Improved Analysis of Rock Mass Geometry using Mathematical and Photogrammetric Methods

A thesis submitted to the University of London
(Imperial College of Science, Technology and Medicine)
in fulfilment of the requirements for the degree of Doctor of Philosophy
in the Faculty of Engineering

by

John Paul Harrison

June 1993

This page intentionally left blank

Abstract

Most engineering in, or on, rock close to the surface of the Earth is affected by the presence of discontinuities. Hence, an accurate and complete description of rock mass geometry in terms of the geometry of these features is essential for economic operations. Here, a review of customary methods for the measurement and analysis of rock mass geometry has been presented, and has revealed that they are deficient in a number of ways.

Detailed reviews of photogrammetry and cluster identification schemes have shown that the integrated use of modern non-metric photogrammetric techniques with fuzzy partitioning algorithms promises improved methods of the analysis of rock mass geometry, and this has led to further development in these fields.

A high-performance analytical plotter, incorporating sophisticated closed-loop control systems, has been designed and constructed to allow determination of photograph coordinates at a resolution appropriate for the measurement of rock mass geometry. This instrument, in conjunction with an original method of introducing high-accuracy photogrammetric control at a rock face, has allowed development of simplified procedures for photogrammetric surveying.

A novel computer program is introduced that implements a sequential application of three fuzzy partitioning algorithms—namely; fuzzy *c*-means, fuzzy covariance and fuzzy maximum likelihood estimation—for the analysis of discontinuity orientation. Exhaustive tests on both synthetic and natural data sets reveals the enhanced form of the analysis over customary methods. This analysis leads to the generation of two new types of hemispherical projection—*viz.* cluster assignment plots and membership level plots—which together allow improved assimilation of rock mass structure.

A report of the application of the new techniques at two quarry sites within the UK demonstrates their overall efficacy. It has been found that photogrammetric methods permits surveying of the entire rock face at an accuracy sufficient for the computation of discontinuity orientation, and the use of fuzzy partitioning algorithms allows an improved analysis of rock mass structure in terms of discontinuity orientation.

Acknowledgements

This enterprise has taken many years to complete and consequently, given the transient nature of the personnel in a University research group, a great number of people have helped me during this time in one way or another; essentially everyone connected with rock mechanics at Imperial College during this period. To all of these erstwhile colleagues—wherever they may be now—I can do no more than issue a blanket 'thank you'.

Of course, there are a number of people without whose help this document would not have seen the light of day. Consequently, I am especially grateful to the following:

- Ed Antonio and Chris Windsor, for stimulation during the early days of my rock mechanics adventure;
- Professors John Archer, Ted Brown and John Hudson, for agreeing to let me start off along this tortuous route;
- Dean Millar for his good humour whilst driving a rain-drenched theodolite, and Frank Coombe for his skill with a universal milling machine;
- Gwen Field for her diligent checking of my prose, whilst tolerating what must have seemed like my insatiable love affair with a computer; and
- Gordon and Celia Harrison for not only starting the whole thing off, but also endless offering of hope when all seemed hopeless.

Part of the funding for this project was obtained in the form of a grant from the University of London Central Research Fund, with the remainder from Imperial College courtesy of Professors Archer, Brown and Hudson.

The field work was undertaken with the permission of Mr C. Peachey, General Manager, Mountsorrel Quarry, and Mr R. Webb, Manager, Tytherington Quarry; this permission is gratefully acknowledged.

Contents

Abstract 3

Acknowledgements 4

Contents 5

List of Figures 13

List of Tables 21

Part I
Introduction and Background

Chapter 1 **Introduction** 25

1.1 Discontinuous Rocks and Rock Engineering 25

1.2 Discontinuity Geometry 27

1.3 Purpose and Structure of this Thesis 31

Chapter 2 **Customary Methods in Discontinuity Geometry** 35

2.1 Measurement of Discontinuity Geometry 36
2.1.1 Discontinuity spacing 36
2.1.2 Discontinuity roughness 37
2.1.3 Discontinuity extent 40
2.1.4 Discontinuity orientation 41
2.1.5 Summary 43

2.2 Analysis of Discontinuity Spacing 44
2.2.1 Mechanistic models 45
2.2.2 Probabilistic techniques 46
2.2.3 Fractals 47
2.2.4 Summary 49

2.3 Analysis of Discontinuity Roughness 49
2.3.1 Data from profile measurements 49
2.3.2 Data from areal measurements 51
2.3.3 Summary 51

2.4 Analysis of Discontinuity Extent 52
2.4.1 Methods based on discontinuity trace length 52
2.4.2 Stochastic modelling of discontinuities 53
2.4.3 Summary 55

2.5 Analysis of Discontinuity Orientation 55
2.5.1 Contouring on the hemisphere 56
2.5.2 Objective functions in clustering algorithms 68

2.5.3 Summary 70
2.6 Synopsis 71

Part II
Fundamental Techniques

Chapter 3 Photogrammetry 77

3.1 An Overview Of Photogrammetry 78
3.1.1 Cameras 79
3.1.2 Viewing photogrammetric images 83
3.1.3 Photogrammetric measurements 85
3.1.4 Field of application 90
3.1.5 Summary 91

3.2 Non-Metric Analytical Photogrammetry 93
3.2.1 Non-metric cameras 93
3.2.2 Data processing 94
3.2.3 Measurement of photograph coordinates 95
3.2.4 Summary 102

3.3 Direct Linear Transform 103
3.3.1 Formulation of the collinearity equations 103
3.3.2 Formulation of the DLT equations 106
3.3.3 Least squares solution of the DLT equations 108
3.3.4 Computation of object space coordinates 111
3.3.5 Summary 112

3.4 Photogrammetry in Rock Mechanics 113
3.4.1 Photogrammetry applied to general rock mechanics problems 113
3.4.2 Photogrammetry applied to rock mass geometry 115
3.4.3 Summary 120

3.5 Synopsis 120

Chapter 4 Cluster Identification Schemes and Fuzzy Sets 123

4.1 Fundamental Clustering Methods 124
4.1.1 Partitioning methods 124
4.1.2 Hierarchical methods 126
4.1.3 Miscellaneous methods 127
4.1.4 Summary 127

4.2 Partitioning Algorithms and Objective Functions 128
4.2.1 Cluster shape and distance measurement 128
4.2.2 The hard *c*-means algorithm 132
4.2.3 Summary 136

4.3 Uncertainty and Fuzzy Sets 136
4.3.1 Deterministic, probabilistic and fuzzy models in engineering rock mechanics 137
4.3.2 Probabilistic *versus* fuzzy models 140
4.3.3 Summary 141

4.4 Fuzzy Partitioning Algorithms 142
4.4.1 The uni-modal algorithm 143
4.4.2 The fuzzy *c*-means algorithm 144
4.4.3 The fuzzy covariance algorithm 148
4.4.4 The fuzzy maximum likelihood estimation algorithm 154
4.4.5 Non-metric fuzzy partitioning algorithms 158
4.4.6 Summary 160

4.5 Validity of Fuzzy Clusters 160
4.5.1 Cluster validity functions 161
4.5.2 Summary 169

4.6 Hardening of Fuzzy Partitions 170
4.6.1 'Nearest maximum membership' hard partition 171
4.6.2 α-Level core of a fuzzy set 172
4.6.3 Summary 174

4.7 Synopsis 174

Part III
Development of an integrated system for surveying and analysis of rock mass geometry

Chapter 5 Non-Metric Photogrammetry Applied to Measurements of Rock Mass Geometry 179

5.1 An Analysis Of The Computation Of Discontinuity Orientation From Measured Photograph Coordinates 180
5.1.1 Analysis using total differential error 180
5.1.2 Numerical analysis 183
5.1.3 Summary 199

5.2 Design and Construction of an Analytical Plotter 200
5.2.1 General specification 200
5.2.2 Mechanical components and construction 202
5.2.3 Electrical and electronic components and construction 207
5.2.4 Control algorithms and software 211
5.2.5 Summary 213

5.3 Procedures for Introducing Photogrammetric Control 215
5.3.1 Survey configuration 216
5.3.2 Survey adjustment 218
5.3.3 Summary 224

5.4 Enhancements to the DLT Method of Photogrammetric Data Reduction 231
5.4.1 Real-time implementation of the DLT 231
5.4.2 Adaptation to control points of poor spatial distribution 233
5.4.3 Summary 235

5.5 Accuracy Test Of The Photogrammetric System 235
5.5.1 Operation of the modified DLT program 236
5.5.2 Coordination accuracy of the photogrammetric system 237
5.5.3 Summary 242

5.6 Synopsis 242

Chapter 6 Analysis of Rock Mass Geometry Using Fuzzy Partitioning Algorithms 245

6.1 Application of Fuzzy Objective Function Algorithms to the Analysis of Discontinuity Orientation Data 246
6.1.1 Geometry of orientation data 246
6.1.2 Distance measures 250
6.1.3 Summary 252

6.2 A Clustering Scheme based on Fuzzy Partitioning Algorithms for Discontinuity Orientation Data 253
6.2.1 Choice of the FCM, FCV and FMLE algorithms 253
6.2.2 Detail of the algorithms 255
6.2.3 Special procedures 255
6.2.4 Cluster validity functions incorporating sharpening 263
6.2.5 Summary 266

6.3 Investigations into the Effectiveness of FADD for Partitioning Orientation Data 266
6.3.1 Empirical verification of FADD 267
6.3.2 Algorithm convergence criteria 279
6.3.3 Assessment of cluster validity functions applied to sharpened partitions 282
6.3.4 Summary 303

6.4 Graphical Presentation of Partitioning Results for Orientation Data 305
6.4.1 Cluster assignment plots 305
6.4.2 Membership level plots 308
6.4.3 Summary 312
6.5 Synopsis 312

Chapter 7 Application of the New Techniques 315

7.1 Equipment Used and Procedures Followed 315
7.1.1 Field operations 316
7.1.2 Office procedures 319
7.1.3 Summary 321
7.2 Application at a Limestone Rock Face 322
7.2.1 Photogrammetric survey 323
7.2.2 Compass survey 329
7.2.3 Discussion 331
7.3 Application at a Granodiorite Rock Face 333
7.3.1 Photogrammetric survey 333
7.3.2 Compass survey 339
7.3.3 Discussion 341
7.4 Synopsis 343

Part IV
Topics for Further Research & Conclusions

Chapter 8 Topics for Further Research 347

8.1 Further Development of Non-Metric Photogrammetry 347
8.1.1 Optimisation of the DLT 347
8.1.2 Total least squares applied to the solution of the DLT 348
8.1.3 Absolute performance of the analytical plotter 349
8.1.4 Automation of the measurement process 350
8.1.5 Application of digital photogrammetry 350
8.1.6 Further study of the photogrammetric configuration at a rock face 351
8.1.7 Simplified methods of introducing photogrammetric control 351
8.2 Further Development of Fuzzy Partitioning Algorithms 352
8.2.1 The use of Mahalanobis angles 353
8.2.2 Further study of cluster validity functions 353

8.2.3 Other partitioning algorithms 354
8.2.4 Interpolation of partitioning results 354
8.2.5 Characterising fuzzy clusters 355
8.2.6 Higher dimensional clustering 356
8.2.7 Non-metric methods 356

8.3 Advanced Study of Rock Mass Geometry 357
8.3.1 Determination of spacing, extent and roughness from photographs 357
8.3.2 Interrelations between extent, orientation and spacing of discontinuity surfaces 358
8.3.3 The analysis of scanline techniques on rock faces 358
8.3.4 Visualisation techniques for complete rock mass geometry 359
8.3.5 Enhanced stability analyses 359

8.4 Synopsis 360

Chapter 9 Conclusions 363

Part V
Supporting Material

Chapter 10 References 371

Appendix A Analytical Plotter Construction Details 383

A.1 General 383

A.2 Closed-Loop Control System 384

A.3 HCTL-1000 General Purpose Motion Control IC 386
A.3.1 Internal operation 386
A.3.2 Host computer interface 390
A.3.3 Motor driver, rotary encoder and limit interfaces 390

A.4 ISA-bus Interface Circuit 392

A.5 Stepping Motor Switchmode Constant-Current Chopper Driver 403

A.6 Miscellaneous Electrical Issues 403
A.6.1 Optical limit switches 404
A.6.2 Floating spot brightness control 405
A.6.3 Repeater circuit 406
A.6.4 Power supply, cabling and connectors 407

A.7 HCTL-1000 Programming 408

A.8 Mechanical Aspects 411

A.8.1Components 411
A.8.2Fabrication and construction 415

Appendix B Program APCS: Analytical Plotter Control System 421

B.1 Structure of APCS 421

B.2 Listing of program APCS 425

B.3 Supporting Assembly Language Routines 454

Appendix C Program MDLT: Modified Direct Linear Transformation 465

C.1 Structure of MDLT 465

C.2 Listing of program MDLT 467

Appendix D Program FADD: Fuzzy Analysis of Discontinuity Data 487

D.1 Operation of FADD 487

D.2 Input and Output Files 489

D.3 Listing of Program FADD 490

This page intentionally left blank

List of Figures

Figure 1.01 A typical rock mass as exposed in a quarry face; the dumptruck gives scale 26
Figure 1.02 Interdependence between discontinuity geometry parameters 29
Figure 1.03 Outline of the thesis 33
Figure 2.01 Discontinuity distance measurements taken at an exposure (from Attewell and Farmer [9]) 37
Figure 2.02 Interpretation of discontinuity spacing from borehole core (from Brown [22]) 38
Figure 2.03 Measurement of discontinuity surface roughness by means of laser profiling (from Priest [109]) 39
Figure 2.04 Window sampling at a rock exposure (from Priest [109]) 40
Figure 2.05 Numerical simulation of discontinuity networks under conditions of plane stress (from Renshaw and Pollard [113]) 45
Figure 2.06 Discontinuity spacing histogram for scanlines in chalk (from Priest and Hudson [110]) 46
Figure 2.07 Fractal fracturing in plane stress (from Skjeltorp and Meakin [128]) 48
Figure 2.08 Scale effect in roughness determination from linear profiles (from Bandis [11]) 50
Figure 2.09 Analysis of three-dimensional roughness measurements (from Brown [22]) 51
Figure 2.10 Complete, semi- and censored semi-trace lengths 53
Figure 2.11 Examples of stochastic models of rock mass geometry (from Dershowitz and Einstein [34]) 54
Figure 2.12 Program PATCH: division of the hemisphere into 100 equal area spherical quadrilaterals (from Mahtab *et alia* [86]) 57
Figure 2.13 Program PATCH: equal area upper hemispherical projection of porphyry copper data (from Mahtab *et alia* [86]) 58
Figure 2.14 Discrepancy in clustering due to the use of counting circles and Schmidt contouring (from Bayer [14]) 60

Figure 2.15 Schmidt and Fisher contouring, as implemented in the program DIPS (from DIPS user's manual [61]) 61
Figure 2.16 Porphyry copper data: upper hemispherical equal area projections of Schmidt contouring (data from Mahtab *et alia* [86]) 63
Figure 2.17 Porphyry copper data: upper hemispherical equal area projections of Fisher contouring (data from Mahtab *et alia* [86]) 65
Figure 2.18 Spherical counting cell (from Shanley and Mahtab [124]) 69
Figure 3.01 Principal distance and principal point for a perfect pin-hole camera 80
Figure 3.02 Radial and decentering lens distortions 81
Figure 3.03 Binocular vision and stereoscopic photographs 84
Figure 3.04 The mirror stereoscope 85
Figure 3.05 Geometry of stereoscopic photographs 87
Figure 3.06 Causes of y-parallax in stereoscopic photographs 89
Figure 3.07 Interactions between equipment and techniques for aerial photogrammetry 92
Figure 3.08 Configurations of analytical photogrammetric instruments (after Petrie [104]) 96
Figure 3.09 Systematic position error in a digitising tablet (from Carson [24]) 98
Figure 3.10 Photogrammetric systems for defining camera orientation 105
Figure 3.11 Roughness profiles for a rock discontinuity surface (from Miller *et alia* [96]) 116
Figure 3.12 Object space control system for non-metric photogrammetry applied to rock mass geometry, and typical discontinuity orientation results (from Brandow *et alia* [21]) 117
Figure 3.13 Stereoscopic non-metric camera system (from Hagan [55]) 119
Figure 4.01 Single-link agglomerative clustering (data from Dunn and Everitt [37]) 126
Figure 4.02 Result of partitioning a data set in $\mathbb{R}^2$ with the min$|\mathbf{W}|$ algorithm (from Everitt [39]) 130
Figure 4.03 Butterfly data: hard c-means partitioning with three different initial partitions 135

Figure 4.04 The uni-modal fuzzy clustering technique (from Gitman and Levine [47]) 144
Figure 4.05 Illustration using the butterfly data of fuzziness as a function of weighting exponent m (after Bezdek [15]) 148
Figure 4.06 'Cross' data: application of the fuzzy c-means and the fuzzy covariance algorithms (after Gustafson and Kessel [54]) 152
Figure 4.07 Data consisting of two ill-defined subgroups, partitioned using the FCM and FMLE algorithms (from Gath and Geva [46]) 157
Figure 4.08 Geometry of the separation coefficient 166
Figure 4.09 Geometry of the separation index 167
Figure 4.10 Butterfly data: nearest maximum membership hard partition 172
Figure 4.11 α-Level hard cores associated with the butterfly data 173
Figure 5.01 Configuration of a simplified representation of a discontinuity surface for the purposes of analysis of error in orientation 184
Figure 5.02 Definitions of triad and discontinuity orientations 185
Figure 5.03 Definition of possible discontinuity plane, tangential to error spheres associated with discontinuity triad points 186
Figure 5.04 Geometry of the error analysis 191
Figure 5.05 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=0.1m, inclination=80°, object-base distance=15m 193
Figure 5.06 Error in discontinuity orientation determination as a function of distance from camera base and height above plane of camera base: image interpretation error=0.04mm, diameter=0.1m, inclination=80°, distance from camera base centreline=5m 194
Figure 5.07 Error in discontinuity orientation determination as a function of discontinuity orientation and image interpretation error: discontinuity diameter=0.1m, object-base distance=15m, height above plane of camera base=10m, distance from camera base centreline=5m 195

Figure 5.08 Error in discontinuity orientation determination as a function of distance from camera base and image interpretation error: diameter=0.1m, inclination=80°, height above plane of camera base=10m, distance from camera base centreline=5m 196
Figure 5.09 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=0.1m, inclination=40°, object-base distance=11m 196
Figure 5.10 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=0.5m, inclination=40°, object-base distance=11m 197
Figure 5.11 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=1.0m, inclination=40°, object-base distance=11m 197
Figure 5.12 Deviation between angular error computed using (5.18) and angular error derived numerically 199
Figure 5.13 Detail of stepping motor and lead screw mountings and coupling 205
Figure 5.14 Coupling between lead screw and carriage 206
Figure 5.15 Customary and independent photograph configurations 206
Figure 5.16 HEDS5540 rotary encoder mounted on a lead screw 208
Figure 5.17 Typical HOA2001 optical limit switch and wiper 209
Figure 5.18 Component side of wire-wrap interface board 210
Figure 5.19 Stepping motor driver board 211
Figure 5.20 The complete analytical plotter 215
Figure 5.21 Cable control network 217
Figure 5.22 Ground control network 218
Figure 5.23 Elements of a survey network 220
Figure 5.24 Example Ground Control Data Reduction worksheet 225
Figure 5.25 Example Ground Control Coordinate Adjustment worksheet 226

Figure 5.26 Example Marker Survey Data Reduction worksheet 227
Figure 5.27 Example worksheet Marker Coordinate Adjustment (sheet 1) 228
Figure 5.28 Example worksheet Marker Coordinate Adjustment (sheet 2) 229
Figure 5.29 Example worksheet Marker Coordinate Adjustment (sheet 3) 230
Figure 6.01 Spherical polar coordinates in Riemannian geometry 249
Figure 6.02 Illustration of antipodal vectors 260
Figure 6.03 Test for antipodal vector and computation of associated Euclidean distance 261
Figure 6.04 FCM partitioning of data containing four clusters drawn from Fisher distributions 270
Figure 6.05 FCM partition of test data containing an anisotropic cluster 273
Figure 6.06 FCV partition of test data containing an anisotropic cluster 274
Figure 6.07 Classification of clusters in orientation data: (a) typical patterns for observed values of eigenvalues (from Fisher *et alia* [44]); (b) results of FCV algorithm applied to test data 276
Figure 6.08 Anisotropic test data: Convergence of partitioning iteration with m=2. (a) and (b): FCM algorithm, (c): FCV algorithm 281
Figure 6.09 Porphyry data: partition coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 285
Figure 6.10 Porphyry data: normalised partition coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 287
Figure 6.11 Porphyry data: partition entropy for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 289
Figure 6.12 Porphyry data: proportion exponent for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 291
Figure 6.13 Porphyry data: separation coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 293
Figure 6.14 Porphyry data: separation index for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 295
Figure 6.15 Porphyry data: modified partition coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 297

Figure 6.16 Porphyry data: fuzzy hypervolume for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 299
Figure 6.17 Porphyry data: partition density for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5 301
Figure 6.18 Porphyry data: cluster assignment plot for FCM partitioning with m=2 and c=3 306
Figure 6.19 Porphyry data: cluster assignment plot for FCV partitioning with m=2 and c=3 307
Figure 6.20 Porphyry data: membership level plot for FCM partitioning with m=2 and c=3 310
Figure 6.21 Porphyry data: membership level plot for FCV partitioning with m=2 and c=3 311
Figure 7.01 Typical cable markers 316
Figure 7.02 Weights used for tensioning control cables 317
Figure 7.03 Limestone face photogrammetric survey: control marker configuration and relative coordinates 323
Figure 7.04 Limestone face photogrammetric survey: left hand stereophotograph 324
Figure 7.05 Limestone face photogrammetric survey: right hand stereophotograph 325
Figure 7.06 Limestone face photogrammetric survey: discontinuity triangles and control points 326
Figure 7.07 Limestone face photogrammetric survey: cluster assignment plot 328
Figure 7.08 Limestone face compass survey: equal angle contoured pole plot 329
Figure 7.09 Limestone face compass survey: membership level plot 330
Figure 7.10 Granodiorite face photogrammetric survey: control marker configuration and relative coordinates 333
Figure 7.11 Granodiorite face photogrammetric survey: left hand stereophotograph 334
Figure 7.12 Granodiorite face photogrammetric survey: right hand stereophotograph 335
Figure 7.13 Granodiorite face photogrammetric survey: discontinuity triangles and control points 336
Figure 7.14 Granodiorite face photogrammetric survey: cluster assignment plot 338

Figure 7.15 Granodiorite face compass survey: equal angle contoured pole plot 339
Figure 7.16 Granodiorite face compass survey: membership level plot 340
Figure A.01 Block diagram of closed-loop control system 384
Figure A.02 HCTL-1000 internal block diagram (after Hewlett-Packard [60]) 387
Figure A.03 HCTL-1000 register block diagram (after Hewlett-Packard [60]) 388
Figure A.04 HCTL-1000 operation flowchart (after Hewlett-Packard [60]) 389
Figure A.05 HCTL-1000 interface timing diagrams (after Hewlett-Packard [60]) 391
Figure A.06 Interface circuit block diagram 394
Figure A.07 Interface board schematic: sheet 1 of 2 400
Figure A.08 Interface board schematic: sheet 2 of 2 401
Figure A.09 Interface circuit timing diagrams 402
Figure A.10 Schematic of stepping motor switchmode constant-current chopper driver 404
Figure A.11 Constant-current, pulse width modulated control of stepping motor windings 405
Figure A.12 Power supply cabinet 406
Figure A.13 L200-based floating spot brightness control 406
Figure A.14 Analytical plotter terminal board 407
Figure A.15 Stepping motor driving sequences using HCTL-1000 commutator port 409
Figure A.16 Typical OFFSET optimisation curves 410
Figure A.17 Recirculating ball elements and single- and twin-track linear bearings 413
Figure A.18 Analytical plotter: base plate general arrangement 417
Figure A.19 Analytical plotter: sectioned general arrangements 418
Figure A.20 Analytical plotter: general arrangement of left carriage plate 419
Figure D.1 Flowchart of program FADD. 488

This page intentionally left blank

List of Tables

Table 2.01 Porphyry copper orientation data (formatted as dip direction, dip angle) (from Mahtab *et alia* [86]) 59

Table 3.01 Advantages and disadvantages of non-metric cameras with respect to metric cameras (after Karara [68]) 82

Table 4.01 Cross data: sample covariance matrices and the fuzzy scatter matrices (after Bezdek [15]) 153

Table 4.02 Cross data: eigenvalues and corresponding eigenvectors for covariance and fuzzy scatter matrices (after Bezdek [15]) 153

Table 4.03 Basic relations between elements of the FCV algorithm and Bayesian classifiers 155

Table 5.01 Linear motion drive and measurement systems 203

Table 5.02 Photograph movement procedures 212

Table 5.03 Object space coordinates and measured photograph coordinates of the test control points 237

Table 5.04 Control point object space coordinates, computed using measured photograph coordinates 238

Table 5.05 Interior and exterior camera orientation parameters for the two camera stations used in the coordination tests 239

Table 5.06 Error in y-parallax determination 240

Table 5.07 Error in position determination resulting from the use of photograph coordinates computed using a parallax-free model 241

Table 6.01 The FCM fuzzy partitioning algorithm 256

Table 6.02 The FCV fuzzy partitioning algorithm 257

Table 6.03 The FMLE fuzzy partitioning algorithm 258

Table 6.04 Test data comprising four isotropic clusters 269

Table 6.05 Comparison between prototypical and computed cluster centroids for isotropic test data 271

Table 6.06 Test data containing an anisotropic cluster 272

Table 6.07 Comparison between prototypical and computed cluster centroids for the test data containing an anisotropic cluster 275

Table 6.08 Eigenvalues and corresponding classification coordinates for anisotropic test data 276

Table 6.09 Porphyry data: comparison of cluster orientation and size for five different cluster identification schemes 277
Table 6.10 Porphyry data: cluster orientations for FCV and FMLE partitioning 278
Table 6.11 Empirical ranking of cluster validity functions applied to orientation data 303
Table 6.12 Mapping scheme for membership level plots 309
Table 7.01 Limestone face photogrammetric survey: discontinuity surface coordinates 327
Table 7.02 Limestone face photogrammetric survey: discontinuity orientations 328
Table 7.03 Limestone face compass survey: discontinuity orientations 330
Table 7.04 Limestone face survey: cluster validity functions 332
Table 7.05 Granodiorite face photogrammetric survey: discontinuity surface coordinates 337
Table 7.06 Granodiorite face photogrammetric survey: discontinuity orientations 338
Table 7.07 Granodiorite face compass survey: discontinuity orientations 340
Table 7.08 Granodiorite face survey: cluster validity functions 342
Table B.01 Structure of program APCS: sheet 1 422
Table B.02 Structure of program APCS: sheet 2 423
Table C.01 Structure of program MDLT 466

Part I
Introduction and Background

This page intentionally left blank

1 Introduction

This thesis represents a comprehensive report on fundamental research undertaken by the Author. This research was aimed at the development of improved methods for the measurement and analysis of the geometry of discontinuities in rock masses.

This Chapter serves to both introduce the basic problem and identify the context of the research, as well as explain the outline of the thesis. To this end, it is composed of three Sections: the first briefly describes the elements of rock as an engineering material, showing how its behaviour is influenced by the presence of fractures within the rock itself; the second goes on to consider the geometrical attributes of these fractures, and highlights why the methods currently used for their measurement and characterisation could usefully be improved; and the third Section presents an outline of the structure of the thesis.

1.1 Discontinuous Rocks and Rock Engineering

The *in situ* **rock mass** is the basic material of rock engineering. In order to predict the engineering behaviour of a rock mass, the various elements that comprise it must be quantified, and to this end engineers have both identified these elements and developed methods of measurement, subsequent analysis and characterisation for them.

Commonly, rock masses are regarded as being formed from two constituent elements—blocks of **intact rock**, surrounded by mechanical breaks known as **discontinuities**. This assemblage is acted upon by environmental factors such as ***in situ* stress** and **groundwater**, with the whole forming the engineering material. A typical rock mass as it is exposed in an engineering project is shown in Figure 1.01, and this photograph serves to illustrate the nature of the basic material.



Figure 1.01 A typical rock mass as exposed in a quarry face; the dumptruck gives scale.

The discontinuities that exist in rocks close to the Earth's surface may be observed on a variety of scales, from the very small (*e.g.* intra-crystal) to the very large (*e.g.* tectonic plate boundaries). Although such a vast range exists, when one observes rock exposed in a rock face (say at a cliff or a quarry), it is usually fracturing occurring on the scale of tens of centimetres to a few metres that is immediately apparent, as Figure 1.01 dramatically demonstrates. It is this fracturing that has a profound effect on engineering in rock, for the simple reason that, on the scale of most structures engineered in or on rock, these fractures alter the fundamental mechanical nature of the rock material from that of an essentially isotropic, homogenous and continuous medium to the converse: an anisotropic, inhomogenous and discontinuous medium.

Given the world-wide distribution of engineering in rock masses, it is surprising that a great number of engineers are not aware, in an explicit fashion, of the effects each of these three attributes (*i.e.* anisotropy, inhomogeneity, discontinuousness) of rock masses may have on the engineering problem. The popular engineering press abounds with reports of operations costing far more and taking far longer to complete than expected, and yet the reasons for such problems can often be simply explained in terms of the manifestation of these basic attributes of a rock mass.

The reason for the regular appearance of reports of such problems may arise from the methods used to characterise rock masses. Currently, rock mass

characterisation is undertaken on the basis of measurements and characterisation of individual components (*e.g.* intact rock, discontinuities, *in situ* stress and groundwater), which are commonly regarded as being independent. Indeed, the fact that the customary characterisation methods for these components are embodied in separate published documents [22][§] exemplifies this, widely regarded, independence.

One possible means of improving the perception of anisotropy, inhomogeneity and discontinuousness in rock masses is through the development and application of methods that explicitly consider these phenomena in a linked manner. Although work is being undertaken in this direction [62], it is not yet sufficiently advanced for general use. As an adjunct to this work, it is important to ensure that methods are available that allow both for the complete measurement and analysis of these phenomena. Thus, given the central rôle played by discontinuities in the manifestation of these phenomena, it is therefore appropriate to study in detail the customary methods applied to the measurement and analysis of their physical attributes, to ensure that such adequate techniques exist.

A detailed study of all of the physical attributes of discontinuities is beyond the scope of a work such as this, and as a consequence it is necessary to identify some coherent fraction of them in order to reduce the problem to manageable proportions. Customarily, the physical attributes of discontinuities are regarded as being either mechanical or geometrical in nature, with properties such as shear strength belonging to the former group and properties such as location and orientation belonging to the latter. This customary division is convenient in the context of this work, with either division forming a consistent whole worthy of study; consequently, discontinuity geometry has been adopted for further consideration.

1.2 Discontinuity Geometry

Studying the photograph of the quarry face in Figure 1.01 provokes a number of questions about the geometry of the rock mass. For example: how big are the individual rock blocks; what shape are they; what do their surfaces look like; what patterns of fracturing are apparent; and what variation in geometry exists? It is to help address these questions that various suggested methods—

§ References are arranged alphabetically in Chapter 9 and numbered sequentially. They are then identified in the text by this number, surrounded by a pair of brackets.

for example, those proposed by the International Society for Rock Mechanics (ISRM) [22]—have been developed.

The adoption of these suggested methods generally involves the geotechnical engineer or geologist in making measurements of a number of parameters that are considered to describe discontinuities. For example, the ISRM Suggested Methods identify ten such parameters, namely: orientation; spacing; extent (or persistence); roughness; wall strength; aperture; filling; seepage; number of sets; and block size. Limiting ourselves to a consideration of geometrical parameters only, we can disregard wall strength, filling and seepage on the basis that they are either mechanical or environmental parameters. Furthermore, of the seven geometrical parameters, it can be argued that aperture, number of sets and block size are not independent parameters, but are functions of orientation, spacing, extent and roughness. Thus, discontinuity geometry in rock masses may be regarded as a four-parameter phenomenon, and to adequately characterise it we need only to measure these four principal parameters.

Indeed, it is such an approach that is customarily followed in geotechnical engineering. In order to characterise a rock mass, a geotechnical engineer or geologist will measure four geometrical attributes: the spacing between adjacent discontinuities; the orientation of individual discontinuities; the lengths of discontinuity traces exposed on a rock face (as a way of determining discontinuity extent); and the roughness of discontinuity surfaces themselves.

Consequently, measurements of these four principal attributes represent the basis of most, if not all, geotechnical surveying operations that seek to gather data for subsequent analysis of rock mass geometry. However, these geometrical attributes of orientation, spacing, extent, and roughness are regarded as independent parameters when describing rock masses. Now, whilst this independence may be fundamentally correct for individual discontinuities, it can be argued that, when these attributes are measured at a rock exposure (for example, either a rock face or a length of borehole core), the opposite is true—they are interdependent. An example of how our interpretation of the geometry of a rock mass may be affected by this interdependence is shown in the matrix presented in Figure 1.02

In this matrix, the leading diagonal contains the 'independent' (or principal) geometrical attributes, with each off-diagonal (or secondary) element illustrating the effect the corresponding two principal attributes have on each other. The flow of 'cause and effect' is clockwise in this matrix: for example, the

SPACING	Close spacing generates small blocks, the faces of which may be too small to display large scale roughness	If discontinuities in one set tend to terminate at discontinuities in another set, then the spacing of those in the second set will dictate the extent of those in the first	Measurements of orientation may be imprecise and inaccurate in a highly fractured rock mass, due to only surfaces of limited extent being available
Spacing between adjacent discontinuities is not uniquely defined for rough surfaces	**ROUGHNESS**	Roughness can merge into waviness and non-planarity. Is extent measured along a mean plane, along the actual surface, or some projection of it?	Depending on the relative scales of roughness and the instrument used for measuring orientation, roughness may be identified as a spread of orientations
In a set of discontinuities possessing finite length, spacing values change at terminations	Extensive discontinuities may have been sheared and hence smoothed (*e.g.* slickenslided fault planes)	**EXTENT**	Discontinuities of small extent may be unrelated in orientation to the major rock mass structure
Measured spacings (along a scanline, for example) are related to the orientation of the discontinuities relative to the survey line by the function $\lambda_s = \sum_{i=1}^{n} \lambda_i \cos\theta_i$	If a failure plane is defined by a series of non-parallel intersecting discontinuities, it will have an effective roughness due to its step-like form	As two intersecting discontinuity sets move away from being orthogonal, so extent (exhibited as the size of block faces) will increase	**ORIENTATION**

Figure 1.02 Interdependence between discontinuity geometry parameters.

top right hand element is the effect of spacing on orientation, the bottom left hand element is the effect of orientation on spacing. The descriptions of the secondary elements have been chosen to reflect how the interdependence of the principal attributes would be manifested at an exposure of the rock mass.

A study of this matrix shows, quite clearly, that customary measurements of rock mass geometry do no more than reveal the values of these secondary elements. Consequently, any attempt at measuring the principal attributes at some sort of exposure is bound to failure—unless either a complete physical dissection of the rock mass is undertaken, or some means of extracting the values of the principal attributes from the measurements of the secondary elements is found. Of course, physical dissection is usually impossible, which

means that it is the latter route that must be pursued. Hence, having determined that a sophisticated method of analysing the data resulting from measurements of geometrical attributes is required, it is fitting to ask whether such a method is used in practice.

At present, rock masses are generally characterised through reference to empirical schemes, such as those proposed in the ISRM Suggested Methods [22]. Although more sophisticated techniques have been developed, they often remain unused by practising engineers. It is tempting to suggest that this may be due to human nature: if a method is either not universally and easily applicable, or if it does not give predictions that are subsequently found to be in agreement with prevailing conditions, or if there is no tangible benefit resulting from its application, then it will fall into disuse. Thus, although consideration of a matrix such as that shown in Figure 1.02 indicates that the development of more sophisticated analytical techniques is warranted, care must be taken to ensure their suitability for general use.

Of course, the sophisticated analysis of rock mass geometry cannot proceed without adequate geometrical data, and so we must also determine whether the equipment commonly used for acquiring such data is adequate. Usually, measurements of the principal attributes of orientation, spacing, extent, and roughness are undertaken using hand-held equipment such as geological compasses, measuring tapes and profile gauges. As a consequence, the tendency is for only simple measurements of geometry to be made at easily and safely accessible locations on a rock exposure, with the resulting data being limited to simple statistical manipulations and graphical analysis. Thus, it can be argued that the development of sophisticated analysis is, in part, hampered by the limited nature of the geometrical data, and that enhanced analytical techniques will only appear when the data exist to be analysed.

The need for improved measurement and analytical techniques cannot be overstated, as the following points show.

(i) In recent years great advances have been made in the fundamental understanding and prediction of the mechanical behaviour of discontinuous rock masses, through developments in both analytical and numerical methods. However, the confident application of these methods is probably hampered by a lack of good geometrical data.

(ii) Manual measurement of discontinuities is laborious, prone to error due to inaccessibility, time consuming and sometimes dangerous. In addition, the

lack of a permanent record from which alternative assessments may be made can be a serious drawback in certain circumstances.

The importance of those aspects listed in (ii) must not be underestimated. Figure 1.01 demonstrates that a rock face some 20m high is inaccessible to a human for the purposes of physical measurement over 90% of its area. Furthermore, if it appears in the production face of a quarry it will probably be unstable and ephemeral, and if its complexity approaches that shown in Figure 1.01, measurement and analysis by current means is all but intractable.

It is the lack of sophisticated, and yet easily applicable, techniques for the measurement and analysis of discontinuity geometry that has prompted the line of research reported herein. This research has been concerned with the identification and appropriate development of techniques and equipment such that an integrated system for the measurement and analysis of rock mass geometrical attributes may be created. In order that the resulting system may be acceptables to practising engineers, a number of constraints have been applied, and these are outlined below.

(i) The system should be universally applicable: it should be able to acquire and analyse geometrical data from all rock masses and, within reason, all environments (*i.e.* above and below ground, and at exposures of limited life).

(ii) The system should be straightforward to operate. Specialist personnel, other than geotechnical engineers, must not be required.

(iii) Any equipment required should be of reasonable cost and easily obtainable.

(iv) Speed and quality of analysis should be in advance of that currently available.

These constraints have been used to guide the research, and are referred to throughout this work.

1.3 Purpose and Structure of this Thesis

This thesis is a report of research undertaken by the Author, the aim of which was to develop an original and coherent system for characterising the geometry of discontinuous rock masses through the development and application of improved methods of data acquisition and analysis. Consequently, following a detailed review of the customary methods used in the measurement and analysis of discontinuity geometry, the path of the research bifurcated: one

branch concentrated on measurement equipment and procedures, and the other on analytical techniques. These two paths finally converged with the creation of an integrated system for the measurement and analysis of rock mass geometry. A consideration of this structure indicates a number of distinct phases to the work, in particular the three stages of assimilation, development and application. Thus, this thesis is arranged in Parts and Chapters such that these separate strands and phases are presented in a structured fashion. The layout of the thesis is shown in Figure 1.03.

As Figure 1.03 shows, this document is organised into five main Parts: introduction and background; fundamental techniques; development and application of an integrated system for surveying and analysis of rock mass geometry; topics for further research and conclusions; and supporting material. The second and third parts represent the bifurcation in the path of the research, with separate Chapters in each part relating to the measurement and analysis of discontinuity geometry.

The introductory work in Part I is concerned with a review of customary methods in the measurement and analysis of rock mass geometry. On the strength of this review, the further study of both photogrammetry and the mathematics of cluster analysis is suggested, as these fields offer appropriate techniques for the measurement and analysis of discontinuity geometry. Thus, Part II considers these topics in depth, with the result that non-metric analytical photogrammetry and cluster identification schemes based on fuzzy partitioning algorithms are identified as candidates for development in the context of rock mass geometry.

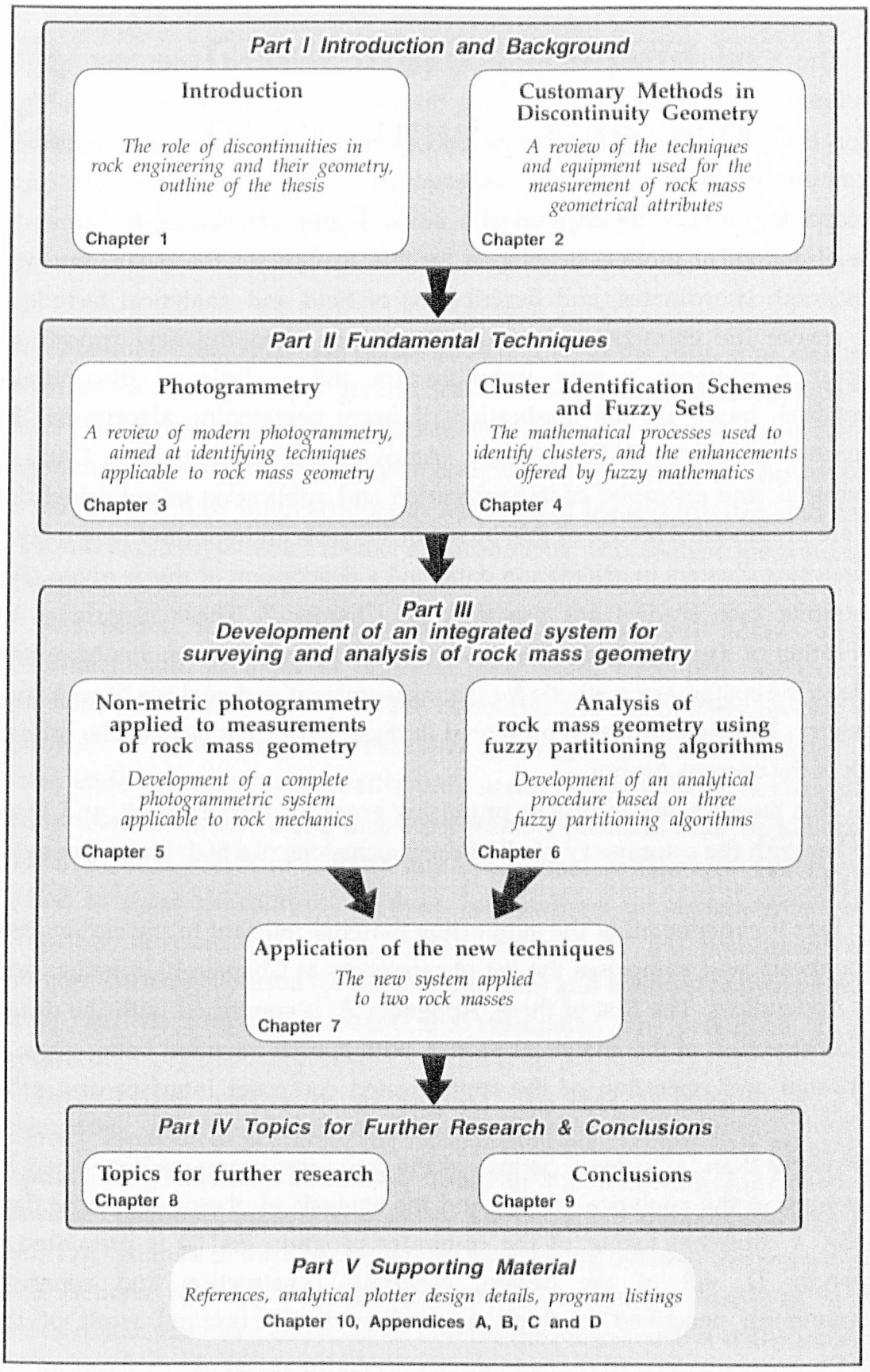


Figure 1.03 Outline of the thesis.

The Chapters comprising Part III are concerned with the original development work undertaken during this research. Chapter 5 consists of three major elements: an original analysis of the errors to be expected when discontinuity orientation is determined using measurements from stereophotographs; a description of a novel, high-performance, instrument (a type of analytical plotter) developed by this Author for the measurement of photograph coordinates; and descriptions of field and analytical techniques that ensure the extraction of maximal accuracy from the new procedures. Chapter 6 presents a new technique for the analysis of discontinuity orientation, based on the application of fuzzy partitioning algorithms. The computer program FADD (**F**uzzy **A**nalysis of **D**iscontinuity **D**ata) is introduced, and examples of its verification and application to published data sets are presented. The use of this program suggests an important new method of analysing clusters in orientation data, and a description of this is given. Two illustrative case studies are presented in Chapter 7. These reports of the application of the new integrated system show how improvements have been made to the customary methods for the measurement and analysis of rock mass geometry. The developments presented in these three Chapters are the original work of the current Author.

This research has indicated promising areas for further work, and these, together with the a summary of all of the conclusions reached, are presented in Part IV.

Part V contains all of the supporting material relevant to the earlier Parts of the thesis, and comprises the list of references at Chapter 10, together with four Appendices. The first of these, Appendix A, is concerned with the design and construction of the analytical plotter, with special attention being given to the design and operation of the sophisticated computer interface and other electronic circuits that provide closed-loop control of the instrument. Appendices B and C contain listings of the computer programs developed for the control of the analytical plotter and the analysis of photogrammetric data. Finally, a complete listing of the computer program FADD is presented in Appendix D. All of the design, electronic construction and computer programming described in these Appendices is the original work of this Author.

2 Customary Methods in Discontinuity Geometry

In the previous Chapter, it was demonstrated how, in the context of rock mass geometry, the four principal geometrical attributes of discontinuity spacing, surface roughness, extent and orientation should not be regarded as independent. Instead, we should recognise that any geometrical measurements that we make will be influenced by at least two of these attributes, and hence we should explicitly attempt to take measurements, and analyse the results, on this basis.

The search for equipment and techniques that will allow explicit measurement and analysis of multiple geometrical attributes can valuably begin with a review of those practices currently, and customarily, in use; it is important to determine whether any such practices do account for interdependence of geometrical attributes, and if not, whether it is possible to develop them to do so. Failing this, other suitable practices must be identified. In addition, it is useful to identify any operational deficiencies that may be attached to these procedures, as these can perhaps be circumvented with appropriate development. Thus, critical reviews of the equipment and techniques customarily used for the measurement and analysis of discontinuity spacing, surface roughness, extent and orientation are presented in this Chapter.

This Chapter consists of six Sections. The first of these reviews the customary methods of acquiring the fundamental geometrical data relating to the principal geometrical attributes. Then, in the following four Sections, reviews of customary analytical practices are presented sequentially. The last Section is a synopsis of the Chapter which, by drawing together the work of the earlier Sections, serves to highlight the various deficiencies that exist in the methods of measurement and analysis of these attributes, such that a promising route for research can be identified.

2.1 Measurement of Discontinuity Geometry

In the ISRM Suggested Methods for the Quantitative Description of Discontinuities [22], comprehensive accounts are given on how to determine the ten fundamental discontinuity parameters of orientation, spacing, persistence, roughness, wall strength, aperture, filling, seepage, number of sets and block size. Only the first four are appropriate here, as it can be argued that the others either are not geometrical descriptors of the entire rock mass attributes (*e.g.* wall strength, aperture, filling, seepage), or are a function of other attributes (*e.g.* number of sets and block size).

Consequently, customary measurement techniques for each of the four attributes of orientation, spacing, persistence and roughness are briefly described in the following four sub-Sections.

2.1.1 Discontinuity spacing

The most satisfactory method—in terms of reducing subjectivity—for measuring discontinuity spacing is to use either scanline or window sampling strategies at a rock exposure. A complete discourse on the modern use of these techniques is given by Priest [109]. Window sampling is most commonly used for measuring discontinuity extent, and so consideration of this technique is deferred until sub-Section 2.1.3.

Scanline techniques require measurements to be taken of discontinuity location relative to a long straight tape measure affixed to the rock surface at an exposure. If borehole core is being used, then the core itself represents a scanline through the rock mass, and the same measurement techniques are employed as when the scanline is at an exposure. A complete discussion on the mechanics of undertaking a scanline survey is presented by Attewell and Farmer [9]; a schematic of a scanline, illustrating the various measurements that may be taken, is shown in Figure 2.01.

Spacing measurements on borehole core are taken in a similar fashion to those on an exposure, although the added complication exists of where within the cross-section of the core the measurement should be taken. Figure 2.02 illustrates three interpretations: tip-to-tip, centre line and fully circular. The ISRM Suggested Method concludes that centre line length is most appropriate, and is therefore the recommended measurement.

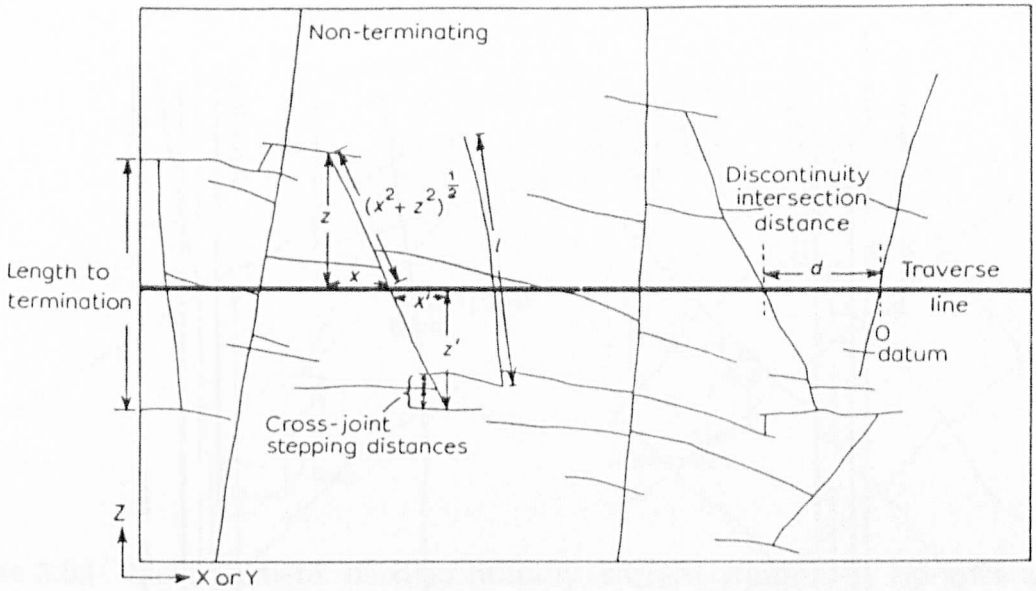


Figure 2.01 Discontinuity distance measurements taken at an exposure (from Attewell and Farmer [9]).

Clearly, the measurement of spacings between adjacent discontinuities along a scanline or length of borehole core does not directly relate to the spacing between discontinuities that are approximately parallel, and a subsequent analysis—based on an assessment of set membership—is required to determine this measurement. This may either be explicit (*i.e.* following identification of two features that belong to the same set of parallel discontinuities, the distance between them along the scanline is determined) or implicit (whereby a least squares analysis of discontinuity frequency measured along a number of non-parallel scanlines is performed [70]).

2.1.2 Discontinuity roughness

Discontinuity roughness is usually measured using some form of linear profile along a discontinuity surface. The ISRM Suggested Method [22] describes measuring such profiles in terms of the physical procedure: taking offsets to the surface from a datum line above the surface itself. This same basic procedure is described by Attewell and Farmer [9], although in this case the use of an instrument commonly used for making laboratory measurements of roughness on metallurgical surfaces is suggested.

The use of linear profiling ignores an important aspect of discontinuity surface roughness, in that it is usually anisotropic (consider, for example, a ripple marked bedding plane). To account for the three-dimensional nature of roughness, a combined compass and clinometer can be used to measure the

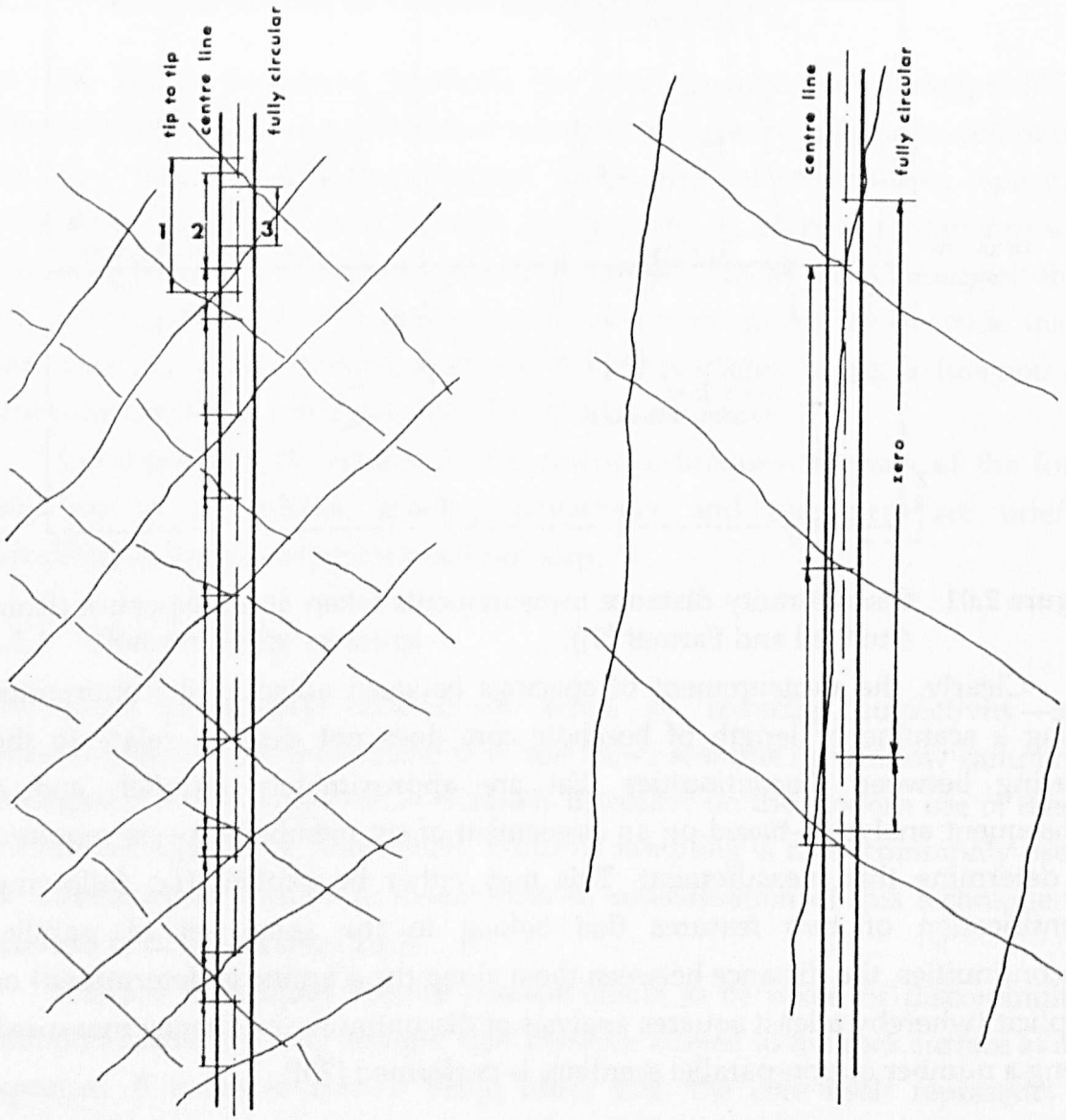


Figure 2.02 Interpretation of discontinuity spacing from borehole core (from Brown [22]).

orientation of a surface at a number of locations, with the resulting range of orientations being considered a measurement of roughness. The use of differently sized plates attached to the instrument allows the orientation range to be determined at different scales; this point is covered further in sub-Section 2.3.1. There are two major deficiencies with this procedure: the first is that of the long time required for its conduct (the ISRM Suggested Method calls for approximately 200 plate positions, representing at least one hours work, on each sampled plane); the second is the lack of a suitable analytical procedure, which is discussed in sub-Section 2.3.2.

A commonly used method of assessing roughness is simply to directly compare a discontinuity surface with a 'standard' profile. Whilst this technique

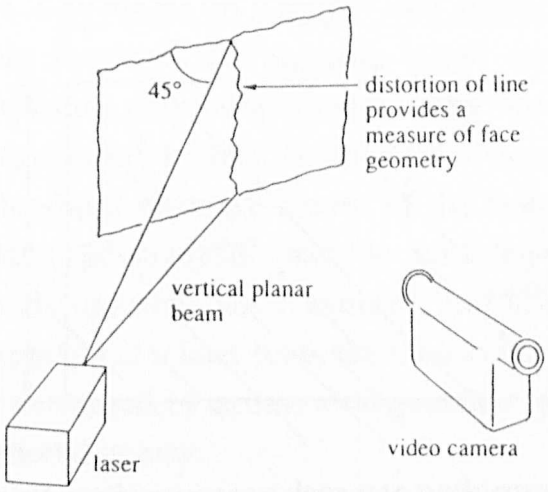


Figure 2.03 Measurement of discontinuity surface roughness by means of laser profiling (from Priest [109]).

is rapid and easily applied, it suffers from the lack of any objectivity, as no physical measurement is made of the surface. Milne [97] reported on an attempt to increase the degree of objectivity by means of taking a photograph of a shadow, cast by a straight edge, onto a rock surface. A similar approach is described by Priest [109] who, referring to other workers, illustrates a method by which a planar beam of light from a scanning laser is projected onto a rock surface for sampling by a video camera attached to a microcomputer, as shown in Figure 2.03.

The use of photogrammetry for measuring roughness is briefly considered both by Priest and as part of the ISRM Suggested Method. For reasons that are discussed in Chapter 3 these techniques have not gained widespread use, although research aimed at improving their ease of use is still being undertaken (see, for example, Miller *et alia* [96]).

It is noteworthy that the ISRM Suggested Method recommends the use of a three-dimensional method of measuring roughness (for example, a compass and clinometer) whenever the direction of sliding of rock blocks is unknown, with linear profiles being reserved for those situations when the sliding direction is known at the time of measurement. Bearing this in mind, it is useful to consider the following two observations: firstly, that the current trend in industry is for site investigation and rock mechanics design to be regarded as two completely different—and often temporally separated—processes; and secondly, that roughness (or a derived parameter, such as angle of friction) will be required as an input condition to complex three-dimensional numerical

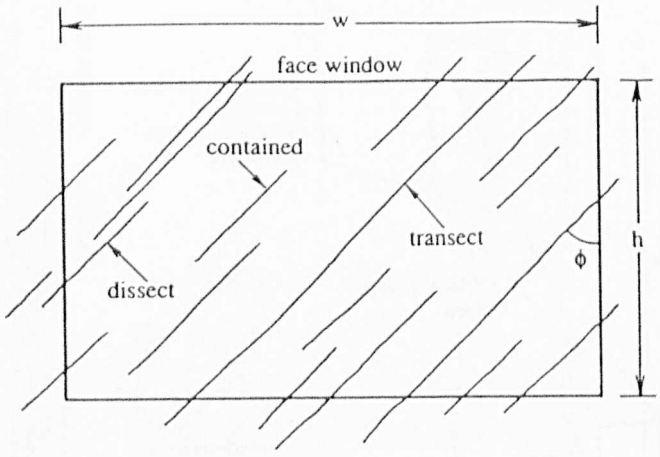


Figure 2.04 Window sampling at a rock exposure (from Priest [109]).

models, where inter-block movement cannot be predicted in advance of computer simulation.

Regardless of the correctness of considering roughness as a three-dimensional phenomenon, until suitable (*i.e.* simple and rapid in use) measurement techniques become available, such an approach will not become widespread. Thus, in order to generally improve the applicability of discontinuity roughness to rock mechanics problems, improved measurement techniques are required.

2.1.3 Discontinuity extent

Discontinuity extent (or persistence) refers to the areal size of the discontinuity plane itself. As is noted in the ISRM Suggested Method, "It is one of the most important rock mass parameters, but one of the most difficult to quantify in anything but crude terms". The basic technique for measuring extent involves the use of window sampling at a rock exposure to determine the trace length of those discontinuities that cross a scanline. This is indicated in Figure 2.01, and also in Figure 2.04.

The measurement of extent in the terms illustrated in Figures 2.01 and 2.04 is straightforward, but time consuming and highly biased by the sampling process. As a consequence of this latter point, sophisticated analytical techniques are required to determine extent accurately, and this is discussed in sub-Section 2.4.1.

An alternative form of window sampling is to use photogrammetry, whereby stereoscopic photographs are taken of the rock face and used for later analysis. Perhaps the prime advantage this method possesses over direct

measurement is that limitations on window size imposed by the height of the human observer are removed. As a result, larger and more representative samples of rock exposure may be analysed. There are further advantages to photogrammetry, connected to factors other than sampling conditions. For example: a permanent and objective record of the face is obtained, such that independent checking or analysis may be undertaken; working in close proximity of a possibly unstable face is avoided; and because the measurement process can be undertaken at a later time, the time available for this work is not constrained to that permitted by access arrangements as, for example, may be the case at an operational quarry.

There are, however, disadvantages to photogrammetry. Although taking a photograph of a rock face is simple, ensuring that the resulting photograph can be used to obtain accurate measurements is not straightforward. Strictly, if measurements are to be obtained from photographs, standard photogrammetric principles must be adopted in order to guarantee a known relation between photograph coordinates and real-world coordinates; these principles are discussed in Chapter 3. Priest [109], however, simply suggests minimising the effect of camera tilt relative to the plane of the rock face, and recommends the use of ordinary photographic procedures to achieve this. As examples, Priest offers: the use of a camera fitted with a long focal length lens and positioned a long way from the face (to minimise the range of camera-to-object distances present, and hence displayed on the photographs); the use of specialised 'tilt-shift' lenses that remove distortion; and 'differential printing', whereby at the time of printing, the tilt introduced at the time of exposure is removed by imposing—between the enlarger and photographic paper—the reverse angle of tilt. Although these techniques do reduce distortion, they cannot be recommended for high accuracy work.

2.1.4 Discontinuity orientation

Three techniques for measuring discontinuity orientation are recognised in the ISRM Suggested Method: the use of a combined magnetic compass and clinometer at a rock exposure; the use of photogrammetry at a rock exposure; and the use of a protractor on oriented borehole core.

Little needs to be said about the method of using a compass and clinometer to determine the orientation of discontinuities visible at an exposure. A geological compass is the preferred instrument, because when it is correctly aligned on an exposed surface, direct readings of dip direction and

dip angle—the most convenient parameters for use in subsequent mathematical or graphical manipulation—are obtained. Both Attewell and Farmer [9] and Priest [109] recommend the use of a scanline to bring a degree of objectivity to the procedure, in that every discontinuity intersecting the scanline is measured, rather than only those that the observer considers critical.

Care must be taken in the vicinity of ferrous objects (either man-made or naturally occurring mineral deposits) as interference with the magnetic compass will result, and in these circumstances the ISRM Suggested Method recommends a procedure in which angles to a scanline of known orientation (determined by customary survey techniques) are measured using a protractor or similar instrument. In the case of discontinuities indicated only as a line intersecting the plane of an exposure, Attewell and Farmer [9] suggest that these can effectively be extended by inserting a non-ferrous plate into them and measuring the orientation of that.

An important aspect of the use of a compass and clinometer is the reproducibility of the results. Priest [109], quoting the results of other workers, suggests that the maximum errors present in measurements of dip direction and dip angle will be ±10° and ±5°, respectively.

The use of photogrammetry for measurement of discontinuity orientation is covered in some depth as part of the relevant ISRM Suggested Method. In essence, this Suggested Method relies on analogue metric photogrammetry (this technique is described in detail in Chapter 3), whereby the real-world location and orientation of a discontinuity is determined from a pair of low-distortion photographs taken at precisely known locations in front of a rock exposure. The Suggested Method has its origins in original research work conducted by Ross-Brown [115], and is a direct descendant of a report produced by the same author [117]. However, as the work presented in Chapter 3 demonstrates, this technique should now be regarded as obsolete and of historical interest only.

Measurement of discontinuity orientation from oriented core is an indirect technique, in that orientations relative to the axis of the core are measured initially and from these, orientations relative to a global frame of reference are determined by either graphical or mathematical procedures. Providing both the inclination relative to the core axis and the rotation relative to a longitudinal datum on the core are measured for a discontinuity, determining the global orientation is straightforward.

The inverse of this procedure—*i.e.* taking measurements from the borehole wall—may also be implemented, by lowering a specially adapted

small diameter video camera down the hole. This instrument is capable of rotating within the hole, and thus transmitting an image of any part of the borehole wall; the image of a discontinuity appears as a sinusoidal trace. The camera usually superimposes an orientation mark (*e.g.* a scale graduated in degrees or a compass) on the image, with the result that discontinuity orientation can be computed from measurements of a discontinuity trace displayed on a television screen. This is, of course, a specialised procedure and one that requires careful planning before implementation.

2.1.5 Summary

Following this review of the customary measurement procedures used for discontinuity geometrical attributes, one common characteristic is apparent: the use of a human observer, working at an exposure with simple instruments (for example, tape measure and compass), and measuring the attributes directly. Notwithstanding the widespread use this method of working commands, two aspects of its utilisation demand further consideration: bias due to sampling location and safety of the observer.

The direct use of a human observer at a rock exposure means that the choice of sampling location is not totally free and unbiased. If human access is required, sampling locations at rock exposures will be limited to those that are no more than 2m from the base of the exposure (*e.g.* invert of a tunnel, muckpile at the base of a quarry face, or ground level at a natural exposure). Similarly, in the event of borehole core, the sampling location is obviously limited to the line of the borehole, which may in turn be constrained by a number of other criteria. Thus, the vast majority of a rock exposure or rock mass will not be surveyed.

In an attempt to remove bias, scanlines are suggested. Unfortunately, although they are a useful means of overcoming subjectivity, when direct human access is required they introduce their own bias, in that they can only easily be set up parallel to, and no more than 1.5m from, the base of an exposure. Orthogonal scanlines are recommended as a means of overcoming directional bias [9, 22, 109], but it is hard to see how this can be easily achieved at a rock face.

At a large number of surface exposures, human access may be either dangerous or impossible, due to the unstable nature of the rock face. When the rock face is unstable, it is difficult for an observer to maintain concentration on the measurement of rock mass geometrical attributes, as personal safety

assumes overriding importance. Furthermore, exposure of a human observer to risk of injury is usually regarded as unacceptable, and so safer alternative practices should be sought.

As a result, we are faced with two seemingly incompatible criteria: the beneficial objectivity brought to the measurement of discontinuity geometry by scanlines means that they should be retained; however, the problems of bias and risk of injury to an observer means that it is difficult to see how, in the case of measurements made at an exposure, the basic principle of using a tape measure affixed to the rock can be regarded as an acceptable *modus operandi*. The conclusion to be drawn from this is that a technique that allows scanlines to be made anywhere on an exposure, and from a safe remote location, should be developed.

This review indicates that photogrammetry is suggested as a means of determining all of the geometrical attributes, and it is without doubt a remote, and hence safe, method that is not biased—in the way scanlines are—with respect to position or orientation on an exposure. However, little use is made of this procedure, probably due to a perceived need for specialised techniques and equipment. This probably stems from the anachronistic nature of the ISRM Suggested Method, and therefore begs the question, can modern photogrammetry be applied to the problem of measuring rock mass geometrical attributes? An attempt is made by the current Author to answer this question in the remainder of this thesis. It is appropriate now, however, to consider how these geometrical attributes are analysed; this is the subject of the next four Sections.

2.2 Analysis of Discontinuity Spacing

The spacing between adjacent discontinuities in a rock mass is of fundamental importance in rock engineering as it defines, to a large degree, the size of the blocks of intact rock that exist within a rock mass. Given the significant nature of discontinuity spacing, we should expect that robust techniques exist for use in its analysis; this is not the case, however. Thus, whilst geologists have long been able to account for discontinuities in a qualitative way, for example by recognising the effects of lithology, *in situ* stress, fluid pressure and tectonic events, a general rigorous quantitative model for predicting spacings has not been developed. Many attempts have been made to quantitatively model spacings, and the application of three techniques—mechanistic, probabilistic

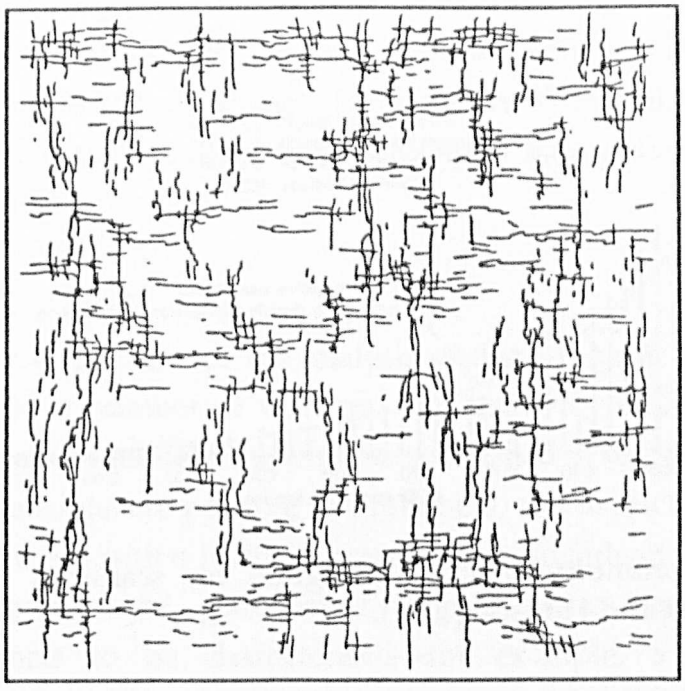


Figure 2.05 Numerical simulation of discontinuity networks under conditions of plane stress (from Renshaw and Pollard [113]).

and fractal models—has met with modest success. These three techniques are now described in the following three sub-Sections.

2.2.1 Mechanistic models

Recognising that joint spacings in a stratum tend to be proportional to the thickness of the stratum itself and approximately uniform, Price [107] developed a simple model based on the mechanics of fracture to predict these spacings. This has recently been extended, using fracture mechanics techniques, by Pollard and Segall [105] and Engelder [38], who discusses patterns of multiple fractures and joints, invoking the principles of structural geology. Similarly, Aydin and deGraff [10] studied the development of columnar jointing in lava flows, and concluded that fracture surface energy and strain energy of the system are the prime factors in controlling the evolution of discontinuity networks. Invoking these principles, Renshaw and Pollard [113] have succeeded in numerically simulating the generation of discontinuity networks in plates subjected to plane stress conditions, as shown in Figure 2.05.

Notwithstanding these advances, no mechanistic model yet exists for predicting three-dimensional discontinuity patterns resulting from the application of a general stress state.

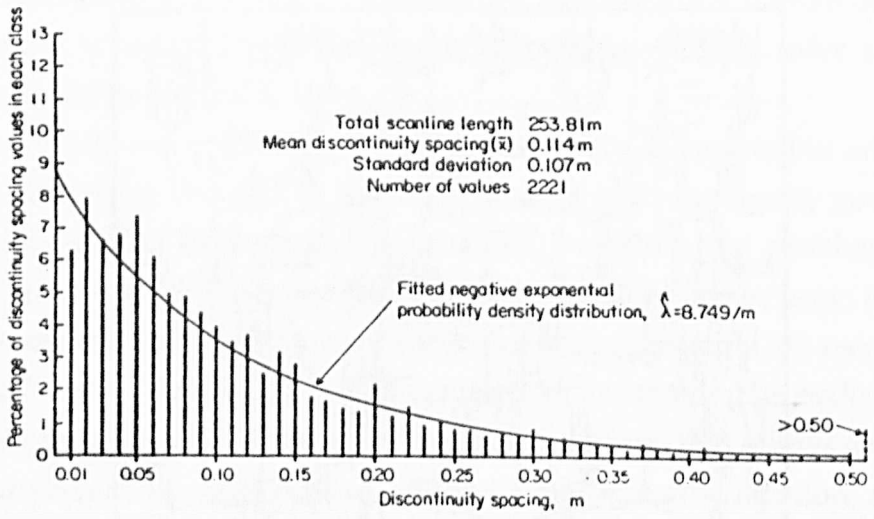


Figure 2.06 Discontinuity spacing histogram for scanlines in chalk (from Priest and Hudson [110]).

2.2.2 Probabilistic techniques

Probabilistic techniques have been found to be most useful in analysing rock mass geometry based on discontinuity spacings. In a series of seminal papers [63, 64, 110, 111], Hudson and Priest demonstrated that discontinuity spacings often tend to follow a negative exponential distribution, *i.e.* $f(x) = \lambda \exp(-\lambda x)$, as shown in Figure 2.06. These workers also showed that, through the application of probabilistic statements, the negative exponential nature of spacing distributions leads to the derivation of useful engineering parameters, such as determining the likelihood of encountering zones of highly fractured rock.

However, such probabilistic techniques are not universally applicable, as there are a great number of rock masses that do not exhibit negative exponential distributions of spacing (columnar jointing in lava flows, for example, tends to possess uniform spacings). The introduction of probabilistic distributions that are more complex—even the well-known and often-used normal distribution, for example—as a means of overcoming this deficiency soon leads to formulations that are analytically intractable.

As an alternative approach, the derivation of discontinuity spacings from the construction of probabilistic joint system models has been attempted by a number of authors. Dershowitz and Einstein [34] and Kulatilake [76] performed reviews of those stochastic models available up to mid-1987, and reported that they generated either Poisson, negative exponential or lognormal

distributions. Interestingly, whilst not demonstrating that discontinuity spacings result from a stochastic process, these workers nevertheless produced field data that supported all of their (very different) hypotheses. This would appear to indicate that there is no universally applicable probabilistic model for discontinuity spacing.

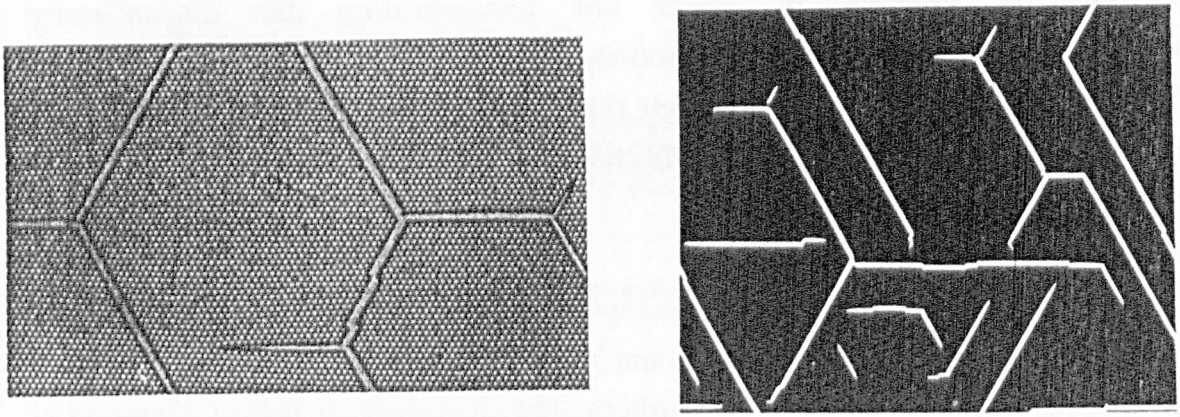
2.2.3 Fractals

Recently, the use of fractals for the analysis of discontinuity spacing has been given attention by a number of workers. The discovery of fractal methods is generally credited to Mandelbrot, who has published a key text on the subject [89]. Fractal dimensions are positive real numbers, and as such can be regarded as a superset of the positive integers representing Euclidean dimensions. This allows the departure of geometrical objects from standard Euclidean geometrical objects to be characterised—for example, a surface that is represented by a Euclidean plane has both fractal and Euclidean dimensions of 2 (or, more precisely, 2.0 and 2 respectively) but as the roughness of this surface (*i.e.* departure from a Euclidean plane) increases, so its fractal dimension increases towards 3.0, at which stage the surface has become volume-filling. One further property of fractal geometry is that it is scale independent: regardless of the scale of study, the fractal dimension of an object is invariant. This property is difficult to assess, as all real objects are of finite size, and real measuring techniques have a finite resolution.

The ability to characterise objects that can be regarded as having non-Euclidean attributes means that fractals have been applied in the search for solutions to a large number of problems in various fields, and the nature and occurrence of fracture is one of them. In the literature, 'successful' application is usually signalled by the reporting of plausible fractal dimensions, together with a range of scales over which the dimension has been found to be invariant. A number of examples relevant to discontinuity spacing follow.

(i) Barton and Larsen [13] mapped rock pavements in Miocene ash-flow tuffs, discovering that the discontinuity spacings had fractal dimensions of between 1.12 and 1.16 over two orders of magnitude of scale.

(ii) Turcotte [134] examined the size distributions of fragments of broken coal, granite fragmented in underground nuclear explosions and basalt fragmented by an impacting projectile, and showed how they all follow a fractal relationship over more than 3 orders of magnitude in scale. In each case a fractal dimension of approximately 2.5 was indicated.



(a) experimental (b) numerical

Figure 2.07 Fractal fracturing in plane stress (from Skjeltorp and Meakin [128]).

(iii) Solla [129] used a probabilistic generation of rock strength to model the propagation of cracks along geological faults, and found a scale invariant mechanism for growth and coalescence of cracks.

(iv) Louis and Guinea [84] modelled fracture commencing on an atomic scale. In their model, which uses the equilibrium equations with the Lamé constants simulated by the behaviour of atomic bonds, fracturing was found to produce a fractal relation between the number of bonds broken and the amount of material affected. In this case the fractal dimension was 1.55 for uniform dilation and 1.60 for shear.

(v) Skjeltorp and Meakin [128] noted the experimental development of fractal fracture patterns. These workers took 3.4μm diameter spheres of sulphonated polystyrene, and fabricated sheets of strained material exactly one sphere thick. The strain was subsequently dissipated and the sheets allowed to fracture. A numerical model, based on bond energies and a probabilistic statement regarding their breaking, produced strikingly similar results, as shown in Figure 2.07.

The use of fractals in modelling fracture is at a very early stage of development. However, the work that has been performed has identified fractal relations on the bases of both prototypical measurements and of the development of fracture from a theoretical base, which suggests that a useful technique for the analysis of discontinuity spacing may result in due course. Perhaps more importantly, given that fractals are one manifestation of chaotic dynamic systems [48], this suggestion can be made that the process of fracturing should be regarded as a similar dynamic process, and modelled accordingly.

2.2.4 Summary

The greatest success to date in the analysis of discontinuity spacings has been achieved through the use of probabilistic techniques, particularly those allied to the negative exponential distribution function. However, these are not universally applicable, as a number of rock masses possess spacing distributions that are not adequately characterised by tractable distribution functions.

Deterministic models, and models based on fractal geometry, are presently the subject of research, and although they demonstrate promise, require a great deal of further development before they approach the level of application enjoyed by the probabilistic methods.

We must conclude, therefore, that no universally satisfactory method exists for the analysis of discontinuity spacing.

2.3 Analysis of Discontinuity Roughness

Given that the majority of measurements of discontinuity surface roughness result from the application of a linear profiling technique, it is on this basis that most analysis takes place. However, three-dimensional techniques for measuring roughness are also used, and it is useful to consider the analytical methods applied in such cases. Consequently, the following two sub-Sections consider the analysis of linear profiles and the analysis of areal (*i.e.* three-dimensional) measurements, respectively.

2.3.1 Data from profile measurements

Bandis [11] notes that the quantitative analysis of discontinuity surface roughness may be undertaken using a number of methods, and quotes three: estimation of the maximum and median asperity angle i, from

$$i = \arctan\left(\frac{2a}{L}\right) \tag{2.01}$$

where a=amplitude and L=baselength of the irregularities; arithmetic mean of the elevations of the peaks and troughs measured with respect to the mean plane; and computation of the 'amplitude index', the ratio of the sum of projected asperities to the total length of the measured profile.

A number of statistics for the characterisation of surface roughness may be borrowed from the field of roughness measurement in metallurgy. Goodfellow

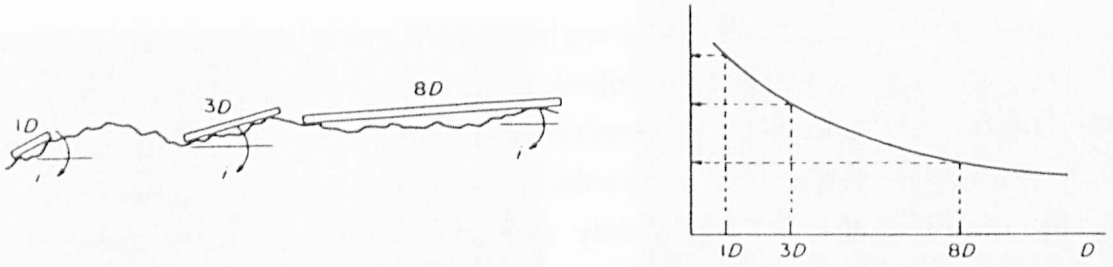


Figure 2.08 Scale effect in roughness determination from linear profiles (from Bandis [11]).

[50] considered the use of centreline average roughness, R_a, and root mean square roughness, R_q. These are calculated from offset measurements taken along a linear profile, using the formulae

$$\left.\begin{aligned} R_a &= \frac{y_1 + y_2 + y_3 + \cdots + y_n}{n} \\ R_q &= \frac{\sqrt{y_1^2 + y_2^2 + y_3^2 + \cdots + y_n^2}}{n} \end{aligned}\right\} \qquad (2.02)$$

where the individual offset measurements y_i are made at uniform spacing values.

In all cases, the problem exists of determining the optimal sampling length over which the roughness value is to be computed. This is a major deficiency, as roughness determined in this fashion is scale dependent, as Figure 2.08 shows. Consequently, if the results are to be relevant to an engineering problem, the scale at which the determination of roughness takes place is crucial. Thus, if the engineering scale is $8D$ or more (in the context of the scales illustrated in Figure 2.08), roughness measurement and computation at the customary scale of, say, $1D$ will be meaningless.

This scale effect problem is one that cannot be lightly disregarded, as a commonly applied semi-quantitative method of assessing roughness is by means of reference to 'standard roughness profiles'. Such profiles customarily refer to a fixed assessment length (as, for example, the profiles of Joint Roughness Coefficient (JRC) [22], in which case the assessment length is 10cm), and this invalidates their application at different scales. A method of overcoming this difficulty in the case of JRC is proposed by Bandis [11] who, quoting some of his earlier work, proposes a scale effect correction such that JRC may be applied to lengths in excess of the assessment length. However,

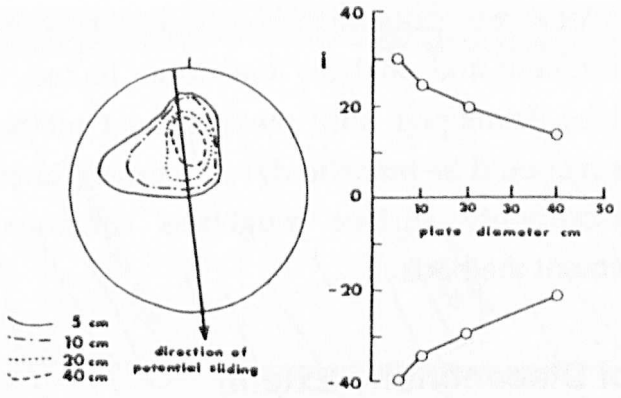


Figure 2.09 Analysis of three-dimensional roughness measurements (from Brown [22]).

some reservations must be aired at this seemingly indiscriminate use of extrapolation.

2.3.2 Data from areal measurements

Notwithstanding the great potential of an analysis of roughness based on areal (or three-dimensional) measurements, quantitative techniques for undertaking such an analysis are scarce.

The ISRM Suggested Method refers extensively to the pioneering work of Fecker and Rengers [43], but this indicates an essentially qualitative technique, whereby surface orientation measurements are plotted on the hemispherical projection. An example of this analysis is shown in Figure 2.09, both as a spread of orientations on the hemispherical projection and as the effective asperity angle along a potential sliding direction.

Recent work by Harrison and Goodfellow [56] suggests that the use of multivariate statistics offers scope for improved quantitative analysis. In this method, the eigenvalues of a matrix generated from the direction cosines of the orientation measurements are used to quantify the spread of orientations. Although these workers demonstrated the efficacy of the technique, there is no extensive experience of using it for the analysis of roughness, and at present it remains a research topic.

2.3.3 Summary

It is clear from the work presented in this Section that the customary quantitative analysis of roughness takes place using data obtained from a linear profile. It can be argued that this is the case simply because the majority of

roughness measurements are obtained from linear profiles. However, the techniques of measurement and analysis tend to be linked, in that analytical procedures will not be developed until measurement methods are, and *vice versa*. Thus, an attempt should be made to develop three-dimensional analytical procedures for discontinuity surface roughness contemporaneously with appropriate measurement methods.

2.4 Analysis of Discontinuity Extent

Customary methods of analysing discontinuity extent rely on probabilistic principles. This can be taken as a reflection of the unknown nature of the extent of discontinuities; after all, such information can only be gained through a complete dissection of a rock mass, and this enterprise has not yet been undertaken.

The mathematics of this probabilistic analysis is complex, and not strictly relevant to the work in hand here. Consequently, it may be regarded as being beyond the scope of this work, and so the review of it has been limited to brief descriptions. The following two sub-Sections relate to two aspects of this work: analysis based on sampled measurements of discontinuity trace length; and analysis based on stochastic models of discontinuities.

2.4.1 Methods based on discontinuity trace length

The study of discontinuity persistence in terms of the two-dimensional traces formed by the intersection of a discontinuity plane with a rock exposure surface has been undertaken by Priest and Hudson [111]. These workers formulated general relations between complete trace length (*i.e.* the lengths of discontinuity traces that would be found on a rock exposure of infinite extent) and samples of the various lengths that occur in reality.

Thus, recognising that the majority of field measurements would be either of semi-trace lengths or censored semi-trace lengths, where these terms are defined in Figure 2.10, these workers sought to discover links between the probability density distributions applicable to these features and to the distribution of complete lengths.

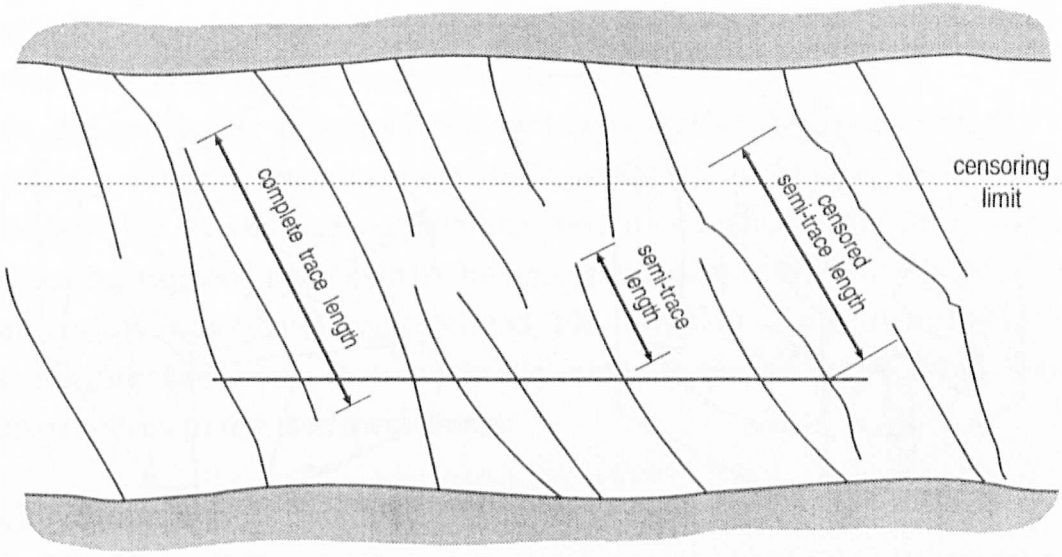


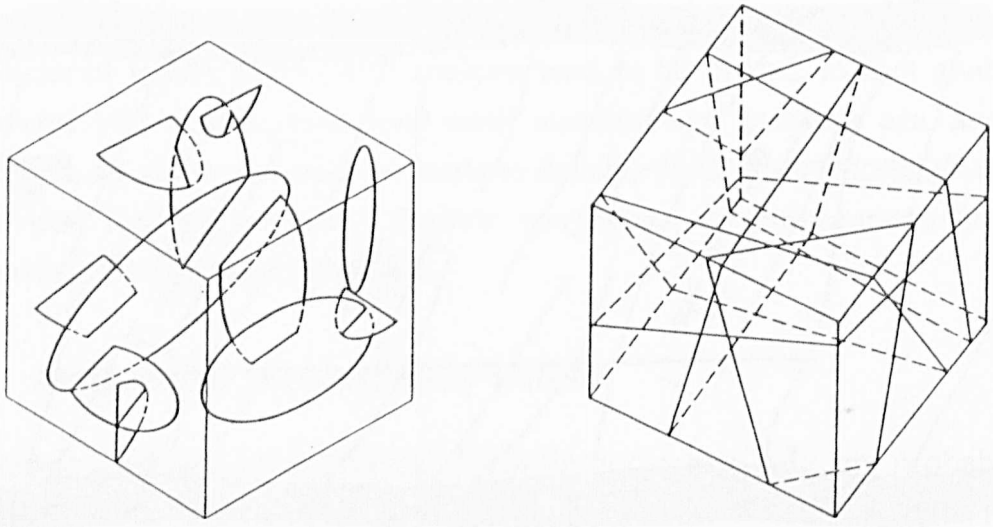
Figure 2.10 Complete, semi- and censored semi-trace lengths.

A report on a numerical study of the applicability of normal, lognormal, exponential, Weibull and gamma distributions to discontinuity extent (and also spacing) has been published by Mahtab *et alia* [87]. These workers took discontinuity extent data from six exposures in different rock types, separated them into sets on the basis of discontinuity orientation, and then attempted to fit the various distributions. By their own admission, the results are inconclusive: no one distribution fared better than any other, and in some cases a satisfactory fit could not be achieved with any model. They concluded that the use of probabilistic distributions for the characterisation of discontinuity extent must be undertaken with care, and alternative approaches adopted if no reasonable fit is apparent.

2.4.2 Stochastic modelling of discontinuities

As the reviews of Dershowitz and Einstein [34] and Kulatilake [76] show, a number of workers have tackled the problem of predicting trace length from stochastic models of discontinuities within a three-dimensional space. These models are based upon two-dimensional geometric figures—representing complete discontinuity surfaces—distributed in a three-dimensional space, and allow numerical sampling of various geometrical parameters, including discontinuity extent.

The distribution of the geometric figures is usually based on probabilistic criteria, applied to both location and orientation in space, with various forms being adopted for the figures themselves: for example, orthogonal infinite



(a) Circular and elliptical disks (b) 3-D Poisson plane process

Figure 2.11 Examples of stochastic models of rock mass geometry (from Dershowitz and Einstein [34]).

planes, circular or elliptical planar shapes, polygons resulting from the application of a two-dimensional Poisson process and mosaic tessellation. Examples of models based on circular and elliptical disks, and the three-dimensional Poisson plane process, are shown in Figure 2.11.

Warburton [139, 140] has applied stereology (geometric probability, as used to extrapolate measurements obtained in space of a given dimensionality to results in space of a higher dimensionality—see, for example, Santaló [122]) to the problem of interpreting trace length data. This work is, perhaps, the first application of stereology to rock mass geometry, but although it produced some interesting results it has not been taken further. One aspect of stereology that may hamper its application in this field is the requirement for a model of the shape of the objects under consideration. Presently, this is limited to simple Euclidean solids, and hence is not applicable to the arbitrary polyhedra that constitute the blocks in a rock mass.

The difficulty of using stochastic models is highlighted in recent work by Kulatilake and Wathgula [77]. These workers used eight modelling schemes in an attempt to reconcile field data from, and hence determine the geometry of, a granitic rock mass in Sweden. Their results indicated that no one scheme was particularly applicable, with each being superior to the others in one or more aspects of producing a geometry that compared well with the field data. In conclusion, these workers recommended that a number of such schemes should

always be used in order to determine the most applicable model, and that further work is needed to produce improved models.

Notwithstanding the views of Dershowitz and Einstein [34], who feel that such rigorous techniques are advantageous, there are serious fundamental criticisms to be levelled at stochastic modelling, one of which is that the underlying stochastic nature of rock masses has not been demonstrated. Indeed if, as a number of workers have suggested [38, 105, 107, 113], discontinuities form by deterministic processes, then stochastic methods can never be more than an approximation to the true mechanism.

2.4.3 Summary

The analysis of discontinuity extent is customarily undertaken on the basis of either models founded on probabilistic distributions of trace lengths, or stochastic three-dimensional models of rock mass geometry. The intractability of the underlying probabilistic models means that analytical procedures have been obtained for only the simplest of cases, with numerical analysis being the most commonly adopted procedure.

The examples presented in the previous two sub-Sections show that evidence for the unequivocal use of either models based on the distribution of trace lengths or stochastic geometrical models does not exist. Indeed, the published evidence seems to indicate that, depending on the data being analysed, any part of the complete spectrum of applicability may be appropriate: from complete inapplicability, through a good prediction offered by one or two distribution, to a reasonable prediction being offered by many distributions. This variability in suitability points to a fundamental deficiency in these methods of analysis, and one that is important to identify and overcome.

2.5 Analysis of Discontinuity Orientation

Rock mass characterisation for engineering purposes almost always includes an analysis of discontinuity orientation. As prediction of discontinuity orientation from a model of discontinuity genesis is currently not possible, the accepted approach is to identify clusters of preferred orientation within large sets of *in situ* orientation data. The basis for this analysis is that each orientation measurement is considered to represent the true orientation of a planar

discontinuity, and that these planes can then be grouped, on the basis of similar orientation, into sets.

Customarily, the identification of sets of discontinuities possessing similar orientation is undertaken by methods that involve the identification of significant clusters of orientation vectors. Because these vectors are undirected and may be considered as being of unit magnitude, these clusters are often determined on the basis of contours of vector density on the surface of the unit hemisphere. In addition, other techniques based on the identification of a mathematically optimal partition of the data have also been proposed. Consequently, these two techniques are described in the following two sub-Sections.

2.5.1 Contouring on the hemisphere

In the field of geomechanics, contouring orientation data on the hemispherical projection as a means of identifying significant clusters is not only well established but, in some cases, is recommended as a technique for determining preferred orientation of discontinuities [29, 59]. Using this technique, the total problem is usually solved in three distinctly separate parts: determination the contours; identification of any clusters of preferred ranges of orientations from these contours; and evaluation of parameters that quantitatively describe such clusters. Thus, it is convenient to discuss the complete contouring procedure in terms of these operations, and this is done in the following three Divisions.

Contouring process

Although the contouring of orientation data may be performed either manually, using semi-graphical contouring techniques associated with hemispherical projection methods [108], it is increasingly undertaken using a computer-based numerical adaptation of a variant of the manual scheme. Development of such computer-based schemes began in the early 1970s [20, 86], probably in recognition of the time consuming nature of this analytical process.

Almost all computer programs for the analysis of orientation data numerically implement the manual method of **Schmidt contouring**, in which a grid is laid over a hemispherical projection plot of discontinuity poles, a counting circle placed at each grid point in turn and the number of poles within the circle counted, and then contouring of these values taking place by either visual or numerical interpolation of the grid values.

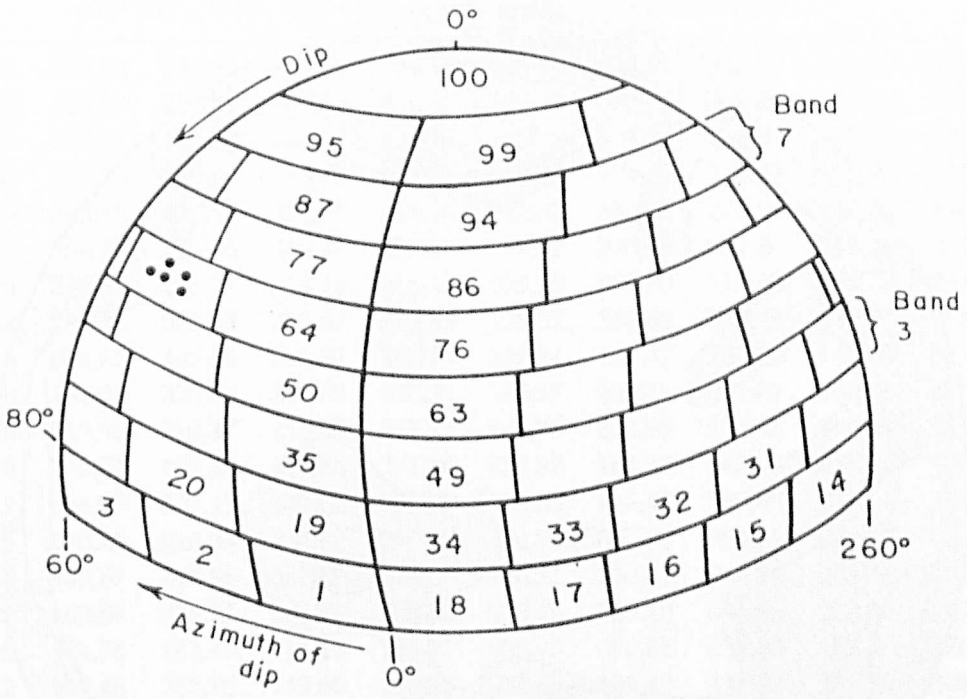


Figure 2.12 Program PATCH: division of the hemisphere into 100 equal area spherical quadrilaterals (from Mahtab *et alia* [86]).

One of the first computer implementations of the Schmidt contouring procedure—the program PATCH by Mahtab *et alia* [86]—did not count projected poles, but simply determined the number of discontinuity normals falling in each of 100 equal area spherical quadrilateral patches drawn on the hemisphere, as shown in Figure 2.12. This procedure was probably adopted due to constraints of processor execution time and limited computer memory. Projection of the data only took place in the production of output, which typically took the form of a printed projection of the grid values at the centre of each patch.

A contoured equal area upper hemispherical projection resulting from an analysis using PATCH is shown in Figure 2.13. The data used in the generation of Figure 2.13 have since been used by a number of workers (for example, [52] and [125]), with the result that this set offers an excellent means of directly comparing different techniques for the analysis of preferred orientation. As such, it will be used in this work for comparative studies. The data themselves represent 286 measurements of discontinuity orientation in a porphyry copper deposit, and are reproduced in Table 2.01 below.

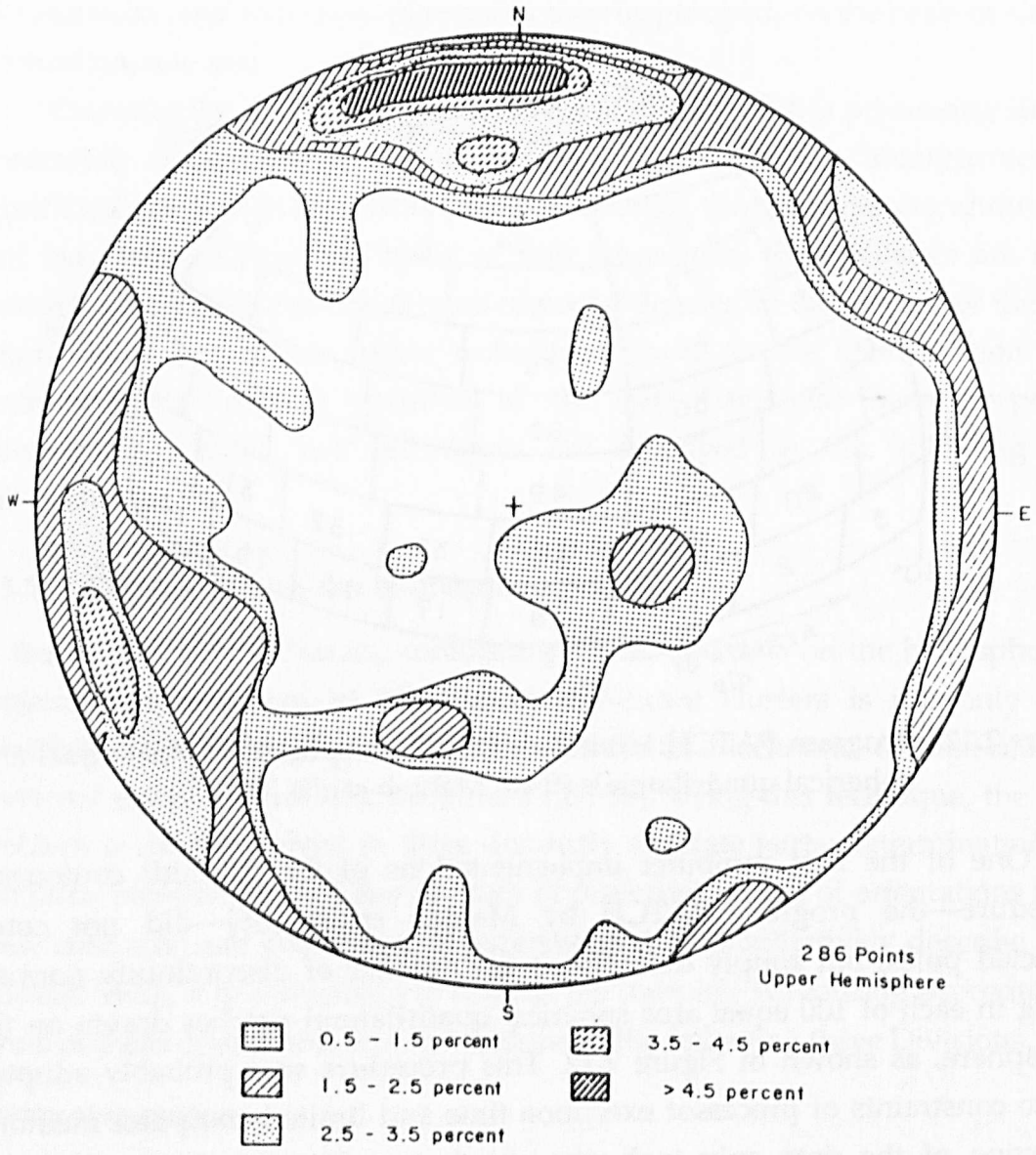


Figure 2.13 Program PATCH: equal area upper hemispherical projection of porphyry copper data (from Mahtab *et alia* [86]).

An analysis of the traditional counting circle approach of Schmidt contouring is offered by Bayer [14], and this work demonstrates how Schmidt contouring produces incorrect results through giving unjustified weight to certain data. Figure 2.14 shows how the size of the counting circle (in this example 5% and 20% of the surface area of the hemisphere) affects the outcome of the clustering: there is little similarity between the results. This is a major disadvantage, as the clustering should reveal some *intrinsic* property of the data, *i.e.* one that does not depend on the method by which it is determined.

Table 2.01 Porphyry copper orientation data (formatted as dip direction, dip angle) (from Mahtab *et alia* [86]).

186,83	255,50	255,65	345,87	255,87	325,86	342,85	270,57	242,70	062,83
127,23	355,80	232,82	240,85	075,80	007,90	060,37	062,26	202,75	015,90
083,42	250,87	360,77	244,81	275,35	245,79	314,88	258,87	357,74	009,56
352,80	265,73	030,80	240,77	136,87	218,45	274,87	225,67	355,90	230,55
248,70	343,85	235,53	050,90	249,30	045,87	356,85	244,58	244,65	248,82
056,87	356,87	230,86	266,84	252,86	255,82	240,55	012,86	229,64	232,60
327,90	210,80	255,77	218,78	216,33	008,70	084,80	175,87	345,87	232,82
234,82	247,32	028,58	053,87	350,83	105,32	360,88	037,85	057,24	167,86
006,70	024,87	345,88	240,87	360,70	330,84	217,47	260,20	162,88	077,18
327,80	132,06	220,86	343,60	030,34	150,87	230,85	041,87	350,86	250,77
228,60	013,90	210,47	255,68	247,78	161,70	220,65	177,46	147,85	342,88
251,78	355,75	153,16	355,84	214,18	057,87	163,85	045,03	085,80	180,00
110,82	028,90	007,15	140,67	050,90	342,80	216,43	190,07	010,64	237,87
345,85	070,20	016,34	013,82	247,75	314,90	065,80	354,75	033,20	081,32
008,47	143,87	240,85	067,87	015,67	126,31	250,87	346,86	243,88	081,13
251,86	182,86	025,21	245,78	338,80	020,25	325,20	044,90	326,90	264,80
133,35	302,78	181,40	111,19	186,83	086,86	090,65	036,88	155,80	276,82
158,22	168,46	357,78	149,60	291,80	013,85	199,43	332,82	331,80	346,70
176,43	287,90	154,40	280,90	257,86	354,75	129,26	116,86	115,25	121,20
105,90	257,80	167,30	217,65	341,70	360,70	085,12	315,90	296,90	336,82
270,74	119,32	205,36	344,74	010,90	118,37	270,88	126,20	014,87	106,44
109,85	193,65	077,58	101,80	141,86	167,57	093,86	328,80	341,90	177,36
314,70	286,90	133,20	351,75	157,34	358,87	270,90	230,50	147,70	145,25
353,70	270,73	084,75	327,83	297,80	097,28	308,27	233,60	337,70	292,57
344,70	266,65	344,70	291,45	268,65	010,68	328,90	337,70	270,54	355,60
254,25	350,65	281,60	267,22	147,85	270,85	150,83	247,63	328,60	286,42
347,62	353,85	285,75	334,66	277,66	192,77	333,67	276,54	346,79	073,83
297,65	356,74	328,78	313,58	304,70	013,63	263,63	297,55	074,90	279,75
301,52	334,72	083,82	340,65	304,53	350,90				

Bayer identifies the step function represented by the boundary of the counting circle in the Schmidt method as causing this phenomenon. In essence, the method determines 'membership of the set of poles close to a grid point', and the step function ensures membership is binary and is precisely determined at the boundary of the circle. As a means of overcoming this deficiency, Bayer advocates the use of non-constant weighting functions, whereby 'degree of membership of the set of poles close to a grid point' is determined by radial location within the counting circle. The family of weighting functions is infinite, but Bayer offers an analysis demonstrating the optimality of a linear function, monotonically decreasing from a maximum at the circle centre to zero at the boundary. In Chapter 4 of this work, it is demonstrated that the apparent improvement due to this technique is mathematically sound.

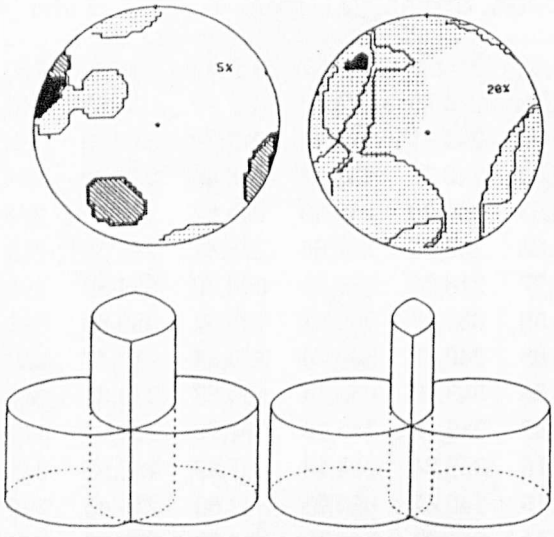


Figure 2.14 Discrepancy in clustering due to the use of counting circles and Schmidt contouring (from Bayer [14]).

This reasoning offers insight into one of the deficiencies associated with PATCH. In this program, set membership is not only defined exactly at the boundary of the spherical quadrilateral patches, but these boundaries are fixed in location relative to the global coordinate system. This means that small clusters of data may be centred over the common corner of four adjacent areas, with the result that erroneously low densities will be determined. Although exacerbated through the use of a coarse counting cell structure in PATCH, this is a deficiency inherent in all counting cell approaches, and can only be avoided through the use of methods that do not rely on determining density within some area of the hemispherical surface.

Notwithstanding the serious deficiencies associated with the use of contouring for determination of preferred orientation, much work continues with variants of the scheme. Two recent examples are those of Charlesworth *et alia* [27] and vanEvergingen *et alia* [136]; the programs presented in these two publications are both based on such an approach, with the only innovations appearing to be enhanced ease of use and output quality. The problem of objectively, rather than subjectively or customarily, setting the size of the counting circle has been ignored.

Having identified these drawbacks and given the customary use of Schmidt contouring, it is important to examine how variable the results may be. A contouring analysis of the porphyry copper data has been undertaken using the computer program DIPS [61]. This program is typical of most current

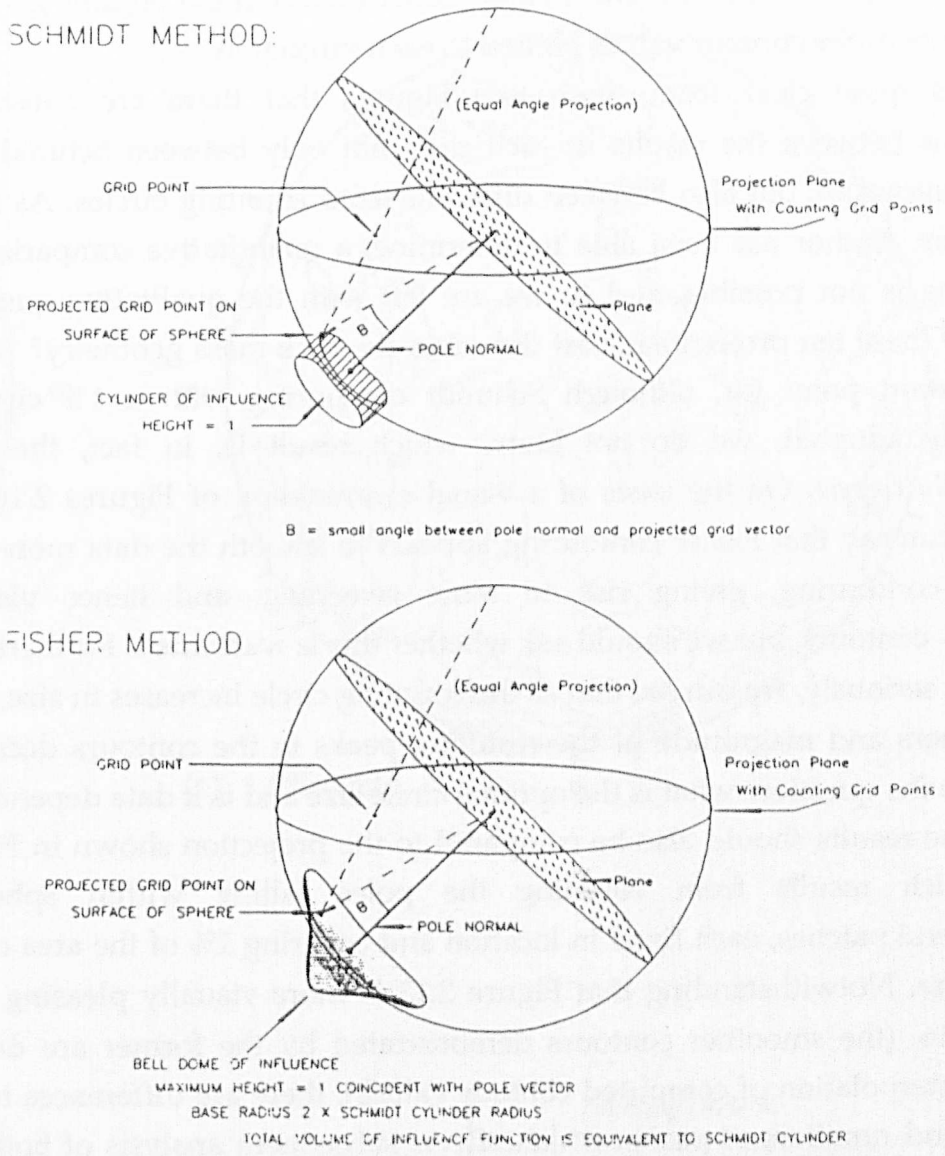


Figure 2.15 Schmidt and Fisher contouring, as implemented in the program DIPS (from DIPS user's manual [61]).

offerings of the genre, providing the user with the facility to produce various types of hemispherical projection, use both Schmidt and **Fisher contouring** methods, and produce hardcopy output. Schmidt contouring has been described above; Fisher contouring, according to the authors of DIPS, replaces the constant weighting function of the Schmidt method with a variable weighting function derived from the Fisher distribution [90]. The methods are shown graphically in Figure 2.15.

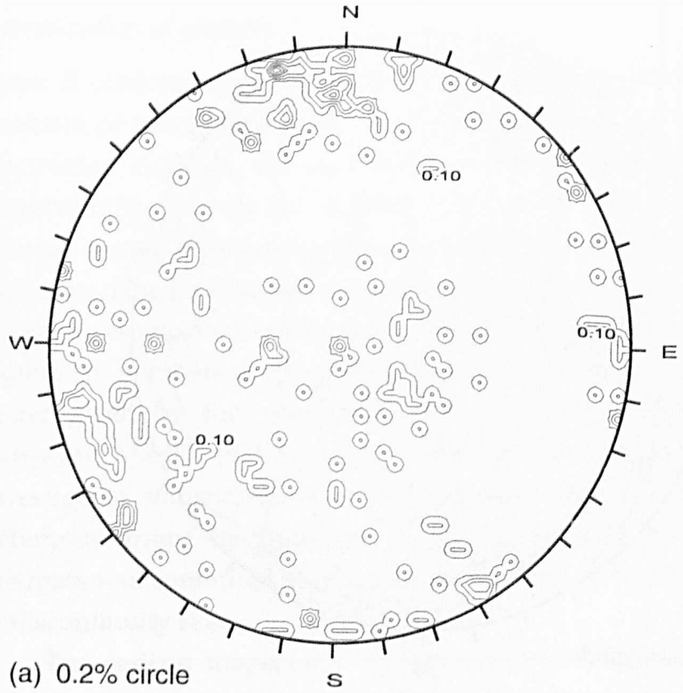
Figures 2.16 and 2.17 show the results of contouring the porphyry data. The contours result from the application of counting circles of 0.2%, 0.5%, 1.0%, 2.0% and 5.0% of the surface area of the unit hemisphere, with Schmidt and Fisher contouring being adopted in the first and second of these Figures,

respectively. The table in the lower right corner each of these Figures gives the equivalence of the contour values plotted in each projection.

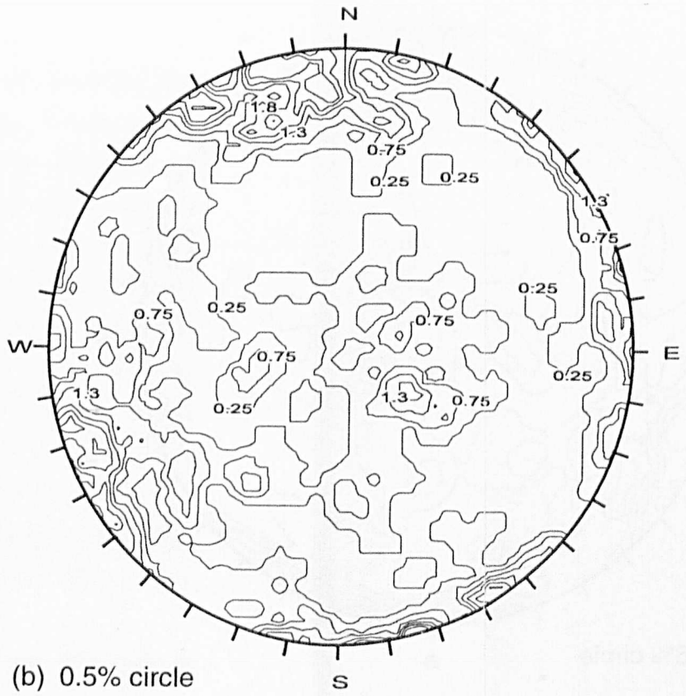
It is quite clear from these two Figures that there are substantial differences between the results in each case: not only between Schmidt and Fisher contouring, but also between different sized counting circles. As far as the present Author has been able to determine, a quantitative comparison of contouring is not possible, and so we are left with the qualitative question: which, of these ten projections, best describes the rock mass geometry? This is an important point for, although Schmidt contouring with a 1% circle is *customarily* adopted, we do not know which result is, in fact, the most *intrinsically correct.* On the basis of a visual examination of Figures 2.16 and 2.17, we can say that Fisher contouring appears to smooth the data more than Schmidt contouring, giving rise to more sweeping—and hence visually pleasing—contours, but we should ask whether this is warranted. Furthermore, and more seriously, we can see that as the counting circle increases in size, both the numbers and magnitude of the resulting peaks in the contours decrease, leading to the question, what is the optimal circle size and is it data dependent?

These results should also be compared to the projection shown in Figure 2.13, which results from counting the poles falling within spherical quadrilateral patches, each fixed in location and covering 1% of the area of the hemisphere. Notwithstanding that Figure 2.13 is more visually pleasing than Figure 2.16, (the smoother contours demonstrated by the former are due to manual interpolation of computed contour values), there are differences in the location and numbers of peaks, which affects subsequent analysis of both the orientation and numbers of cluster mean vectors. Again, no quantitative analysis can be performed to determined relative correctness.

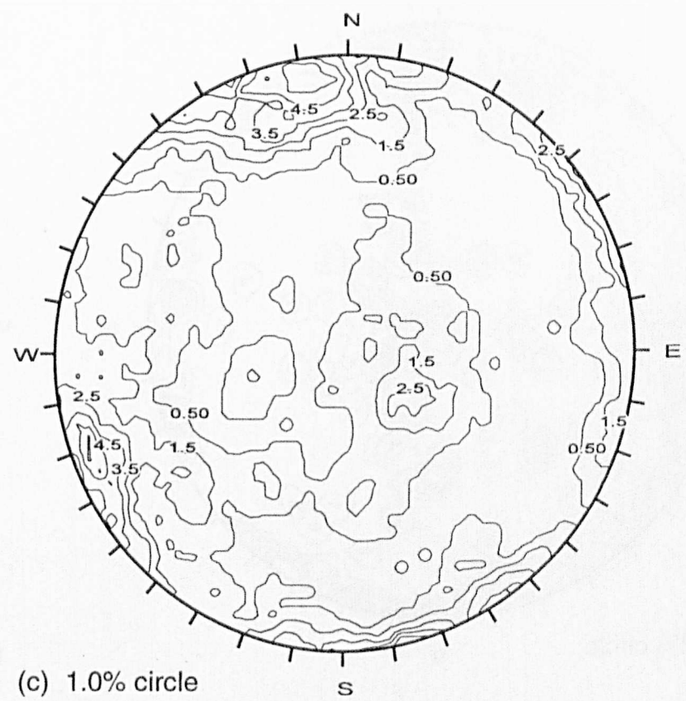
In attempting to answer the question of which contouring approach is most correct, sight must not be lost of the fundamental reason for contouring orientation data; *i.e.* the determination of preferred orientation. Contouring, based on the counting of poles, is an indirect technique for this analysis, depending on an arbitrary means of agglomerating data into local sets of similarity. If the choice of the underlying controlling parameters of circle size and weighting function cannot be quantified, with a view to determining their optimality, then arbitrary choices leading to arbitrary results will always be the consequence. We can conclude, therefore, that contouring of orientation data, based on counting circles, is a wholly unreliable technique, and an objective and mathematically sound replacement method must be sought.



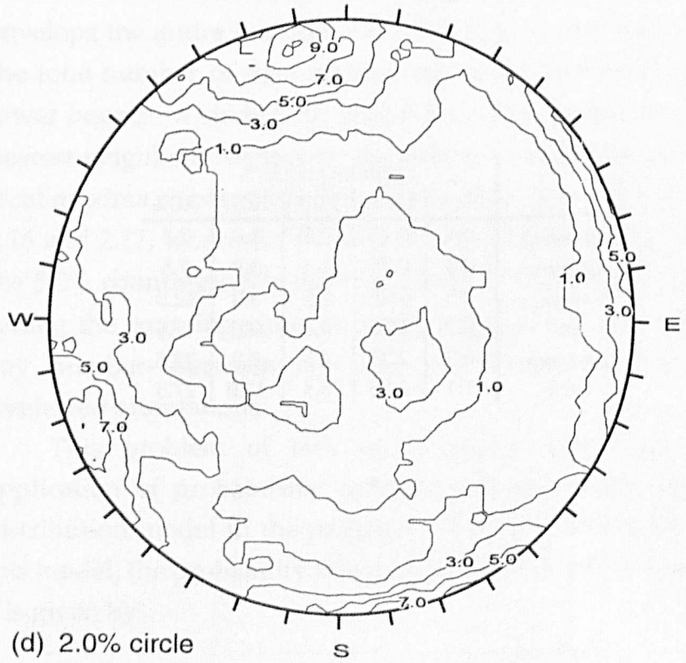
(a) 0.2% circle



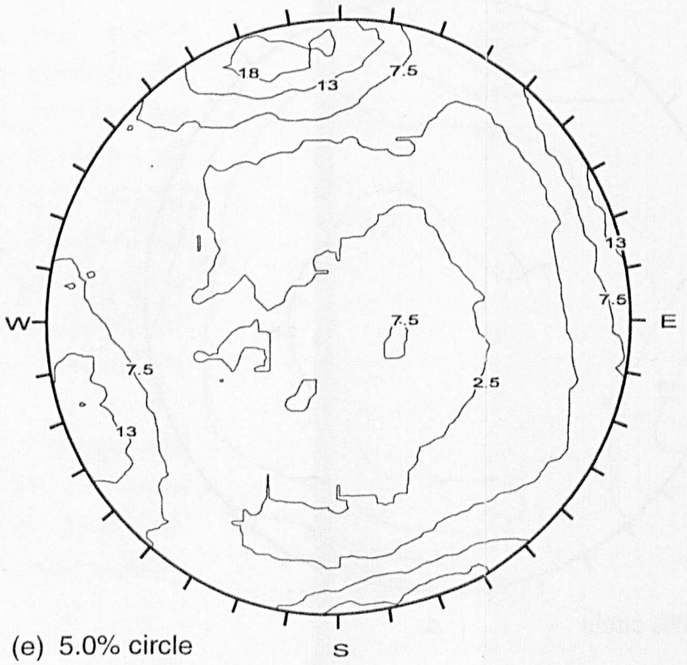
(b) 0.5% circle



(c) 1.0% circle



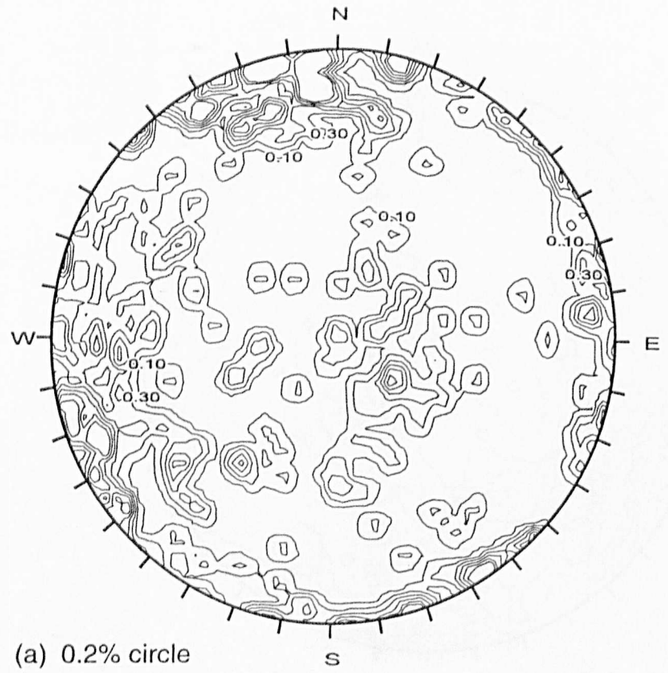
(d) 2.0% circle



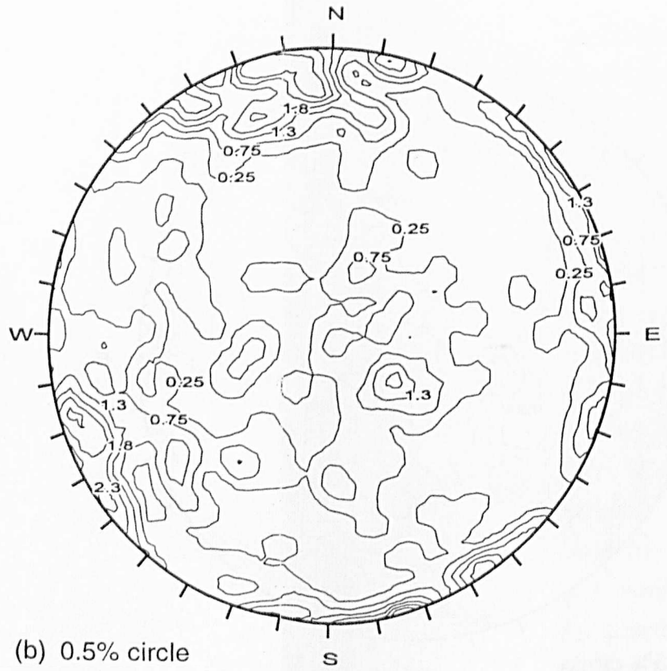
(e) 5.0% circle

	counting circle %				
	0.2%	0.5%	1.0%	2.0%	5.0%
equivalent contours: % of poles in counting circle	0.1	0.25	0.5	1.0	2.5
	0.3	0.75	1.5	3.0	7.5
	0.5	1.25	2.5	5.0	12.5
	0.7	1.75	3.5	7.0	17.5
	0.9	2.25	4.5	9.0	22.5
	1.1	2.75	5.5	11.0	27.5

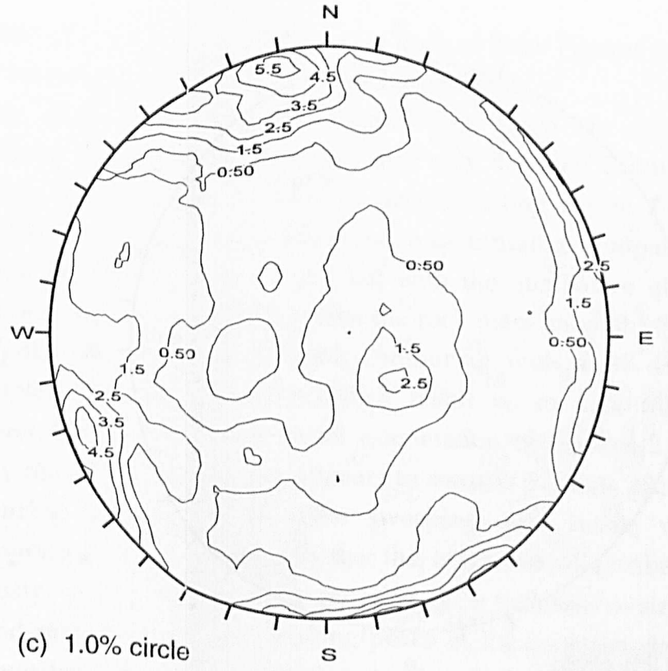
Figure 2.16 Porphyry copper data: upper hemispherical equal area projections of Schmidt contouring (data from Mahtab *et alia* [86]).



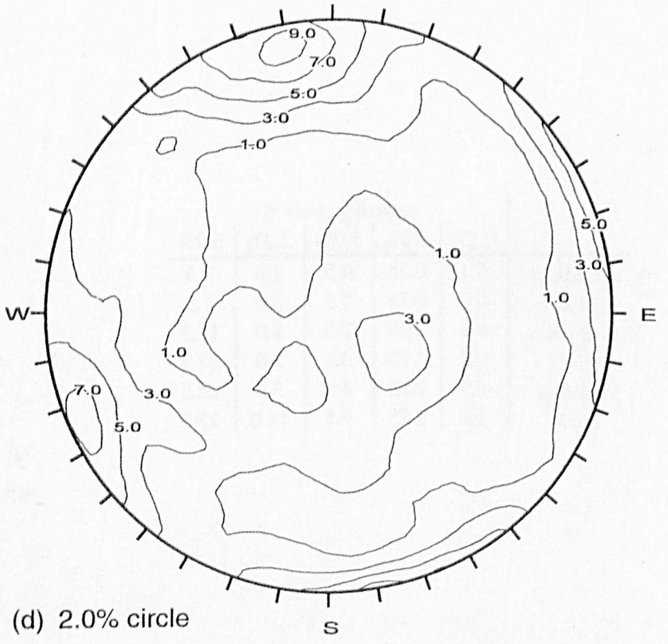
(a) 0.2% circle



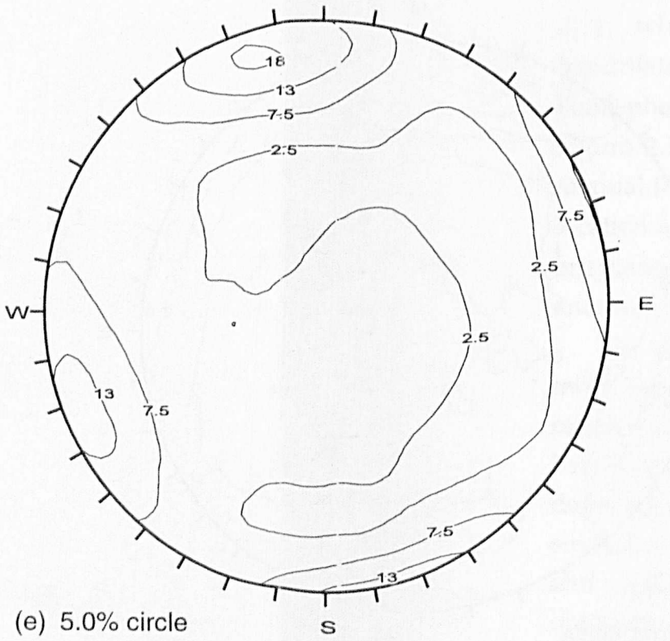
(b) 0.5% circle



(c) 1.0% circle



(d) 2.0% circle



(e) 5.0% circle

	counting circle %				
	0.2%	0.5%	1.0%	2.0%	5.0%
equivalent contours: % of poles in counting circle	0.1	0.25	0.5	1.0	2.5
	0.3	0.75	1.5	3.0	7.5
	0.5	1.25	2.5	5.0	12.5
	0.7	1.75	3.5	7.0	17.5
	0.9	2.25	4.5	9.0	22.5
	1.1	2.75	5.5	11.0	27.5

Figure 2.17 Porphyry copper data: upper hemispherical equal area projections of Fisher contouring (data from Mahtab *et alia* [86]).

Identification of clusters

Even if contouring *per se* could be made more objective, there remains the problem of identifying preferred orientations from the contours. When using contouring methods, the usual procedure is to assume that only densities of discontinuity normals (*i.e.* number of poles per unit area of the hemisphere) above a certain level are significant, and below the contour associated with this value the data are assumed to exist in no preferred orientation. All local density maxima, as revealed by the contouring, are then treated as representative of significant numbers of almost parallel discontinuities, with their mean direction being given by the orientation of the maximal point in the contouring. However, Mardia [90] notes that "these methods do not seem to have been investigated statistically" and, as far as the current Author has been able to determine from the literature, it has never been demonstrated that the orientation of contoured maxima do, in fact, coincide with the mean orientation of discontinuity sets.

The serious inadequacy of this technique is easily judged, through a consideration of the contours that result from the upper and lower bounds to the size of the counting circle. The upper bound—a counting circle that envelops the entire hemisphere—results in an isotropic density value equal to the total number of data divided by the surface area of the hemisphere; the lower bound—a circle with radius less than the arc distance between the two nearest-neighbour vectors in the data set—produces a contoured surface with local maxima corresponding to each datum. This effect is clearly seen in Figures 2.16 and 2.17, where the 0.2% counting circle identifies many small peaks, and the 5.0% counting circle tends to produce an isotropic result. Thus, simply by setting the counting circle to an appropriate size, we will be able to generate any number—between one and the number of data under analysis—of 'preferred orientations'.

This problem of lack of objectivity has been tackled through the application of probabilistic methods. Mahtab *et alia* [86] assumed a Poisson distribution model in the program PATCH to determine threshold density. In this model, the probability that a random density D exceeds a threshold density x is given by

$$\Pr(D > x) = 1 - \sum_{j=0}^{x} \frac{e^{-m} m^{j}}{j!} \qquad (2.03)$$

where m is the average density over the 100 patches. In the program PATCH, the threshold was arbitrarily selected as $\Pr(D > x) \leq 0.05$ but could, of course, have been selected at any level within $0 \leq \Pr(D > x) \leq 1$.

Using this test, PATCH operates by identifying all significant patches, and then agglomerating adjacent significant patches to form clusters. With this method, the porphyry copper data was considered to contain 3 clusters, the mean orientations (dip direction/dip angle) of which were 249/84 (57 data), 348/81 (77 data) and 120/23 (7 data). This result will be referred to later in sub-section 6.3.1.

Cheeney [29] used a different approach to introduce objectivity: assuming that the counting cells are placed randomly over uniformly distributed data, then the binomial coefficient is applicable. Thus, if n uniformly distributed data are plotted, then for a counting cell with an area which is a proportion h of the hemispherical surface area (*i.e.* $0 \leq h \leq 1$), the threshold density D is computed, using the cumulative binomial probability, as

$$\Pr(D) = \binom{n}{D} h^D (1-h)^{(n-D)} \tag{2.04}$$

Interestingly, as the Poisson distribution is a limiting case of the binomial distribution (resulting when $h \to 0$ and $n \to \infty$), which of these two approaches is the most applicable is debatable. However, regardless of the statistical method used, the result is to identify those areas of the projection where significant clustering occurs (*i.e.* densities significantly greater than those that would result from a uniform distribution).

Of course, the application of these tests are not the only approaches to identifying the existence of a cluster. It is likely that, had contouring not been followed, some other test would have been used—for example, the angular separation between nearest-neighbour vectors being less than some specified amount.

It is quite clear, therefore, that we must question the appropriateness of identifying clusters on the basis of significance testing with respect to the magnitudes of peaks developed in the contouring of pole densities. There is little or no objectivity in any step of the procedure, with the result that any preferred orientations identified on this basis must be regarded as entirely arbitrary.

Assessment of cluster validity

Although any clusters identified in orientation data on the basis of contouring may be regarded as being of an arbitrary nature of any, the customary analytical procedure continues with an assessment of their validity. In discontinuity orientation analysis, the application of null-hypothesis testing with respect to the probabilistic distributions of Fisher and Bingham [90] is applied as a matter of course [86, 108, 125] as a means to this end. Extensions have been made by both Alldredge and Mahtab [6] and Grossman [52], who have used the bivariate normal distribution fitted to the projection of data onto that tangent plane of the hemisphere corresponding to the mean vector of a cluster. Alldredge and Mahtab showed how elliptical confidence regions could be drawn on the projection, and Grossman showed how the fit could be tested using the Kolmogorov-Smirnov test. Other spherical distributions exist in the literature (see, for example, Mardia [90] and Upton and Fingleton [135]) but they appear not to have been applied to discontinuity orientation analysis.

Two main arguments can be levelled against the application of null-hypothesis testing using any probabilistic distributions. Firstly, there is no evidence to suggest that the distribution of intra-set orientation of discontinuities is, in fact, probabilistic in nature. Secondly, all distributions have intrinsic and fixed shapes on the sphere: for example, the Fisher distribution is rotationally symmetric, and the Bingham and bivariate normal distributions are ellipto-form with two orthogonal axes of symmetry. Of course, the actual distributions themselves differ, but all demonstrate a basic bell-shape. Consequently, testing against these distributions is not testing for an intrinsic validity, but rather for validity in terms of the shape of the distributions.

A smaller criticism relates to the null-hypothesis testing. As Press *et alia* [106] point out, the chi–square test (χ^2), using the statistic

$$\chi^2 = \sum_i \frac{(N_i - n_i)^2}{n_i} \tag{2.05}$$

is applicable to discrete (or binned) data, with N_i being the observed number of data in the ith bin, and n_i being the expected number according to some known distribution. In contrast, the Kolmogorov-Smirnov (K-S) test uses the statistic

$$D = \max_{-\infty<x<\infty} \left| S_{N_1}(x) - S_{N_2}(x) \right| \tag{2.06}$$

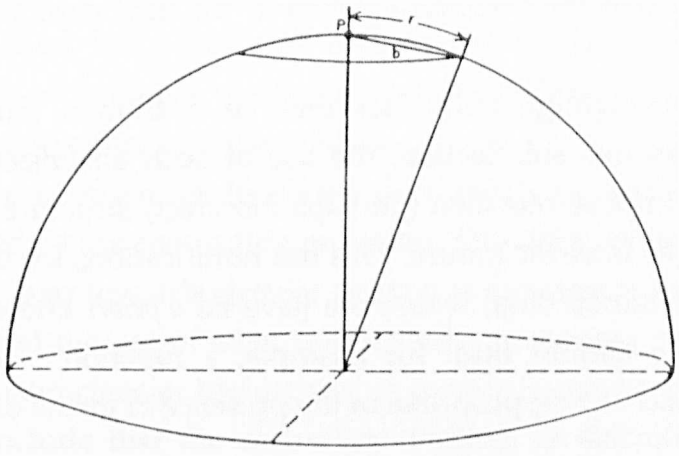
and is applicable to continuous data, with S_{N_1} and S_{N_2} being two different cumulative distribution functions. The argument to be resolved is whether orientations are discrete events (and if so, how are they binned) with the result that the χ^2 test is applicable, or are they continuous (*i.e.* following some distribution) indicating the use of the K-S test?

Notwithstanding the difficulty of determining the correct testing procedure, it is quite wrong to limit the assessment of cluster validity to testing for a fit against only those distributions for which we have statistical apparatus. Such testing amounts not to determining intrinsic cluster validity, but rather testing for validity in terms of an arbitrarily chosen distribution. Indeed, evidence for the inappropriateness of this technique, when applied to assessment of clusters in discontinuity orientation measurements, exists in the literature. In a state-of-the-art-review, Kulatilake [76]—after citing the results of other workers who could not fit probabilistic distributions at the 5% significance level for 18 out of 22 data sets they analysed—concludes that these techniques are inadequate.

Although goodness-of-fit testing is commonplace for orientation data, there is scant evidence of the adoption of techniques that permit assessment of the optimality of the resulting data partition. Such an assessment can be achieved through the use of monitoring the behaviour of an objective function, and is discussed in the next sub-Section.

2.5.2 Objective functions in clustering algorithms

An alternative, and less subjective, technique to contouring as a means of identifying clusters in orientation data is to use an algorithm that incorporates an **objective function**. These functions are chosen such that their magnitude depends on how the data are partitioned, on the basic premise that an optimal partition is indicated by a small number of dense clusters. This approach has been used for the analysis of discontinuity orientation by Mahtab and Yegulalp [88] and Shanley and Mahtab [124, 125, 126] with some, although not widely applied, success. The means by which the data are partitioned remains similar to the counting cell approach: spheres of some arbitrary radius r (see Figure 2.18) are constructed around either set points on the hemisphere [124] or around the data themselves [125, 126]; the number of data falling within the spheres are determined; those spheres containing more than a threshold number of data are labelled 'dense' (*i.e.* considered to represent the location of a



NOTE : Percent of hemispherical surface contained in sphere of radius $b = 2 \sin(\frac{r}{2})$ about the point P is $(1-\cos r)$.

Figure 2.18 Spherical counting cell (from Shanley and Mahtab [124]).

cluster); and, finally, dense points that are less than the sphere radius apart are linked to form clusters. A randomness test based on the Poisson process is used to determine the threshold k, where k is the smallest integer for which

$$\Pr(>k,c) = 1 - \sum_{j=0}^{k} \frac{(cn)^j e^{-cn}}{j!} = c \tag{2.07}$$

where $c = 1 - \cos\alpha$, α=the semi-apical angle of the cone shown in Figure 2.18 and n=number of data in the sample.

The algorithm programmed by Shanley and Mahtab evaluated the objective function

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n(i)} d^2(\mathbf{x}_{ij}, \mathbf{v}_i) + \sum_{i=1}^{c-1} \sum_{j=1+1}^{c} d^2(\mathbf{v}_j, \mathbf{v}_i) \tag{2.08}$$

where $n(i)$=number of data in cluster i, c=number of clusters, $\mathbf{x}$ and $\mathbf{v}$ are datum and centroid vectors, respectively, with $\mathbf{x}_{ij}$ being the jth element of cluster i, and the function $d^2(\cdot)$ represents the square of the Euclidean distance between two vectors. The choice of this function, variously known as the sum-of-squared-distance criterion, the principle of n best centroids and the variance criterion [130], was reportedly subjective, although the authors did state that it performed well in tests, including some undertaken with the porphyry data. However, as is explained further in Chapter 4, this should be regarded as good fortune, because clustering algorithms based on either the partitioning algorithm described or this type of objective function will intrinsically tend to partition the data into circular clusters. The results would

not have been good had the intrinsic clusters in the data conformed to some other shape.

This last point highlights the fact that, in a fashion similar to that discussed in the previous sub-Section, the use of such an objective function means that we will impose our own (perhaps incorrect) structure on the data, rather than identify its intrinsic nature. This has ramifications for the clustering of discontinuity orientation data, where we have no *a priori* knowledge of the shape of clusters we should find: for example, a function biased towards circular clusters would be inappropriate in the presence of girdle distributions.

2.5.3 Summary

The two customary methods of analysing orientation data—contouring on the hemisphere and the use of objective functions—have been discussed in this Section. A number of serious deficiencies have been identified with both ofthese techniques, and these are summarised in the points below.

(i) The Schmidt counting circle approach induces binary membership of a set on the data, with the result that within the counting circle all data are regarded as definite members of a cluster, and outside the circle the complementary case is true. The effect of this is to make the result of the clustering extremely dependent on circle size.

(ii) Changing the circle size will change the number of peaks found in the resulting contours and hence—as these peaks are commonly taken to represent evidence of preferred orientation—the number discontinuity sets identified. As the size of the counting circle increases, so the contouring becomes more isotropic, with the result that the number of sets identified is inversely proportional to circle size.

(iii) The mean orientation of a discontinuity set is commonly taken to be indicated by the orientation of the corresponding maximum in the contours. There is no statistical evidence to suggest that this is the case.

(iv) Identification of clusters is usually performed on the basis of the application of probabilistic tests derived from Poisson or binomial distributions. These tests require a level of significance to be chosen and, given that these tests are applied to the results of a subjective contouring scheme, there is no objective means of determining this threshold.

(v) Statistics relating to the validity of a cluster are ordinarily derived from null-hypothesis testing against spherical probabilistic distributions, particularly those of Fisher and Bingham. However, the particular shapes

possessed by these distributions means that intrinsic validity is not being determined, but rather a fit with respect to the distributions themselves. There is no evidence to suggest that discontinuity orientation data should conform to either of these models.

(vi) The use of objective functions does much to overcome the subjectivity imposed by the contouring processes. However, in the schemes developed to date, bias towards circular clusters is inherent in the objective functions used, and the use of what is effectively a counting circle approach means that the deficiencies highlighted in points (i) and (ii) above will still exist.

We may conclude that the customary analysis of discontinuity orientation is inherently subjective and consequently deficient. If the analysis of this geometrical attribute is to improve, an objective analytical procedure that overcomes these deficiencies must be sought.

2.6 Synopsis

As this review has shown, there are a great many inadequacies associated with the current methods of measuring and analysing rock mass geometrical attributes. It is appropriate to assemble these findings in this Synopsis, to aid identification of procedures that, with suitable development, will allow improvements to be made.

Commencing with the measurement of discontinuity geometrical attributes, the customary procedures commonly call for a human observer, working at an exposure with simple instruments (for example, tape measure and compass), and measuring the attributes directly. Not only does this method require the observer to work in potentially hazardous locations, it also raises questions about the objectivity of the measurement process. For example, the direct use of a human observer at a rock exposure means that the choice of sampling location is not totally free and unbiased, because if human access is required, sampling locations at rock exposures will be limited to those that are no more than 2m from the base of the exposure. Consequently, the vast majority of a rock exposure or rock mass will not be surveyed. Although scanlines or area sampling strategies are suggested as a means of overcoming bias and subjectivity, they introduce their own bias when direct human access is required. Although photogrammetry figures in most of the ISRM Suggested Methods relating to the measurement of geometrical attributes, little use is made of this procedure, probably due to a perceived need for specialised

techniques and equipment. This perception probably stems from the fact that the Suggested Methods do not reference modern photogrammetric procedures. However, because photogrammetry is not biased in position or orientation on an exposure, and furthermore is a remote and hence safe method, further consideration should be given to its development in the field of rock mass geometry.

Moving on to the analytical procedures, it is interesting to note that, although there is great interdependence between the geometrical attributes of discontinuity spacing, surface roughness, extent and orientation, they are customarily analysed independently, with essentially different techniques applied to each. Various forms of probabilistic methods figure highly in the analysis of spacing and extent, but notwithstanding their widespread use, they are not universally applicable. The results of a number of workers have been cited in this Chapter, the common thread between them being that probabilistic techniques must be applied with great care, and abandoned if they appear inappropriate.

Roughness is customarily analysed through the application of deterministic methods that characterise the geometry of a discontinuity surface. Although these methods are, in themselves, satisfactory, they are used with data from linear profiles and, as the appropriate ISRM Suggested Method notes, this should only be done in the particular circumstances when the direction of inter-block movement is known. In the general case, analysis should be undertaken using three-dimensional methods. However, the lack of suitable measurement and quantitative analysis methods for the three-dimensional case prevents this being done. To improve matters, contemporaneous development of readily applicable measurement methods and three-dimensional analytical procedures is required.

Finally, the analysis of orientation has been found to be deficient in a number of aspects. Neither of the two customary methods for analysing orientation data—contouring on the hemisphere and the use of objective functions—are rigorously correct. Subjectivity is implicit in both methods, procedures are used that are without mathematical foundation, and probabilistic models, the appropriateness of which to discontinuity orientation data has not been demonstrated, are customarily used to both identify and quantify preferred orientations.

Thus, we can see two possible approaches for the improvement of the analysis of rock mass geometry. Firstly, the use of modem photogrammetric

techniques may allow development of measurement techniques that will both remove the bias imposed by human observers, and allow acquisition of three-dimensional data that currently is unavailable. Secondly, mathematical procedures should be sought that do not unnecessarily impose probabilistic forms on rock mass geometrical data, whilst permitting more sophisticated and objective analysis than is currently possible.

The remainder of this thesis is dedicated to the application of photogrammetry and cluster identification schemes to the measurement and analysis of rock mass geometry, with particular emphasis being placed on discontinuity orientation. In Part II following, these fields are studied in detail, such that appropriate means of applying them to the problem may be identified.

This page intentionally left blank

Part II
Fundamental Techniques

This page intentionally left blank

3 Photogrammetry

The discussion in Chapter 2 illustrated how current methods for the acquisition of rock mass geometrical data are inadequate and can require personnel to work in hazardous locations, *e.g.* at the foot of unstable rock faces. To overcome these problems a rapid and remote data acquisition technique is required, and modern photogrammetry was identified in Chapter 2 as a possible candidate. The aim of this Chapter is study photogrammetry in detail and identify those aspects that may usefully developed in the context of determining rock mass geometry.

Photogrammetry has been defined [143] as 'the art, science and technology of obtaining reliable information about physical objects...through processes of recording [and] measuring...photographic images...'. It is without doubt a remote technique and, providing it is sufficiently fast and accurate, may be applicable to rock mass geometrical data acquisition. The question to which we require an answer is, do photogrammetric equipment and techniques exist that, with suitable development, will permit such an application to be made, whilst obeying the four principles laid down in Section 1.2? These principles are worth reiterating, with the first three being: universal (and hence, flexible) in application, straightforward in use, and comprising equipment that is of reasonable cost and readily available. The fourth principle—that a new method should be an advance on those customarily used—has not been included in this list on the understanding that it will automatically be upheld.

It is essential to consider these three principles in the context of photogrammetry, if it is to be studied for suitable equipment and techniques. This is done in the following points.

(i) **Applicability.** The system is to be used initially for research in the field of rock mass geometry, although if successful it may find wider application. It therefore needs to be able to cope with the whole range of possible circumstances under which it may be used.

(ii) **Accuracy and precision.** Although the surveying contemplated will be of relatively low standard (in comparison to cartographic standards), data are required that possess errors low enough to permit meaningful analysis. It is important that measurements of discontinuity surface orientation and extent should be at least equal in accuracy to those obtained by customary techniques involving a geological compass and tape measure.

(iii) **Cost.** In connection with flexibility, if the system has to be changed or upgraded, this must be done at minimum cost. Similarly, if the system is to become acceptable to a wider range of users, it must be affordable.

(iv) **Availability.** Components should be readily available (ideally off-the-shelf), in order that the concepts of high flexibility and low cost may be supported.

Thus, the aim of the review presented in this Chapter is to identify those elements within the field of photogrammetry that will allow—with a reasonable amount of development—the principles outlined above to be achieved.

This Chapter consists of five Sections. The first of these gives an overview of photogrammetry in general, so that the specialised terminology may be introduced. By considering the various facets of the specific areas of photogrammetry, so-called non-metric photogrammetry is identified as a suitable area for further investigation. This subject is discussed in detail in Section 3.2, with particular reference to the equipment and data processing techniques used. As a result of this discussion, the Direct Linear Transform is revealed to be the most suitable data processing technique, and this is in turn then examined, in Section 3.3. At this stage, having identified—with a view to an application in the study of rock mass geometry—promising avenues for development, it is useful to consider what previous photogrammetric work has been undertaken in rock mechanics. This is the subject of Section 3.4. Finally, a synopsis in given in Section 3.5.

3.1 An Overview Of Photogrammetry

Photogrammetry is applied in a great many fields, and as a consequence is extremely rich in equipment and techniques. This richness has resulted in the development of a specialised nomenclature that, whilst necessary for succinctness, can be confusing to the point of obscuring the underlying

principles. It is helpful, therefore, to introduce some of this nomenclature in terms of these principles, whilst simultaneously attempting to identify those aspects of photogrammetry that may be of use in the field of rock mass geometry.

The following sub-Sections deal with four fundamental elements of photogrammetry: the cameras used; how the resulting photographs are viewed; how measurements are taken from these photographs; and how the customary distinction between applications has affected the development of the science. Finally, sub-Section 3.1.5 summarises the position in terms of photogrammetry for rock mass geometry.

3.1.1 Cameras

Photographic imagery may be obtained by means of either **metric** or **non-metric** cameras. Technically, the distinction between metric and non-metric cameras is made [41] in terms of their **interior orientation**, the term used to describe the parameters that model the passage of light rays through the lens and onto the film. The four parameters **principal distance, principal point, radial lens distortion** and **decentering lens distortion** are used to define interior orientation, and great care is taken in the design and construction of metric cameras to ensure these parameters are both quantified and constant. A description of these parameters is presented in the following numbered points, partly to elucidate the differences between metric and non-metric cameras, and partly as background for the remainder of this Chapter.

(i) The principal distance is the perpendicular distance from the perspective centre of the lens to the film plane. For a perfect camera, modelled as a pin-hole camera, the principal distance is as shown in Figure 3.01, and is customarily denoted by the symbol c. For a lens system possessing variable focus, the principal distance is also variable, with the focal length of the lens being the principal distance when the lens is focused at infinity.

(ii) The principal point is that location in the film plane where an image due to a ray passing directly through the centre of the lens would form if the focal plane of the camera were perfectly perpendicular to the ray itself, as shown in Figure 3.01. In a metric camera, the principal point and the coordinate origin coincide, *i.e.* $x_p = 0$ and $y_p = 0$; this is generally not the case with a non-metric camera. The photograph coordinate system shown

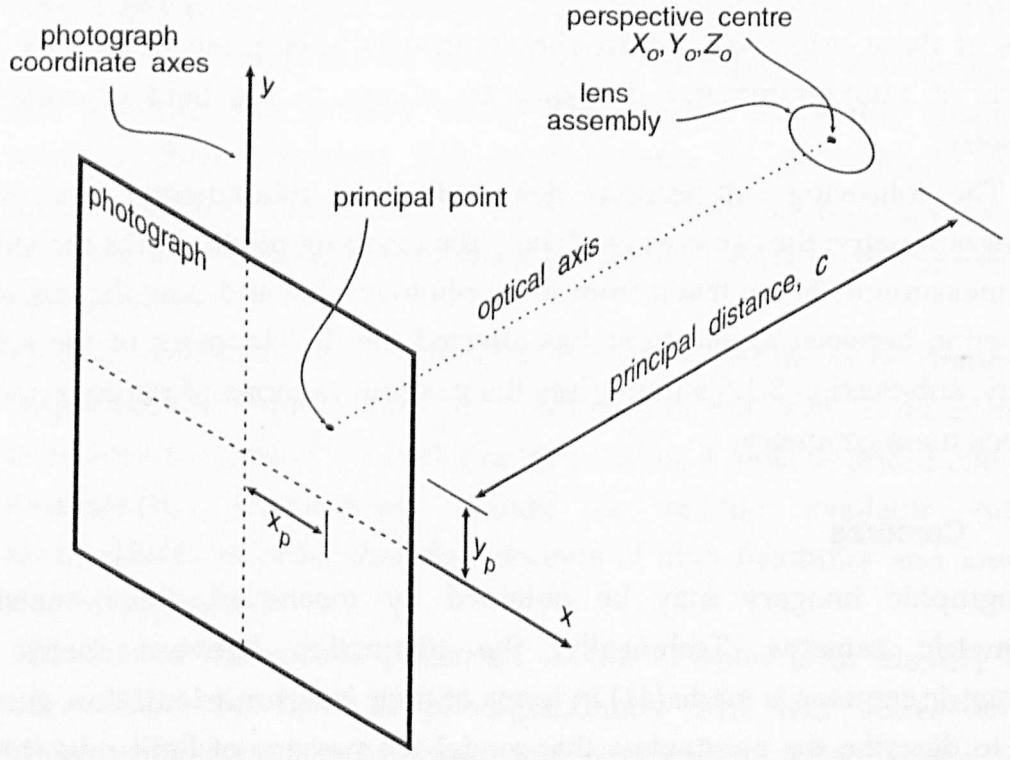


Figure 3.01 Principal distance and principal point for a perfect pin-hole camera.

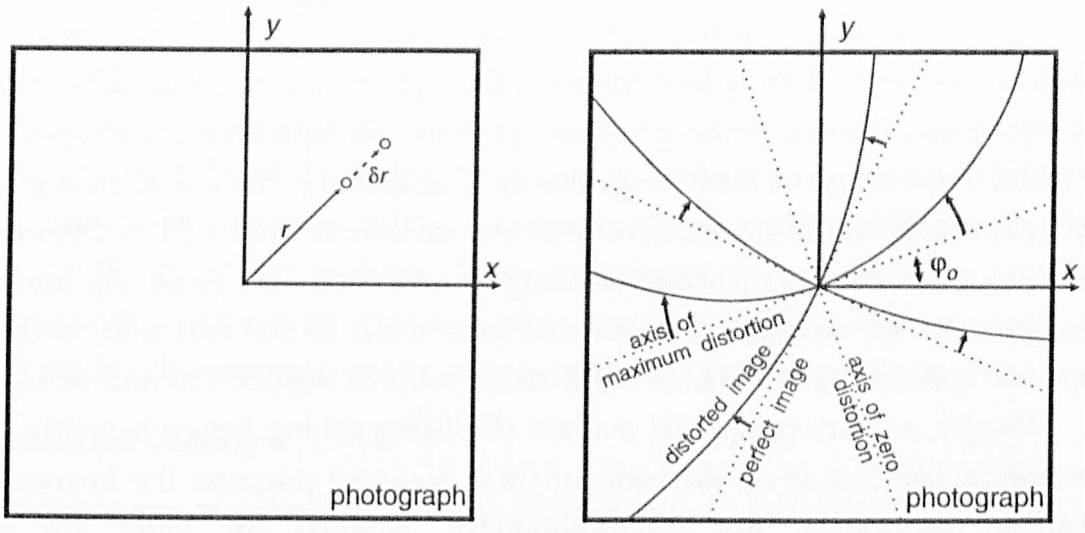
in Figure 3.01 is arranged such that the origin coincides with the centre of the photograph itself. Fiducial marks—which appear at the corners of the photograph—are built into a metric camera, such that the intersection of diagonal lines drawn between them locates the principal point.

(iii) If the image formed by an off-axis target is not in the position that a perfect lens would produce, but is either radially closer to or farther away from the principal point, then the image suffers from radial distortion, as shown in Figure 3.02. The magnitude of the distortion at a radius r from the principal point is usually adequately accounted for by the first three terms of the odd-powered polynomial series [7]

$$\delta r = K_1 r^3 + K_2 r^5 + K_3 r^7 \qquad (3.01)$$

where the coefficients K are determined by calibration. The magnitudes of these coefficients in metric cameras are generally much smaller than in non-metric cameras.

(iv) Decentering lens distortion is due to imperfect centring of the lens components and other manufacturing defects. Figure 3.02 shows how decentering distortion appears on a photograph. In terms of the angle φ_o,



(a) radial distortion (b) decentering distortion

Figure 3.02 Radial and decentering lens distortions.

shown in the Figure, the x- and y-axis components of this distortion may adequately be expressed using the first three terms of the even-powered polynomial series [7]

$$\left.\begin{aligned}\Delta x &= \left[P_1\left(r^2+2x^2\right)+2P_2xy\right]\cdot\left[1+P_3r^2+P_4r^4\right]\\ \Delta y &= \left[2P_1xy+P_2\left(r^2+2y^2\right)\right]\cdot\left[1+P_3r^2+P_4r^4\right]\end{aligned}\right\} \qquad (3.02)$$

where, as with radial lens distortion, the various coefficients P are determined by calibration.

Whilst metric cameras are designed and built to ensure the interior orientation remains constant, and are then calibrated in order to quantify these four parameters, in the case of a non-metric camera the interior orientation may be unstable, and hence cannot generally be quantified through calibration.

A further distinction between metric and non-metric cameras may be made on the basis of film deformation. This can be considered as comprising components of uniform distortion, differential distortion and irregular distortion. Uniform distortion represents a uniform shrinkage or expansion, and can be corrected for by introducing a scale factor. Differential distortion results when the degree of film shrinkage or expansion along the length of the film roll is different from that across the width of the film roll. This gives an **affine deformation** (*i.e.* different scales in different directions) of the photograph, which can be corrected using an **affine coordinate transformation**. Finally, irregular distortion—which is due to causes such as

variations in the elastic properties and lack of flatness of the film—cannot be corrected for, and can only be minimised through the use of a **réseau** (a grid of accurately coordinated points projected onto the photograph during exposure) and film tensioning and flattening devices. Traditionally, the use of glass plates rather than roll film has been advocated as a means of reducing film distortion, with the result that metric cameras using flexible film are a relatively modern development. Of course, a camera that uses small format roll film, with no means or controlling film tension or flatness, must be regarded as non-ınetric.

Finally, a more pragmatic method of distinguishing between metric and non-metric cameras is on the basis of their designed purpose: the former are designed specifically for photogrammetry, whereas the latter are not. Accompanying this distinction, non-metric cameras possess a series of advantages and disadvantages with respect to metric cameras, and these are summarised in Table 3.01.

We can see from Table 3.01 that flexibility, availability and price dictate that a non-metric camera must be considered for this work. The question that this poses is, can non-metric photogrammetry produce data of the required standard? This question is probed further in Section 3.2, but the subject of viewing the resulting photographs and taking measurements from them is considered first.

Table 3.01 Advantages and disadvantages of non-metric cameras with respect to metric cameras (after Karara [68]).

Advantages
• General availability. • Flexibility of focusing range. • Can be hand-held and thereby oriented in any direction. • The price is considerably less than for metric cameras.
Disadvantages
• The lenses are designed for high resolution at the expense of high distortion. • Instability of interior orientation. • Lack of fiducial marks. • Absence of level bubbles and orientation provisions precludes the determination of exterior orientation before exposure.

3.1.2 Viewing photogrammetric images

Photogrammetry is primarily used to determine measurements of some attribute of the geometry of the object. This can only be done by taking two or more photographs of the object, from known locations, and then applying the basic principles of triangulation to measurements taken from the photographs to solve the geometric problem. These measurements may be taken from one photograph at a time, or from two photographs simultaneously. As such, the viewing is either **monoscopic** or **stereoscopic**, and these are described below.

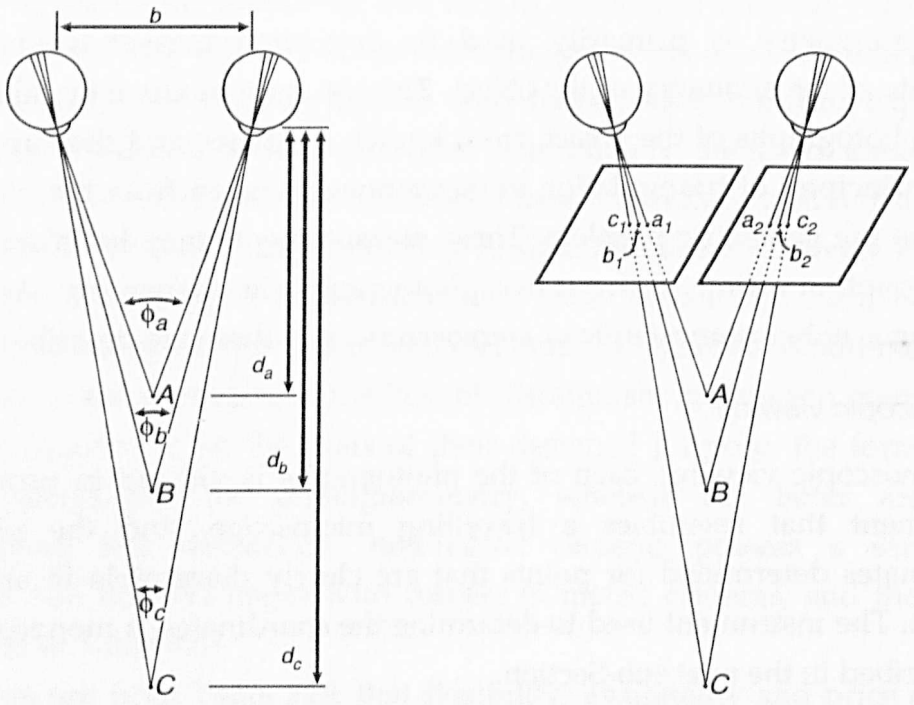
Monoscopic viewing

In monoscopic viewing, each of the photographs is viewed in turn using an instrument that resembles a travelling microscope, and the photograph coordinates determined for points that are clearly discernible in at least two images. The instrument used to determine the coordinates, a monocomparator, is described in the next sub-Section.

Monoscopic viewing has one main deficiency associated with it: humans are not able to gain much perception of distances to objects, or depth, from monocular vision. Monoscopic perception of depth is limited to the relative sizes of objects, hidden objects, shadows and differences in focusing [143]. Of course, on a photograph only the first three of these techniques can be used. Because humans usually use binocular vision as a means of determining depth (this is described in the next Division), when limited to the equivalent of monocular vision (through the use of monoscopic viewing) we find that the associated difficulty of determining depth means that identification of homologous points in two or more photographs becomes very difficult. It is therefore not easy to undertake the photogrammetric process using monoscopic viewing, and so stereoscopic viewing is preferred.

Stereoscopic viewing

When the eyes focus on an object, the optical axes of the two eyes converge and intersect at the object, subtending an angle known as the **parallactic angle**, as shown in Figure 3.03(a). The observer can perceive depth because the brain interprets parallactic angle as distance to the object. Now, if two photographs are taken of an object from slightly different locations and subsequently placed in front of an observer such that each eye can see only one photograph—as shown in Figure 3.03(b)—, then the brain immediately interprets the scene in terms of binocular vision, and once again determines depth in terms of



(a) depth in binocular vision is perceived in terms of parallactic angle

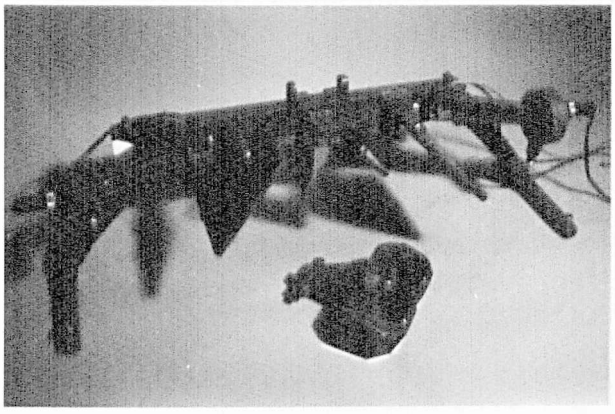
(b) position of an object on a photograph is partly a function of depth

Figure 3.03 Binocular vision and stereoscopic photographs.

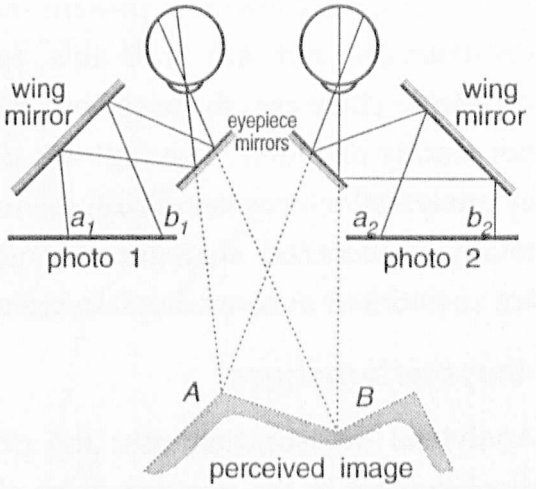
parallactic angle. As a result the observer perceives a stereoscopic image, known as a **stereomodel**; two photographs that give rise to this phenomenon are called a **stereopair**.

Comfortable stereoscopic viewing requires a **stereoscope**, an optical instrument that permits each of the observer's eyes to view one of the photographs independently, whilst allowing the eyes to focus at a distance that is commensurate with the perceived parallactic angles in the stereomodel. Many designs of stereoscope exist, from simple devices consisting of a pair of lenses in a frame, up to sophisticated prismatic models. One of the commonest types is the **mirror stereoscope**, pictured in Figure 3.04(a). The optical operating principle of the mirror stereoscope is illustrated in Figure 3.04(b), and this shows how the instrument provides for comfortable viewing, whilst simultaneously accommodating much larger photographs than could be simply placed in front of the eyes.

The stereoscopic visual cues that are received by the brain as a result of using a stereoscope are very powerful, allowing an observer to clearly perceive a stereomodel. As it is also possible to use a stereoscope for photograph



(a) mirror stereoscope



(b) optical operating principle of the mirror stereoscope

Figure 3.04 The mirror stereoscope.

coordinate measurement (this is described in the next sub-Section) stereoscopic viewing is generally the preferred technique in photogrammetry.

3.1.3 Photogrammetric measurements

Photogrammetric measurements may be taken by either **analogue** or **analytical** means. These two techniques are fundamentally different, and as such are described separately in the following two Divisions.

Analogue techniques

Analogue mensuration is only possible with metric photography of known **exterior orientation** (this is defined in terms of the global Cartesian coordinates of the camera perspective centre and its orientation with respect to the global Cartesian axes), as it relies on the principle that two such photographs, when correctly and accurately positioned to represent the original camera positions, allow an observer to restitute a stereoscopic model that is a small-scale analogue of the original object. An **analogue plotter**—an instrument based on a stereoscope but designed as an analogue computer—is used to correctly position the photographs relative to each other such that they can then be used to directly plot a correctly projected two-dimensional scale map of the corresponding three-dimensional object.

These instruments represent the zenith of analogue computer design and construction, but are inflexible (only metric photography can be used), expensive (they are composed entirely of precision engineered parts) and hence not readily available. The first two of these disadvantages have been overcome by analytical instruments, with the result that analytical techniques have almost totally superseded analogue techniques. Consequently, analytical techniques are considered in some detail in the next Division.

Analytical techniques

Analytical mensuration uses the principal of **parallax**. This is the apparent displacement in the position of an object, with respect to a frame of reference, when the position of an observer changes. Figure 3.03(b) shows how, for two observation positions represented by the left and right photographs, the positions of the images of objects *A*, *B* and *C* change relative to both themselves and the photographs. Because this parallax occurs parallel to the *x*-axis of the photograph coordinate system, it is known as ***x*-parallax**.

It is clear from the geometry of Figure 3.03(b) that, providing the relative positions of the cameras and their principal distances are known, it is straightforward to compute the relative locations of the objects from measurements made of their parallaxes. Indeed, providing the camera inner orientations and the **object space coordinates** (*i.e.* coordinates in a three-dimensional Cartesian system that is fixed relative to the object) of the camera perspective centres are known, it is possible to compute the object space coordinates of any point that appears in two or more photographs. This requires the parallax equations, which are derived from the geometry of two stereoscopic photographs—as shown in Figure 3.05—on the basis of similar triangles. The parallax equations are

$$Z_a = H - \frac{Bf}{p_{x_a}} \tag{3.03}$$

$$X_a = B\left(\frac{x_a}{p_{x_a}}\right) \tag{3.04}$$

$$Y_a = B\left(\frac{y_a}{p_{x_a}}\right) \tag{3.05}$$

where p_{x_a}, the *x*-parallax of point *A*, is given by $p_{x_a} = x_a - x'_a$.

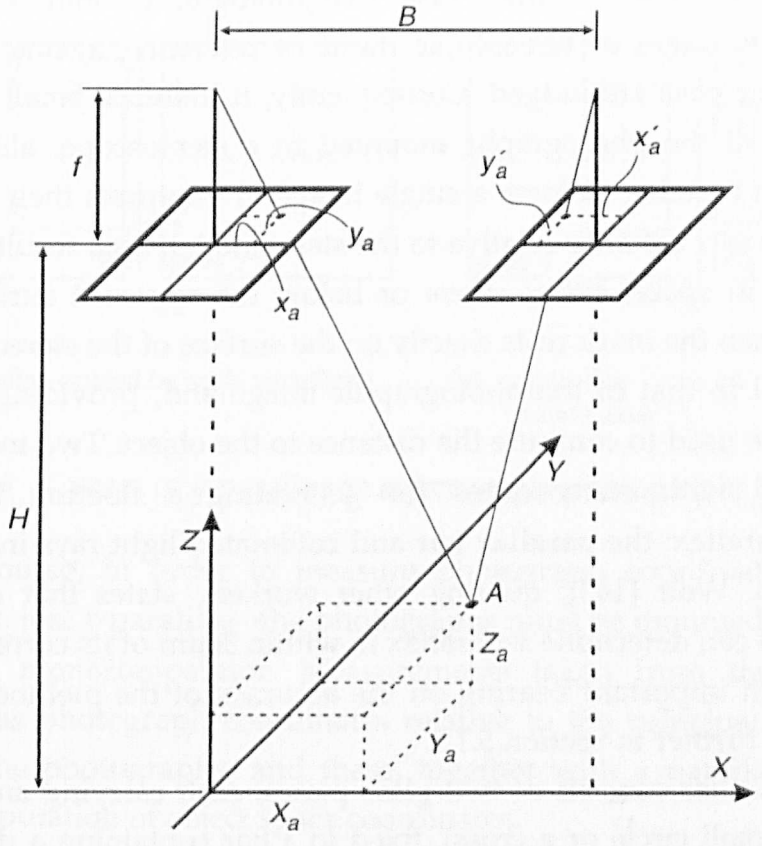


Figure 3.05 Geometry of stereoscopic photographs.

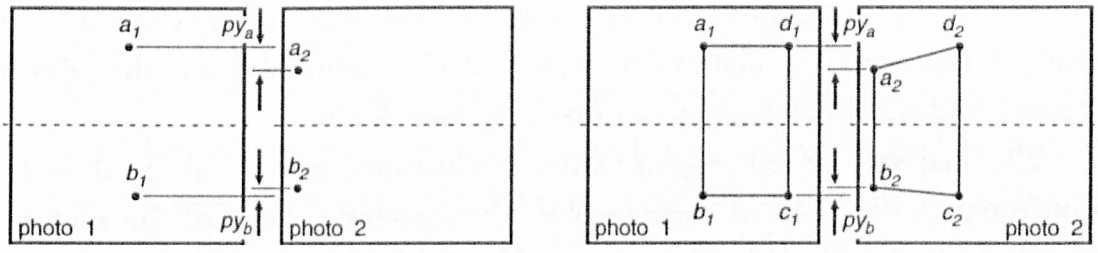
If monoscopic viewing is being used, it is simply a matter of measuring (in at least two photographs), the photograph coordinates relative to the principal point of an image point, and then computing the parallax of the point and hence the object space coordinates. The instrument used for the measurement is a **monocomparator**, which consists simply of a low-power microscope (magnification in the range 10× to 40×) mounted over a leadscrew-driven *x*-*y* stage that carries the photograph. In order that photograph coordinates can be determined, the leadscrews in the *x*-*y* stage are equipped with either micrometers or electronic digital encoders. Although commercial monocomparators are generally able to resolve distances of 1μm, the difficulty of locating homologous points in a pair of photographs, together with the fact that two complete cycles of measurement are required (one for each photograph), means that they are neither easy nor fast to use for photogrammetry. The use of stereoscopic viewing overcomes these deficiencies.

Parallax determination whilst viewing photographs stereoscopically makes use of the principle of the **floating mark**. The visual intensity of the

stereomodel formed when the brain fuses two photographic images together is so great that it becomes a stereoscopic frame of reference, against which all other stereoscopic cues are judged. Consequently, if identical small marks are placed on each of the photographs mounted in a stereoscope, although the brain fuses them together to form a single image, it interprets their associated parallactic angle as a distance relative to the stereomodel. As a result, the mark seems to 'float' in space, either above or below the apparent surface of the stereomodel. When the mark rests exactly on the surface of the stereomodel, its parallax is equal to that of the photographic image and, providing it can be measured, can be used to compute the distance to the object. Two methods are commonly used with stereoscopes for generating a floating mark and measuring its parallax: the **parallax bar** and collimated light rays injected into the optical train. Wolf [143], quoting other workers, states that reasonably skilled observers can determine x-parallax to within 30μm of its correct value, a result that has an important bearing on the accuracy of the method, and one that is discussed further in Section 5.1.2.

The parallax bar consists of two glass plates, each carrying an inscribed mark (either a small circle or a cross), fixed to a bar containing a micrometer assembly. The two plates lie on the photographs, and by adjusting the micrometer the floating spot is brought to rest on the surface of the stereomodel. Some initial calibration of the bar is required, so that parallaxes relative to the principal points of the photographs are measured, but following that it is a quick and easy method of determining distances to objects.

The use of collimated light rays injected into the optical train is more versatile, as no equipment lies on the photographs to possibly distract the observer. A simple method of injecting the rays into a mirror stereoscope is to direct small collimated beams of light from behind half-silvered wing mirrors, parallel to the plane of the photographs, towards the eyepiece mirrors. This results in the observer seeing a spot of light 'floating' in the stereomodel. One method of varying the parallax of the light spot, and hence its apparent height above the stereomodel, is to change the height above the plane of the photographs of one of the light sources. This is simply achieved by means of a micrometer, with the result that a direct reading of the parallax is also available. An alternative approach, and one that is more convenient to implement in certain instruments, is to change the separation of the photographs; the effect is the same.



(a) y-parallax caused by scale variation

(b) y-parallax caused by non-parallel camera axes

Figure 3.06 Causes of y-parallax in stereoscopic photographs

Of course, in order to measure photograph coordinates completely—rather than just x-parallax—the photographs must be mounted on an x-y stage, as in the monocomparator. Measurements taken from the x-y stage are regarded as photograph coordinates relative to the principal point of one or other of the photographs, and these, together with a parallax measurement, allow computation of object space coordinates.

If the photographs are either of slightly different scale or not carefully aligned, or if the original camera axes were not parallel, then **y-parallax** will exist in the stereomodel, as shown in Figure 3.06. Depending on the magnitude of the y-parallax, stereoscopic viewing will either be uncomfortable or impossible, factors that mean it is important for it to be removed. This is simply done by moving one of the photographs in the y-direction relative to the other, and so the x-y stage of a stereoscope usually incorporates the facility for this. Although the existence of y-parallax does not affect customary aerial photogrammetric measurements of elevation, it may not be determined in these circumstances. However, when complete coordination of a point is being attempted, the **coplanarity condition** (*i.e.* the condition when the object, the two camera perspective centres and the two images are coplanar in space) is important and, as it is possible to compute the amount of y-parallax present at any point in a stereomodel, some means of accurately removing it is provided (for example, a vernier gauge and adjusting screws).

There are many methods by which the principles of stereoscopy may be implemented in analytical instruments, and they tend to be categorised in terms of this. This subject is covered in some detail in Section 3.2.

3.1.4 Field of application

Traditionally, photogrammetric applications have been categorised as either **aerial** or **terrestrial**; a distinction that probably resulted from the different accuracy and equipment requirements of the two fields.

The majority of photogrammetric applications are aerial, as it is used extensively in the field of cartography. Consequently, most of the literature, equipment and techniques are applicable primarily to this field. The end product of cartographic photogrammetry has tended to be a printed map, with the result that analogue techniques have been developed to permit contours and physical features to be traced directly onto a map from the photographs. In turn, the development of these analogue techniques has been aided by the geometric simplifications arising from horizontal flying and simple, fixed-focus, metric cameras. Thus, analogue plotters have tended to produce only graphical output, and as they are capable of handling only small angular and scale changes between photographs, immense care must be taken in obtaining the photography. It is fair to say that aerial photogrammetry has the highest accuracy requirements of customary applications, and not only does this require great attention to detail in the planning and execution of a survey project, it has also led to the development of sophisticated precision equipment.

Historically, terrestrial photogrammetry has been a more difficult undertaking, and consequently has not been as widely practised as aerial photogrammetry. Under normal circumstances, the cameras will not be focused at infinity and, unless the camera stations are accurately positioned, there is no guarantee that the camera axes will be parallel, or that the photograph scales will be equal. These factors can lead to large angular and scale changes between photographs and, taken in conjunction with the fact that terrestrial photogrammetry is more likely to be performed in order to determine spatial measurements than to produce a physical contour map, produce conditions that do not facilitate solution by analogue plotters.

Equipment has been developed to overcome these difficulties. Probably the most important is the **phototheodolite**, an instrument combining a metric terrestrial camera and a theodolite. This instrument allows the camera stations to be carefully positioned and the camera axes to be correctly oriented, such that the resulting photography possesses angular and scale changes that are within the limits imposed by analogue plotters. Adding digital output devices to the plotters themselves then allows object space coordinates to be determined directly. The limitations of photogrammetry undertaken in this

way are lack of flexibility in camera positions and orientations, and slow speed (neither obtaining the photography with care, nor manually reading coordinates from an analogue plotter, are rapid procedures).

With the advent of analytical instruments and techniques, the necessity for the distinction between aerial and terrestrial applications has reduced. There is still equipment that is only appropriate for one of the categories—an example is the fixed-focus metric aerial camera—but photogrammetry seems to be moving more towards equipment and techniques chosen on the basis of output requirements, rather than on field of application.

3.1.5 Summary

This overview has been presented in terms of the cameras, the methods of viewing and taking measurements from photographs, and the historical distinction between aerial and terrestrial photogrammetry.

Cameras are distinguished on the basis of being metric or non-metric, the optical geometry of the former being much more highly refined than the latter. Accompanying this distinction are aspects of cost, flexibility and availability: metric cameras are more expensive, less flexible and less readily available than non-metric cameras.

The resulting photography may be viewed monoscopically or stereoscopically. Stereoscopic viewing makes use of the powerful visual cues associated with parallax, which allow an observer to readily determine location of points in space, and as such is generally preferred to monoscopic viewing. Stereoscopic images formed from metric photography can be plotted directly onto two-dimensional media to produce maps and projections, although greater flexibility results from the direct measurement of parallax. This is achieved through the use of a so-called floating mark, with real-world coordinates of points being easily computed from parallax measurements.

Thus, it is apparent from this overview that the availability of equipment and techniques in any one of these areas of photogrammetry has affected the development of techniques and equipment in the others, and this has been driven by the requirements of the largest field of application, namely aerial photogrammetry for cartography. It is useful to summarise this in graphical form, as shown in Figure 3.07.

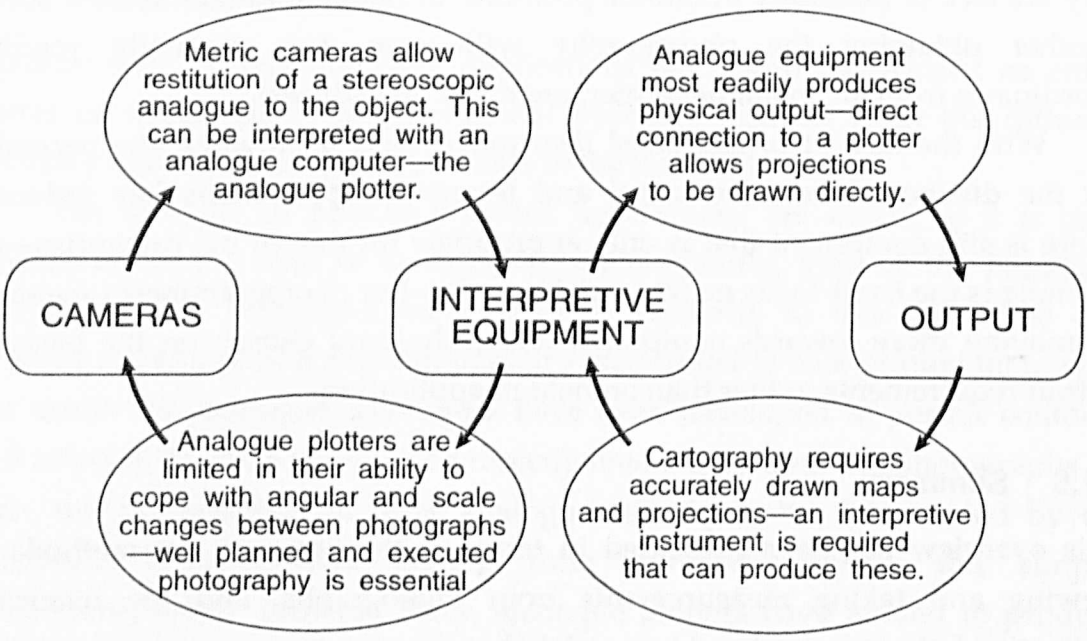


Figure 3.07 Interactions between equipment and techniques for aerial photogrammetry.

This Figure shows how it is impossible to break out of this series of interrelations if only one of the three main elements—cameras, interpretive equipment or output—is changed. However, as the previous sub-Sections have shown, the basic principles of photogrammetry are straightforward: take two photographs of an object from different locations; view them stereoscopically; whilst doing this take parallax measurements of points on the object; and from these measurements compute real-world coordinates and dimensions.

It is beneficial to consider photogrammetry in this light, when examining it in the context of rock mass geometry. Under these circumstances, we are freed from the constraints of cartographic accuracy and precision, but have the additional constraints of flexibility, low cost, and ready availability placed upon us. The only possible strategy becomes clear: use a non-metric camera together with a simple stereoscope, measure parallax directly and at a resolution suitable for the task in hand, and use a desk-top computer to determine real-world coordinates. This is the field of **non-metric analytical photogrammetry**, and a detailed discussion of it forms the next Section.

3.2 Non-Metric Analytical Photogrammetry

The need for flexibility, availability and reasonable cost dictates that non-metric analytical photogrammetry is the only technique applicable to the analysis of rock mass geometry. However, the availability and efficacy (in terms of *required* versus *obtainable* accuracy and precision) of the various elements of a photogrammetric system must be considered. The following sub-Sections address this in terms of **non-metric cameras**, **data processing techniques** and **instruments for photograph coordinate measurement**.

3.2.1 Non-metric cameras

Karara [66, 67] has reviewed the use of non-metric cameras in close range and industrial photogrammetry, and concludes that the use of such cameras should not be dismissed out of hand, but rather "such cameras should be used only if the accuracy requirements permit". Adams [3] performed accuracy tests using non-metric cameras, and concluded that accuracies better than 1:1000 (defined as the ratio of mean square error in camera-to-object distances to actual camera-to-object distances) were readily achievable. This can be placed in the context of photogrammetry for rock mass geometry, where a camera-to-object distance of, say, 20m will result in positional errors of 20mm or less. The current Author contends that determining point positions on rock faces to such an accuracy is more than adequate for the purposes of analysing rock mass geometry: indeed, an extensive numerical simulation of this photogrammetric problem is presented in Section 5.1, and the results concur with this.

A major problem with non-metric cameras is that of determining interior orientation through calibration. Faig [40] has clearly set out the various methods available for camera calibration, and shows that for non-metric cameras—in which no assumptions regarding stability, construction or lens distortion can be made—complete calibration of every photograph using coordinated control points in the object space (so-called 'on-the-job' calibration) is necessary. As a corollary, we can see that a data processing technique is required in which the camera calibration constants are not required as input data, but direct relations between photograph coordinates and object space coordinates are formulated instead. This 'on-the-job' calibration moves the accuracy requirements from the photography to the data mensuration and processing stages of the process, with the result that these stages must be considered in this light. This is the subject of the next two sub-Sections.

3.2.2 Data processing

The requirement for 'on-the-job' calibration has led to the development of a number of suitable numerical techniques. These have been summarised by Gruen [53] as falling into the categories of: bundle solutions; direct linear transform; metric photo perspective transformation; rectification; and stepwise and combined procedures. Only the first two of these are candidates for the work in hand, as the final three methods are based on, respectively, the existence of metric photography, digitised photographs being manipulated by specialised computers, and an *ad hoc* mixture of any techniques that can provide photogrammetric control. Consequently, bundle solutions and the direct linear transform are discussed in greater detail.

Bundle solutions

Bundle solutions are based on the geometry of 'bundles' of light rays emanating from points in the object space, passing through the camera lens and onto the film plane. For a perfect camera and lens, the object point, image point and perspective centre must lie on a straight line, a condition represented by the **collinearity equations** (presented in sub-Section 3.3.1, below), and these represent the underlying theory of bundle solutions.

Bundle solutions are applicable to most non-metric photogrammetric problems, but require *a priori* knowledge of a number of geometric constraints in the object space. These constraints can be generated from a knowledge of either control point coordinates, the geometry of objects themselves (distances, angles, parallel or straight lines, plane conditions and so on), or elements of the exterior orientation of the camera, or a mixture of these.

The collinearity equations represent a non-linear system of equations, and are not solved directly. Instead, a first-order Taylor series expansion is taken, and the resulting linear system of equations solved iteratively, using *a priori* estimates of the unknown parameters. In order to account for lens distortion, so-called 'additional parameters' are introduced into the system of equations, and adjusted simultaneously during the solution with the other parameters in the collinearity equations.

The drawbacks associated with the use of bundle solutions are the need for both a great number of constraints and their corresponding initial estimates, and the time consuming nature of the iterative solution. To address these drawbacks, an alternative approach was developed: the direct linear transform.

Direct linear transform

The **direct linear transform** (DLT) was developed by Abdel-Aziz and Karara [1] as a method that requires neither a calibrated camera, nor one with fiducial marks, nor the knowledge of either geometric constraints or initial estimates of the unknown camera interior orientation parameters. Instead, a number of coordinated control points surrounding the object are used to solve a linearised system of collinearity equations directly.

The method incorporates a direct transform from comparator coordinates (not image coordinates, because the comparator—like the camera—may not be perfect) to object space coordinates, and therefore solves the complete analytical photogrammetric problem. In the DLT approach, a system of linear equations is assembled and solved to generate direct relations between comparator coordinates and object coordinates. These equations contain coefficients that account for linear components of film deformation, lens distortion and comparator errors as a minimum, and may be extended to account for symmetrical and asymmetrical lens distortion. However, the linear equations used in the DLT method are linearised approximations to the collinearity equations, and as such the results are of lower accuracy than those achieved with rigorous iterative bundle solutions.

Notwithstanding the essentially approximate nature of the solution, the simplicity of the DLT method has meant that it has found great application in fields where relatively low-accuracy non-metric photogrammetry is customarily used for the determination of three-dimensional object coordinates. Particular examples are the fields of biomedical engineering [57] and robotics [81]. The wealth of collected knowledge indicates that, providing the accuracy is acceptable, the DLT is a useful method. Consequently, the direct linear transformation has been selected for this work, and is therefore considered in depth in Section 3.3.

3.2.3 Measurement of photograph coordinates

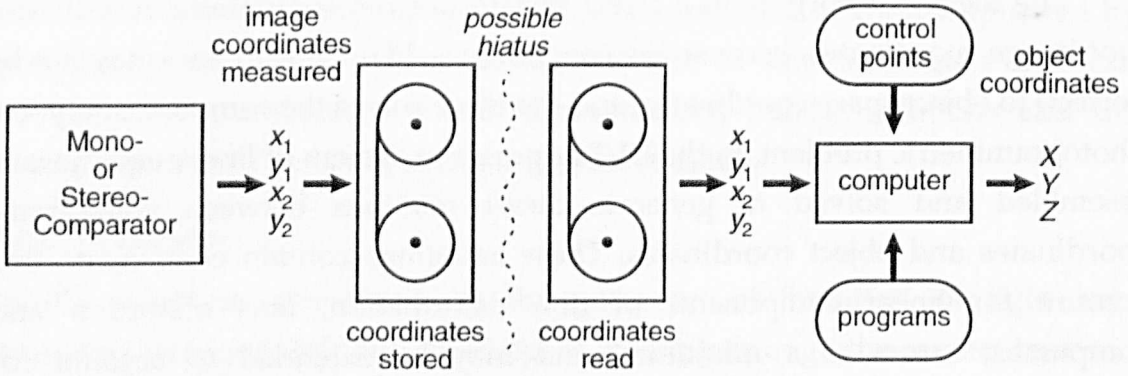
Petrie [104] has surveyed the instruments available for analytical photogrammetry, and has chosen to classify them into three main groups:

(i) a recording device for measuring photograph coordinates, with the photogrammetric solution being performed as a separate operation;

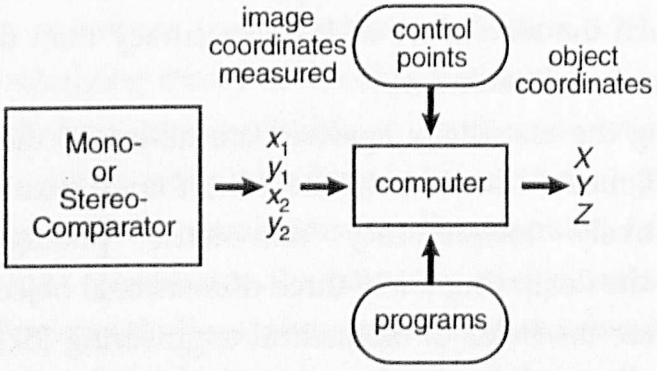
(ii) a device for measuring photograph coordinates which is connected directly to a computer for real-time processing of the photogrammetric solution; and

(iii) a coordinating device in which real-time feedback from the attached computer is used to continually adjust the relative positions of the photographs such that a correctly formed stereomodel is always visible.

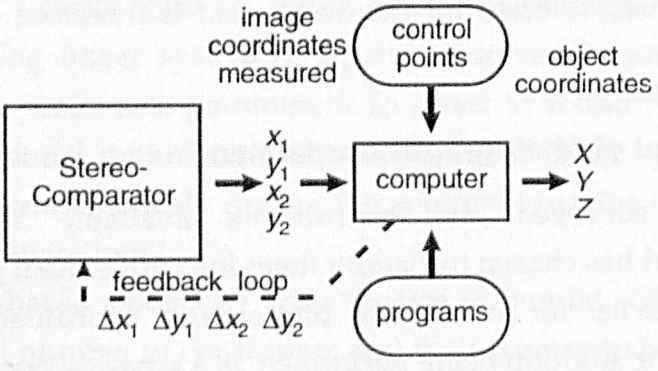
These three categories of instrument are shown schematically in Figure 3.08; for further reference it is convenient to refer to the instruments in these categories as **off-line comparators, real-time comparators** and **analytical plotters**, respectively.



(a) comparator with off-line solution



(b) comparator with real-time solution



(c) comparator with feedback

Figure 3.08 Configurations of analytical photogrammetric instruments (after Petrie [104]).

Instruments falling into all of these categories are available commercially, but they tend to be of cartographic standards (*i.e.* with a positioning resolution of about 1μm and a repeatability of about ±2μm), which means that they are very expensive and hence not readily available. The combination of these two factors appears to be the major drawback to using non-metric techniques, and as a consequence a number of workers have attempted to use other, more readily available although less accurate, equipment for particular applications.

This search for appropriate instrumentation means that reports of the use of novel equipment in the development of low cost off-line comparators, real-time comparators and analytical plotters can be found in the literature. It is useful to review some of these reports, and at the same time consider the practicalities of using such instruments, in the context of determining the suitability of these developments for an application in rock mass geometry.

Off-line comparators

Perhaps the most readily available instrument that could be used measuring photograph coordinates is a digitising tablet attached to a desktop computer, and this technique has been evaluated by both Oimoen [103] and Carson [24]. The work of these authors was motivated by a need in the fields of geography and forestry, respectively, for rapid determination of ground coordinates at moderate accuracy (*e.g.* errors of less than 0.5m). Both of these authors used customary metric aerial photography in their evaluations, with coordinates determined using first-order comparators as controls.

The digitising tablet used by Oimoen had a stated resolution of 0.025mm and an accuracy (usually quoted by manufacturers as 'repeatability') of 0.25mm. His methodology was to digitise the two photographs of a stereopair separately, recording the coordinates for later analysis by a rigorous bundle adjustment. He found root-mean-square (RMS) errors of around 0.1mm in photograph coordinates which, for the scale of the photography used in the test (about 1/4000), gave rise to RMS errors in horizontal ground coordinates of less than 0.2m, and an RMS error in ground elevation of 0.34m.

Although the work of Carson [24] was aimed at development of a real-time comparator, he performed a series of off-line tests in order to quantify the accuracy of a digitising tablet with quoted resolution of 0.025mm and repeatability of 0.14mm. Two very important findings resulted from this work: firstly, random errors in position measurement were about ±7μm; and, secondly, there was a large systematic error with a recurrency of 1 inch—a

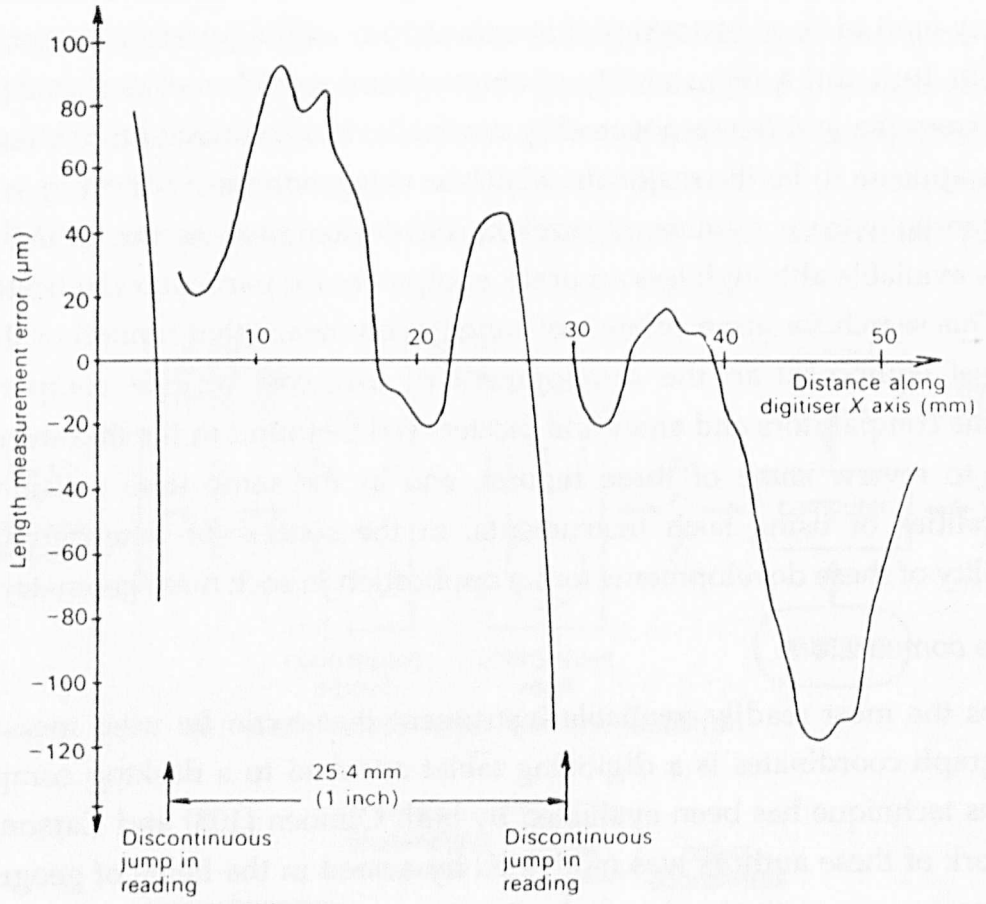


Figure 3.09 Systematic position error in a digitising tablet (from Carson [24]).

distance that corresponded to the spacing of the transducing conductors embedded in the tablet. The form that this systematic error took is indicated in Figure 3.09, and Carson assessed the magnitude of the discontinuity in error as being between 100μm and 140μm.

These two assessments of the use of digitising tablets produced the same conclusions: RMS errors in photograph coordinates of about 0.1mm should be expected, and this must be considered in the context of the particular application before deciding to use such equipment. An analysis of photograph coordinate errors in the context of determining discontinuity orientation is presented in this volume at Section 5.1, and the results of this indicate that a digitising tablet is of insufficient accuracy.

It is appropriate to ask whether it is feasible to use an off-line comparator for the analysis of rock mass geometry. Certainly, stereoscopic viewing will be essential, as it is only with binocular vision that homologous points on discontinuity surfaces appearing in two photographs will be easily identified.

However, the off-line nature of the analysis introduces other difficulties, though. Firstly, book-keeping that records data in terms of specific discontinuities will be required, and this implies that a unique system of identifying such features will be required. Secondly, because a rock mass can contain discontinuities with any orientation, detection of errors in computed discontinuity orientation will not be possible without reference to the original stereomodel. These two problems indicate that the use a real-time comparator is preferable.

Real-time comparators

Carson [24] was primarily interested in the development of a real-time stereocomparator, and he chose to use an 'instrumented parallax bar' in conjunction with a digitising tablet and mirror stereoscope. The resulting instrument he termed an 'image space plotter'.

As Carson was using metric aerial photography, it was possible for him to mount the two photographs on the digitising tablet with the correct relative orientation, so that y-parallax did not need to be accounted for. Thus, by connecting the cursor of the digitising tablet to the left-hand plate of the parallax bar, and electronically recording the x-parallax as the micrometer reading of the bar itself, he was able to determine object space coordinates of points on the photographs. The mechanical motion of the parallax bar micrometer was converted to an electrical signal by means of a precision multi-turn potentiometer, and an analogue-to-digital converter then produced a digital word (suitable for processing by the host minicomputer) from this signal.

In this system, the resolution of the parallax bar was 0.01mm, and that of the two digitising tablets tested was 0.1mm and 0.025mm. All of the electronic equipment was interfaced to a PDP MiniMINC minicomputer *via* serial (RS232) communication ports. The control program was configured such that, after the user pressed a button on the cursor to signal that tablet coordinates should be recorded, the computer did so, and then interrogated the control electronics of the parallax bar to determine the x-parallax, finally computing the corresponding object space coordinates. The time taken for one cycle of the complete measurement procedure was 0.15s.

The instrumented parallax bar worked well, but the system was let down by the large errors associated with the digitising tablet (as discussed in the previous Division) and the poor numerical approximations used in the

computation procedure. As a result, Carson concluded that although an instrument based on a commercially available minicomputer could be developed, position measurement with less than ±50μm error and more robust computational procedures were required.

In the context of analysing rock mass geometry using terrestrial non-metric photography, a real-time stereocomparator would work well. However, the *ad hoc* nature of the photography means that it is likely that large amounts of y-parallax will need to be removed from the stereomodel, and if operator comfort is to be maintained, an automatic method of doing so will be necessary. As a minimum, this requires feedback from the host microcomputer to drive a mechanism that moves the two photographs relative to each other in the y-direction, and this implies some form of analytical plotter.

Analytical plotters

The ready availability and low cost of modern desk-top computing equipment prompted Carson [25] to develop an inexpensive analytical plotter. This required identification of the minimum features that such an instrument should possess, and Carson offered the following criteria:

(i) the optical system must offer illuminated spot floating marks, variable magnification and be capable of viewing customary 230mm×230mm metric aerial photographs;

(ii) all mechanical and electronic components must provide for a repeatable resolution of at least ±10μm;

(iii) the control system must allow for positioning the photographs under the fixed optics, automatic control of y-parallax, adjustment of x-parallax and recording of photograph coordinates; and

(iv) the associated microcomputer and control software must be able to maintain a stereomodel free from y-parallax in real time.

An instrument based on this specification was subsequently produced, using a mirror stereoscope for the optics, precision linear bearings for the photograph guidance system, stepping motors for driving the x- and y-parallax adjustment system, linear electronic encoders for determination of photograph coordinates associated with the left photograph, and rotary electronic encoders for determining the x- and y-parallax. The interface circuitry for the electronic systems was based on a Z80 microprocessor, and connected to an IBM-compatible Olivetti M24 microcomputer using an RS232 serial communication port.

The choice of a mirror stereoscope for the optical system is interesting. The general low-level of sophistication and flexibility of these instruments has led to their wide-spread use—one will be found in most engineering offices where even only occasional study of aerial photographs is undertaken—and this in turn has tended to keep their cost low. Providing the accuracy achievable through the use of such an instrument is acceptable, then these factors mean that consideration must be given to adopting a mirror stereoscope in this work.

A brief report of the accuracy attainable with his analytical plotter has been published by Carson [23]. An accuracy test was implemented by inviting visitors to a photogrammetric exhibition to measure the photograph coordinates of 12 points on an aerial stereopair, and then comparing their measurements with those taken by an experienced photogrammetrist using a first-order cartographic instrument. Of the forty visitors who accepted the invitation, 28 achieved standard deviations less than 20μm, and the maximum standard deviation recorded was 27μm. Experienced users were reported as being able to achieve standard deviations under 12μm. On this basis—and considering the results presented in later, Section 5.1 of this volume,—there is little doubt that the combination of an optical system provided by a mirror stereoscope, and an electro-mechanical system of linear bearings and electronic encoders, can lead to an instrument of sufficient accuracy for the work in hand here.

Although no reports of the general efficacy of Carson's instrument appear to have been published, an improved version of the instrument has been produced and marketed by Cartographic Engineering Ltd. of Salisbury, England. Presumably, the commercial availability of the instrument can be taken as a demonstration of its adequacy for certain applications. The current Author has been fortunate to briefly test one of these instruments, and whilst it has not been possible to produce a detailed assessment of the instrument, a number of design and constructional details of the instrument mean that this particular form is unsuitable for the work proposed here. In particular, the instrument displayed insufficient smoothness of operation, and the analysis and control programs were suitable for metric photography only. Of course, these shortcomings could easily be addressed by suitable enhancements to the electronic and software systems. More serious, though, are the lack of provision for complete computer controlled motion and closed-loop measurement of coordinates and parallax.

According to Petrie [104], complete computer controlled motion allows the user of an analytical plotter greater flexibility of application, in that it is possible to choose any position in object space and drive the floating spot to the corresponding location within the stereomodel. In particular, this allows a digital terrain model to be constructed and taking an example in the context of analysing rock mass geometry, such a facility would allow great opportunity for the validation of geometrical models. However, if complete computer controlled motion is to be implemented, then closed-loop measurement of coordinates and parallax is essential, as stepping motors can lose synchronisation, with the result that 'command position' may not equal 'actual position'. Under these conditions erroneous data may be passed to an analytical program, and incorrect object space coordinates computed as a result.

3.2.4 Summary

A non-metric photogrammetric system can be considered in terms of three components: non-metric cameras; data processing techniques; and instruments for the measurement of photograph coordinates.

In this Section, we have seen that the adoption of non-metric cameras means that a data processing technique that implements 'on-the-job' calibration must be used. A suitable technique for this is the direct linear transform, and this is considered in greater detail in the next Section.

Of the three categories of instrument for measuring photograph coordinates, only an analytical plotter appears appropriate: the impossibility of error detection in off-line comparators, and the lack of a means of removing y-parallax in on-line comparators, leads to this conclusion. As a means of measuring photograph coordinates, the lowest-cost option of using a digitising tablet is inappropriate as published results show that such devices cannot produce errors of sufficiently low magnitude.

Finally, we have seen that an adequate analytical plotter can be developed using a mirror stereoscope for the optical system, and linear bearings and electronic encoders for the positioning system. However, in order to maximise the potential of such an instrument, complete closed-loop, computer-controlled positioning should be incorporated.

The development of a low-cost analytical plotter incorporating complete closed-loop, computer-controlled positioning is described in detail in Chapter 5. This Chapter continues with a discussion of the Direct Linear Transform for solution of the non-metric photogrammetric problem.

3.3 Direct Linear Transform

The reviews presented in Section 3.2 above have indicated the use of an analytical plotter for measurement of photograph coordinates and the direct linear transform (DLT) method for subsequent data processing as being most appropriate for non-metric photogrammetry applied to the analysis of rock mass geometry. The computer-controlled nature of an analytical plotter means that the method used for the solution of the photogrammetric problem is intimately bound into the operation of such an instrument. Consequently, as a precursor to the work presented in Chapter 5 that describes the design and construction of an analytical plotter, a detailed description of the DLT method is given here. This Section consists of five sub-Sections. The first two sub-Sections are concerned, respectively, with the formulation of the collinearity equations and, from these, the formulation of the DLT equations. The next two sub-Sections cover the least squares solution of the DLT equations and the computation of object space coordinates. The final sub-Section draws the essential details together in a summary.

3.3.1 Formulation of the collinearity equations

The DLT, as first proposed by Abdel-Aziz and Karara [1] and subsequently enhanced by Marzan and Karara [94], is simply an analytical formulation of the customary photogrammetric problem, but one that does not depend on either a metric or a calibrated camera. In recognition of the fact that every component of the photogrammetric system may be imperfect, the DLT seeks to account for all errors by forming a direct relation between the coordinates of control points in object space and the corresponding comparator coordinates (*i.e.* as-measured photograph coordinates). This relation follows from a linearisation of the collinearity equations.

The collinearity equations represent a three-dimensional coordinate transform from object space coordinates to photograph coordinates, the origin of the latter system being the camera perspective centre. The fundamental theoretical concept of collinearity assumes that the photograph is a perfect plane and that the camera is free from distortion, with the result that an image point, the projection centre of the camera lens and the object point are collinear [7]. In essence, the collinearity equations relate one coordinate system (the object space system) with another (the photograph system), and this is revealed when they are written in matrix form as

$$\begin{bmatrix} x - x_p \\ y - y_p \\ -c \end{bmatrix} = \lambda \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X - X_c \\ Y - Y_c \\ Z - Z_c \end{bmatrix} \tag{3.06}$$

where

x, y = photograph coordinates of a point,
x_p, y_p = photograph coordinates of the camera principal point
c = camera principal distance
λ = scale factor
m_{ij} = elements of the rotation matrix
X, Y, Z = object space coordinates of the point
X_c, Y_c, Z_c = object space coordinates of camera perspective centre

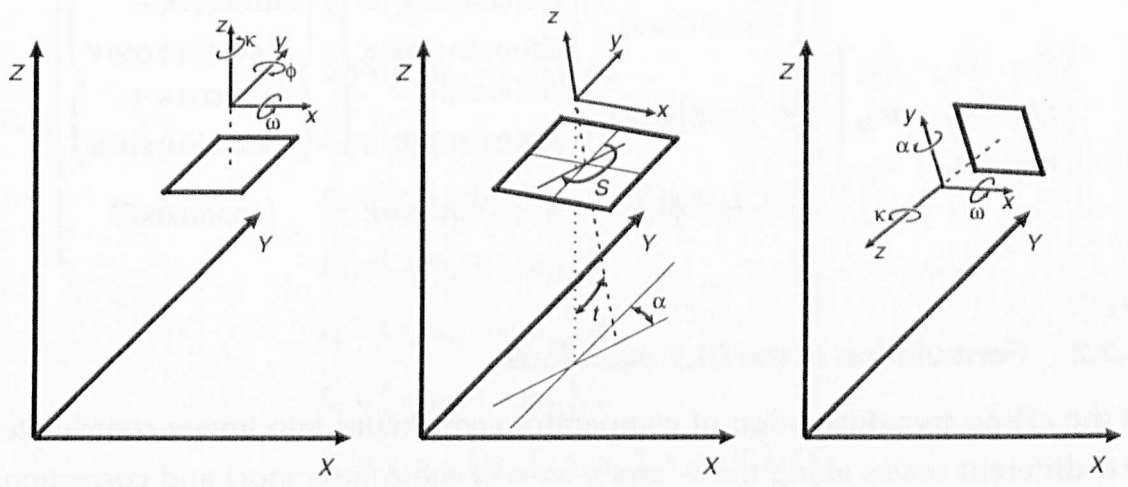
The m_{ij} elements of the rotation matrix enable the camera orientation to be quantified relative to the global Cartesian coordinate system. Although three rotations (one about each of the Cartesian axes) are sufficient to uniquely define this orientation, three major systems for defining these rotations are in common photogrammetric usage [7]. These are:

(i) the **omega-phi-kappa** (ω–ϕ–κ) system, commonly used for aerial photogrammetry and shown in Figure 3.10(a), in which the rotation angles are all counter-clockwise positive about the local *x*-, *y*- and *z*-axes;

(ii) the **tilt-swing-azimuth** (t-S-α) system, illustrated in Figure 3.10(b) and also used for aerial photogrammetry, in which tilt is the always-positive angle between the camera principal axis and the global *Z*-axis, swing is the clockwise-positive angle—measured in the plane of the photograph—between the local *y*-axis and the line of maximum tilt and azimuth is the clockwise-positive angle—measured in the global *X*-*Y* plane—between the line of maximum tilt and the global *Y*-axis; and

(iii) the **omega-alpha-kappa** (ω-α-κ) system, shown in Figure 3.10(c) and commonly used for terrestrial photogrammetry, in which all of the angles are measured about the local *x*-, *y*- and *z*-axes, but with ω and κ counter-clockwise positive and α clockwise positive.

Customarily, the choice of system is made on the basis of which will give rise to the smallest rotations. This stems from the use of analogue photogrammetry, in which the mechanical constraints applied by analogue plotters made the use of a coordinate system that produced small rotations a necessity. No such constraints are present when using analytical photogrammetry, and a



(a) omega-phi-kappa (ω–ϕ–κ) system

(b) tilt-swing-azimuth (t-S-α) system

(c) omega-alpha-kappa (ω-α-κ) system

Figure 3.10 Photogrammetric systems for defining camera orientation.

numerically convenient system is more important than a system that results in small rotations.

Thus, although photogrammetry in rock mechanics is a terrestrial application and therefore indicates that the ω-α-κ system should be adopted, the non-standard sense (in terms of the mathematics of right-handed Cartesian axes) of the α-rotation—clockwise-positive—suggests otherwise. Indeed, there is no disadvantage incurred if the ω–ϕ–κ system (with its standard counter-clockwise positive rotations) is adopted for analytical terrestrial photogrammetry: the magnitude of ω (the *x*-axis rotation) will be computed as being in the region of -90°, but that represents no difficulty in an analytical solution.

Hence, adopting the ω–ϕ–κ system, the elements of the rotation matrix may be computed by the following sequence of matrix multiplications [114]:

$$\mathbf{M} = \mathbf{M}_z \cdot \mathbf{M}_y \cdot \mathbf{M}_x \tag{3.07}$$

where

$$\mathbf{M}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix}, \quad \mathbf{M}_y = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix}, \quad \mathbf{M}_z = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The individual elements of the rotation matrix are, then,

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} (\cos\phi\cos\kappa) & \begin{pmatrix}\cos\omega\sin\kappa + \\ \sin\omega\sin\phi\cos\kappa\end{pmatrix} & \begin{pmatrix}\sin\omega\sin\kappa - \\ \cos\omega\sin\phi\cos\kappa\end{pmatrix} \\ (-\cos\phi\sin\kappa) & \begin{pmatrix}\cos\omega\cos\kappa - \\ \sin\omega\sin\phi\sin\kappa\end{pmatrix} & \begin{pmatrix}\sin\omega\cos\kappa + \\ \cos\omega\sin\phi\sin\kappa\end{pmatrix} \\ (\sin\phi) & (-\sin\omega\cos\phi) & (\cos\omega\cos\phi) \end{bmatrix} \quad (3.08)$$

3.3.2 Formulation of the DLT equations

If the **affine transformation** of comparator coordinates into image coordinates (*i.e.* different scales along the x- and y-axes of the comparator) and corrections for lens distortion are incorporated into the collinearity equations, they can be written in the form

$$\left.\begin{aligned} x + \delta_x &= x_p - c_x \frac{m_{11}(X - X_c) + m_{12}(Y - Y_c) + m_{13}(Z - Z_c)}{m_{31}(X - X_c) + m_{32}(Y - Y_c) + m_{33}(Z - Z_c)} \\ y + \delta_y &= y_p - c_y \frac{m_{21}(X - X_c) + m_{22}(Y - Y_c) + m_{23}(Z - Z_c)}{m_{31}(X - X_c) + m_{32}(Y - Y_c) + m_{33}(Z - Z_c)} \end{aligned}\right\} \quad (3.09)$$

where x and y are the image coordinates in the comparator coordinate system, δ_x and δ_y are the systematic errors of the image coordinates induced by lens distortion, and c_x and c_y represent the camera principal distances relative to the x and y coordinate directions. It is as a result of the differential scaling inherent in the affine transformation that c_x and c_y are not equal.

Equations (3.09) may be written in the basic form of the DLT equations as

$$\left.\begin{aligned} x + \delta_x &= \frac{L_1 X + L_2 Y + L_3 Z + L_4}{L_9 X + L_{10} Y + L_{11} Z + 1} \\ y + \delta_y &= \frac{L_5 X + L_6 Y + L_7 Z + L_8}{L_9 X + L_{10} Y + L_{11} Z + 1} \end{aligned}\right\} \quad (3.10)$$

where L_1 through to L_{11} are the DLT coefficients, given by

$$\left.\begin{aligned}
L_1 &= \left(x_p m_{31} - c_x m_{11}\right)/L \\
L_2 &= \left(x_p m_{32} - c_x m_{12}\right)/L \\
L_3 &= \left(x_p m_{33} - c_x m_{13}\right)/L \\
L_4 &= x_p + c_x\left(m_{11}X_c + m_{12}Y_c + m_{13}Z_c\right)/L \\
L_5 &= \left(y_p m_{31} - c_y m_{21}\right)/L \\
L_6 &= \left(y_p m_{32} - c_y m_{22}\right)/L \\
L_7 &= \left(y_p m_{33} - c_y m_{23}\right)/L \\
L_8 &= y_p + c_y\left(m_{21}X_c + m_{22}Y_c + m_{23}Z_c\right)/L \\
L_9 &= m_{31}/L \\
L_{10} &= m_{32}/L \\
L_{11} &= m_{33}/L \\
L &= -\left(m_{31}X_c + m_{32}Y_c + m_{33}Z_c\right)
\end{aligned}\right\} \qquad (3.11)$$

The terms δ_x and δ_y may be expanded at this stage, by including explicit expressions for lens distortion derived from (3.01) and (3.02), so that

$$\begin{aligned}
\delta_x &= x'\left(K_1 r^2 + K_2 r^4 + K_3 r^6\right) + \left[r^2 + 2(x')^2\right]P_1 + 2y'x'P_2 \\
\delta_y &= y'\left(K_1 r^2 + K_2 r^4 + K_3 r^6\right) + 2x'y'P_1 + \left[r^2 + 2(y')^2\right]P_2
\end{aligned} \qquad (3.12)$$

where $x' = x - x_p$, $y' = y - y_p$, $r^2 = x'^2 + y'^2$ and only the distortion terms K_1, K_2, K_3, P_1 and P_2 are considered of importance.

Further algebraic manipulation allows the elements of interior and exterior camera orientation to be calculated from the DLT parameters [69], through the use of (3.08) and (3.11), as

$$
\left.
\begin{aligned}
L &= -1\Big/\sqrt{\left(L_9^{\,2} + L_{10}^{\,2} + L_{11}^{\,2}\right)} \\
x_p &= \left(L_1 L_9 + L_2 L_{10} + L_3 L_{11}\right) L^2 \\
y_p &= \left(L_5 L_9 + L_6 L_{10} + L_7 L_{11}\right) L^2 \\
c_x &= \sqrt{\left[\left(L_1^{\,2} + L_2^{\,2} + L_3^{\,2}\right) L^2 - x_p^{\,2}\right]} \\
c_y &= \sqrt{\left[\left(L_5^{\,2} + L_6^{\,2} + L_7^{\,2}\right) L^2 - y_p^{\,2}\right]} \\
m_{11} &= L\left(x_p L_9 - L_1\right)\Big/ c_x \\
\omega &= \tan^{-1}\left(-L_{10}/L_{11}\right) \\
\phi &= \sin^{-1}\left(L_9 L\right) \\
\kappa &= \left[\cos^{-1}\left(m_{11}\right)\right]\Big/\cos\phi \\
\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} &= \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix}^{-1} \begin{bmatrix} L_4 \\ L_8 \\ 1 \end{bmatrix}
\end{aligned}
\right\} \qquad (3.13)
$$

3.3.3 Least squares solution of the DLT equations

Equations (3.10) and (3.12) may be simplified through three assumptions [94], namely: the terms δ_x and δ_y are, by themselves, insignificant; of the lens distortion parameters only K_1 needs to be taken into account; and the magnitudes of x_p and y_p are sufficiently small to be ignored. With these assumptions, a direct solution to the DLT equations can be derived, with equations (3.10) then being written as

$$
\left.
\begin{aligned}
L_1 X_i + L_2 Y_i + L_3 Z_i + L_4 - x_i X_i L_9 - x_i Y_i L_{10} - x_i Z_i L_{11} - x_i K_1 r_i^2 A - x_i &= r_x \\
L_5 X_i + L_6 Y_i + L_7 Z_i + L_8 - y_i X_i L_9 - y_i Y_i L_{10} - y_i Z_i L_{11} - y_i K_1 r_i^2 A - y_i &= r_y
\end{aligned}
\right\} \qquad (3.14)
$$

where r_x and r_y represent residuals arising from the simplification due to the inclusion of the assumptions. Equations (3.14) show how the photograph coordinates (x_i, y_i) of the ith object are related to the corresponding object space coordinates (X_i, Y_i, Z_i) through the DLT coefficients. If random errors are incorporated into (3.10), non-linear equations result, and these can only be solved iteratively, following linearisation by a Taylor expansion. A method similar to a customary bundle solution therefore results.

Continuing with the approximations given by (3.14), a system of **condition equations** can be written for a number of object points with known object space coordinates, as

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1r^2 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1r^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -x_iX_i & -x_iY_i & -x_iZ_i & -x_ir^2 \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -y_iX_i & -y_iY_i & -y_iZ_i & -y_ir^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_nX_n & -x_nY_n & -x_nZ_n & -x_nr^2 \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_nX_n & -y_nY_n & -y_nZ_n & -y_nr^2 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_{10} \\ L_{11} \\ K_1' \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_i \\ y_i \\ \vdots \\ x_n \\ y_n \end{bmatrix} + \begin{bmatrix} r_{x_1} \\ r_{y_1} \\ \vdots \\ r_{x_i} \\ r_{y_i} \\ \vdots \\ r_{x_n} \\ r_{y_n} \end{bmatrix} \quad (3.15)$$

where $K_1' = (XL_9 + YL_{10} + ZL_{11} + 1)K_1$. This system has the general form

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} = \mathbf{v} \quad (3.16)$$

where

A = real $m \times n$ matrix ($m \geq n$) of constants, with rank n

x = column vector of n unknowns

b = column vector of m independent variables

v = column vector of m residuals

The constraint $m \geq n$ is necessary in order to produce a system of equations that does not have infinitely many solutions. Although a unique solution exists when $m=n$, when a larger number of control points are used then equations (3.15) will become overdetermined, necessitating a least squares solution.

Equations (3.15) clearly demonstrate the need for at least 6 control points when the basic DLT solution of eleven parameters plus the first component of lens distortion is applied. Similarly, the number of control points rises to eight when the additional four parameters associated with symmetrical and asymmetrical lens distortion are taken into account [8].

The **weighted least squares solution** to equations (3.15) is that which minimises the sum of the squares of the residuals, Q, which may be calculated from the expression

$$Q = \mathbf{v}^{\mathrm{T}} \cdot \mathbf{W} \cdot \mathbf{v} \quad (3.17)$$

where **W** is a diagonal matrix, $\mathbf{W} = \mathrm{diag}\{w_{1x}, w_{1y}, \cdots, w_{ix}, w_{iy}, \cdots, w_{nx}, w_{ny}\}$. The elements of this matrix are given by $w_{ij} = 1/\sigma_{ij}^2$, where σ_{ij} is the standard deviation associated with the x (j=1) or y (j=2) condition equation associated with the measurement of the ith independent variable. The inclusion of the weight matrix **W** ensures that those individual equations in which we have

little confidence (*i.e.* the data used to form them possess a large standard deviation) have correspondingly little effect on the solution.

Minimising Q is accomplished by taking the derivative with respect to x of equations (3.17). We begin by substituting for $\mathbf{v}$ in (3.17) to give

$$Q = \mathbf{v}^T \cdot \mathbf{W} \cdot \mathbf{v} = (\mathbf{Ax} - \mathbf{b})^T \mathbf{W}(\mathbf{Ax} - \mathbf{b}) \tag{3.18}$$

This expression is then expanded, making use of the multiplicative properties of transposed matrices, to give

$$Q = \mathbf{x}^T\mathbf{A}^T\mathbf{WAx} - \mathbf{AxWb} - \mathbf{b}^T\mathbf{WAx} + \mathbf{b}^T\mathbf{Wb} \tag{3.19}$$

Taking the partial derivative with respect to x results in

$$\frac{\partial Q}{\partial x} = 2\mathbf{A}^T\mathbf{WAx} - \mathbf{A}^T\mathbf{Wb} - \left(\mathbf{b}^T\mathbf{WA}\right)^T \tag{3.20}$$

which, making use of matrix transpositions, may be written as

$$\frac{\partial Q}{\partial x} = 2\mathbf{A}^T\mathbf{WAx} - \mathbf{A}^T\mathbf{Wb} - \mathbf{A}^T\mathbf{Wb} \tag{3.21}$$

Collecting terms, and noting that this expression must be equal to zero for a minimisation of Q, we find that

$$\left(\mathbf{A}^T\mathbf{WA}\right)\mathbf{x} = \mathbf{A}^T\mathbf{Wb} \tag{3.22}$$

with the result

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{WA}\right)^{-1}\mathbf{A}^T\mathbf{Wb} \tag{3.23}$$

Equations (3.23) are known as the **normal equations** of the system, and may be solved on a computer using standard subroutine library procedures. By this method $\mathbf{x}$, the vector of DLT parameters, is determined for a given photograph.

Marzan and Karara [94] developed the following expressions for standard deviations used to compute the elements w_{ij} of the weight matrix $\mathbf{W}$:

$$\left.\begin{aligned}
\sigma_x^2 &= \left(\frac{xL_9 - L_1}{A}\right)^2 \sigma_X^2 + \left(\frac{xL_{10} - L_2}{A}\right)^2 \sigma_Y^2 + \left(\frac{xL_{11} - L_3}{A}\right)^2 \sigma_Z^2 + (\sigma_x)^2_{\text{comp}} \\
\sigma_y^2 &= \left(\frac{yL_9 - L_5}{A}\right)^2 \sigma_X^2 + \left(\frac{yL_{10} - L_6}{A}\right)^2 \sigma_Y^2 + \left(\frac{yL_{11} - L_7}{A}\right)^2 \sigma_Z^2 + (\sigma_y)^2_{\text{comp}}
\end{aligned}\right\} \tag{3.24}$$

where

σ_x, σ_y = standard deviations associated with the x and y condition equations, respectively;

$\sigma_X, \sigma_Y, \sigma_Z$ = standard deviations of the object space coordinate measurements;

$(\sigma_x)_{comp}, (\sigma_y)_{comp}$ = standard deviations of the photograph coordinate measurements; and

$A = L_9X + L_{10}Y + L_{11}Z + 1$

These standard deviations must be computed iteratively, as they are dependent on the DLT parameters. Marzan and Karara suggest that two iterative cycles are sufficient, the initial cycle being performed with values of unity for all elements of the weight matrix.

As a means of quantifying the statistical errors in the analysis, the so-called **variance-covariance matrix** of the solution is computed [98]:

$$\sigma_x^2 = \sigma_0^2 \left[\mathbf{A}^T \mathbf{W} \mathbf{A}\right]^{-1} \tag{3.25}$$

This matrix contains variances on the leading diagonal and covariances in all off-diagonal locations, and is used to determine how well the individual condition equations in (3.15) fit the final solution. The scaling factor σ_0^2, the **variance of unit weight**, is determined from

$$\sigma_0^2 = \frac{\mathbf{v}^T \mathbf{W} \mathbf{v}}{m - n} \tag{3.26}$$

The title 'variance of unit weight' reflects the fact that any equation in equations (3.25) will have this variance if its weight, computed from equation (3.26), is unity.

3.3.4 Computation of object space coordinates

Once the DLT parameters have been computed for both photographs, then the object space coordinates of any point which is imaged in both photographs can be computed. For such a point, four condition equations—each derived from (3.14)—can be written, as follows:

$$\begin{bmatrix} (L_1^l - x^l L_9^l) & (L_2^l - x^l L_{10}^l) & (L_3^l - x^l L_{11}^l) \\ (L_5^l - y^l L_9^l) & (L_6^l - y^l L_{10}^l) & (L_7^l - y^l L_{11}^l) \\ (L_1^r - x^r L_9^r) & (L_2^r - x^r L_{10}^r) & (L_3^r - x^r L_{11}^r) \\ (L_5^r - y^r L_9^r) & (L_6^r - y^r L_{10}^r) & (L_7^r - y^r L_{11}^r) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -(L_4^l - x'^l) \\ -(L_8^l - y'^l) \\ -(L_4^r - x'^r) \\ -(L_8^r - y'^r) \end{bmatrix} \tag{3.27}$$

where $x'^{l} = x^{l}\left[1+K_1\left(r^{l}\right)^2 A\right]$, $y'^{l} = y'^{l}\left[1+K_1\left(r^{l}\right)^2 A\right]$, $x'^{r} = x^{r}\left[1+K_1\left(r^{r}\right)^2 A\right]$ and $y'^{r} = y^{r}\left[1+K_1\left(r^{r}\right)^2 A\right]$. In these equations, the superscripts *l* and *r* refer to the DLT parameters and the photograph coordinates associated with the left and right photographs, respectively. A further least squares analysis of this system is then performed to solve for the object space coordinates *X,Y,Z*, and this is repeated for as many object points as desired.

In a manner similar to that used for quantifying the overall error in the solution, the errors associated with determination of individual point coordinates are computed from the variance-covariance matrix arising from equations (3.27). A **mean positional error** may be defined in terms of the root mean square error [98] as

$$\overline{\sigma}_p = \frac{1}{N}\sum_{i=1}^{N}\left(\left(\sigma_x\right)_i^2 + \left(\sigma_y\right)_i^2 + \left(\sigma_z\right)_i^2\right)^{1/2} \tag{3.28}$$

where $\left(\sigma_x\right)_i^2$, $\left(\sigma_y\right)_i^2$ and $\left(\sigma_z\right)_i^2$ are the variances taken from the variance-covariance matrix for point *i*.

3.3.5 Summary

In this Section, an illustration has been given of how the DLT formulation is derived from the basic collinearity equations—equations (3.06)—and how, following rearrangement and simplification, simple condition equations—equations (3.14)—result.

The steps involved in using the DLT method are straightforward:

(i) for each point in object space for which both the object space coordinates and the photograph coordinates are known, two condition equations are written;

(ii) a sufficient number of condition equations are assembled to form an overdetermined system of linear equations;

(iii) the normal equations of the system are formed and solved to determine the DLT coefficients; and

(iv) using the DLT coefficients corresponding to two photographs, and the photograph coordinates of objects visible thereon, a further system of condition equations is assembled and solved, through the normal equations, to compute the object space coordinates of the object itself.

This last step is repeated for as many object points as desired.

Although the DLT method is simple, it is not without its deficiencies. Some of these result from the simplifying assumptions used in the development of the DLT equations, but others result from the form of the condition equations themselves. This point is covered in Section 5.4, where enhancements to the numerical solution process are discussed.

Having completed an overview of photogrammetry in general, and non-metric photogrammetry in particular, a review of how photogrammetry has been applied in the past to the solution of problems in rock mechanics can be usefully and clearly undertaken. This is the subject of the next Section.

3.4 Photogrammetry in Rock Mechanics

The appeal of using photogrammetry as a remote and rapid survey technique in rock mechanics is not original, and many workers have reported on such applications. Studying this literature in the context of photogrammetric principles—as outlined in Sections 3.1 and 3.2 above—enables a critical review to be made, thus revealing the shortcomings in this previous work. An analysis and understanding of these shortcomings is essential if advances are to be made.

The wide ranging nature of the previous use of photogrammetry in rock mechanics means that a review can be organised in many ways. In keeping with the general theme of this work, the primary division chosen here is between general applications and those directed principally at the measurement of rock mass geometry. These two subjects are presented in the two sub-Sections below.

3.4.1 Photogrammetry applied to general rock mechanics problems

Photogrammetry of various configurations has been used in rock mechanics to study a range of phenomena apart from rock mass geometry. The examples discussed below cover the range from metric analogue to non-metric analytical techniques (at various degrees of photogrammetric sophistication), both monoscopic and stereoscopic applications, and include measurements of blast fragmentation, particle size distribution, discontinuity surface roughness and slope stability.

Single image photogrammetry has seemed to many workers to promise a rapid, quantitative technique, and as such its use has been widely reported. The

common factor in these reports is the use of photogrammetry to simply measure the two-dimensional projected size of objects, using a device of some sort to measure photograph (or, more generally, image) coordinates.

Noren and Porter [102] used a photogrammetric technique to study the fragmentation performance of explosives. Photographic images of the muckpile resulting from a blast were projected onto a screen carrying a grid, and the projected area of rock fragments thus determined. From these measurements of projected area, an equivalent particle size distribution was obtained. Although these workers were well aware of the assumptions necessary to compute a particle size distribution from such measurements, it seems they were unaware of any photogrammetric principles. However, because the scope of the work was only to compare the results of different blast conditions in the same rock mass, with no attempt being made to calibrate the technique with particle size distributions determined by mechanically screening the muckpiles, any errors present would have been constant throughout the work. The general usefulness of the technique was, however, unproven.

Similar work has been conducted by Aimone and Dowding [4], Carter [26], Franklin and Maerz [45], Gozon *et alia* [51] and Maerz *et alia* [85]. In all of this work, the deficiencies of a single image—in terms of it being unable to convey three-dimensional coordinates—have been addressed through the use of assumptions such as the existence of spherical particles and the representative nature of the surface of a pile. Indeed, only the work of Franklin and Maerz [45] and Maerz *et alia* [85] has attempted to overcome the limitation of particle shape and partially hidden fragments through the application of stereology [122]. Notwithstanding the application of such a sophisticated mathematical technique, this must still be regarded as an approximation given the intractability of mathematically describing rock fragment shape.

Given the assumptions inherent in all of the techniques discussed here, and noting that true three-dimensional object coordinates cannot be obtained from a single photograph, then applications of single image or monoscopic photogrammetry to studies in rock mechanics must be limited. For three-dimensional mensuration without assumptions regarding shape, stereoscopic photogrammetry must be used.

Ross-Brown [115] presented a complete discourse of the use of stereoscopic metric analogue terrestrial photogrammetry in rock mechanics. This work was essentially concerned with slope stability, and so the surveying tended to concentrate on determining the location and orientation of large,

planar discontinuities and faults relative to the side slopes of open pit mines. The photography was obtained through the use of a metric phototheodolite, with the images captured on glass plates. Measurement of photograph coordinates was undertaken using an analogue plotter (of cartographic standard), fitted with micrometers. The number of publications resulting from this work [116, 117, 118, 119] probably reflect its originality, and one of these [117] is still in use as an ISRM Suggested Method [22]. However, there can be little doubt that this Suggested Method is an anachronism, concentrating as it does solely on metric analogue techniques.

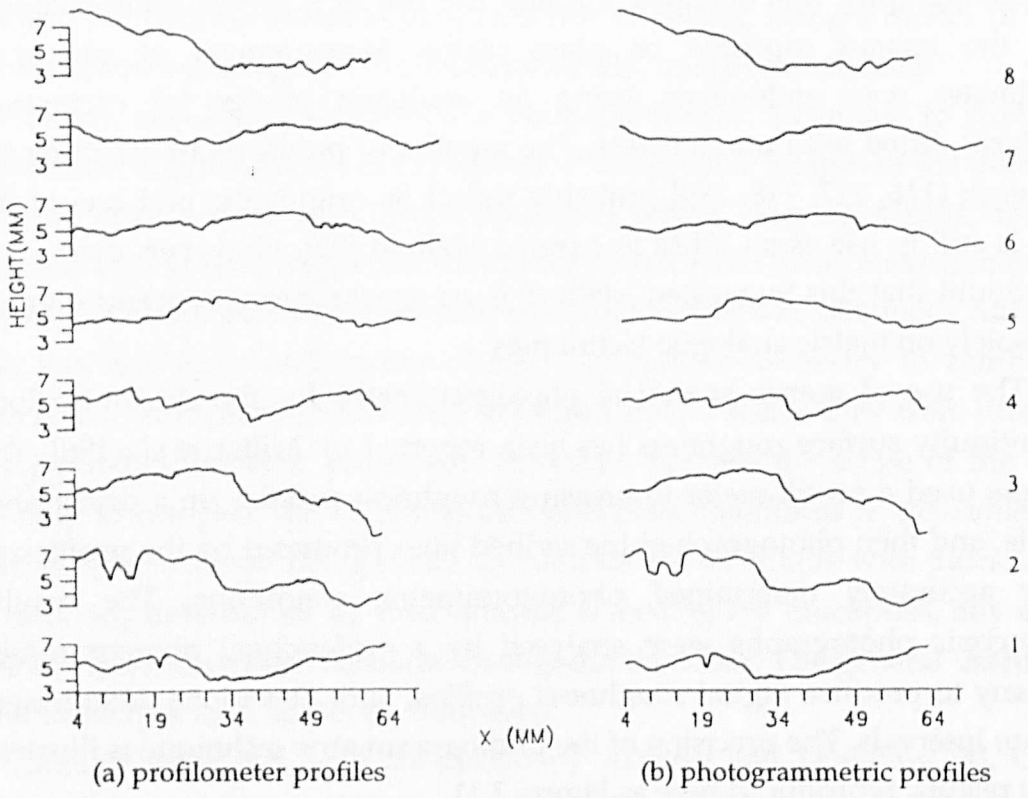
The use of metric analytical photogrammetry for the determination of discontinuity surface roughness has been reported by Miller *et alia* [96]. These workers used a profilometer to measure roughness profiles on a discontinuity sample, and then photographed the scribed lines produced by the profilometer under accurately determined photogrammetric conditions. The resulting stereoscopic photographs were analysed by a professional photogrammetric company to produce digital roughness profiles, with elevations determined at 0.05mm intervals. The precision of the photogrammetric technique is illustrated by the results, reproduced here as Figure 3.11.

Although the work in these last two examples was primarily directed at the analysis of rock slope stability and discontinuity surface roughness, respectively, in both cases it required the determination of some geometric properties of discontinuities. A great deal of work has been performed in which this was the sole purpose, and this work is reviewed in the next sub-Section.

3.4.2 Photogrammetry applied to rock mass geometry

As with general applications of photogrammetry in rock mechanics, examples of applications concerned primarily with the determination of spatial distribution and orientation of discontinuities exist which cover the whole range of possible photogrammetric configurations.

Extensive use of metric analytical photogrammetry for the determination of discontinuity geometry has been reported by Allam [5]. This work was motivated by the need to find a reliable, rapid and safe technique for determining rock mass geometry as an input for stability analysis. A phototheodolite was used to photograph a quarry face, and then a stereocomparator in conjunction with a mainframe computer was used to determine discontinuity orientation, "width" and "length", together with roughness profiles. Whether "width" and "length" refer to in-plane extent or



(a) profilometer profiles (b) photogrammetric profiles

Figure 3.11 Roughness profiles for a rock discontinuity surface (from Miller *et alia* [96]).

aperture and trace length is not clear from the publication. Three particular aspects of this work warrant specific attention, and these are in the following three points.

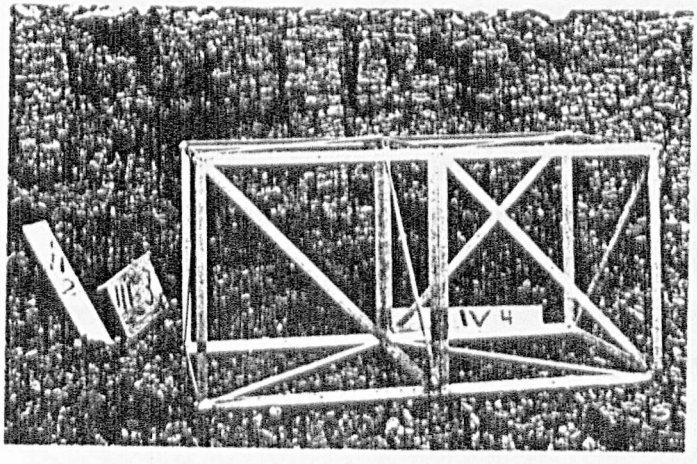
(i) Object space control was achieved by means of targets painted on the rock face. Whilst they gave excellent control for the photogrammetric analysis, their emplacement defeated the safety aspect of the technique.

(ii) A number of points were coordinated on each discontinuity surface, and the least-squares plane fitted through them taken to represent the orientation of the discontinuity surface itself. Although the author noted that, in the case of a non-planar discontinuity surface, several such planes could be determined, unfortunately no guidance was given as to how a non-planar surface could be identified, or how it should subsequently be analysed.

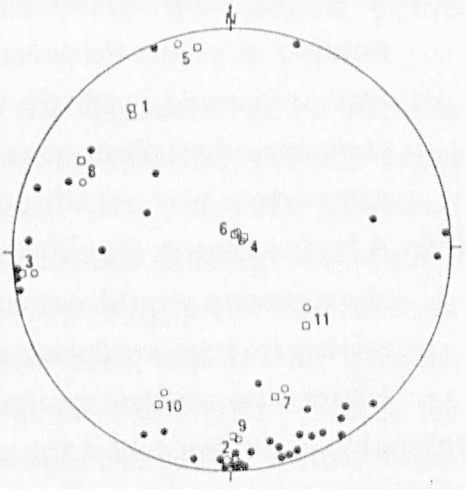
(iii) The metric equipment used in the work—phototheodolite and cartographic standard stereocomparator—means that the general technique is inappropriate for widespread use.

Some interesting points regarding error emerged from this work. The reported error in determining discontinuity orientation was ±2°, resulting from root-mean-square errors of less than 12μm in photograph coordinates. Unfortunately, no information was given to enable the scale of the photographs relative to the object to be determined. Notwithstanding this last omission, the author reported that such photograph coordinate errors were acceptable in the context of geotechnical engineering.

A solution to the problem of requiring specialised equipment was developed in part by Brandow *et alia* [21]. This system used non-metric cameras of both 55mm×55mm and 35mm format, a cartographic standard stereocomparator and the DLT approach. The system was developed for use in underground coal mines, and so object space control was contrived by means of a precision fabricated space frame constructed from aluminium bar, placed at the foot of the coal face, as shown in Figure 3.12(a). Part of the aim of this work was to compare the photogrammetric method with the traditional geological compass technique. Following a rigorous series of tests, which included using the system in a children's playground in order to compare



(a) aluminium frame for object space control in place before a coal face.



lower hemisphere projection of poles

1-11 test surfaces, with:
- O geological compass measurement
- □ photogrammetric measurement
- ● photogrammetric measurements of other surfaces

(b) comparison between geological compass and photogrammetric measurements of discontinuity orientation.

Figure 3.12 Object space control system for non-metric photogrammetry applied to rock mass geometry, and typical discontinuity orientation results (from Brandow *et alia* [21]).

plane orientations measured by compass, photogrammetric and ground surveys, it was concluded that such a photogrammetric technique was not only technically feasible but practical. Indeed, the feasibility tests indicated that, using the ground survey results as a datum, the compass method gave orientations typically 4° in error whereas photogrammetry resulted in 1° error. This can be clearly seen in Figure 3.12(b).

Although the work of these authors would appear to have demonstrated beyond doubt the efficacy of the non-metric techniques for rock mass geometry, a number of facets to the investigation bear criticism, as the following four numbered points demonstrate.

(i) The space frame used for object control was composed of cubes with 600mm sides. Such a system would be manifestly inadequate for applications of typical rock engineering scale—say, of the order of a few metres—as photogrammetric measurements should not extrapolate beyond the control points.

(ii) The space frame was oriented using a geological compass. Although the test results indicated good accuracy for the photogrammetry, errors resulting from this imprecise orientation were not considered.

(iii) A simple least-squares solution for the mean plane passing through a number of points (between four and ten) measured on each discontinuity was performed, with the result being compared to the mean of several compass orientation measurements. The mathematical validity of such a comparison was not addressed.

(iv) A high-accuracy (±1μm) analytical stereocomparator was used. In general, as engineers would not have access to such an instrument, a technique relying on such equipment is impractical.

Use of a non-metric system using the bundle method of self-calibration, a computer digitising tablet for measurement of photograph coordinates and a microcomputer for data reduction has been reported by Kondoh *et alia* [73] and Kyoya *et alia* [79]. This system was used, respectively, in the context of rock slope stability and assessing the development of fracturing in a rock mass. No assessment of error is given in either of these publications but, as a digitising tablet is used for measurement of photograph coordinates, a comparison with the work performed by Carson [23, 24] and Oimoen [103] (see sub-Section 3.2.3, above) means the conclusion must be drawn that it cannot be good.

A photogrammetric system that did not depend on any sophisticated equipment has been described by Hagan [55]. As Figure 3.13 shows, two 35mm

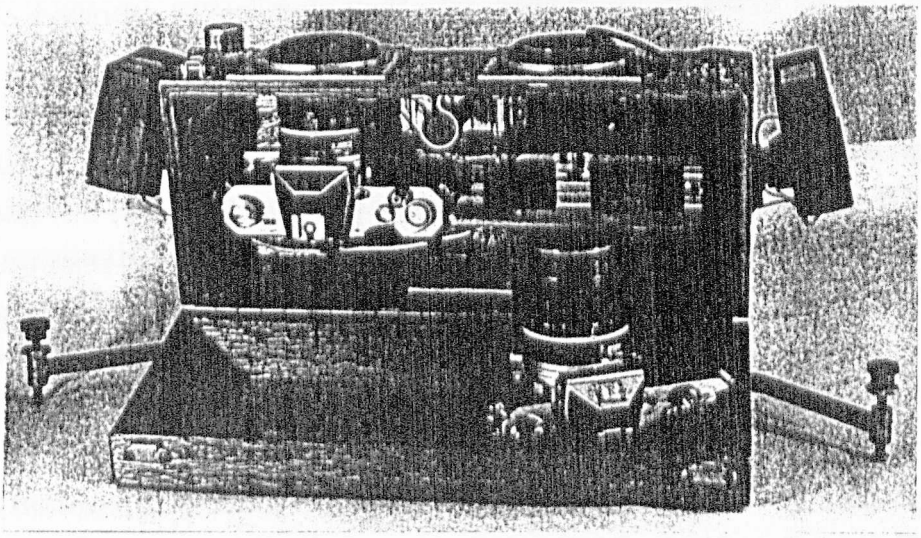


Figure 3.13 Stereoscopic non-metric camera system (from Hagan [55]).

cameras were mounted, with optical axes parallel and shutter releases synchronised, in a robust container. This enabled pairs of stereoscopic photographs to be taken, which were subsequently viewed under a simple pocket stereoscope, and homologous points on discontinuity surfaces imaged in each photograph marked. The photographs were then transferred to a digitising tablet and the photograph coordinates of the marked points determined.

The photogrammetric analysis consisted of the application of the parallax equations (equations (3.03), (3.04) and (3.05), above), with control being provided by the orientation of the camera optical axes (this was measured using a magnetic compass) and the camera separation. Although the author reported final errors in orientation of ±2°, this is surprising, given that no form of calibration—either on- or off-the-job—was undertaken, and that the parallax equations—applicable only to metric photography—were used. Whilst the system seemed to give acceptable results, its non-rigorous nature means that it must be disregarded.

Similar non-rigorous procedures were adopted by Sullivan [131]. In this work, the author used an extensive literature review to conclude that sophisticated adjustment was unnecessary for rock mechanics applications, and that a ruler calibrated in 1mm divisions was sufficient for determination of photograph coordinates. However, as the various applications reviewed were not quantified in terms of required accuracy of object space coordinates, such a statement is worthless. Notwithstanding this, the author concluded that errors in discontinuity orientation would be in the region of 0.2°: this should be

compared to the error in discontinuity orientation of 2° reported by both Brandow *et alia* [21] and Allam [5] discussed above!

Other work concentrating on the measurement of discontinuity extent or trace length on rock faces has been reported by DeRycker [35] and Thomas *et alia* [133]. These authors all used single image photogrammetry, but none of it was rigorous in terms of assessing the various sources of error resulting from the photogrammetry itself. DeRycker's work involved using a digitising tablet to measure trace length directly from photographs, whereas the work of Thomas *et alia* was novel in that the photographic images were displayed on a computer screen, and a light pen used to determine image coordinates in terms of screen pixels.

3.4.3 Summary

The number of both the different applications of photogrammetry in rock mechanics, and the techniques used, is impressive. Unfortunately, the greater majority of these applications have used measurements of photograph coordinates directly, rather than processing them with a rigorous photogrammetric analysis.

The work of Allam [5], Brandow *et alia* [21], Miller *et alia* [96] and Ross-Brown [115], are the most thorough with respect to photogrammetric principles, and demonstrate beyond doubt the feasibility of using photogrammetry for rock mass geometrical problems. However, only the second of these works tackled the question of non-metric analytical photogrammetry, but this still used a cartographic-standard stereocomparator. The question of whether non-metric photography, coupled with low accuracy photograph coordinates and a rigorous photogrammetric analytical procedure, can be used to determine properties of rock mass geometry remains unanswered.

3.5 Synopsis

This Chapter has attempted to review the various techniques and equipment used in photogrammetry, for the purpose of identifying a means of applying photogrammetry to the measurement of rock mass geometry. Guiding the review has been four basic principles: the system must be flexible in use; the accuracy and precision of the equipment and techniques should be appropriate

for the intended use; the cost of the equipment must be modest; and the individual components of the system must be readily available.

Following a general overview, in which the relevant terminology of photogrammetry is summarised, non-metric analytical photogrammetry has been identified as the branch most likely to fulfil the above criteria. This style of photogrammetry requires a commercially available 'High Street', or amateur, camera coupled with an instrument for measuring photograph coordinates on two photographs. In order to optimise the process by which homologous points are identified in the photographs, this latter instrument should incorporate a stereoscope, to allow stereoscopic viewing of the images.

A non-metric camera can only be calibrated at the instant it is used to take a photograph, and so the processing of measured photograph coordinates into object space coordinates must make use of an analytical technique that uses 'on-the-job' calibration. The direct linear transform (DLT), using the coordinate system customarily applied in aerial photogrammetry, has been identified as appropriate in this case.

In order to measure photograph coordinates with a reasonable degree of accuracy and precision, an instrument that uses a system of linear bearings to move the photographs, and electronic encoders to measure the displacement, is required. Furthermore, for flexibility in use, the instrument should take the form of an analytical plotter, in which the movements are under closed-loop control of a computer.

No recorded previous use of such non-metric analytical photogrammetry in rock mechanics has been found. Indeed, previous applications of rigorous photogrammetry in rock mechanics have tended to use metric photography, and the question of required accuracy for measurement of photograph coordinates seems not to have been addressed. These applications have, however, demonstrated the feasibility of using photogrammetry for measurements of rock mass geometry.

Thus, armed with the knowledge gained from the reviews presented in this Chapter, a suitable route for the development of a non-metric analytical photogrammetric system appropriate for the measurement of rock mass geometry can be identified. The development of such a system is presented in Chapter 5, in terms of the following elements: the required accuracy; design and construction of an analytical plotter; a method of introducing photogrammetric control at rock faces; and enhancements to the DLT method to ensure acceptable results are computed.

Although these developments will result in an improved method of acquiring rock mass geometrical data, the question of how these data may best be analysed remains unanswered. Consequently, the next Chapter now goes on to consider how cluster identification schemes may be used in this context.

4 Cluster Identification Schemes and Fuzzy Sets

Identification of key features within large sets of data is often made difficult by the vast, and as a consequence confusing, quantity of information present in the data. Schemes whereby the data is grouped into clusters, such that the elements of each group possess greater similarity with each other than with the elements of other groups, are commonly employed as a means of identifying key features and hence introducing clarification.

All sciences in which large data sets play a part use cluster identification in this fashion, and rock mechanics is no exception. Here, the most common use of a cluster identification scheme is in the classification of rock discontinuity orientation data: clusters are sought in which groups of discontinuities are identified, the elements of each group being similar in that they are mutually sub-parallel. The review presented in Chapter 2 showed that contouring methods are most commonly used for this work, with the few objective cluster identification schemes that have been developed and applied exhibiting either deficiencies, or insufficient advantage, to be universally employed.

A robust and mathematically well-defined cluster identification scheme would be valuable as an analysis tool not only for cases where rock masses are to be characterised using discontinuity orientation but, if it could be extended in an appropriate manner, to cases in which other geometrical attributes were included. This Chapter seeks to review such cluster identification schemes, in an attempt to identify those that may be applied to such analysis.

This Chapter consists of seven Sections. The first of these presents a brief review of the fundamental types of cluster identification schemes, together with examples of their applications. On the basis of this review, partitioning algorithms and objective functions are selected for further study in Section 4.2. Example data presented in this Section demonstrate the difficulty surrounding the concept of set membership, and this concept is considered further in Section 4.3 in terms of uncertainty and fuzzy sets. This leads naturally to a discussion of fuzzy partitioning algorithms, and this is presented in Section 4.4 with

specific reference to the fuzzy *c*-means (FCM), fuzzy covariance (FCV) and fuzzy maximum likelihood estimate (FMLE) algorithms. The quality of the partitioning (in terms of the validity of the resulting clusters) is considered in Section 4.5, where a number of cluster validity functions are reviewed. The important question of how to condense the fuzzy set membership into a readily assimilable form based on single set membership is addressed in Section 4.6. Finally, a summary of the work is presented in Section 4.7.

4.1 Fundamental Clustering Methods

Identifying clusters within sets of data is a central task in most sciences where measured data are to be characterised. Späth [130] surveys the available methods and Blashfield *et alia* [17] provide a useful (but now somewhat dated) review of cluster analysis software. These workers all choose to cluster the clustering methods thus: partitioning methods; hierarchical methods; and miscellaneous methods. The brief description of clustering methods that follows will adhere to this classification.

4.1.1 Partitioning methods

This important group consists of iterative methods that seek to partition the data by repeating, until some termination criterion is reached, the cycle of: assigning entities to the 'nearest' or 'most similar' cluster; computing new cluster centroids; and determining whether any more reassignments are required. They consist of heuristic methods and methods that apply some statistical criterion to the analysis.

The family of heuristic methods is infinite, as these methods simply partition the data on an *ad hoc* basis—say, for example, by assuming the first random sample taken from a population is a cluster prototype, and then either assigning subsequent random samples to this cluster or using them in turn as new prototypes, depending on some measurement of similarity. Clearly, the clusters that result from such a method will tend to be random, as they depend to a great degree on the order in which the data are extracted to form the cluster prototypes. As such, these methods are neither robust nor widely used, and will not be considered further here.

Methods that use a statistical function tend to be based on either some combination of either the **inter-cluster covariance matrix** and the **intra-cluster**

covariance matrix, or the **within-group sum-of-squared-distances** criterion. The inter-cluster covariance matrix is defined as

$$\mathbf{W} = \sum_{j=1}^{n}\sum_{i=1}^{c}(\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T \tag{4.01}$$

the intra-cluster covariance matrix is defined as

$$\mathbf{B} = \sum_{j=i}^{c}\sum_{i=1}^{c-1}(\mathbf{v}_j - \mathbf{v}_i)(\mathbf{v}_j - \mathbf{v}_i)^T \tag{4.02}$$

and the within-group sum-of-squared-distances criterion is defined as

$$J_W = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}\left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2 \tag{4.03}$$

where n is the number of data, c is the number of clusters, u_{ij} is a member of $\mathbf{U}$, the $c \times n$ matrix of binary values indicating cluster membership, and $\mathbf{x}$ and $\mathbf{v}$ are column vectors representing the data and the centroids, respectively. In (4.03) the following are defined:

$$u_{ij} = \begin{cases} 1; & \mathbf{x}_j \in \text{cluster } i \\ 0; & \text{otherwise} \end{cases} \tag{4.04}$$

and

$$\left\|\mathbf{x}_j - \mathbf{v}_i\right\| = \left[\sum_{k=1}^{p}(x_{kj} - v_{ki})^2\right]^{1/2} \tag{4.05}$$

where p represents the number of dimensions in the clustering space.

The similarity between (4.01), (4.02) and (4.03) is clear, with all three functions being based on either scalar or matrix **measures of dissimilarity** between the data and the cluster centroids. The partitioning methods based on these functions attempt to optimise a particular criterion, for example trace($\mathbf{W}$), trace($\mathbf{W}^{-1}$), det($\mathbf{W}$) or the magnitude of J_W. The popularity of these methods becomes clear once the nomenclature specific to individual fields has been deciphered. For example, the covariance matrices are variously known in the literature as orientation matrices [44], sum-of-squares-and-products (SSP) matrices [90, 92] or scatter matrices [15]. Similarly, examples of pseudonyms for the J_W criterion include the WGSS (within-group sum-of-squares) criterion [15] and the variance squared criterion [86]. The former will be exclusively referred to here as **scatter matrices**, and the latter as the **sum-of-squared-distances** criterion.

The operation of partition methods can be broken down into three clearly defined procedures: generation of an initial partition; reassignment of data; and evaluation of the termination criterion. All of these steps have difficulties associated with their implementation, but experience has shown that partitioning methods are the most robust clustering methods available, and have thus been used most widely. For these reasons this method is used exclusively in this work, and so is described in greater detail in Section 4.2.

4.1.2 Hierarchical methods

Hierarchical methods operate stepwise, and at each step either combine two previously existing clusters into one (**agglomerative methods**) or split a previously existing cluster into two (**divisive methods**). In each case, there are n steps corresponding to the n data: divisive methods start with one cluster containing n elements and finish with n clusters each containing one datum; agglomerative methods do the reverse. In a divisive method, the cluster chosen for splitting may be, for example, the one with the largest sum-of-squared-distances; in an agglomerative method the two chosen for coalescing may be those with the lowest inter-cluster variance. As an alternative agglomerative technique, the data may simply be added to a cluster on the basis of some measure of dissimilarity from other data.

Figure 4.01 gives an example of this last type of agglomerative technique. The dissimilarity matrix at the left of the Figure shows how the data are related to each other, in terms of some arbitrary measure of dissimilarity. At each stage of the process, either the element-element pair or the cluster-element pair with

dissimilarity matrix:

	1	2	3	4	5
1	0.0				
2	2.0	0.0			
3	6.0	5.0	0.0		
4	10.0	9.0	4.0	0.0	
5	9.0	8.0	5.0	3.0	0.0

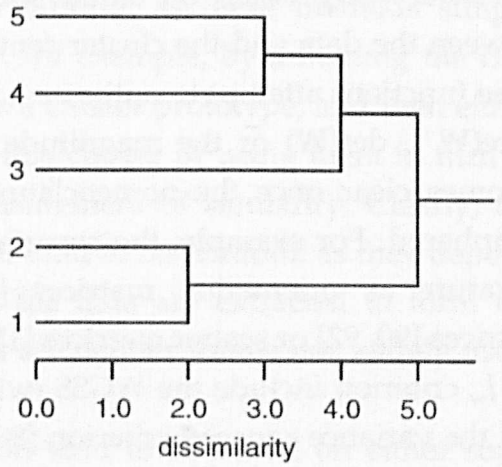


Figure 4.01 Single-link agglomerative clustering (data from Dunn and Everitt [37]).

the lowest dissimilarity is chosen for agglomeration. Consequently, with this data the process begins with elements 1 and 2, as shown in the accompanying dendrogram, at a dissimilarity of 2.0 units. Elements 4 and 5 are next to agglomerate, followed by element 3 joining their cluster. The dendrogram reveals at what level of dissimilarity the various elements can be considered to be part of the same cluster: in this case, at a value of 5.0 units, the data can all be considered to be part of the same cluster.

Unfortunately, there is no guarantee that, other than at the first and the last steps, the partition formed at any step will be the optimal partition for that step. The result of this is that these techniques show how one partition will optimally lead to another, rather than how the data may optimally be partitioned. Furthermore, as the underlying structure to these methods is dendritic, they are best applied to data possessing a dendritic structure, *e.g.* biological taxonometric data.

4.1.3 Miscellaneous methods

This group includes methods with some thermodynamical basis to the corresponding objective function, for example mimicking cooling and solidifying of a liquid, annealing of a metal, or maximising 'bond energy', and all manner of optimisation techniques. Some of these cannot be guaranteed to converge to anything other than a local minimum, and others are computationally intractable for all but trivially sized problems.

As Press *et alia* [106] point out, wide ranging practical experience with most of these methods is insufficient to determine their efficacy in general circumstances. As such, they have not been pursued further in this work.

4.1.4 Summary

On the basis of the descriptions presented above, we may conclude that the partitioning methods of cluster identification are currently most applicable to the problem of interpreting rock mass geometry. Hierarchical methods are not applicable as the data is not of a dendritic form, and the operation of the various miscellaneous methods is not sufficiently well known to guarantee any success. Indeed, Späth [130] notes that, out of partitioning, hierarchical, and other methods, only partitioning algorithms based on the sum-of-squared-distances criterion are computationally tractable, with—for well designed algorithms—a guarantee of convergence to a global minimisation (or maximisation) of the corresponding objective function.

As this work is concerned with the identification and application of a suitable cluster identification method to problems in rock mass geometry, it is partitioning algorithms that have been selected for use. Various aspects of partitioning algorithms and objective functions are now considered in greater detail in the following Section.

4.2 Partitioning Algorithms and Objective Functions

Heuristic partitioning algorithms were briefly discussed in sub-Section 4.1.1, and it was noted there that the family of these algorithms was infinite. The same is true of the families of partitioning algorithms based on statistical functions; any algorithm or objective function that optimises at certain partitions of the data may be regarded as suitable for cluster identification.

Sub-Section 4.1.1 briefly presented some common algorithms formulated around either **W**—the data scatter matrix (4.01)—or the sum-of-squared-distances criterion (4.03). These algorithms have been found over the years to perform well under different conditions and with different data, and have consequently tended to become standards against which other algorithms are judged. Almost all references in the literature to a partitioning algorithm will make reference to one of these algorithms. However, different algorithms partition data according to different rules, and different objective functions quantify different aspects of a partition. These facts mean that for the same data set partitioned with different algorithms and objective functions, different clusters will be identified. It is useful, therefore, to look at how these partitioning methods tend to behave when applied to data sets displaying particular geometrical attributes.

To this end, the following sub-Section considers how the partitioning results are influenced by the method of dissimilarity (or distance) measurement inherent in the algorithm. This leads naturally to a review—presented in sub-Section 4.2.2—of what is perhaps the most widely used partitioning algorithm: the so-called hard *c*-means algorithm.

4.2.1 Cluster shape and distance measurement

It is well known amongst numerical taxonomists that partitioning algorithms and objective functions tend to identify clusters of preferred shapes, and these shapes are dependent on the method used for determining the distances between cluster centroids and individual data. A qualitative example from the

analysis of rock discontinuity orientation data, using the customary method of counting the number of data appearing in a circular region (either manually on a hemispherical projection, or numerically on the surface of the unit sphere itself) illustrates this phenomenon. Implicit in the use of a circular region is the fact that we consider dissimilarity to be best determined as distance from the centre of the circle, rather than based on some intrinsic form of the data. The counting circle gives equal weight to distances, regardless of their orientation (on the projection or on the unit sphere) relative to the cluster. Not surprisingly, the data will tend to be partitioned into circular clusters.

The effect of this phenomenon on two aspects of partitioning methods—an algorithm itself and an objective function—are now described.

Behaviour of the min|W| algorithm

Everitt [39] considered the effect of choice of partitioning algorithm on the resulting partition. This author considered a number of algorithms based on maximising or minimising various functions of the scatter matrix, **W** (defined in (4.01) above) associated with the data and the cluster centroids.

As an example of Everitt's findings, Figure 4.02 shows the result of clustering a data set in $\mathbb{R}^2$§ which, to the eye, contains one dense circular cluster and one variable density elongate cluster. In this case, the min|**W**| algorithm has been applied with c=1,2 and 5. The results clearly demonstrate that such an algorithm has been unable to satisfactorily cope with at least two attributes of the visually apparent clusters: their differing shape and density. For the case of c=2, a number of the members of the elongate cluster have clearly been misclassified as belonging to the circular cluster. For the case of c=5, the elongate cluster has been partitioned into three visually acceptable clusters, but the circular cluster has additionally been partitioned into two unacceptable clusters.

The reason for this behaviour can be traced to the geometry implicit in the mathematics associated with the form of distance measurement used in the algorithm. If we multiply out the right hand side of (4.01) for the top left element of **W**, we find

$$W_{11} = (x_1 - v_1)(x_1 - v_1) \tag{4.06}$$

---

§ The symbol $\mathbb{R}^2$ can be read as 'two-dimensional space, the coordinates of which are real numbers'. Throughout this work, reference will often be made to data disposed in space of various dimensions, and this symbol represents convenient shorthand.

(a) c=1

(b) c=2

(c) c=5

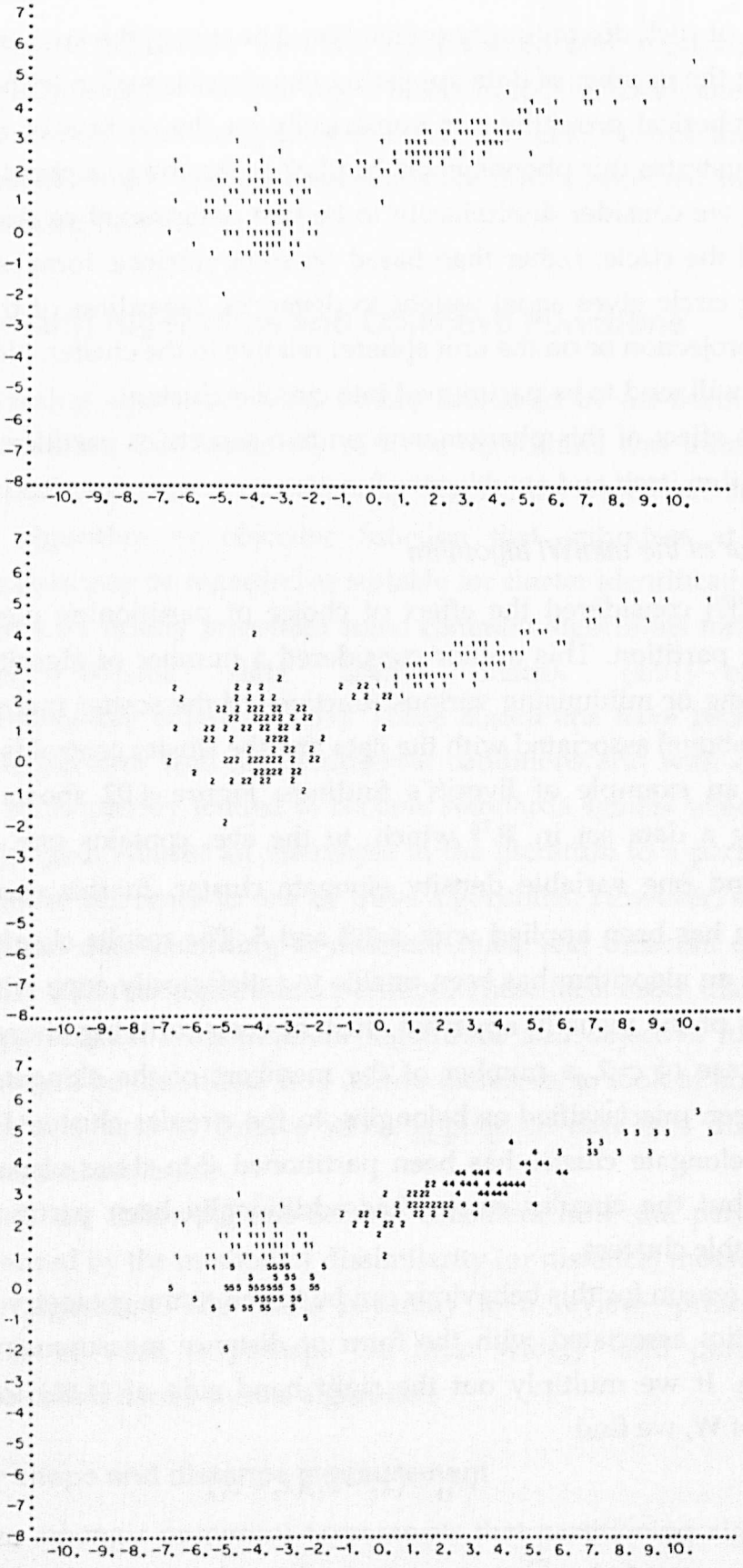


Figure 4.02 Result of partitioning a data set in $\mathbb{R}^2$ with the min|**W**| algorithm (from Everitt [39]).

which represents the ordinary notion of the square of a distance. We find that all elements of **W** take this form, indicating that **Euclidean distances** are used in the formulation. Now, because the Euclidean distances are isotropic (*i.e.* distances are not 'greater' measured in some direction than in others), we will find that the elements of **W** will tend to be of the same magnitude. Remembering that **W** represents the covariance matrix of the data, we know that if all of the elements of this matrix are of approximately equal magnitude, then there are no correlations greater than any other in the underlying data, with the result that an isotropic distribution is indicated. Of course, an isotropic distribution corresponds to circular clusters (in $\mathbb{R}^2$ space; spherical or hyperspherical in space of higher dimensions), and so the min|**W**| algorithm will tend to identify clusters of this form. This is borne out by the shape of the clusters shown in Figure 4.02.

Behaviour of the sum-of-squared-distances criterion

The sum-of-squared-distances criterion, equation (4.03) above, falls into the same trap as the algorithms based on the scatter-matrix, in that it tends only to identify isotropic clusters.

This criterion evaluates the **inner product** $(\mathbf{x}_j - \mathbf{v}_i)^T(\mathbf{x}_j - \mathbf{v}_i)$, which may be interpreted as the square of a Euclidean distance, *i.e.*

$$d^2(\mathbf{x}_j, \mathbf{v}_i) = (\mathbf{x}_j - \mathbf{v}_i)^T(\mathbf{x}_j - \mathbf{v}_i) \tag{4.07}$$

Such an inner product is variously known in the literature as a vector norm or a **metric**, and signified by the operator $\|\cdot\|$. Metrics are used extensively in partitioning algorithms because they possess the following geometrical properties [75]:

(i) $\|\mathbf{x}\|$ is a non-negative real number;

(ii) $\|\mathbf{x}\|=0$ if and only if $\mathbf{x}=0$; and

(iii) $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\|+\|\mathbf{y}\|$ (triangle inequality)

These properties are intuitive, and reflect our understanding of distances. However, the Euclidean metric is isotropic and incorporating it into an objective function ensures that the partitioning algorithm will tend—depending on the dimensionality of the space in question—to identify circular, spherical or hyper-spherical clusters.

Implications for the analysis of rock mass geometry

This phenomenon of identifying clusters of preferred shape must be considered in the context of the customary approach to partitioning orientation data; *i.e.* counting poles falling within a given area and then applying a probabilistic test to identify those areas containing significant numbers. In the presence of elongate clusters (say, for example, poles forming a small girdle on the hemisphere) the counting circle may be inappropriate, depending on the concentration of poles, size of the circle and 'length' and 'width' of the cluster. Under certain circumstances a visually apparent, and physically real, elongate cluster would not be identified. Thus, this implies that we will not discover any intrinsic structure to the data, but we will impose our own structure on the data. In general, we have no *a priori* knowledge of the shape of clusters we should find in discontinuity orientation data, indicating an algorithm that is not biased towards predetermined cluster shapes should be adopted.

This theme is continued in a later Section, but it is important to look in detail at other aspects of the operation of partitioning algorithms, in particular how the data are placed into clusters. This is the subject of the next sub-Section.

4.2.2 The hard *c*-means algorithm

Notwithstanding the shortcomings demonstrated by the application of the Euclidean metric to (4.03), this function is widely used and forms the basis of a number of algorithms. For example, Späth [130] introduces it as the *k*-means algorithm (because he assigns the variable k to the number of clusters identified) and Bezdek [15] refers to it as the *c*-means algorithm (because he assigns the variable c to the number of clusters). Both of these workers accept that their algorithms are a descendent of the ISODATA algorithm, the name given by earlier workers. Here, c is the adopted variable, and so *c*-means is appropriate. Furthermore, to uniquely identify this algorithm from later algorithms presented here, the adjective 'hard' is applicable.

Formulation of the HCM algorithm

The hard *c*-means algorithm (HCM) is computationally simple. Späth [130] and Bezdek [15] both give essentially the same procedure, which is as follows:

(1) Initialise **U**, *i.e.* partition the data into c clusters using some initialisation procedure and evaluate the cluster centroids $\mathbf{v}_i$ (i=1,c) using

$$\mathbf{v}_i = \frac{\sum_{j=1}^{n} u_{ij}\mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}} \tag{4.08}$$

(2) For each datum i=1,n experimentally transfer the datum from its previously assigned cluster j into cluster k=1,c ($k \neq j$). For any k that

$$\frac{n_j}{n_j - 1}\left\|\mathbf{x}_i - \mathbf{v}_j\right\|^2 > \frac{n_k}{n_k + 1}\left\|\mathbf{x}_i - \mathbf{v}_k\right\|^2 \tag{4.09}$$

then the datum is reallocated to the corresponding cluster that minimises the right hand side of (4.09), *i.e.*

$$\begin{aligned} u_{ij} &= 0 \\ u_{ik} &= 1 \end{aligned} \tag{4.10}$$

and both $\mathbf{v}_i$ and $\mathbf{v}_k$ are recomputed using (4.08).

(3) Evaluate J_W. If either J_W has decreased, or any point has been reallocated, return to step (2).

Whilst the algorithm is computationally simple, the incorporation of the Euclidean metric means that it obviously suffers from the deficiency of tending to identify isotropic clusters. A further drawback exists in terms of assigning set membership. At each reassignment, a datum ceases to become a member of one set and joins another, which—as will be shown in the next Division—can lead to bizarre results. It should be noted that binary cluster memberships are indicated by the membership function of (4.10), with the result that in step (2) of this algorithm it is not possible to apply the computational equivalent of 'datum i looks best in set k, but perhaps should still be considered as a member of set j'. The clustering is said to be hard—a result due to the use of a binary membership function—and it this behaviour that gives rise to the name hard c-means for the algorithm.

Operation of the HCM algorithm

The clustering operation of the HCM algorithm, and typical difficulties encountered due to the binary nature of cluster membership, are easy to demonstrate, especially in $\mathbb{R}^2$. A commonly used [15] data set for testing

clustering algorithms is Ruspini's‡ 'butterfly', illustrated in Figure 4.03. The results shown in this Figure represent the outcome of an analysis performed on a microcomputer spreadsheet program equipped with an iterative capability: details of the actual spreadsheet used are immaterial, as any current program will perform this analysis. The top section of Figure 4.03 illustrates the data and its visualisation in $\mathbb{R}^2$, with the following three sections illustrating three example analyses, each performed using a different initialisation of **U**. The initial and the final partitions, together with the coordinates of the cluster centroids and number of iterations required for convergence, are also given in the Figure.

It is apparent from Figure 4.03 that the hard *c*-means algorithm has insurmountable difficulties in dealing with a symmetrical data set that cannot be partitioned into two symmetrical clusters. Algorithmically, datum 8 can only be included in one of the two clusters, with the result that the centroids of the two clusters are different (relative to the data). Intuitively, we can see that the *x*-coordinate of the left hand cluster should be greater than -2.29 (the value when datum 8 is excluded from the cluster) and less than -2.00 (the value when datum 8 is included in the cluster).

Similarly, the dependency of the final partition on the initialisation of **U** is demonstrated by examples 1 and 2. The implication here is that hard *c*-means will not necessarily find a 'correct' partition, although it will find a numerically optimal partition. Furthermore, as examples 1 and 3 show, the convergence of the algorithm is also dependent on the initialisation; the closer the initialisation is to the final partition, the sooner the algorithm will converge.

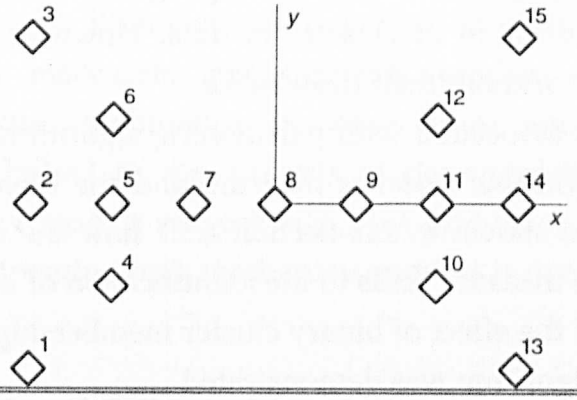
Of course, it is extremely unlikely that a set of rock mass geometry data will display the perfect symmetry of the butterfly data. What is likely though, is that the data will not display hard characteristics, *i.e.* set membership will not be binary. Indeed, the hemispherical projections of orientation data presented in Figures 2.16 and 2.17 demonstrate this: the clusters 'blend' into one another. Under such circumstances, algorithms that can somehow account for concurrent membership in more than one cluster, perhaps by assigning membership values that lie between zero and one, may be more efficacious. This concept is embodied in the mathematics of fuzzy sets, and is considered further in the next Section.

---

‡ *This reference has not been retrieved as part of the current work:*
Ruspini, E., 1970. Numerical Methods for Fuzzy Clustering, Inf. Sci., 2, pp.319-350.

'Butterfly' data:

element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x-coordinate	-3	-3	-3	-2	-2	-2	-1	0	1	2	2	2	3	4	3
y-coordinate	-2	0	2	-1	0	1	0	0	0	-1	0	1	-2	0	2



Example 1

Initial membership matrix:

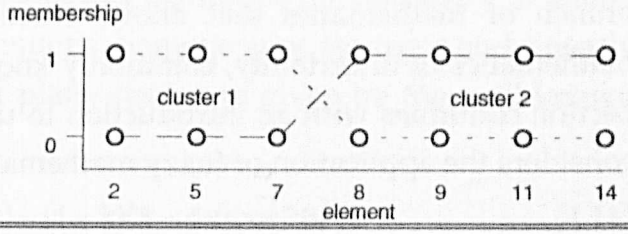
	element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	cluster 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	cluster 2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Membership matrix after 5 iterations:

	cluster 1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	cluster 2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

resulting centroids:

$x_1 = -2.29 \quad x_2 = 2.00$
$y_1 = 0.00 \quad y_2 = 0.00$



Example 2

Initial membership matrix:

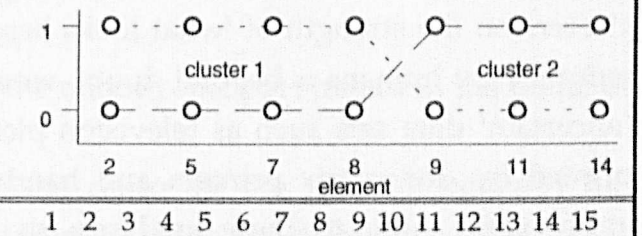
	element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	cluster 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	cluster 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Membership matrix after 5 iterations:

	cluster 1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
	cluster 2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

resulting centroids:

$x_1 = -2.00 \quad x_2 = 2.29$
$y_1 = 0.00 \quad y_2 = 0.00$



Example 3

Initial membership matrix:

	element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	cluster 1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	cluster 2	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Membership matrix after 3 iterations:

	cluster 1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	cluster 2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

resulting centroids:

$x_1 = -2.29 \quad x_2 = 2.00$
$y_1 = 0.00 \quad y_2 = 0.00$

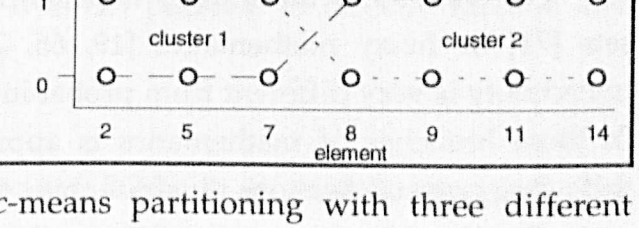


Figure 4.03 Butterfly data: hard *c*-means partitioning with three different initial partitions.

4.2.3 Summary

The literature demonstrates that partitioning algorithms are extremely popular tools for the identification of clusters in data. However, as this Section has shown, they are not without their drawbacks.

The problems associated with partitioning algorithms in general are due to the use of an isotropic distance measure and the binary nature of cluster membership. It was shown in sub-Section 4.2.1 how the use of the Euclidean metric as a distance measure leads to the identification of isotropic clusters, and in sub-Section 4.2.2 the effect of binary cluster membership on the operation of the hard *c*-means algorithm was demonstrated.

A method for overcoming the tendency for the Euclidean metric to identify isotropic clusters will be described later in this Chapter. For now, in an attempt to mitigate the effects of binary cluster membership, we must look to a branch of mathematics that allows fractional set membership. This is the mathematics of uncertainty, commonly known as fuzzy mathematics. The next Section continues with an introduction to uncertainty and fuzzy sets, and then considers the application of fuzzy mathematics to partitioning algorithms.

4.3 Uncertainty and Fuzzy Sets

When humans apply reasoning to the problem of clustering, the procedure for aggregating data into clusters is probably subjective, with data assigned to clusters on the strength of 'what looks best'. That such a technique is applied naturally by humans is beyond doubt: we are able to extract information from 'uncertain' data sets such as television pictures, newspaper photographs, text printed by dot-matrix printers and handwriting. Indeed, humans generally apply great store to these analyses; so much so, that it is possible that algorithms for partitioning data will only be considered acceptable if they produce the same partitions as a human skilled in the interpretation of such data!

The analysis of uncertainty is performed using the mathematics of fuzzy sets [71] or fuzzy mathematics [19, 65, 72]. It is important to realise that uncertainty is very different from probability, and that the application of either of these branches of mathematics is appropriate only in certain cases. The following two sub-Sections illustrate this.

4.3.1 Deterministic, probabilistic and fuzzy models in engineering rock mechanics

Bezdek [15] presents an illuminating discussion on how uncertainty can be attributed to either inaccurate measurements, random occurrences, vague descriptions, or some combination of these three, and how these three categories may be linked to the analysis of deterministic, probabilistic (or stochastic), and fuzzy models respectively. This discussion can usefully be set in the context of engineering rock mechanics, and this is done below.

Deterministic models

In the case of a **deterministic model**, the outcome can be predicted with absolute certainty, providing the circumstances defining the model are determined with absolute accuracy.

For example, the radial displacement of a point on the boundary of a long circular opening in an infinite, continuous, homogenous, isotropic and linearly elastic medium under conditions of plane strain are given by the well-known equation

$$u_r = -\frac{\sigma_1 a}{4G}\{(1+k)-(1-k)(3-4\nu)\cos 2\theta\} \tag{4.11}$$

where G = shear modulus of the rock,
ν = Poisson's ratio of the rock,
a = the radius of the opening,
σ_1 = the major principal stress,
k = the ratio of the major to the minor principal stresses in the plane of the excavation, and
θ = the angle subtended at the centre of the opening between the orientation of the minor principal stress and the point in question.

Providing all of these parameters can all be determined precisely, the displacement can also be determined precisely. The deterministic model demonstrates that there is no ambiguity or uncertainty about the outcome. Indeed, with a model of this kind it is not possible to directly include any method of accounting for uncertainty.

Probabilistic models

As has been discussed in sub-Section 2.2.2, it is widely accepted that discontinuity spacings follow the negative exponential distribution $f(x)=\lambda\exp(-\lambda x)$. Assuming that the Poisson process is valid for the occurrence

of discontinuities along a sampling line, then a **probabilistic model** can be developed using the equation

$$\Pr(k,x) = \frac{(\lambda x)^k e^{-\lambda x}}{k!} \tag{4.12}$$

This equation allows us to determine the probability of k discontinuities occurring in a length x of sampling line. The confidence with which $\Pr(k,x)$ can be estimated depends on the confidence that can be attached to λ, which in turn increases with the number of trials conducted (*i.e.* length of sampling line used) to determine it. Thus, uncertainty in the analysis of a probabilistic process can be reduced simply by performing more trials.

What can never be removed from the probabilistic process is chance: the chance that, even after sampling, say, 5000m of borehole core and finding a mean discontinuity spacing of 0.5m, a 100m length of core could be recovered free from discontinuities. Thus, although (4.12) allows us to quantify the uncertainty present in the probabilistic model, it does not allow us to predict it, *i.e.* we can quantify *when* an event *will* occur, rather than *whether* an event *has* occurred.

Fuzzy models

Unfortunately, uncertainty of the 'whether' kind abounds in rock mechanics. Vague descriptions are commonplace, with most nouns being modified by one or more adjectives: examples are high stress, strong rock, extremely weathered and good quality. Even quantities that are measured with precise and accurate equipment may be vague—orientation of a wavy (on the scale of the measuring instrument) discontinuity is a case in point. In such circumstances it is quite wrong to imbue the measured parameters with probabilistic attributes, for the simple reason that none are implied. The analysis should not continue with the use of probabilistic statements, but with the aid of a **fuzzy model**, which enables uncertainty to be quantified. An analysis of what we mean by 'strong rock' illustrates this.

Consider a rock mass classification scheme in which 'strong rock' is regarded as material for which the unconfined compressive strength is exactly 150MPa, and that this is applied to a population X consisting of n specimens of rock. The set of specimens regarded as strong, S_1, is then

$$S_1 = \{x \in X \mid s(x) = 150\,\text{MPa}\} \tag{4.13}$$

which is understood to mean 'S_1 is the set of elements x, drawn from the population X, for which the strength s is exactly equal to 150MPa'.

Of course, very few specimens will be members of the set S_1, if membership is understood to mean a strength of *exactly* 150MPa. The customary means of overcoming this deficiency is to admit specimens with a range of strengths into the set, thus:

$$S_2 = \{x \in X \mid s(x) = 150\text{MPa} \pm 10\text{MPa}\} \tag{4.14}$$

in which case S_2 is a less restrictive set than S_1, because it can contain all specimens with strengths between, and including, 140MPa and 160MPa.

However, we now find that a specimen with a strength of 139.9MPa would be excluded from S_2, even though it would sensibly be regarded as no different from a specimen with a strength of 140MPa. The problems arise from the binary-valued functions u_1 and u_2 used to indicate membership of the sets S_1 and S_2, namely

$$u_1(x) = \begin{Bmatrix} 1; & x \in X \mid s(x) = 150\text{MPa} \\ 0; & \text{otherwise} \end{Bmatrix} \tag{4.15}$$

and

$$u_2(x) = \begin{Bmatrix} 1; & x \in X \mid s(x) = 150 \pm 10\text{MPa} \\ 0; & \text{otherwise} \end{Bmatrix} \tag{4.16}$$

If we generalise these functions to allow for more values than 0 or 1, then it is possible to allow for the concept of 'strong rock' by assigning values to u which give **grade of membership** in the set 'strong rock', and thus overcome these problems. For example, we may wish to generalise the **membership function** to

$$u_3(x) = \begin{Bmatrix} 1; & 140\text{MPa} \le s(x) \le 160\text{MPa} & & \\ 0.95; & 130\text{MPa} \le s(x) \le 140\text{MPa} & \text{or} & 160\text{MPa} \le s(x) \le 170\text{MPa} \\ \vdots & \vdots & & \vdots \\ 0.05; & \cdots & & \cdots \\ \vdots & \vdots & & \vdots \end{Bmatrix} \tag{4.17}$$

We can assign as many values to u as we require, up to and including the case when u becomes a continuous function of $s(x)$. Thus, the fuzzy model retains, for further analysis, a large amount of the information regarding uncertainty that is present in the data.

By using this technique, we are able to deal with uncertainty where it occurs which, in the example given here, is in determining the strength of a rock specimen and then categorising the strength value. A probabilistic statement is wholly inappropriate under such circumstances: this would only allow us to determine what strengths to expect from the testing process, not how to quantify our uncertainty in categorising them. This distinction between probabilistic and fuzzy models is taken up in the next sub-Section.

4.3.2 Probabilistic *versus* fuzzy models

To determine whether fuzzy models are more appropriate than probabilistic models, the difference between probability and fuzziness must be illustrated. Continuing with the example on rock strength presented above, let us assume that the strength of a number of specimens of rock has been determined, and that as a result we have estimated that

$$\Pr(140\text{MPa} \le s(x) \le 160\text{MPa}) = \Pr(x \in S_2) = 0.95 \tag{4.18}$$

which, in the context of the example presented here, may be read as 'the probability that a specimen x, drawn at random from the population X, will be a member of the set S_2, which consists of those specimens for which the strength s lies between 140MPa and 160MPa, is 0.95'. Because the specimen x has not yet been tested, this is a **prior probability**.

Using Bayes theorem, upon determination of the strength of the next specimen of rock, the prior probability $\Pr(x \in S_2) = 0.95$ becomes a **posterior probability**, with

$$\Pr(x \in S_2 \mid x) = \left\{ \begin{matrix} 1; & \text{if } s(x) = 150\text{MPa} \pm 10\text{MPa} \\ 0; & \text{otherwise} \end{matrix} \right\} \tag{4.19}$$

The left-hand side of (4.19) is read as 'the probability that x is a member of set S_2, on the basis of having measured the value of x in question'. This result shows that the uncertainty about the occurrence of the event—in this case the strength of the specimen—disappears. However, regardless of the outcome of the strength test, the membership value defined in (4.17) above—namely $u_3(x) = \{1.00; \quad 140 \le s(x) \le 160\}$—still holds, because this is a measure of the extent to which $s(x)$ is regarded as 'strong rock' and must remain unaltered.

Thus, the application of a probabilistic model results in certainty, whilst the application of the fuzzy model maintains uncertainty. If, after strength

testing for example, we are uncertain as to whether the rock is strong or not, then the use of fuzzy analytical methods are indicated.

This concept is immediately applicable in the context of rock mass geometrical data. If, for example, we have *a priori* knowledge of both the number of discontinuity sets in a rock mass and the probabilistic distributions of discontinuity normals within these sets, and furthermore, we know that any discontinuity within the rock mass is a member of one of these sets, then we are correct in posing probabilistic questions. Under such circumstances, questions concerning topics such as the probability of a sampled discontinuity possessing an orientation within certain limits, or the probability of intersecting a discontinuity of a given orientation in a borehole, are valid.

However, we do not in general have such *a priori* knowledge, and so should not pose such questions. If we do pose such questions, we are testing against the hypothesis that there are a certain number of sets, with specific distribution parameters, and that all discontinuities belong to one of these sets. The form that our questions should take is 'is this discontinuity a member of that set?', and the answers we should be willing to accept are of the form 'highly unlikely'. Under such circumstances, a fuzzy model is applicable.

Analysis using fuzzy models has found some application in geomechanics, for example: tunnel support design [42]; rock mass classification [100, 101, 127]; hazard evaluation [99]; and assessment of rock slope stability [121]. Its use in the analysis of rock mass geometry appears, however, to be unexplored.

4.3.3 Summary

This Section has considered the differences between deterministic, probabilistic and fuzzy models, and given examples of their application to problems in engineering rock mechanics. The simple examples of displacement around an underground opening, occurrence of a discontinuity in borehole core and strength of rock illustrate how these three concepts should be applied. The example highlighting the difficulties associated with categorising the results of strength testing demonstrates how this kind of uncertainty, as opposed to determining the probability of a given strength occurring, can only be analysed using fuzzy methods. This indicates that a cluster identification scheme for rock mass geometry must be implemented using this technique if it is to be meaningful.

The next Section considers how fuzzy partitioning algorithms have been applied to other clustering problems in the past, and gives some insight into how they may be applied to the problem of cluster identification in rock mass geometry.

4.4 Fuzzy Partitioning Algorithms

Most work associated with the development of fuzzy partitioning algorithms has been connected with the solution of problems in pattern recognition. However, these algorithms are completely general, in that the originators of them have not concentrated on particular types of data. Pattern recognition can be thought of as identifying clusters in data sets, as the aim is to partition the data into groups, the elements of which are mutually similar. The basic principle in all pattern recognition is the same: to take data, the distribution and cluster membership of which is unknown, and determine its 'best' partition into clusters.

General algorithms developed for pattern recognition may be applicable to the analysis of rock mass geometry through the identification of clusters in geometrical data. Taking the particular example of discontinuities in rock masses, intuitively we can see that such pattern recognition could be based on identifying parallelism, although attributes such as, for example, genesis, roughness, aperture and extent could be used. Continuing with the example of discontinuity orientation, if we use unit normal vectors to the discontinuity planes to represent orientation, then parallelism is apparent in terms of clustering of the ends of these vectors on the surface of the unit sphere. The pattern recognition problem can therefore be couched in terms of finding patterns in the three-dimensional data of direction cosines representing the normals to discontinuity surfaces.

In fuzzy partitioning algorithms, an individual datum is assigned a membership value, u, which reflects the degree of certainty with which it possesses membership of a set—in the present context, 'membership of a given cluster of orientations'. As has been discussed in the previous Section, membership values are positive real numbers such that $0 \leq u \leq 1$, with $u=0$ representing total uncertainty and $u=1$ representing total certainty. This should be compared with hard memberships (as used in the hard c-means algorithm), which are binary valued. By permitting degrees of membership to take on values other than 0 and 1, we are admitting the possibility of a datum being a

member of a given set, whilst not excluding its membership in others. Put another way, we assume that a datum is a member of all clusters concurrently, but it is more likely to be a member of one than of the others.

The development and application of general pattern recognition algorithms using fuzzy partitioning algorithms are covered in detail in the monographs by Bezdek [15] and Kandel [65]. The work by Bezdek in particular is extremely thorough, and is referred to extensively in the rest of this Chapter.

This Section considers four fuzzy partitioning algorithms—the uni-modal, fuzzy *c*-means, fuzzy covariance and fuzzy maximum likelihood estimation algorithms. There are advantages and disadvantages to each of the algorithms, as the following sub-Sections demonstrate. Throughout, reference is made as to how these algorithms might be applied to the problem of cluster identification in rock mass geometry data, with particular reference to discontinuity orientation. Finally, a brief review of additional aspects associated with the application of fuzzy partitioning algorithms is presented. This includes partitioning using non-metric methods, and partitioning data in which the components of the data vectors have different physical units.

4.4.1 The uni-modal algorithm

Probably the earliest development of a fuzzy cluster identification method was reported by Gitman and Levine [47]. Strictly, their uni-modal algorithm was not a partitioning scheme, but implemented a particularly intuitive hierarchical clustering scheme.

The scheme commenced with the assignment of membership values to the data, and the datum with the largest value was considered to be the mode of a cluster. The data around this datum, which must possess lower membership values, were then assigned to the same cluster. Figure 4.04 shows how the algorithm operates for data in $\mathbb{R}^1$, how new clusters are identified in terms of their prototypical mode, and how these clusters would coalesce at lower levels of certainty.

It is useful to attach a geographical interpretation to the operation of this algorithm. Following the assignment of membership values, each datum can be considered as representing a region of landscape possessing a particular 'height'. If this landscape is then considered to be totally submerged, the clustering operation simply represents a case of identifying the mountains as they appear from the ocean as it recedes. The highest summit appears first, and becomes the mode of the first cluster. Other, quite unconnected, peaks will

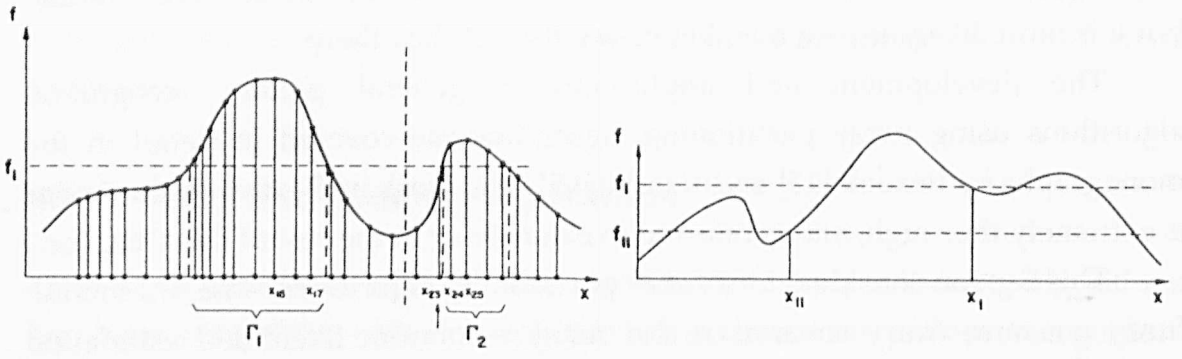


Figure 4.04 The uni-modal fuzzy clustering technique (from Gitman and Levine [47]).

appear as the ocean recedes, and at certain stages some of these peaks may merge to form a single mountain. Land located at the mid-point of a valley or col could not be regarded as belonging any more to one of the adjoining peaks than to the other.

Such a procedure is almost identical to that followed in the customary analysis of rock discontinuity orientation data. The heights represent the number of normals falling within the counting circle, and it is then simply a case of connecting together points of equal height. Thus, this algorithm seems ideally suited to the analysis of orientation data, and as such was programmed by this Author as part of this investigation.

Unfortunately, this algorithm has a number of shortcomings, not the least of which are the lack of an objective function basis and the implicit use of a Euclidean distance measure. It is thus impossible to determine whether the resulting partition is optimal, and whether sub-peaks should be regarded as separate entities or coalesced, as well as avoid bias towards circular clusters. It is possible to overcome these shortcomings, but in so doing the algorithm ceases to be the uni-modal algorithm, and becomes the fuzzy *c*-means algorithm instead. It is therefore appropriate to continue with a description of the fuzzy *c*-means algorithm, and make no further mention of the uni-modal algorithm.

4.4.2 The fuzzy *c*-means algorithm

Both Bezdek [15] and Kandel [65] have used fuzzy objective functions in the development of general pattern recognition algorithms. Bezdek, on the basis of a wide literature survey, discusses in detail the development of fuzzy objective

functions for pattern recognition, based mostly on extensions to the hard *c*-means algorithm.

Formulation of the FCM algorithm

A fuzzy extension of the hard *c*-means algorithm is intuitive. Rather than assign hard membership values using (4.04), (4.09) and (4.10), a similarity function based on the Euclidean metric (4.07) is used to determine fuzzy membership values. Thus, the objective function associated with this algorithm is

$$J_m(\mathbf{U},\mathbf{V}) = \sum_{j=1}^{n}\sum_{i=1}^{c}\left(u_{ij}\right)^m\left(d_{ij}\right)^2 \tag{4.20}$$

where $d_{ij} = \left\|\mathbf{x}_j - \mathbf{v}_i\right\|$ (the Euclidean metric), $0 \le u_{ij} \le 1$ and $1 \le m < \infty$. The parameter m is termed the **fuzzy exponent**.

There are three matrix arrays inherent in (4.20)—the **membership matrix, U**, the **data matrix, X** and the **centroid matrix, V**. It is essential to be aware of the architecture of these matrices in order to comprehend the operation of a fuzzy partitioning algorithm. These matrices take the following form:

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & \ddots & & \\ \vdots & & u_{ij} & \\ & & \ddots & \vdots \\ u_{c1} & & \cdots & u_{cn} \end{bmatrix} \tag{4.21}$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & \ddots & & \\ \vdots & & x_{ij} & \\ & & \ddots & \vdots \\ x_{p1} & & \cdots & x_{pn} \end{bmatrix} \tag{4.22}$$

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1c} \\ v_{21} & \ddots & & \\ \vdots & & v_{ij} & \\ & & \ddots & \vdots \\ v_{p1} & & \cdots & v_{pc} \end{bmatrix} \tag{4.23}$$

where c=number of clusters, n=number of data and p=number of dimensions in the clustering space. The matrices represented by (4.21), (4.22) and (4.23) also

represent the actual form of the arrays used when this fuzzy partitioning algorithm is implemented on a computer, as described in Appendix D.

Many possibilities exist for a method of computing u. The membership function that is used varies from algorithm to algorithm, and an objective choice is not straightforward [28, 144]. In general, some inverse function of distance is used, with the result that for small distances the value of u is close to unity, decreasing with increasing distance. Bezdek [15] proves that

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}} \tag{4.24}$$

will provide for at least a local minimisation of J_m. This function has the properties that $u_{ij} \to 1$ as $d_{ij} \to 0$ (providing that in any numerical application the singularity at $d_{ij}=0$ is properly dealt with), and u_{ij} decreases monotonically as d_{ij} increases. Interestingly, for the case of orientation data, the use of such a membership function produces a scheme very similar to one using a hemispherical projection and counting circles with non-constant weighting functions applied. This is the enhancement suggested by Bayer [14], and described in sub-Section 2.5.1, but developed using totally different reasoning.

Inspection of (4.24) reveals that

$$\sum_{i=1}^{c} u_{ij} = 1 \tag{4.25}$$

i.e. each column of **U** sums to unity. This result is to be expected: whilst we may be uncertain about the allocation of datum j to a particular cluster i, there can be no uncertainty about it being allocated somewhere!

The fuzzy c-means (FCM) algorithm is, then, similar to the hard c-means algorithm, and may be formalised as follows:

(1) Initialise **U**, *i.e.* partition the data into c clusters using some initialisation procedure.

(2) Evaluate the cluster centroids $\mathbf{v}_i$ (i=1,c) using

$$\mathbf{v}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{n} (u_{ij})^m} \qquad \text{for } i = 1, c \tag{4.26}$$

(3) Compute membership values using (4.24).

(4) Evaluate $\max_{ij} \left\| u_{ij} - \hat{u}_{ij} \right\|$, where $\hat{u}_{ij}$=corresponding value of u_{ij} from the previous iteration and $\| \cdot \|$ is a suitable matrix norm. If the magnitude of the norm exceeds a threshold value ε ($0 < \varepsilon < 1$), repeat from step (2).

Operation of the FCM algorithm

As Bezdek [15] demonstrates, the weighting parameter m in (4.20) controls the 'fuzziness' of the resulting partitions. Figure 4.05 replicates his example using the butterfly data, and shows how, for m=2, the final membership values are lower, and are more variable, than for the case of m=1.25. These two characteristics both indicate increased fuzziness: from the point of view of an individual datum, lower membership values represent greater uncertainty of membership in a particular set; and from the point of view of the set as a whole, increased variability represents uncertainty as to the limits of the set.

One aspect of introducing the fuzzy exponent m that is implicit in the results shown in Figure 4.05, is the increased number of iterations required for convergence. The increase (over the number required for convergence with hard c-means) for this data is not large, but is to be expected, as the elements of **U** are now positive real numbers rather than binary integers.

A further aspect not demonstrated in Figure 4.05 is the problem of initialising **U**, which can be considered in two parts: how the initialisation affects the final partition; and how it affects the convergence. The final partition is not as dependent on the initialisation as was the case with hard c-means, as it is possible—up to the limit of machine precision—to assign membership values such that within any column of the membership matrix both the criteria of $u_{1j} = u_{2j} = \cdots = u_{cj}$ and (4.25) hold, with the result that symmetry in set membership is preserved. This is demonstrated in Figure 4.05 for the case of

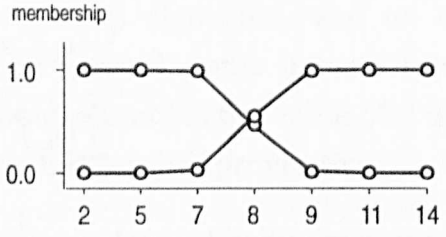
Example 1: **U** after 6 iterations with m=1.25

element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cluster 1	0.99	1.00	0.99	1.00	1.00	1.00	0.99	0.47	0.01	0.00	0.00	0.00	0.01	0.00	0.01
cluster 2	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.53	0.99	1.00	1.00	1.00	0.99	1.00	0.99

resulting centroids:

$x_1 = 0.83 \quad x_2 = 5.14$

$y_1 = 2.00 \quad y_2 = 2.00$



Example 2: **U** after 6 iterations with m=2.00

element	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cluster 1	0.86	0.97	0.86	0.94	0.99	0.94	0.88	0.50	0.12	0.06	0.01	0.06	0.14	0.03	0.14
cluster 2	0.14	0.03	0.14	0.06	0.01	0.06	0.12	0.50	0.88	0.94	0.99	0.94	0.86	0.97	0.86

resulting centroids:

$x_1 = 0.85 \quad x_2 = 5.14$

$y_1 = 2.00 \quad y_2 = 2.00$

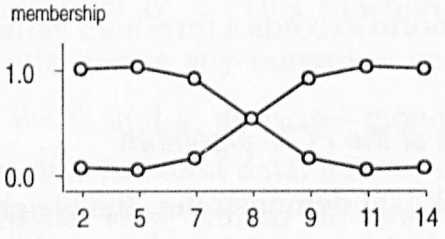


Figure 4.05 Illustration using the butterfly data of fuzziness as a function of weighting exponent m (after Bezdek [15]).

m=2. However, certain initial partitions are known to be unsuitable (for example, when all columns of **U** are identical), and certain partitions are put forward on the basis of heuristics. For example, Bezdek advocates choosing an 'average' partition using

$$\mathbf{U}^{(0)} = \alpha\mathbf{U} + \beta\mathbf{W} \tag{4.27}$$

where $\alpha = 1 - 1/\sqrt{2}$, $\beta = 1/\sqrt{2}$, $\mathbf{U} = [1/c]$ and **W** is a hard partition. Initialisation of **U** when this algorithm is applied to orientation data is taken up further in Chapter 6. The problem of convergence is more difficult and, as far as this Author has been able to determine from the literature, has not been solved.

4.4.3 The fuzzy covariance algorithm

Implicit in the fuzzy c-means algorithm is the use of the Euclidean metric for determining distances and this, as in all other clustering schemes that use it, results in the identification of isotropic clusters. Gustafson and Kessel [54] attempted to circumvent this problem by replacing the Euclidean metric with the **Mahalanobis metric** and using **Mahalanobis distances** in their partitioning

algorithm. Mahalanobis distances are commonly used in statistical work where it is important to take account of the underlying distribution of data. In order to demonstrate how the Mahalanobis distance is constructed, it is useful to rewrite (4.07) as

$$d^2(\mathbf{x}_j, \mathbf{v}_i) = (\mathbf{x}_j - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_j - \mathbf{v}_i) \tag{4.28}$$

where **A=norm inducing matrix**. Providing **A** is positive-definite, then the resulting **weighted inner product** is a metric. For the case of the Euclidean metric, the identity matrix is used as the norm inducing matrix, *i.e.* $\mathbf{A}_E=\mathbf{I}$. For the Mahalanobis metric however, the norm inducing matrix is defined as the inverse of the sample covariance matrix, *i.e.*

$$\mathbf{A}_M = [\text{cov}(X)]^{-1} \tag{4.29}$$

where the sample covariance matrix is computed as

$$\text{cov}(X) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{v})(\mathbf{x}_i - \mathbf{v})^T \tag{4.30}$$

for n data in the sample. In a partitioning algorithm, however, there are c cluster centroids, and so it should be borne in mind that subscripts should be added to both the entire left-hand side and the centroid vectors **v** on the right-hand side of (4.30). A consequence of this is that there are c norm-inducing matrices defined by (4.29), one for each cluster. The use of the sample covariance matrix in this way means that the effect of statistical dependence between the components of the vector joining a datum to the centroid is taken into account when determining the length of the vector. The result of this is that distances thus computed reflect the underlying distribution of the data, with the possibility that ellipsoidal clusters may be identified.

Formulation of the FCV algorithm

The application of the Mahalanobis norm to the FCM algorithm follows in a logical way, with (4.29) replacing the Euclidean norm wherever it appears. Gustafson and Kessel commence by defining the **fuzzy scatter matrix, SF**, as

$$\mathbf{SF}_i = \sum_{j=1}^{n} (u_{ij})^m (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T \tag{4.31}$$

which connects the covariance matrix for the cluster with the membership values in that cluster. The name for this matrix follows from a comparison with (4.06), the data scatter matrix. In order to ensure that the resulting algorithm

could be numerically minimised, Gustafson and Kessel then normalised the fuzzy scatter matrix to produce a fuzzy Mahalanobis-like norm-inducing matrix thus

$$(\mathbf{A}_{FM})_i = \left[\rho_i \det(\mathbf{SF}_i)\right]^{(1/p)} (\mathbf{SF}_i)^{-1} \tag{4.32}$$

where the scaling term, $\left[\rho_i \det(\mathbf{SF}_i)\right]^{(1/p)}$, is used to add the constraint $\det(\mathbf{A}_{FM})_i = \rho_i$. Geometrically, this constraint ensures that, for all clusters i, the u_{ij} are distributed in a hyperellipsoid, the volume of which is ρ_i. Minimising the objective function is, then, equivalent to minimising the volume of this hyperellipsoid, which is the same as ensuring the data are partitioned into clusters of small variation in u.

Whilst the subscript i in (4.31) and (4.32) trivially indicates that a different norm inducing matrix is calculated for each cluster, the result of this is more profound: the clusters may all take on different shapes, and the shapes are defined by the data themselves. Although they are not directly used in the numerical implementation of the algorithm, Gustafson and Kessel coined the term **fuzzy covariance matrix** for

$$\mathbf{F}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T}{\sum_{j=1}^{n} (u_{ij})^m} \tag{4.33}$$

from which the algorithm gets its name.

The enhanced fuzzy c-means algorithm, fuzzy covariance (FCV) is, then:

(1) Initialise **U**, *i.e.* partition the data into c clusters using some initialisation procedure.
(2) Evaluate the cluster centroids $\mathbf{v}_i$ (i=1,c) using (4.26).
(3) Calculate the fuzzy scatter matrices, using (4.31), together with their inverses and their determinants.
(4) Calculate the norm inducing matrices $(\mathbf{A}_{FM})_i$ using (4.32).
(5) Compute membership values using (4.24) and the norm inducing matrices from step (4).
(6) Evaluate $\max_{ij}\left\|u_{ij}-\hat{u}_{ij}\right\|$, where $\hat{u}_{ij}$=corresponding value of u_{ij} from the previous iteration and $\|\cdot\|$ is a suitable matrix norm. If the magnitude of the norm exceeds a threshold value ε ($0<\varepsilon<1$), repeat from step (2).

Application of the FCV algorithm

Gustafson and Kessel demonstrated the efficacy of the FCV algorithm on a synthesised data set in $\mathbb{R}^2$. This data set consists of two long, chain-like, data structures arranged approximately mutually perpendicular. The salient results from this demonstration are reproduced in Figure 4.06.

In this example, the circular clusters identified by the FCM algorithm and attributable to the isotropic Euclidean norm are clearly demonstrated. Similarly, it is quite clear that the use of the Mahalanobis-like norm-inducing matrix permits extremely elongate clusters to be identified, circumstances that are impossible with the Euclidean norm.

There is a further advantage in using this algorithm, that stems from the geometry of the norm-inducing matrices, and that it that it is straightforward to extract information from these matrices that relates to the geometry of the final clusters. This is described below.

Cluster orientation

Additional geometrical information that relates to the orientation of the clusters is available from the fuzzy covariance matrix. Table 4.01 replicates the information given by Bezdek relating to the sample covariance matrices and the fuzzy scatter matrices (at the termination of iterations) for the example of Figure 4.06.

Element coordinates		Final membership values (1 cluster only)	
x	y	FCM	FCV
-9.75	-0.15	0.25	0.03
-6.44	0.34	0.18	0.03
-4.69	-0.30	0.11	0.03
-2.04	0.37	0.08	0.02
-1.24	0.45	0.10	0.32
0.33	-0.08	0.22	0.02
5.04	-0.21	0.98	0.03
5.86	-0.25	0.97	0.03
7.54	0.16	0.91	0.03
7.67	0.24	0.91	0.03
-0.30	-8.07	0.30	0.97
0.13	-7.13	0.29	0.97
-0.37	-5.18	0.19	0.97
0.03	-3.33	0.13	0.97
0.35	-2.63	0.13	0.98
0.23	-2.68	0.12	0.98
-0.05	-2.00	0.07	0.98
0.41	0.37	0.29	0.74
0.69	4.75	0.57	0.97
0.74	8.87	0.57	0.97

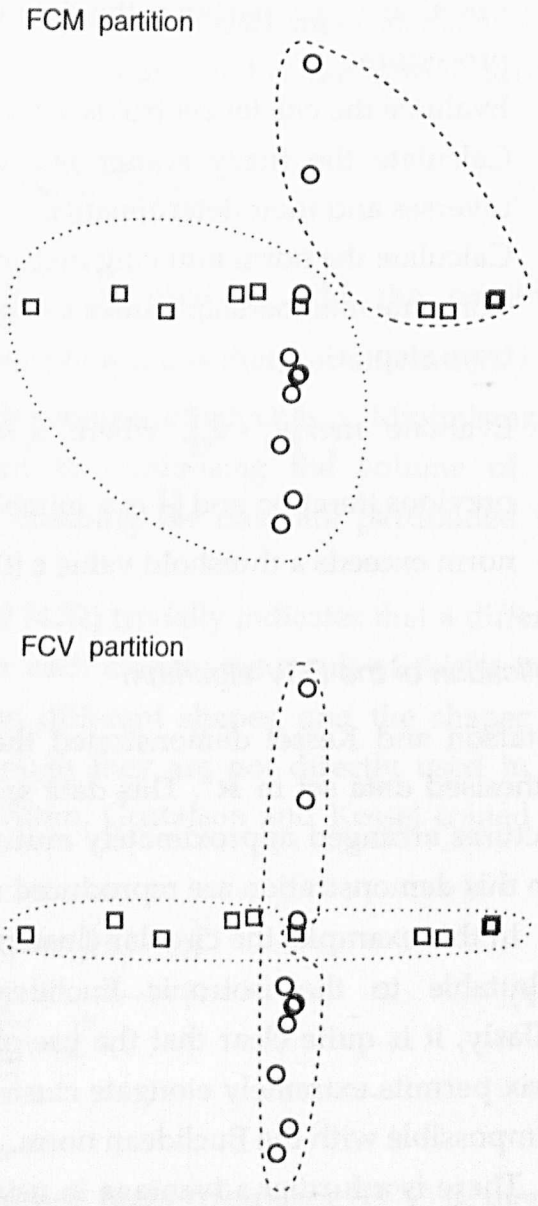


Figure 4.06 'Cross' data: application of the fuzzy c-means and the fuzzy covariance algorithms (after Gustafson and Kessel [54]).

In each case, the small off-diagonal terms indicate little correlation between x and y coordinates, demonstrating a linear structure. The leading diagonal terms indicate large variance in the x coordinates for cluster 1 and large variance in the y coordinate for cluster 2, demonstrating their alignment parallel to the coordinate axes.

Table 4.01 Cross data: sample covariance matrices and the fuzzy scatter matrices (after Bezdek [15]).

	covariance matrices	fuzzy scatter matrices
cluster 1	$\mathbf{cov}_1 = \begin{bmatrix} 33.92 & -0.03 \\ -0.03 & 0.07 \end{bmatrix}$	$\mathbf{SF}_1 = \begin{bmatrix} 36.33 & 0.88 \\ 0.88 & 0.05 \end{bmatrix}$
cluster 2	$\mathbf{cov}_2 = \begin{bmatrix} 0.13 & 1.50 \\ 1.50 & 24.44 \end{bmatrix}$	$\mathbf{SF}_2 = \begin{bmatrix} 0.06 & 0.72 \\ 0.72 & 25.65 \end{bmatrix}$

Extracting the eigenvalues and eigenvectors of the covariance matrix gives a means of quantifying this information (see, for example, [58], [90] and [92]). The eigenvalues indicate the lengths of the principal axes of the best-fit ellipsoid around the data, and the corresponding eigenvectors give the orientations within the coordinate system of the principal axes. Table 4.02 shows, for each of the clusters, the eigenvalues and corresponding eigenvectors for both the sample covariance matrix and the fuzzy covariance matrix. This demonstrates that the fuzzy covariance matrix can be used, to a good approximation, to determine the orientation of the cluster. This is convenient, because it is then not necessary to separately compute the sample covariance matrix in order to be able to recover this information.

In the context of the analysis of rock discontinuity orientation data, the ability to extract information regarding the orientations of the clusters in this way is extremely useful, because the relative magnitudes of the three eigenvalues give an indication of the cluster shape (compact, anisotropic, girdle-like) [44] and the corresponding eigenvectors indicate both the preferred orientation of the orientation vectors and the alignment of any anisotropy in the

Table 4.02 Cross data: eigenvalues and corresponding eigenvectors for covariance and fuzzy scatter matrices (after Bezdek [15]).

	sample covariance matrix		*fuzzy covariance matrix*	
	eigenvalue	eigenvector (*x*,*y*)	eigenvalue	eigenvector (*x*,*y*)
cluster 1	33.920	1.000,-0.001	36.351	1.000, 0.024
(horizontal)	0.070	0.001, 1.000	0.029	-0.024, 1.000
cluster 2	0.038	0.998,-0.061	0.040	1.000,-0.028
(vertical)	24.532	0.061, 0.998	25.670	0.028, 1.000

cluster [90, 92].

Although the FCV algorithm is capable of identifying elongate clusters through the use of cluster covariance matrices as norm inducing matrices in the computation of distance, it is possible to further enhance the clustering performance through the introduction of a membership function that attempts to identify crisper clusters. A suitable formulation for a membership function can be found in Bayesian statistics, and the incorporation of this into the FCV algorithm leads to the algorithm discussed in the next sub-Section.

4.4.4 The fuzzy maximum likelihood estimation algorithm

According to Gath and Geva [46], one of the paradoxes associated with fuzzy cluster identification is that humans tend to regard good clusters as those that are crisp, rather than fuzzy or ill-defined. In an attempt to address this paradox, these workers—in an extension to earlier work of Bezdek [15]—exploited the similarity between Bayesian classifiers and both the FCM and FCV algorithms to develop the **fuzzy maximum likelihood estimation** (FMLE) algorithm.

Formulation of the FMLE algorithm

Table 4.03 shows the functional similarity between the main relations of the FCV algorithm and **Bayesian classifiers**. As a description and critique of Bayesian classifiers is beyond the scope of this work, this Table is presented without proof.

On the strength of Table 4.03, it is apparent that with maximum likelihood estimation there are analogues to the fuzzy set formulations of cluster centroid, set membership, distance and covariance matrix. Indeed, superficially (and with the exception of distance and weight) the only difference between the formulations is one of nomenclature. In the case of weight, it is the assumption inherent in the development of Bayesian classifiers—namely, that events follow a normal distribution—that leads the corresponding function to incorporate an exponential term.

According to Gath and Geva, it is the inclusion of the weight function—defined by (4.36)—as a distance measure incorporating both exponentially increasing distances and a norm-inducing covariance matrix, which produces an algorithm that tends to seek:

(i) clear separation between clusters;
(ii) minimal volume of clusters; and

Table 4.03 Basic relations between elements of the FCV algorithm and Bayesian classifiers.

FCV	*Bayesian classifiers*
centroid: $$\mathbf{v}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{n} (u_{ij})^m}$$	mean vector: $$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)\mathbf{x}_j}{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)} \quad (4.34)$$
membership: $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}}$$	posterior probability: $$h(i \mid \mathbf{x}_j) = \frac{1}{\sum_{k=1}^{c} \left(\frac{w_{ij}}{w_{kj}} \right)} \quad (4.35)$$
distance: $$d_{ij} = \left[(\mathbf{x}_j - \mathbf{v}_i)^T \mathbf{A}_{FM} (\mathbf{x}_j - \mathbf{v}_i) \right]^{1/2}$$ where $$(\mathbf{A}_{FM})_i = [\rho_i \det(\mathbf{SF}_i)]^{(1/p)} (\mathbf{SF}_i)^{-1}$$ and $$\mathbf{SF}_i = \sum_{j=1}^{n} (u_{ij})^m (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T$$	weight: $$w_{ij} = \frac{[\det(\Sigma_i)]^{1/2} \exp\left[\frac{1}{2} (\mathbf{x}_j - \boldsymbol{\mu}_i)^T (\Sigma_i)^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_i) \right]}{p_i} \quad (4.36)$$
	prior probability: $$p_i = \frac{1}{n} \sum_{j=1}^{n} h(i \mid \mathbf{x}_j) \quad (4.37)$$
covariance matrix: $$\mathbf{F}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T}{\sum_{j=1}^{n} (u_{ij})^m}$$	covariance matrix: $$\Sigma_i = \frac{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)(\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)_{ij}} \quad (4.38)$$

(iii) maximum number of data points concentrated in the vicinity of the cluster centroid.

These attributes tend to produce well-defined clusters within the data, whilst still permitting uncertainty to be accounted for. Of course, we must be aware that the assumption of a normal distribution of events (or, in this context, set memberships) is implicit in the formulation of the FMLE algorithm, and this assumption may not always be correct.

The connection between FMLE and both FCM and FCV, indicated in Table 4.03, means that the FMLE algorithm is readily formulated, as follows:

(1) Initialise **U**, *i.e.* partition the data into c clusters using some initialisation procedure;
(2) Evaluate the cluster mean vectors $\boldsymbol{\mu}_i$ (i=1,c) using (4.34);
(3) Evaluate prior probabilities using (4.37);
(4) Calculate the fuzzy covariance matrices using (4.38), compute their norm-inducing inverses and their determinants;
(5) Compute posterior probabilities based on maximum likelihood estimation using (4.35), with weights given by (4.36) and the norm inducing matrices from step (4);
(6) Evaluate $\max_{ij}\left\|h_{ij}-\hat{h}_{ij}\right\|$, where $\hat{h}_{ij}$=corresponding value of h_{ij} from the previous iteration and $\|\cdot\|$ is a suitable matrix norm. If the magnitude of the norm exceeds a threshold value ε ($0<\varepsilon<1$), repeat from step (2).

Application of the FMLE algorithm

Gath and Geva tested their algorithm on synthesised data sets in $\mathbb{R}^2$ and $\mathbb{R}^5$, and then applied it to the analysis of human sleep patterns using electroencephalogram data in $\mathbb{R}^5$. In testing, the FMLE algorithm correctly partitioned the cross data of Gustafson and Kessel [54], presented above, indicating its ability to detect anisotropic clusters. A synthetic data set which consisted of two subgroups, one large and sparse, the other small and dense, was also used as a test. This data set is shown in the top diagram of Figure 4.07, where it can be seen that there is no visually apparent boundary between the subgroups, although their existence, and the fact that there are two such subgroups, is obvious. The manner in which these subgroups were synthesised is of interest: the customary technique of drawing random variables from two bivariate normal distributions, each possessing different means and standard deviations, was employed. As a consequence, the data in the subgroups are normally distributed.

(a) data

(b) FCM

(c) FMLE

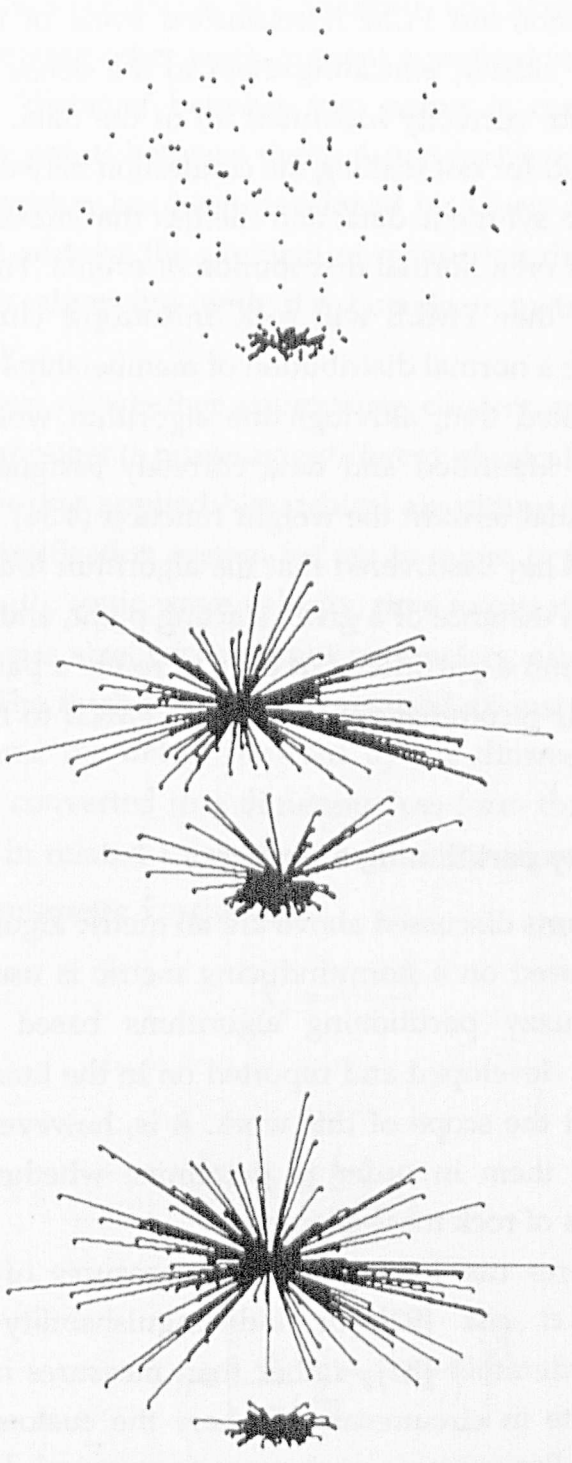


Figure 4.07 Data consisting of two ill-defined subgroups, partitioned using the FCM and FMLE algorithms (from Gath and Geva [46]).

The lower two diagrams in Figure 4.07 show, respectively, how the FCM and FMLE algorithms partitioned these data. Both algorithms correctly identified the two clusters, but FCM misclassified some of the data on the periphery of the sparse cluster, allocating them to the dense cluster instead. FMLE on the other hand, correctly identified all of the data. However, Gath and Geva can be criticised for not making the connection between the normally distributed nature of this synthetic data, and the fact that maximum likelihood estimation itself is based on a normal distribution of events. Thus, just as FCM seeks isotropic clusters, then FMLE will seek anisotropic clusters, but these clusters will tend to have a normal distribution of memberships within them.

Gath and Geva noted that, although the algorithm worked well, with clusters being correctly identified and data correctly assigned to them, the presence of the exponential term in the weight function (4.36) tended to make the algorithm unstable. They discovered that the algorithm found optima only within a small Euclidean distance of a given starting point, and concluded that it could not be used to find a partition, but only to refine a partition found by some other means. Their recommendation was for FMLE to be applied after pre-processing by FCM.

4.4.5 Non-metric fuzzy partitioning algorithms

The partitioning algorithms discussed above are all metric algorithms, in that a membership function based on a norm-inducing metric is used to determine membership values. Fuzzy partitioning algorithms based on non-metric methods have also been developed and reported on in the literature, but have been considered beyond the scope of this work. It is, however, interesting to briefly review some of them in order to determine whether they may be applicable to the analysis of rock mass geometry.

Non-metric measures can be regarded as measures of dissimilarity—according to Mardia *et alia* [92]—or indistinguishability—according to Demantaras and Valverderather [33]—rather than measures of distance, and are therefore appropriate in circumstances where the customary concept of distance is inapplicable. For example, rock mass geometrical data may contain not only the Euclidean components of orientation vectors, but also non-Euclidean components such as measures of discontinuity roughness and genesis. Under such circumstances, the implementation of a non-metric method may prove to be fruitful.

With non-metric methods the requirements for a metric, as presented in association with (4.07) above, do not hold, with the result that determining membership values is difficult, as Chaudhuri and Majunder [28] discuss. These authors, in reviewing other work, present a method based on rank ordering of data, in which 'distance' between two points is computed in terms of the number of other points between them. A non-metric algorithm similar in form to the FCM algorithm has been developed by Libert and Roubens [82]. These authors did not address the problem of measuring dissimilarity, but chose to implement their algorithm with the Euclidean metric and demonstrate its efficacy that way.

The question of whether appropriate clusters can be identified in data formed from components possessing different physical units was addressed by Lin [83]. This worker applied hierarchical algorithms to the development of a rock stability classification system for use in mines in China. This system used point load strength, sonic wave velocity, time to cessation of displacement and 'index of rock mass size' (*sic*) as input parameters, all of which have different physical units. The Euclidean metric was used exclusively in the work, and in order to overcome the defect imposed by the dimensions of the units, each component was converted to a dimensionless form through a process akin to normalisation. Lin related a parameter with dimensions, $\mathbf{x}_{ij}$, to a corresponding dimensionless parameter $\hat{\mathbf{x}}_{ij}$, through

$$\hat{\mathbf{x}}_{ij} = \frac{\mathbf{x}_{ij} - \overline{\mathbf{x}}_j}{S_j} \tag{4.39}$$

where

$$\overline{\mathbf{x}}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{ij} \tag{4.40}$$

and

$$S_j = \left[\frac{1}{n-1}\sum_{i=1}^{n}\left(\mathbf{x}_{ij} - \overline{\mathbf{x}}_j\right)^2\right]^{1/2} \tag{4.41}$$

No results of the analysis are given, other than to say that it performed well.

The application of non-metric methods, together with techniques for making parameters dimensionless, is untried in the field of rock mass geometry analysis. This is a promising avenue for further research.

4.4.6 Summary

This Section has discussed the uni-modal, fuzzy *c*-means (FCM), fuzzy covariance (FCV) and fuzzy maximum likelihood estimation (FMLE) algorithms for cluster identification, and briefly introduced non-metric methods.

This discussion has shown how the FCM algorithm, whilst computationally straightforward, will tend to identify isotropic (*i.e.* circular, spherical or hyperspherical) clusters through the use of the Euclidean metric for distance measurement. Such cluster shapes may be inappropriate for certain data. This defect is overcome in the FCV algorithm by the use of a Mahalanobis-like metric, in which the fuzzy scatter matrix is used as the norm-inducing matrix, with the result that anisotropic clusters may be identified. This improvement is gained at the expense of a large increase in the amount of computation required. However, a further advantage of the FCV algorithm is that it is possible to determine the orientation of the resulting clusters from the norm-inducing matrices themselves. Finally, the FMLE algorithm is able to produce improved partitions through the use of the maximum likelihood estimation weight function as a membership function. The resulting membership function thus contains an exponential term that optimises the partition through the rejection of data at large distances. However, the FMLE algorithm is reported to be unstable, is strictly only defined for maximum likelihood estimation, and will tend to find clusters that contain normally distributed data.

The general nature of the FCM, FCV and FMLE algorithms means that their application to the specific task of identifying clusters in rock mass geometrical data should prove successful. As such, these three algorithms have been selected for incorporation into a scheme for determining clusters in rock mass discontinuity orientation data. This scheme is described in detail in Chapter 6, and its application in conjunction with photogrammetric techniques of data collection is described in Chapter 7.

4.5 Validity of Fuzzy Clusters

It is important to be able to answer two related, but separate, questions regarding a partition of a data set: is the partition the optimum; and does a partition exist at all? Statistically, these questions would be answered using null-hypothesis testing, but for a non-statistical technique such as the

application of a fuzzy partitioning algorithm, these measures are not valid as the expected distribution of the data is not known.

In the event that a partitioning scheme using an objective function is used, whereby the function is either maximised or minimised at a given partition, then it would seem plausible to use the magnitude of the function itself as a measure of cluster validity. However, as Bezdek [15] notes, a number of workers have demonstrated that objective functions often display a global extremum that does not coincide with the optimal—in terms of human perception—partition. As an example, a function that seeks to maximise the density of data in a cluster will reject sparse, yet obvious, clusters. The result of this is that various heuristic scalar functions, based on well known statistical models, have been developed to give an indication of the validity of a fuzzy partition.

The heuristic nature of these functions means that large numbers of them—mostly based on a few fundamental models and subsequently adapted to the form of particular data—have been proposed in the literature. A useful review of the most commonly used examples, together with their properties where known, is given by Bezdek [15]. These functions are intended to either maximise, minimise or else display a large deviation from an overall trend at the optimal partition.

As far as the current Author has been able to determine, fuzzy partitioning algorithms with attendant cluster validity functions have never been applied to the problem of cluster identification in rock mass geometry. It is therefore useful to survey the more commonly used functions, together with some interesting variants, in the context of the analysis of rock mass geometry.

4.5.1 Cluster validity functions

Notwithstanding the heuristic nature of cluster validity functions, they are all based on some mathematical principle. As such, it is easy to suggest simple functions that could be considered: for example, a measure of density; or the magnitude of a resultant vector. However, such simple functions do not account for all of the attributes that make up a 'good' partition; for example, well defined, clearly separated and dense clusters.

To address this, functions have been developed which attempt to account for these attributes in different ways. These functions can be grouped into three categories: those which use the membership values of the data, those which consider the geometry of the resulting clusters, and those which make use of

parameters associated with a matrix such as the scatter matrix. Ten functions are described below, which fit into these three categories as follows:

(i) *membership values* partition coefficient, normalised partition coefficient, partition entropy, proportion exponent and modified partition coefficient;

(ii) *cluster geometry* separation coefficient and separation index;

(iii) *matrix parameters* fuzzy hypervolume, partition density and average partition density.

Partition coefficient

Evaluating the resultant vector is commonly used as a means of detecting good clusters in orientation data, as the texts by Fisher [44], Mardia [90], Mardia *et alia* [92] and Priest [108, 109] demonstrate. In the case of fuzzy partitioning, an analogue to this is found in the partition coefficient, and so this may be expected to perform well for such data. The partition coefficient is defined as

$$F = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{c} \left(u_{ij}\right)^2 \tag{4.42}$$

The term defined by the double summation sign can be seen to represent the square of the Euclidean distance of the resultant in $c \times n$ space, which accords with the notion of resultant.

One disadvantage of the partition coefficient is obvious: as c increases so the magnitude of the function may be expected to increase, because the number of terms involved in its evaluation increases. The limit is when $c=n$, in which case all membership values will equal unity, and $F=n$. Whilst such a case should not be taken as optimal, it does indicate that a hard partition (when the membership values are either zero or unity) will be preferred to a fuzzy partition in the case when low membership values predominate.

A further disadvantage is the lack of any direct connection to the data themselves, for example through the inclusion of the data matrix **X**.

Normalised partition coefficient

The deficiency present in the partition coefficient of (4.42), due to it being a function of c, may be overcome by normalisation thus:

$$B = \frac{cF - 1}{c - 1} \tag{4.43}$$

Of course, the singularity present at c=1 prevents this function being used to evaluate the quality of a partition consisting of a single cluster.

Interestingly, (4.43) may be rewritten—by making use of the upper bound of unity demonstrated by (4.42) when c=n—as

$$B = \frac{1}{c-1} \sum_{j=1+1}^{c} \sum_{i=1}^{c-1} \left[\frac{1}{n} \sum_{k=1}^{n} \left(u_{ik} - u_{jk} \right)^2 \right] \tag{4.44}$$

This demonstrates how B sums the inter-cluster variances, and is thus a measure of the overlapping present between clusters. It is this behaviour that gives rise to the alternative name of average partition separability.

It is interesting to note that (4.44) compares with the objective function used by Shanley and Mahtab [124, 125, 126] in their partitioning scheme for rock discontinuity orientation data, discussed in sub-Section 2.5.2. This function was based on the sum of intra-cluster and inter-cluster variances, and was reportedly used with some success. On the basis of this success, we may expect the comparable normalised partition coefficient to perform equally well with such data.

Notwithstanding the improvement (4.44) represents over (4.42) in terms of representing a function that is invariant in terms of the number of clusters, there is still no direct connection to the structure of the data, and the implicit use of the Euclidean norm (the square under the innermost summation sign reveals an inner product metric), will ensure that partitions consisting of (hyper)spherical clusters will be regarded as 'better' than those containing 'good' (hyper)ellipsoidal clusters.

Partition entropy

A well-known result in statistics is that a distribution, given by i probabilities p_i, has a statistical entropy of

$$H = -\sum_{i=1}^{n} p_i \ln p_i \tag{4.45}$$

It is easy to see that H=0 if any single probability is unity, and that entropy increases as the probabilities decrease. We thus see that entropy is inversely proportional to the likelihood (or certainty) of a given outcome. This concept can be extended to a measure of fuzzy partition quality: if cluster memberships are high (*i.e.* close to unity), then the entropy of the partition will be low. By including membership of all clusters, and normalising to account for the

number of data present, we arrive at the expression for partition entropy, namely

$$H = -\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}\log_a\left(u_{ij}\right) \tag{4.46}$$

which clearly produces a low value for good partitions. The base a of the logarithms may take any convenient positive value greater than one, and the expression within the summation signs is defined as being equal to zero when $u_{ij}=0$.

An advantage of (4.46) is that it is independent of any matrix norm, and therefore does not favour clusters of any particular shape. It does, however, demonstrate a monotonic increase with respect to c, but this can be removed by normalising with respect to c to give

$$\hat{H} = \frac{H}{1-(c/n)} \tag{4.47}$$

although this expression is insensitive to practical values of c and n (*i.e.* c likely to be less than 10, n likely to be in the hundreds) and offers no great advantage over (4.46).

Proportion exponent

Windham [142] tackled the cluster validity problem from a different standpoint. Rather than use all cn elements of the membership matrix, only the c largest elements are considered. The magnitude of these elements can be thought of as representing the quality of each cluster. The formulation of the function is then based on evaluating the proportion of membership values that exceed an expected amount, which in turn is a function of the number of data and the number of clusters. The proportion exponent is defined as

$$P = -\ln\left\{\prod_{k=1}^{n}\left[\sum_{j=1}^{\left[\mu_k^{-1}\right]}(-1)^{j+1}\binom{c}{j}\left(1-j\cdot\max_{1\le i\le c}\{u_{ik}\}\right)^{c-1}\right]\right\} \tag{4.48}$$

where

$$\left[\mu_k^{-1}\right] = \left(\text{greatest integer} \le \frac{1}{\max\limits_{1\le i\le c}\{u_{ik}\}}\right) \tag{4.49}$$

Once more, this function is independent of matrix norms, and is therefore not biased towards clusters of a particular shape. To its advantage, it is normalised

with respect to both the number of data and the number of clusters, but it is deficient in that it does not use any information implicit in the data themselves.

Separation coefficient

The alternative approach to formulating a cluster validity function is to examine the data themselves, on the basis of the results of the partitioning process. The heuristic involved is that a good partition will have well-defined clusters (which implies clearly separated clusters), with the elements in each cluster possessing high membership values.

On the assumption that these attributes can be determined from the geometry of the clusters, the fuzzy partition is in some way hardened (*i.e.* converted to a hard partition, in which cluster membership values of only either 0 or 1 are present), and geometrical parameters evaluated for the hard clusters: for example, number of elements, radius of the (hyper)spherical ball that surrounds the cluster, density in terms of elements per unit volume, and so on. In addition, some measure of the distance between adjacent clusters may be included.

The separation coefficient embodies these concepts, and is defined as

$$G = 1 - \max_{i+1 \le j \le c} \left\{ \max_{1 \le i \le c-1} \left\{ \frac{r_i + r_j}{c_{ij}} \right\} \right\} \tag{4.50}$$

where

$$r_i = \max_{1 \le k \le n} \left\{ u_{ik} d(\mathbf{X}_k, \mathbf{V}_i) \right\} \tag{4.51}$$

and

$$c_{ij} = d(\mathbf{V}_i, \mathbf{V}_j) \tag{4.52}$$

The separation coefficient is defined for a hard partition, and so the 'first moment of membership' indicated by (4.51) is, in fact, only a measure of distance. On this basis, the separation coefficient is defined entirely in terms of geometry, and this geometry is indicated in Figure 4.08.

In common with the functions described above, the separation coefficient makes use of Euclidean distances, with the result the (hyper)spherical clusters are preferred. It may be possible to use the Mahalanobis distance in (4.51), but as Figure 4.08 shows, because the separation coefficient is computed on the basis of distances determined along the line connecting the two cluster centroids, evaluating these under such circumstances would be extremely time

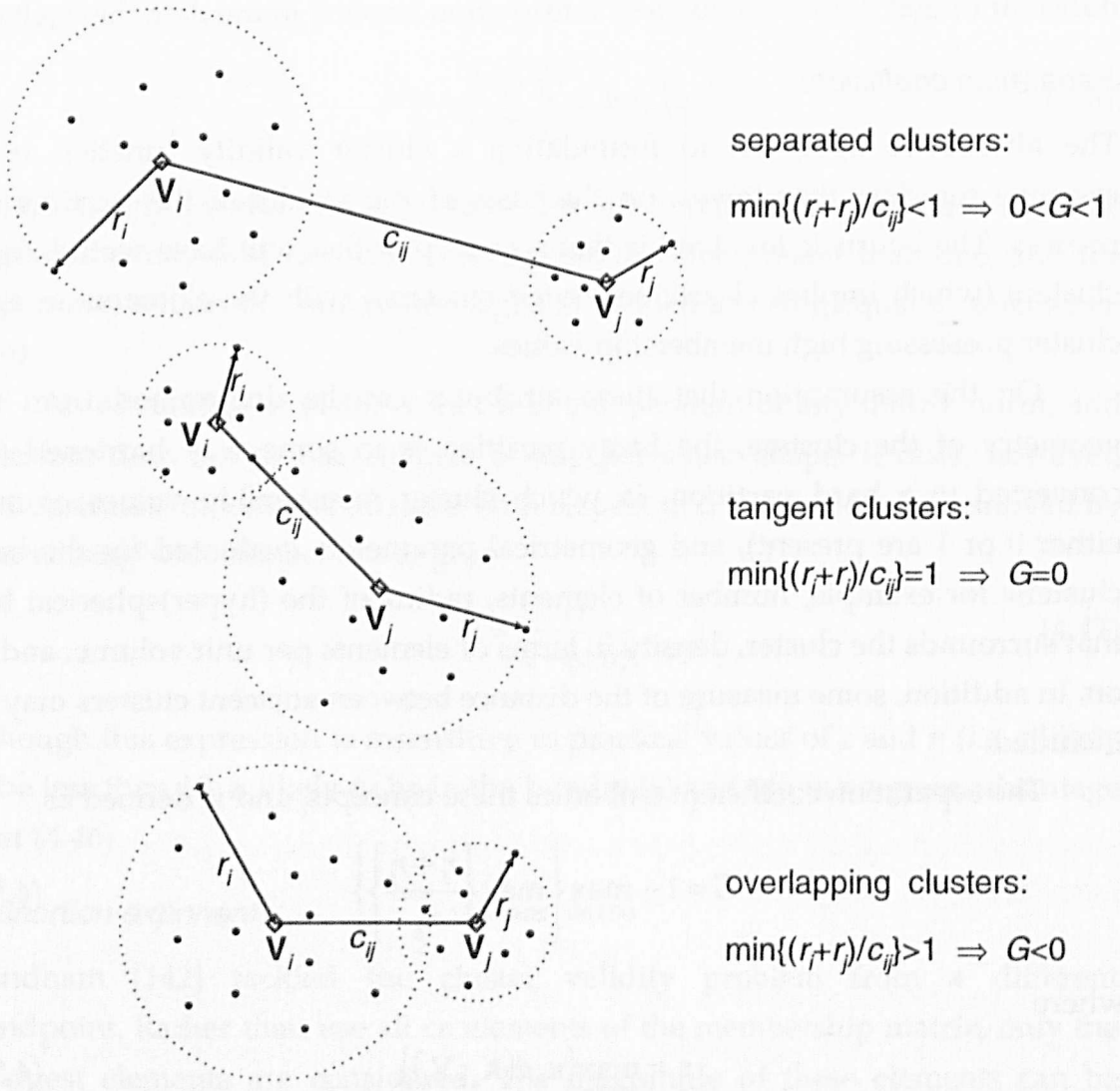


Figure 4.08 Geometry of the separation coefficient.

consuming. A further deficiency with the separation coefficient is that it is evaluated between all pairs of clusters, but only one pair is effectively chosen to represent the entire partition. Finally, we should note that there is no attempt to normalise the coefficient with respect to either the number of data or the number of clusters, with the result that the magnitude of the coefficient will be more likely to approach unity as the number of clusters increases.

It is interesting to consider how the coefficient would behave in the event that a fuzzy partition was used to evaluate (4.51). We can say that, as membership is inversely proportional to distance, neither large poorly defined clusters nor small well-defined clusters would be preferred to the other, and as a transition will occur from the former to the latter as the number of clusters

increases, normalisation will have been introduced. This point is considered further in Chapter 6.

Separation index

A cluster validity function that is very similar to the separation coefficient is the separation index. This is evaluated using a hardened partition, but does not invoke the locations of the cluster centroids. The separation index is defined as

$$D = \min_{i+1 \le j \le n} \left\{ \min_{1 \le i \le n-1} \left\{ \frac{d(u_i, u_j)}{\max_{1 \le k \le c} \{\text{dia}(u_k)\}} \right\} \right\} \tag{4.53}$$

where $d(u_i, u_j)$ and $\text{dia}(u_k)$ represent the distance between the two data vectors i and j, and the diameter of cluster k, respectively. Figure 4.09 demonstrates graphically how the separation index is evaluated, in terms of the largest cluster diameter and the minimum inter-cluster distance (measured between elements of the clusters).

Clearly, the same deficiencies exist with the separation index as with the separation coefficient: the use of Euclidean distances; only specific elements of the partition are used in its evaluation; and no attempt is made at normalisation.

Modified partition coefficient

According to Libert and Roubens [82], one of the disadvantages suffered by the partition coefficient is its inability to identify clearly separated clusters. These

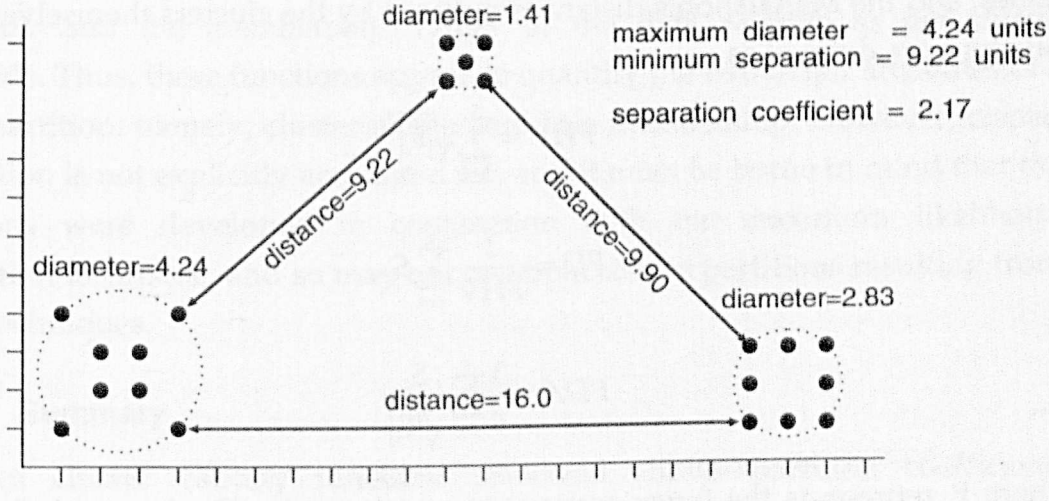


Figure 4.09 Geometry of the separation index.

workers proposed a modified form of the partition coefficient to include a term—the magnitude of the smallest maximum membership value—that is intended to account for overlapping clusters. Their modified partition coefficient is defined as

$$\tilde{F} = \frac{1}{2}\left[\frac{1}{n}\sum_{j=1}^{n}\max_{1\le i\le c}\{u_{ij}\} + \min_{1\le i\le c}\left\{\max_{1\le j\le n}\{u_{ij}\}\right\}\right] \tag{4.54}$$

and may be normalised with respect to c in the same fashion as the customary partition coefficient to give

$$\tilde{B} = \frac{c\tilde{F} - 1}{c - 1} \tag{4.55}$$

The rationale behind (4.54) is that the first term gives the intuitive 'average membership value', and the second term represents the maximum membership present in the most poorly defined cluster. This is intended to account for overlapping clusters, on the assumption that these are indicative of a poor overall partition. These two terms each have an upper bound of unity, and so by summing them and dividing by two an overall coefficient with an upper bound of unity is obtained.

Fuzzy hypervolume, partition density and average partition density

In their implementation of FMLE, Gath and Geva [46] proposed three cluster validity functions: fuzzy hypervolume (*FHV*); partition density, (*PD*); and average partition density, (*PDA*). These measures make use of all aspects of the partitioning process, namely the data and centroid vectors, the membership values, and the Mahalanobis distances induced by the clusters themselves. The functions are defined as

$$FHV = \sum_{i=1}^{c}\sqrt{|\mathbf{F}_i|} \tag{4.56}$$

$$PD = \frac{1}{FHV}\sum_{i=1}^{c} S_i \tag{4.57}$$

$$PDA = \frac{1}{c}\sum_{i=1}^{c}\frac{S_i}{\sqrt{|\mathbf{F}_i|}} \tag{4.58}$$

where $\mathbf{F}_i$ represents the fuzzy covariance matrix of the ith cluster, defined in (4.33) and S_i, the 'sum of the central members', is given by

$$S_i = \sum_{j=1}^{n} u_{ij} \tag{4.59}$$

In their application of (4.59), Gath and Geva suggested that only those u_{ij} for which

$$\left[\left(\mathbf{x}_j - \mathbf{v}_i \right)^T \mathbf{F}_i^{-1} \left(\mathbf{x}_j - \mathbf{v}_i \right) \right] < 1 \tag{4.60}$$

should be included in the evaluation. When used in the context of maximum likelihood estimation, as these authors intended, then (4.60) corresponds to using only those data that are within one standard deviation of the cluster centroid. This is an arbitrary threshold and, indeed, the constraint indicated by (4.60) is not one that must be enforced. It is, in fact, an interesting means of 'sharpening' the partition; this concept is dealt with in detail in the following Section.

Through the geometric interpretation of matrices we can see that *FHV* relates to the volume of the best-fit ellipsoids surrounding the clusters. According to Gath and Geva, *PDA* and *PD* reflect the presence of single dense clusters and the normal physical definition of density, respectively. In use, *FHV* tends to minimise, and both *PDA* and *PD* tend to maximise, at the optimal partition. In the event that the data contains clusters of differing fuzziness and shape, together with noise, then the use of these three measures in concert permits identification of the optimal partition of the data.

The three measures *FHV*, *PDA* and *PD* promise to be the most exacting of the cluster validity functions reviewed here. All of the functions are dependent on the fuzzy covariance matrices, which in turn are a function both of the geometry and the membership values of the data relative to the cluster centroids. Thus, these functions appear to quantify the two major attributes of a good partition: namely, cluster shape and data membership. However, cluster separation is not explicitly accounted for, and it must be borne in mind that the functions were developed in conjunction with the maximum likelihood estimation technique, and so may not be applicable to partitions resulting from other techniques.

4.5.2 Summary

The ten cluster validity functions reviewed above—partition coefficient, normalised partition coefficient, partition entropy, proportion exponent, modified partition coefficient, separation coefficient, separation index, fuzzy

hypervolume, partition density and average partition density—attempt to define a scalar value that is representative of the 'quality' of the clusters within a partitioned data set. They do this by computing a statistic based on either the magnitudes of some or all of the elements of the membership matrix, the sizes and separation of the proposed clusters, or the mathematical attributes of the cluster scatter matrices.

There are advantages and disadvantages associated with each of these functions. Those which make use only of the membership matrix can be regarded as ignoring the geometry of the clusters, and *vice versa*. The functions that use the cluster scatter matrices—fuzzy hypervolume, average partition density and partition density—appear to suffer least from disadvantages associated with ignoring information regarding either membership value or cluster geometry, but they strictly are defined only for maximum likelihood estimation.

It is apparent, therefore, that no one cluster validity function is available that will quantify all aspects of a partition. Furthermore, we may expect these various functions to be biased towards identifying different aspects of a partition as optimal. Pragmatism therefore leads us to the conclusion that, in order to identify the optimal partition, we should evaluate a number of these functions and accept as the optimal partition the one that optimises functions based on membership values, cluster shape and scatter matrix attributes.

4.6 Hardening of Fuzzy Partitions

Fuzzy partitioning schemes present an interesting paradox: although it seems natural to analyse data on the basis of concurrent membership of multiple sets, the results are most easily understood in terms of membership of a single set. It is easy to imagine the case where an analyst is presenting results arising from a fuzzy partitioning scheme, only to be asked 'but which single set is the datum a member of'. Thus, although data may be analysed on the basis of fuzzy sets, the results are often required to be presented in terms of hard sets. Indeed, this is apparent in the functions reviewed in the previous Section, in that the search for maximum memberships—which amounts to determining the 'best' cluster membership—is commonly undertaken. Two techniques are used to harden a fuzzy partition, and these are discussed in the following sub-Sections.

4.6.1 'Nearest maximum membership' hard partition

Formation of the **nearest maximum membership hard partition** is intuitive. For a given datum, the membership values in each cluster are considered and the maximum identified. This is then set equal to unity, and the other memberships set to zero. Formally, the procedure is written as

$$\left(\mathbf{U}_{MM}\right)_{ij} = \begin{Bmatrix} 1, & u_{ij} = \max\limits_{1 \le k \le c}\{u_{kj}\} \\ 0, & \text{otherwise} \end{Bmatrix} \tag{4.61}$$

The result of applying this procedure is that the data are assigned to only one cluster.

Although (4.61) simplifies the resulting partition, the problem of symmetrical data reappears. In the event that, to the limit of machine precision, a datum possesses equal maximum membership in more than one cluster, then the use of (4.61) must be accompanied by a process in which a datum is arbitrarily assigned to one of the clusters in which it exhibits maximum membership. In general, this will not be necessary, as equal membership to the limit of machine precision will occur only in the most pathological of cases—for example, the butterfly data. However, if this case does occur, or if the maximum membership for a particular datum is *nearly* equal for one or more clusters, then a graphical presentation of the resulting hard partition may result in cluster boundaries being drawn which are inconsistent with the cluster centroids. Figure 4.10 demonstrates this with the butterfly data, where the final memberships have been taken from Figure 4.05. The central datum must be assigned to one of the two clusters, but the resulting cluster boundaries are clearly inconsistent with the symmetrical cluster centroids.

A detail not demonstrated by this data, but which may be of importance in data that exhibits dense regions (corresponding to the location of the cluster centroids) surrounded by sparsely populated regions, is the abrupt transition from one cluster to another. We would naturally regard data in sparsely populated regions as not being members of any cluster, and to see such data assigned may seem counter-intuitive. To overcome this, a generalised form of producing a hard partition is used, and this is discussed in the next sub-Section.

Finally, it must be stressed that analysing data using FCM and hardening the results is not equivalent to analysing with HCM, as a comparison between Figures 4.03 and 4.05 shows—the locations of the cluster centroids are quite different.

Final membership matrix with m=2.00:

element	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
cluster 1	0.86	0.97	0.86	0.94	0.99	0.94	0.88	0.50	0.12	0.06	0.01	0.06	0.14	0.03	0.14
cluster 2	0.14	0.03	0.14	0.06	0.01	0.06	0.12	0.50	0.88	0.94	0.99	0.94	0.86	0.97	0.86

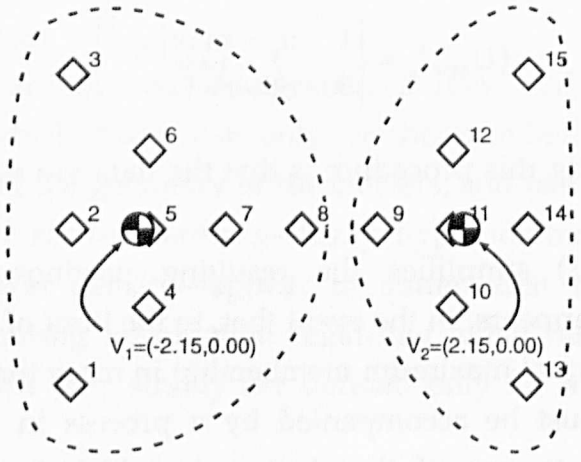


Figure 4.10 Butterfly data: nearest maximum membership hard partition.

4.6.2 α-Level core of a fuzzy set

It is quite logical to continue with the concept of uncertainty in set membership whilst producing a hardened form of a fuzzy partition. For example, we may wish to see set boundaries drawn around only those data for which our level of certainty regarding membership exceeds a given threshold. This is analogous to identifying those data for which the membership value exceeds a certain value. The name **α-level core** is suggested for the set of data for which this is true, as such a set will form a core to a set defined as the nearest maximum membership partition. Bezdek [15] defines the α-level core as

$$C_\alpha(u_i,\alpha)=\{x\in X \mid u_i(x)>\alpha\} \tag{4.62}$$

which states that any element x will be a member of the α-level core C_α if its membership in the ith cluster exceeds the threshold value α. Obviously, if ambiguity in cluster assignment is to be avoided, then the lower bound on the threshold of $\alpha \not< 1/c$ must be observed. For example, in the case of two clusters, then the threshold must exceed 0.5, and in the case of four clusters it must exceed 0.25.

The results presented in Figure 4.10 may be recast in this form, as shown in Figure 4.11. This reveals two useful aspects of α-level hardening: the boundary of a cluster is always consistent with the location its centroid; and the size of a cluster reduces as the level of certainty indicated by it increases.

This latter point is very important, for if we were to superimpose the three diagrams shown in Figure 4.11, a diagram that resembles a contour plot would be obtained. Notwithstanding the fact that the curves surrounding each α-level core are placed arbitrarily rather than interpolated, they still indicate degree of certainty, which is an analytical property of the data.

The ability to produce such plots is extremely useful, especially when considered in the context of the analysis of rock discontinuity orientation data. A successful implementation of a fuzzy partitioning scheme for such data will result in membership values being assigned to individual orientation vectors. The boundaries of the resulting α-level cores can then be plotted to allow an assessment of cluster assignment to be made using a diagram similar in form to

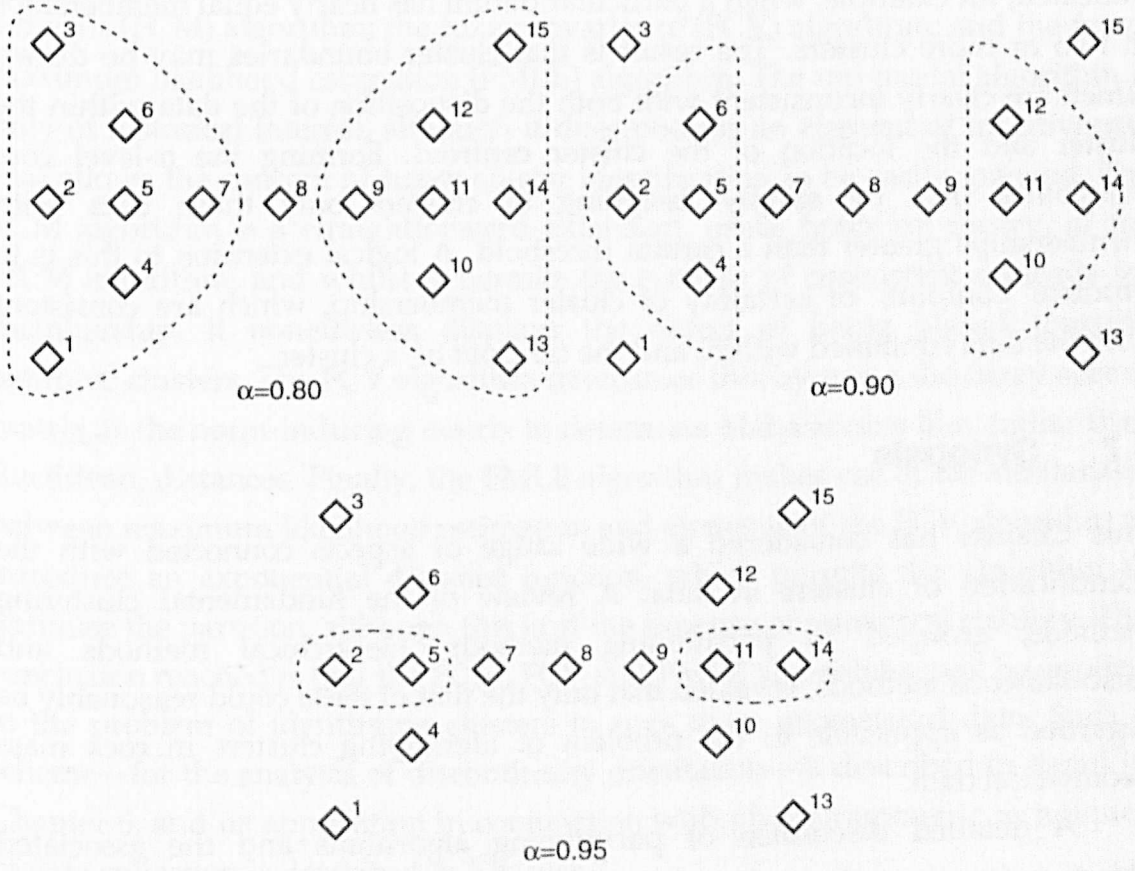


Figure 4.11 α-Level hard cores associated with the butterfly data.

the customary contoured hemispherical projection. It must, however, be stressed that the resemblance is entirely cosmetic—the review presented in Chapter 2 demonstrated that contouring performed in the customary fashion reveals arbitrary and extrinsic properties, whereas α-level cores reveal an intrinsic property regarding cluster membership. The use of α-level plots in this way is novel, and the technique is applied to orientation data in Chapters 6 and 7.

4.6.3 Summary

Hardening of fuzzy partitions gives us a means of addressing the paradox that, whilst it appears natural to analyse data on the basis of concurrent membership of multiple sets, the results are most easily assimilated on the basis of single set membership.

Both the nearest maximum membership partition and the α-level core of a fuzzy set allow the partitioning results to be simplified in a way that assists human interpretation. Nearest maximum membership hardening suffers from the defect of assigning data to clusters when such an assignment is not indicated; for example, when a particular datum has nearly equal memberships in two or more clusters. The result is that cluster boundaries may be drawn which are clearly inconsistent with both the disposition of the data within the cluster and the location of the cluster centroid. Forming the α-level core overcomes this, by simply assigning to clusters only those data with memberships greater than a certain threshold. A logical extension to this is to produce 'contours' of certainty of cluster membership, which are consistent with the data contained within, and the centroid of, a cluster.

4.7 Synopsis

This Chapter has considered a wide range of aspects connected with the identification of clusters in data. A review of the fundamental clustering methods, grouped as partitioning methods, hierarchical methods and miscellaneous methods, revealed that only the first of these could reasonably be regarded as applicable to the problem of identifying clusters in rock mass geometrical data.

A detailed discussion of partitioning algorithms and the associated objective functions showed how the implicit use of the Euclidean metric for distance measurement led to the identification of isotropic clusters, even when

none were indicated. The ramifications for the analysis of rock mass geometry were highlighted: the fact that as we have no *a priori* knowledge of the shape of clusters we should find in discontinuity orientation data, an algorithm that is not biased towards predetermined cluster shapes should be adopted. The hard *c*-means (HCM) algorithm was presented, together with examples of how it partitioned data. The major defect apparent in this algorithm is its inability to consider data as concurrently being partly contained within a number of clusters, with the result that inappropriate clusters may be identified.

Identification of this defect leads naturally to a discussion of uncertainty and fuzzy sets. Examples of deterministic, probabilistic and fuzzy models in rock mechanics indicate clearly that it is the latter that are appropriate for the identification of clusters in rock mass geometrical data. Fuzzy models allow quantification of uncertainty in set membership, whereas probabilistic models allow the probability of occurrence to be assessed; these two concepts are clearly different.

A review of the application of fuzzy set theory to fuzzy partitioning algorithms has been presented in Section 4.4. Four fuzzy algorithms for cluster identification were discussed in detail: the uni-modal algorithm; the fuzzy *c*-means (FCM) algorithm; the fuzzy covariance (FCV) algorithm; and the fuzzy maximum likelihood estimation (FMLE) algorithm. The uni-modal algorithm is only of historical interest, although it does possess an element of intuitiveness that allows the concept of fuzzy cluster identification to be easily grasped. The FCM algorithm is a straightforward extension, using fuzzy set theory, of the HCM algorithm, and whilst it permits the concept of concurrent multiple set membership, it nonetheless displays the defect of being biased towards isotropic clusters. The FCV algorithm overcomes this by using the fuzzy scatter matrix as the norm-inducing matrix to determine Mahalanobis-like, rather than Euclidean, distances. Finally, the FMLE algorithm makes use of the similarities between maximum likelihood estimation and elements of the FCV algorithm to introduce an exponential distance function, which permits the algorithm to optimise the partition, although this is at the expense of numerical stability. The conclusion reached is that the FCM, FCV and FMLE algorithms may be applied to the problem of identifying clusters in rock mass geometrical data. Such a scheme—for the analysis of discontinuity orientation—is described in detail in Chapter 6, and its application in conjunction with photogrammetric techniques of data collection is described in Chapter 7.

The important question of assessing whether a particular partition is the optimum, or whether a partition exists at all, is addressed through a review of ten cluster validity functions. The functions reviewed—partition coefficient, normalised partition coefficient, partition entropy, proportion exponent, modified partition coefficient, separation coefficient, separation index, fuzzy hypervolume, partition density and average partition density—may be grouped in terms of three different geometrical bases. The result is that a validity statistic, based on either the magnitudes of some or all of the elements of the membership matrix, the sizes and separation of the proposed clusters, or the mathematical attributes of the cluster scatter matrices, is evaluated. Various advantages and disadvantages associated with the mathematical bases, and hence all of the functions, indicate that no one function will quantify all aspects of a partition, and so optimality will only be identified through the application and assessment of a number of functions.

Finally, the concept of converting a fuzzy partition into one that is hard has been addressed in Section 4.6. Hard partitions are much easier for humans to assimilate than fuzzy partitions, and on this basis two hardening techniques have been reviewed: determining the nearest maximum membership partition; and evaluating the α-level core of a fuzzy set.

Determining the nearest maximum membership partition involves identifying to which cluster a datum has greatest membership, and then assuming that cluster is the only one to which it belongs. A generalised form of this hardening is the α-level core of a fuzzy set, which may be thought of as being equivalent to determining an uncertainty threshold. As a consequence, it is possible to produce a 'contour' diagram of the partition, and it is suggested by the Author that this can be applied in a novel way to the analysis of rock discontinuity orientations. This application of α-level plots is original, and is presented in Chapters 6 and 7.

Part III

Development of an Integrated System for Surveying and Analysis of Rock Mass Geometry

This page intentionally left blank

5 Non-Metric Photogrammetry Applied to Measurements of Rock Mass Geometry

The overview of photogrammetry presented in Chapter 3 indicated how traditional metric analogue photogrammetry has been successfully applied to problems in rock mechanics, and how new procedures associated with non-metric techniques are well established in the field of photogrammetry at large. The purpose of this Chapter is to show how, with appropriate developments, these two aspects of photogrammetry can be combined to form a non-metric photogrammetric system for determination of rock mass geometry.

In order to develop a useful system for the non-metric photogrammetric surveying of rock mass geometry, a number of questions regarding the specification of the equipment and techniques must be answered. The most important of these is, to what accuracy must the photograph coordinates of a discontinuity be measured in order to determine its location and orientation in space with acceptably low errors? In order to answer this, solutions to two supplementary questions are required: what photogrammetric configuration will produce photography of the required standard; and, what resolution is required from the instrument that measures the photograph coordinates? Related to these questions in turn is, how can photogrammetric control, in terms of points with known object space coordinates, be introduced to the system?

The first four Sections of this Chapter consider and answer each of these questions in turn. In Section 5.1, an analysis is presented of discontinuity orientation as computed from measured photograph coordinates, and this permits conclusions to be drawn regarding both the photogrammetric configuration adopted and the resolution of the analytical plotter to be used for the measurement of these coordinates. Section 5.2 goes on to present the design and construction (both aspects being the work of the current Author) of a novel high-performance, low-cost analytical plotter of appropriate specification. A method of introducing photogrammetric control, together with recommendations for the associated survey network adjustment computations,

is presented in Section 5.3, and enhancements to the DLT method of photogrammetric data reduction are outlined in Section 5.4. An accuracy test of a complete photogrammetric system that incorporates all of these features is described in Section 5.5, from which it can be concluded that non-metric analytical photogrammetry is appropriate for the measurement of rock mass geometry.

Finally, the Chapter closes with a synopsis at Section 5.6.

5.1 An Analysis Of The Computation Of Discontinuity Orientation From Measured Photograph Coordinates

Although a great deal of error analysis has been performed for terrestrial photogrammetry in general, this has been limited to simple spatial error (see, for example, [7], [18] and [69]). This work, however, is also concerned with photogrammetric measurement of discontinuity orientation, and as far as this Author has been able to determine, no error analysis exists for this.

The development of an orientation-based analysis can use one of two methods: an analytical approach, in which a mathematical model of the problem is derived, and errors determined through the evaluation of derivatives with respect to the various input parameters; and a numerical approach, in which a numerical model of the problem is constructed, and overall errors determined through the direct introduction of errors in the measured parameters. These two methods are considered in the following two sub-Sections.

5.1.1 Analysis using total differential error

The accepted method of error analysis in photogrammetry is to compute the **total differential error** of a system. In this technique, all of the variables that may contain random errors are identified, and the partial derivatives of the **functional model** (*i.e.* the relation between the element to be evaluated and those that are measured) with respect to each variable are evaluated and combined. A simple example, taken from Wolf [143], illustrates the technique.

Example application in vertical aerial photography

In the case of vertical aerial photography, the functional model for the height of a point above a datum is one of the parallax equations, *i.e.*

$$Z_a = H - \frac{Bf}{p_{x_a}} \tag{5.01}$$

where Z_a = height of point *a* above datum,
H = flying height of aircraft above datum,
B = distance along flight line between photographs,
f = principal distance of camera, and
p_{x_a} = parallax (difference between the *x*-coordinate values measured in the two photographs) of point *a*.

If, for example, random errors exist in the determination of H, B and p_{x_a}, then the partial derivatives $\frac{\partial Z_a}{\partial H}$, $\frac{\partial Z_a}{\partial B}$ and $\frac{\partial Z_a}{\partial p_{x_a}}$ are formed, *i.e.*

$$\frac{\partial Z_a}{\partial H} = 1, \quad \frac{\partial Z_a}{\partial B} = \frac{f}{p_{x_a}} \quad \text{and} \quad \frac{\partial Z_a}{\partial p_{x_a}} = \frac{Bf}{p_{x_a}^2} \tag{5.02}$$

and combined in root-mean-square fashion to determine their total effect on the functional model. Thus, we have

$$\delta Z_a = \sqrt{(1 \times \partial H)^2 + \left(\frac{f}{p_{x_a}} \times \partial B\right)^2 + \left(\frac{Bf}{p_{x_a}^2} \times \partial p_{x_a}\right)^2} \tag{5.03}$$

It is then a case of assessing the errors ∂H, ∂B and ∂p_{x_a} present in the measurement of H, B and p_{x_a}, and computing the magnitude of δZ_a.

Application to discontinuity orientation

The application of this technique to the case of analysing error in discontinuity orientation is now demonstrated. If we define error in orientation as the angle subtended by two vectors, one representing the true normal to a discontinuity surface and the other the normal to a displaced (*i.e.* erroneously measured) surface, then the error is computed as the scalar product of the vectors *tn* (true normal) and *dn* (displaced normal), thus

$$\theta_{error} = \cos^{-1}(l_{tn}l_{dn} + m_{tn}m_{dn} + n_{tn}n_{dn}) \tag{5.04}$$

Where l, m and n are the direction cosines of the vectors. Consequently, the partial derivatives with respect to the direction cosines of the displaced normal are required. However, in a photogrammetric application, these direction cosines will themselves be computed from the space coordinates of points

known to define the discontinuity surface, and these space coordinates will, in turn, be computed from measurements of the corresponding photograph coordinates. Thus, taking the simplest case of three points defining a planar discontinuity surface, the functional model that relates angular error to error in photograph measurement may be constructed as outlined in the following paragraphs.

Firstly, the space coordinates of each of the three points on the discontinuity surface are computed, using the DLT parameters, L, and the x- and y-coordinate values from both the left and right photographs, as the least squares estimate

$$\begin{bmatrix} (L_1^l - L_9^l x^l) & (L_2^l - L_{10}^l x^l) & (L_3^l - L_{11}^l x^l) \\ (L_5^l - L_9^l y^l) & (L_6^l - L_{10}^l y^l) & (L_7^l - L_{11}^l y^l) \\ (L_1^r - L_9^r x^r) & (L_2^r - L_{10}^r x^r) & (L_3^r - L_{11}^r x^r) \\ (L_5^r - L_9^r y^r) & (L_6^r - L_{10}^r y^r) & (L_7^r - L_{11}^r y^r) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -(L_4^l - x^l) \\ -(L_8^l - y^l) \\ -(L_4^r - x^r) \\ -(L_8^r - y^r) \end{bmatrix} \tag{5.05}$$

where the superscripts l and r refer to left and right photographs, respectively.

Secondly, the space coordinates of these three points, i, j and k, are formed into the determinant

$$\begin{vmatrix} x - x_i & y - y_i & z - z_i \\ x_j - x_i & y_j - y_i & z_j - z_i \\ x_j - x_i & y_j - y_i & z_j - z_i \end{vmatrix} = 0 \tag{5.06}$$

which, upon expansion, yields the coefficients a, b, c and d of the plane

$$ax + by + cz + d = 0 \tag{5.07}$$

which passes through the three points. The direction cosines of the normal to the plane are then computed from these coefficients as

$$l = \frac{a}{p}, \quad m = \frac{b}{p} \quad \text{and} \quad n = \frac{c}{p} \quad \text{where} \quad p = \sqrt{a^2 + b^2 + c^2} \tag{5.08}$$

Thirdly, the angular error may be computed from these parameters using (5.04).

Now, each of the photograph coordinates in (5.05) are subject to measurement error, arising primarily from random errors in the electromechanical photograph positioning system, and errors of interpretation due to the human observer. Thus, each of equations (5.04), (5.05), (5.06), (5.07)

and (5.08) must be expanded in order to determine the partial derivatives $\frac{\partial\theta}{\partial x_i^l}$, $\frac{\partial\theta}{\partial y_i^l}$, $\frac{\partial\theta}{\partial x_i^r}$, $\frac{\partial\theta}{\partial y_i^r}$, $\frac{\partial\theta}{\partial x_j^l}$, $\frac{\partial\theta}{\partial y_j^l}$, $\frac{\partial\theta}{\partial x_j^r}$, $\frac{\partial\theta}{\partial y_j^r}$, $\frac{\partial\theta}{\partial x_k^l}$, $\frac{\partial\theta}{\partial y_k^l}$, $\frac{\partial\theta}{\partial x_k^r}$ and $\frac{\partial\theta}{\partial y_k^r}$ and hence their root-mean-square combined effect on θ_{error}.

Clearly, derivation of the total differential error for orientation determination will be a lengthy process, and one that cannot be guaranteed to result in an expression that will offer a concise insight into the problem. Indeed, any resulting expression is likely to require numerical evaluation simply for the purposes of visualisation. Consequently, the problem has been tackled in this work pragmatically, from a purely numerical point of view.

5.1.2 Numerical analysis

A numerical approach to the analysis of error in determination of discontinuity orientation has been developed by the Author as part of this work. The approach followed seeks to represent the actual procedures adopted in photogrammetric determination of discontinuity orientation.

As orientation is computed from the coordinates of points known to lie on a discontinuity surface, and the space coordinates of such points are determined from photograph coordinates, the analysis firstly models the photographic process, then the coordinate measurement process and finally computation of orientation. The model attempts to account for the effects of discontinuity size, orientation and location relative to the cameras, and the errors present in photograph coordinate measurement. The model is deterministic, not probabilistic, and therefore gives worst-case, rather than most likely, estimates of error.

The discussion of this analysis, presented in the four Divisions below, is arranged as: an outline of the analysis, including the various simplifying assumptions adopted in order to make the problem tractable; the computational procedure adopted; the application of the method to a representative photogrammetric configuration, together with results; and, in conclusion, an empirical predictive equation.

Outline of the analysis

In order to concentrate solely on the problem of orientation analysis, a simplified representation of a discontinuity surface has been adopted. A triad

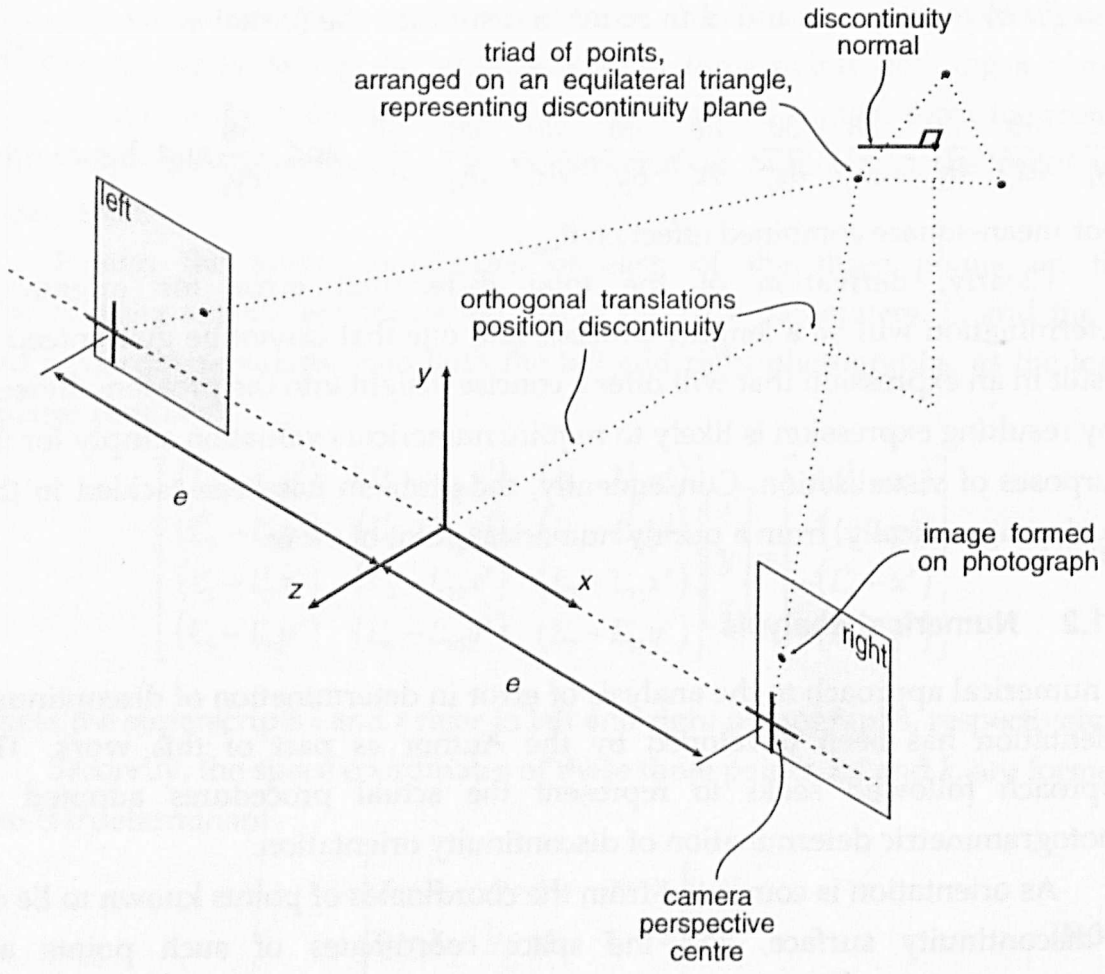


Figure 5.01 Configuration of a simplified representation of a discontinuity surface for the purposes of analysis of error in orientation.

of points, arranged to fall on the vertices of an equilateral triangle, has been taken to define the discontinuity surface. Using three points removes the added complication of lack of coplanarity of the points (*i.e.* roughness of the surface), and a triad in the form of an equilateral triangle allows the size (*i.e.* two-dimensional extent) of the discontinuity surface to be defined as the radius of the circumscribed circle, with the centre of this circle being able to represent the location of the discontinuity surface in space. The location of the discontinuity centre relative to the photogrammetric configuration is defined in terms of its Cartesian coordinates. Figure 5.01 shows this representation of a discontinuity relative to the photogrammetric configuration and global Cartesian axes.

The local orientation of a triad (*i.e.* orientation relative to the discontinuity surface) is defined in terms of a rotation about the local z'-axis, which is

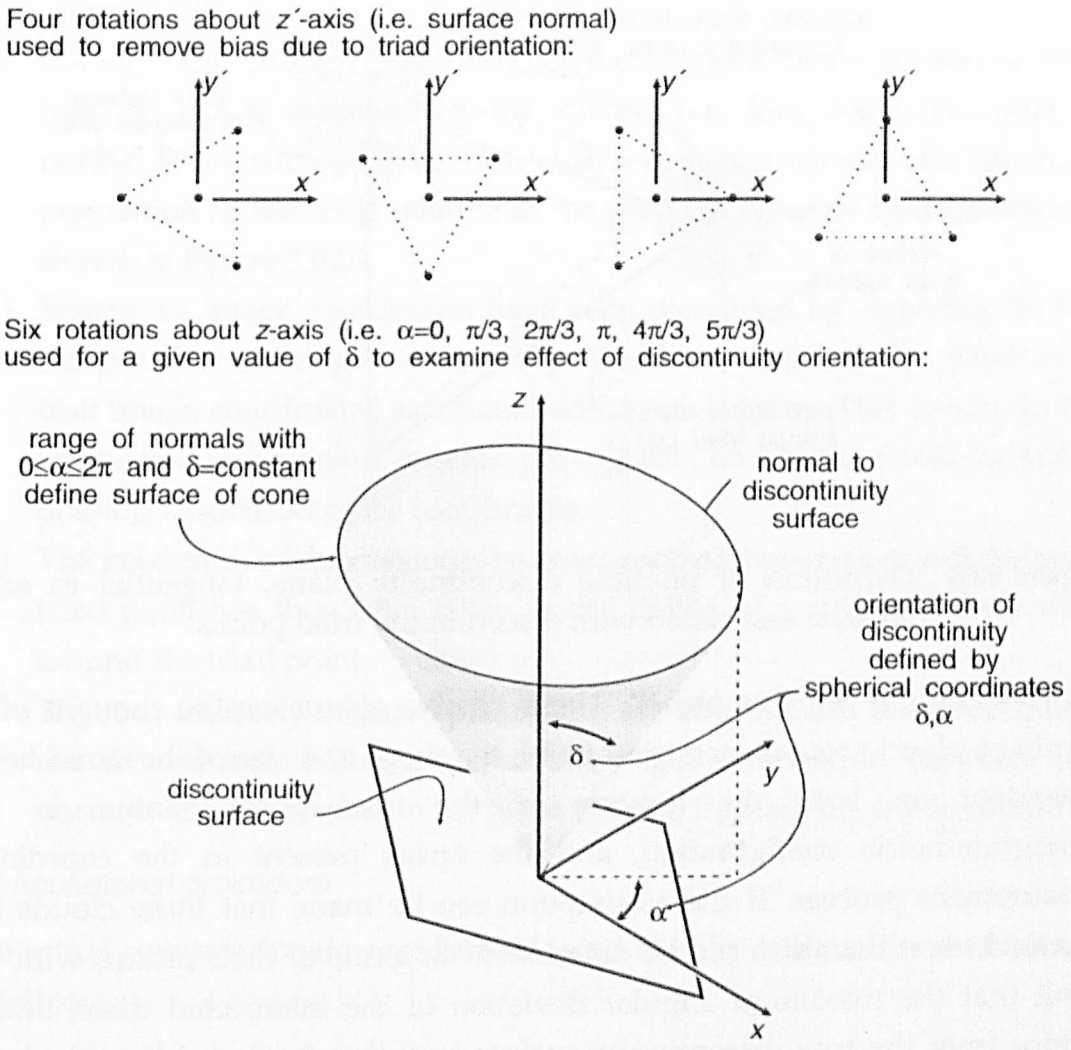


Figure 5.02 Definitions of triad and discontinuity orientations.

parallel to the discontinuity normal and passes through the centre of the triad. Spherical coordinates define the orientation of the discontinuity surface, relative to the photogrammetric configuration, in terms of δ, the subtended angle between the normal to the surface and the global z-axis, and α, the azimuth of the normal (measured in the global x-y plane, counter-clockwise positive from the x-axis), as shown in Figure 5.02. These parameters are sufficient to compute the space coordinates of the triad points and (assuming the use of perfect cameras) the image coordinates, on both the left and the right photographs, of each point in the triad.

These image coordinates are then varied to mimic measurement error, and the resulting erroneous coordinates used to determine the space coordinates of

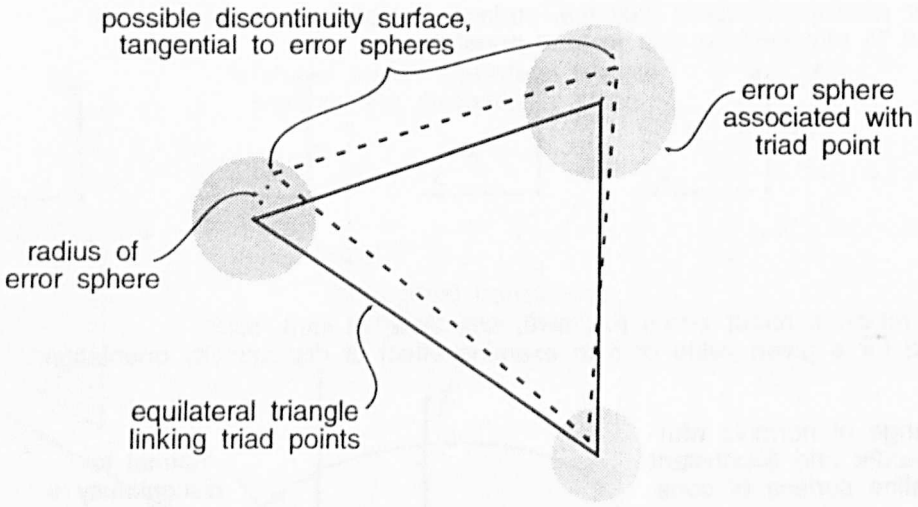


Figure 5.03 Definition of possible discontinuity plane, tangential to error spheres associated with discontinuity triad points.

a corresponding 'illusive object'. These illusive objects can be thought of as forming a cloud around each triad point, the shape and size of the cloud being dependent on both the location of the triad point relative to the photogrammetric configuration, and the errors present in the coordinate measurement process. If the assumption can be made that these clouds are spherical, then their size can be determined in terms of their radius, with the result that the maximum angular deviation of the interpreted discontinuity surface from the true discontinuity surface may then be found by examining the orientations of those planes that are tangential to the error spheres, as shown in Figure 5.03. Although such planes may be tangential either above or below the plane containing the triad, it is easily demonstrated that only four unique angular deviations exist. These may be found by analysing the cases $(+i,+j,+k)$, $(-i,+j,+k)$, $(-i,-j,+k)$ and $(+i,-j,+k)$ for the points i, j and k, where the plus and minus signs indicate above and below the true plane, respectively.

It is clearly impossible to analyse the error clouds around the infinite number of point triads that represent discontinuity surfaces of all sizes and orientations, and so a number of limitations and assumptions have been introduced to render the analysis computationally tractable. These are described in the following points.

(i) Discontinuity locations have been limited to a coarse grid and, for a given discontinuity size, the analysis has been arbitrarily limited to six values of

α, for each of a number of specifically chosen values of δ (as shown in Figure 5.02).

(ii) For all discontinuity orientations considered, four local orientations of the triad of points representing the surface (*i.e.* four rotations about the normal to the surface) have been analysed, to remove any bias due to the orientation of the triad relative to the photogrammetric configuration (as shown in Figure 5.02).

(iii) Erroneous image coordinates have been computed by applying, in both the positive and negative x- and y-directions, a coordinate offset to the true image coordinates associated with a triad point. This results in four erroneous image points on each photograph, and hence a total of sixteen possible associated space coordinates.

(iv) The maximum of the distances between each of these points and the actual triad point has then been taken as the radius of a spherical error cloud around the triad point.

(v) The four uniquely oriented planes that are tangential to the error spheres are determined, and the maximum angular deviation between their normals and the normal to the plane passing through the triads computed.

Computational procedure

The actual computational procedure adopted may be outlined in **pseudocode**[†] as follows:

[†] Pseudocode is essentially a plain-English representation of the procedures followed by a computer program. Although the standard symbols for arithmetic operations (i.e. +, -, * and /) and parentheses to indicate precedence of operations are used, the syntax of specific programming languages with respect to, for example, controlling program flow, input/output and variable type and value declaration are not. In addition, a level of detail is adopted which is appropriate to the discussion in hand, and the code is indented to reveal loop structures. The result is a program representation that is easy to follow and assimilate.

```
Set discontinuity size, subtended angle between global z-axis and
local (to the discontinuity) z'-axis, and error associated with
measurement of image coordinates
For all required discontinuity locations:
  For all representative orientations of z'-axis relative to global
  axes:
    For all representative orientations of triad of discontinuity
    points relative to local axes:
      Compute image locations in left and right photographs of triad
      of discontinuity points
      For each image point:
        For all combinations of error in image position:
          Compute space coordinates of point defined by erroneous
          image coordinates
          Compute distance between true and erroneous space
          coordinates
        Next image error
        Determine maximum of error distances, assign to radius of
        error sphere
      Next image point
      For all planes tangential to error spheres:
        Determine subtended angle between normal to tangential plane
        and plane passing through the triad of discontinuity points
      Next tangential plane
      Determine maximum of subtended angles for all tangential planes
    Next triad orientation
    Determine maximum of subtended angles computed for all triad
    orientations
  Next local axis orientation
  Determine maximum of subtended angles computed for all local axis
  orientations
Next discontinuity location
```

The computational burden associated with this analysis is eased by making extensive use of transformation matrices, in a manner directly analogous to the generation of computer graphics. In this technique, matrices for the fundamental transformations of translation, reflection, rotation, shearing, scaling and projection may be concatenated by matrix multiplication to form a combined transformation matrix. Using **homogenous coordinates** (*i.e.* where four values are used locate a point in three-dimensional space—the three customary Cartesian coordinate values and a fourth scaling parameter), the fundamental transformation matrices required for this analysis are [114]:

(i) rotation θ about x-axis,

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.09}$$

(ii) rotation ϕ about y-axis,

$$\mathbf{R}_y = \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

(iii) rotation ψ about z-axis,

$$\mathbf{R}_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.11}$$

(iv) translations x, y and z,

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.12}$$

(v) projection onto left photograph, camera of focal length d located at $(-e,0,0)$ (see Figure 5.01),

$$\mathbf{P}_l = \begin{bmatrix} 1 & 0 & 0 & e \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{d} & 1 \end{bmatrix} \tag{5.13}$$

(vi) projection onto right photograph, camera of focal length d located at $(e,0,0)$ (see Figure 5.01),

$$\mathbf{P}_r = \begin{bmatrix} 1 & 0 & 0 & -e \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{d} & 1 \end{bmatrix} \tag{5.14}$$

The combined transformation matrices $\mathbf{M}_l$ and $\mathbf{M}_r$, for the left and right photographs respectively, are

$$\mathbf{M}_l = \mathbf{P}_l \cdot \mathbf{T} \cdot \mathbf{R}_y \cdot \mathbf{R}_x \cdot \mathbf{R}_z \quad \text{and} \quad \mathbf{M}_r = \mathbf{P}_r \cdot \mathbf{T} \cdot \mathbf{R}_y \cdot \mathbf{R}_x \cdot \mathbf{R}_z \tag{5.15}$$

and using these the photograph coordinates of a point are computed as

$$\mathbf{V}_l' = \mathbf{M}_l \cdot \mathbf{V} \quad \text{and} \quad \mathbf{V}_r' = \mathbf{M}_r \cdot \mathbf{V} \tag{5.16}$$

where $\mathbf{V} = (x \quad y \quad z \quad 1)^T$ is the vector of homogenous coordinates of a triad point. Although (5.15) and (5.16) indicate the required sequence of transformations as being:

(i) rotation about the *z*- axis (to orientate the triad of points);

(ii) rotation about the *x*-axis followed by rotation about the *y*-axis (to orientate the discontinuity);

(iii) translation (to position the discontinuity surface relative to the cameras); and

(iv) projection to form the left and right camera images;

these do not reflect the order in which the computational loops take place (*i.e.* this ordering indicates the outermost loop as being triad orientation, the innermost loop as projection). The order adopted for the computational loops, as indicated in the pseudocode above, reduces the overall amount of arithmetic necessary, but at the expense of requiring some additional matrix storage (for transformation matrices consisting of only some of the fundamental matrices indicated in (5.13)) and subsequent multiplication to form the combined matrix. Care is required in computation of the rotation angles θ and ϕ (step (ii) above), in order to ensure that they represent the required (α,δ) orientation of the discontinuity (as defined in Figure 5.02), whilst maintaining the x'-axis parallel to the x-y plane. This constraint—equivalent to ensuring the x'-axis represents the strike of the surface—is induced by the requirement to be able to independently rotate the triad on the surface. An analysis of the geometry of the situation reveals that the required rotations are given in terms of α and δ as

$$\theta = \sin^{-1}(-\sin\delta\sin\alpha) \quad \text{and} \quad \phi = \tan^{-1}(\tan\delta\cos\alpha) \tag{5.17}$$

Application of the method and results

The procedure described above has been applied to the determination of error in orientation, arising from the photogrammetric configuration described in the following points.

(i) Cameras of 125mm effective focal length, separated by 10m, each camera axis parallel with the global *z*-axis.

(ii) Discontinuity centres on a 2m grid within a 10m×10m×10m volume and located relative to the cameras as shown in Figure 5.04. The volume does not extend below the plane of the camera base for two reasons: the

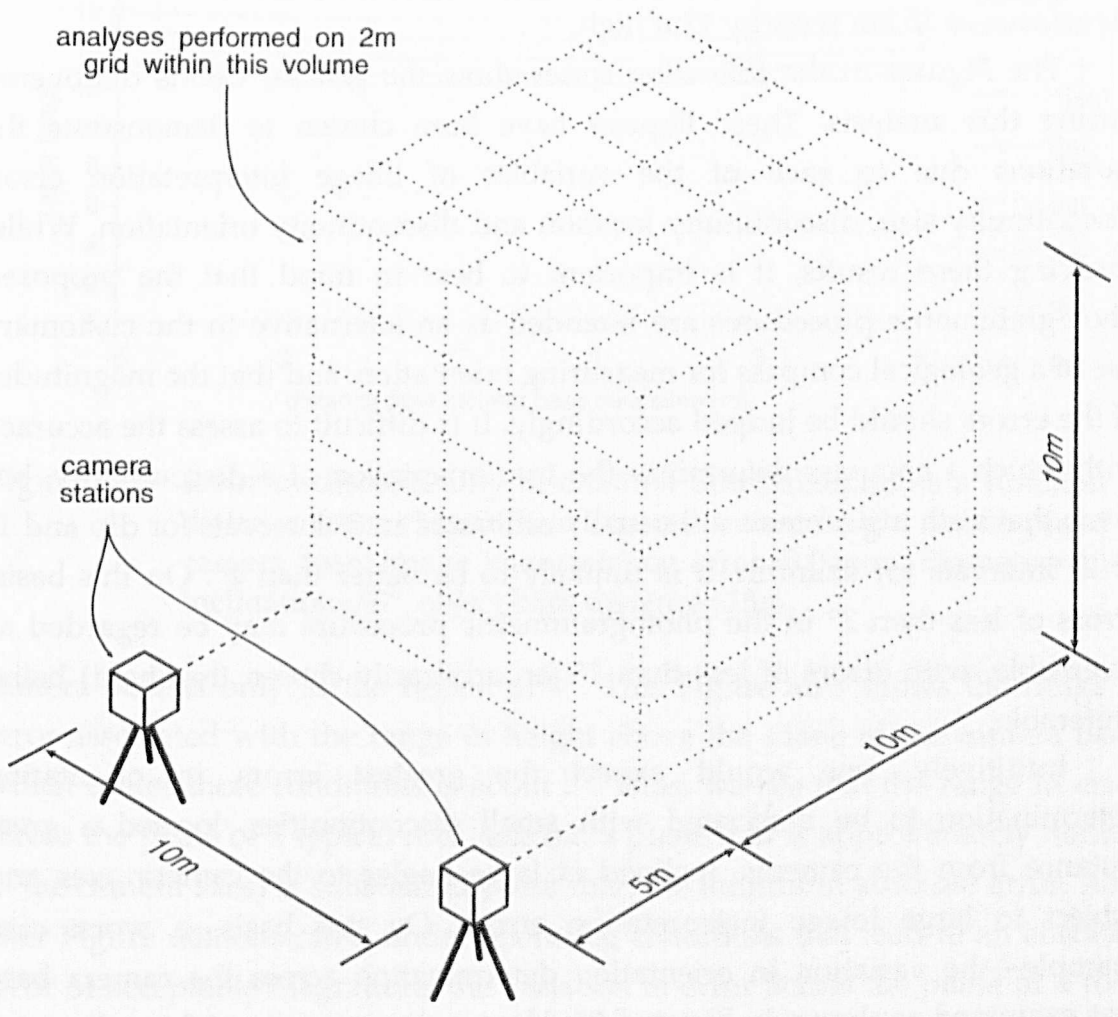


Figure 5.04 Geometry of the error analysis.

presence of loose rock at the base of a rock face would mean that few, if any, instances of discontinuities being visible below this plane would occur in practice; and, the errors will be symmetrical about this plane. The volume does not extend beyond the vertical plane through the camera axes on the assumption that, in keeping with good photogrammetric practice, the photography does not 'extrapolate' beyond the camera base.

(iii) discontinuity triad diameters of 0.1m, 0.5m and 1.0m;

(iv) discontinuities inclined at angles of 0°, 40° and 80° to the camera axes; and

(v) errors in photograph coordinate measurement of 10μm, 30μm and 50μm.

It should be noted that the focal length used in the analysis, 125mm, is not a physical dimension, but rather gives the scale of the photographic image. Thus,

at an object distance of 10m, the scale is 1:80 (*i.e.* 10m divided by 0.125m), indicating that a photograph 210mm wide and 150mm high would show an object area of 16.8m wide by 12m high.

The Figures in the following series show the general trends discovered during this analysis. These Figures have been chosen to demonstrate the variations due to each of the variables of image interpretation error, discontinuity size, discontinuity location and discontinuity orientation. Whilst studying these results, it is important to bear in mind that the proposed photogrammetric procedures are intended as an alternative to the customary use of a geological compass for measuring orientation, and that the magnitudes of the errors should be judged accordingly. It is difficult to assess the accuracy with which a compass determines the true orientation of a discontinuity, but given that such instruments are usually calibrated in 5° intervals for dip and 1° or 2° intervals for azimuth, it is unlikely to be better than 2°. On this basis, errors of less than 2° in the photogrammetric procedure may be regarded as acceptable, with errors of less than 1° (an arbitrarily chosen threshold) being preferable.

Intuitively, one would expect the greatest errors in orientation determination to be associated with small discontinuities, located a great distance from the cameras, inclined at large angles to the camera axes and subject to large image interpretation errors. On this basis—a worst case example—the variation in orientation determination across the camera base was examined as shown in Figure 5.05. Here, a discontinuity of 0.1m diameter and inclined at 80° to the camera is subject to an image interpretation error of 0.04mm. The discontinuity is located 15m from the camera base, and the curves in the Figure show how the orientation error varies with height above the plane of the camera base.

As would be expected, the errors are symmetrical about the mid-point of the camera base, and increase as the height above the plane of the camera base (*i.e.* the distance from the cameras) increases. Similarly, the errors increase away from the camera base mid-point, as the object-to-image distance of one of the cameras increases markedly in these circumstances. This increase in distance could be expected to lead to greatly increasing errors as the discontinuity location moves away from the camera base centreline. However, as the Figure clearly demonstrates, a very small range in errors across the camera base exists. Notwithstanding the absolute magnitude of these errors (due to the choice of extreme operating conditions), the variation across the

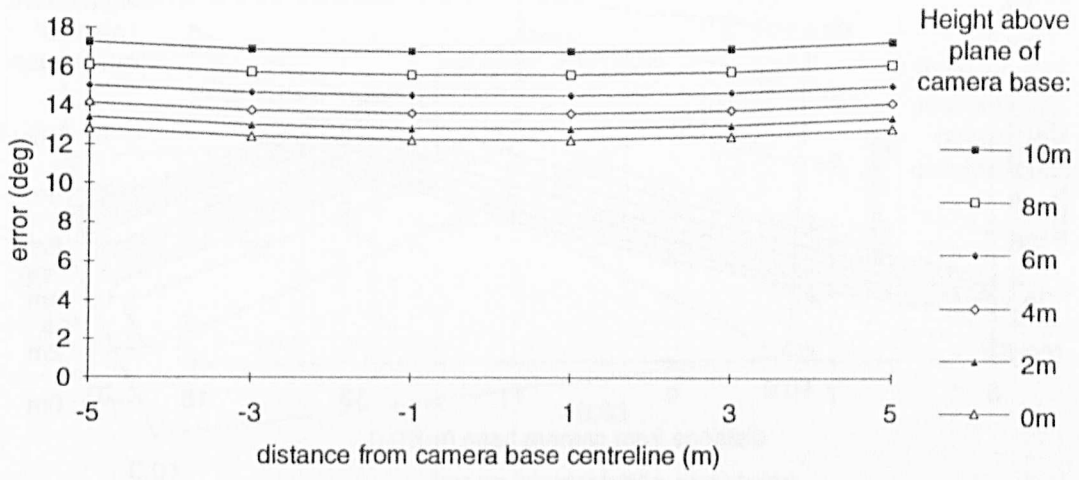


Figure 5.05 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=0.1m, inclination=80°, object-base distance=15m.

camera base is only in the region of 1°. This Figure also shows the range in error associated with the range in height above the plane of the camera base, which under these conditions is about 5°. Thus, we see that the range in error across the plane of a typical rock face (*i.e.* a plane that is approximately normal to the camera axes) is substantially less than the minimum absolute error. As a later Figure demonstrates, under operating conditions that lead to an absolute error of acceptable magnitude, the variation in error across the plane of a rock face is, for practical purposes, negligible.

Figure 5.06 shows how, for a discontinuity of 0.1m diameter, inclined at 80° to the camera axes, located in line with one of the camera axes and subject to an image interpretation error of 0.04mm, the magnitude of the orientation error is extremely dependent on the distance from the camera base line. This is in keeping with the well-known photogrammetric principle that spatial error is proportional to the ratio of camera base to object-to-image distance [143]. In terms of height above the plane of the camera base, the range is both substantially less than this and largely invariant with respect to distance from the camera base. Figure 5.06 must be set in the context of surveying a rock face, for although the range in error indicated in the Figure is large—around 12° for a 10m range in distance from the camera base—the range in distance that is applicable to a given rock face is much less than this. For example, a rock face

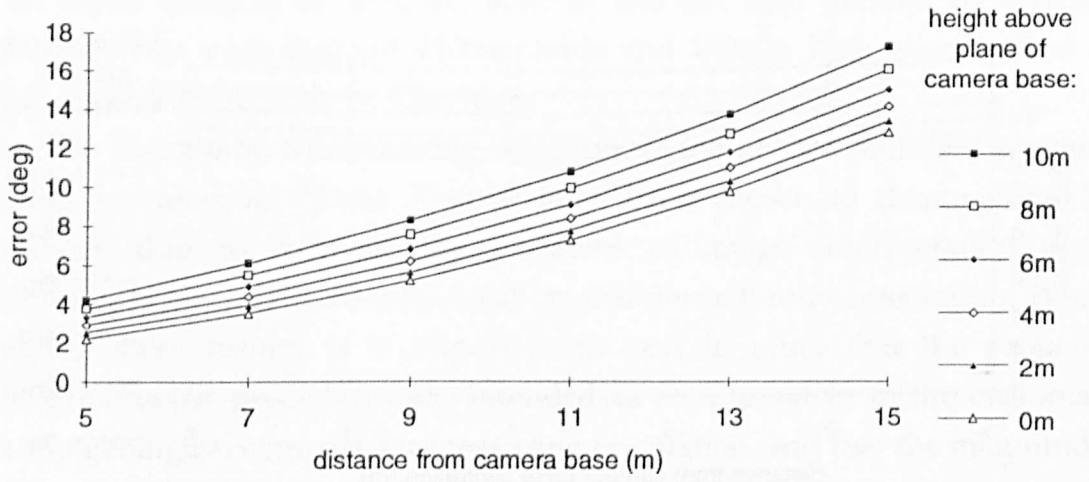


Figure 5.06 Error in discontinuity orientation determination as a function of distance from camera base and height above plane of camera base: image interpretation error=0.04mm, diameter=0.1m, inclination=80°, distance from camera base centreline=5m.

10m high and dipping at 65°, has a range of distances of $10\text{m} / \tan 65° = 4.7\text{m}$, with the result that a range in orientation error of around 5° is appropriate.

Two factors that may be considered critical in the procedure are the orientation of a discontinuity under analysis relative to the camera axes and the image interpretation error. Figure 5.07 shows how these factors affect the error in orientation for a discontinuity with a diameter of 0.1m, located 15m from the camera base, 10m above the plane of the camera base and 5m from the camera base centreline. The Figure clearly shows that it is image interpretation error that is critical, rather than discontinuity orientation. Although image interpretation error may be difficult to assess, the fact that the results are largely independent of discontinuity orientation means that, in a practical situation, discontinuities of all orientations may be surveyed in the knowledge that they will all suffer the same magnitude of error in orientation determination.

That image interpretation error has a significant effect on the error in orientation determination is clearly demonstrated in Figure 5.07. The form of the relation between image interpretation error and distance from the camera base is shown in Figure 5.08. Once more, a discontinuity of 0.1m diameter, inclined at 80° to the camera axes and located 10m above the plane of the camera base and 5m from the camera base centreline has been studied. As can clearly be seen from the Figure, for distances from the camera base ranging between 5m and 15m, the range in orientation error varies from around 3° to

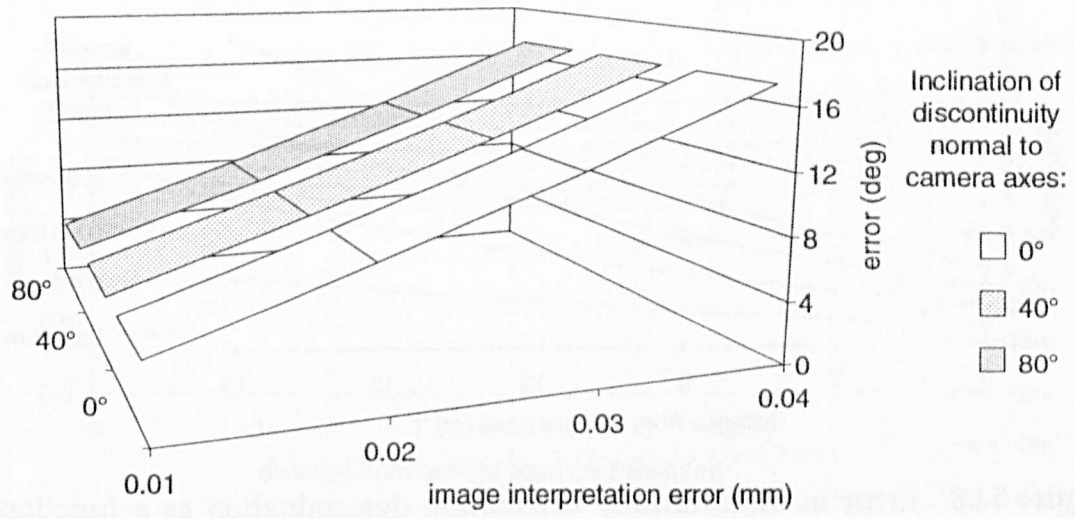


Figure 5.07 Error in discontinuity orientation determination as a function of discontinuity orientation and image interpretation error: discontinuity diameter=0.1m, object-base distance=15m, height above plane of camera base=10m, distance from camera base centreline=5m.

around 13°, as the image interpretation error varies from 0.01mm to 0.04mm, respectively. Again, over a smaller range of distances from the camera base (say, 5m), which is more in keeping with surveying a rock face, the range in orientation error is much reduced—to around 5° in the case of an image interpretation error of 0.04mm.

The results presented in Figures 5.05 through to 5.08 are intended to demonstrate trends in the variation of discontinuity orientation determination, and as such are all for extreme cases: small discontinuities at large distances from the camera, subject to large image interpretation errors. Figures 5.09, 5.10 and 5.11 show more realistic cases. These three Figures show, respectively, discontinuities of 0.1m, 0.5m and 1.0m diameter, located 11m from the camera base and subject to an image interpretation error of 0.04mm. The conclusion that may be immediately drawn from these Figures is that, providing discontinuities with a diameter greater than 0.5m are surveyed, the errors due to this photogrammetric procedure will be at least equal to, and probably better than, those due to a geological compass. To put the results into the context of the 1:80 scale photography assumed for this analysis, discontinuities that appear on the photographs with a size greater than around 6.5mm will be suitable for this low-error surveying.

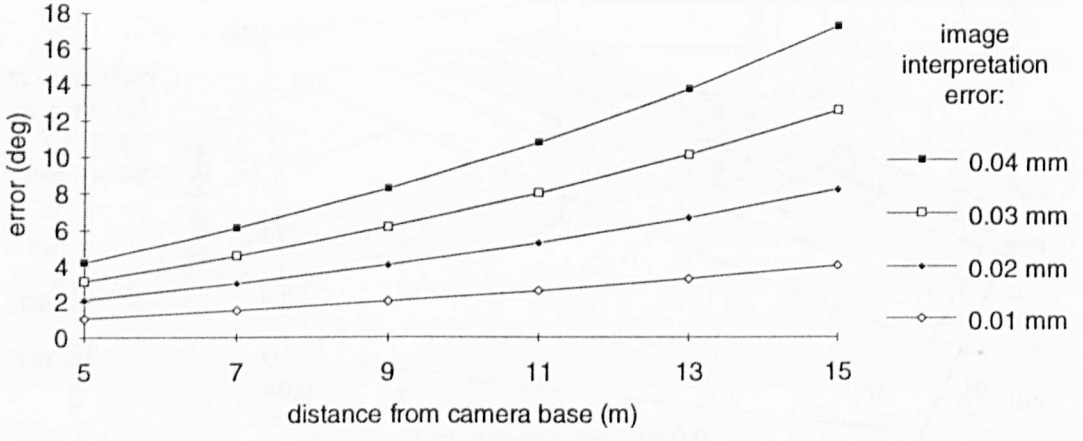


Figure 5.08 Error in discontinuity orientation determination as a function of distance from camera base and image interpretation error: diameter=0.1m, inclination=80°, height above plane of camera base=10m, distance from camera base centreline=5m.

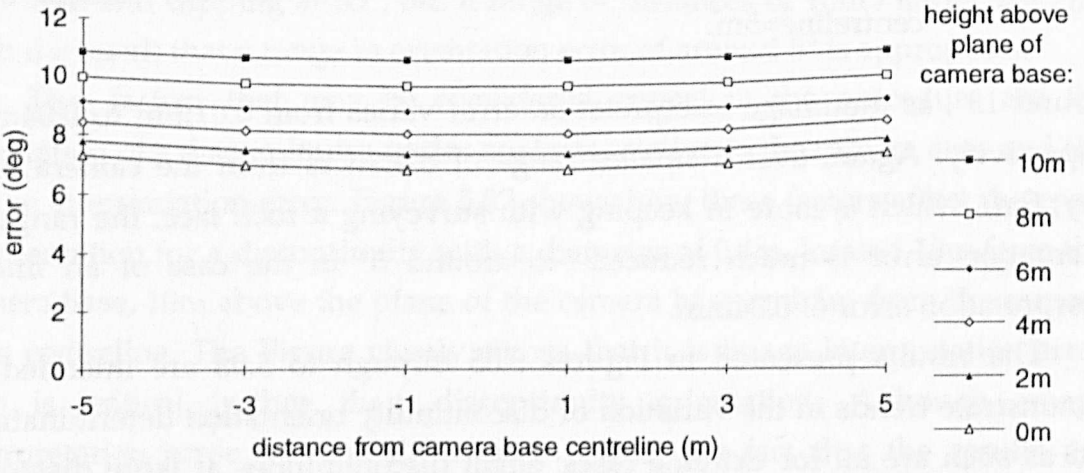


Figure 5.09 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=0.1m, inclination=40°, object-base distance=11m.

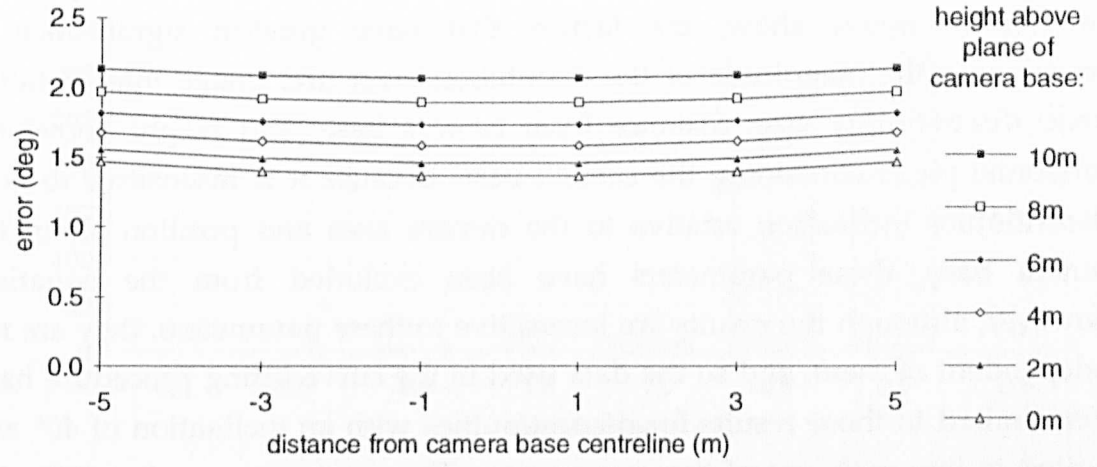


Figure 5.10 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=0.5m, inclination=40°, object-base distance=11m.

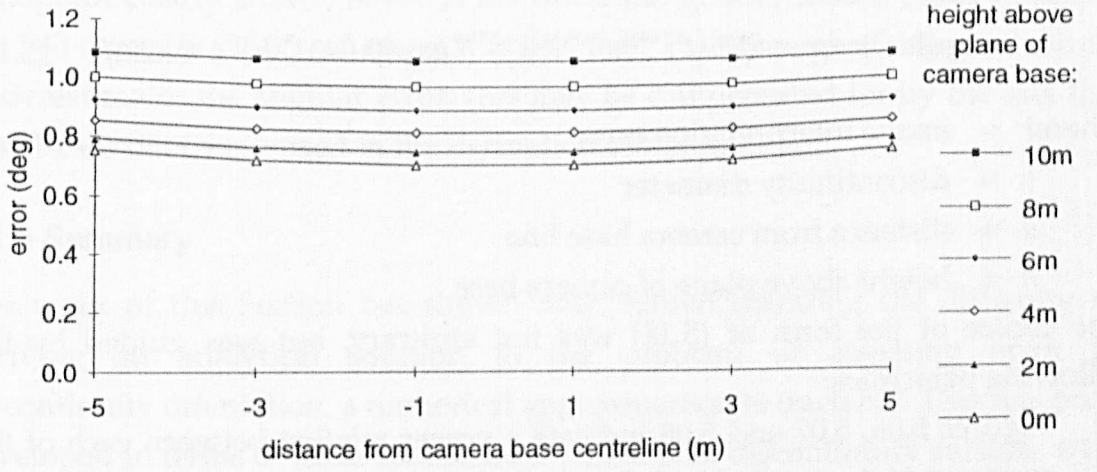


Figure 5.11 Error in discontinuity orientation determination as a function of distance from camera base centreline and height above plane of camera base: image interpretation error=0.04mm, diameter=1.0m, inclination=40°, object-base distance=11m.

An empirical predictive equation

Whilst this numerical analysis has produced interesting results, it is impractical to have to resort to such an analysis in order to predict orientation error. Therefore, a predictive equation has been produced by fitting a curve—on the

basis of a least-squares estimate—to the results of the numerical analysis. As the Figures above show, the factors that have greatest significance in determining the magnitude of the orientation error are: image interpretation error; discontinuity size; distance from camera base; and height above the horizontal plane containing the camera base. Because it is insensitive to both discontinuity inclination relative to the camera axes and position along the camera base, these parameters have been excluded from the equation. However, although the results are insensitive to these parameters, they are not independent of them, and so the data used in the curve fitting procedure have been limited to those results for discontinuities with an inclination of 40° and located in line with one of the camera axes. These values were chosen on the basis of (i) a practical limit to the inclination of a discontinuity to the camera axes, and (ii) the location where the angular errors are greatest.

Thus, on the basis of 540 data (*i.e.* all combinations of 5 values of image interpretation error, 3 values of discontinuity diameter, 6 values of distance from the camera base line and 6 values of height above the plane of the camera base), the following equation results:

$$\text{angular error} = \sin^{-1}\left(t^{1.00435} s^{-0.98948} d^{1.37749} \exp(0.04317h - 4.42652)\right) \qquad (5.18)$$

where t = image interpretation error
s = discontinuity diameter
d = distance from camera base line
h = height above plane of camera base

The choice of the form of (5.18) was not arbitrary, but was guided by the following principles:

(i) Figures 5.06, 5.07 and 5.08 indicate a power relation between each of the parameters t, s, and d and the error in orientation determination, and a linear relation between h and the error in orientation determination. In order to fit a curve using linear least squares, logarithms must be used, with the result that the final form of the function contains an exponential term.

(ii) The angular error is computed from linear data (*e.g.* discontinuity size), and so a trigonometric ratio must be used to convert between the two parameters. Although a tangent is indicated by the ratio between error sphere radius and discontinuity diameter, during the curve fitting process the sine function was found to result in a better fit.

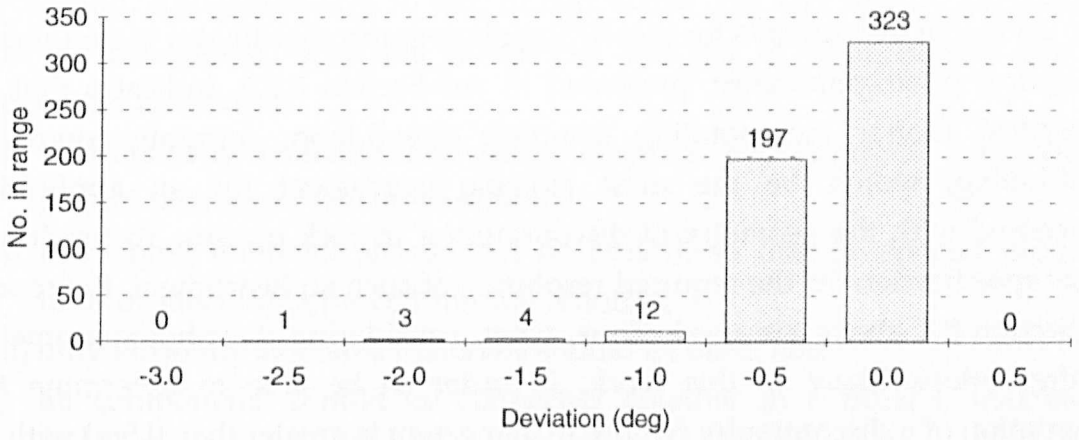


Figure 5.12 Deviation between angular error computed using (5.18) and angular error derived numerically.

The accuracy with which (5.18) predicts orientation error can be judged from Figure 5.12, in which a histogram of the deviation between the numerical and the predicted results for all 540 cases analysed is presented. As this histogram clearly shows, never is the deviation greater than 3°, and in 520 of the 540 cases (*i.e.* 96.3% of cases) it is less than 1°. Although (5.18) consistently underestimates the angular error, this may be compensated for by the fact that worst-case data were used in the derivation of the equation.

5.1.3 Summary

The work of this Section has shown how, notwithstanding the difficulty of deriving an analytical solution to the problem of assessing error in discontinuity orientation, a numerical approximation is tractable. This has been developed in terms of three coordinated points on a discontinuity surface, with an image interpretation error associated with each point. The results of the analysis are encouraging, showing that for likely photogrammetric and rock mass geometrical configurations, an orientation error of less than 1° will result even with a large image interpretation error of 0.04mm. On the strength of these results, we can see that an analytical plotter of only modest resolution—say, 25μm—will be appropriate for photogrammetric surveying of a rock mass. The next Section highlights how such an instrument was designed and constructed as part of this work.

5.2 Design and Construction of an Analytical Plotter

The review of instruments for measuring photograph coordinates in non-metric analytical photogrammetry, presented in sub-Section 3.2.3, indicates that an analytical plotter incorporating complete closed-loop, computer-controlled positioning, would be the most suitable instrument for an application concerned with the geometry of discontinuities in rock masses. To produce a basic specification for the required resolution of such an instrument, the results of Section 5.1 above are used. Thus, when considering the photogrammetric configurations likely in this work, in order to be able to determine the orientation of a discontinuity (whose planar extent is greater than 0.5m) with an error of less than 1°, photograph coordinates should be measured with a resolution of 25μm or better.

Analytical plotters of this specification are not commercially available, because there is no industrial requirement for this level of sophistication at such a low (in the context of photogrammetry) resolution. Thus, as part of this work, the current Author has designed and constructed a unique low-cost analytical plotter incorporating these features. The ensuing four sub-Sections describe this novel development in the following terms: general specification; mechanical components and construction; electrical and electronic components and construction; and control algorithms and software. These four sub-Sections present only a summary of the instrument: a thorough description, together with complete design and construction details, is presented in Appendix A. All of the design details given in both this Section and Appendix A are the original work of the Author.

5.2.1 General specification

At this point, it is useful to state all of the fundamental requirements of the analytical plotter, as these form its general specification. These are:

(i) the optical system must provide illuminated floating spot marks;
(ii) the instrument must be able to hold customary 230mm×230mm photogrammetric prints;
(iii) the photograph positioning system must have a repeatable resolution of 25μm or better;
(iv) the control system must be able to drive the photographs to predetermined positions; and
(v) real-time removal of y-parallax is essential.

Furthermore, the principles laid down in Section 1.2—*i.e.* flexibility in application, straightforward in use, of reasonable cost and readily available—must be considered, with the result that this list of requirements can be augmented as follows:

(vi) commercially available mechanical, electrical and electronic components should be utilised wherever possible;

(vii) machining tolerance should not be critical, so that no special machine tools or machining procedures are required;

(viii) only normally available materials should be used; and

(ix) all components should be connected together in a manner that eases fabrication, assembly and—should modifications be required—subsequent disassembly.

On the basis of these nine requirements, two aspects of the design and construction become clear, namely that (i) as indicated in sub-Section 3.2.3, a commercially available mirror stereoscope provides the most convenient source of both the viewing optics and the illuminated floating spot system; and (ii) the most suitable microcomputer for implementation of the control system is an IBM-PC based machine, given the ubiquitous nature and low cost of these machines, together with the wealth of engineering data regarding the interfacing of other equipment to them.

One further aspect of the design that is indicated by these criteria is the choice of x-parallax measurement method. Although mirror stereoscopes incorporate a system for this, whereby the position of one of the collimated light beams that form the illuminated floating spot can be manually adjusted using a micrometer, such a scheme is inconvenient for implementation in an analytical plotter. This is because systems are required in such an instrument to mechanically move the photographs, and so it is simpler to adapt these to vary the x-direction spacing of the photographs, rather than include a separate system for adjusting the position of the light beam. This is the approach adopted here.

The remaining aspects of the design and construction of the analytical plotter are, then, to produce both an electromechanical system for positioning the photographs beneath the optics of the stereoscope, and a suitable computer interface. The next two sub-Sections address these aspects from the point of view of the mechanical and the electronic components, respectively.

5.2.2 Mechanical components and construction

The fundamental purpose of the mechanical system in an analytical plotter is to allow two stereophotographs to be moved in a controlled fashion, over the complete scene covered by each photograph, such that an observer can both perceive a stereomodel of the object and measure parallax. For the application under consideration here, a photograph movement system is required that is able to coordinate points anywhere in a region 230mm square—the customary size of stereophotographs used in photogrammetry—to a resolution of 25μm or better.

The following two Divisions show how the various mechanical components required for such a system have been selected, and how they have been assembled to form the chassis of an analytical plotter.

Component selection

The core of an analytical plotter—a plane coordination system with 230mm of travel and a resolution of 25μm—is a mechanical system of limited application, with the result that manufacturers do not mass produce such things. Although standard laboratory *x-y* stages are available that provide coordination in a plane through the use of micrometers, their development as tools for use in microscopy means that their travel is limited to only a few tens of millimetres. Consequently, it has been necessary to identify components from which a system suitable for photogrammetric measurements can be constructed.

A number of manufacturers of precision engineering components produce linear bearing systems suitable for constructing custom *x-y* movement systems, and these allow the limitation of insufficient movement to be overcome. However, although such linear bearings allow construction of an *x-y* movement system capable of covering a region 230mm square, some means of both moving the system and determining its position is necessary. Table 5.01 shows the various possible combinations of engineering components for implementing linear movement and associated measurement. Of these, the linear motor/encoder combination can be discounted on cost (several hundreds of Pounds each), and the linear actuator/encoder combination can be discounted on terms of both cost and lack of fine resolution control. Of the rotary electric motors, stepping motors have an inherent appeal for use in a measuring instrument, due to the manner in which they settle at fixed rotor orientations and hence are extremely stable when stationary; servo motors are more applicable to situations in which constant running speed is required. On the

Table 5.01 Linear motion drive and measurement systems.

Drive system			*Measurement system*
linear electric motor			linear encoder
rotary electric motor (servo or stepping motor)	connected to	lead screw & drive nut rack & pinion chain & sprockets toothed belt & pulleys	rotary encoder (attached to motor) linear encoder (attached to carriage)
linear actuator (hydraulic, pneumatic or electromagnetic)			linear encoder

grounds of cost, rotary rather than linear encoders are the most suitable, and the selection of these components in discussed in the following sub-Section.

Now, the greatest number of steps per revolution attainable with commercially available stepping motors is 400, and in order to convert this angular step directly into a linear step of 25μm (the desired resolution of the analytical plotter), the required pitch circle radius of a pinion, sprocket or toothed pulley is

$$r = \frac{s}{\theta} = \frac{0.025}{(2\pi/400)} = 1.59\text{mm} \tag{5.19}$$

which is smaller than any practical components. However, using a lead screw for the same task requires a screw pitch of only

$$p = 400 \times 0.025 = 10\text{mm} \tag{5.20}$$

which is an extremely coarse thread. Indeed, lead screws of a size appropriate for this analytical plotter (about 6mm diameter) tend to have a pitch of about 1mm, which results in a linear step of 2.5μm—a resolution that is an order of magnitude better than required.

Following a survey of components readily available on the market, the following have been selected to form the basic mechanical system:

(i) single- and twin-track linear bearings (linear accuracy 10μm/m), formed from running elements—comprising two recirculating tracks of ball bearings—arranged on 12mm diameter hardened and ground steel shafts mounted in a light alloy extrusion, distributed in the UK by Unimatic Engineers Limited, London;

(ii) $\frac{1}{4}$-20 (*i.e.* 0.25" diameter, 0.05" pitch), precision ground (lead accuracy of ±0.003"/ft, ±0.00025"/in), stainless steel, Acme thread lead screws,

manufactured by Precision Industrial Components Corporation, Connecticut, USA; and

(iii) size 17 (*i.e.* 42mm overall dimension of stator), 4 phase, 0.6A/phase, 1.8° step angle (positional accuracy ±0.054°) stepping motors, manufactured by Sanyo Denki, Japan.

As a result, the instrument possesses a theoretical fundamental linear resolution of either 6.35μm or 3.175μm, depending on whether the stepping motors are driven in full-step or half-step mode, respectively (stepping motor driving is discussed fully in Appendix A). Hence, with these components the instrument easily achieves the required fundamental linear resolution of 25μm, whilst the pitch accuracy of the lead screws, the linear accuracy of the bearings and the positional accuracy of the stepping motors are sufficiently high to generate systematic errors of very low magnitude.

Construction

A great deal of attention has been paid to constructional requirements and methods, to ensure that only standard workshop equipment and practices would be required for the manufacture of the analytical plotter. To this end, the skeleton of the analytical plotter has been constructed from aluminium alloy parts, milled from pieces of plate and bar, with the various components connected by socket head capscrews and countersunk screws. Whilst this strategy obviates the requirement for close-tolerance machining, the resulting freedom of movement introduces problems of non-orthogonality of the movement axes, lack-of-fit between the stepping motors and leadscrews, and non-parallelism between each lead screw and its corresponding linear bearing. The first of these problems was overcome through the use of the DLT for computation of object space coordinates, but to overcome the other two necessitated inclusion of particular assemblies.

To account for lack-of-fit between the stepping motors and the leadscrews, the two components have been mounted independently of each other on a base plate: the lead screws are restrained by plummer-blocks containing miniature roller bearings; and the stepping motors are held in a simple mount. Between the two, a backlash-free Oldham coupling accommodates both axial and radial lack-of-fit. This arrangement is shown in Figure 5.13.

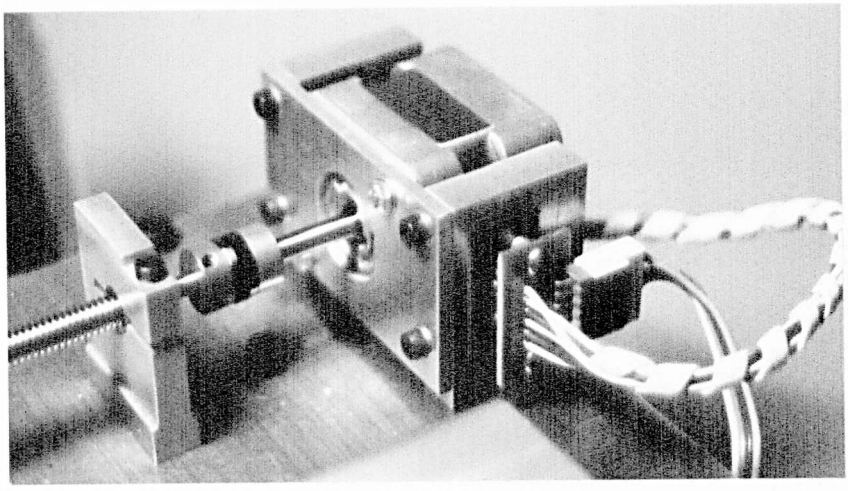
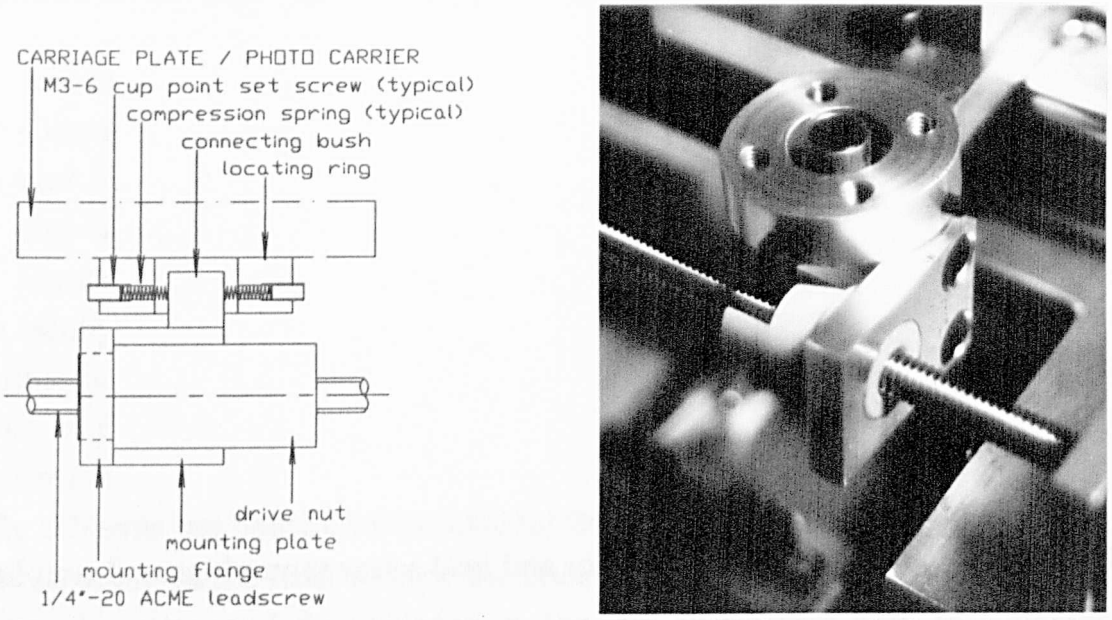


Figure 5.13 Detail of stepping motor and lead screw mountings and coupling.

The problem of non-parallelism between a lead screw and its corresponding linear bearing is exacerbated by the possibility of the lead screws themselves not being straight (no manufacturing tolerance is given for this, as it has negligible effect on the linear accuracy of a lead screw). Consequently, a novel form of connection has been devised, in which a bush on the drive nut is connected to a ring under a plate connected to the bearing carriage *via* high-stiffness (2.61N/mm) compression springs. A sectioned engineering drawing of the device is shown in Figure 5.14(a), and a photograph of the complete assembly is shown in Figure 5.14(b). By this means, a relative movement of about ±1mm between the bush and the ring is accommodated, with the pre-compression in the springs ensuring that when the system comes to rest, the bush always takes up the same horizontal position within the ring.

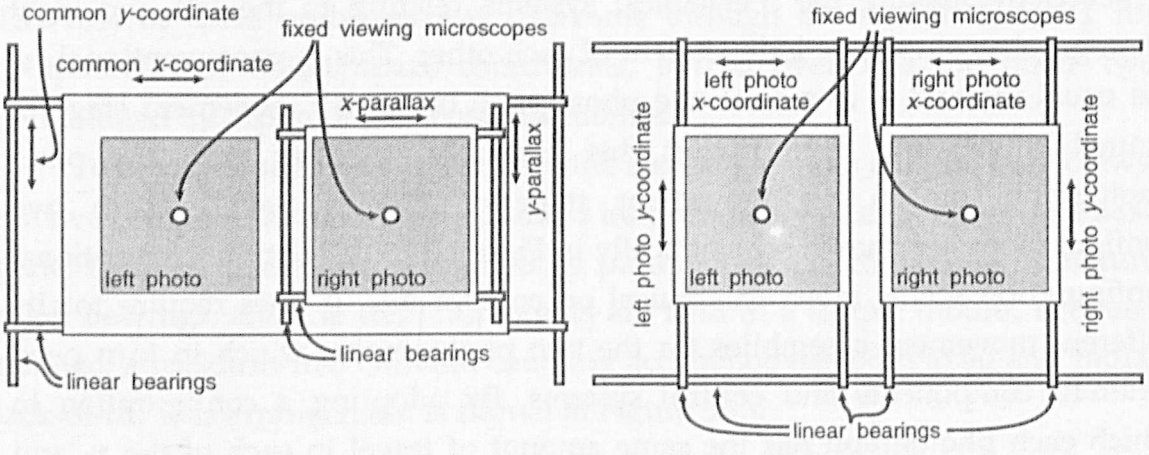
Finally, a great deal of simplification in the construction has been achieved by making the mechanical systems relating to the left and right photographs completely independent of each other. This is unconventional, as the usual method is to mount one photograph on an x-y movement stage of limited travel, and then mount this assembly, together with the other photograph, on an x-y movement stage of 230mm travel. These two configurations are shown schematically in Figure 5.15. Whilst the conventional configuration seems more economical on components, it does require totally different movement assemblies for the two photographs, which in turn need different components and control systems. By adopting a configuration in which each photograph has the same amount of travel in each of the x- and y-directions, four identical movement assemblies are needed, which makes for rationalised component and control system requirements. Of course, an



(a) sectioned arrangement (b) device as built

Figure 5.14 Coupling between lead screw and carriage.

independent system such as this means that there is no direct *mechanical* link between the two photographs, only a *virtual* link through the control system. Whilst this arrangement permits simpler construction and increased accommodation of x- and y-parallax, it does so at the expense of needing sophisticated control techniques. This point is considered further in the next two sub-Sections.



(a) customary (b) independent

Figure 5.15 Customary and independent photograph configurations.

5.2.3 Electrical and electronic components and construction

The previous sub-Section has shown how operational requirements led to the choice of stepping motors for the analytical plotter. In this sub-Section, the method by which these motors—together with appropriate rotary position encoders—have been assembled into a computer-controlled closed-loop positioning system is discussed. The first Division is concerned with component and technique selection, whereas the second is concerned with specific construction details.

Component selection

The implementation of closed-loop position control for stepping motors has traditionally been very complex, but the recent introduction of a general purpose motion controller in integrated circuit form has simplified matters. The HCTL-1000, manufactured by Hewlett-Packard, is ideally suited for the development of the analytical plotter, containing as it does all of the time-intensive computational functions necessary for position and velocity control of a stepping motor/rotary encoder system. Whilst this device provides high-performance control and is fully programmable and hence extremely flexible, it is designed to interface directly with a host microprocessor *via* a multiplexed 8-bit address and data bus, with the result that the simplest means of accommodating in the IBM-PC is to mount the device on an interface board that plugs directly into the main system board of the machine. This was the route followed, with the four devices required for the analytical plotter (one device to control each of the x- and y-axes of motion, for both the left and right photographs) then interfaced to the host machine through the use of standard TTL integrated circuits. These circuits provide all of the address decoding, bus multiplexing and control signals for the HCTL-1000 devices.

This closed-loop control circuit—the design and construction of which was entirely the work of the Author—represents a major development in the design of high-performance photogrammetric coordination instruments. Although the circuit is unique, it has demonstrated the feasibility of mounting a complete and sophisticated multi-channel digital closed-loop control system on a single microcomputer interface board.

To close the control loop, the HCTL-1000 is designed to interface with position encoders of modest operating characteristics, and Hewlett-Packard have developed a series of such rotary encoders, in modular form, that connect directly between the motor and the HCTL-1000. Thus, the rotary encoders

selected for the analytical plotter were three-channel, 500pulse/revolution HEDS5540 optical encoders, manufactured by Hewlett-Packard. The two count trains output by these encoders are used by the HCTL-1000 in quadrature (*i.e.* out of phase by 90 electrical degrees), such that an effective resolution of 2000 pulses per revolution is obtained. This high resolution is not used directly for position measurement, but rather for closed-loop position control, with the result that the operational flexibility offered by the HCTL-1000 is maximised. Figure 5.16 shows one of these encoders mounted on a lead screw in the analytical plotter. Finally, although the use of a stepping motor, lead screw and rotary encoder assembly for linear position measurement introduces systematic errors due to both motor step accuracy and screw pitch accuracy, the high accuracy of the mechanical components selected for the analytical plotter (see Section 5.2.2) means that this has been restricted to negligible levels.

The operating potential of stepping motors is optimised when they are driven using a constant-current source. This, in turn, is most conveniently implemented by connecting a high-voltage, high-current power supply to the motor, and using a closed-loop control system to maintain a constant current in the motor coils by rapidly switching the power supply on and off. These driving circuits are known as **switch-mode constant-current chopper drivers**, and until recently were extremely difficult to construct. Modern developments in high-current integrated circuit technology, however, mean that devices are now available that implement all of the required control and amplification functions in standard layout packages. For the driver built here, a matched pair of such devices—a L6506 current controller and a L293E push-pull amplifier,

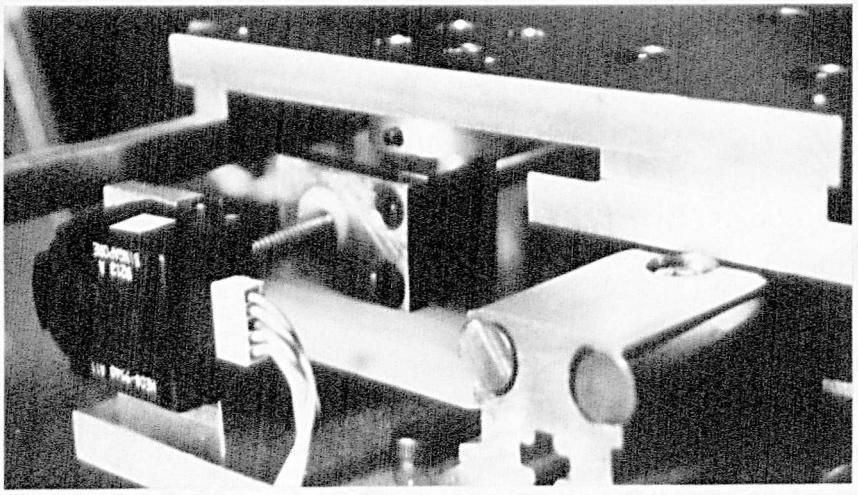


Figure 5.16 HEDS5540 rotary encoder mounted on a lead screw.

both manufactured by SGS-Thomson Microelectronics Ltd—were selected for each motor. A standard power supply unit from an IBM-PC microcomputer is used as the power supply for the driver, with the result that each motor phase is driven at 12V and 0.5A, to provide a minimum starting torque of 46.6Nmm.

Finally, protection for the instrument against physical damage has been included through the use of limit switches, hard-wired into an emergency stop input on the HCTL-1000 controllers. Five such switches have been incorporated: one for each of the x- and y-axes of movement for both the left and right photographs, and one for the x-distance between the two photographs. Traditional mechanical switches were not used, due to the requirement for additional de-bouncing circuitry in order to interface such devices to high-speed electronics. Instead, integrated optical switches incorporating TTL-compatible circuitry were adopted. The particular devices used—type HOA2001, manufactured by Honeywell—are constructed with an infra-red emitter and detector on either side of a slot in the switch body, and operate when the light beam passing between these elements is interrupted by a wiper. Figure 5.17 shows a typical limit switch and wiper arrangement in the analytical plotter.

Construction

The construction of the electronic systems can be broken down into three separate parts: the computer interface circuitry; the stepping motor driver circuitry; and miscellaneous connecting circuits. These are described sequentially in the following paragraphs.

Because of the non-proprietary architecture employed in the IBM-PC, and

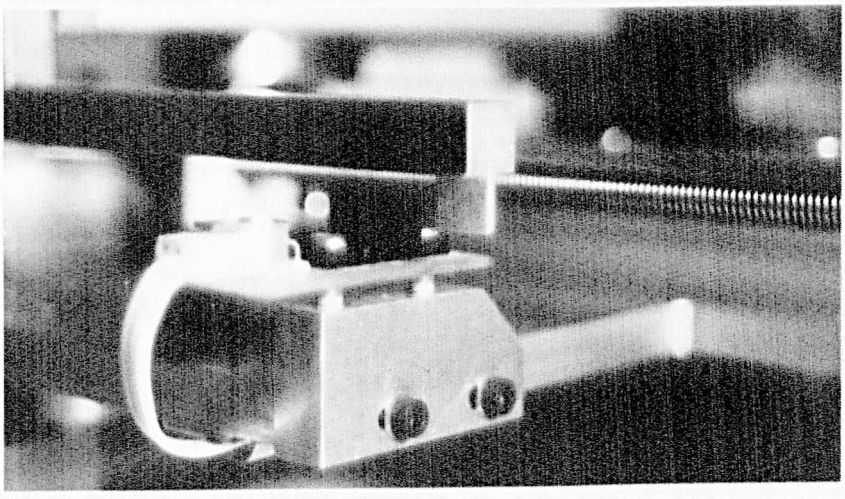


Figure 5.17 Typical HOA2001 optical limit switch and wiper.

the widespread scientific and industrial use enjoyed by such machines, interface prototyping cards that plug directly into the main system board are readily available. In recognition of the fact that a four-channel, closed-loop control circuit, based on the HCTL-1000 processor would be completely novel, a prototyping card based on a wire-wrap system was selected, such that the inevitable circuit wiring changes could be easily made. Similarly, in order to be able to implement extensive logic changes, all of the integrated circuits were mounted in sockets. Although all of the logic circuit connections were made using wire-wrap, the power and ground circuits were soldered to avoid problems of spurious switching due to noise. Figure 5.18 shows the component side of the final interface board.

The board holding the driver circuitry for the stepping motors contains both logic and power circuits. The combined current carried by the power circuits is 4A (4 motors, each with two phases energised at 0.5A/phase), which is beyond the capacity of a wire-wrap system. This, coupled with the fact that the integrated circuits selected for motor current supply and control were from the same manufacturer, and could be hard-wired with confidence, led to the decision to produce a custom-built printed circuit board with fixed connections, with the various components being soldered to it directly. The printed circuit board was produced using the customary exposure and etching method, with the artwork for the wiring being produced with the aid of a CAD package. Figure 5.19 shows both the component and wiring sides of the completed driver board.

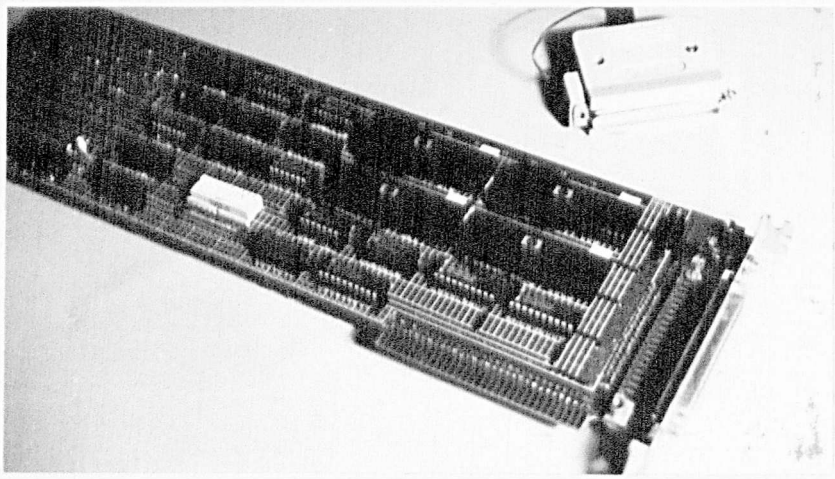
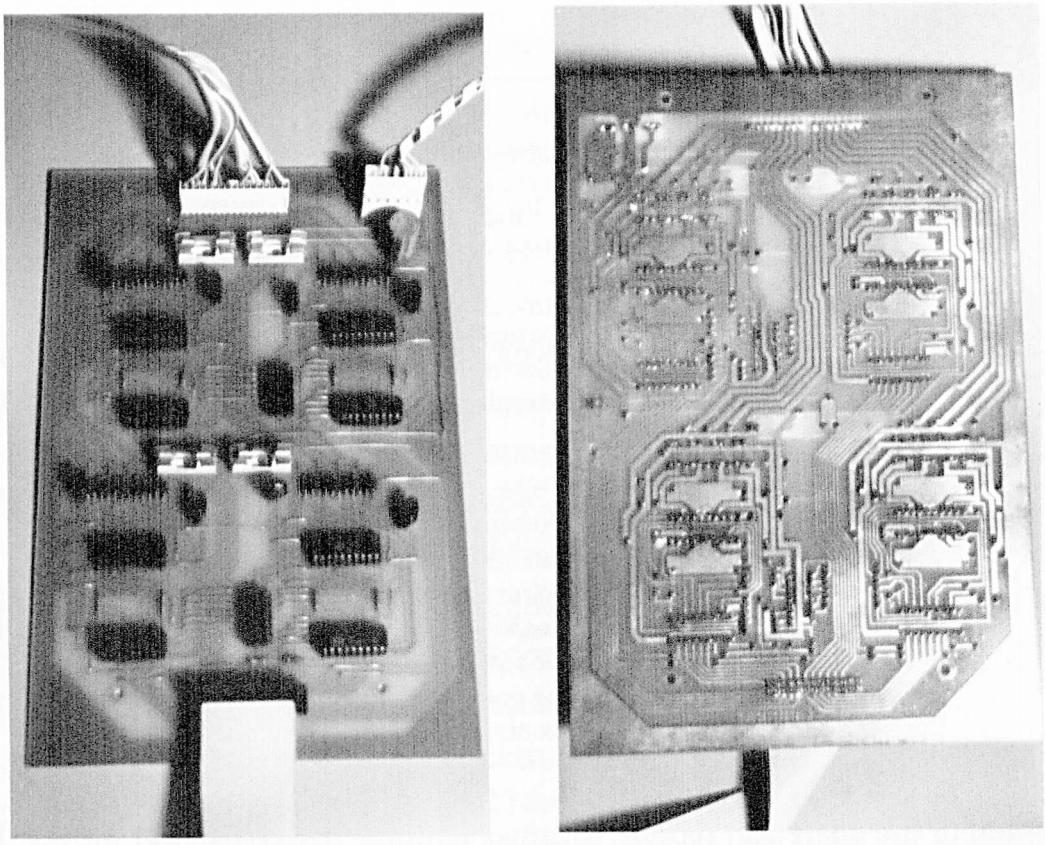


Figure 5.18 Component side of wire-wrap interface board.



(a) component side (b) wiring side

Figure 5.19 Stepping motor driver board.

It was felt important that all of the individual components—stepping motors, rotary encoders, limit switches, interface board, driver board and power supply board—should be connected by cables terminated in plugs or sockets (as appropriate) in order to maintain a high standard of circuit integrity, ease of use and flexibility for component replacement. Most cabling was undertaken using ribbon cable terminated in 0.1"-pitch PKK-type connectors, with the exception of the main cables connecting the interface board to the power supply unit, and the power supply unit to the analytical plotter. In these latter two cases, custom-made screened cables were used to guard against problems of noise and cross-talk. Details of all of these cables and connections is given in Appendix A.

5.2.4 Control algorithms and software

The design of the analytical plotter is such that the user is required to issue a movement command to the host microcomputer, which in turn interprets this and issues the appropriate command(s) to the relevant motion controller(s). Table 5.02 shows the three photograph movement procedures required for

Table 5.02 Photograph movement procedures.

movement required:	each photograph independently	the two photographs in a congruous manner, such that the stereomodel is maintained	one photograph relative to the other, such that the x-parallax varies
during:	stereomodel set-up phase, whilst moving homologous control points to coincide with the illuminated spots	measurement phase, whilst moving around the photographs	measurement phase, whilst changing the elevation of the floating spot relative to the surface of the stereomodel
user inputs:	command x- and y-movement for the relevant photograph	command x- and y-movement for one photograph, the corresponding position of the other photograph being computed by the control program	command x-movement for one photograph, all other axes stationary

operation of the analytical plotter, together with the necessary user inputs. This shows that, at any time, no more than two inputs are required: x- and y-axis command movements. The most convenient method for a user to issue these commands is through the use of a joystick, with displacement of the joystick from the null position representing magnitude of the desired move. Interfacing a joystick to an IBM-PC is trivial, as these machines commonly incorporate a port specifically for these devices, and so this was the user interface adopted for the analytical plotter.

Interpreting the displacement of the joystick as a command velocity allows the HCTL-1000 motion controllers to move the photographs under velocity—rather than position—control, with the result that the instrument is extremely responsive to the user. Thus, the control program simply executes a continuous loop, whereby a reading of joystick displacement is converted to a velocity, this is issued to the relevant HCTL-100 controller as a velocity command, and the loop repeated. The control program has been designed such that the user signals when to record a photograph coordinate by pressing a button on the joystick; at this point, the control program temporarily leaves the velocity control loop to read the current position indicated by the motion controller.

Total velocity control is only appropriate for the first and last phases shown in Table 5.02. During the measurement phase of operation, if the control

program is to maintain a stereomodel free from y-parallax, then it must be able to determine the location of the illuminated spot in one of the photographs relative to object space, and then compute the required position of the other photograph for output as a command position to the relevant HCTL-1000 motion controller. The design of the HCTL-1000 controllers means that this is easily accomplished, as position information is continually available when operating in velocity mode, and output of command positions when operating in position mode can occur regardless of whether a previous command has been completed. Thus, in the measurement phase the control program executes a continuous loop, both issuing velocity commands and reading the position of one photograph, and computing and issuing position commands to the other, until the user requests otherwise by signalling with a joystick button. The speed at which this loop is executed means that the user perceives a smooth motion, with a stereomodel free from y-parallax always being visible.

Many programming languages are available to implement such a control program, but a combination of FORTRAN and assembly language has been chosen here. FORTRAN was chosen for the bulk of code for the simple reasons that the Author is practised in this language, and interfacing to assembly language is straightforward. Assembly language has been reserved for those subroutines that interrogate the joystick and both issue commands to, and read information from, the HCTL-1000 motion controllers. Because it is the HCTL-1000 motion controllers that are dedicated to the computationally intensive tasks associated with maintaining closed-loop control of photograph movement, a relatively modest host microcomputer is sufficient for implementation of the control program.

Appendix B contains a complete listing of the Analytical Plotter Control System (APCS), including all of the assembly language routines for communicating with the interface board and joystick.

5.2.5 Summary

The specification, design and construction of a novel high-performance analytical plotter has been presented in this Section. The instrument uses a standard mirror stereoscope as an optical system and an IBM-PC microcomputer as a host for the control programs, with the electronic and mechanical systems for moving the photographs in a controlled fashion being custom designed and built as part of this project.

The mechanical core of the instrument is an assembly of linear bearings, stepping motors and lead screws that allows independent movement of the photographs over an area of 230mm square, with a fundamental linear step of either 6.35μm or 3.175μm, depending on whether the stepping motors are driven in full-step or half-step mode, respectively. These elements are all carried on a skeleton formed from aluminium alloy parts, milled from pieces of plate and bar, with the various components connected by socket head capscrews and countersunk machine screws.

The electronic control system is based on high-performance, motion control integrated circuits that have been assembled into a unique four-channel, closed-loop control circuit mounted on an interface board that plugs directly into the main system board of an IBM-PC. Position feedback is gained from the use of rotary encoders attached to each lead screw.

In order to optimise the performance of the stepping motors, a switch-mode constant-current chopper driver has been designed and built using integrated circuits mounted on a custom printed circuit board. This circuit (and hence the complete instrument) is powered by a standard IBM-PC power supply unit.

Mechanical protection to the instrument is provided by optical limit switches, hard-wired into the motion control integrated circuits.

The user interfaces with the instrument through a simple joystick connected to the host microcomputer. This machine executes a control program, written in a combination of FORTRAN and assembly language, that allows for the full functionality of an analytical plotter to be developed. The use of dedicated processors for implementation of closed-loop motion control allows a microcomputer of modest power to be used.

The complete instrument is shown in Figure 5.20. At the right of the photograph is the host microcomputer with screen, joystick, keyboard and (internally and hence out of view) the interface circuit. This is sitting on the cabinet that houses the stepping motor drivers and power supply unit. The left of the photograph shows the mechanical part of the instrument. The two photographs are held on Perspex carriers, which in turn are mounted on the assembly of linear bearings, stepping motors and lead screws. Surmounting everything is the mirror stereoscope.

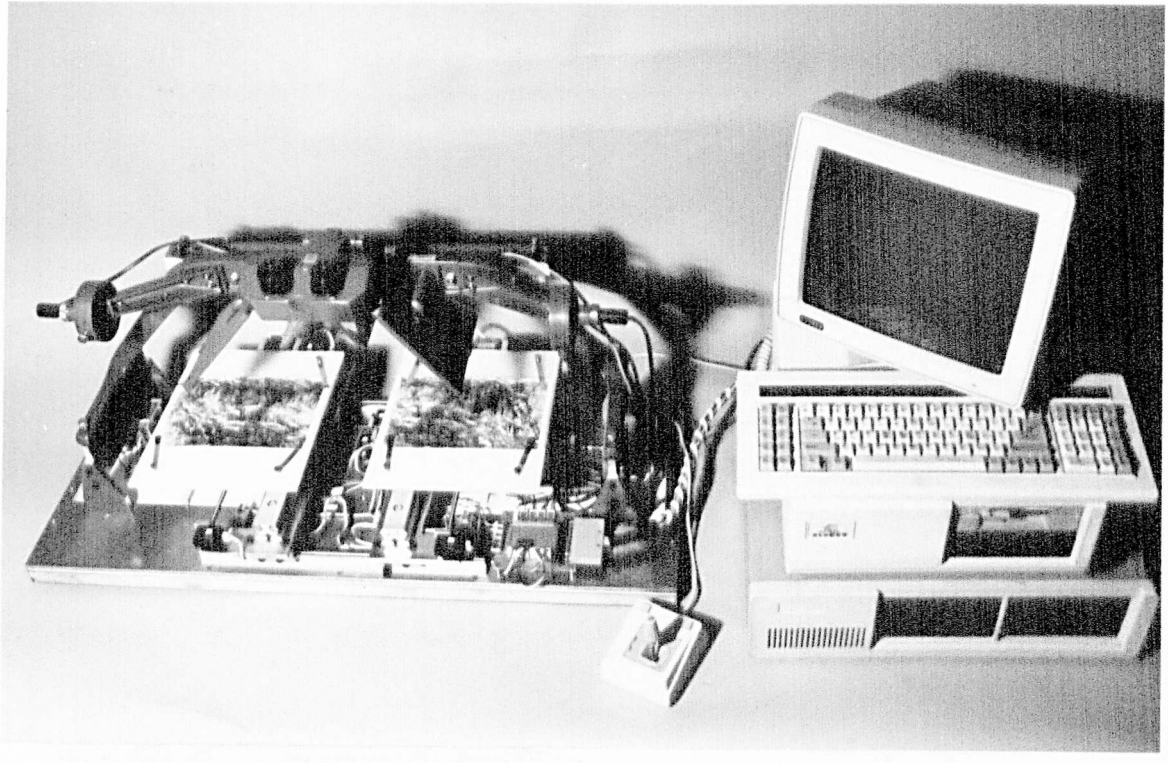


Figure 5.20 The complete analytical plotter.

This instrument is now being successfully used for gathering rock mass geometrical data, using enlarged prints from 35mm non-metric photography, and the results of some of this work are presented in Chapter 7.

Of course, the analytical plotter represents only part of the requirements for successful photogrammetry: the field and data analysis procedures are of equal importance. The particular configuration of photogrammetry at a rock face means that developments to these have been necessary in these areas, and these are described in the next two Sections, respectively.

5.3 Procedures for Introducing Photogrammetric Control

A number of points of known three-dimensional coordinates are required as control for the DLT, in order that the coefficients of the mathematical model relating photograph coordinates to space coordinates may be determined. The challenge in this work has been to develop a technique for introducing this control, subject to the following constraints:

(i) work directly at, or on, the rock face should be avoided;

(ii) the equipment required should be limited to that which is readily accessible to geotechnical engineers;

(iii) specialised, and high cost, items of equipment should not be required;

(iv) the technique should be flexible and easy to follow; and
(v) the requirements of the DLT procedure for control points must be upheld.

Previous attempts to introduce photogrammetric control for rock mechanics applications by other workers has already been discussed in Chapter 3, and a number of deficiencies in their equipment and techniques identified. In order to overcome these deficiencies, whilst adhering to the criteria outlined above, the current Author has developed a control system that uses lengths of thin steel wire rope tensioned in front of the rock face, with each length carrying a number of markers that are coordinated using a straightforward theodolite survey. High precision is gained through the use of an advanced least squares adjustment procedure. This system is described in the following two sub-Sections: firstly, the overall configuration of the system is presented in 5.3.1; and secondly the method by which the least squares adjustment is undertaken is presented in 5.3.2.

5.3.1 Survey configuration

The control survey can be thought of as being composed of two distinct—but connected—three-dimensional networks, that are surveyed simultaneously and subsequently adjusted sequentially. The primary network comprises the cable markers and the two theodolite stations from which they are surveyed, as shown in Figure 5.21. The secondary network is formed by the two theodolite stations used in the primary network, together with one other theodolite station, all arranged in a triangular array. Both the theodolite optical centres, and the ground marker over which they are positioned, are included in this network, as shown in Figure 5.21.

The primary network, shown in Figure 5.21, is used to coordinate the individual control markers attached to each steel cable, on the basis of horizontal and vertical angles measured at the theodolite stations A and B. Rather than coordinate each marker solely on the basis of triangulation, an over-constrained network is produced by including the spacings, measured along the wires, between adjacent control markers. These spacings are recorded before the cables are located at the rock face. In order to be able to compute the coordinates of the markers, accurate coordinates of the two theodolite stations are required, and these are determined from the secondary network.

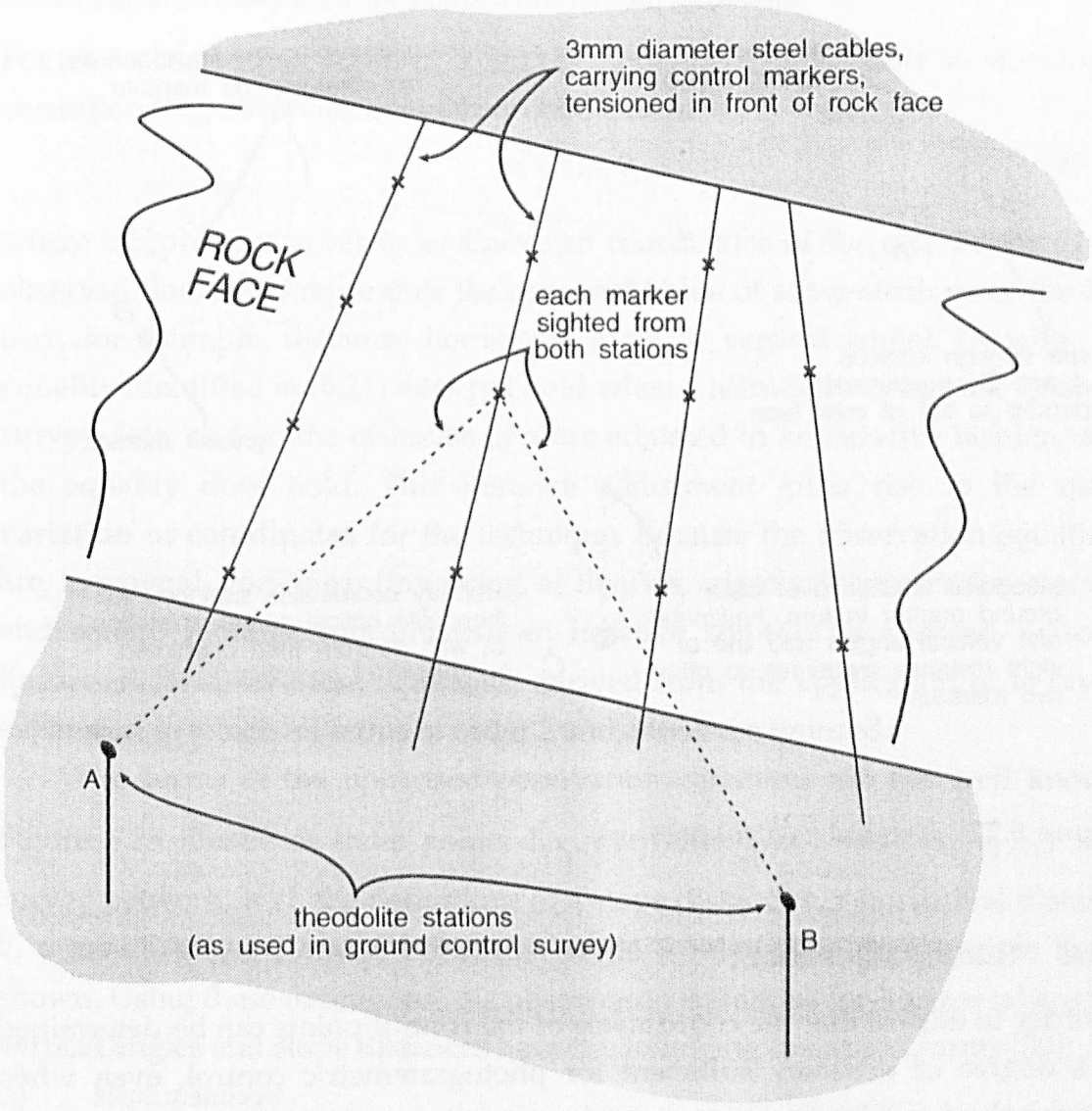


Figure 5.21 Cable control network.

Figure 5.22 shows how the ground network consists of the three ground markers and the optical centre of the theodolite that is set up over each marker in turn. Measuring horizontal angles, vertical angles and line-of-sight distances at each of the three theodolite stations allows the relative three-dimensional coordinates of the theodolite optical centres to be determined. These relative coordinates are computed on the basis of arbitrary coordinates being assigned to one station, with a magnetic compass being used to orientate the network with respect to magnetic north. The coordinates of the theodolite optical centres at stations *A* and *B* are those used as fixed points for the primary network.

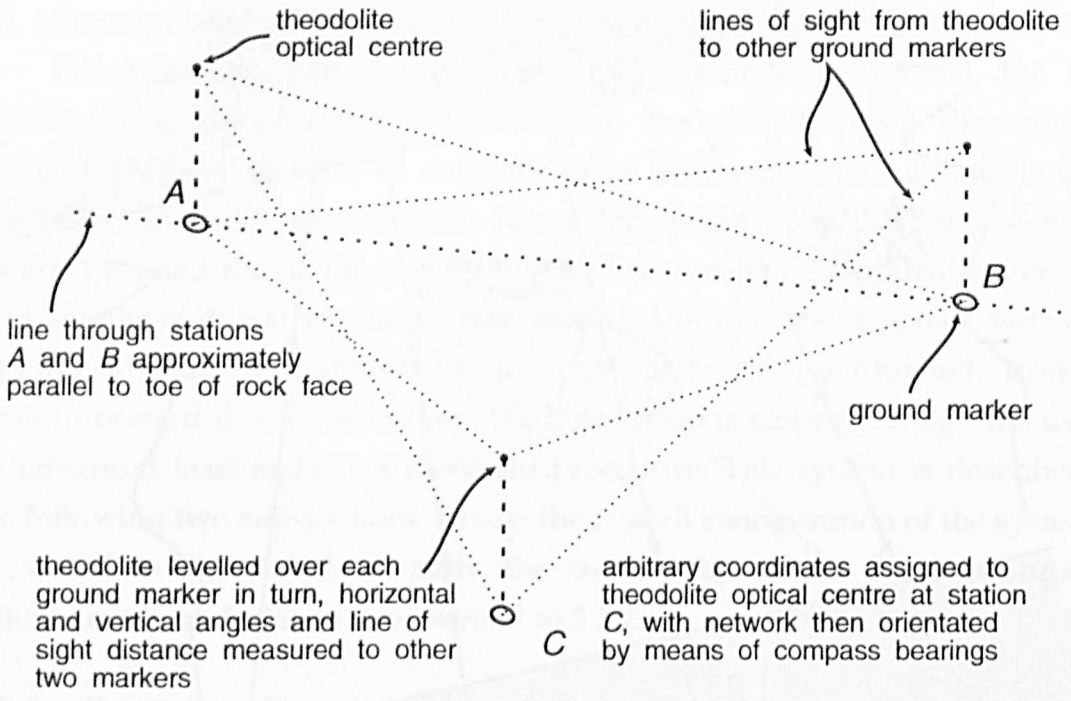


Figure 5.22 Ground control network.

5.3.2 Survey adjustment

In order to ensure that the coordinates of the control points can be determined to a degree of accuracy sufficient for photogrammetric control, even when personnel of minimal survey experience using survey instruments of only moderate accuracy are involved, a sophisticated and rigorous least squares estimation procedure has been developed by the writer for adjusting both the ground and the cable marker networks. The networks involved in this work are complex, and to aid reduction and adjustment of the survey data, an integrated suite of analysis programs, running on a computer spreadsheet program, have been produced by the Author.

As the basis and technique of least squares estimation of coordinates in three-dimensional networks such as those used here is not well known, but is important for an understanding of the technique, a brief description of them is now given.

Least squares estimates for three-dimensional networks

For any observation made in a survey network it is possible to develop a corresponding **observation equation** of the form

$$f(\mathbf{x},\ell)=0 \tag{5.21}$$

where $\mathbf{x}$ represents a vector of Cartesian coordinates of the end points of the observed line, and ℓ represents the observed value of some attribute of the line (say, for example, distance, horizontal angle or vertical angle). Usually, the equality identified in (5.21) does not hold when $\mathbf{x}$ is initially computed from the survey data, and so the elements of $\mathbf{x}$ are adjusted in an iterative fashion until the equality does hold. This iterative adjustment gives rise to the name **variation of coordinates** for the technique. Because the observation equations are, in general, non-linear (involving, as they do, trigonometrical ratios, squares and square roots of coordinates), an iterative solution is adopted to solve linearised forms of these equations, derived from the application of a Taylor expansion in which all terms of order 2 and above are ignored.

The forms of the linearised observation equations are not well known. Figure 5.23 illustrates three points $I(x_i,y_i,z_i)$, $J(x_j,y_j,z_j)$, and $K(x_k,y_k,z_k)$ in a survey network, with the definitions of a slope distance s, a horizontal distance h, a zenith angle ζ and a horizontal angle θ relative to these points being shown. Using these definitions, the observation equations for horizontal angles, vertical angles and slope distances have the following linearised forms [30]:

(i) *slope distances*

$$\begin{aligned}&-\frac{(x_j-x_i)}{\hat{s}_{ij}}\delta x_i-\frac{(y_j-y_i)}{\hat{s}_{ij}}\delta y_i-\frac{(z_j-z_i)}{\hat{s}_{ij}}\delta z_i\\&+\frac{(x_j-x_i)}{\hat{s}_{ij}}\delta x_j+\frac{(y_j-y_i)}{\hat{s}_{ij}}\delta y_j+\frac{(z_j-z_i)}{\hat{s}_{ij}}\delta z_j=\left({}_o s_{ij}-\hat{s}_{ij}\right)+v\end{aligned} \tag{5.22}$$

(ii) *horizontal angles*

$$\begin{aligned}&\left[\frac{(y_j-y_i)}{\hat{h}_{ij}^{\,2}}-\frac{(y_k-y_i)}{\hat{h}_{ik}^{\,2}}\right]\delta x_i-\left[\frac{(x_j-x_i)}{\hat{h}_{ij}^{\,2}}-\frac{(x_k-x_i)}{\hat{h}_{ik}^{\,2}}\right]\delta y_i\\&-\frac{(y_j-y_i)}{\hat{h}_{ij}^{\,2}}\delta x_j+\frac{(x_j-x_i)}{\hat{h}_{ij}^{\,2}}\delta y_j+\frac{(y_k-y_i)}{\hat{h}_{ik}^{\,2}}\delta x_k-\frac{(x_k-x_i)}{\hat{h}_{ik}^{\,2}}\delta y_k=\left({}_o\theta_{ij}-\hat{\theta}_{ij}\right)+v\end{aligned} \tag{5.23}$$

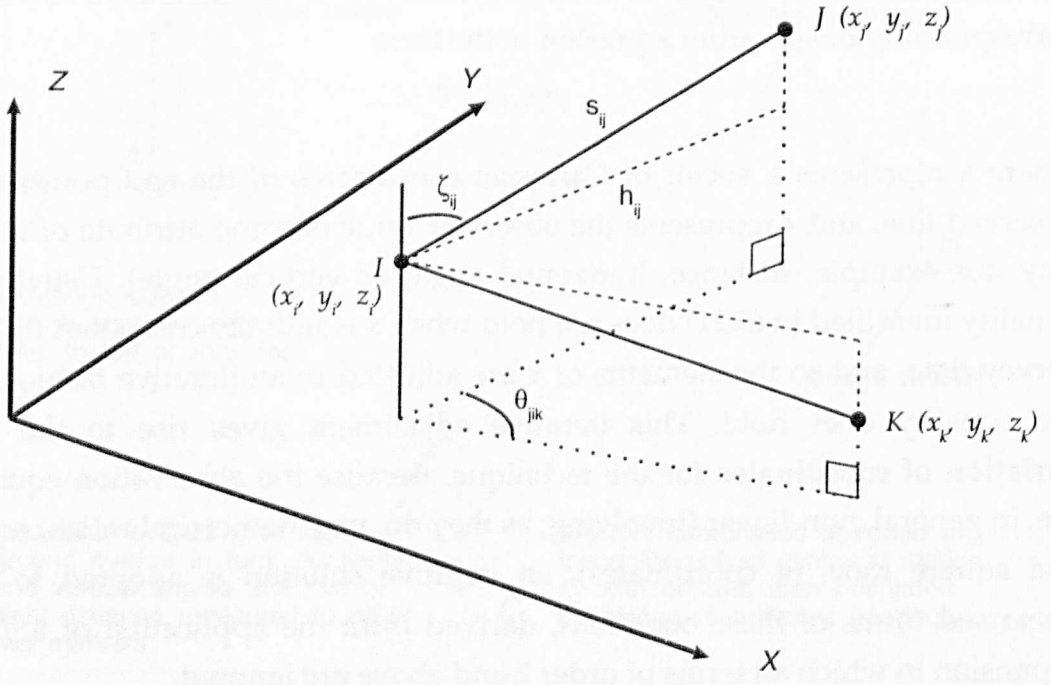


Figure 5.23 Elements of a survey network.

(iii) *zenith angles*

$$\begin{aligned} &-\frac{(x_j-x_i)}{\hat{s}_{ij}^{\,2}\tan\hat{\zeta}_{ij}}\delta x_i-\frac{(y_j-y_i)}{\hat{s}_{ij}^{\,2}\tan\hat{\zeta}_{ij}}\delta y_i+\frac{\hat{h}_{ij}}{\hat{s}_{ij}^{\,2}}\delta z_i \\ &+\frac{(x_j-x_i)}{\hat{s}_{ij}^{\,2}\tan\hat{\zeta}_{ij}}\delta x_j+\frac{(y_j-y_i)}{\hat{s}_{ij}^{\,2}\tan\hat{\zeta}_{ij}}\delta y_j-\frac{\hat{h}_{ij}}{\hat{s}_{ij}^{\,2}}\delta z_j=\left({}_o\zeta_{ij}-\hat{\zeta}_{ij}\right)+v \end{aligned} \tag{5.24}$$

In these equations, the terms ${}_os$, ${}_o\theta$, ${}_o\zeta$ represent observed values of slope distance, horizontal angle and zenith angle, respectively, and the terms $\hat{s},\hat{h},\hat{\theta},\hat{\zeta}$ represent the computed values (using the current Cartesian coordinates of the end points of the lines in question) of the respective parameters.

As there are usually many more observation equations than unknowns (*i.e.* station coordinates) in a survey network, it is necessary to make a least squares estimate of the station coordinates. Thus, the linearised observation equations are assembled into an overconstrained system of linear equations

$$\mathbf{Ax}=\mathbf{b}+\mathbf{v} \tag{5.25}$$

in which, by reference to equations (5.22), (5.23) and (5.24), it can be seen that the matrix **A** (of dimensions $m \times 3n$) contains the terms of the m linearised observation equations, the vector **x** contains the $3n$ increments to the approximate coordinate values of the n stations in the network, the vector **b** contains the m differences between the observed and the computed values, and the vector **v** contains the m residuals. The least squares solution seeks to minimise the function

$$\varphi = \mathbf{v}^T \mathbf{W} \mathbf{v} \tag{5.26}$$

where the diagonal matrix **W** contains the weights (*i.e.* reciprocal of the standard deviations) associated with the observations. Thus, in a manner identical to that presented in Chapter 3 for the solution of the DLT equations, the normal equations of the system are formed and solved to give a least squares estimate $\hat{\mathbf{x}}$ of the increments to the station coordinates:

$$\hat{\mathbf{x}} = \left(\mathbf{A}^T \cdot \mathbf{W} \cdot \mathbf{A}\right)^{-1} \mathbf{A}^T \cdot \mathbf{W} \cdot \mathbf{b} \tag{5.27}$$

These increments are then added to the current values of the approximate coordinates, and the process repeated until the increments are acceptably small.

Although this process minimises the errors within a network, it does not, as it stands, yield a unique solution for the station coordinates. For this, constraints that fix the network in space are required. These constraints are included as additional observation equations of the form

$$\mathbf{B}\mathbf{x} = \mathbf{0} \tag{5.28}$$

and, in general, augment the system of linearised observation equations to give

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix} \tag{5.29}$$

In the case of the triangular ground control network, the coordinates of one of the triangle vertices and the bearing of one of the sides of the triangle were assumed fixed. In the case of the marker network, the coordinates of the two theodolite stations were assumed fixed.

For a fixed bearing, the additional linearised observation equation takes the form

$$-(y_j - y_i)\delta x_i - (x_j - x_i)\delta y_i - (y_j - y_i)\delta x_j - (x_j - x_i)\delta y_j = 0 \tag{5.30}$$

and for each component of a fixed coordinate the linearised observation equations are

$$\delta x_i = 0, \quad \delta y_i = 0, \quad \delta z_i = 0 \tag{5.31}$$

Rather than include equations (5.31) in an augmented system, it is computationally more convenient to simply remove the relevant rows from **x**, **b** and **v** and the corresponding columns from **A**, and solve the reduced system

$$\mathbf{A}'\mathbf{x}' = \mathbf{b}' + \mathbf{v}' \tag{5.32}$$

Application of the method

In the triangular ground control networks used in this work, 15 observation equations (2 slope distances, 2 zenith angles and 1 horizontal angle at each of 3 theodolite stations) and 1 constraint equation (a fixed bearing) are generated to solve for 9 coordinate values (x, y, z_{ground} and $z_{theodolite}$ at each of two stations, and $z_{theodolite}$ at the third station, where the values of x, y, and z_{ground} are fixed). For the cable marker networks, a cable containing 2 markers generates 9 observation equations (1 horizontal angle and 1 vertical angle to each marker from each of two theodolite stations, and 1 pre-measured slope distance between the markers themselves) to solve for 6 coordinate values (x, y and z for each marker), and cables containing 3 and 4 markers generate 16 and 19 observation equations to solve for 9 and 12 coordinate values, respectively.

To assist in the computation of the marker coordinates, an integrated analysis system based on a series of four linked spreadsheets has been developed by the Author. Examples of the sheets are given in Figures 5.24 through to 5.29. The formulae underlying the entries in these sheets are not shown due to their verbosity. In any case, the linearised observation equations are presented above and the basic observation reduction formulae and intersection formulae may be found in any standard survey text (for example, Bannister and Raymond [12]). A brief description of the function of each sheet is now given.

(i) **Ground Control Data Reduction** (Figure 5.24)

The user enters the observed angles, distances and bearings relevant to the triangular ground control network. Horizontal and vertical angles observed on both faces of the theodolite are included, with subsequent reduction to included and mean angles. Visual comparison of computed mean angles and included angles offers the opportunity for detection of blunders (arising from instrument reading or booking). The unadjusted errors in the network are tabulated at the bottom of the sheet. The

approximate coordinates of the ground stations and theodolite axes are computed using standard intersection formulae [12].

(ii) **Ground Control Coordinate Adjustment** (Figure 5.25)

This sheet is in three parts: fundamental data and adjusted coordinates, intermediate computed values, and linearised observation equations.

The fundamental data takes the form of reduced observations of included horizontal angle, mean vertical angles and slope distances at each theodolite station, and approximate coordinates of the theodolite stations and ground markers. These data are taken automatically from the Ground Control Data Reduction sheet.

The adjusted coordinates part of the sheet indicates, at each stage of the iteration, the current values of the coordinates of each station and the adjusted values of the observations. The example shown indicates the results of the final iteration, with the maximum increment applied to a coordinate value being 0.14mm (x-coordinate of station A) and the maximum discrepancies between observed and adjusted values being 21 seconds of arc and 3mm for vertical angles and slope distances, respectively.

The intermediate computed values are used in computation of the observation equations. Each of these parameters is reused several times in the observation equations, and this justifies their explicit inclusion in the worksheet.

Not shown on this sheet is the system of normal equations resulting from the observation equations. Their inclusion is of no direct benefit to the user and, indeed, they are not explicitly formed on the sheet. Instead, the implicit worksheet functions dealing with matrix multiplication, transposition and inversion are simply concatenated to transparently construct and solve the system of equations.

(iii) **Marker Survey Data Reduction** (Figure 5.26)

This is similar to the Control Data Reduction sheet, in that the user enters raw observations onto this sheet. Data input by the user are horizontal and vertical angles (observed on both the left and right faces of the theodolite) from each of the left and right theodolite stations to all markers, together with measured marker spacings. The adjusted coordinates of these theodolite stations are taken automatically from the Ground Control Coordinate Adjustment sheet, and approximate coordinates of the markers computed using intersection formulae.

The errors in the unadjusted network are tabulated in the form of the difference between computed and observed marker spacings for each pair of collinear markers on the cables.

(iv) **Marker Coordinate Adjustment** (three separate sheets—Figures 5.27, 5.28 and 5.29)

This is similar in format and function to the Ground Control Coordinate Adjustment sheet. All of the fundamental data are taken automatically from the Ground Control Coordinate Adjustment and Marker Survey Data Reduction sheets. In this case these are horizontal and vertical angles to cable markers, and their approximate coordinates. The part containing the intermediate computed values is used heavily by the part containing the observation equations. This part is itself subdivided into sections relating to each of the cables installed on the rock face, as a system of normal equations is constructed and solved for each cable.

5.3.3 Summary

This Section has detailed a simple system appropriate for the introduction of high-precision photogrammetric control at a rock face.

The system developed makes use of lengths of thin steel wire rope, tensioned in front of the rock face. These wires are used to support markers, representing photogrammetric control points, in position in front of the rock face. The three-dimensional coordinates of these points are determined from a simple survey using a theodolite and steel tape measure, followed by a rigorous adjustment of the survey using advanced least squares estimation. The survey used to coordinate the control points is unconventional, using as it does both horizontal and vertical triangulation to previously trilaterated points, but coupled with a sophisticated least squares adjustment, it has also proved to be extremely accurate. It is also relatively fast, simple to conduct, does not require the operator to work close to potentially unsafe rock faces, and uses field equipment that is readily acquired and easily transported and used.

Ground Control Survey Data Reduction:
Observations shown in bold

	Horizontal Angles			*Vertical Angles*		*Distances*				*Control Triangle*		*Approximate coordinates*			
	Observed	Inc. Angle	Mean	**Observed**	Angle	**Observed**	comp. hrz.	comp. vrt.	ht. diff.	**Bearing**	sides	E	N	Z	Z(instrmnt)
At A															
Circle Left									B-C	A-B		A			
B	**096-27-10**	060-13-40	060-13-30	**097-03-10**	-007-01-40	**9.931**	9.856	-1.215	0.002	**284-00-00**	9.854	102.841	90.092	99.622	101.214
C	**156-40-50**			**096-44-50**	-006-42-55	**10.378**	10.307	-1.214							
Circle Right															
B	**276-27-10**	060-13-20		**262-59-50**											
C	**336-40-30**			**263-19-00**											
At B															
Circle Left									C-A	C-B		B			
C	**248-32-50**	062-06-40	062-06-45	**098-23-30**	-008-21-35	**10.228**	10.119	-1.487	-0.378	**225-00-00**	10.121	93.270	92.438	99.998	101.487
A	**310-39-30**			**100-45-10**	-010-43-20	**10.027**	9.852	-1.866							
Circle Right															
C	**068-32-30**	062-06-50		**261-40-20**											
A	**130-39-20**			**259-18-30**											
At C															
Circle Left									A-B	C-A		C			
A	**281-43-40**	057-40-10	057-40-15	**100-11-10**	-010-09-40	**10.473**	10.309	-1.848	0.375	**164-00-00**	10.308	100.000	100.000	100.000	101.470
B	**339-23-50**			**098-18-20**	-008-16-40	**10.230**	10.123	-1.473							
Circle Right															
A	**101-43-50**	057-40-20		**259-51-50**											
B	**159-24-10**			**261-45-00**											
	Angle sum:		*180-00-30*				*Height sum:*		*-0.002*	*Angle Sum:*	*180-00-00*				

Figure 5.24 Example Ground Control Data Reduction worksheet

Ground Control Coordinate Adjustment:

All input data is shown in 'Initial Coordinate Values' section and is taken from survey reduction sheet

Initial Coordinate Values														
	xA	**yA**	**zAg**	**zAt**	**xB**	**yB**	**zBg**	**zBt**	**xC**	**yC**	**zCg**	**zCt**	*'g' represents value at ground marker*	
	102.841	90.092	99.622	101.214	93.270	92.438	99.998	101.487	100.000	100.000	100.000	101.470	*'t' represents value at trunnion axis*	
Observations														
	angle CAB	**angle ABC**	**angle BCA**	**elev AtBg**	**elev AtCg**	**elev BtCg**	**elev BtAg**	**elev CtAg**	**elev CtBg**	**lgth AtBg**	**lgth AtCg**	**lgth BtCg**	**lgth BtAg**	**lgth CtAg**	**lgth CtBg**
	060-13-30	062-06-45	057-40-15	-007-01-40	-006-42-55	-008-21-35	-010-43-20	-010-09-40	-008-16-40	9.931	10.378	10.228	10.027	10.473	10.230
Adjusted Values														
	060-13-19	062-06-36	057-40-05	-007-01-59	-006-42-46	-008-21-24	-010-42-59	-010-09-57	-008-16-43	9.9286	10.3785	10.2307	10.0288	10.4717	10.2286
errors	000-00-11	000-00-09	000-00-10	-000-00-19	000-00-09	000-00-11	000-00-21	-000-00-17	-000-00-03	0.002	-0.001	-0.003	-0.002	0.001	0.001
Computed Coordinates														
	xA	**yB**	**zAg**	**zAt**	**xB**	**yB**	**zBg**	**zBt**	**xC**	**yC**	**zCg**	**zCt**		
last inc	3.18E-08	1.11E-07	-1.34E-07	-1.85E-09	-9.04E-09	1.72E-08	-2.62E-07	1.32E-07	0	0	0	-1.61E-07		
adj. vals	102.841	90.092	99.622	101.213	93.271	92.439	99.997	101.487	100.000	100.000	100.000	101.470		

Computed values											
	AtBg	**AtCg**	**BtCg**	**BtAg**	**CtAg**	**CtBg**	**angle CAB**	**angle ABC**	**angle BCA**		
diff. X	-9.570	-2.841	6.729	9.570	2.841	-6.729	1.0511	1.0840	1.0065	*radians*	
diff. Y	2.347	9.908	7.561	-2.347	-9.908	-7.561	60.2220	62.1099	57.6682	*decimal degrees*	
diff. Z	-1.216	-1.213	-1.487	-1.865	-1.848	-1.473	060-13-19	062-06-36	057-40-05	*deg-min-sec*	
H^2	97.09976	106.24153	102.45573	97.09976	106.24153	102.45573					
tan(zen.)	-8.10576	-8.49625	-6.80766	-5.28411	-5.57689	-6.87279					
zenith	1.69355	1.68796	1.71665	1.75783	1.74822	1.71528					
length	9.92863	10.37850	10.23066	10.02882	10.47175	10.22863					

Observation equations														
	d(xA)	**d(yA)**	**d(zAg)**	**d(zAt)**	**d(xB)**	**d(yB)**	**d(zBg)**	**d(zBt)**	**d(xC)**	**d(yC)**	**d(zCg)**	**d(zCt)**	***B***	***weights***
angle A	-0.06909	0.07182	0	0	-0.02417	-0.09856	0	0	0.09326	0.02674	0	0	0.00005	1.70E+09
angle B	-0.02417	-0.09856	0	0	0.09797	0.03288	0	0	-0.07380	0.06568	0	0	0.00005	1.70E+09
angle C	0.09326	0.02674	0	0	-0.07380	0.06568	0	0	-0.01946	-0.09242	0	0	0.00005	1.70E+09
elev AtBg	-0.01198	0.00294	0	0.09996	0.01198	-0.00294	-0.09996	0	0	0	0	0	-0.00009	4.73E+07
elev AtCg	-0.00310	0.01083	0	0.09569	0	0	0	0	0.00310	-0.01083	-0.09569	0	0.00004	4.73E+07
elev BtCg	0	0	0	0	0.00944	0.01061	0	0.09671	-0.00944	-0.01061	-0.09671	0	0.00005	4.73E+07
elev BtAg	-0.01801	0.00442	-0.09797	0	0.01801	-0.00442	0	0.09797	0	0	0	0	0.00010	4.73E+07
elev CtAg	-0.00465	0.01620	-0.09400	0	0	0	0	0	0.00465	-0.01620	0	0.09400	-0.00008	4.73E+07
elev CtBg	0	0	0	0	0.00936	0.01052	-0.09675	0	-0.00936	-0.01052	0	0.09675	-0.00001	4.73E+07
lgth AtBg	0.96392	-0.23634	0	0.12244	-0.96392	0.23634	-0.12244	0	0	0	0	0	0.00237	1.00E+06
lgth AtCg	0.27377	-0.95466	0	0.11689	0	0	0	0	-0.27377	0.95466	-0.11689	0	-0.00050	1.00E+06
lgth BtCg	0	0	0	0	-0.65774	-0.73909	0	0.14533	0.65774	0.73909	-0.14533	0	-0.00266	1.00E+06
lgth BtAg	0.95429	-0.23398	-0.18595	0	-0.95429	0.23398	0	0.18595	0	0	0	0	-0.00182	1.00E+06
lgth CtAg	0.27133	-0.94616	-0.17650	0	0	0	0	0	-0.27133	0.94616	0	0.17650	0.00125	1.00E+06
lgth CtBg	0	0	0	0	-0.65787	-0.73924	-0.14399	0	0.65787	0.73924	0	0.14399	0.00137	1.00E+06
brg CtAg	-9.90799	2.84134	0	0	0	0	0	0	9.90799	-2.84134	0	0	0	0

Figure 5.25 Example Ground Control Coordinate Adjustment worksheet

Marker Survey Data Reduction:

	Horizontal						*Vertical*						**Approximate Coordinates**				**Marker Spacings**			
Obs.	**Observations at left station**			**Observations at right station**			**Observations at left station**			**Observations at right station**				X	Y	Z		*comp.*	*obs.*	*error*
Station:	*Circle Left*	*Circle Right*	*Mean*	*Circle Left*	*Circle Right*	*Mean*	*Circle Left*	*Circle Right*	*Mean*	*Circle Left*	*Circle Right*	*Mean*	L	**102.841**	**90.092**	**101.213**				
													R	**93.271**	**92.439**	**101.487**				
R.O.	*096-27-10*	*276-27-10*		*310-39-30*	*130-39-20*															
1	*042-34-10*	*222-34-20*	053-52-55	*030-05-50*	*210-05-40*	079-26-20	*079-30-40*	*280-33-30*	010-31-25	*078-40-10*	*281-24-00*	011-21-55	1	92.657	81.515	103.686	1-2:	2.425	2.423	0.002
2	*040-54-50*	*220-55-00*	055-32-15	*033-01-30*	*213-01-20*	082-22-00	*072-50-10*	*287-13-40*	017-11-45	*070-46-20*	*289-18-00*	019-15-50	2	91.974	80.389	105.722	2-3:	2.551	2.550	0.001
3	*039-28-30*	*219-28-40*	056-58-35	*035-31-20*	*215-31-30*	084-52-00	*067-17-20*	*292-46-40*	022-44-40	*064-30-40*	*295-33-50*	025-31-35	3	91.261	79.218	107.873	3-4:	2.422	2.416	0.006
4	*038-19-20*	*218-19-20*	058-07-50	*037-28-50*	*217-29-00*	086-49-30	*063-05-50*	*296-58-10*	026-56-10	*059-58-30*	*300-05-50*	030-03-40	4	90.590	78.114	109.921				
7	*034-25-20*	*214-25-40*	062-01-40	*026-27-10*	*206-27-50*	075-48-05	*064-44-20*	*295-20-10*	025-17-55	*063-34-40*	*296-30-20*	026-27-50	7	93.366	79.476	107.939	7-8:	2.502	2.500	0.002
8	*034-21-10*	*214-21-00*	062-06-05	*021-24-20*	*201-24-20*	070-44-55	*069-29-40*	*290-35-00*	020-32-40	*069-22-20*	*290-42-10*	020-39-55	8	94.404	80.614	105.968	8-9:	2.462	2.460	0.002
9	*034-15-30*	*214-15-20*	062-11-45	*015-26-50*	*195-26-20*	064-47-10	*075-48-50*	*284-15-10*	014-13-10	*076-53-10*	*283-11-00*	013-08-55	9	95.435	81.744	104.038	9-10:	2.573	2.570	0.003
10	*034-07-20*	*214-07-10*	062-19-55	*008-01-20*	*188-01-20*	057-21-55	*085-09-00*	*274-55-40*	004-53-20	*086-57-00*	*273-07-20*	003-05-10	10	96.518	82.931	102.029				
11	*023-27-00*	*203-27-10*	073-00-05	*000-37-40*	*180-37-40*	049-58-15	*087-44-40*	*272-19-20*	002-17-20	*089-35-50*	*270-28-30*	000-26-20	11	98.239	82.365	101.573				
12	*013-09-00*	*193-09-20*	083-18-00	*357-12-50*	*177-12-50*	046-33-25	*078-40-20*	*281-24-40*	011-22-10	*082-53-20*	*277-11-00*	007-08-50	12	99.581	81.361	103.086	12-13:	2.236	2.233	0.003
13	*008-24-40*	*188-24-50*	088-02-25	*356-07-40*	*176-07-40*	045-28-15	*067-55-40*	*292-08-40*	022-06-30	*074-56-50*	*285-07-20*	015-05-15	13	100.214	80.769	105.148	13-14:	2.570	2.565	0.005
14	*003-28-10*	*183-28-10*	092-59-00	*355-03-10*	*175-03-10*	044-23-45	*058-11-40*	*301-47-00*	031-47-40	*067-30-30*	*292-34-00*	022-31-45	14	100.935	80.091	107.520				

Observations shown in italics
Data from control adjustment shown in bold

Figure 5.26 Example Marker Survey Data Reduction worksheet

Marker Coordinate Adjustment:

All input data taken from marker survey data reduction sheet

Fixed Coordinate Values

xL	yL	zLt	xR	yR	zRt	
102.841	90.092	101.213	93.271	92.439	101.487	*resulting from ground network adjustment*

Observations

	1	2	3	4	7	8	9	10	11	12	13	14	
angle L	053-52-55	055-32-15	056-58-35	058-07-50	062-01-40	062-06-05	062-11-45	062-19-55	073-00-05	083-18-00	088-02-25	092-59-00	*horizontal angle from L station*
angle R	079-26-20	082-22-00	084-52-00	086-49-30	075-48-05	070-44-55	064-47-10	057-21-55	049-58-15	046-33-25	045-28-15	044-23-45	*horizontal angle from R station*
elev L	010-31-25	017-11-45	022-44-40	026-56-10	025-17-55	020-32-40	014-13-10	004-53-20	002-17-20	011-22-10	022-06-30	031-47-40	*elevation from L station*
elev R	011-21-55	019-15-50	025-31-35	030-03-40	026-27-50	020-39-55	013-08-55	003-05-10	000-26-20	007-08-50	015-05-15	022-31-45	*elevation from R station*

	1-2	2-3	3-4	7-8	8-9	9-10	12-13	13-14	
spacings	2.423	2.550	2.416	2.500	2.460	2.570	2.233	2.565	*measured along cables before installation*

Initial Values of Marker Coordinates

X01	Y01	Z01	X02	Y02	Z02	X03	Y03	Z03	X04	Y04	Z04	X07	Y07	Z07	X08	Y08	Z08
92.657	81.515	103.686	91.974	80.389	105.722	91.261	79.218	107.873	90.590	78.114	109.921	93.366	79.476	107.939	94.404	80.614	105.968
X09	**Y09**	**Z09**	**X10**	**Y10**	**Z10**	**X11**	**Y11**	**Z11**	**X12**	**Y12**	**Z12**	**X13**	**Y13**	**Z13**	**X14**	**Y14**	**Z14**
95.435	81.744	104.038	96.518	82.931	102.029	98.239	82.365	101.573	99.581	81.361	103.086	100.214	80.769	105.148	100.935	80.091	107.520

Computed Coordinates

	X01	Y01	Z01	X02	Y02	Z02	X03	Y03	Z03	X04	Y04	Z04	X07	Y07	Z07	X08	Y08	Z08
last inc	-4.44E-07	-8.92E-07	6.30E-07	-3.79E-07	-1.00E-06	2.06E-07	-1.47E-07	-1.09E-07	7.81E-07	-1.45E-07	-2.61E-07	-9.71E-07	-1.43E-07	-9.30E-07	1.01E-07	-8.87E-08	-1.14E-06	1.65E-07
new vals	92.657	81.513	103.687	91.974	80.389	105.722	91.261	79.218	107.872	90.592	78.119	109.917	93.368	79.480	107.937	94.405	80.617	105.966
	X09	**Y09**	**Z09**	**X10**	**Y10**	**Z10**	**X11**	**Y11**	**Z11**	**X12**	**Y12**	**Z12**	**X13**	**Y13**	**Z13**	**X14**	**Y14**	**Z14**
last inc	-5.14E-07	-1.29E-06	8.75E-08	-4.63E-07	-7.40E-07	1.06E-06	2.44E-10	2.21E-10	-1.46E-11	1.24E-06	-8.58E-08	2.96E-06	2.07E-07	-1.50E-06	1.05E-06	-4.59E-07	-7.11E-07	-3.85E-06
new vals	95.435	81.745	104.038	96.517	82.929	102.030	98.239	82.365	101.573	99.581	81.359	103.088	100.214	80.770	105.147	100.935	80.099	107.515

Figure 5.27 Example worksheet Marker Coordinate Adjustment (sheet 1)

Marker Coordinate Adjustment:

Parameters determined from computed coordinate values on previous sheet, used in observation equations

Computed values													
From L	**1**	**2**	**3**	**4**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**R**
diff. X	-10.185	-10.867	-11.580	-12.249	-9.473	-8.436	-7.406	-6.324	-4.602	-3.260	-2.627	-1.907	-9.570
diff. Y	-8.579	-9.703	-10.874	-11.973	-10.612	-9.475	-8.347	-7.163	-7.727	-8.733	-9.322	-9.993	2.347
diff. Z	2.474	4.509	6.659	8.704	6.723	4.753	2.825	0.817	0.360	1.875	3.934	6.302	0.274
H^2	177.32929	212.24749	252.33279	293.40391	202.35899	160.94762	124.52736	91.29590	80.89089	86.90015	93.79533	103.49983	97.09976
tan(zen.)	5.38304	3.23108	2.38538	1.96798	2.11576	2.66905	3.95044	11.69384	25.01191	4.97240	2.46195	1.61427	36.00312
zen.	1.38712	1.27065	1.17383	1.10066	1.12927	1.21232	1.32287	1.48549	1.53084	1.37233	1.18497	1.01618	1.54303
bearing	3.84162	3.87045	3.89556	3.91559	3.98362	3.98494	3.98666	3.98911	4.17526	4.35511	4.43769	4.52386	2.90115
inc. angle	0.94048	0.96930	0.99441	1.01445	1.08247	1.08380	1.08551	1.08797	1.27411	1.45396	1.53655	1.62272	
length	13.544	15.251	17.224	19.214	15.734	13.548	11.511	9.590	9.001	9.509	10.453	11.967	9.858
From R	**1**	**2**	**3**	**4**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**L**
diff. X	-0.614	-1.297	-2.009	-2.679	0.097	1.134	2.164	3.247	4.969	6.310	6.943	7.664	9.570
diff. Y	-10.926	-12.050	-13.220	-14.320	-12.959	-11.822	-10.694	-9.509	-10.074	-11.080	-11.668	-12.340	-2.347
diff. Z	2.200	4.235	6.386	8.430	6.450	4.480	2.551	0.543	0.086	1.601	3.660	6.029	-0.274
H^2	119.74646	146.87471	178.81606	212.23299	167.93306	141.04230	119.04223	100.96738	126.17159	162.58418	184.35927	211.00495	97.09976
tan(zen.)	4.97383	2.86152	2.09411	1.72810	2.00920	2.65121	4.27684	18.49174	130.77939	7.96401	3.70971	2.40954	-36.00312
zen.	1.37239	1.23460	1.12529	1.04621	1.10898	1.21011	1.34111	1.51677	1.56315	1.44589	1.30749	1.17741	1.59856
bearing	4.65622	4.60517	4.56155	4.52745	4.71988	4.80805	4.91208	5.04139	5.17058	5.23010	5.24917	5.26815	6.04274
inc. angle	1.38652	1.43757	1.48119	1.51529	1.32286	1.23469	1.13066	1.00135	0.87216	0.81264	0.79357	0.77459	
length	11.162	12.838	14.819	16.832	14.475	12.693	11.205	10.063	11.233	12.851	14.063	15.727	9.858
spacings	**1-2**	**2-3**	**3-4**	**7-8**	**8-9**	**9-10**	**12-13**	**13-14**					
diff. X	-0.683	-0.713	-0.669	1.037	1.030	1.082	0.633	0.720					
diff. Y	-1.124	-1.171	-1.099	1.137	1.128	1.185	-0.588	-0.672					
diff. Z	2.035	2.150	2.045	-1.970	-1.928	-2.008	2.059	2.368					
length	2.423	2.550	2.416	2.500	2.460	2.570	2.233	2.565					

Observation equations	***Cable 1-2-3-4***													
	dX01	**dY01**	**dZ01**	**dX02**	**dY02**	**dZ02**	**dX03**	**dY03**	**dZ03**	**dX04**	**dY04**	**dZ04**	**B**	**W**
angle L-01	0.04838	-0.05743	0	0	0	0	0	0	0	0	0	0	-0.00006	4.25E+08
angle L-02	0	0	0	0.04572	-0.05120	0	0	0	0	0	0	0	0.00001	4.25E+08
angle L-03	0	0	0	0	0	0	0.04309	-0.04589	0	0	0	0	0.00002	4.25E+08
angle L-04	0	0	0	0	0	0	0	0	0	0.04081	-0.04175	0	0.00012	4.25E+08
angle R-01	-0.09124	0.00513	0	0	0	0	0	0	0	0	0	0	-0.00005	4.25E+08
angle R-02	0	0	0	-0.08204	0.00883	0	0	0	0	0	0	0	0.00000	4.25E+08
angle R-03	0	0	0	0	0	0	-0.07393	0.01124	0	0	0	0	0.00001	4.25E+08
angle R-04	0	0	0	0	0	0	0	0	0	-0.06747	0.01262	0	0.00009	4.25E+08
elev L-01	-0.01031	-0.00869	-0.07259	0	0	0	0	0	0	0	0	0	0.00000	4.25E+08
elev L-02	0	0	0	-0.01446	-0.01291	-0.06264	0	0	0	0	0	0	0.00002	4.25E+08
elev L-03	0	0	0	0	0	0	-0.01636	-0.01537	-0.05354	0	0	0	0.00000	4.25E+08
elev L-04	0	0	0	0	0	0	0	0	0	-0.01686	-0.01648	-0.04640	0.00001	4.25E+08
elev R-01	-0.00099	-0.01763	-0.08783	0	0	0	0	0	0	0	0	0	0.00005	4.25E+08
elev R-02	0	0	0	-0.00275	-0.02555	-0.07353	0	0	0	0	0	0	-0.00002	4.25E+08
elev R-03	0	0	0	0	0	0	-0.00437	-0.02875	-0.06090	0	0	0	-0.00001	4.25E+08
elev R-04	0	0	0	0	0	0	0	0	0	-0.00547	-0.02925	-0.05142	-0.00008	4.25E+08
dist 01-02	0.28170	0.46388	-0.83992	-0.28170	-0.46388	0.83992	0	0	0	0	0	0	-0.00002	1.00E+08
dist 02-03	0	0	0	0.27942	0.45912	-0.84328	-0.27942	-0.45912	0.84328	0	0	0	-0.00002	1.00E+08
dist 03-04	0	0	0	0	0	0	0.27709	0.45506	-0.84625	-0.27709	-0.45506	0.84625	-0.00002	1.00E+08

Figure 5.28 Example worksheet Marker Coordinate Adjustment (sheet 2)

Marker Coordinate Adjustment:

Observation equations: ***Cable 7-8-9-10***

	dX07	dY07	dZ07	dX08	dY08	dZ08	dX09	dY09	dZ09	dX10	dY10	dZ10	B	W
angle L-07	0.05244	-0.04681	0	0	0	0	0	0	0	0	0	0	0.00012	4.25E+08
angle L-08	0	0	0	0.05887	-0.05242	0	0	0	0	0	0	0	0.00008	4.25E+08
angle L-09	0	0	0	0	0	0	0.06703	-0.05947	0	0	0	0	0.00001	4.25E+08
angle L-10	0	0	0	0	0	0	0	0	0	0.07846	-0.06927	0	-0.00007	4.25E+08
angle R-07	-0.07716	-0.00058	0	0	0	0	0	0	0	0	0	0	0.00012	4.25E+08
angle R-08	0	0	0	-0.08382	-0.00804	0	0	0	0	0	0	0	0.00011	4.25E+08
angle R-09	0	0	0	0	0	0	-0.08983	-0.01818	0	0	0	0	0.00007	4.25E+08
angle R-10	0	0	0	0	0	0	0	0	0	-0.09418	-0.03215	0	-0.00013	4.25E+08
elev L-07	-0.01809	-0.02026	-0.05746	0	0	0	0	0	0	0	0	0	-0.00002	4.25E+08
elev L-08	0	0	0	-0.01722	-0.01934	-0.06912	0	0	0	0	0	0	-0.00009	4.25E+08
elev L-09	0	0	0	0	0	0	-0.01415	-0.01595	-0.08422	0	0	0	-0.00025	4.25E+08
elev L-10	0	0	0	0	0	0	0	0	0	-0.00588	-0.00666	-0.10390	-0.00002	4.25E+08
elev R-07	0.00023	-0.03078	-0.06185	0	0	0	0	0	0	0	0	0	-0.00007	4.25E+08
elev R-08	0	0	0	0.00266	-0.02768	-0.07372	0	0	0	0	0	0	0.00001	4.25E+08
elev R-09	0	0	0	0	0	0	0.00403	-0.01992	-0.08690	0	0	0	0.00020	4.25E+08
elev R-10	0	0	0	0	0	0	0	0	0	0.00173	-0.00508	-0.09923	0.00016	4.25E+08
dist 07-08	-0.41493	-0.45468	0.78810	0.41493	0.45468	-0.78810	0	0	0	0	0	0	-0.00003	1.00E+08
dist 08-09	0	0	0	-0.41867	-0.45852	0.78388	0.41867	0.45852	-0.78388	0	0	0	-0.00006	1.00E+08
dist 09-10	0	0	0	0	0	0	-0.42111	-0.46089	0.78119	0.42111	0.46089	-0.78119	-0.00008	1.00E+08

Observation equations: ***Marker 11***

	dX11	dY11	dZ11	B	W
angle L-11	0.09553	-0.05689	0	0.00000	4.25E+08
angle R-11	-0.07984	-0.03938	0	0.00000	4.25E+08
elev R-11	0.00030	-0.00061	-0.08902	-0.00001	4.25E+08
elev L-11	-0.00227	-0.00381	-0.11101	0.00001	4.25E+08

Observation equations: ***Cable 12-13-14***

	dX12	dY12	dZ12	dX13	dY13	dZ13	dX14	dY14	dZ14	B	W
angle L-12	0.10050	-0.03752	0	0	0	0	0	0	0	-0.00010	4.25E+08
angle L-13	0	0	0	0.09938	-0.02801	0	0	0	0	0.00005	4.25E+08
angle L-14	0	0	0	0	0	0	0.09655	-0.01842	0	0.00015	4.25E+08
angle R-12	-0.06815	-0.03881	0	0	0	0	0	0	0	-0.00007	4.25E+08
angle R-13	0	0	0	-0.06329	-0.03766	0	0	0	0	0.00005	4.25E+08
angle R-14	0	0	0	0	0	0	-0.05848	-0.03632	0	0.00027	4.25E+08
elev L-12	-0.00725	-0.01943	-0.10310	0	0	0	0	0	0	0.00003	4.25E+08
elev L-13	0	0	0	-0.00977	-0.03465	-0.08863	0	0	0	-0.00004	4.25E+08
elev L-14	0	0	0	0	0	0	-0.00825	-0.04322	-0.07103	-0.00030	4.25E+08
elev R-12	0.00480	-0.00842	-0.07721	0	0	0	0	0	0	0.00017	4.25E+08
elev R-13	0	0	0	0.00946	-0.01591	-0.06866	0	0	0	-0.00002	4.25E+08
elev R-14	0	0	0	0	0	0	0.01286	-0.02070	-0.05873	0.00018	4.25E+08
dist 12-13	-0.28353	0.26347	-0.92206	0.28353	-0.26347	0.92206	0	0	0	-0.00008	1.00E+08
dist 13-14	0	0	0	-0.28086	0.26180	-0.92335	0.28086	-0.26180	0.92335	-0.00006	1.00E+08

Figure 5.29 Example worksheet Marker Coordinate Adjustment (sheet 3)

Field trials conducted by the writer have demonstrated the efficacy of the technique, and the results presented in Chapter 7 bear testimony to this. However, although this system can be easily used to introduce high-precision photogrammetric control, unfortunately it does not produce control points that deviate substantially from a plane. This can be unsatisfactory when the control is to be used for the DLT, as ill-conditioned systems of equations will result. Enhancements to the customary solution of the DLT are therefore required, and these—together with those required in order to use the DLT with an analytical plotter—are described in the following Section.

5.4 Enhancements to the DLT Method of Photogrammetric Data Reduction

The DLT method, as described in Chapter 3, is usually used to reduce photograph coordinate data gathered using a monocomparator or stereocomparator. In this respect the method operates 'off-line', with the photogrammetric solution being determined some time after the coordinate measurement phase of the operation. Additionally, the customary formulation of the method also makes use of the normal equations of the system to solve the least squares problem.

In this work, the DLT is used in conjunction with an analytical plotter, and as such is required to operate in a real-time environment. Additionally, the photogrammetric control points that are available do not offer the spatial distribution required for a well-conditioned solution to the normal equations. This Section is therefore concerned with developments to overcome these deficiencies; the first sub-Section describes the changes required in order to use the DLT for control of an analytical plotter; and the second sub-Section is concerned with numerical techniques for avoiding problems of ill-conditioning.

5.4.1 Real-time implementation of the DLT

The DLT method, as described in Chapter 3, is usually used for 'off-line' photogrammetry, in which a prepared data file of photograph coordinates and object space coordinates of control points is analysed to determine the DLT parameters and interior and exterior orientation elements for each of the photographs in a stereopair. Subsequent entries in the data file, containing photograph coordinates corresponding to points of unknown location, are then analysed and the object space coordinates of the points computed.

In the application developed here, the DLT method is used in real time, to firstly determine the DLT parameters for each of photographs, and subsequently to both remove, in real time, y-parallax from the stereoscopic model as the photographs are moved on the stereocomparator and compute object space coordinates of unknown points. Consequently, the published form of the DLT method [94] is of little use here, and so a new and totally revised form has been produced by the writer to account for its new rôle.

Rather than operate in a straightforward sequential manner, the revised version is required to:

(i) be able to incorporate additional control points at any stage of the analysis, such that the benefits to an operator of the analytical plotter of even a rudimentary stereomodel can be made available as soon as possible, with subsequent refinement being made as additional control points are added to the solution;

(ii) remove y-parallax from the stereomodel, by computing the necessary y-coordinate of one photograph, given both the x- and y-coordinates of the other photograph and the current x-parallax; and

(iii) compute the object space coordinates of a point imaged on the two photographs whenever its photograph coordinates become available.

Requirements (i) and (iii) of this list have been met by carefully coding the DLT such that these specific functions are available to the control program of the analytical plotter as independent subroutines, able to share the necessary data. In order to meet requirement (ii), a new subroutine based on the four condition equations generated by two photographs has been written. These condition equations—presented as equations (3.27) and reproduced here for convenience (but without the terms for lens distortion correction)—are

$$\begin{bmatrix} \left(L_1^{\,l} - x^l L_9^{\,l}\right) & \left(L_2^{\,l} - x^l L_{10}^{\,l}\right) & \left(L_3^{\,l} - x^l L_{11}^{\,l}\right) \\ \left(L_5^{\,l} - y^l L_9^{\,l}\right) & \left(L_6^{\,l} - y^l L_{10}^{\,l}\right) & \left(L_7^{\,l} - y^l L_{11}^{\,l}\right) \\ \left(L_1^{\,r} - x^r L_9^{\,r}\right) & \left(L_2^{\,r} - x^r L_{10}^{\,r}\right) & \left(L_3^{\,r} - x^r L_{11}^{\,r}\right) \\ \left(L_5^{\,r} - y^r L_9^{\,r}\right) & \left(L_6^{\,r} - y^r L_{10}^{\,r}\right) & \left(L_7^{\,r} - y^r L_{11}^{\,r}\right) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -\left(L_4^{\,l} - x^l\right) \\ -\left(L_8^{\,l} - y^l\right) \\ -\left(L_4^{\,r} - x^r\right) \\ -\left(L_8^{\,r} - y^r\right) \end{bmatrix} \tag{5.33}$$

These equations are used to determine the y-parallax at a point in a stereomodel by treating photograph coordinate y^r as an unknown variable, and solving for it in terms of x^l, y^l and x^r, using the previously determined DLT coefficients of both photographs. This is a two stage process: firstly, the current values of x^l, y^l and x^r are used with equations

$$\begin{bmatrix} (L_1^l - x^l L_9^l) & (L_2^l - x^l L_{10}^l) & (L_3^l - x^l L_{11}^l) \\ (L_5^l - y^l L_9^l) & (L_6^l - y^l L_{10}^l) & (L_7^l - y^l L_{11}^l) \\ (L_1^r - x^r L_9^r) & (L_2^r - x^r L_{10}^r) & (L_3^r - x^r L_{11}^r) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -(L_4^l - x^l) \\ -(L_8^l - y^l) \\ -(L_4^r - x^r) \end{bmatrix} \tag{5.34}$$

to compute the object space coordinates of the point in space indicated by the illuminated spot of the analytical plotter; and secondly these object space coordinates are used to compute y^r through

$$y^r = \frac{L_5^r X + L_6^r Y + L_7^r Z + L_8^r}{(1 + L_9^r X + L_{10}^r Y + L_{11}^r Z)} \tag{5.35}$$

In the interests of program execution speed, the direct expanded form of equations (5.34) has been implemented in the revised program. The revised program has been coded in FORTRAN, and its operational details, together with the complete commented code, are presented in this volume at Appendix C.

Of course, the operation of this revised, real-time form of the DLT method is intimately bound up with the operation of the analytical plotter, as demonstrated both by the numbered points (i) to (iii) above and the discussion presented in Section 5.2, but care has been taken to ensure that the code is completely general, and it is therefore not specific to the particular instrument developed for this project.

5.4.2 Adaptation to control points of poor spatial distribution

It is well known that, when the underlying system of equations is ill-conditioned, round-off error will give rise to meaningless answers when an attempt is made to solve a least squares problem through the use of the normal equations [49]. Consequently, the DLT program developed during this project makes use of the numerically stable **singular value decomposition** (SVD) to solve the least squares problem, rather than the customary use of the normal equations.

Golub and vanLoan [49] show how, through the repeated application of Householder reflections (the process by which a reflection is applied to a matrix with the result that all but one element of a column is zeroed) and Givens rotations (the process by which a rotation is applied to a matrix with the result that selective off-diagonal terms are zeroed), any matrix **A** (dimensions $m \times n$) can be decomposed into two orthogonal matrices **U** and **V** ($m \times n$ and $n \times n$, respectively) and one diagonal matrix Σ ($n \times n$) such that

$$\mathbf{A} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T \tag{5.36}$$

The right hand side of (5.36) is the SVD of **A**. The Householder and Givens transformations are chosen because of their orthogonal nature—the multiplying factors are only either 0 or 1, with the result that round-off error is much reduced.

A system of non-singular linear equations can be written as,

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \tag{5.37}$$

and thus, as (5.36) clearly shows, the SVD permits the solution vector **x** to be simply computed as

$$\mathbf{x} = \mathbf{V} \cdot \left[\text{diag}\left(1/\sigma_j\right)\right] \cdot \left(\mathbf{U}^T \cdot \mathbf{b}\right) \tag{5.38}$$

where σ_j represents the *j*th element of the diagonal matrix Σ.

Similarly, for the case of an overdetermined system of equations (*i.e.* when $m>n$), the corresponding solution vector is computed as

$$\mathbf{x} = \sum_{i=1}^{m} \left(\frac{\mathbf{U}_i \cdot \mathbf{b}}{\sigma_i} \right) \cdot \mathbf{V}_i \tag{5.39}$$

which is the least squares estimate of **x** [49].

Furthermore, the elements of the diagonal matrix Σ may be used directly to compute the variance of any parameter a_j, an issue that is useful in the case of photogrammetry, given that variances are used extensively in the DLT solution. Thus, we have

$$\text{var}\left(a_j\right) = \sum_{i=1}^{m} \left(\frac{V_{ji}}{\sigma_i} \right)^2 \tag{5.40}$$

Computing the SVD is more expensive than solving the normal equations—a total of $3mn^2 + (23/3)n^3$ additional floating point operations are required—and more storage space is required, but these are trivial quantities when a typical sized photogrammetric problem of, say, m=20 and n=12 is under consideration. Moreover, the SVD does have advantages that definitely outweigh these possible disadvantages. As Golub and vanLoan prove, the errors that accumulate due to the finite precision arithmetic available on a computer are much less with the SVD than with the normal equations. Additionally, the SVD is able to deal with cases of column rank deficiency in **A** (*i.e.* not all of the parameters in the system of equations are independent), whereas solutions to

the normal equations are extremely sensitive to this issue. This is important, as the control points that can be provided on a rock face will unavoidably be close to coplanar, and this is a condition that leads to breakdown of the DLT method when its solution is based on the normal equations [1]. It is for these reasons that the SVD has been implemented in this version of the DLT method.

5.4.3 Summary

This Section has described two refinements to the customary DLT method. The first represents a completely new computer implementation of the method, such that it is suitable for use with—and as part of the control program for—an analytical plotter. The second involves the use of singular value decomposition—rather than the normal equations—for the solution of the least squares problem, such that inaccuracies due to computer round-off error are minimised. Incorporating these two refinements has required the production of a completely new and revised coded form of the DLT, and this is presented in Appendix C.

A discussion on the efficacy of the revised form of the DLT is presented in the following Section, where an accuracy test of the complete photogrammetric system is described.

5.5 Accuracy Test Of The Photogrammetric System

Before using the photogrammetric system in a routine fashion, an important investigation into the accuracy capabilities of the entire scheme was undertaken. The intention of these tests was to prove the complete system, rather than identify specific weaknesses within it, and hence a pragmatic approach to the testing was adopted. Thus, two stereophotographs resulting from the field work undertaken as part of this project were used for the test, and nine of the control points visible on them coordinated with the analytical plotter. The control points had previously been surveyed and adjusted using the procedures described in Section 5.3. A series of analyses was then undertaken with the modified DLT program, to both examine the efficacy of the analysis and determine with what accuracy points could be coordinated.

Within the general framework of analysing the accuracy with which the DLT model fitted the transformation from object space coordinates to photograph coordinates, the tests studied two particular aspects: what algorithmic parameters optimised the transformation; and what position errors

resulted following computation of object space coordinates from both measured photograph coordinates and photograph coordinates computed on the basis of a parallax-free stereomodel. The results of these tests are described in the following two sub-Sections, the first dealing with the operation of the DLT program, and the second with the coordination accuracy of the system.

5.5.1 Operation of the modified DLT program

Three general findings were made with respect to the operation of the modified DLT program, and these are described in the following points.

(i) Providing that realistic estimates are made of the variances of both the object space coordinates of the control points and the photograph coordinates, their specific values have little effect on the outcome of the analysis. Variances ranging between $0.00m^2$ and $0.01m^2$ for object space coordinates and $0.0006mm^2$ to $0.06mm^2$ (*i.e.* 1 to 10 square encoder counts) for photograph coordinates were tested. In all cases, the object space coordinates resulting from the computation never differed by more than a few millimetres.

(ii) Incorporation of asymmetric lens corrections, and symmetric corrections beyond the first term, rendered the DLT unstable. In order to incorporate these corrections, an initial solution was obtained using only the eleven basic DLT parameters, and from this solution initial estimates of the correction terms made. Further iteration was then undertaken to refine these estimates. In all of the cases tested, as the iteration for the evaluation of these terms proceeded, it was observed that the values of all of the DLT parameters oscillated around the values found from first (uncorrected) pass, and rapidly diverged to meaningless values. As a consequence of this, it was found impossible to determine at what stage of the iterative procedure optimal correction factors were obtained. Although an arbitrary iteration count can be imposed, the results are then worthless, and so the use of higher correction terms with the DLT solution must be abandoned.

(iii) When the first symmetric correction term is incorporated into the solution, convergence (measured as the reduction, to a threshold value of 0.0005, of the difference between successive values of χ^2 for the equation fitting process) occurs after about 6 cycles.

The interesting findings of item (ii) above must be compared with the descriptions of the DLT in the literature. In all cases, the general statement is

made that only the first symmetric correction term is effective in improving the solution, but no supporting reasoning given. On the basis of the tests undertaken here, it would appear that the reason is simply that the higher correction terms lead to an unstable numerical scheme, a state of affairs that probably arises out of the approximate linearised form of the collinearity equations used in the derivation of the DLT equations.

5.5.2 Coordination accuracy of the photogrammetric system

To test the coordination accuracy of the system, nine control points, visible on each of two stereophotographs, were coordinated using the analytical plotter and then analysed using the modified DLT program. Both the object space coordinates and the measured photograph coordinates of the control points are presented in Table 5.03. The order of the points given in this and the subsequent Tables is the order in which the points where observed. This order was selected simply on the basis of convenience—having set the photographs on the analytical plotter, point 18 was found to be the nearest to the illuminated spot and hence was coordinated first; the others followed as a path was taken around the photographs.

The object space coordinates were determined on the basis of a survey and subsequent adjustment as described in Section 5.3. The measured photograph coordinates are referenced to arbitrary local coordinate systems (one for each photograph), as the design of the analytical plotter is such that fixed coordinate systems are neither required nor implemented.

A series of four tests was undertaken using these data, incorporating differing degrees of lens distortion correction. These were: no corrections; first symmetrical term; first asymmetrical term; and both first symmetrical and

Table 5.03 Object space coordinates and measured photograph coordinates of the test control points.

point number	object space coordinates (m)			photograph coordinates (mm)			
	X	Y	Z	*x* left	*y* left	*x* right	*y* right
18	104.894	61.448	103.544	0.559	0.272	0.290	0.318
4	97.160	60.625	109.308	-93.940	60.331	-110.675	62.087
3	97.197	61.090	106.937	-99.708	37.722	-110.638	37.714
16	99.541	60.217	102.255	-72.011	-22.993	-76.832	-23.562
28	102.110	60.864	100.249	-40.004	-51.742	-37.036	-50.623
27	102.112	60.460	104.597	-37.452	8.954	-44.233	9.190
15	99.431	60.016	105.016	-69.952	11.411	-80.272	11.411
20	105.135	59.119	111.138	-0.010	74.269	-21.072	76.429
26	102.112	60.082	109.003	-35.281	57.897	-50.686	58.955

asymmetrical terms of lens distortion. The number of DLT parameters solved for in each of these cases was 11, 12, 13 and 14 respectively, and in each case a total of 18 condition equations was assembled (*i.e.* two equations for each control point).

Table 5.04 shows the computed object space coordinates resulting from the application of each of these DLT models, using the measured photograph coordinates as input data. This Table shows, *via* the root mean square (RMS) position error (computed as the linear distance between computed and observed control point coordinates), the dramatic effect of introducing the lens distortion corrections to the solution. The minimum RMS error is obtained when the first correction term for symmetric distortion is used, and the maximum is obtained when the correction term for asymmetric distortion is used. However, as was noted in the previous sub-Section, the numerical instability induced when the asymmetric correction terms are included means that the poor results obtained in this case are to be expected.

If we study the results shown in the top part of Table 5.04, *i.e.* those obtained either with no correction or the first correction term for symmetric

Table 5.04 Control point object space coordinates, computed using measured photograph coordinates.

point	no correction				1st symmetrical term			
number	X	Y	Z	error	X	Y	Z	error
18	104.880	61.475	103.541	0.031	104.883	61.477	103.545	0.031
4	97.156	60.604	109.314	0.022	97.160	60.618	109.315	0.010
3	97.192	61.052	106.954	0.042	97.196	61.084	106.936	0.006
16	99.538	60.222	102.256	0.006	99.539	60.230	102.261	0.014
28	102.117	60.810	100.246	0.054	102.113	60.844	100.244	0.021
27	102.129	60.457	104.583	0.022	102.122	60.427	104.593	0.035
15	99.430	60.052	105.012	0.036	99.432	60.033	105.014	0.017
20	105.168	59.031	111.191	0.108	105.139	59.109	111.144	0.012
26	102.064	60.292	108.901	0.239	102.101	60.119	108.985	0.043
	RMS position error (m):			0.092	RMS position error (m):			0.024
point	1st asymmetrical term				1st symmetrical & asymmetrical terms			
number	X	Y	Z	error	X	Y	Z	error
18	104.888	61.485	103.539	0.038	104.879	61.546	103.519	0.103
4	97.174	60.706	109.263	0.093	97.168	60.692	109.273	0.076
3	97.202	61.183	106.897	0.102	97.204	61.145	106.909	0.062
16	99.540	60.241	102.248	0.025	99.544	60.263	102.252	0.047
28	102.126	60.821	100.246	0.046	102.121	60.893	100.234	0.034
27	102.136	60.419	104.594	0.048	102.126	60.450	104.587	0.020
15	99.428	60.093	105.000	0.079	99.432	60.045	105.011	0.029
20	105.002	59.495	110.898	0.466	105.104	59.254	111.051	0.164
26	102.047	60.405	108.842	0.367	102.1	60.158	108.967	0.085
	RMS position error (m):			0.206	RMS position error (m):			0.081

distortion only, we can see that the inclusion of the correction terms reduces the position error in all cases, with the exception of points 18, 16 and 27. Nevertheless, the increases in these cases are modest—ranging from 0mm to 13mm—and must be set against the magnitudes of the reductions in position error, the maximum of which is 196mm for point 26. The overall reduction in RMS. position error, from 92mm to 24mm, shows the benefit of including the first symmetric correction term, and also demonstrates that the system is extremely capable of coordinating points to an accuracy acceptable for the measurement of rock mass geometry.

It is interesting to examine the effect these four degrees of distortion correction have on the computed interior and exterior orientation parameters. Table 5.05 shows the values corresponding to the left and right photographs for each of the four types of correction used. The most striking effects are to be seen in the coordinates of the photograph principal point, shown in the Table as x_p and y_p. The principal point is that location where all of the lens distortions are zero and, as the Table shows, if we take the coordinates obtained for the no-correction case as a datum, this point moves by between 9mm to 16mm. It is not surprising, therefore, that the incorporation of these corrections has such an effect on the solution. No pattern to the changes in any of these parameters can be discerned from the Table, which demonstrates that these data are insufficient *per se* to determine an optimal algorithmic configuration for the solution.

Apart from the computation of object space coordinates for points on the surface of the rock face, the other use of the solution to the DLT is to generate a

Table 5.05 Interior and exterior camera orientation parameters for the two camera stations used in the coordination tests.

	no correction		1st symmetrical term		1st asymmetrical term		1st symmetrical & asymmetrical terms	
	left photo	*right photo*	*left photo*	*right photo*	*left photo*	*right photo*	*left photo*	*right photo*
x_p (mm)	-27.016	-90.543	-20.718	-97.353	-26.730	-97.650	-24.420	-98.317
y_p (mm)	9.499	9.422	10.562	10.391	-4.600	8.790	5.711	13.813
c (mm)	190.894	192.801	194.339	201.251	189.898	199.971	198.087	201.971
ω	71.5°	75.1°	71.5°	74.4°	75.6°	74.8°	72.6°	73.5°
ϕ	7.0°	1.4°	5.9°	3.2°	6.0°	3.0°	6.8°	3.5°
κ	7.4°	3.0°	6.3°	4.2°	6.1°	3.9°	7.1°	4.5°
X (m)	104.643	98.994	104.828	98.970	104.403	98.883	104.805	99.011
Y (m)	73.813	74.734	73.968	75.303	73.833	75.223	74.197	75.372
Z (m)	100.125	100.991	100.149	100.729	100.080	100.737	99.987	100.729

parallax-free stereomodel. Although the implementation of controlled removal of y-parallax is essential for general stereoscopic viewing, if the computed y-parallax is not exactly equal to the actual value at a given position within the stereomodel, then use of the floating spot will be impossible. This is because the human brain will take the visual cues associated with the majority of the scene, *i.e.* the photographs, as being without y-parallax with the result that two floating spots will be perceived. It is essential, therefore, to determine the amount of y-parallax that will remain in the stereomodel after application of the DLT to remove it.

To evaluate the error in y-parallax, equations (5.34) and (5.35) have been applied to the observed values of the photograph coordinates representing x_l, y_l, and x_r in order to compute the value of y_r according to the DLT model. The computed values are shown in Table 5.06, together with both the observed values and the resulting error in y-parallax. However, these results conflict with those of control point coordination presented above. Here, we that although the application of the asymmetrical corrections is once more of no benefit, it appears that not applying the first correction term for symmetric distortion produces smaller absolute errors in y-parallax. Although the magnitude of these errors is small—with no corrections the maximum is 0.3mm, and with the first correction term for symmetric distortion the maximum is 0.6mm—use of the analytical plotter has demonstrated that they are sufficient to prevent perception of a single floating spot. Consequently, in the operation of the analytical plotter, the automatic removal of y-parallax used during general movement operations, with the user being given the option of manual control when coordination is being undertaken.

Table 5.06 Error in y-parallax determination.

point number	y (right) observed (mm)	no correction		1st symmetrical term		1st asymmetrical term		1st symmetrical & asymmetrical terms	
		computed (mm)	error (mm)	computed (mm)	error (mm)	computed (mm)	error (mm)	computed (mm)	error (mm)
18	0.318	0.331	-0.013	0.398	-0.080	0.494	-0.176	0.235	0.083
4	62.087	62.064	0.023	61.470	0.617	60.486	1.601	60.906	1.181
3	37.714	37.637	0.077	37.529	0.185	36.856	0.858	37.200	0.514
16	-23.562	-23.445	-0.117	-23.138	-0.424	-23.934	0.372	-23.551	-0.011
28	-50.623	-50.645	0.022	-50.683	0.060	-52.146	1.523	-51.617	0.994
27	9.190	9.057	0.133	9.072	0.118	9.030	0.160	8.965	0.225
15	11.411	11.360	0.051	11.404	0.007	11.160	0.251	11.287	0.124
20	76.429	76.323	0.106	76.707	-0.278	75.820	0.609	75.702	0.727
26	58.955	59.235	-0.280	59.150	-0.195	58.567	0.388	58.596	0.359

Notwithstanding the difficulties associated with the perception of a single floating spot when a small amount of y-parallax is present, it is useful to consider what position errors would result if the computed, rather than the observed, value of y_r was to be used for coordination purposes. The motivation for this is that, should satisfactory results be obtained, then the difficulties associated with the perception of a single floating spot may be secondary to the simplified and faster use made possible when computed values of y_r are adopted.

The results of this analysis are presented in Table 5.07. Interestingly, the smallest errors result when no lens distortion corrections are applied, with those resulting from the application of the first term of symmetrical lens distortion being about three times as large. The reason for this, and the general difficulty of minimising the error in y-parallax when lens distortion correction is applied, is the impossibility of determining the corrections required for the

Table 5.07 Error in position determination resulting from the use of photograph coordinates computed using a parallax-free model.

	no correction					1st symmetrical term				
	y (right)	computed object space coordinates				y (right)	computed object space coordinates			
point number	computed (mm)	X (m)	Y (m)	Z (m)	error (m)	computed (mm)	X (m)	Y (m)	Z (m)	error (m)
18	0.331	104.880	61.475	103.542	0.030	0.398	104.886	61.469	103.550	0.023
4	62.064	97.158	60.617	109.304	0.009	61.470	97.184	60.806	109.158	0.236
3	37.637	97.196	61.081	106.937	0.009	37.529	97.203	61.131	106.906	0.052
16	-23.445	99.539	60.207	102.263	0.013	-23.138	99.540	60.198	102.282	0.033
28	-50.645	102.116	60.811	100.246	0.053	-50.683	102.112	60.846	100.241	0.020
27	9.057	102.121	60.490	104.568	0.043	9.072	102.118	60.448	104.582	0.020
15	11.360	99.429	60.076	105.002	0.062	11.404	99.432	60.036	105.013	0.020
20	76.323	105.136	59.108	111.132	0.012	76.707	105.193	58.977	111.255	0.193
26	59.235	102.116	60.054	109.053	0.057	59.150	102.123	60.015	109.056	0.086
	RMS position error (m): 0.038					RMS position error (m): 0.108				

	1st asymmetrical term					1st symmetrical & asymmetrical terms				
	y (right)	computed object space coordinates				y (right)	computed object space coordinates			
point number	computed (mm)	X (m)	Y (m)	Z (m)	error (m)	computed (mm)	X (m)	Y (m)	Z (m)	error (m)
18	0.494	104.896	61.469	103.553	0.023	0.235	104.877	61.551	103.513	0.109
4	60.486	97.256	61.401	108.726	0.974	60.906	97.223	61.129	108.922	0.638
3	36.856	97.230	61.413	106.733	0.383	37.200	97.222	61.283	106.811	0.231
16	-23.934	99.538	60.279	102.223	0.070	-23.551	99.544	60.262	102.253	0.045
28	-52.146	102.111	60.846	100.153	0.098	-51.617	102.106	60.945	100.186	0.103
27	9.030	102.125	60.467	104.571	0.029	8.965	102.116	60.490	104.562	0.047
15	11.160	99.422	60.246	104.940	0.243	11.287	99.430	60.109	104.985	0.098
20	75.820	104.774	60.067	110.475	1.212	75.702	104.839	59.935	110.550	1.049
26	58.567	101.986	60.682	108.661	0.702	58.596	102.046	60.411	108.803	0.391
	RMS position error (m): 0.590					RMS position error (m): 0.441				

photograph whose y-coordinate is unknown. This is because both the x- and y-coordinate values are required in order to compute the symmetric lens distortion term. Thus, DLT coefficients associated with a model that incorporates lens distortion corrections are applied to uncorrected coordinate values, with the result that large errors are generated. We can conclude, therefore, that unless some form of iterative procedure can be developed and adopted, the use of computed y-parallax to determine photograph coordinates is not a viable proposition.

5.5.3 Summary

In this Section, the results of a series of tests undertaken to examine the accuracy of the photogrammetric system have been presented. They show how only DLT models incorporating either no lens distortion corrections, or only the first symmetric correction term, are numerically stable. Similarly, the results of the DLT analysis have been found insensitive to realistic values of photograph coordinate and object space coordinate variances.

Most importantly, it has been demonstrated that, when the first correction term for symmetric lens distortion is incorporated into the DLT solution, the photogrammetric system is capable of determining the coordinates of points in space such that root mean square errors of position of around 25mm are achievable. Hence, the system may be regarded as being suitable for the measurement of rock mass geometry.

A study of the removal of y-parallax through computation has revealed that, although its routine removal in order to produce a parallax-free model is practicable, the errors are too large to allow satisfactory computation of object space coordinates. As a consequence, in the operation of the analytical plotter automatic removal of y-parallax has been limited to operations involving general photograph positioning, with manual removal of y-parallax being adopted for coordination procedures.

5.6 Synopsis

In this Chapter, five aspects associated with the development of a non-metric photogrammetric system appropriate to the analysis of rock mass geometry have been discussed, and it is useful to summarise the findings arising from these discussions in a series of numbered points.

(i) A numerical study of the effect of errors in the measurement of photograph coordinates on the computed orientation of a discontinuity surface has led to a number of conclusions regarding the photogrammetric configuration and the accuracy with which photograph coordinates must be determined. This work has shown that, with the photogrammetric configurations that are possible in front of a rock face, the orientation of discontinuities larger than 0.5m in extent can be determined with an error of less than 1°, providing photograph coordinates can be measured with a resolution of around 25μm.

(ii) Using this result as part of a general specification, a novel high-performance analytical plotter has been designed and constructed. Constructed from commercially available linear bearings, leadscrews and stepping motors, this instrument is able to determine photograph coordinates to a resolution of 6.35μm. The optical system of the instrument consists of a standard mirror stereoscope, with an IBM-PC compatible microcomputer being used as the host for the control software. Modern developments in integrated circuit technology have enabled a closed-loop control system for the instrument to be implemented on an interface board that connects directly with the system board of the microcomputer.

(iii) A novel method for easily introducing high-precision photogrammetric control at a rock face has been developed. This method makes use of simple field equipment and techniques (*i.e.* thin steel wire rope, moderate precision theodolite, a simple survey), coupled with a sophisticated least squares data reduction scheme implemented on a microcomputer spreadsheet program, to determine the object space coordinates of a number of markers positioned in front of the rock face.

(iv) An enhanced form of the DLT method has been described. This completely new implementation of the DLT is able to both operate in real time with an analytical plotter and, through the inclusion of singular value decomposition as a least squares solver, use photogrammetric control points of non-optimal configuration.

(v) Finally, the results of an accuracy test of the complete photogrammetric system have been described. These show how, when the DLT is implemented with the first term of symmetric lens distortion included, it is possible to determine the position in space of a point to within 25mm of its correct position. The use of other corrections for lens distortion has

been found to lead to numerical instability, and hence they cannot be implemented. The automatic removal of y-parallax has also been studied, and it has been found that whilst the errors are sufficiently small to allow generation of a parallax-free stereomodel during routine photograph positioning procedures, manual removal of y-parallax is required when coordination is being undertaken.

The result of the work presented in this Chapter is a new and highly integrated system for photogrammetric surveying of rock faces. The outcome of field tests of the new system are presented in Chapter 7, where it can be seen that the system permits accurate surveying of rock mass geometry. However, this system only allows geometrical data associated with rock masses to be acquired—a method of analysing this data is also required. Thus, an enhanced scheme for the analysis of discontinuity orientation, based on the fuzzy partitioning algorithms described in Chapter 3, has been developed by the Author, and this is described in detail in Chapter 6, following.

6 Analysis of Rock Mass Geometry Using Fuzzy Partitioning Algorithms

The review of clustering techniques presented in Chapter 4 concluded that it may be possible to use partitioning algorithms based on fuzzy objective functions to form the kernel of an efficacious scheme for the analysis and classification of rock mass geometry.

In this Chapter, a novel scheme is described in which a sequential application of the fuzzy *c*-means (FCM), fuzzy covariance (FCV) and fuzzy maximum likelihood estimation (FMLE) algorithms are used to optimally partition discontinuity orientation data, and thus discover the underlying nature of the rock mass geometry. This scheme has been developed and produced as a computer program—FADD (**F**uzzy **A**nalysis of **D**iscontinuity **D**ata)—by this Author, the FORTRAN code for which is reproduced in Appendix D.

This Chapter consists of five Sections. The first of these, Section 6.1, discusses both the geometry of, and distance measures available for, orientation data. This leads to a description of how fuzzy objective functions may be applied to the analysis of such data. The development of the scheme, in terms of the detailed implementation of the FCM, FCV and FMLE algorithms, is discussed in Section 6.2. This discussion is centred on the tactics adopted for ensuring the scheme correctly and efficiently initialises the matrix of membership values, computes centroid vectors, and accounts for antipodal vectors, together with modified formulations of various cluster validity functions. Examples are presented in Section 6.3 that demonstrate how the scheme correctly identifies clusters within sets of test data. In this same Section, a study of algorithm termination criteria and cluster validity functions is presented, together with recommendations for their use in the context of analysing discontinuity orientation data. Section 6.4 considers how the results of the analysis may be presented, in a form that is readily assimilable, using the hemispherical projection. The result is a totally new method of visually presenting clusters in orientation data—*i.e.* discontinuity sets—that is

impressive in its clarity. As a demonstration, the scheme is applied to the analysis of the copper porphyry data of Mahtab *et alia* [86], and this clearly demonstrates the enhanced nature of the analysis over both customary and previously proposed techniques. The final Section of the Chapter contains a synopsis of the findings.

6.1 Application of Fuzzy Objective Function Algorithms to the Analysis of Discontinuity Orientation Data

In order to successfully apply a partitioning algorithm to the analysis of orientation data, it is necessary to consider in some detail the geometry of the data itself, and then how similarity (or its inverse, dissimilarity—usually regarded as 'distance') between any two data, or a datum and a cluster centroid, may be quantified. These two issues are discussed in the following two sub-Sections. Whilst considering these issues, it is also important to bear in mind the weaknesses associated with individual partitioning algorithms (as highlighted in Chapter 4), if these are to be accounted for.

6.1.1 Geometry of orientation data

Determining how to formulate a partitioning algorithm for discontinuity orientation data is made difficult because of the dichotomy that exists in terms of whether the data are of two- or three-dimensional form. For example, discontinuity orientations are customarily measured in the field using a two-dimensional coordinate system—either dip angle and dip direction, or dip angle and strike direction—whereas mechanical or geometrical analysis involving an oriented object is customarily undertaken using a three-dimensional Cartesian coordinate system, with the orientations being described in terms of the Cartesian components of the corresponding vector. It is important to determine which—if, indeed, either—of these systems is correct and should be adopted as the basis for a partitioning algorithm. To do so requires a consideration of the geometry of both the three-dimensional space we are normally aware of, and of the geometry of the surface of the sphere. These are Euclidean and Riemannian geometries, respectively.

Orientation data in Euclidean space

If we consider a three-dimensional Cartesian system within a Euclidean space, it is easy to show that the three components of an orientation vector are not

independent. Orientation vectors are usually considered to emanate from the origin of the Cartesian coordinate system, with the added constraint that they are of equal magnitude (because we have no notion of the 'magnitude of an orientation'; *i.e.* we do not regard certain orientations as 'bigger' than others). Arbitrarily, these vectors are assigned unit magnitude, with the result that the locus of the tips of the vectors forms a unit sphere. Now, the magnitude of a vector **a** in three-dimensional Euclidean space is computed using the Pythagorean Theorem, *i.e.*

$$|\mathbf{a}| = \sqrt{a_x^{\,2} + a_y^{\,2} + a_z^{\,2}} \tag{6.01}$$

where a_x, a_y and a_z represent the Cartesian components of the vector. In the case of orientation vectors, the magnitude is constrained to equal unity, with the result that only two of the three components are independent. Thus, on the basis of this dependence, we can say that the use of a three-dimensional coordinate system for orientation data involves redundancy, and so a two-dimensional coordinate system is appropriate.

However, the coordinate system will be used—in the present context—primarily to determine dissimilarity between vectors, and this can be thought of as the distance between points on the surface of the unit sphere. Such distances are, in fact, the lengths of curves on the surface of the sphere, and the analysis of curves in Euclidean space is performed using **differential geometry**. More particularly, in the case of a finite curved surface such as a sphere, **Riemannian geometry** is applicable.

Thus, we are led to the result that the analysis of orientation data should take place using a two-dimensional coordinate system in a Riemannian space. It is therefore necessary to consider how this may be undertaken.

Orientation data in Riemannian space

In Riemannian geometry [74], we find that a surface may be represented by the vector function $\mathbf{x}(u^1, u^2)$ where u^1 and u^2 are parametric functions of axis 1 and axis 2, respectively, of some two-dimensional coordinate system. An arc on this surface, with end points a and b, may be represented by the parametric vector function

$$u^1 = u^1(t), \quad u^2 = u^2(t) \qquad (a \le t \le b) \tag{6.02}$$

and on this basis the length of an arc, l, between two points a and b is given by

$$l = \int_a^b \sqrt{\left(\sum_{j=1}^{2} \sum_{k=1}^{2} g_{jk} \dot{u}^j \dot{u}^k \right)} \, dt \tag{6.03}$$

where $g_{jk} = \mathbf{x}_j \cdot \mathbf{x}_k$, the scalar product of two vectors, and the overdots in (6.03) represent differentiation with respect to t. Thus, with the adoption of a coordinate system u^1u^2, it is possible to use a two-dimensional representation on the surface of the sphere in terms of Riemannian geometry.

Hence, we can see how the customary representation of orientation data using two-dimensional spherical polar coordinates—say, dip angle and dip direction, or dip angle and strike direction—may be used. Under these circumstances, we have

$$\mathbf{x}(u^1, u^2) = (r \cos u^2 \cos u^1, r \cos u^2 \sin u^1, r \sin u^2) \tag{6.04}$$

where, for the unit sphere r=1 and the angles u^1 and u^2 are defined in Figure 6.01.

Unfortunately, this coordinate system is singular at certain points, with the result that only one coordinate value is necessary to identify a vector at these points. In the case of the customary geological polar coordinates dip angle and dip direction, the singularity exists for vectors possessing a dip angle of 90°, as azimuthal information is not required with vertical vectors. Consequently, some other coordinate system on the sphere is required; one that does not possess singularities.

The coordinate system to be adopted should also fulfil criteria associated with the computational implementation of the scheme. Bearing in mind that the purpose of the work described here is to develop a useful tool for the analysis of discontinuity geometry, and considering that such analysis may involve computation associated with many tens, if not hundreds, of orientation vectors, then solutions to (6.03) are required which may be rapidly evaluated on a computer, if computation time is not to render the resulting computer program useless for all but trivial numbers of vectors. This implies that, as the differentiation and integration in (6.03) are both performed with respect to the parameter t, a variable that indicates position along an arc, closed-form solutions to this calculus must be found if (relatively slow) numerical evaluation in a piecewise manner is to be avoided. The question to be asked then is: can a two-dimensional representation of the sphere be found which

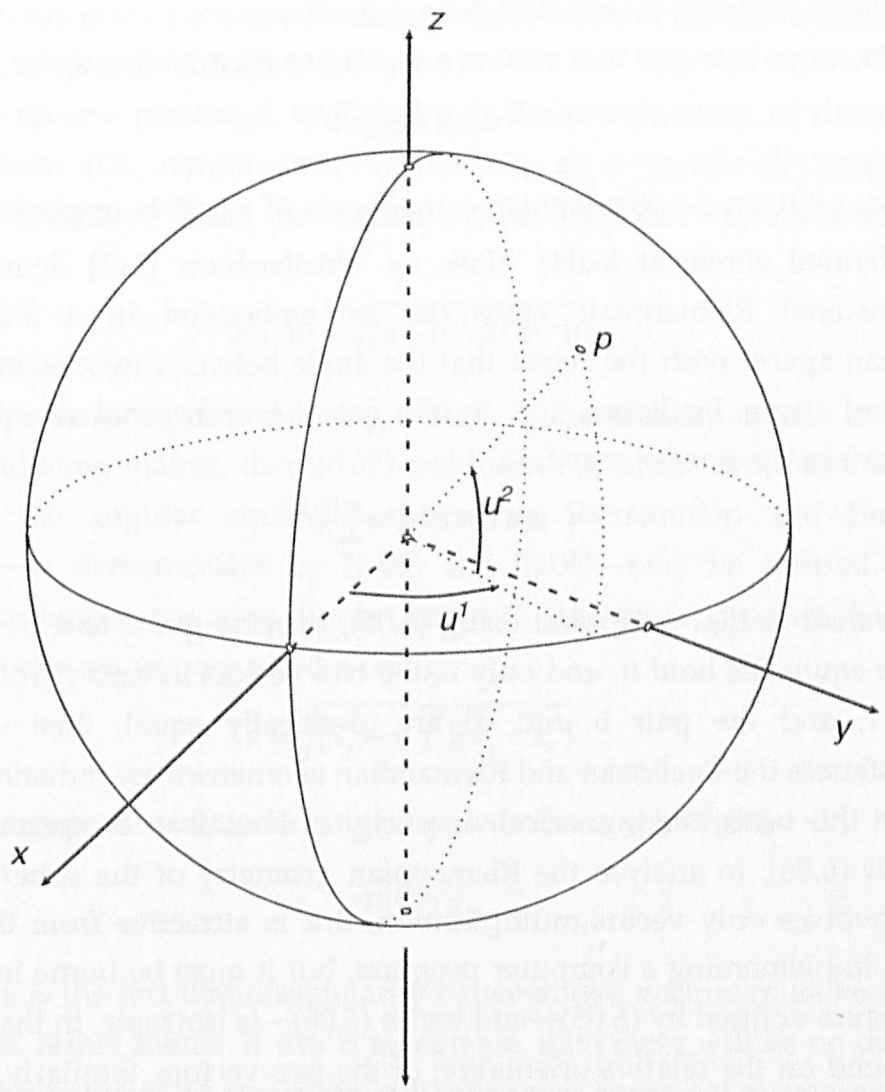


Figure 6.01 Spherical polar coordinates in Riemannian geometry.

contains no singularities, which is both integrable and differentiable, and does not involve numerically expensive functions?

Rather than answer this question directly, it is possible to make use of the connection between Euclidean and Riemannian geometries to find a solution to the problem that is appropriate for numerical analysis.

Connection between Euclidean and Riemannian geometries

By adopting a pragmatic approach to the problem of computing distances in Riemannian space, we can avoid the search for a coordinate system that fulfils the criteria outlined above. If we decide to use the angle subtended by two vectors—rather than arc distance—as a measure of dissimilarity, this approach

allows the Riemannian geometry of the sphere to be connected with the numerically straightforward Euclidean geometry.

The angle between two vectors **a** and **b** in a Riemannian space is given by

$$\cos\theta = g_{jk}a^j b^k \tag{6.05}$$

where a and b are the Riemannian components of **a** and **b**, respectively and g_{jk} is as defined above at (6.03). Now, as Weatherburn [141] demonstrates, a 2-dimensional Riemannian space can be embedded in a 3-dimensional Euclidean space, with the result that the angle between two vectors **c** and **d**, computed using Euclidean (or, in the case of orthogonal coordinate axes, Cartesian) components c and d, *i.e.*

$$\cos\theta = \mathbf{c}\cdot\mathbf{d} = \sum_i c^i d^i \tag{6.06}$$

is equivalent to that computed using (6.05), providing **a**=**c** and **b**=**d**. As these last two equalities hold if, and only if, the two vectors in each pair (*i.e.* the pair **a** and **c**, and the pair **b** and **d**) are identically equal, then under such circumstances the Euclidean and Riemannian geometries are indistinguishable.

On this basis, the numerical simplicity of Euclidean components may be used, *via* (6.06), to analyse the Riemannian geometry of the sphere. Now, as (6.06) involves only vector multiplication, this is attractive from the point of view of implementing a computer program, but it must be borne in mind that the measure defined by (6.05)—and hence (6.06)—is isotropic, in that angles do not depend on the relative orientation of the two vectors. Similarly, arc length as defined by (6.03) is isotropic. However, it has been shown in Chapter 4 that using an isotropic measure of dissimilarity in a partitioning algorithm will lead to the identification of isotropic clusters. If there is no *a priori* knowledge of the existence of isotropic clusters, then some measure of dissimilarity that embodies anisotropy must be used.

We are thus led to a need to examine distance measures for their applicability in this respect.

6.1.2 Distance measures

The previous sub-Section has demonstrated how a pragmatic approach to the computation of dissimilarity in Riemannian space, through the use of subtended angle rather than arc distance, allows the use of numerically efficient Euclidean distances, but at the cost of partitioning the data into isotropic

clusters. We are thus led to the requirement to identify distance measures that allow the connection between Euclidean and Riemannian geometries to be maintained, whilst admitting anisotropy.

In the review presented in Chapter 4, the computation of distances as inner products (*i.e.* norms) was introduced, as a means of maintaining numerical versatility. Thus, the distance d between the vectors **a** and **b** is computed as

$$d(\mathbf{a},\mathbf{b}) = \sqrt{(\mathbf{a}-\mathbf{b})^T \mathbf{A}(\mathbf{a}-\mathbf{b})} \tag{6.07}$$

where **A** represents a norm-inducing matrix. If the identity matrix is used as the norm-inducing matrix, then (6.07) produces Euclidean distances, with the result that the angular equivalence between Riemannian and Euclidean geometries—as demonstrated by (6.05) and (6.06)—may be utilised. Under these circumstances we may firstly compute the inner product Euclidean distance, d, between vectors **a** and **b** using

$$d = \sqrt{(\mathbf{x}_a - \mathbf{x}_b)^T \mathbf{I}(\mathbf{x}_a - \mathbf{x}_b)} \tag{6.08}$$

and then compute the subtended angle between the vectors from

$$\sin\theta = \frac{d}{2} \tag{6.09}$$

Implicit here is the fact that dissimilarity between two vectors must be defined in terms of θ, rather than d. If this is acceptable, then there will be no difficulty in implementing the FCM algorithm in Riemannian space. Of course, as it does in Euclidean space, the FCM algorithm will demonstrate a tendency to seek isotropic clusters which, in the case of orientation data, will appear as circles on the surface of the unit sphere.

Continuing with the theme of using inner products for distance measurement, by incorporating the sample covariance matrix as the norm-inducing matrix in (6.07), we should be able to account for anisotropy in cluster shape in a fashion that is completely analogous to the extensions to FCM represented by the FCV and FMLE algorithms. Using the sample covariance matrix as the norm-inducing matrix induce the anisotropic Mahalanobis distance in Euclidean space, and (6.09) may then be used to exploit the angular equivalence between the Riemannian and Euclidean geometries. Unfortunately, this is flawed: as there is no upper-bound to Mahalanobis distances, the

constraint imposed on d by the arc sine function in (6.09), *i.e.* $d \leq 2$, cannot be guaranteed.

We must conclude, therefore, that it is not possible to exploit the angular equivalence between Riemannian and Euclidean geometries, and yet maintain the capability of detecting anisotropic clusters through the use of the Mahalanobis distance. The following sub-Section summarises the position.

6.1.3 Summary

The discussion above, relating to the geometry of orientation data, and the choice of distance measure, may be summarised as follows:

(i) orientation data are implicitly Riemannian in nature, representing as they do the surface of the unit sphere;

(ii) to analyse orientation data in terms of their implicit 2-dimensional Riemannian geometry we should calculate distances using (6.03), but this is computationally unattractive;

(iii) whilst it is possible, through (6.05) and (6.06), to make use of the link between the intrinsically correct Riemannian geometry and the computationally preferable Euclidean geometry, the resulting angles are isotropic, which will prevent a partitioning algorithm from identifying anisotropic clusters in the data;

(iv) to account for anisotropy, Mahalanobis distances—calculated in Euclidean space and then converted to angles (which are equal in Euclidean and Riemannian spaces) using (6.09)—must be used; and

(v) the Mahalanobis distance has no upper bound, with the result that the arcsine function in (6.09) may become undefined.

Consequently, we are led to the conclusion that, notwithstanding the fact that it is intrinsically correct to analyse orientation data in terms of 2-dimensional Riemannian geometry, constraints on computation time and the requirement to be able to identify anisotropic clusters through the use of the Mahalanobis distance means that the analysis must be conducted using 3-dimensional Euclidean geometry and Cartesian coordinates. This will result in only an approximate solution, but one that may be adequately close to the correct Riemannian solution. As an amelioration, it must be borne in mind that an extension of the partitioning algorithm to account for, say, discontinuity surface roughness, would require an implementation in Euclidean space. Thus, the successful application of a fuzzy partitioning algorithm to discontinuity

orientation data in Euclidean space is an essential precursor to any subsequent analyses that incorporate other geometrical attributes.

The following Section describes how the application of a fuzzy partitioning algorithm to discontinuity orientation data in Euclidean space has been made, and the tactics adopted to overcome the difficulties due to the approximate nature of the geometry.

6.2 A Clustering Scheme based on Fuzzy Partitioning Algorithms for Discontinuity Orientation Data

This Section describes a novel numerical scheme for the identification of clusters in orientation data. This scheme has been developed by the Author, and is based on the application of fuzzy partitioning algorithms. The scheme involves the sequential application of three algorithms—fuzzy *c*-means (FCM), fuzzy covariance (FCV) and fuzzy maximum likelihood estimation (FMLE)—implemented in Euclidean space. The scheme has been programmed in FORTRAN, and the code of the resulting program—known as FADD (Fuzzy Analysis of Discontinuity Data—is presented in Appendix D.

The following three sub-Sections describe how this scheme has been implemented, dealing in particular (and respectively) with: the rationale behind the choice of these algorithms and the order of their application; the organisation of the algorithms with respect to programming; and aspects of the coding with respect to both the initialisation of the data partition and the analysis of orientation data in Euclidean space. A fourth sub-Section both discusses how the concept of hardening a fuzzy partition through the application of 'sharpening' may be applied to enhance the evaluation of cluster validity functions, and presents a number of modified cluster validity functions that incorporate sharpening.

6.2.1 Choice of the FCM, FCV and FMLE algorithms

The FCM, FCV and FMLE algorithms have been discussed in detail in Section 4.4. This discussion demonstrated how an increase in sophistication and computational effort, at the expense of numerical stability, is exhibited in the order of FCM, FCV and FMLE. The sequential application of these algorithms has been adopted to optimise their application, as the following paragraphs explain.

(i) FCM operates on an initial arbitrary initialisation of **U**, the membership value matrix, to produce a basic fuzzy partition of the data. The use of Euclidean distances in this algorithm dictates that the resulting partition will comprise of essentially isotropic clusters (which may not be fully appropriate in the presence of anisotropic clusters), but the robustness, numerical efficiency and rapid convergence of the algorithm render it ideal for the production of a first fuzzy partition.

(ii) The basic fuzzy partition resulting from FCM is then used as an initialisation for FCV, which produces a refined partition that accounts for variation in individual cluster shape. Working from a fuzzy, rather than an arbitrary, partition ensures that the fuzzy scatter matrices—*i.e.* the sample covariance matrices used as norm-inducing matrices in the Mahalanobis-like distance—computed by this algorithm rapidly converge to those representing the true anisotropic shapes of the clusters in the data, rather than the circular shapes induced by the Euclidean metric.

(iii) Finally, the refined partition resulting from the FCV algorithm is used as an initialisation for the FMLE algorithm. This stage of the scheme incorporates an exponential distance measure into the algorithm, defined by the adoption of maximum likelihood classifiers. This exponential distance, together with the norm-inducing covariance matrix, is used to finally partition the data into locally compact ellipsoidal clusters, whilst allowing any variations in cluster shape and cluster density to be taken into account. The result is that the final partition can be regarded as being optimal.

The FCM algorithm is numerically simple, requiring nothing more computationally sophisticated than the evaluation of matrix products. However, the FCV and FMLE algorithms require a great deal of matrix inversion procedures in order to generate the norm-inducing matrices required for the Mahalanobis-like distance, and FMLE additionally requires a large number of exponential functions to be computed; both of these procedures are computationally slow. The sequential implementation adopted here ensures that the initial data partitions presented to FCV and FMLE are reasonable approximations to the partitions subsequently determined by these algorithms, with the result that the computation time is minimised.

By carefully considering the computational steps in each of the three algorithms, it has been possible to optimise the use of individual sections of the resulting computer code, as the following sub-Section demonstrates.

6.2.2 Detail of the algorithms

The FCM, FCV and FMLE algorithms described in Chapter 4 are presented in a coherent manner in Tables 6.01, 6.02 and 6.03. These Tables show the major computational steps involved in the algorithms, and demonstrate how each algorithm may be considered as being formed from four modules, namely: initialisation; computation of centroids; computation of membership values; and assessment of convergence. The final three modules are executed cyclically until the algorithm converges.

The layout of the Tables shows how these modules correspond between the algorithms, and this correspondence has been exploited in the code for FADD. Thus, within FADD we find that the code associated with FCM is essentially common to all three algorithms, and hence this forms the basis of the scheme; within each module there are then additional sections of code specific to the FCV and FMLE algorithms. The benefit of coding the scheme in this fashion is twofold: a great deal of the code and the data storage is reused three times, with the result that the code is both easy to follow (and hence maintain) as well as being compact. This latter point is important, as the resulting program is designed to be implemented on unsophisticated microcomputers.

The complete FORTRAN code for FADD, together with details of the program structure, variable storage, subroutine functionality and data file format, is presented in Appendix D.

6.2.3 Special procedures

The application of the fuzzy objective function algorithms FCM, FCV and FMLE to the analysis of orientation data has necessitated the development of numerical procedures that account for the geometry of the data and efficiently initialise the fuzzy membership matrix, **U**. These procedures are now described.

Computation of centroid vectors

The FCM, FCV and FMLE algorithms are usually implemented in Euclidean space, where the Cartesian components of the coordinate values are independent of each other and may take values anywhere within the range of real numbers. Furthermore, the clusters defined in Euclidean space by a partition of the data according to these algorithms are convex ball-like (either spheroidal or ellipsoidal) structures, with the cluster centroid defined in terms

of the data's first moment of area. Thus, the Cartesian components of the centroid vector are also independent and similarly may take values anywhere within the range of real numbers.

However, in the case of orientation data, the data are defined as unit vectors, and therefore represent points on the surface of the unit sphere. Consequently, independence of the coordinate components is not permitted, as the clusters constitute circular or elliptical caps on the unit sphere, with the cluster centroids being constrained to be unit vectors.

Table 6.01 The FCM fuzzy partitioning algorithm.

Step	*Operation*
1	Adopt c, number of clusters, and m, fuzzy exponent. Initialise **U**, the fuzzy membership matrix.
2	Compute centroids, **v**, associated with all clusters: $$\mathbf{v}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{n} (u_{ij})^m} \quad \text{for } i = 1,c$$
3	Update **U**, by computing membership, u, based on new centroids: $$u_{ij} = \frac{\left[1/d^2(\mathbf{x}_j, \mathbf{v}_i) \right]^{\frac{1}{(m-1)}}}{\sum_{k=1}^{c} \left[1/d^2(\mathbf{x}_j, \mathbf{v}_k) \right]^{\frac{1}{(m-1)}}} \quad \text{for } i = 1,c \text{ and } j = 1,n$$ where Euclidean distances are computed using $$d^2(\mathbf{x}_j, \mathbf{v}_i) = (\mathbf{x}_j - \mathbf{v}_i)^T \mathbf{I} (\mathbf{x}_j - \mathbf{v}_i)$$
4	Test convergence criterion over all data and all clusters: if $\max_{ij} \left\| u_{ij} - \hat{u}_{ij} \right\| > \varepsilon$, where $0 \le \varepsilon \le 1$, and $\hat{u}$=corresponding value of membership value from previous iteration, repeat from step 2

Table 6.02 The FCV fuzzy partitioning algorithm.

Step	*Operation*
1	Adopt final fuzzy membership matrix, **U**, from FCM as initialisation of **U** for FCV.
2	Compute centroids, **v**, associated with all clusters: $$\mathbf{v}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{n} (u_{ij})^m} \qquad \text{for } i = 1,c$$ Compute fuzzy scatter matrices, **SF**, associated with all clusters: $$\mathbf{SF}_i = \sum_{j=1}^{n} (u_{ij})^m (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T \qquad \text{for } i = 1,c$$ Compute Mahalanobis-like norm-inducing matrices, $\mathbf{A}_{FM}$, associated with all clusters: $$(\mathbf{A}_{FM})_i = [\rho_i \det(\mathbf{SF}_i)]^{(1/p)} (\mathbf{SF}_i)^{-1} \qquad \text{for } i = 1,c$$ where ρ_i=volume associated with $\mathbf{SF}_i$ and p=number of space dimensions.
3	Update **U**, by computing membership, u, based on new centroids: $$u_{ij} = \frac{\left[1 / d^2(\mathbf{x}_j, \mathbf{v}_i) \right]^{\frac{1}{(m-1)}}}{\sum_{k=1}^{c} \left[1 / d^2(\mathbf{x}_j, \mathbf{v}_k) \right]^{\frac{1}{(m-1)}}} \qquad \text{for } i = 1,c \text{ and } j = 1,n$$ where Mahalanobis-like distances are computed using $$d^2(\mathbf{x}_j, \mathbf{v}_i) = (\mathbf{x}_j - \mathbf{v}_i)^T (\mathbf{A}_{FM})_i (\mathbf{x}_j - \mathbf{v}_i)$$
4	Test convergence criterion over all data and all clusters: if $\max_{ij} \left\| u_{ij} - \hat{u}_{ij} \right\| > \varepsilon$, where $0 \le \varepsilon \le 1$, and $\hat{u}$=corresponding value of membership value from previous iteration, repeat from step 2

Table 6.03 The FMLE fuzzy partitioning algorithm.

Step	*Operation*
1	Adopt final fuzzy membership matrix, **U**, from FCV as initialisation of **U** for FMLE.
2	Compute centroids, $\boldsymbol{\mu}$, associated with all clusters: $$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)\mathbf{x}_j}{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)} \quad \text{for } i = 1,c$$ Compute prior probability of cluster selection, p, associated with all clusters: $$p_i = \frac{1}{n}\sum_{j=1}^{n} h(i \mid \mathbf{x}_j) \quad \text{for } i = 1,c$$ Compute, for use as norm-inducing matrices, inverses of fuzzy covariance matrices, $\boldsymbol{\Sigma}$, associated with all clusters: $$\boldsymbol{\Sigma}_i = \frac{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)(\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} h(i \mid \mathbf{x}_j)} \quad \text{for } i = 1,c$$ where $h(i \mid \mathbf{x}_j)$ is the probability of selecting the ith cluster given the jth datum.
3	Update **U**, by computing membership, h, based on new centroids and maximum likelihood estimation: $$h(i \mid \mathbf{x}_j) = \frac{1}{\sum_{k=1}^{c}\left(\frac{w_{ij}}{w_{kj}}\right)}$$ where weights, w, are computed using $$w_{ij} = \frac{[\det(\boldsymbol{\Sigma}_i)]^{1/2}}{p_i}\exp\left[\frac{1}{2}(\mathbf{x}_j - \boldsymbol{\mu}_i)^T(\boldsymbol{\Sigma}_i)^{-1}(\mathbf{x}_j - \boldsymbol{\mu}_i)\right]$$
4	Test convergence criterion over all data and all clusters: if $\max_{ij}\left\|h_{ij} - \hat{h}_{ij}\right\| > \varepsilon$, where $0 \le \varepsilon \le 1$, and $\hat{h}$=corresponding value of membership value from previous iteration, repeat from step 2

Although the computation of centroid vectors in FCM, FCV and FMLE are shown in Step 2 of Tables 6.01, 6.02 and 6.03 respectively, this reflects the Euclidean case where, for each coordinate direction, the end point of the vector lies at the centre of area of the vectors comprising the cluster. In the case of orientation data, because the end points of the data vectors form a cap lying on the surface of a sphere, the coordinates of the end point of the computed centroid will lie inside this sphere, *i.e.* these vectors will possess a magnitude less than unity. In the case of small, compact clusters, the magnitude of the centroid vector will differ negligibly from unity, but as the size of the cluster increases the error will become appreciable. The limit will be for uniformly distributed orientation data, in which case the centroid vector becomes undefined, as the end point converges on the coordinate origin (the centre of the unit sphere).

In this application, the centroid has the physical interpretation of representing a unit vector defined in terms of direction cosines, and so this variable magnitude is not acceptable. To overcome this, the expedient of constantly scaling the magnitude of the centroid vector to unity has been adopted. The end point of the unit vector thus formed is not the centroid of the cluster in terms of the first moment of area, but is the centroid in terms of 'mean orientation'. This, of course, is the output required of the partitioning scheme. The pseudocode for centroid computation procedure is as follows:

```
For all clusters, j:
    magv = 0.0                          initialise vector magnitude
    For all dimensions, i:              loop over all dimensions
    magv = magv + v(i,j) * v(i,j)       build magnitude of V
    Next dimension
    magv = SQRT(magv)
    For all dimensions, i:
        v(i,j) = v(i,j)/magv            scale components of V
    Next dimension
Next cluster
```

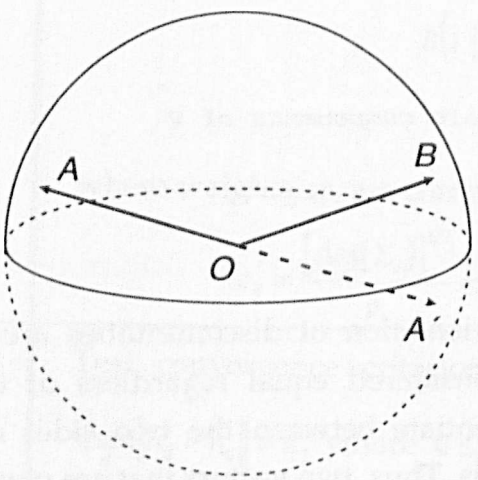
Accounting for antipodal vectors

By definition, vectors representing the orientation of discontinuities are not directed. That is, parallel vectors are considered equal regardless of their direction of action, as we do not differentiate between the two sides of a discontinuity surface in geometrical analysis. Thus, two vectors that are parallel but mathematically directed in opposite directions are, from the point of view of orientation analysis, considered to belong to the same cluster of vectors. In

the context of orientation data, such vectors appear as antipodal to one another, and special attention must be paid to the problem of accounting for antipodal vectors, in order to prevent them being assigned to different clusters.

The numerical representation of orientation data in terms of Cartesian components is implicitly directed, as the sign of each component gives directional information. Customarily, we assume orientation data to be mathematically directed either upwards or downwards, depending on the sign convention adopted. Geometrically, the data may therefore be regarded as points on the surface of either an upper hemisphere or a lower hemisphere, respectively. Considering the vectors corresponding to orientation data as undirected means that, as Figure 6.02 illustrates, two vectors that are almost horizontal and separated by an azimuth angle close to 180° are not maximally dissimilar, but rather, are very similar. It is critically important that a partitioning algorithm recognises this, and is able to determine which of the two oppositely directed vectors representing an orientation should be incorporated in the analysis.

If we make use of the geometry of the right angled triangle containing the vectors *AO*, *BO* and *A'O*, as illustrated in Figure 6.03, it can be seen that the reverse vector *A'O* is required when the computed Euclidean distance between the two vectors *AO* and *BO* is greater than $\sqrt{2}$, the hypotenuse distance of a unit right angle isosceles triangle. In a partitioning algorithm, it is not the distances between any two data vectors themselves that are of interest, but the distances between centroid vectors and data vectors. Thus, a datum may be



Vectors *AO* and *BO* are almost antipodal. The vector *A'O* should be used in calculations with *BO*, rather than *AO*, as orientation vectors are undirected.

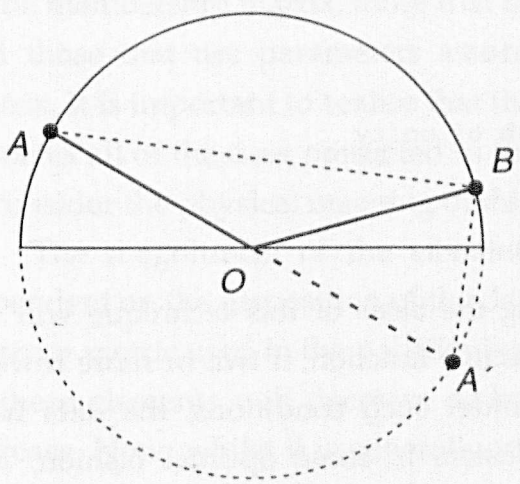
Figure 6.02 Illustration of antipodal vectors.

required in the computation procedure in both its natural (say, downward directed) and reversed forms, depending on the distances to the various centroids under consideration. It is necessary, therefore, to separately record the intrinsic components of a datum, and the information regarding its direction relative to the computed centroids. Fortunately, when working with Cartesian components, a vector may be reversed simply by changing the signs of each component, and so only the sign information needs to be recorded.

In order to implement this, **U** (the membership matrix) has been augmented with sign information such that in conjunction with **X**, the data component matrix, the reverse direction for any datum j may simply be taken by applying the sign of u_{ij} to the components of x_j. This technique, which is only possible because each u_{ij} is (by definition) positive, has been adopted—rather than storing the information in a separate matrix—to reduce the storage requirements of the program.

During the procedure to determine which of the vector directions is required, it is convenient to compute the distance between the datum and the prototype centroid of the corresponding cluster. The geometry of the triangle $AA'B$ indicates that it is simpler (and, incidentally, computationally faster) to work with squared distances, and compute the Euclidean distance associated with the antipodal vector as shown in Figure 6.03.

Pseudocode for the procedure adopted in FADD for augmenting **U** is as follows:



AO and BO are both unit vectors. Instead of A, the antipodal vector A' is required when $A\hat{O}B > 90°$, or $(AB)^2 > 2$. The distance $A'B$ is conveniently given by $A'B = \sqrt{4-(AB)^2}$.

Figure 6.03 Test for antipodal vector and computation of associated Euclidean distance.

```
For all data, i:
    For all clusters, j:
        COMPUTE distance between x(i) and v(j)
        IF (distance*distance)>2 THEN
            sense = -1              antipodal vector
        ELSE
            sense = +1
        ENDIF
        COMPUTE effective components of x, using sense
        COMPUTE u(i,j)
        u(i,j) = u(i,j)*sense       augment with sign
    Next cluster
Next datum
```

In all further computation involving u_{ij}, the absolute value is taken, with the sign being available for reversing x_j, as this example for computing the centroids demonstrates:

```
For all clusters, i:
  scale = 0.0
    For all dimensions, k:
        v(k,i) = 0.0
    Next dimension
    For all data, j:
        sense = SIGN(u(i,j))
        memb = ABS(u(i,j))**fuzzy_exponent
        scale = scale + ABS(memb)
        For all dimensions, k:
            1st_moment(k) = 1st_moment(k) + x(k,j)*memb*sense
        Next dimension
    Next datum
    For all dimensions, k:
        v(k,i) = 1st_moment(k) / scale
    Next dimension
    COMPUTE v(i) normalised to length of unity
Next cluster
```

Initialisation of membership value matrix

The fuzzy clustering algorithms forming the basis of this technique will not converge to a minimisation of J_m, the objective function, if two or more rows of **U** are identical [15]. This is because, under such conditions, the data have effectively already been assigned to clusters in some optimal fashion, and further iterations of the algorithm will have no effect on **U**. The use of the 'average' partition—given by (4.19)—as an initialisation seems to offer no

advantage in this application given the spherical nature of the data, and so an alternative is required.

The only requirements for an initialisation of **U** are: that each column should sum to unity; and that all rows should be different. Providing these requirements are met, any convenient procedure may be used. Consequently, an initialisation has been adopted here in which a hard partition is generated based on column address. This has the two advantages that it is numerically easy to implement and is fast. The pseudocode listing for the procedure is

```
For all data, j:
    For all clusters, i:
        u(i,j) = 0.0                          set all elements to zero
    Next cluster
    u(1+MOD(j,no_of_clusters),j)=1.0    set one element to unity
Next datum
```

This procedure results in an initialisation of the form:

$$\mathbf{U}^{(0)} = \begin{bmatrix} \underbrace{0.0 & 0.0 & 1.0}_{\text{cyclic pattern}} & \underbrace{0.0 & 0.0 & 1.0}_{\text{cyclic pattern}} & \underbrace{\cdots}_{\text{pattern repeats}} \\ 1.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & \cdots \\ 0.0 & 1.0 & 0.0 & 0.0 & 1.0 & 0.0 & \cdots \end{bmatrix} \tag{6.10}$$

6.2.4 Cluster validity functions incorporating sharpening

The discussion on cluster validity functions presented in Chapter 4 indicated that these functions may be grouped into three categories: those that make use of the membership matrix; those that use the geometry of the resulting clusters; and those that use parameters associated with a matrix such as the scatter matrix. It is important to realise that the customary definition of these functions involves all of the data presented to the partitioning algorithm, and it is useful to consider the physical meaning of this.

The magnitudes of the elements in the final membership matrix are dependent on the disposition of the data, the number of clusters sought and the distance metric used in the partitioning algorithm; for example, the magnitudes of these elements will decrease as both sparsity and the number of clusters increase. Now, whilst it is generally regarded that a good partition will consist of relatively few clusters, the elements of which have membership values close to unity, by evaluating cluster validity functions on the basis of all the data presented to the algorithm, we may risk including some low-membership data

that are widely separated from well-formed and compact clusters, with the erroneous result that a poor partition will be indicated.

An example of how humans overcome this effect can readily be found in the analysis of rock discontinuity orientation data. Experience in the analysis of such data tells us that we should expect a number of 'randomly' disposed elements in any set of orientation measurements; physically, these will represent discontinuities that do not clearly belong to any main discontinuity set, and are likely to be fractures of small extent. A skilled interpreter of orientation data plotted on the hemispherical projection will take such spurious data into account, and mentally remove it from the interpretation, with the effect that it is regarded as background noise. On the basis of such experience, therefore, it may be regarded as valid to attempt such a procedure in the evaluation of a cluster validity function. Certainly, it can be argued that a function such as separation coefficient or separation index—the magnitude of which is dependent on the size of the resulting clusters—will give misleading results if such spurious data are incorrectly included as cluster members.

Data sharpening

One method of overcoming the problem of incorrectly including spurious data in the evaluation of cluster validity is to somehow disregard those data that possess low membership values. This is the idea of 'sharpening', as discussed in sub-Section 4.6.2, and although Bezdek [15] notes that a scheme for objectively excluding data is untried, he goes on to say (p.144) that "...the idea of paring or pruning fuzzy clusters to find distinctive 'cores' is intriguing".

However, developing a 'sharpened partition' by simply excluding data on the basis of membership value introduces new difficulties, for example: at what membership value should an exclusion threshold be set; how many data can be excluded before the remainder are not representative of the whole; and should the cluster centroids be recomputed once the data have been excluded? No proposals are offered here to overcome these difficulties, but rather an investigative approach has been followed, in which a data sharpening scheme has been implemented and the results examined for enhanced identification of the optimal partition.

Incorporating a sharpened partition into cluster validity functions

The method adopted here as a means of forming a sharpened partition may be regarded as a combination of forming the nearest maximum membership hard

partition and the α-level core of a cluster (both described in Section 4.6), and is formalised as

$$\left[u_{ij}\right]_{s\alpha} = \begin{cases} u_{ij}; & \text{if } u_{ij} = \max\limits_{1\le k\le c}\{u_{kj}\} \text{ and } u_{ij} \ge \alpha \\ 0; & \text{otherwise} \end{cases} \tag{6.11}$$

where $\left[u_{ij}\right]_{s\alpha}$ is taken to represent 'the partition sharpened at a level given by α'. The two underlying processes employed in forming the sharpened partition are clear from (6.11), namely that: the α-level core is formed by only accepting those maximum memberships that exceed the threshold α; and the maximum membership partition is introduced by only allowing one of the rows in the membership matrix to be non-zero. It can be argued that applying (6.11) results in an invalid membership matrix, **U**, as the columns of the matrix no longer sum to unity. However, this difficulty can be overcome by applying the conditions in (6.11) to a supplementary cluster membership vector, and leaving **U** unchanged. This supplementary vector may be used to determine into which clusters data are to be assigned during the evaluation of the cluster validity functions. This is computationally convenient, and because of this is the approach adopted in the program FADD.

Incorporating a sharpened partition into cluster validity functions is straightforward. We simply redefine any such function, on the basis that all references to elements of **U** are replaced with elements of $\mathbf{U}_{s\alpha}$. As an example, we can redefine the partition coefficient as

$$F_{s\alpha} = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c}\left(\left[u_{ij}\right]_{s\alpha}\right)^2 \tag{6.12}$$

where $\left[u_{ij}\right]_{s\alpha}$ is defined in (6.11). An alternative definition can be written, in which the original elements of **U** are retained and the function is supplemented with the cluster membership vector. Under these circumstances the sharpened partition coefficient may be written as

$$F_{s\alpha} = \frac{1}{n}\sum_{j=1}^{n}\left[u_{c_j j}\,\mathrm{sign}\left(c_j\right)\right]^2 \tag{6.13}$$

The double subscript to the element $u_{c_j j}$ indicates that the row address of the element is contained within the cluster membership vector *c*. Incorporating this

vector highlights the hard nature of the sharpened partition, with the result that the function needs only to be summed once, over the number of data.

The basic principles embodied in (6.12) and (6.13) are easily incorporated into the other cluster validity functions presented in 4.5.1, and as such there is no need to rewrite the sharpened cluster validity functions here. Their use is, however, discussed in the next Section.

6.2.5 Summary

A detailed description of the implementation of an original scheme for identifying clusters in discontinuity orientation data, based on the fuzzy partitioning algorithms FCM, FCV and FMLE, has been given in this Section. This scheme has been developed as the computer program FADD (Fuzzy Analysis of Discontinuity Data) by the Author.

The rationale behind the choice of these algorithms, and the order of their implementation in terms of increasing capability to define an optimal fuzzy partition at the expense of numerical speed and stability was discussed in 6.2.1.

The necessity to implement in Euclidean space what is strictly a problem of Riemannian geometry requires the adoption of special numerical procedures. The methods used to both compute centroid vectors and cope with antipodal data, and these, together with a procedure for initialising the membership matrix, were described in sub-Section 6.2.3.

Finally, a method by which a sharpened fuzzy partition may be formed was presented in sub-Section 6.2.4, together with proposals for its use with customary cluster validity functions. This concept was put forward as a means of reducing the effect of data that clearly does not belong to any cluster, and hence increasing the efficacy with which an optimal partition may be identified.

On the strength of this work, it is now possible to examine in some detail the effectiveness with which the resulting computer program—FADD—identifies clusters in orientation data. This is the subject of the next Section.

6.3 Investigations into the Effectiveness of FADD for Partitioning Orientation Data

This Section reports on trial analyses, with both synthetic and natural data sets, performed as a means of demonstrating the effectiveness with which the computer program FADD identifies clusters in orientation data. Synthetic data

sets, containing known clusters of various shapes and densities, have been used to study—in an empirical fashion—the capability of the program to correctly identify clusters and determine their orientation. Similarly, the natural data set of copper porphyry data, published by Mahtab *et alia* [86], has been used examine how FADD behaves when faced with a noisy data set. These data have also been used to study how the various cluster validity functions described in Chapter 4 may be used to identify the optimal partition, and how this identification is affected by choice of fuzzy exponent m and sharpening of the resulting partition. The copper porphyry data has been chosen because their wide use in the literature allows comparisons to be made with other partitioning processes. An empirical study has also been made of the convergence of the algorithms, in order to select a suitable convergence criterion for the program.

6.3.1 Empirical verification of FADD

The effectiveness with which FADD identifies clusters of both differing shape and density has been examined—in an empirical fashion—through the application of the program to three data sets; two synthetic and one natural. The synthetic data have been used in order that the results from FADD may be compared with a known partition, rather than—as would be the case with a natural data set—a partition determined either by some other partitioning algorithm or heuristically by a human interpreter. The natural data have been used to examine whether, when presented with noisy data, FADD is able to identify the commonly accepted clusters.

The two synthetic data sets have designed to test particular aspects of FADD. The first is a straightforward set consisting of four, well separated, isotropic clusters, and the second is a more taxing set consisting of two clusters, one of which is highly anisotropic. This second set was synthesised such that application of the FCM algorithm by itself would result in an incorrect partition; in both cases, the program FADD correctly identifies the prototypical clusters.

Four, clearly separated, isotropic clusters

A data set consisting of four clearly separated clusters, each comprising random variables drawn from a Fisher distribution [92], was synthesised and used to check that FADD could recover the prototypical clusters, when these

features were clearly separated. The data in each cluster were generated using the following method:

(1) set concentration parameter κ, and $\lambda = \exp(-2\kappa)$;

(2) generate colatitude $\Theta = 2\arcsin\sqrt{\left\{\frac{-\ln[R_1(1-\lambda)+\lambda]}{2\kappa}\right\}}$ where R_1 is a random variable drawn from a uniform distribution, $0 \le R_1 \le 1$;

(3) generate longitude $\Phi = 2\pi R_2$, where R_2 is a random variable drawn from a uniform distribution, $0 \le R_2 \le 1$;

(4) repeat from step (2) until all data in cluster have been generated;

(5) compute actual mean orientation of cluster $(\overline{\Theta}, \overline{\Phi})$;

(6) compute combined rotation matrix which represents rotation of $(\overline{\Theta}, \overline{\Phi})$ to (0,90), followed by rotation to $\overline{\Theta} = 90 - \beta$, $\overline{\Phi} = (450 - \alpha) \bmod 360$, where (α,β) represent required dip direction/dip angle of mean orientation of cluster;

(7) rotate all data in cluster using rotation matrix computed in (6).

Steps (1) to (3) of this procedure are taken from Fisher *et alia* [44]. The data generated for the analysis, together with the relevant distribution parameters, are presented in Table 6.04.

The results of the analysis are shown in Figure 6.04. This Figure is a reproduction of the actual output from FADD, and represents a lower hemispherical equal angle projection of the partition resulting from the FCM algorithm in FADD. In this Figure, the data forming each cluster are marked by letters, the associated cluster centroid by the corresponding numeral, and lower case and upper case letters are used to represent the projected location of single and multiple data, respectively. The boundary and the centre of the projection are marked by the characters x and *, respectively.

The clusters in this data set represent the following conditions:

(i) cluster 1: many members, dense;
(ii) cluster 2: many members, sparse;
(iii) cluster 3: few members, dense;
(iv) cluster 4: few members, sparse.

Table 6.04 Test data comprising four isotropic clusters.

Circular cluster 1: Fisher distribution	
Distribution parameters:	*Planar data (dip direction/dip angle):*
number of data=25 mean orientation=045/30 concentration parameter=70	226/59, 218/64, 230/67, 238/51, 226/69, 239/51, 225/61, 226/52, 214/75, 218/62, 230/67, 215/66, 230/55, 217/64, 211/59, 222/46, 217/59, 227/56, 220/62, 251/63, 230/56, 219/62, 224/62, 230/58, 228/60
Circular cluster 2: Fisher distribution	
Distribution parameters:	*Planar data (dip direction/dip angle):*
number of data=25 mean orientation=135/15 concentration parameter=15	314/83, 292/77, 341/68, 297/65, 313/84, 294/72, 329/79, 331/62, 320/61, 304/85, 334/60, 321/74, 289/43, 323/67, 310/63, 123/82, 345/73, 275/90, 144/85, 308/80, 327/69, 290/64, 135/74, 327/88, 342/86
Circular cluster 3: Fisher distribution	
Distribution parameters:	*Planar data (dip direction/dip angle):*
number of data=10 mean orientation=235/30 concentration parameter=60	055/51, 044/49, 038/54, 055/69, 067/55, 057/66, 068/73, 059/58, 060/59, 045/71
Circular cluster 4: Fisher distribution	
Distribution parameters:	*Planar data (dip direction/dip angle):*
number of data=10 mean orientation=250/80 concentration parameter=15	162/22, 129/24, 080/28, 007/10, 051/09, 003/38, 056/09, 192/20, 276/18, 058/51

In addition, cluster 2 is oriented such that the normals to the planes represented by the data are antipodal, in order that the capability of FADD to detect such clusters is tested.

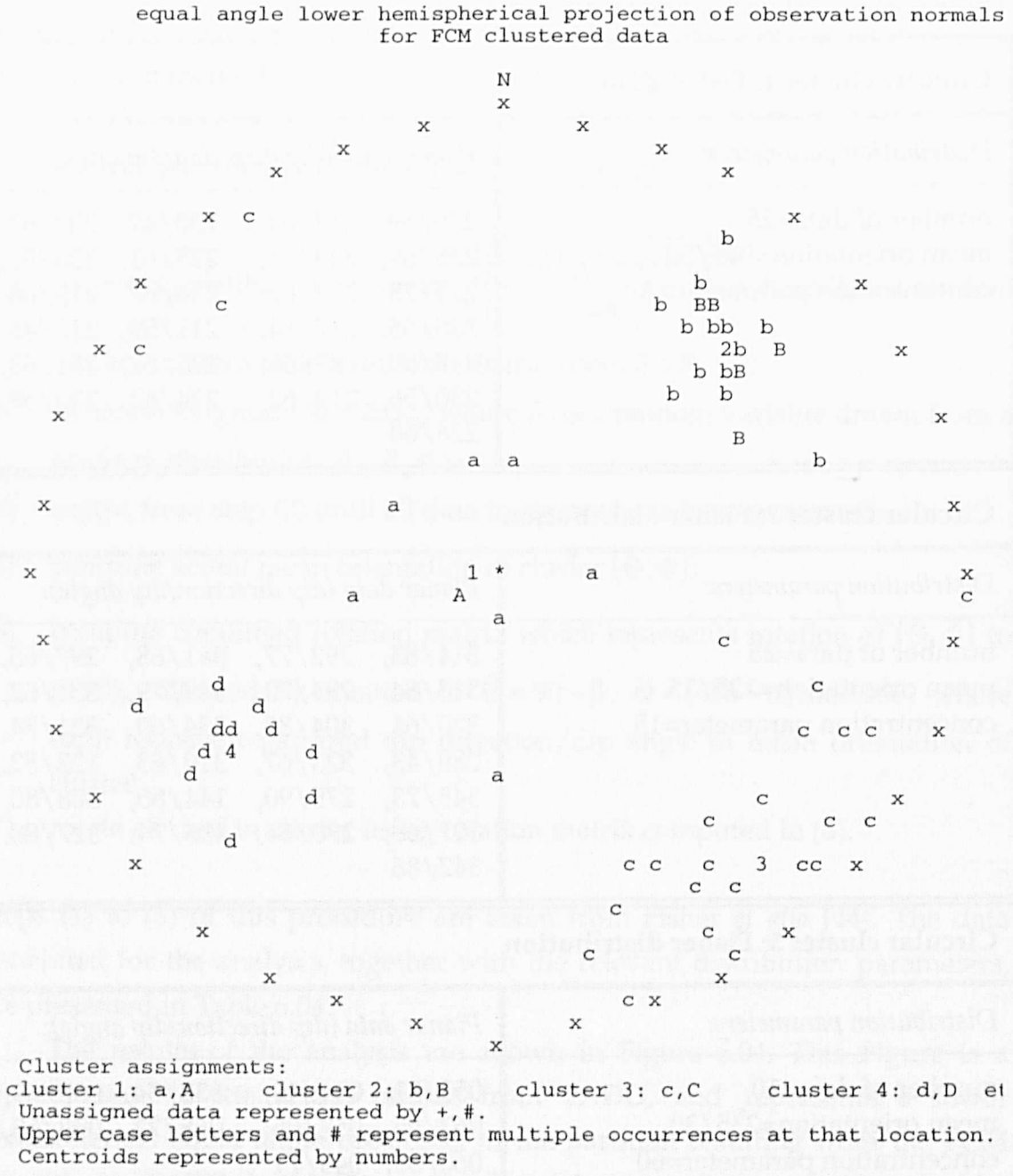


Figure 6.04 FCM partitioning of data containing four clusters drawn from Fisher distributions.

As Figure 6.04 demonstrates, the scheme has correctly identified the four clusters, including the antipodal cluster 2. Table 6.05 compares the various results computed by FADD with the corresponding generated data. In terms of the partition, only one datum—that with orientation 058/51—has been incorrectly assigned on the basis of prototypical clusters. A study of Figure 6.04 shows that this datum is visually more acceptable as a member of the cluster marked with the letters d rather than the cluster marked a. The result of this incorrect assignment, however, is that the two clusters, each of which have few members, have undergone a relatively large change in their membership, and

Table 6.05 Comparison between prototypical and computed cluster centroids for isotropic test data.

Cluster		Number of data		Centroid dip direction/dip angle		
generated	assigned by FADD	generated	assigned by FADD	generated	computed by FADD	angular difference (degrees)
1	4	25	25	045/30	044/31	1.3
2	1	25	25	135/15	137/15	0.0
3	2	10	11	235/30	236/32	2.2
4	3	10	9	250/80	268/86	6.3

hence their computed centroids are some angular distance from the prototypical orientation. Not surprisingly, the effect is most pronounced with the cluster containing few, sparse, data. The cluster centroids computed for those clusters containing many (in this case 25) data are, without doubt for engineering purposes, sufficiently close to the prototypes to be regarded as identical. The differences that exist are due to FADD computing centroids as a function of fuzzy membership (which gives increased weight to those data close to the cluster centroid) rather than the customary method using moments (in which equal weight is assumed for all data). If we accept the heuristic that data close to a cluster centroid are better descriptors of the cluster than those data at the extremes of the cluster, then there is no reason to reject the centroids computed as part of the partitioning process as correct.

Data containing a highly anisotropic cluster

An important capability of FADD is the identification of anisotropic clusters. To check this capability, a data set consisting of one highly anisotropic cluster and one circular cluster has been synthesised. The anisotropic data were generated by drawing two random variables from a standard normal distribution, multiplying these by suitable standard deviations and then, using the resulting numbers as x- and y-axis direction cosines, computing the z-axis direction cosine thus:

$$n = \sqrt{l^2 + m^2} = \sqrt{\sigma_x R_x + \sigma_y R_y} \qquad 6.14$$

Table 6.06 Test data containing an anisotropic cluster.

Anisotropic cluster: 'Binormal' distribution	
Distribution parameters:	*Planar data (dip direction/dip angle):*
number of data=35 mean orientation=000/90 normal to plane of major axis=000/00 standard deviation (major axis)=0.3 standard deviation (minor axis)=0.05	348/06, 079/15, 127/06, 103/06, 021/03, 274/12, 088/53, 109/12, 266/41, 267/15, 098/07, 273/29, 279/35, 096/41, 252/14, 244/05, 107/12, 275/30, 314/06, 099/13, 101/11, 302/02, 256/15, 275/14, 300/11, 108/15, 072/13, 080/02, 126/09, 040/05, 288/08, 112/03, 090/29, 099/21, 277/32
Circular cluster: Fisher distribution	
Distribution parameters:	*Planar data (dip direction/dip angle):*
number of data=15 mean orientation=040/45 concentration parameter=100	213/39, 228/50, 215/53, 228/41, 207/46, 222/51, 218/37, 224/31, 217/46, 228/43, 213/48, 216/56, 216/44, 231/50, 227/44

where σ_x and σ_y are standard deviations, R_x and R_y are standard normal random variables and the subscripts refer to coordinate axes. The circular data were generated as for the previous test data. The distribution parameters and the resulting data set are presented in Table 6.06.

On the basis of the review presented in Chapter 4, we would expect the FCM algorithm to perform poorly with this data, as the Euclidean norm (which is inherent in the distance measure used in the algorithm) will attempt to form isotropic clusters. As the equal angle lower hemispherical projections presented in Figure 6.05 shows, this is clearly the case. The isotropic cluster to the north-east of the projection has been identified, but the data at the eastern end of the anisotropic cluster have also been assigned to this cluster, which is incorrect. What is surprising, however, is the angular distance over which the isotropic cluster exerts its influence. This is in part due to the concentration of the isotropic cluster, and shows how well-defined isotropic clusters will 'capture' data that would better fall within anisotropic clusters. The ramifications for the analysis of rock discontinuity orientation data are clear:

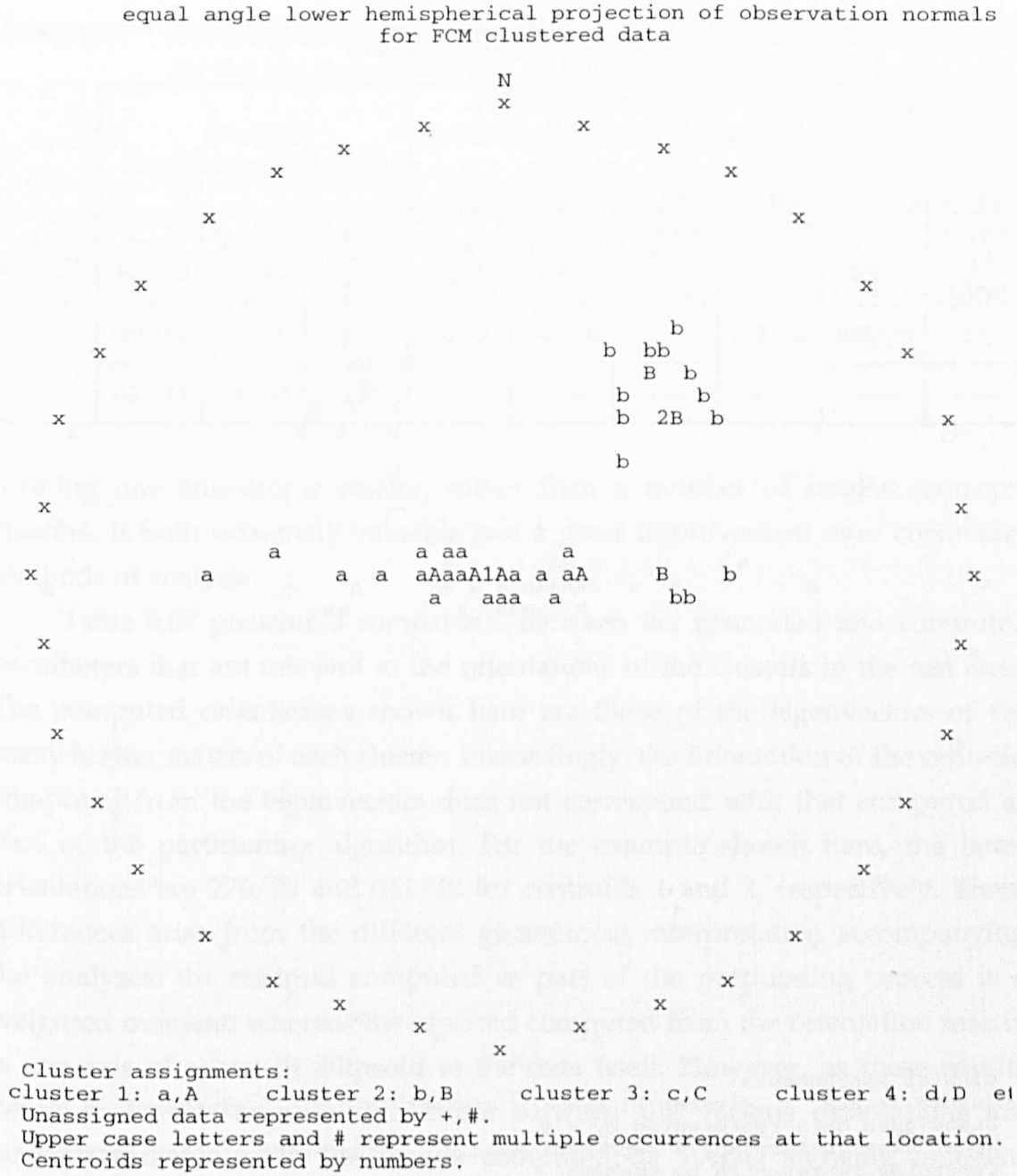


Figure 6.05 FCM partition of test data containing an anisotropic cluster.

the FCM algorithm (and, of course, any clustering algorithm that uses the Euclidean norm) may identify clusters that are not representative of the rock mass structure.

Figure 6.06 shows the partition resulting from the FCV algorithm, the initial membership matrix for which was the final membership matrix computed by the FCM algorithm. In the FCV algorithm, a covariance matrix based on the membership of the data within a cluster is used to form the corresponding Mahalanobis-like norm, with the result that anisotropic clusters may be identified. As the Figure clearly shows, this has indeed taken place,

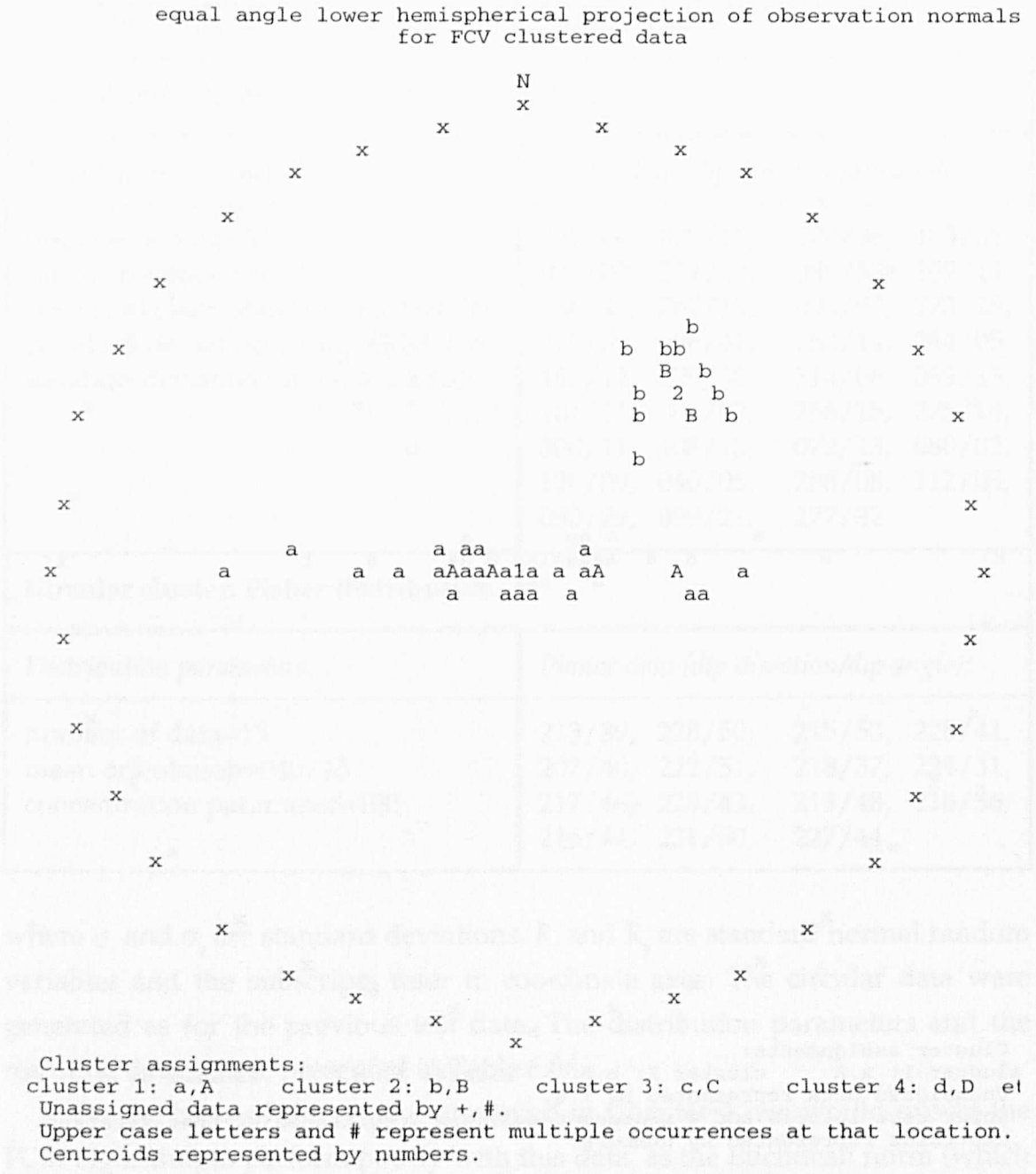


Figure 6.06 FCV partition of test data containing an anisotropic cluster.

with the algorithm correctly identifying both the isotropic and the anisotropic clusters. This demonstrates that the sequential application of the FCM and FCV algorithms will enable clusters of a highly anisotropic form to be identified.

It is useful to consider the data presented in Table 6.06 in practical terms. The data in the anisotropic cluster may be taken to represent, for example, orientation measurements taken with a small magnetic compass at various locations over a large undulating discontinuity surface, or a suite of orientations from a folded structure. The ability to identify these data as

Table 6.07 Comparison between prototypical and computed cluster centroids for the test data containing an anisotropic cluster.

	Centroid dip direction/dip angle			Normal to plane containing major principal axis dip direction/dip angle			Normal to plane containing minor principal axis dip direction/dip angle		
cluster	from FADD	proto-type	angular diff. (deg)	from FADD	proto-type	angular diff. (deg)	from FADD	proto-type	angular diff. (deg)
1	088/89	090/90	1.0	182/00	000/00	2.0	272/00	090/00	2.0
2	040/44	040/45	1.0	——	——	——	——	——	——

forming one anisotropic cluster, rather than a number of smaller isotropic clusters, is both extremely valuable and a great improvement over customary methods of analysis.

Table 6.07 presents a comparison between the generated and computed parameters that are relevant to the orientations of the clusters in the test data. The computed orientations shown here are those of the eigenvectors of the fuzzy scatter matrix of each cluster. Interestingly, the orientation of the centroid computed from the eigenvectors does not correspond with that computed as part of the partitioning algorithm. For the example shown here, the latter orientations are 276/88 and 041/48 for centroids 1 and 2, respectively. These differences arise from the different geometrical interpretation accompanying the analyses: the centroid computed as part of the partitioning process is a weighted moment; whereas the centroid computed from the orientation matrix is one axis of a best-fit ellipsoid to the data itself. However, as these results demonstrate, the angular differences between the various orientations are largely academic, with the results computed by FADD showing excellent agreement with the prototypical values.

The eigenvalues of the cluster orientation matrices are shown in Table 6.08, together with the classification coordinates $\ln(\tau_2/\tau_1)$ and $\ln(\tau_3/\tau_2)$. The eigenvalues are presented in decreasing magnitude (*i.e.* $\tau_3 > \tau_2 > \tau_1$), with the interpretation that the largest eigenvalue represents the orientation of the cluster 'mean', the intermediate eigenvalue the major axis of the cluster and the smallest eigenvalue the minor axis of the cluster. Using the classification diagram of Fisher *et alia* [44], reproduced here as Figure 6.07(a), we can interpret the results as follows:

Table 6.08 Eigenvalues and corresponding classification coordinates for anisotropic test data.

	Eigenvalues			Classification coordinates	
Cluster	τ_3	τ_2	τ_1	$\ln(\tau_2/\tau_1)$	$\ln(\tau_3/\tau_2)$
1	31.0	3.70	0.10	3.61	2.13
2	15.0	0.18	0.11	0.49	4.42

(i) one of the clusters is almost exactly isotropic, and is highly concentrated;
(ii) one of the clusters is highly anisotropic (on the girdle/cluster transition) and of intermediate concentration;
(iii) the anisotropic cluster is more dense than the isotropic cluster.

By comparison with Table 6.06, it can be seen that these are the attributes used to synthesise the data: the isotropic cluster was synthesised from a Fisher distribution with a concentration parameter of 100; and the anisotropic cluster was synthesised from two normal distributions, whose ratio of standard deviations was 6.0.

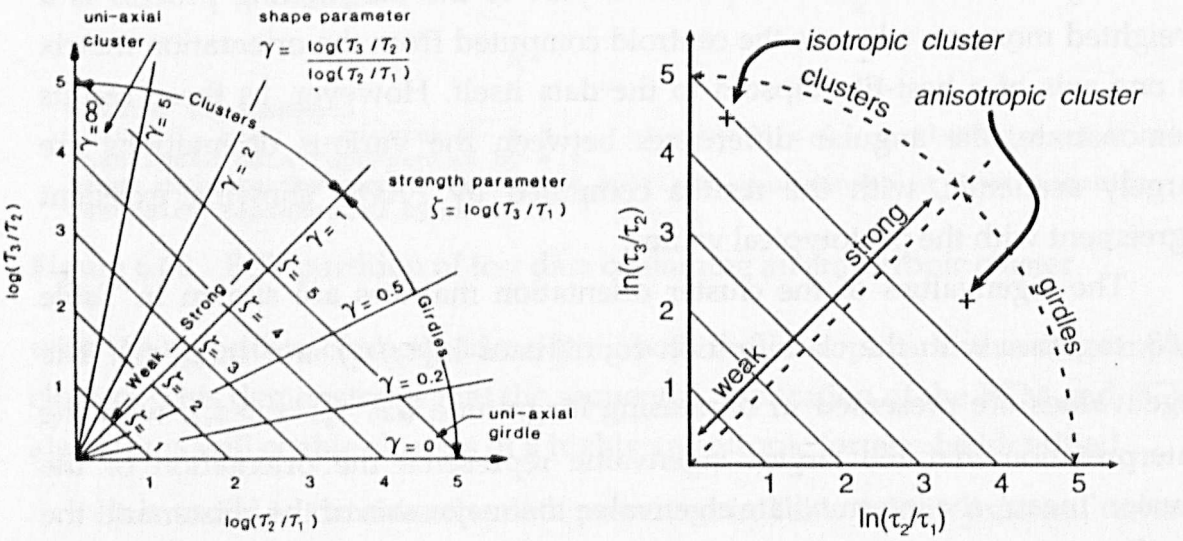


Figure 6.07 Classification of clusters in orientation data: (a) typical patterns for observed values of eigenvalues (from Fisher *et alia* [44]); (b) results of FCV algorithm applied to test data.

Thus we may conclude that, not only is FADD capable of identifying anisotropic clusters in orientation data through the application of the FCV algorithm, but the eigenvectors and eigenvalues of the resulting cluster orientation matrices permit the clusters to be both oriented and classified. These are useful capabilities, as the test results in the next Division show.

Copper porphyry data

A more realistic test of the capabilities of FADD has been undertaken using the copper porphyry data of Mahtab *et alia* [86], reproduced here as Table 2.01. In Section 2.5, this set was used in an analysis of the customary contouring procedure; here it is subject to a sophisticated partitioning analysis.

An analysis using FADD has been undertaken with the algorithmic values of m=2 and α=0.9 (the reasons for the selection of these particular values are described in sub-Section 6.3.3), for the widely accepted case of c=3. The results arising from the FCV and FMLE algorithms (in terms of number of data comprising each cluster, and the azimuth and dip of the plane corresponding to the best-fit normal) are presented in Table 6.09. Also given in this table is a comparison with the results due to other workers; in particular, the original results from PATCH published by Mahtab *et alia* [86], the subsequent results of Shanley and Mahtab [125] in which the Bingham distribution was fitted to the clusters, and the work of Grossman [52] in which the bivariate normal distribution on the tangent plane was fitted.

It is clear from this Table that FADD successfully identifies the same clusters as other schemes, and allocates data to these clusters in the approximately the same proportions. The major differences highlighted by this Table are in the discriminatory performance of the algorithms, with FCV assigning the fewest data to the clusters, and the algorithm FRACTAN, by Shanley and Mahtab [125], assigning every datum to a cluster. The discriminatory behaviour of FCV is linked directly to the magnitude of the

Table 6.09 Porphyry data: comparison of cluster orientation and size for five different cluster identification schemes.

	FCV		FMLE		Mahtab *et alia* [86]		Shanley and Mahtab [125]		Grossman [52]	
cluster	azimuth/ dip	no. of data	azimuth/ dip	no. of data	azimuth/ dip	no. of data	azimuth/ dip	no. of data	azimuth/ dip	no. of data
1	345/80	59	347/82	103	348/81	77	346/81	112	342/82	104
2	249/82	37	255/79	114	249/84	57	252/76	127	253/82	106
3	125/18	21	132/14	55	120/23	7	123/17	47	127/25	15
total		117		272		141		286		225

Table 6.10 Porphyry data: cluster orientations for FCV and FMLE partitioning.

Cluster	Number of data	τ_3	τ_2	τ_1	$\ln(\tau_2/\tau_1)$	$\ln(\tau_3/\tau_2)$	plane to best fit normal	normal to plane containing: major principal axis	normal to plane containing: minor principal axis
FCV									
1	59	56	1.5	1.1	0.31	3.62	345/80	72/15	288/71
2	37	36	0.78	0.53	0.39	3.83	249/82	162/16	314/71
3	21	20	0.54	0.36	0.41	3.61	125/18	140/16	49/04
FMLE									
1	103	91	8.4	3.8	0.79	2.38	347/82	260/20	55/67
2	114	92	15	7.8	0.65	1.81	255/79	166/07	287/76
3	55	45	6.2	4.0	0.44	1.98	132/14	204/04	113/12

sharpening threshold α, as this parameter can be interpreted as 'certainty of set membership': in this example only those data with a certainty greater than 0.9 (or 90%) have been included. In the case of FMLE, the exponential form of the distance metric means that both the 'boundary' of a cluster is marked by a pronounced reduction in membership values, and that all of the data within a cluster possesses nearly equal membership. This results in a great deal more data being assigned to clusters.

As discussed earlier, the algorithms embodied in FADD return additional characterisation parameters, in terms of the eigenvalues and corresponding orientations of the principal axes of the clusters, and these are presented in Table 6.10. The eigenvalue-based classification coordinates—$\ln(\tau_2/\tau_1)$ and $\ln(\tau_3/\tau_2)$—shown in this Table may be placed in the context of Figure 6.07, and when this is done for the results from the FCV algorithm, we can see that all three clusters may be regarded as isotropic and of high density. The density is much lower, and the degrees of isotropy different, for the clusters resulting from the FMLE algorithm, but this is to be expected given the vastly increased number of data in each cluster.

In summary, we can see that FADD operates successfully on sets of natural orientation data, in which clearly defined clusters do not exist. Therefore, can regard FADD as an improvement over the customary counting-circle methods, given both its robust mathematical origins and objective basis, and more general than the schemes that implement a procedure in which a probabilistic distribution is fitted to the data.

6.3.2 Algorithm convergence criteria

The customary convergence criterion used for fuzzy objective function partitioning algorithms is that shown in Step 4 of Tables 6.01, 6.02 and 6.03, namely

$$\max_{ij} \left\| u_{ij} - \hat{u}_{ij} \right\| > \varepsilon \tag{6.15}$$

where the threshold value is set as $0 \leq \varepsilon \leq 1$, and $\hat{u}$=corresponding membership value from the previous iteration. The use of this criterion probably stems from a perceived need to ensure as few misclassifications in the data as possible. However, from the point of view of engineering rock mechanics, it can be argued that the most useful result from a partitioning analysis of orientation data will be the orientation of a cluster centroid, as these analyses are usually a means to an end (for example, stability analysis), and are not an end in their own right. Under such circumstances, it makes sense to examine how the change of cluster centroid might be used as a convergence criterion, such that time consuming and pointless iterations may be avoided. There is, after all, little point continuing with a partitioning analysis until all of the membership values change by less than some threshold value, once the changes in the usable output have fallen below an acceptable threshold.

The anisotropic test data have been used in an empirical study of change in membership value and change in cluster centroid as convergence criteria. Figure 6.08 shows how—for FCM (diagrams (a) and (b)) and FCV (diagram (c)) partitioning with m=2 and c=2—both of these criteria change as the iteration proceeds. It should be noted that the second of the two diagrams displaying the FCM results is an enlarged version of the first, and shows in greater detail both the variation in the maximum change in membership value for each cluster, and the maximum angular change in centroid position, as the algorithm draws close to convergence. A study of this Figure allows us to gain some insight into the operation of the algorithms, and make useful observations about the convergence of the FCM and the FCV algorithms. Thus:

(i) the FCM algorithm appears to regard the initial arbitrary partition as 'good' (*i.e.* small changes in membership values take place initially), but in attempting to improve on the partition will change membership values—and hence the orientation of the centroids—dramatically, before finally converging;

(ii) if required, the FCV algorithm will make major changes to the final FCM partition, and these changes will take place immediately in the iterative process;

(iii) neither criterion monotonically decreases, and they may even display an occasional tendency to jump in magnitude;

(iv) in all cases, changes in membership value appear to converge at a similar rate to the maximum change in centroid orientation criterion, and;

(v) the maximum angular change in centroid orientation may not correspond with the maximum change in membership value, suggesting that data in sparsely populated regions of the data space may undergo large changes in membership values in response to newly computed centroid orientations.

If we consider Figure 6.08 in conjunction with Figures 6.05 and 6.06, we can see that the transition from FCM to FCV partitioning may be accompanied by dramatic changes to cluster shape, centroid orientation, and cluster membership. Furthermore, the evidence in these Figures suggests that, once a datum is effectively assigned to a particular cluster, its membership in that cluster will not change markedly if little change occurs in the centroid orientation. The cyclical nature of the iteration means that this can be recast as the complementary case: if little change occurs in centroid orientation, then there will be little change in membership values.

We may conclude, therefore, that there is little advantage to be gained by adopting the change in centroid criterion over the change in membership criterion for determining convergence of the partitioning process for orientation data. Indeed, the experience of this Author, gained through undertaking numerous analyses with FADD, is that the change in centroid criterion may not adequately reflect the convergence of the partitioning, and will only ever tend to save around 5 iterative cycles over the change in membership value criterion. Thus, in order to guarantee correct convergence, the change of membership criterion must be used, even though the precision with which the centroid orientations are computed may be meaningless in terms of an engineering analysis.

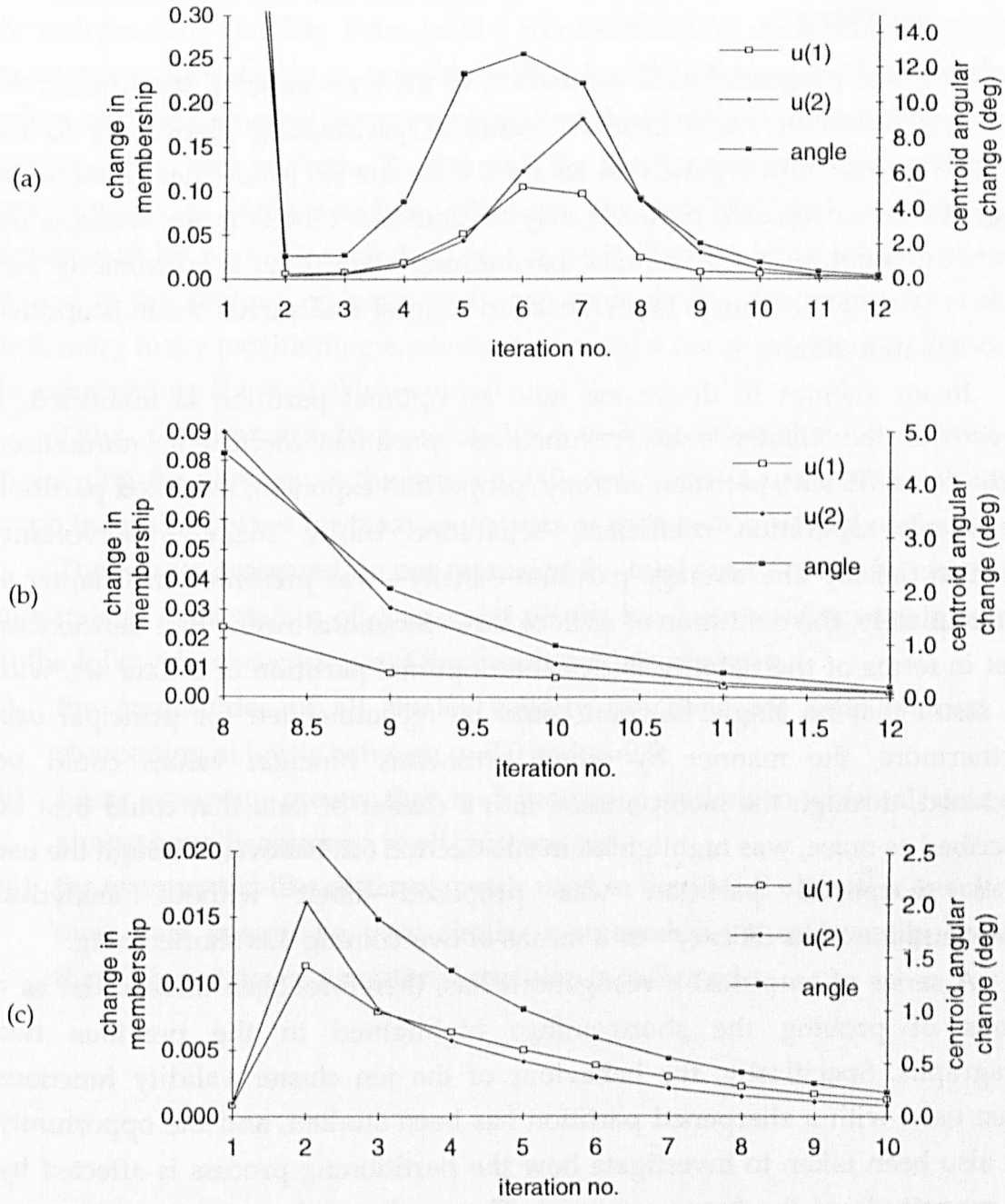


Figure 6.08 Anisotropic test data: Convergence of partitioning iteration with m=2. (a) and (b): FCM algorithm, (c): FCV algorithm.

6.3.3 Assessment of cluster validity functions applied to sharpened partitions

The computer program FADD represents, to the knowledge of the Author, the first application of fuzzy objective function partitioning algorithms to the analysis of orientation data, and as such there are no guidelines available to suggest how an optimal partition may be detected. Furthermore, whereas the fuzzy exponent, m, used in fuzzy partitioning algorithms is customarily and arbitrarily set at 2, there is no evidence to suggest that such a value is optimal for orientation data.

In an attempt to determine how an optimal partition is identified, a review of ten cluster validity functions—partition coefficient, normalised partition coefficient, partition entropy, proportion exponent, modified partition coefficient, separation coefficient, separation index, fuzzy hypervolume, partition density and average partition density—was presented in Chapter 4. Unfortunately, the definition of each of these functions means that deficiencies exist in terms of their ability to detect an optimal partition of a data set, with the result that no single function could be recommended for principal use. Furthermore, the manner by which erroneous function values could be produced, through the incorporation into a cluster of data that could best be described as noise, was highlighted in sub-Section 6.2.4 above, although the use of a sharpened partition was proposed—albeit without analytical demonstration of its efficacy—as a means of overcoming this shortcoming.

A series of empirical investigations has, therefore, been undertaken as a means of probing the shortcomings highlighted in the previous two paragraphs. Specifically, the behaviour of the ten cluster validity functions when used with a sharpened partition has been studied, and the opportunity has also been taken to investigate how the partitioning process is affected by the magnitude of the fuzzy exponent. The results of these investigations are presented here in this sub-Section.

The data used in this investigation are the porphyry data of Mahtab *et alia* [86]. The choice of this data was made because:

(i) their wide use in the literature allows comparisons to be made between this cluster identification scheme and others; and

(ii) the general noisiness of the data, together with the existence of three poorly defined clusters, poses a stiff test of cluster validity functions.

The following ten divisions of this sub-Section contain the results of the investigations, in the form of commentary associated with graphs of each

cluster validity function. In all cases, the number of clusters is assumed to be the independent variable. Four graphs are presented for each function, chosen to cover—as thoroughly as possible with a limited number of Figures—those values of the parameters fuzzy exponent and sharpening threshold that may be used in a practical analysis. To this end, the fuzzy exponents have arbitrarily been limited to integer values, although the case of hard partitioning—represented by m=1—is excluded, on the basis that the scope of this work is limited to the analysis of fuzzy partitions. In order to allow comparison with customary fuzzy partitioning analysis, the case of a fuzzy exponent m=2 and no data sharpening (*i.e.* α=0.0) is included.

Thus, the four graphs presented for each function show the result of sharpening the partition at the levels α=0.0, α=0.5, α=0.7 and α=0.9, with each graph including curves for fuzzy exponents of m=2, m=3, m=4 and m=5.

The results presented do not represent the total extent of the investigation undertaken. Presentation of condensed results has been made possible, based on the following observations of the porphyry data analysis:

(i) the magnitudes of all cluster validity functions are little affected by sharpening at levels between α=0.0 and α=0.5;

(ii) fuzzy exponents greater than m=5 produce partitions in which all data are almost equally assigned to all clusters; and

(iii) the exponential-like distance metric used in the FMLE algorithm results in most data possessing very similar membership values, usually greater than 0.9, and hence no optimal partition is indicated.

Partition coefficient

The partition coefficient is defined in sub-Section 4.5.1 as

$$F = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c}\left(u_{ij}\right)^2 \tag{6.16}$$

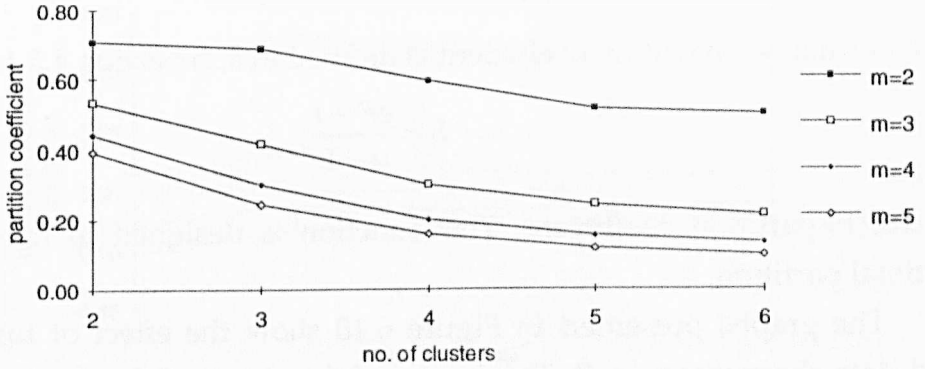
The partition coefficient is expected to show a maximum at the optimal partition, but as was noted in sub-Section 4.5.1, it exhibits a tendency to both monotonically decrease as the number of clusters increases and indicate better partitions associated with decreasing fuzziness.

The graphs in Figure 6.09 concur with these observations. In general, the magnitude of the partition coefficient decreases as both the number of clusters increases and the fuzzy exponent increases. A further observation that can be made about the partition coefficient is that it identifies the increasing fuzziness associated with increasing fuzzy exponent. In each of the four graphs in the Figure, it can clearly be seen that deviations away from the underlying trend of the curve (the means by which the optimal partition is identified) tend to be reduced as the fuzzy exponent increases, with the result that any such features are totally lost with m=5. There is little doubt that, for the range of fuzzy exponents used in this investigation, the customary value of m=2 most easily allows the greatest positive deviation—representative of the optimal partition—to be identified.

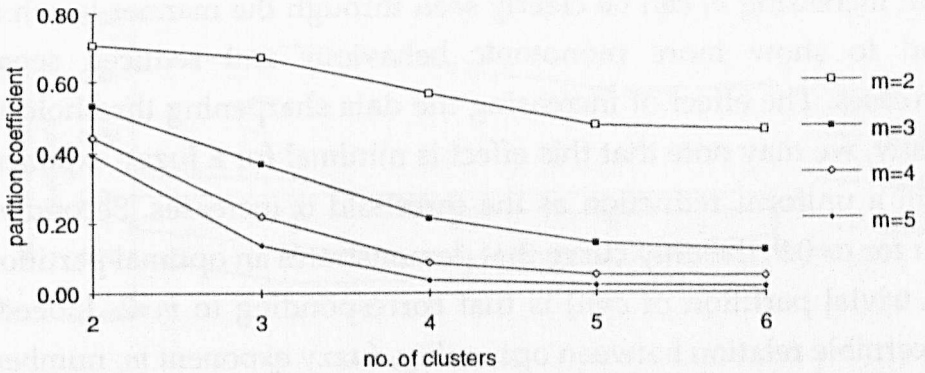
The effect of sharpening the partition is marked. As the sharpening threshold is increased, the magnitude of the function decreases dramatically for fuzzy exponents greater then 2. In the case of an exponent of 2, the optimality indicated at c=2 becomes more pronounced as the sharpening threshold increases, with greatest clarity (for the sharpening values chosen) being achieved at α=0.9.

Although these results indicate a choice of m=2 and α=0.9, there is ambiguity as to whether the optimal partition is indicated—on the basis of deviation from a monotonically decreasing curve—at c=2 or c=3. It is possible to fit a fair curve through either of the sets of points c=2,4,5,6 or c=3,4,5,6, and hence showing either c=3 or c=2 as being optimal. This highlights the need for more than one cluster validity function.

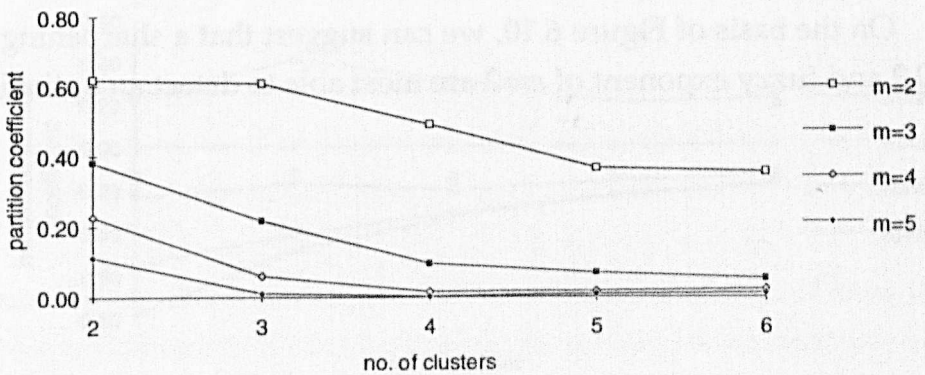
(a) α=0.0



(b) α=0.5



(c) α=0.7



(d) α=0.9

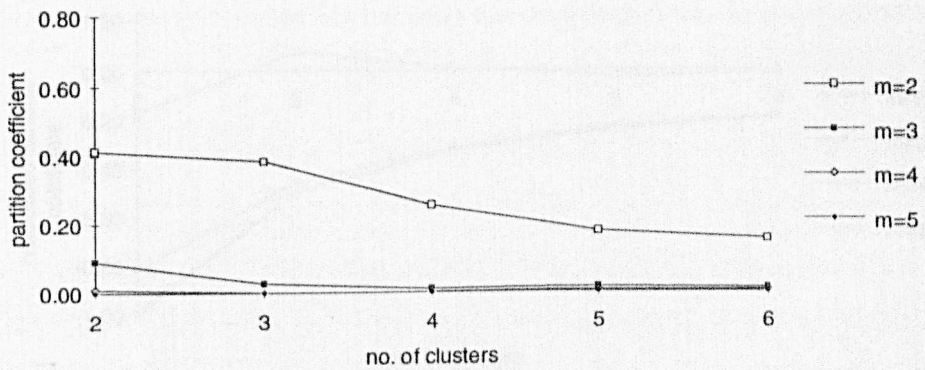


Figure 6.09 Porphyry data: partition coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Normalised partition coefficient

The normalised partition coefficient is defined in sub-Section 4.5.1 as

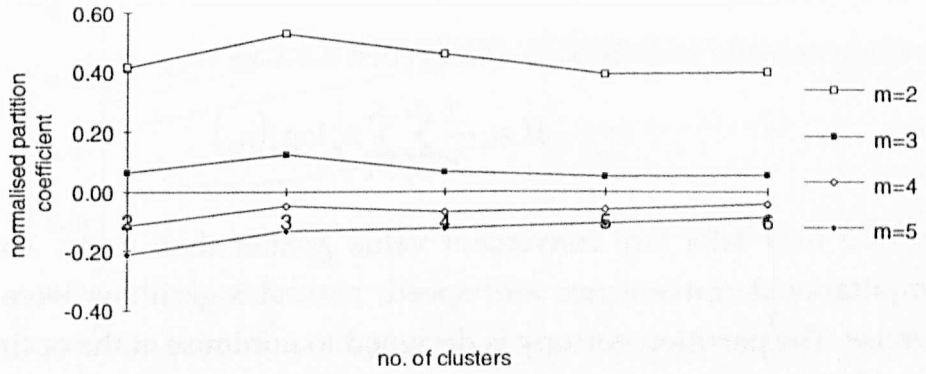
$$B = \frac{cF - 1}{c - 1} \tag{6.17}$$

where F=partition coefficient. This function is designed to maximise at the optimal partition.

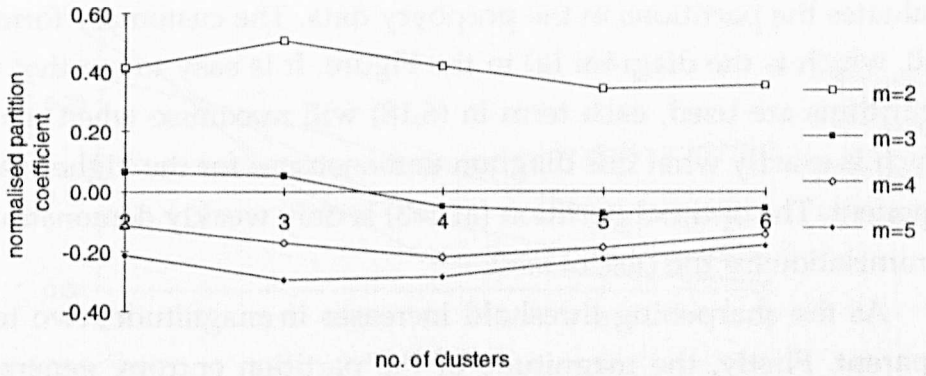
The graphs presented in Figure 6.10 show the effect of fuzzy exponent and data sharpening on B. The increased fuzziness of the partition associated with increasing m can be clearly seen through the manner in which the curves tend to show more monotonic behaviour and reduced separation as m increases. The effect of increasing the data sharpening threshold is interesting. Firstly, we may note that this effect is minimal for a fuzzy exponent m=2, apart from a uniform reduction as the threshold α increases. Secondly, we can see that for α=0.9, the only curve that demonstrates an optimal partition (other than the trivial partition of c=n) is that corresponding to m=2. Indeed, there is no discernible relation between optimality, fuzzy exponent m, number of clusters c and other values of α.

On the basis of Figure 6.10, we can suggest that a sharpening threshold of α=0.9 and fuzzy exponent of m=2 are most able to detect an optimal partition.

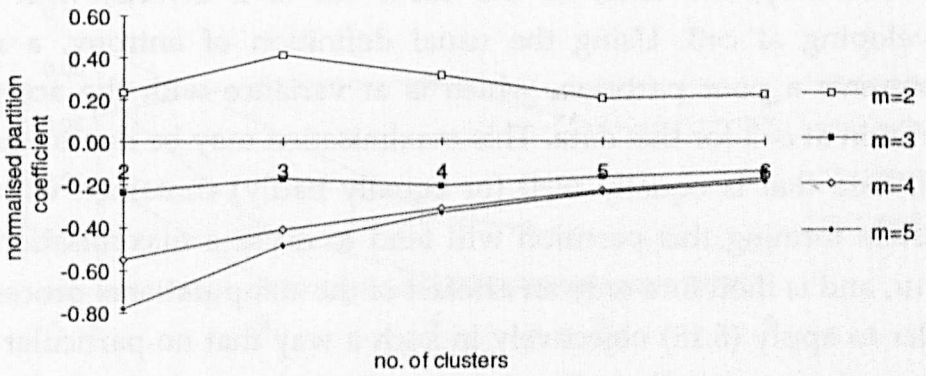
(a) α=0.0



(b) α=0.5



(c) α=0.7



(d) α=0.9

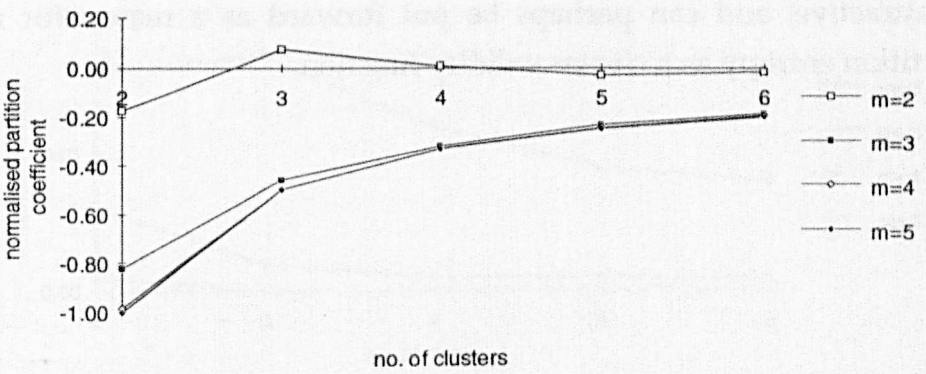


Figure 6.10 Porphyry data: normalised partition coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Partition entropy

Partition entropy is defined in sub-Section 4.5.1 as

$$H = -\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}\log_a\left(u_{ij}\right) \tag{6.18}$$

where a may take any convenient value greater than unity. For the sake of computational convenience and speed, natural logarithms were used in this exercise. The partition entropy is designed to minimise at the optimal partition.

The four diagrams in Figure 6.11 demonstrate how partition entropy evaluates the partitions in the porphyry data. The customary formulation is for α=0, which is the diagram (a) in the Figure. It is easy to see that when natural logarithms are used, each term in (6.18) will maximise when $u = e^{-1} = 0.368\ldots$, which is exactly what this diagram demonstrates for the higher values of fuzzy exponent. The optimal partition (at c=3) is only weakly demonstrated by a local minimisation for the case of m=2.

As the sharpening threshold increases in magnitude, two tendencies are apparent. Firstly, the magnitude of the partition entropy generally decreases and secondly, the form of the curve for m=2 inverts, with a maximum developing at c=3. Using the usual definition of entropy, a maximisation represents a poor partition, which is at variance with the accepted optimal partition at c=3 for this data. This maximisation may be due to the fact that, at c=3, data that is equally well (or equally badly) contained within all of the clusters forming this partition will tend to cause a maximisation of the term $u\ln u$, and is therefore only an artefact of the computational process. Indeed, in order to apply (6.18) objectively in such a way that no particular value of c is favoured, a range of logarithmic bases should be used. This is computationally unattractive, and can perhaps be put forward as a reason for not adopting partition entropy as a cluster validity function.

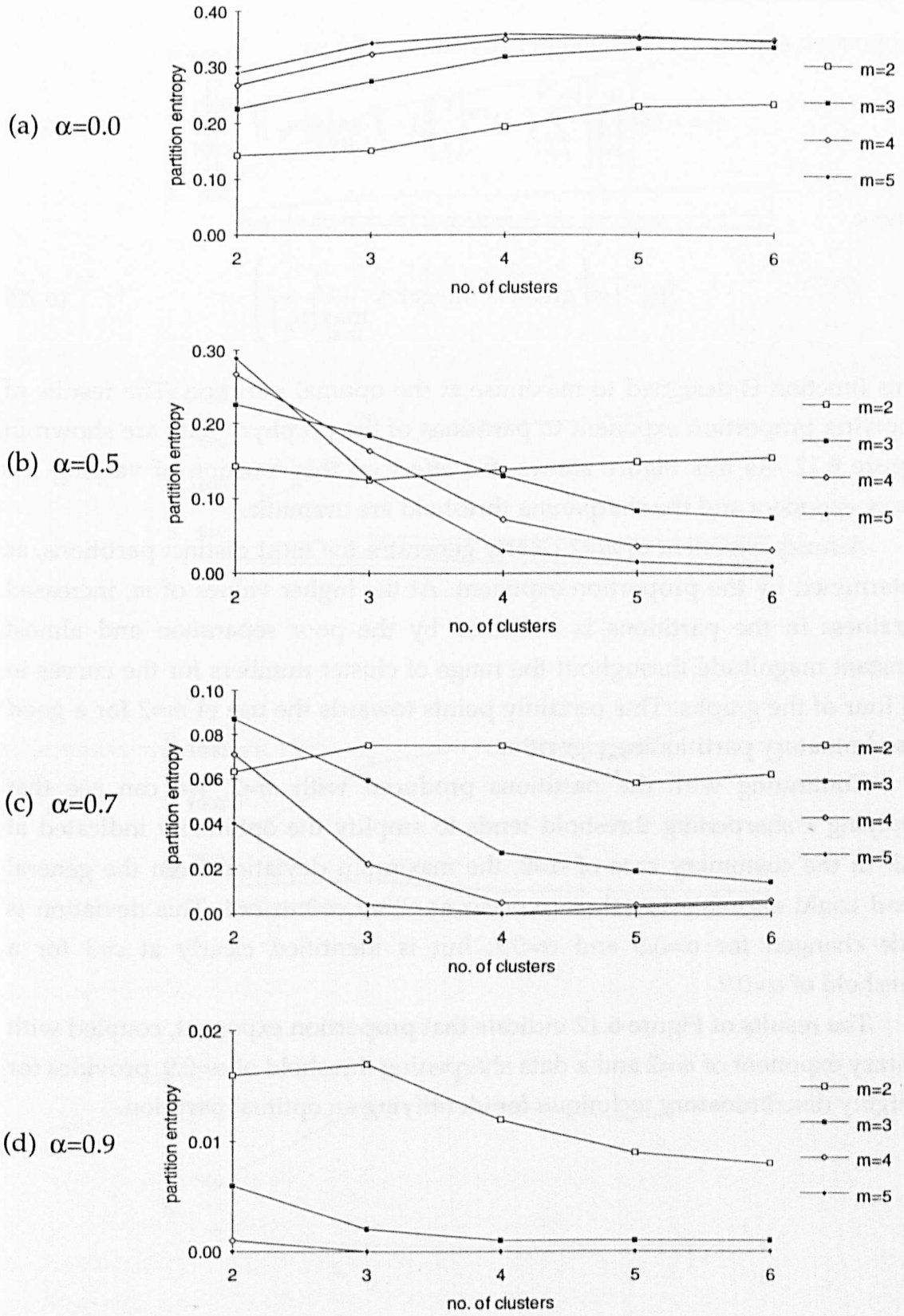


Figure 6.11 Porphyry data: partition entropy for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Proportion exponent

Proportion exponent is defined in sub-Section 4.5.1 as

$$P = -\ln\left\{\prod_{k=1}^{n}\left[\sum_{j=1}^{\left[\mu_k^{-1}\right]}(-1)^{j+1}\binom{c}{j}\left(1-j\cdot\max_{1\le i\le c}\{u_{ik}\}\right)^{c-1}\right]\right\} \tag{6.19}$$

where

$$\left[\mu_k^{-1}\right] = \left(\text{greatest integer} \le \frac{1}{\max\limits_{1\le i\le c}\{u_{ik}\}}\right) \tag{6.20}$$

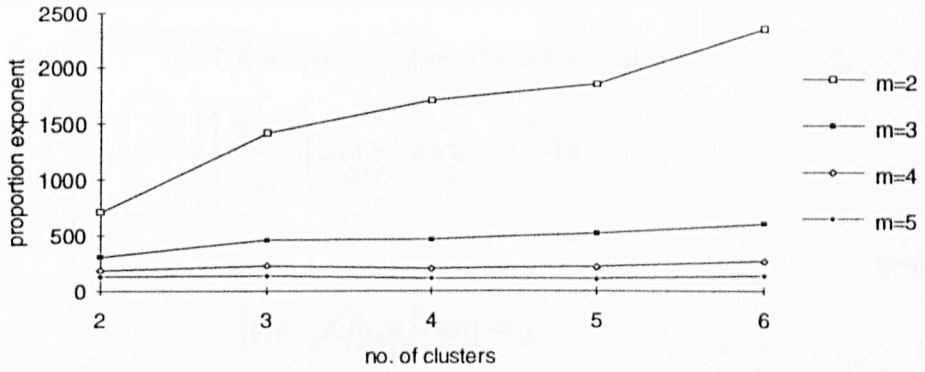
This function is designed to maximise at the optimal partition. The results of applying proportion exponent to partitions of the porphyry data are shown in Figure 6.12. As this Figure shows, the effect on this function of varying the fuzzy exponent and the sharpening threshold are dramatic.

A fuzzy exponent of m=2 clearly generates the most distinct partitions, as determined by the proportion exponent. At the higher values of m, increased fuzziness in the partitions is indicated by the poor separation and almost constant magnitude throughout the range of cluster numbers for the curves in all four of the graphs. This certainly points towards the use of m=2 for a good discriminatory partitioning algorithm.

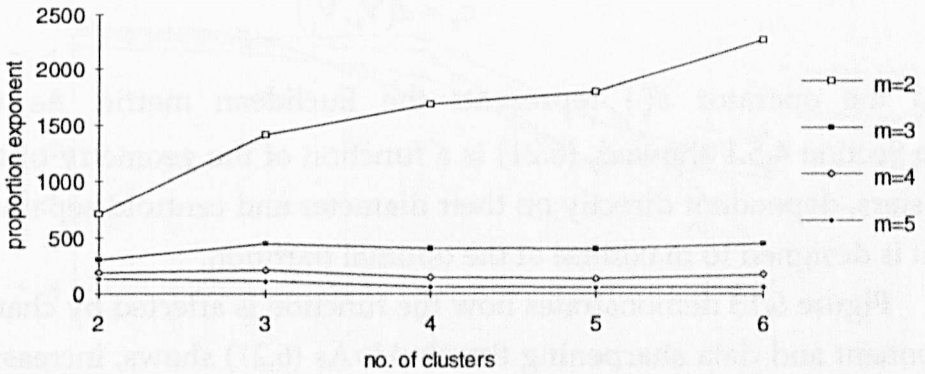
Continuing with the partitions produced with m=2, we can see that applying a sharpening threshold tends to amplify the optimality indicated at c=3. In the customary case of α=0, the maximum deviation from the general trend could arguably be taken to occur at either c=3 or c=4. This deviation is little changed for α=0.5 and α=0.7, but is identified clearly at c=3 for a threshold of α=0.9.

The results of Figure 6.12 indicate that proportion exponent, coupled with a fuzzy exponent of m=2 and a data sharpening threshold of α=0.9, provides for a highly discriminatory technique for identifying an optimal partition.

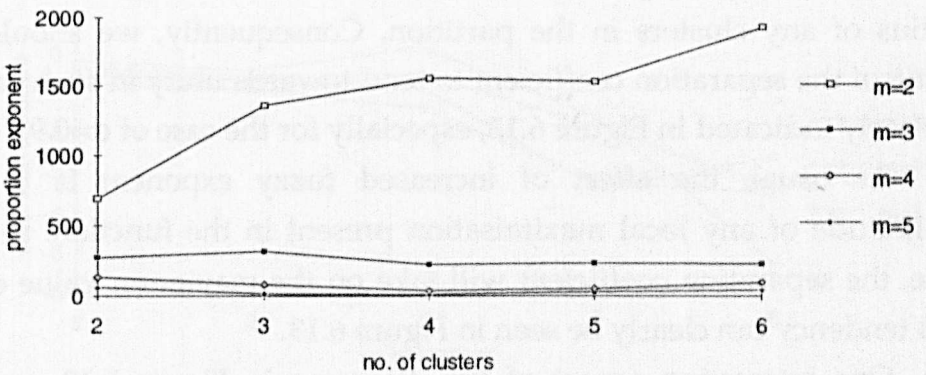
(a) α=0.0



(b) α=0.5



(c) α=0.7



(d) α=0.9

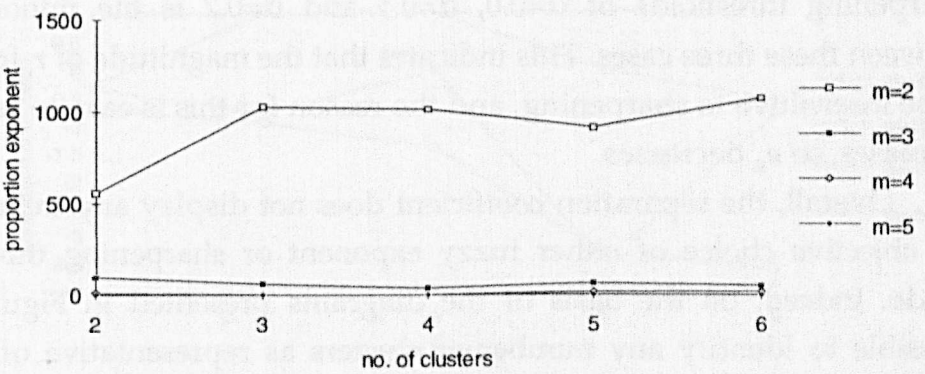


Figure 6.12 Porphyry data: proportion exponent for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Separation coefficient

Separation coefficient is defined in sub-Section 4.5.1 as

$$G = 1 - \max_{i+1 \le j \le c} \left\{ \max_{1 \le i \le c-1} \left\{ \frac{r_i + r_j}{c_{ij}} \right\} \right\} \tag{6.21}$$

where

$$r_i = \max_{1 \le k \le n} \left\{ u_{ik} d(\mathbf{X}_k, \mathbf{V}_i) \right\} \tag{6.22}$$

$$c_{ij} = d(\mathbf{V}_i, \mathbf{V}_j) \tag{6.23}$$

and the operator $d(\cdot)$ represents the Euclidean metric. As the work in sub-Section 4.5.1 showed, (6.21) is a function of the geometry of the identified clusters, dependent directly on their diameter and centroid separation, and one that is designed to maximise at the optimal partition.

Figure 6.13 demonstrates how the function is affected by changes in fuzzy exponent and data sharpening threshold. As (6.21) shows, increasing the value of the data sharpening threshold will have the obvious effect of decreasing the radius of any clusters in the partition. Consequently, we should expect the value of the separation coefficient to tend towards unity in such cases, and this is clearly indicated in Figure 6.13, especially for the case of α=0.9.

As usual, the effect of increased fuzzy exponent is to reduce the magnitude of any local maximisation present in the function. In the extreme case, the separation coefficient will take on the maximum value of unity, and this tendency can clearly be seen in Figure 6.13.

One interesting aspect of the diagrams in Figure 6.13 associated with sharpening thresholds of α=0.0, α=0.5 and α=0.7 is the minor differences between these three cases. This indicates that the magnitude of r_i in (6.22) tends to be insensitive to sharpening, and the reason for this is easy to see: as $d(\mathbf{X}_k, \mathbf{V}_i)$ increases, so u_{ik} decreases.

Overall, the separation coefficient does not display any trends that allow an objective choice of either fuzzy exponent or sharpening threshold to be made. Indeed, on the basis of the diagrams presented in Figure 6.13, it is possible to identify any number of clusters as representative of the optimal partition, simply by choosing appropriate values for these parameters. Furthermore, a clear optimum at c=3 is not indicated by any of the diagrams in the Figure, and hence the function cannot be recommended for general use.

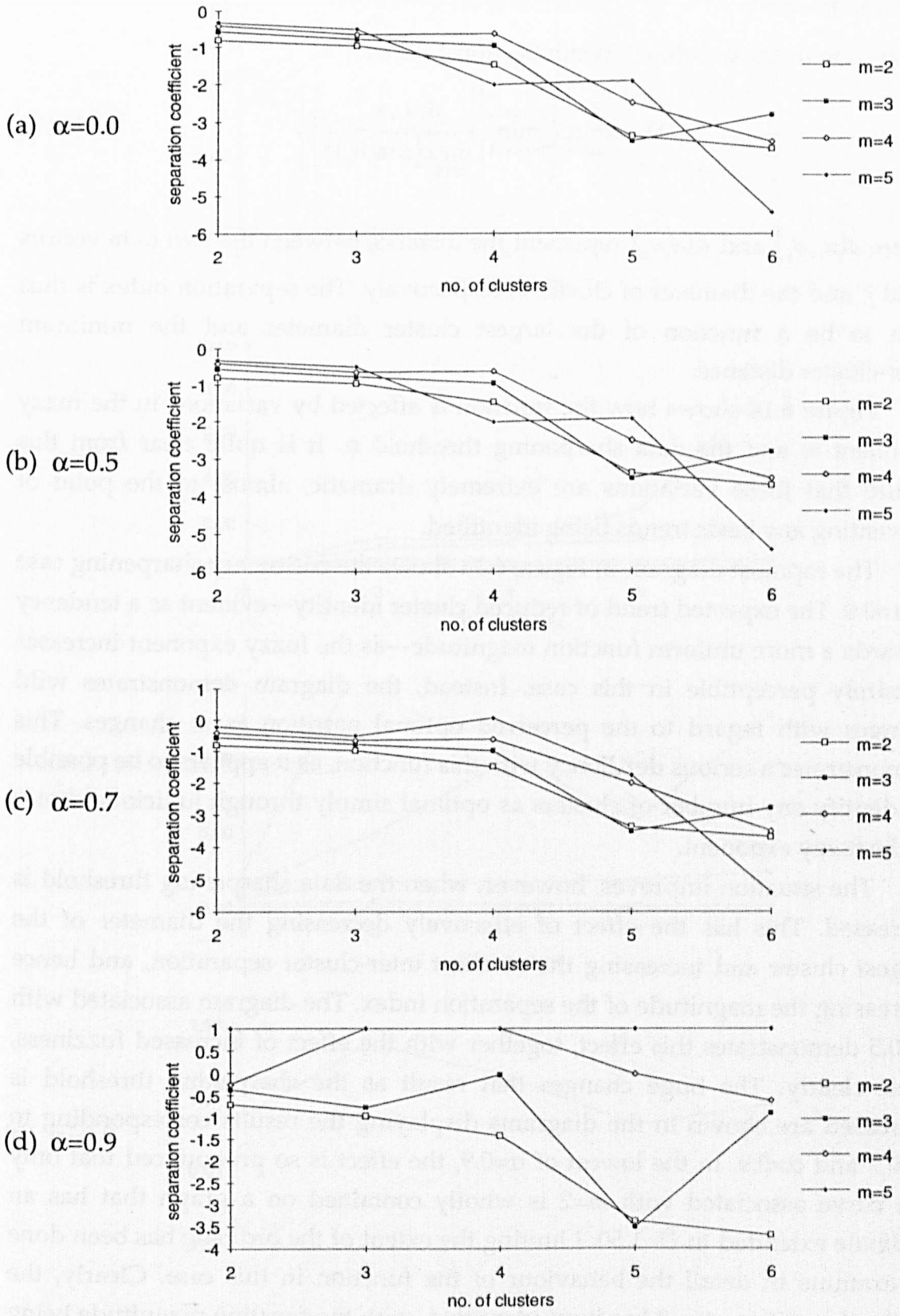


Figure 6.13 Porphyry data: separation coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Separation index

Separation index is defined in sub-Section 4.5.1 as

$$D = \min_{i+1 \le j \le n} \left\{ \min_{1 \le i \le n-1} \left\{ \frac{d(u_i, u_j)}{\max_{1 \le k \le c} \{\text{dia}(u_k)\}} \right\} \right\} \tag{6.24}$$

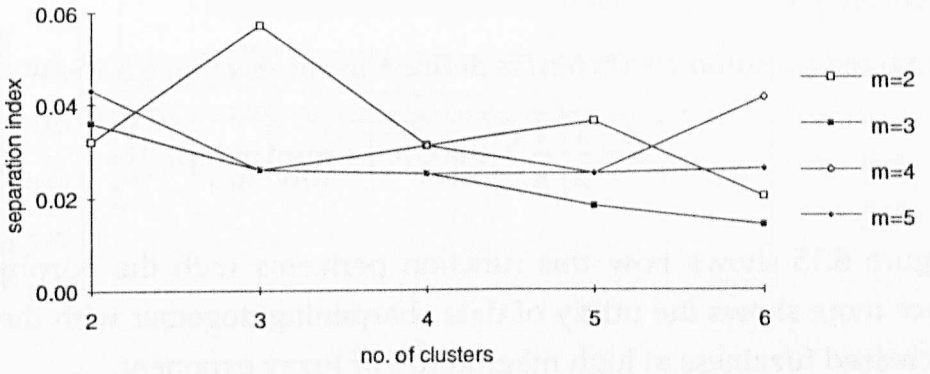
where $d(u_i, u_j)$ and $\text{dia}(u_k)$ represent the distance between the two data vectors i and j, and the diameter of cluster k, respectively. The separation index is thus seen to be a function of the largest cluster diameter and the minimum inter-cluster distance.

Figure 6.14 shows how the function is affected by variations in the fuzzy exponent m and the data sharpening threshold α. It is quite clear from this Figure that these variations are extremely dramatic, almost to the point of preventing any basic trends being identified.

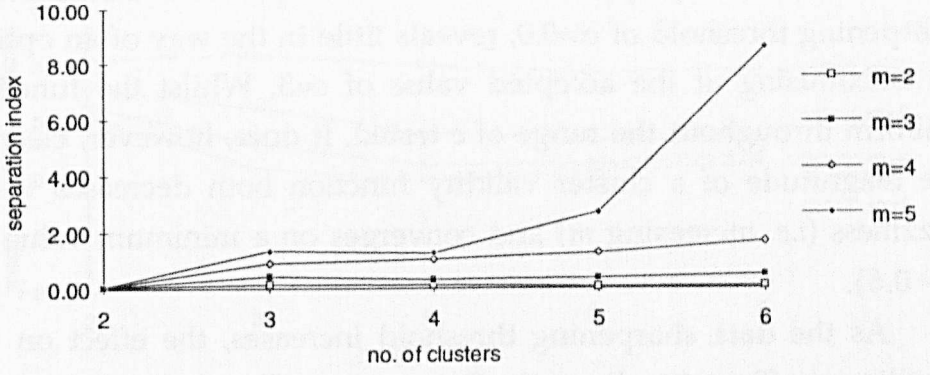
The topmost diagram in Figure 6.14 shows the customary sharpening case of α=0.0. The expected trend of reduced cluster identity—evident as a tendency towards a more uniform function magnitude—as the fuzzy exponent increases is barely perceptible in this case. Instead, the diagram demonstrates wild changes with regard to the perceived optimal partition as m changes. This demonstrates a serious deficiency with this function, as it appears to be possible to identify any number of clusters as optimal simply through judicious choice of the fuzzy exponent.

The situation improves, however, when the data sharpening threshold is increased. This has the effect of effectively decreasing the diameter of the largest cluster and increasing the smallest inter-cluster separation, and hence increasing the magnitude of the separation index. The diagram associated with α=0.5 demonstrates this effect, together with the effect of increased fuzziness, most clearly. The huge changes that result as the sharpening threshold is increased are shown in the diagrams displaying the results corresponding to α=0.7 and α=0.9. In the lowest of α=0.9, the effect is so pronounced that only the curve associated with m=2 is wholly contained on a graph that has an ordinate extended to D=1.50. Limiting the extent of the ordinate has been done to examine in detail the behaviour of the function in this case. Clearly, the optimal partition at c=3 has been identified, with the function magnitude being comparable to that for the non-sharpened case.

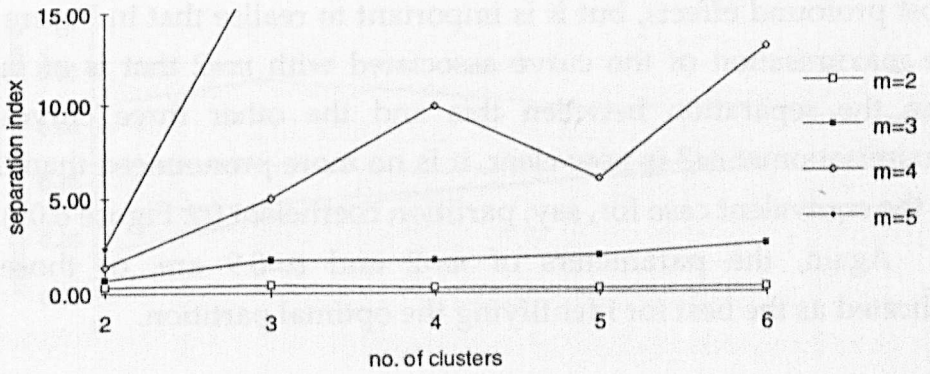
(a) α=0.0



(b) α=0.5



(c) α=0.7



(d) α=0.9

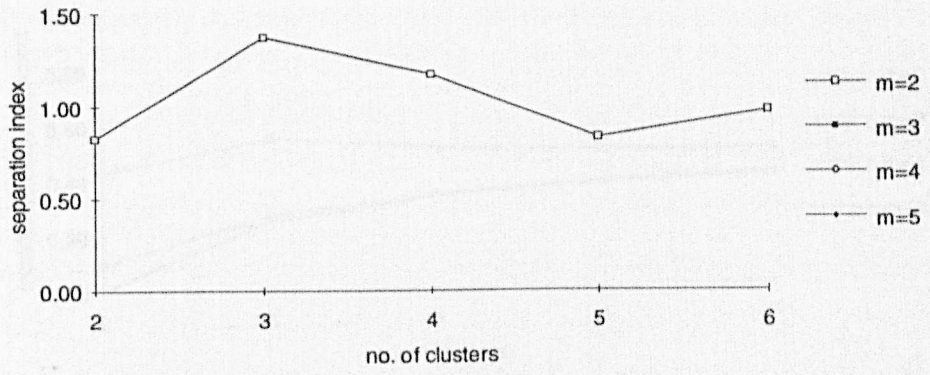


Figure 6.14 Porphyry data: separation index for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Modified partition coefficient

Modified partition coefficient is defined in sub-Section 4.5.1 as

$$\tilde{F} = \frac{1}{2}\left[\frac{1}{n}\sum_{j=1}^{n}\max_{1\le i\le c}\{u_{ij}\} + \min_{1\le j\le n}\left\{\max_{1\le i\le c}\{u_{ij}\}\right\}\right] \tag{6.25}$$

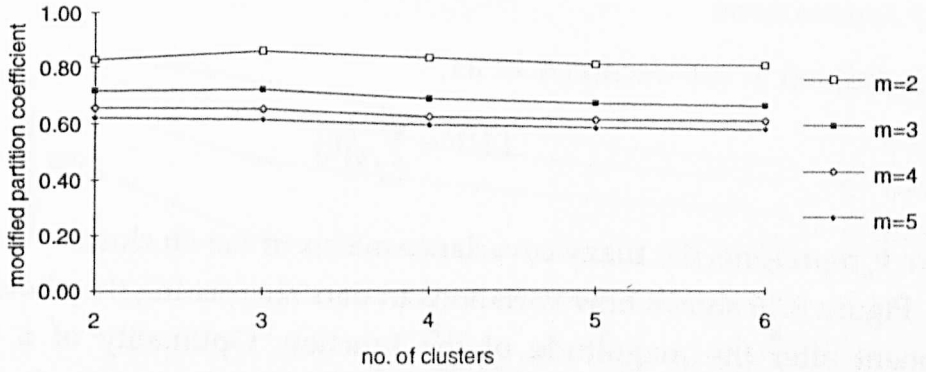
Figure 6.15 shows how this function performs with the porphyry data, and once more shows the utility of data sharpening, together with the tendency for increased fuzziness at high magnitudes of fuzzy exponent.

The customary application of modified partition coefficient, with a data sharpening threshold of α=0.0, reveals little in the way of an optimal partition by maximising at the accepted value of c=3. Whilst the function is almost uniform throughout the range of c tested, it does, however, clearly show how the magnitude of a cluster validity function both decreases with increasing fuzziness (*i.e.* increasing m) and converges on a minimum value (in this case, $\tilde{F} \approx 0.6$).

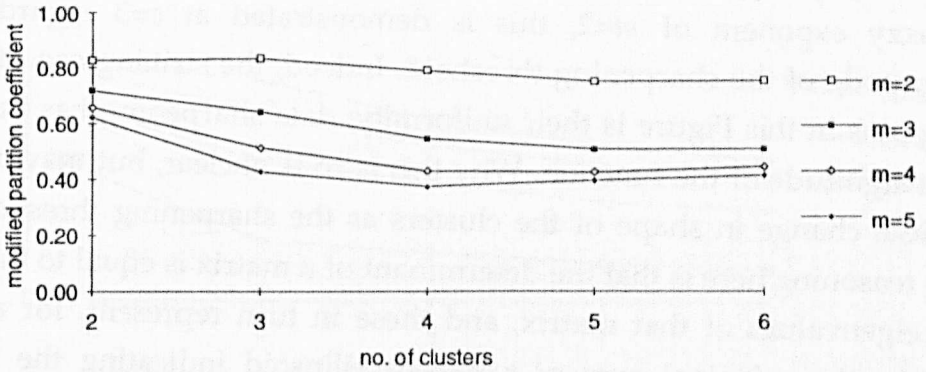
As the data sharpening threshold increases, the effect on the modified partition coefficient is dramatic. Once more, a threshold of α=0.9 produces the most profound effects, but it is important to realise that in Figure 6.15(d) , it is the maximisation of the curve associated with m=2 that is of interest, rather than the separation between this and the other three curves. Whilst the maximisation at c=3 is now clear, it is no more pronounced than that indicated by the equivalent case for, say, partition coefficient (*cf.* Figure 6.09).

Again, the parameters of m=2 and α=0.9 are, of those considered, indicated as the best for identifying the optimal partition.

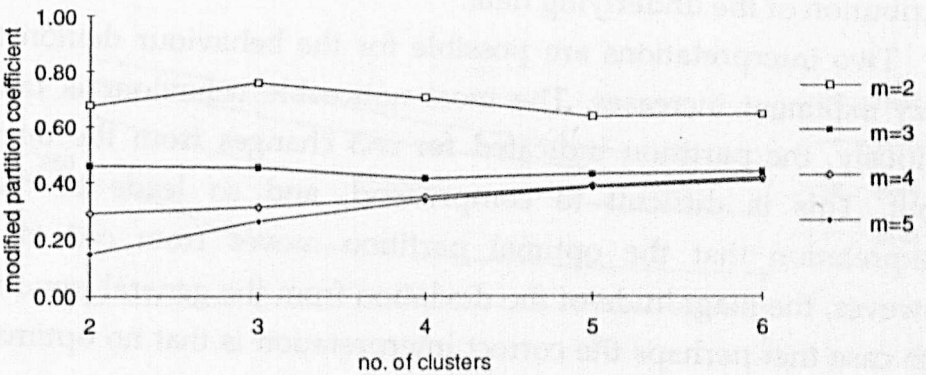
(a) α=0.0



(b) α=0.5



(c) α=0.7



(d) α=0.9

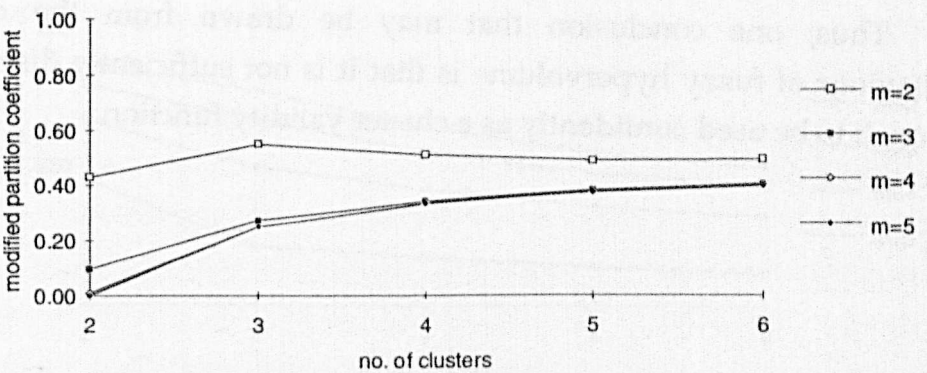


Figure 6.15 Porphyry data: modified partition coefficient for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Fuzzy hypervolume

This is defined in sub-Section 4.5.1 as

$$FHV = \sum_{i=1}^{c} \sqrt{|\mathbf{F}_i|} \tag{6.26}$$

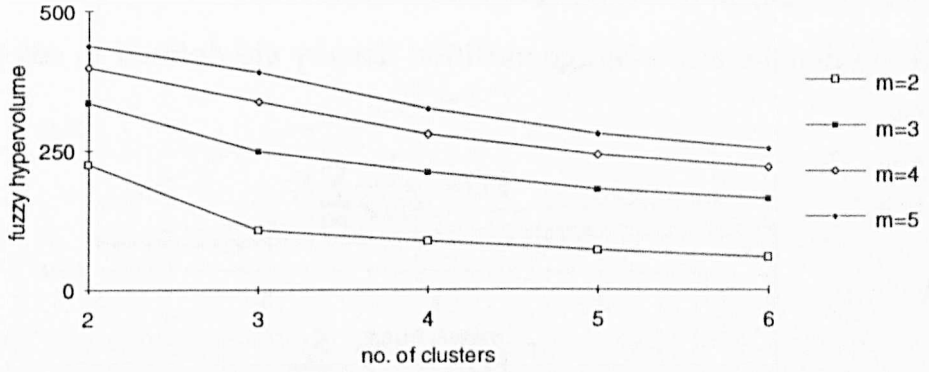
where $\mathbf{F}_i$ represents the fuzzy covariance matrix of the ith cluster.

Figure 6.16 shows how variations in data sharpening threshold and fuzzy exponent alter the magnitude of the function. Optimality of a partition is indicated by a (local) minimum in the function, and the Figure shows how, for a fuzzy exponent of m=2, this is demonstrated at c=3 regardless of the magnitude of the sharpening threshold. Indeed, the striking aspect of the four diagrams in this Figure is their uniformity; data sharpening has little effect on the magnitude of the function. Why this is so is unclear, but may be indicative of little change in shape of the clusters as the sharpening threshold changes. The reasoning here is that the determinant of a matrix is equal to the product of the eigenvalues of that matrix, and these in turn represent, for a covariance matrix, the principal axes of a best-fit ellipsoid indicating the form of the distribution of the underlying data.

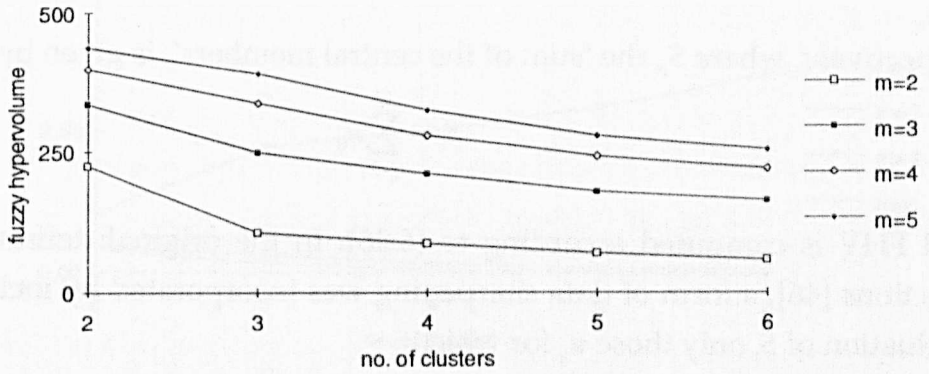
Two interpretations are possible for the behaviour demonstrated as the fuzzy exponent increases. The most noticeable behaviour is that, somewhat curiously, the partition indicated for c=3 changes from the optimum to the worst. This is difficult to comprehend, and so leads to the alternative interpretation that the optimal partition moves from c=3 to that at c=2. However, the magnitude of the deviation from the general trend is so small in each case that perhaps the correct interpretation is that no optimal partition is indicated at all.

Thus, one conclusion that may be drawn from the demonstrated behaviour of fuzzy hypervolume is that it is not sufficiently discriminatory to allow it to be used confidently as a cluster validity function.

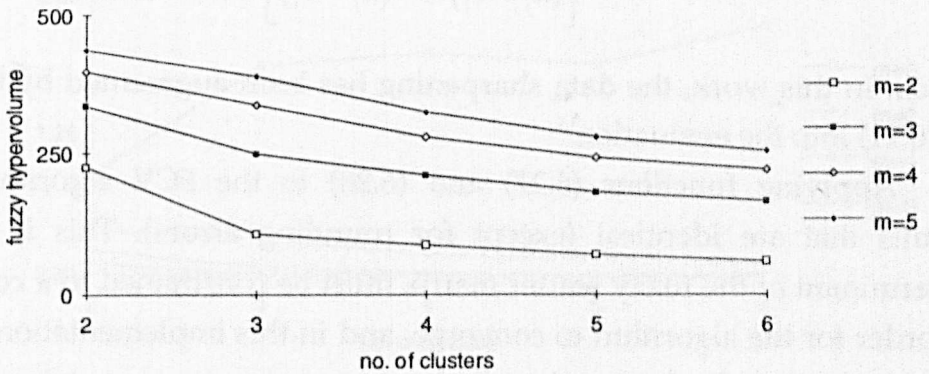
(a) α=0.0



(b) α=0.5



(c) α=0.7



(d) α=0.9

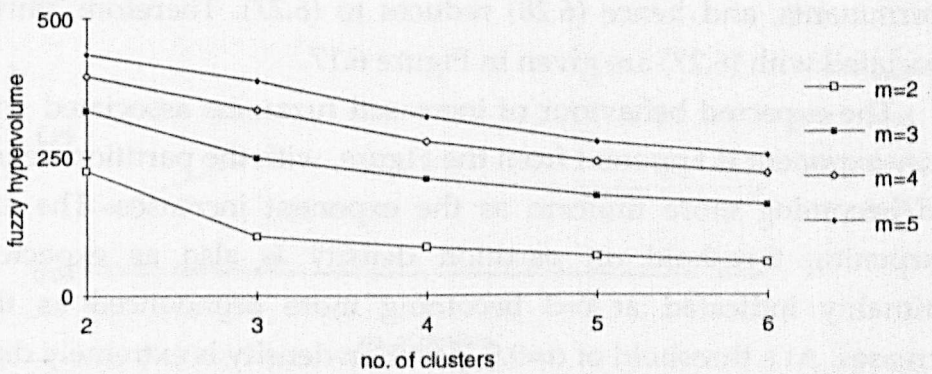


Figure 6.16 Porphyry data: fuzzy hypervolume for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Partition density and average partition density

Partition density and average partition density are defined in sub-Section 4.5.1 as

$$PD = \frac{1}{FHV}\sum_{i=1}^{c} S_i \tag{6.27}$$

and

$$PDA = \frac{1}{c}\sum_{i=1}^{c} \frac{S_i}{\sqrt{|\mathbf{F}_i|}} \tag{6.28}$$

respectively, where S_i, the 'sum of the central members', is given by

$$S_i = \sum_{j=1}^{n} u_{ij} \tag{6.29}$$

and FHV is computed according to (6.26). In the original definition of these functions [46], a form of data sharpening was incorporated by including in the evaluation of S_i only those u_{ij} for which

$$\left[\left(\mathbf{x}_j - \mathbf{v}_i\right)^T \mathbf{F}^{-1}\left(\mathbf{x}_j - \mathbf{v}_i\right)\right] < 1 \tag{6.30}$$

holds. In this work, the data sharpening has been augmented by the inclusion of (6.11) into the evaluation.

Applying functions (6.27) and (6.28) to the FCV algorithm produces results that are identical (except for rounding errors). This is because the determinant of the fuzzy scatter matrix must be constrained to a constant value in order for the algorithm to converge, and in this implementation of FCV that value is unity. Consequently, all the fuzzy covariance matrices have equal determinants, and hence (6.28) reduces to (6.27). Therefore, only the graphs associated with (6.27) are given in Figure 6.17.

The expected behaviour of increased fuzziness associated with increased fuzzy exponent is apparent from the Figure, with the partition density reducing and becoming more uniform as the exponent increases. The effect of data sharpening threshold on partition density is also as expected, with the optimality indicated at c=3 becoming more pronounced as the threshold increases. At a threshold of α=0.9, partition density is extremely discriminatory, and could be regarded as the best of the functions tested here, on the basis of relative change in magnitude from one partition to another.

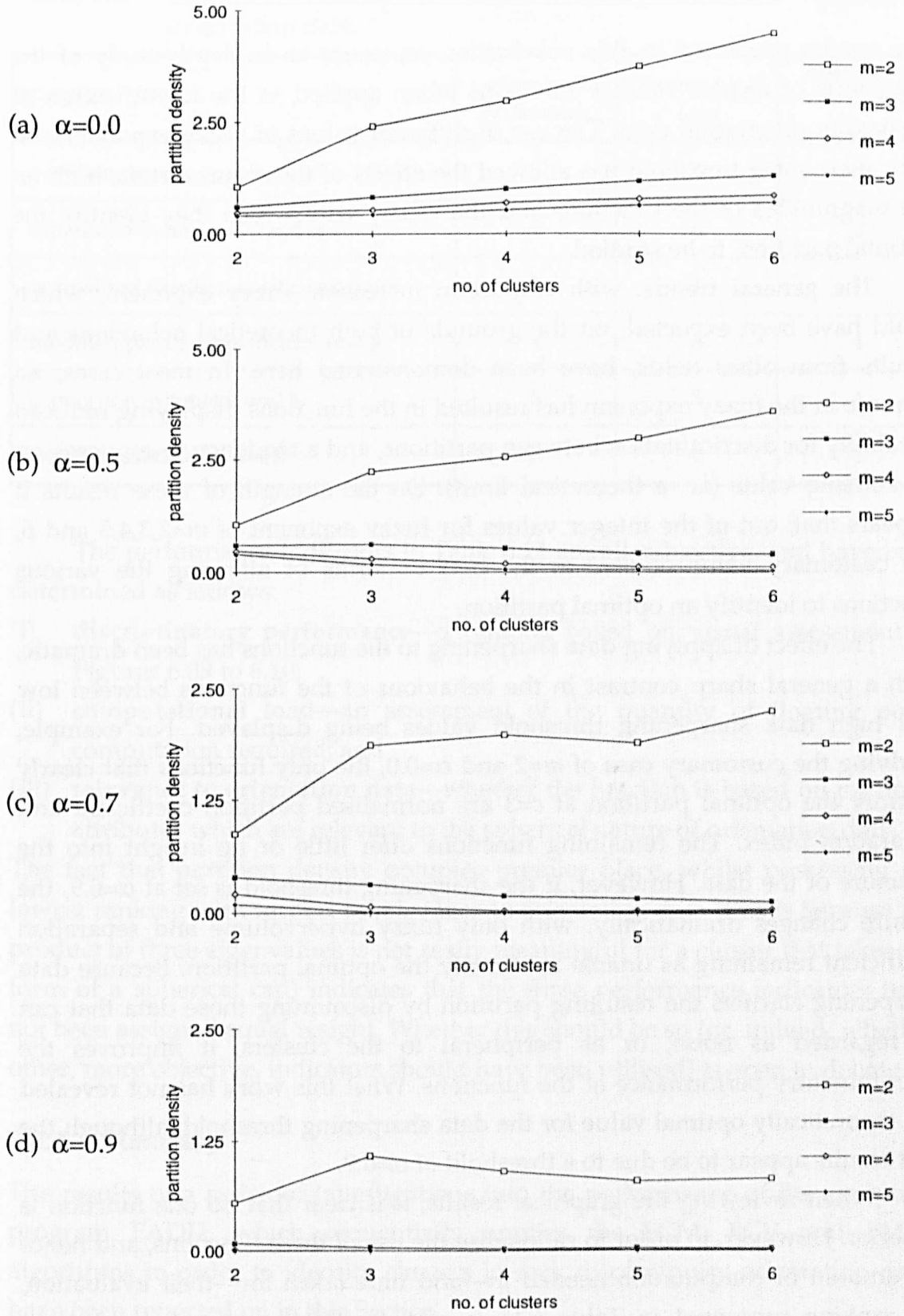


Figure 6.17 Porphyry data: partition density for FCV partitioning with α=0.0, 0.5, 0.7, 0.9 and m=2, 3, 4, 5.

Review

The results presented in this sub-Section represent an in-depth study of the behaviour of cluster validity functions when applied to the identification of clusters in orientation data. The use of different values of fuzzy exponent and data sharpening threshold has allowed the effects of these parameters, both on the magnitudes of the functions and the clarity with which they identify the optimal partition, to be studied.

The general trends, with respect to increasing fuzzy exponent, which could have been expected, on the grounds of both theoretical behaviour and results from other fields, have been demonstrated here. In most cases, an increase in the fuzzy exponent has resulted in the functions displaying reduced capability for discrimination between partitions, and a tendency to converge on an extreme value (*i.e.* a theoretical limit). On the strength of these results it appears that, out of the integer values for fuzzy exponent of m=2,3,4,5 and 6, the customary value of m=2 is the best in terms of allowing the various functions to identify an optimal partition.

The effect of applying data sharpening to the functions has been dramatic, with a general sharp contrast in the behaviour of the functions between low and high data sharpening threshold values being displayed. For example, studying the customary case of m=2 and α=0.0, the only functions that clearly identify the optimal partition at c=3 are normalised partition coefficient and separation index. The remaining functions offer little or no insight into the structure of the data. However, if the sharpening threshold is set at α=0.9, the picture changes dramatically, with only fuzzy hypervolume and separation coefficient remaining as unable to identify the optimal partition. Because data sharpening clarifies the resulting partition by discounting those data that can be regarded as noise, or as peripheral to the clusters, it improves the discriminatory performance of the functions. What this work has not revealed is a theoretically optimal value for the data sharpening threshold, although the best results appear to be due to a threshold of α=0.9.

When reviewing the graphical results, it is clear that no one function is superior. However, in order to rationalise the use of these functions, and hence the amount of computation needed in—and time taken for—their evaluation, the ranking presented in Table 6.11 is offered. This ranking is applied in Chapter 7 to help determine the apparent number of discontinuity sets in a rock mass.

Table 6.11 Empirical ranking of cluster validity functions applied to orientation data.

Function	*Discriminatory Performance*	*Computational Load*	*Relevance to Orientation Data*
partition density, α=0.9	1	4	6
separation index, α=0.0 and α=0.9	2	6	3
normalised partition coefficient, α=0.9	3	1	1
modified partition coefficient, α=0.9	4	3	4
proportion exponent, α=0.9;	5	5	4
partition coefficient, α=0.9	6	1	1

The performance indicators in Table 6.11 are all subjective, and have been determined as follows:

(i) **discriminatory performance**—a ranking based on visual assessment of Figures 6.09 to 6.10;

(ii) **computational load**—an assessment of the quantity of floating point computation required; and

(iii) **relevance to orientation data**—whether the function is based on partition attributes which are relevant to the spherical nature of orientation data.

The fact that partition density occupies premier place, whilst possessing the lowest ranking with respect to relevance to orientation data (this is because the product of three eigenvalues is not really meaningful for a cluster that takes the form of a spherical cap) indicates that the three performance indicators have not been assigned equal weight. Whether this should be so (or, indeed, whether other, more objective, indicators should have been utilised) is open to debate.

6.3.4 Summary

The results of a series of investigations into the performance of the computer program FADD, which sequentially applies the FCM, FCV and FMLE algorithms in order to identify clusters in rock discontinuity orientation data, have been reported on in this Section.

The basic efficacy of the scheme has been demonstrated through testing with two synthetic and one natural data sets. The synthetic data sets were designed such that one consisted of well-separated isotropic clusters, and the

other contained a highly anisotropic cluster. The natural data set is noisy, in that clearly defined clusters are absent. This work demonstrates that:

(i) the numerical scheme will identify clusters in orientation data, and will cope with clusters consisting of different numbers of data disposed with different levels of sparsity;

(ii) whilst the FCM algorithm will tend to identify isotropic clusters, the succeeding FCV and FMLE algorithms identify inherent anisotropic clusters in the data;

(iii) the resulting clusters correspond with known clusters in the data; and

(iv) for each cluster, the eigenvalues and eigenvectors of the fuzzy scatter matrix may be used to determine the mean orientation of the cluster, the orientation of the principal axes of the anisotropy and the shape of the cluster itself.

A study of the nature of the convergence of the FCM and FCV algorithms has revealed that, whilst a criterion based on centroid change of orientation may appear appropriate when attempting to partition orientation data, the customary change-of-membership criterion must be used in order to guarantee convergence to the algorithmically correct partition for a particular analysis.

Finally, a comprehensive study has been undertaken using the copper porphyry data to assess the efficacy with which a number of cluster validity functions identify the optimal partition in orientation data. This study has revealed how a fuzzy exponent of m=2 appears to produce the most clearly defined partitions, and that under such circumstances the discriminatory performance of the cluster validity functions is greatly enhanced when a sharpened partition is used. Consequently, six functions have been subjectively selected and ranked in order of the efficacy with which they are able to identify the optimal partition. This ranking is: partition density; separation index; normalised partition coefficient; modified partition coefficient; proportion exponent and partition coefficient. A data sharpening threshold of α=0.9 has been revealed to offer the greatest enhancement for all functions.

Once the optimal partition in orientation data has been identified through the use of cluster validity functions, it is essential to be able to graphically present the partitioned data in a way that allows various aspects of the quality of the partition to be assimilated. This is the subject of the next Section.

6.4 Graphical Presentation of Partitioning Results for Orientation Data

Whilst orientation data are inherently spherical, and thus could be profitably presented on the surface of a spherical object, the need to use two-dimensional media (*i.e.* sheets of paper) for the dissemination of information means that the graphical presentation of orientation data is customarily undertaken using hemispherical projection techniques.

The aim of this work is to develop an improved form of analysing orientation data through the application of partitioning algorithms, rather than a wholly new form of state-of-the-art graphical presentation, and so the graphical output from FADD has been limited to that which is adequate to portray the results of a partitioning analysis. Consequently, the customary hemispherical projection has been adopted throughout. Furthermore, simple 'line-printer' plots have been adopted, whereby an array of monospaced text is used to represent the area of the projection, with various characters chosen to represent the orientation data and the projection boundary Examples of this form of output have already been presented in Figures 6.04, 6.05 and 6.06. Whilst this leads only to low resolution plots, they possess the advantages that they:

(i) are quick and easy to produce, requiring only text handling procedures from a programming language;
(ii) require no special output devices, being completely device independent;
(iii) are easily transferred to paper using any type of printer; and
(iv) may simply be incorporated into a document produced using a text processing program (as, for example, in this document).

In this Section, these hemispherical projection plots are used to display two forms of graphical partitioning output: **cluster assignment plots**, which show the general form of the partition; and **membership level plots**, which show the specific form of the individual clusters in the partition.

6.4.1 Cluster assignment plots

The concept of the nearest maximum membership partition was discussed in sub-Section 4.6.1, as a means of addressing the paradox that the results of a fuzzy partitioning analysis (in which each datum is considered to be concurrently a member of all clusters in the partition) are most easily understood on the basis that each datum is physically a member of only one

cluster. It is a straightforward matter to plot the nearest maximum membership partition: we simply determine in which cluster a datum possesses greatest membership value, and then plot at the relevant location on the projection a symbol representing this cluster. As examples, Figures 6.18 and 6.19 show, respectively, the nearest maximum membership partition for c=3 resulting from the FCM and the FCV algorithms.

The advantage of these **cluster assignment plots** lies in their simplicity: each cluster centroid is represented by its associated numeral, and the data

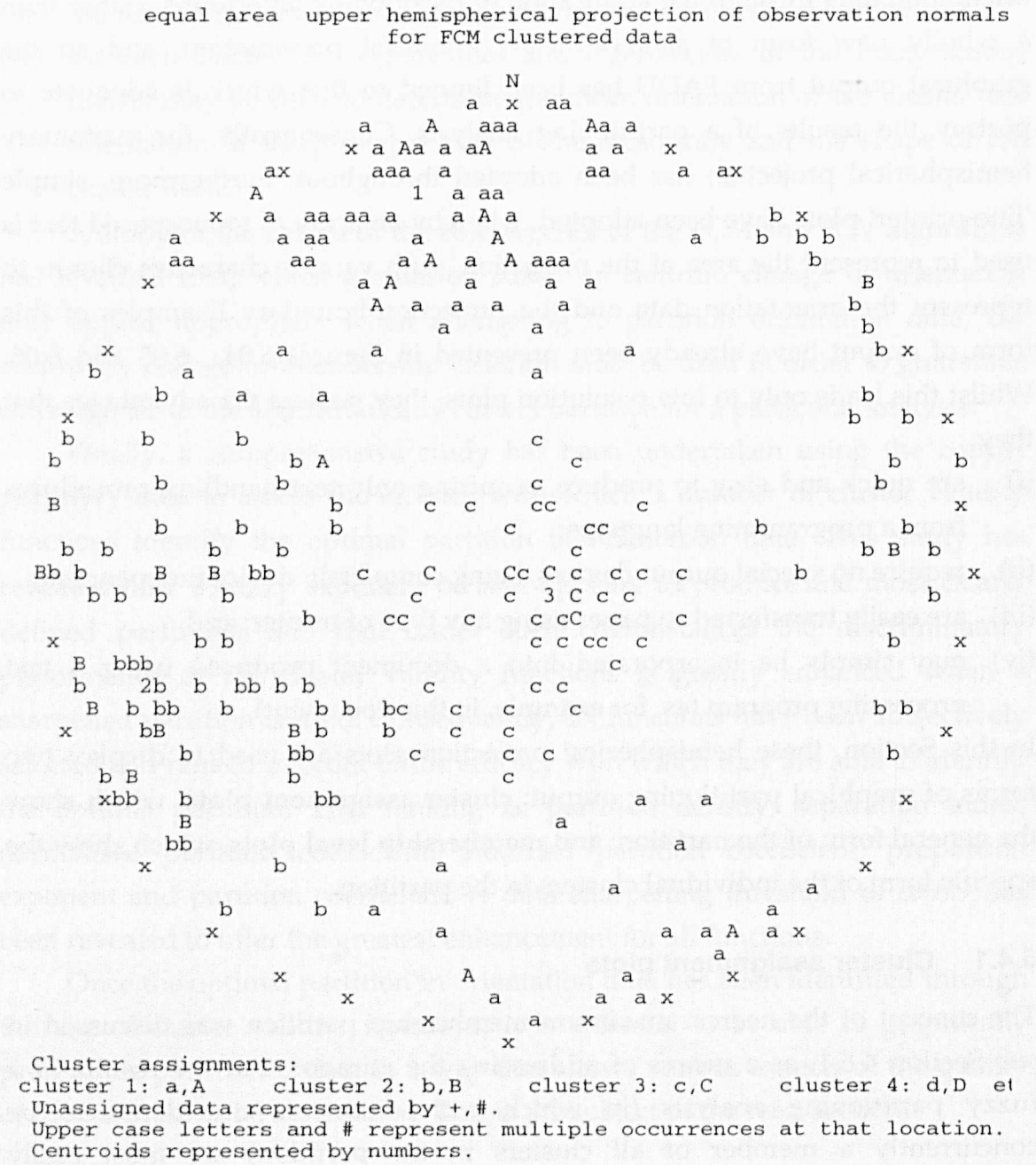


Figure 6.18 Porphyry data: cluster assignment plot for FCM partitioning with m=2 and c=3.

assigned to that cluster are represented by the corresponding letter of the alphabet, with lower case letters used when a single datum falls within an element of the projection, and upper case letters used otherwise. The result is a plot that immediately conveys the key information regarding the locations of clusters and their extent—it is clear from Figures 6.18 and 6.19 where the cluster centroids are located, and how the data have been partitioned.

A criticism of the plots is that they are of limited resolution, with the result that data will not plot at their true projected position. This is true, but it

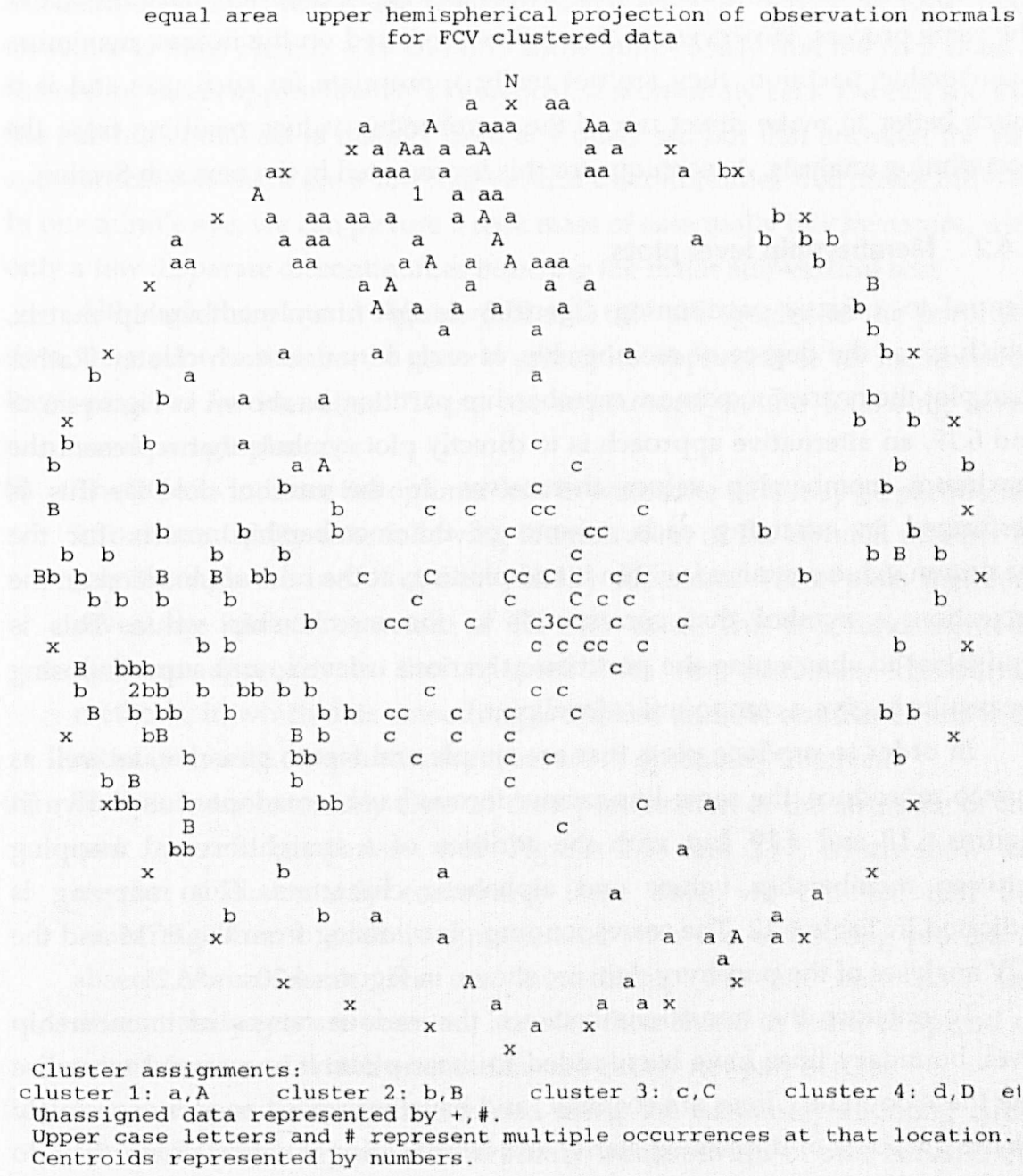


Figure 6.19 Porphyry data: cluster assignment plot for FCV partitioning with m=2 and c=3.

must be remembered that these diagrams are the result of a mathematical analysis, and so results to whatever accuracy are readily available as numerical, rather than graphical, output from this analysis. The aim of these diagrams is to simply convey, in a well-known and easily assimilable format, the overall appearance of the partition. To this end, they perform admirably.

Of course, it is possible to study these diagrams in greater depth. It is easy to compare Figures 6.18 and 6.19, and find data that have changed cluster assignments in the course of moving from the FCM partition to the FCV partition, and to determine by how much the cluster centroids have moved in the same process. However, as these plots are based on the nearest maximum membership partition, they are not really appropriate for such use, and it is much better to make direct use of the membership values resulting from the partitioning analysis. A technique for this is presented in the next sub-Section.

6.4.2 Membership level plots

Central to a fuzzy partitioning algorithm is the fuzzy membership matrix, which gives the degree of membership of each datum in each cluster. Rather than plot the nearest maximum membership partition, as shown in Figures 6.18 and 6.19, an alternative approach is to directly plot symbols that represent the maximum membership values themselves. In the case of FADD, this is performed by scanning each column of the membership matrix for the maximum value contained within it and plotting, at the relevant location on the projection, a symbol that corresponds to the membership value. This is equivalent to sharpening the partition at various α-levels, and superimposing the results to give a compound α-level plot.

In order to produce plots that are simple and fast to generate, as well as easy to reproduce, the same line-printer format has been adopted as shown in Figures 6.18 and 6.19, but with the addition of a straightforward mapping between membership values and alphabetic characters. This mapping is indicated in Table 6.12. The corresponding plots arising from the FCM and the FCV analyses of the porphyry data are shown in Figures 6.20 and 6.21.

To enhance the transitions between the various ranges of membership level, boundary lines have been added to these plots. It is essential to realise that these boundary lines are cosmetic, and have been positioned by eye solely for the purposes of improving clarity and hence aiding interpretation—they do not represent analytically determined contours.

Table 6.12 Mapping scheme for membership level plots.

membership value	0.0 <u≤ 0.1	0.1 <u≤ 0.2	0.2 <u≤ 0.3	0.3 <u≤ 0.4	0.4 <u≤ 0.5	0.5 <u≤ 0.6	0.6 <u≤ 0.7	0.7 <u≤ 0.8	0.8 <u≤ 0.9	0.9 <u≤ 1.0
printed characters	j,J	i,I	h,H	g,G	f,F	e,E	d,D	c,C	b,B	a,A

Presenting discontinuity orientation data in this way is not only completely novel, but—as a comparison with Figures 2.16 and 2.17 immediately demonstrates—is of striking and uncomplicated elegance. Furthermore, interpreting membership level plots in terms of rock mass structure is easy. Figures 6.18 and 6.19 show quite clearly that the rock mass is formed of three, approximately orthogonal, discontinuity sets. We can see that the sub-horizontal set is distinct from any other set, but that between the two sub-vertical sets there are a few sub-vertical discontinuities that strike NE-SW. In our mind's eye, we can picture a rock mass of essentially blocky nature, with only a few disparate discontinuities bisecting the major sub-vertical sets.

Although the plots in Figures 6.20 and 6.21 are specific to the porphyry data, they allow a number of general principles applicable to all membership level plots to be identified. These are expounded in the following seven numbered paragraphs.

(i) These plots indicate the optimal way in which the data may be partitioned into discontinuity sets, following identification of the optimal number of sets through the use of cluster validity functions. These plots therefore reveal the intrinsic structure of the rock mass. This is a fundamentally different approach to that adopted with the customary contouring methods, in which data smoothing is applied and the number of sets then estimated, with no quantitative measure of optimality involved.

(ii) The plots are extremely clear in their presentation of the structure of the rock mass. In comparison with Figures 2.16 and 2.17, which show the results of customarily contouring these data, we can see that the 'smearing' due to smoothing associated with the contouring process is absent. As a result, we can immediately see not only how the discontinuity sets are oriented, but also their shape in terms of spread of orientations, and which orientations are clearly not part of a set.

(iii) In contrast to contours determined using the customary counting circle approach, which represent an arbitrary measurement of pole density, the boundary lines in a membership level plot approximately represent

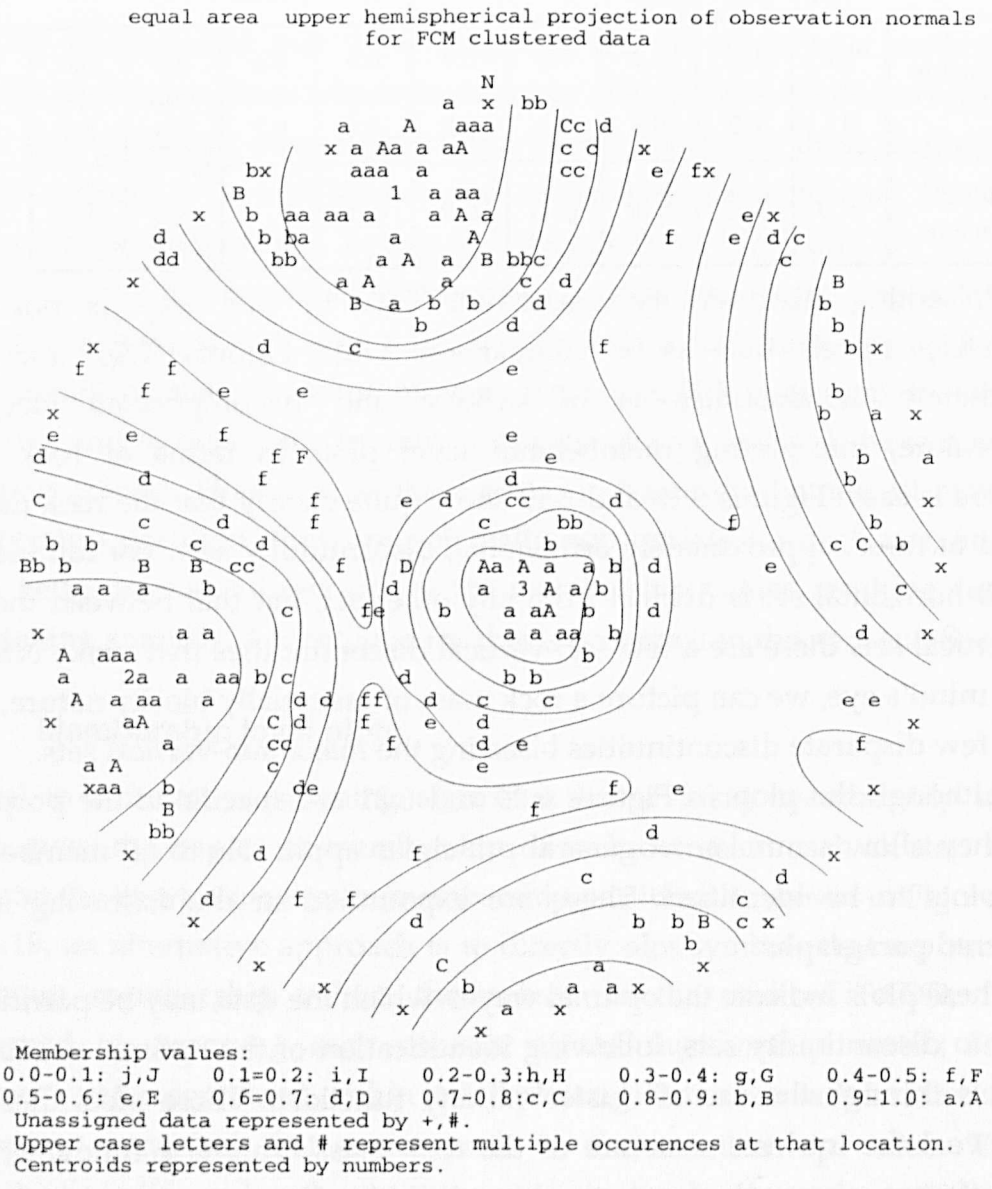


Figure 6.20 Porphyry data: membership level plot for FCM partitioning with m=2 and c=3.

contours of 'degree of certainty'. Degree of certainty is immediately applicable in an engineering context, in that we can choose the angular extent of a set in terms of acceptable levels of uncertainty. As an example, if we wish to sample from a discontinuity such that we are 90% or more certain that it is a member of a particular set, we are able to directly set orientation limits in order that this will be the case.

(iv) The minimal membership values are in the range 0.4 to 0.5, and the corresponding data are distributed in long, narrow regions between the areas of high membership. These low membership values indicate that the corresponding data are more likely to be 'noise' than they are members of

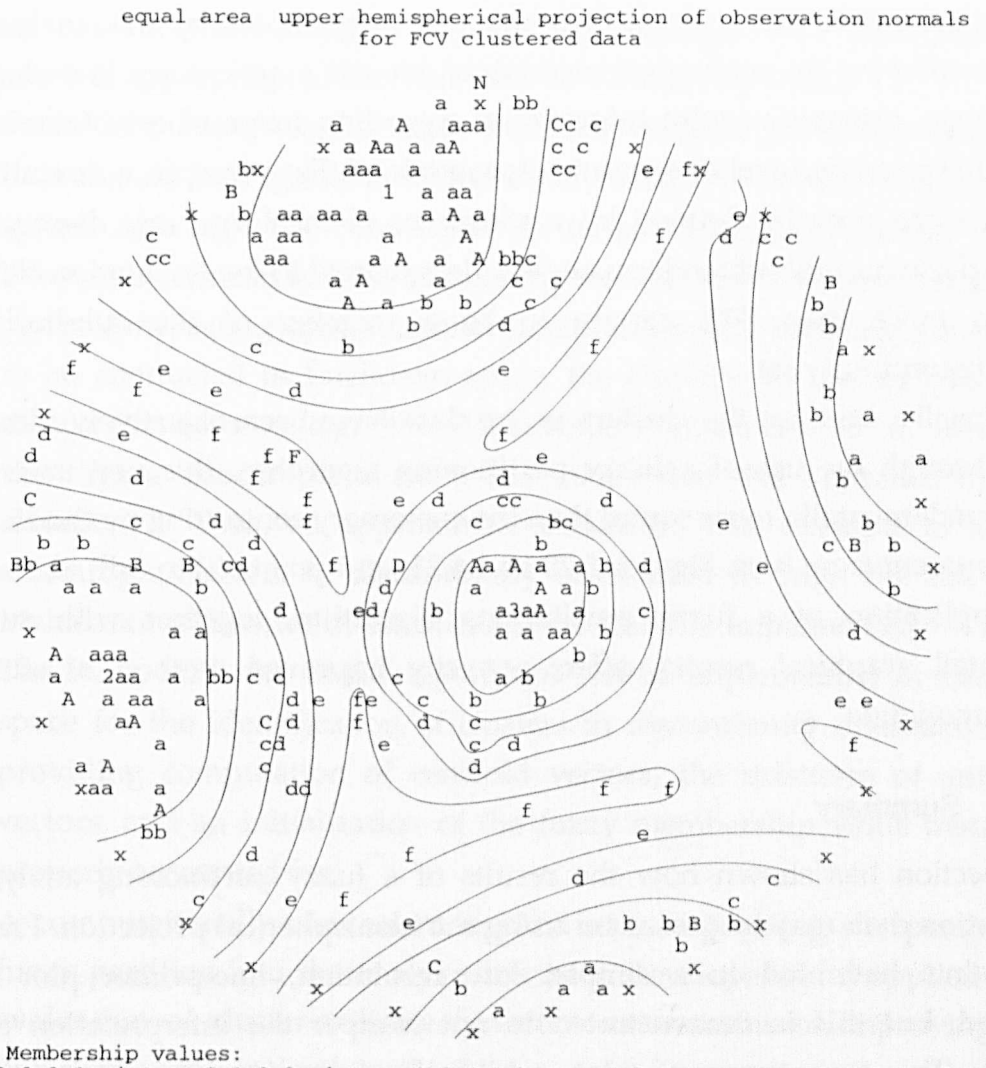


Figure 6.21 Porphyry data: membership level plot for FCV partitioning with m=2 and c=3.

a discontinuity set. Indeed, these data can only be assigned to the adjacent discontinuity sets if extremely low levels of certainty are acceptable.

(v) For any given membership level, the area of the projection covered by a cluster (the plots are equal area) gives a relative indication of the variability of discontinuity orientation within the corresponding discontinuity set. Thus, in the context of the copper porphyry data, we can state categorically that the discontinuities in the sub-vertical sets demonstrate a much greater degree of variability in orientation than those in the sub-horizontal set.

(vi) A new definition of a discontinuity set is prompted by this analysis. By employing the concept of uncertainty, we can augment our knowledge of mean orientation with information regarding range of orientations such that a certain level of certainty is maintained. For example, a discontinuity set can now be defined by a statement of the form 'any discontinuity whose normal subtends an angle of less than 5° to the orientation 105°/85° is more than 80% certain to be a member of the sub-horizontal discontinuity set'.

(vii) Finally, because the clusters in the data have been objectively identified through the use of a fuzzy partitioning algorithm, this new method is fundamentally more sound than the customary contouring methods.

It is apparent from both Figures 6.20 and 6.21, and points (i) to (vii) above, that the application of a fuzzy partitioning algorithm, together with suitably presented graphical results, offers a vastly improved method of studying orientation data.

6.4.3 Summary

This Section has shown how the results of a fuzzy partitioning analysis of orientation data may be presented using the hemispherical projection. Practical constraints have led to a simple, low resolution, line-printer plot being adopted, but this is demonstrated to not hamper the interpretation of the results. The two types of plot exhibited—cluster assignment plots and membership level plots—show, respectively, how the data are distributed on the basis of the nearest maximum membership, and the certainty with which the data may be regarded as cluster members. Cluster assignment plots clearly indicate how the data are disposed within the clusters, but it is membership level plots that most dramatically show the size and shape of the clusters, and hence the geometry of the corresponding rock mass.

6.5 Synopsis

This Chapter has demonstrated how fuzzy partitioning algorithms may be applied to the analysis of rock mass geometry. The Author has developed a computer program—FADD (Fuzzy Analysis of Discontinuity Data)—that implements the FCM, FCV and FMLE algorithms for the case of discontinuity orientation data. This computer program, in conjunction with theoretical

studies, has allowed a number of important findings to be made, and as a synopsis it is appropriate to list them here as numbered points.

(i) From the point of view of analytical procedures, discontinuity orientation data should be regarded as two-dimensional data within a Riemannian space. Although the connection (in terms of angles between vectors) between Euclidean and Riemannian spaces has been identified as a means of accommodating computational practicalities and allowing the analysis to be conducted in Euclidean space, the inability to use an anisotropic distance measure (e.g. the Mahalanobis distance) as a means of identifying anisotropic clusters has meant that this approach must be abandoned. Thus, an approximate procedure that attempts to identify anisotropic clusters in discontinuity orientation through the use of the Mahalanobis distance in Euclidean space must be implemented.

(ii) The FCM, FCV and FMLE algorithms can be implemented in Euclidean space for the identification of clusters in discontinuity orientation data, providing computation of centroid vectors, the existence of antipodal vectors, and an initialisation of the fuzzy membership value matrix are properly accounted for.

(iii) A computationally convenient method of incorporating sharpening of a fuzzy partition into the analysis has been developed, and applied to the evaluation of cluster validity functions as a means of improving their detection of the optimal partition.

(iv) Using two synthetic and one natural data sets, it has been demonstrated how the sequential application of the FCM, FCV and FMLE algorithms will correctly identify clusters, regardless of the number of data within a cluster, their density, the antipodal nature of the data, the anisotropic shape of the cluster or the noisiness of the data. In addition, the orientations of the resulting clusters are correctly determined—not only in terms of mean orientation, but also the alignment of the principal axes of anisotropic clusters—as are cluster shape classification parameters.

(v) Convergence of the algorithms to a stable partition has been demonstrated, and the use of the customary convergence criterion, *i.e.* maximum change in membership value, has been confirmed as being more appropriate than a criterion based on change in orientation of cluster centroid vector.

(vi) An extensive study of the efficacy with which various cluster validity functions identify the optimal partition has been undertaken, using the

copper porphyry data of Mahtab *et alia* [86]. This has revealed that the customary value of m=2 for the fuzzy exponent, together with data sharpening at the α=0.9 level, gives the best results.

(vii) On the basis of the study of the cluster validity functions, a ranking of efficacy in terms of identifying the optimal partition has been proposed. This ranking is: partition density; separation index; normalised partition coefficient; modified partition coefficient; proportion exponent and partition coefficient.

(viii) Two types of hemispherical projection plot have been proposed for the graphical presentation of the partitioning results. The first of these, cluster assignment plots, clearly show how the data are formed into the various clusters. The second of these, membership level plots, dramatically shows the size and shape of the clusters, and hence the geometry of the corresponding rock mass.

(ix) Fuzzy partitioning algorithms, in conjunction with cluster assignment plots and membership level plots, are seen to be advantageous, with respect to the customary methods of identifying clusters in orientation data, in a number of ways. These are:

a) the optimal number of discontinuity sets is determined objectively, rather than a subjective number of discontinuity sets being found;

b) the 'smearing' due to customary contouring is absent, giving a much clearer view of the disposition of the data;

c) in contrast to customary contours, which represent an arbitrary measurement of pole density, the membership values on the projection represent certainty of discontinuity set membership;

d) low membership values indicate directly those data that should be regarded as noise, rather than discontinuity set members;

e) for any given membership level, the amount of an equal area projection covered by a cluster gives an indication of the variability of discontinuity orientation within the corresponding discontinuity set; and

f) on the strength of such an analysis, discontinuity sets can be defined in terms of level of certainty relative to a mean orientation.

Finally, the computer program FADD has been used, in conjunction with the photogrammetric procedures described in the previous Chapter, to analyse rock mass geometry at two sites in the United Kingdom. The results of this analysis are presented in the next Chapter.

7 Application of the New Techniques

As a means of assessing the efficacy of the photogrammetric techniques and fuzzy clustering analysis developed as part of this work, and described in Chapters 5 and 6, tests were undertaken at two sites—one a limestone quarry, the other a granodiorite quarry—in the UK. The results arising from these tests are presented in this Chapter. The tests involved both a photogrammetric survey and a customary orientation survey using a geological compass at each of the sites.

This Chapter consists of four Sections. General details of the equipment used and procedures followed during the work are given in Section 7.1, and this is followed in Sections 7.2 and 7.3 with the results stemming from the use of the system at the quarry sites. Also in these Sections, details are given of the particular survey and photogrammetric configurations adopted, together with copies of the stereoscopic photographic prints used in the analyses, coordinates of the control markers, and measured photograph coordinates and computed orientations of the discontinuities imaged in each stereopair. The orientation data collected with the geological compass are also given for comparison. The results of these surveys are presented as lower hemispherical projection cluster assignment plots and membership level plots (produced using the Author's program FADD), together with an assessment of the optimal partition through the application of cluster validity functions. The compass survey data are also presented as contoured pole plots to aid comparison between the new and the customary techniques.

Finally, a synopsis is presented in Section 7.4.

7.1 Equipment Used and Procedures Followed

This Section presents a description of the field and office procedures followed, together with the equipment used, in the execution of the four field studies. The description is in two parts, each forming a sub-Section: the first considers the

field operations, the second the office procedures. A summary of the procedures is given in a final sub-Section.

7.1.1 Field operations

The field operations performed during this work can be neatly described in terms of three separate but sequential steps, namely: installation of the control markers and carrying wires; establishment of the ground network and surveying of the ground and control markers; and photography. Each of these steps is described in turn in the following Divisions.

Installation of the control markers and carrying wires

The lengths of steel, 3mm diameter, multi-strand cable that were to be used as the carrying wires were first prepared by having suitable markers fastened to them at separations in the range of 2m to 4m. Short lengths of yellow plastic adhesive tape folded around the cable, and numbered to facilitate identification when sighting with the theodolite, were found to form perfectly acceptable markers. Figure 7.01 shows two of the markers, and illustrates the general principle. In keeping with good survey practice, the markers were placed arbitrarily on the wires and their spacings measured subsequently. The spacings were determined with a steel tape measure, whilst the wires were manually placed under tension by the two members of the survey team, with the distances being recorded to the nearest 1mm.

Following this preparation, the cables were installed in front of the face by drawing them up into place, rather than lowering them from the top of the face. This procedure was adopted to avoid the risk of either damaging the markers or of having the cables become fouled up on the rock face. Above the face, the

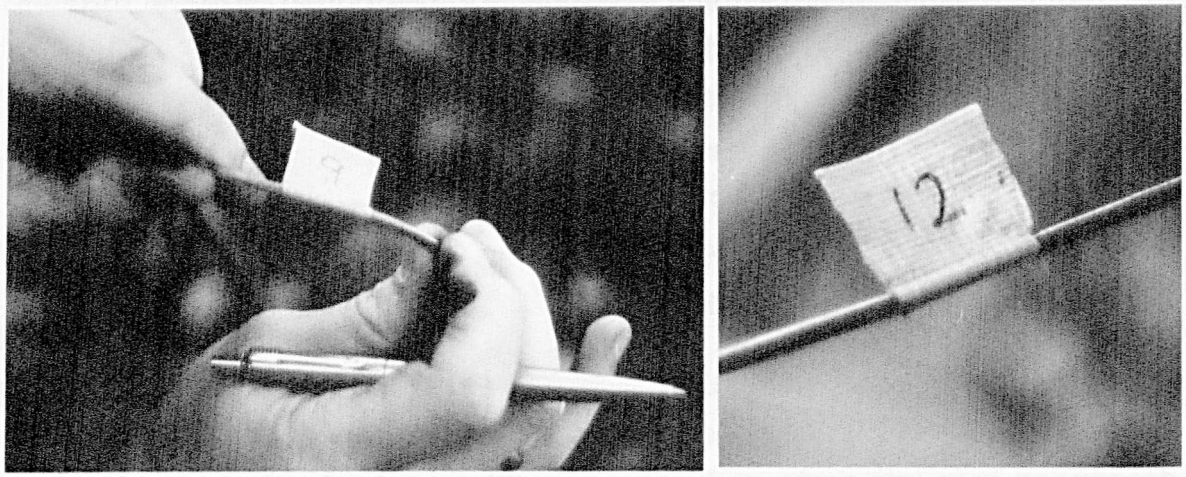


Figure 7.01 Typical cable markers.

cables were anchored by being passed around either a boulder (where one was conveniently located) or a steel pin driven into the ground; in either case they were held fast with cable clamps. Below the face, heavy steel weights—attached by mountaineers' karabiners to loops formed in the cables—were used to hold the cables under tension, as illustrated in Figure 7.02. Other than attempting to meet the basic requirements of the DLT (in that a minimum of 6 non-coplanar points, well distributed about the object space, are required to be visible in the photographs forming the stereopair) by avoiding placing the cables such that any two were approximately coplanar, no precise positioning of the cables took place. This was found to give a great deal of flexibility to the site operation, and allowed for pragmatic placing of the cables.

Establishment of the ground network, surveying of ground and control markers

The general form of the survey networks has been indicated previously in Figures 5.21 and 5.22. The two ground markers representing the two main control survey stations were established at either end of a baseline about 10m in front of, and roughly parallel to, the toe of the rock face. The precise length of the baseline was chosen such that all of the cable-mounted control markers were contained between the projection onto the rock face of the two ground stations, thus avoiding extrapolation beyond the ends of the baseline. The third ground marker was then established such that an approximately equilateral

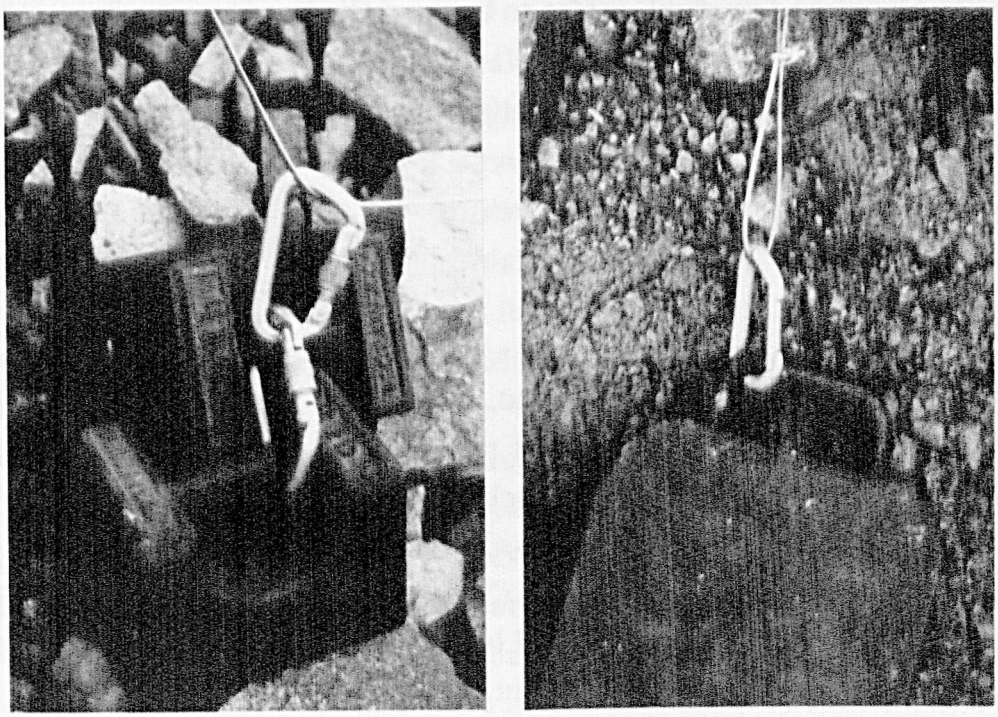


Figure 7.02 Weights used for tensioning control cables.

triangle was formed by the three markers. Site conditions (i.e. width of the bench on which the work was taking place) dictated that the third point lay closer to the rock face than the baseline at the granodiorite quarry, and *vice versa* at the limestone quarry: however, both configurations were found to be equally acceptable and convenient. The ground markers themselves took the form of either steel pins driven into the ground, or marks drawn on painted areas of the rocks forming the quarry floor.

The orientation of the ground network was then determined, relative to magnetic North, by taking bearings with a hand-held compass along each side of the triangle. It was found during this work that the sum of the included angles of the triangle could be determined to within 1° of 180° by taking the mean of bearings measured in both directions along the sides of the triangle. This allowed a bearing to be adopted for one of the sides of the triangle, and used in the subsequent data reduction as a datum.

To coordinate the various markers, a 20-seconds-of-arc theodolite was set up over each of the ground markers in turn, and at each station two complete rounds of horizontal and vertical angles (one on each face of the theodolite) was taken to the other ground markers. In addition, the slope distance from the theodolite trunnion axis to each ground marker was measured—to the nearest millimetre—with a 30m steel tape measure. When the theodolite was set up at either of the two baseline stations, the rounds of angles were extended such that horizontal and vertical angles to each of the markers on the cables were determined.

It is important to realise that, throughout this procedure, it was never necessary to work directly at the rock face, and the only close approaches required were for the positioning of the ground weights. As a consequence, a safe working environment was maintained.

Photography

Along a line approximately parallel to, and at a distance of approximately 10m to 15m from the rock face, a series of photographs was taken of the face itself. The distance of this line from the rock face was dictated by the height of the face relative to the vertical field of view of the camera, as an image of the complete rock face was sought. A Canon A1 35mm camera, fitted with a Tamron 35mm-85mm zoom lens, and loaded with Kodak Technical Pan 25ASA fine-grained black and white print film, was used at both locations.

The flexibility of the analytical plotter developed as part of this work, together with the adoption of the DLT for photogrammetric data reduction, imposed no absolute requirements on either the equipment used for the photography or the positioning and orientation of the camera. However, to maximise the stereoscopy of the resulting photographs, all of the exposures were taken with approximately parallel lens axis orientations. Ten or so exposures, taken at about 2m intervals along the rock face, was found to give sufficient insurance against factors such as inadequate photograph overlap, insufficient control points imaged on both photographs, or simple camera shake, and hence ensure that a suitable stereopair would be formed.

7.1.2 Office procedures

Following the field operations, the work transferred to the office. It was here that the survey data reduction, photogrammetric measurements and the subsequent geometrical analysis took place. These three procedures are described in the following three Divisions.

Survey data reduction

Following the field operations, the survey data were reduced using standard surveying techniques, in order to: identify any blunders in the data; compute mean vertical and horizontal angles in the networks; and determine approximate coordinates of the ground markers and cable control points. A spreadsheet program executing on a desktop computer was used for this (see Figures 5.24 and 5.26), both to guard against the introduction of computation errors as well as increase the speed and remove the tedium of the work.

These reduced data were then used, together with the marker spacing measurements, as input to further spreadsheets designed to implement the least squares adjustment of the two networks in each control survey (as described in Section 5.3). The first (see Figure 5.25) adjusted the ground network, such that accurate coordinates of the theodolite trunnion axes were derived. These coordinates were then used in the second—cable marker—adjustment (see Figure 5.27), in order to determine accurate marker coordinates.

An essential element of least squares adjustment is the use of appropriate weights for the various observation equations (equation (5.27)). In this work it was thought appropriate to apply weights on the basis that horizontal angles, vertical angles and taped distances were measured with errors of ±5", ±30" and ±0.001m, respectively. Of the angular measurements, the lower weight (i.e.

larger errors) associated with the vertical angles corresponds to the customary difficulty of using a theodolite for such measurements.

Using the data from the control surveys, convergence of each adjustment (defined as a maximum coordinate change of less than 0.0001m) was achieved after no more than six or seven iterations of the analysis. Using the observation equations for each network, an estimate of the standard errors of the marker coordinates was made [30] and found to be so small (around 1×10^{-6}m) as to be negligible.

Photogrammetric measurements

The analytical plotter developed as part of this work was used for the photogrammetric analysis. In use, this instrument was found to allow rapid and straightforward determination of photograph coordinates. The procedure followed was as described in the following paragraphs.

By studying prints (enlarged to approximately A5 size) of all of the photographs taken at the rock face, the control points were identified and marked. The large changes in projection between adjacent photographs—due to the relative configuration of the photograph exposure stations, control points and rock face—meant that identifying homologous control points on the basis of their apparent position with respect to the rock face was not possible, and that the carrying wires themselves proved the most reliable means of locating them. Having marked all of the visible control points on all of the photographs, a pair on which more than six points were visible was chosen for analysis.

The photographs forming the stereopair were placed on the photograph carriers of the analytical plotter, covered with a thin transparent acetate sheet, and secured with clips. The flexibility of the analytical plotter meant that no special location or orientation procedures were necessary, other than to ensure that the left hand photograph was on the left hand carrier, and *vice versa*. The various control points were then coordinated with the instrument, their object space coordinates entered into the controlling program, and the DLT equations solved.

Following solution of the DLT equations, a correctly oriented stereomodel of the rock face was available, from which photogrammetric measurements of points on the photographs could be taken using the phenomenon of parallax, as described in Section 3.1. An initial check of the system was made by re-sighting a selection of the control markers and computing their apparent object space coordinates; these results compared well with the actual object space

coordinates of the markers. Coordination of points on arbitrarily chosen, but distinctly visible, discontinuity surfaces was then undertaken. On each of a series of discontinuities, three points (selected from the extremities of the surface) were coordinated, with the results automatically being written to a file by the controlling computer program. It was found convenient to mark, on the acetate sheets covering the photographs, each point as it was coordinated and each surface triangle as it was completed; this prevented the same surface being coordinated a number of times.

Care was taken to ensure that discontinuities of all orientations, and distributed throughout the images, were observed, rather than only those that appeared face on to the camera locations. Indeed, the enhanced depth of view associated with the stereoscopy meant that, with only a little practice, both sub-vertical and sub-horizontal discontinuities (i.e. surfaces almost edge on to the camera locations) could be coordinated with ease.

Subsequent geometrical analysis

Subsequent geometrical analysis began with the computation of the dip direction and dip angle for each of the observed discontinuity surfaces. This was performed using the computed coordinates of the three points observed on each surface, with the assumption that each discontinuity surface was adequately described as a plane passing through these points. The result of this was to produce a list of dip direction and dip angle data, suitable for analysis using FADD.

The first stage of the analysis of this data using FADD involved identification of the optimal partition, on the basis of an assessment of the magnitudes of various cluster validity functions (described in Section 6.2), in order to determine the number of discontinuity sets present. Following identification of the optimal partition, both cluster assignment plots and membership level plots (described in Section 6.4) were produced, together with the cluster shape classification coordinates derived from the eigenvalues of the cluster scatter matrices (described in Section 6.3). The final result of this analysis was an improved interpretation of the rock mass structure.

7.1.3 Summary

A description of the field and office procedures adopted when the new, integrated system for the measurement and analysis of rock mass geometry was used in the case studies has been given in this Section. The various steps in

the procedure are straightforward, and form an easily followed sequence. This sequence can be summarised as:

Field procedures

(i) at the rock face, install the control markers and carrying wires, ensuring that the markers are well distributed about the rock face and that the wires are not coplanar;

(ii) establish the three markers that comprise the ground control network and, using a theodolite and tape measure, survey these markers together with the control point markers on the carrying wires;

(iii) take a series of photographs along a line approximately parallel to, and between about 10m to 15m from, the rock face;

Office procedures

(iv) perform the survey data reduction, including a least squares adjustment of the observations, in order to determine the coordinates of the control points;

(v) select a pair of photographs on which at least six control points have been marked, mount these in the analytical plotter, enter the object space coordinates and determine the photograph coordinates of the control points, and solve the DLT equations to produce a correctly oriented stereomodel of the rock face;

(vi) using the analytical plotter, coordinate three points on each of as many discontinuity surfaces as is desired, and from these coordinates compute the dip direction and dip angle of the discontinuity surface; and

(vi) analyse the resulting orientation data using FADD, initially by identifying the optimal partition (i.e. number of discontinuity sets), and subsequently by producing both cluster assignment plots and membership level plots, together with cluster shape classification coordinates.

Following the final stage in this procedure, the geotechnical engineer is in a position to make an improved interpretation of the structure of the rock mass. Two examples of the procedure in use are given in the following two Sections.

7.2 Application at a Limestone Rock Face

The results of two discontinuity orientation surveys (one photogrammetric, the other using a geological compass) of a rock face exposed in Tytherington

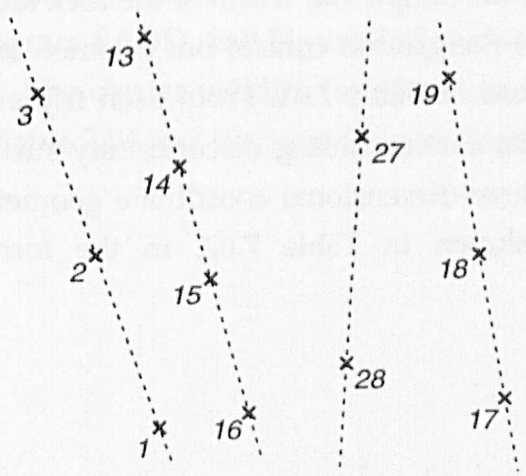
Quarry, near Bristol, UK (National Grid Reference ST661888) are presented here. The rock mass exposed at this site is formed from the Carboniferous limestone, and the quarry face surveyed had a trend of approximately north east-south west, with the cameras being directed towards the north west. The results are presented in the following two sub-Sections.

7.2.1 Photogrammetric survey

An array of four steel cables, carrying a total of 12 control markers, was installed in front of the rock face in the configuration shown in Figure 7.03(a). The reduced relative coordinates of the control markers are given in Figure 7.03(b).

The left and right photographs of the stereopair used in the analysis are shown in Figures 7.04 and 7.05, respectively. These photographs were selected on the basis of maximal number of control markers easily visible on both images, with the result that control markers 13, 14, 15, 18, 19, 27, and 28 were available for the analysis.

In all, one hundred and fifty point coordinates were taken from the photographs shown in Figures 7.04 and 7.05, and the projections on the left hand stereophotograph of the corresponding fifty discontinuity surface triangles (formed by connecting together the points imaged on each surface), together with the control points, are shown in Figure 7.06. These triangles show



(a) marker configuration

marker	x (m)	y (m)	z (m)
1	92.5537	101.3767	102.6443
2	91.6158	101.1907	104.8706
3	90.6416	100.9940	107.2190
13	91.4997	101.9094	109.0009
14	92.7013	102.1718	106.7500
15	93.9221	102.4390	104.4734
16	95.2387	102.7265	102.0532
17	99.2132	105.2561	101.5169
18	97.7849	105.8469	104.6281
19	96.1480	106.5236	108.2191
27	94.1271	104.8365	106.3454
28	94.9998	104.5261	102.0777

(b) marker coordinates

Figure 7.03 Limestone face photogrammetric survey: control marker configuration and relative coordinates.

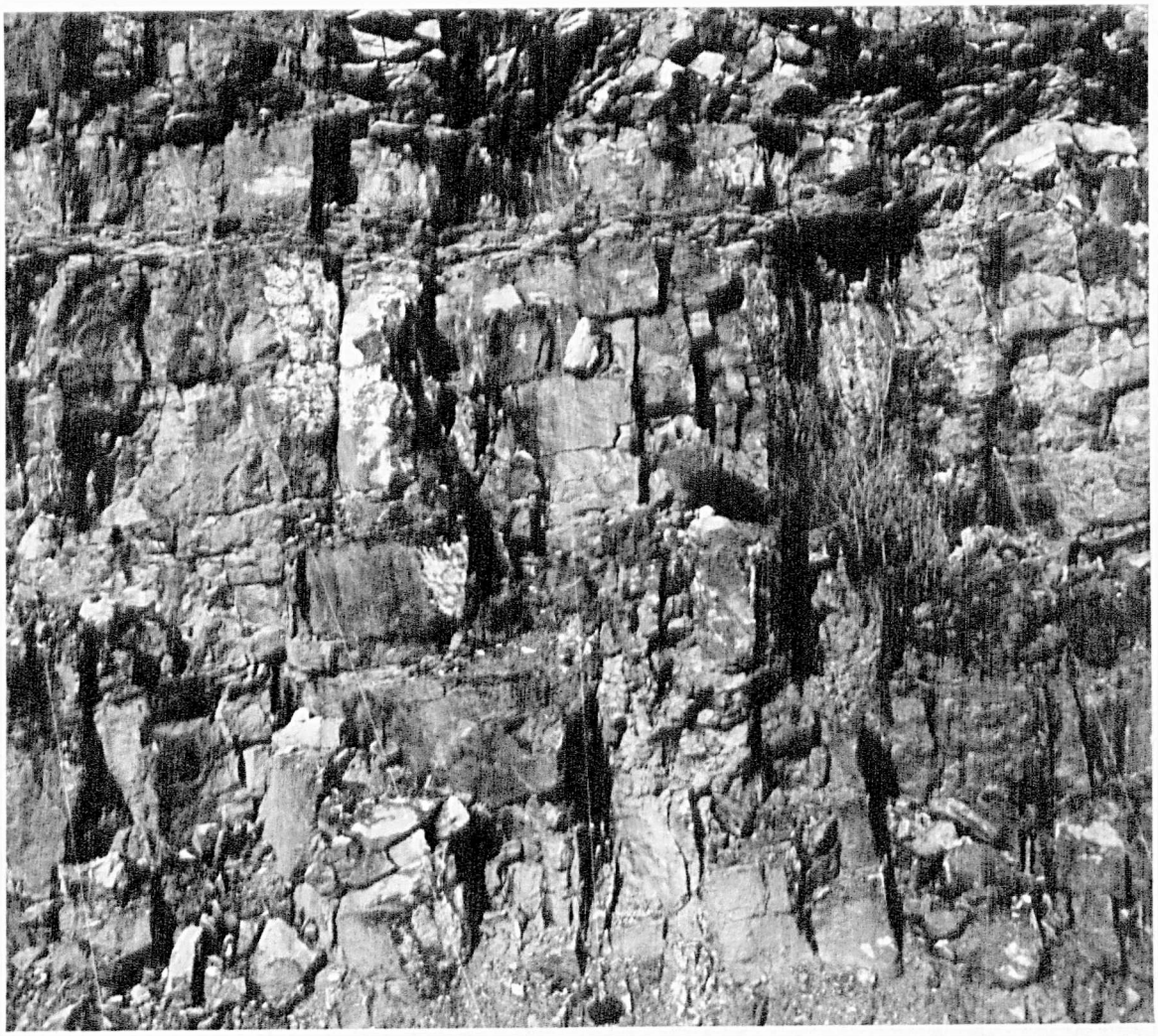


Figure 7.04 Limestone face photogrammetric survey: left hand stereophotograph

how the samples were taken from across the height and width of the rock face.

The surface coordinates, relative to the ground control but oriented with respect to magnetic north, are reproduced in Table 7.01. From each triplet of surface coordinates, the orientation of the corresponding discontinuity surface itself was computed using customary three-dimensional coordinate geometry. These discontinuity orientations are shown in Table 7.02, in the format dip direction/dip angle.

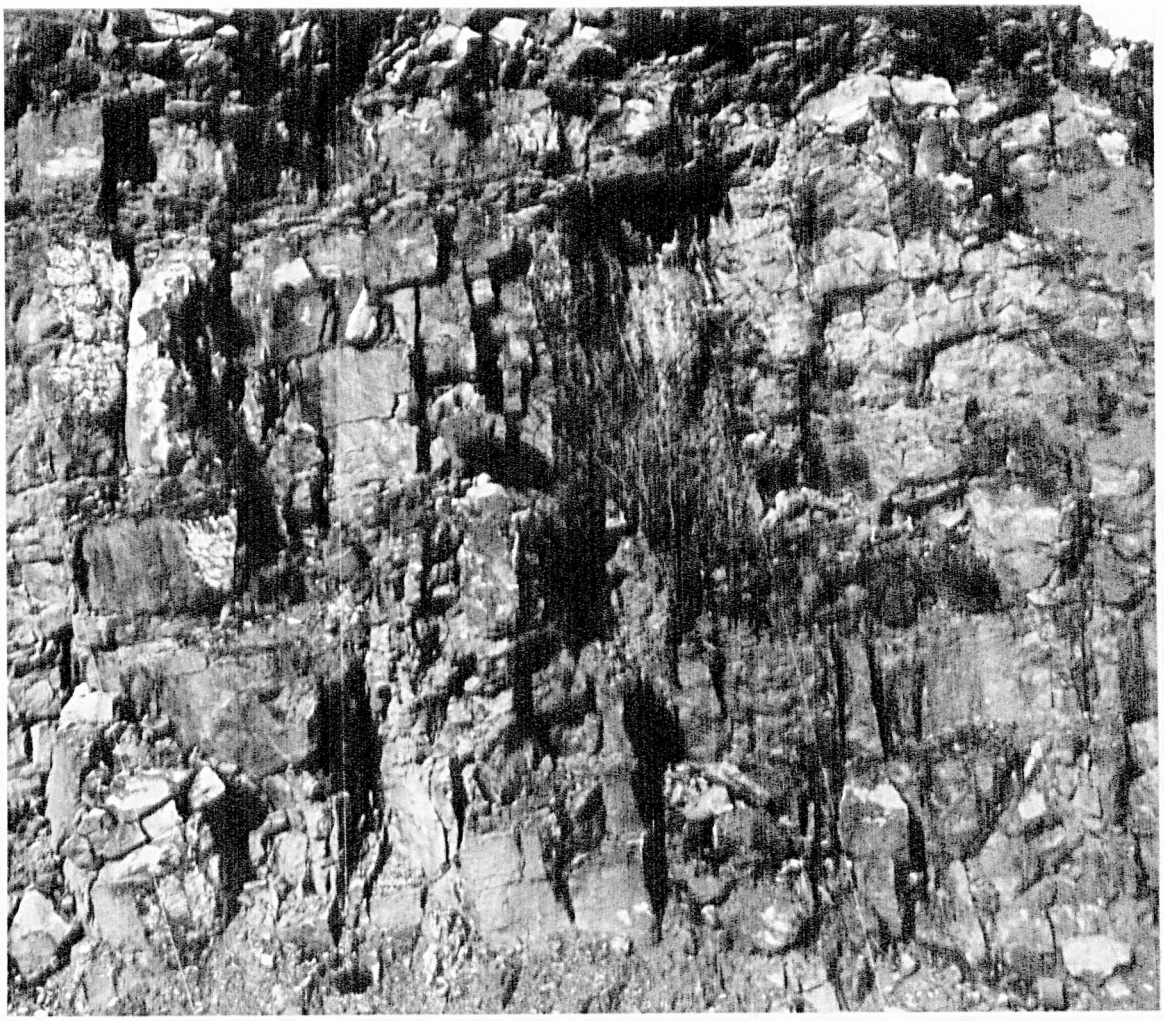


Figure 7.05 Limestone face photogrammetric survey: right hand stereophotograph

These orientation data were subsequently processed with the Author's program FADD, and Figure 7.07 shows the resulting cluster assignment plot for c=4. The cluster validity functions corresponding to this analysis are presented in Table 7.04 and discussed in sub-Section 7.2.3

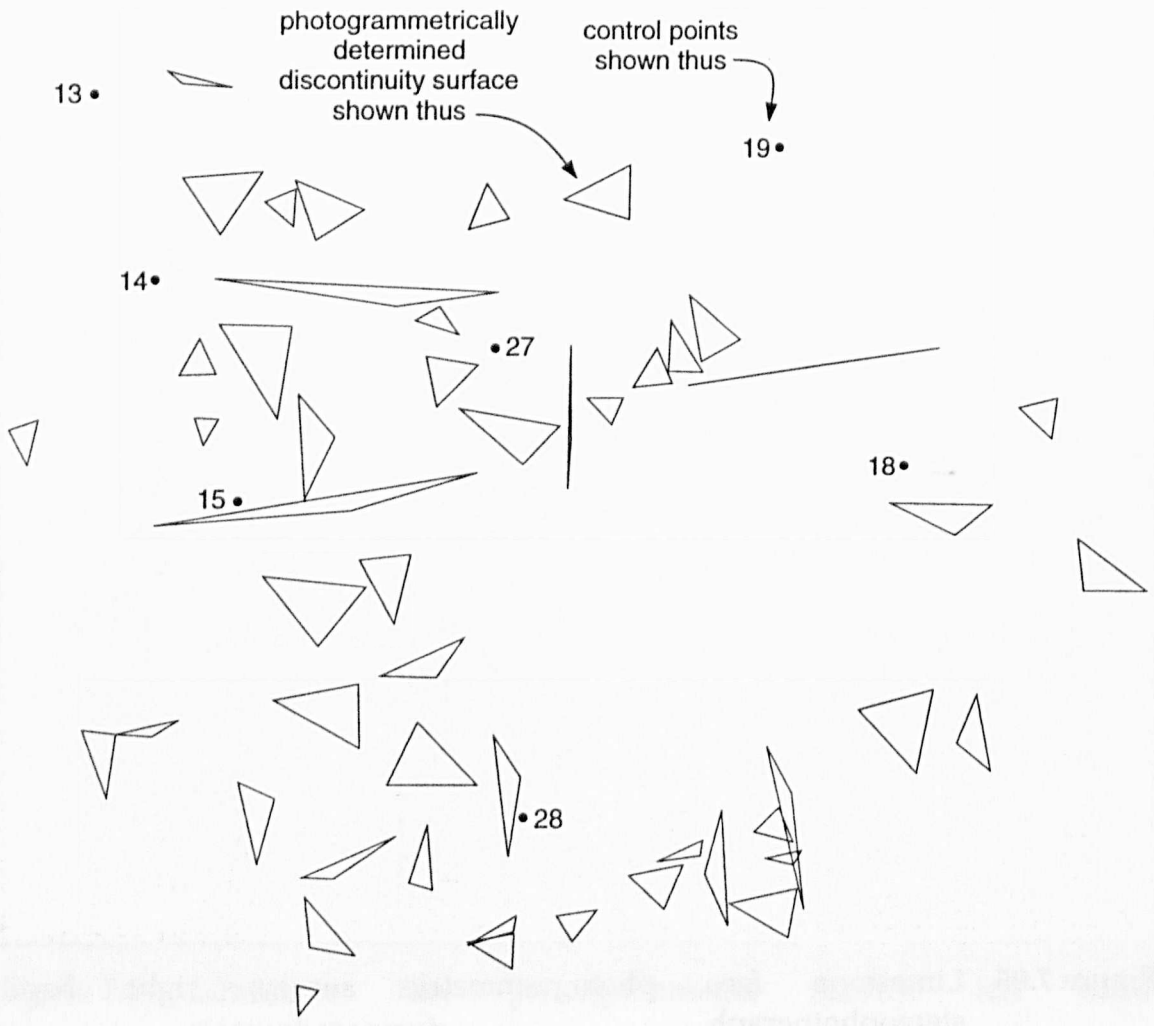


Figure 7.06 Limestone face photogrammetric survey: discontinuity triangles and control points.

Table 7.01 Limestone face photogrammetric survey: discontinuity surface coordinates.

1:	(92.911, 101.861, 102.727),	(93.181, 101.989, 102.218),	(93.082, 102.059, 102.699)
2:	(93.076, 102.119, 102.793),	(93.152, 102.546, 102.835),	(93.162, 102.323, 102.692)
3:	(94.457, 102.945, 101.496),	(94.772, 103.182, 101.095),	(94.359, 103.086, 101.143)
4:	(93.987, 102.826, 101.736),	(93.450, 102.946, 102.361),	(94.068, 102.885, 102.208)
5:	(94.725, 103.527, 101.932),	(94.258, 103.046, 101.633),	(94.555, 103.113, 101.624)
6:	(94.663, 103.864, 102.025),	(94.488, 104.077, 101.542),	(94.571, 103.787, 101.595)
7:	(94.884, 104.116, 101.165),	(95.153, 104.327, 101.226),	(95.167, 104.304, 100.980)
8:	(95.159, 104.745, 101.347),	(95.320, 105.014, 101.387),	(95.294, 104.782, 101.159)
9:	(94.945, 104.063, 101.147),	(95.112, 104.358, 101.238),	(94.968, 104.505, 101.351)
10:	(94.503, 102.931, 100.696),	(94.483, 102.914, 100.890),	(94.666, 102.969, 100.864)
11:	(93.443, 103.211, 103.018),	(94.050, 103.593, 103.121),	(94.139, 103.574, 102.611)
12:	(93.375, 103.063, 104.042),	(93.801, 103.355, 103.443),	(94.091, 103.579, 103.916)
13:	(94.152, 103.968, 104.229),	(94.276, 103.741, 103.609),	(94.063, 103.520, 104.133)
14:	(92.998, 103.506, 105.860),	(93.336, 103.432, 104.780),	(93.618, 103.502, 105.306)
15:	(92.916, 103.430, 106.537),	(92.734, 102.797, 106.447),	(93.111, 103.239, 105.540)
16:	(91.998, 101.577, 105.441),	(91.975, 101.545, 105.018),	(91.822, 101.406, 105.352)
17:	(92.186, 102.577, 108.071),	(92.446, 102.879, 107.470),	(92.523, 103.219, 108.228)
18:	(92.608, 103.225, 107.869),	(92.668, 103.511, 108.076),	(92.554, 103.585, 107.730)
19:	(91.927, 103.145, 109.378),	(91.495, 102.653, 109.558),	(92.067, 102.533, 109.131)
20:	(92.485, 102.787, 105.598),	(92.602, 102.833, 105.323),	(92.767, 102.862, 105.537)
21:	(93.278, 104.541, 106.772),	(93.432, 104.705, 106.914),	(93.635, 104.791, 106.587)
22:	(93.904, 104.621, 105.739),	(94.321, 105.025, 105.184),	(94.627, 105.176, 105.517)
23:	(94.912, 104.132, 102.314),	(93.882, 104.050, 102.363),	(94.534, 103.827, 102.798)
24:	(94.537, 105.594, 105.902),	(94.804, 105.796, 105.879),	(94.744, 105.709, 105.613)
25:	(95.239, 107.518, 102.354),	(96.221, 106.407, 101.380),	(95.507, 106.864, 102.711)
26:	(96.087, 106.496, 101.537),	(95.844, 105.955, 101.428),	(96.044, 106.416, 101.164)
27:	(95.373, 106.170, 101.925),	(95.613, 105.877, 101.798),	(95.480, 105.513, 101.750)
28:	(95.342, 105.339, 101.642),	(95.647, 105.637, 101.723),	(95.416, 105.657, 101.387)
29:	(95.537, 106.230, 102.162),	(95.690, 105.841, 101.657),	(95.796, 105.983, 101.259)
30:	(95.798, 109.722, 103.342),	(95.535, 109.724, 102.893),	(95.898, 109.706, 102.609)
31:	(94.384, 106.353, 106.212),	(94.620, 106.654, 106.259),	(94.504, 106.566, 106.639)
32:	(94.767, 106.757, 107.178),	(94.388, 107.337, 106.703),	(94.089, 108.306, 107.259)
33:	(93.996, 104.693, 103.621),	(94.133, 104.306, 103.232),	(93.738, 104.029, 103.260)
34:	(93.379, 104.619, 106.391),	(93.737, 104.952, 106.290),	(93.461, 104.705, 105.882)
35:	(95.876, 106.837, 101.845),	(96.081, 106.467, 101.722),	(95.646, 106.648, 101.783)
36:	(95.993, 106.584, 101.904),	(95.682, 106.792, 102.194),	(95.532, 106.639, 101.999)
37:	(95.723, 109.102, 103.350),	(95.368, 108.391, 103.137),	(95.348, 109.289, 102.604)
38:	(95.809, 111.950, 105.235),	(95.495, 112.542, 104.771),	(96.315, 112.374, 104.635)
39:	(94.599, 113.037, 107.327),	(95.203, 112.676, 106.701),	(95.414, 112.528, 107.154)
40:	(93.703, 111.459, 105.968),	(94.831, 111.041, 105.318),	(95.112, 111.394, 105.678)
41:	(93.785, 106.901, 108.394),	(93.948, 106.752, 108.945),	(93.810, 105.928, 108.322)
42:	(92.863, 105.700, 108.781),	(93.083, 105.834, 108.321),	(92.699, 105.608, 108.255)
43:	(91.916, 102.811, 104.695),	(93.637, 103.705, 104.668),	(93.774, 104.958, 105.200)
44:	(94.626, 106.871, 106.283),	(94.923, 108.703, 106.846),	(94.820, 111.012, 107.606)
45:	(92.723, 102.714, 106.878),	(92.962, 104.545, 107.009),	(93.785, 105.129, 107.070)
46:	(92.096, 103.852, 108.494),	(92.345, 103.982, 107.768),	(92.802, 104.211, 108.018)
47:	(94.723, 106.535, 106.865),	(94.549, 107.180, 106.511),	(94.516, 106.711, 106.423)
48:	(94.456, 104.971, 102.432),	(94.623, 104.711, 101.800),	(94.885, 104.304, 102.694)
49:	(92.146, 102.776, 106.097),	(92.473, 102.962, 106.068),	(92.191, 102.919, 106.484)
50:	(94.478, 105.436, 106.408),	(94.224, 105.697, 105.579),	(94.417, 105.483, 105.016)

Table 7.02 Limestone face photogrammetric survey: discontinuity orientations.

129.3,75.9	102.3,68.1	349.8,66.1	183.7,82.3	165.3,40.8	256.5,76.0	143.0,83.8
124.4,64.3	148.0,29.4	343.7,86.8	148.8,83.0	144.0,88.7	104.7,74.0	175.7,84.7
108.0,76.2	136.2,89.0	119.3,82.5	268.6,72.0	109.4,36.4	345.8,86.7	140.3,79.0
324.7,80.9	357.0,59.5	142.4,83.1	059.4,74.0	294.0,89.0	131.5,18.8	309.5,60.4
088.1,74.5	182.1,89.0	128.8,84.7	245.7,77.3	141.6,45.0	317.2,89.2	179.6,18.3
177.9,52.9	288.0,60.9	020.5,46.2	032.3,88.4	172.4,49.0	267.3,74.2	329.5,89.6
151.0,27.3	162.0,18.8	200.7,04.2	333.2,89.7	269.1,65.2	237.3,90.0	149.2,75.5
226.9,89.5						

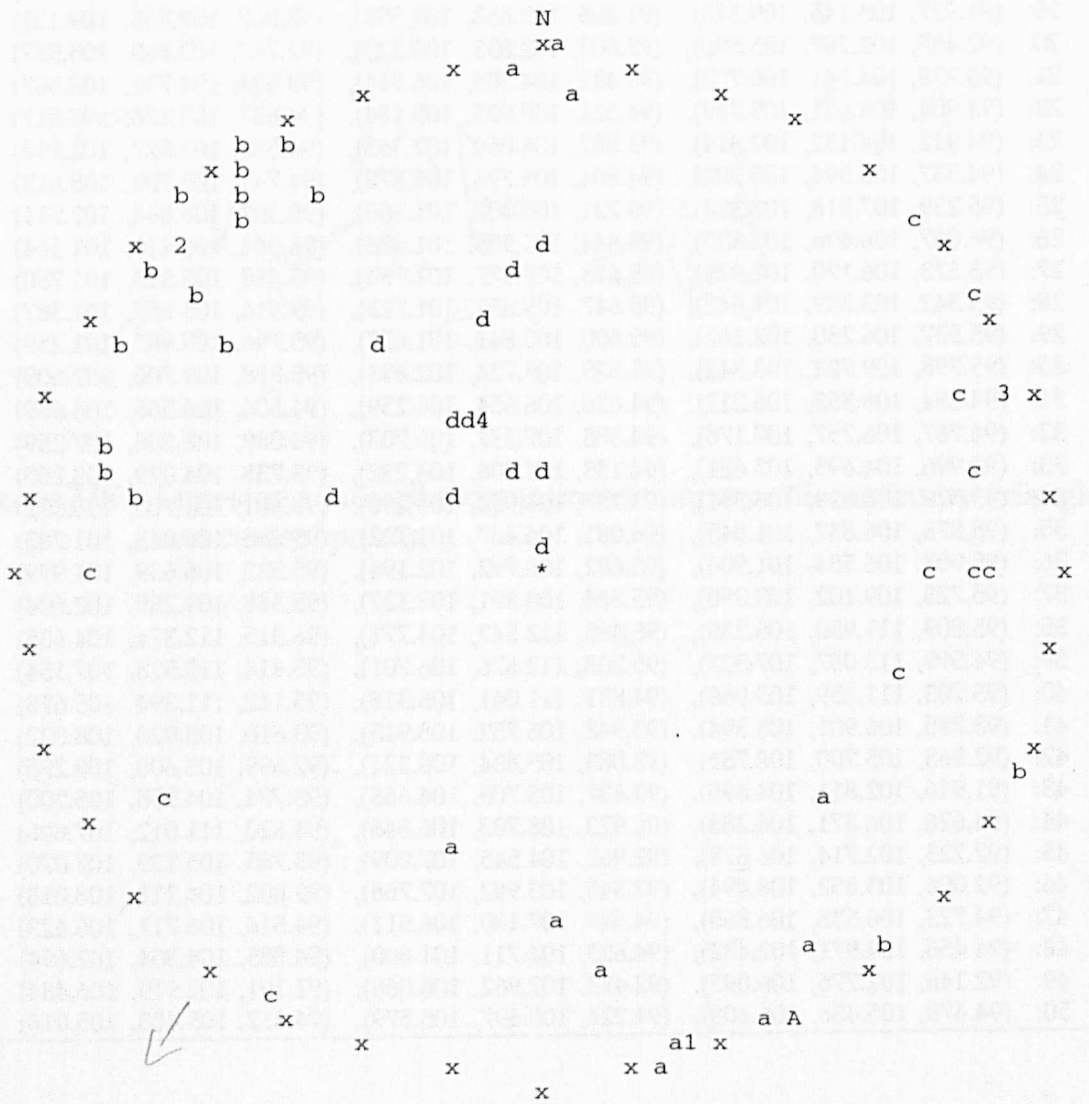


Figure 7.07 Limestone face photogrammetric survey: cluster assignment plot.

7.2.2 Compass survey

A total of fifty discontinuity orientation measurements were taken on the rock face in the vicinity of the control wires, at heights up to about 1.5m from ground level and distributed along the length of the exposure. These measurements are reproduced in Table 7.03, and an equal angle lower hemispherical projection of the corresponding pole densities—determined using Schmidt contouring with a 1% counting circle by the program DIPS—is shown in Figure 7.08. The results of a supplementary analysis, conducted using the Author's program FADD, are shown both in Figure 7.09 as a membership level plot for the case c=4, and in Table 7.04 as the magnitudes of the various cluster validity functions applied to the data.

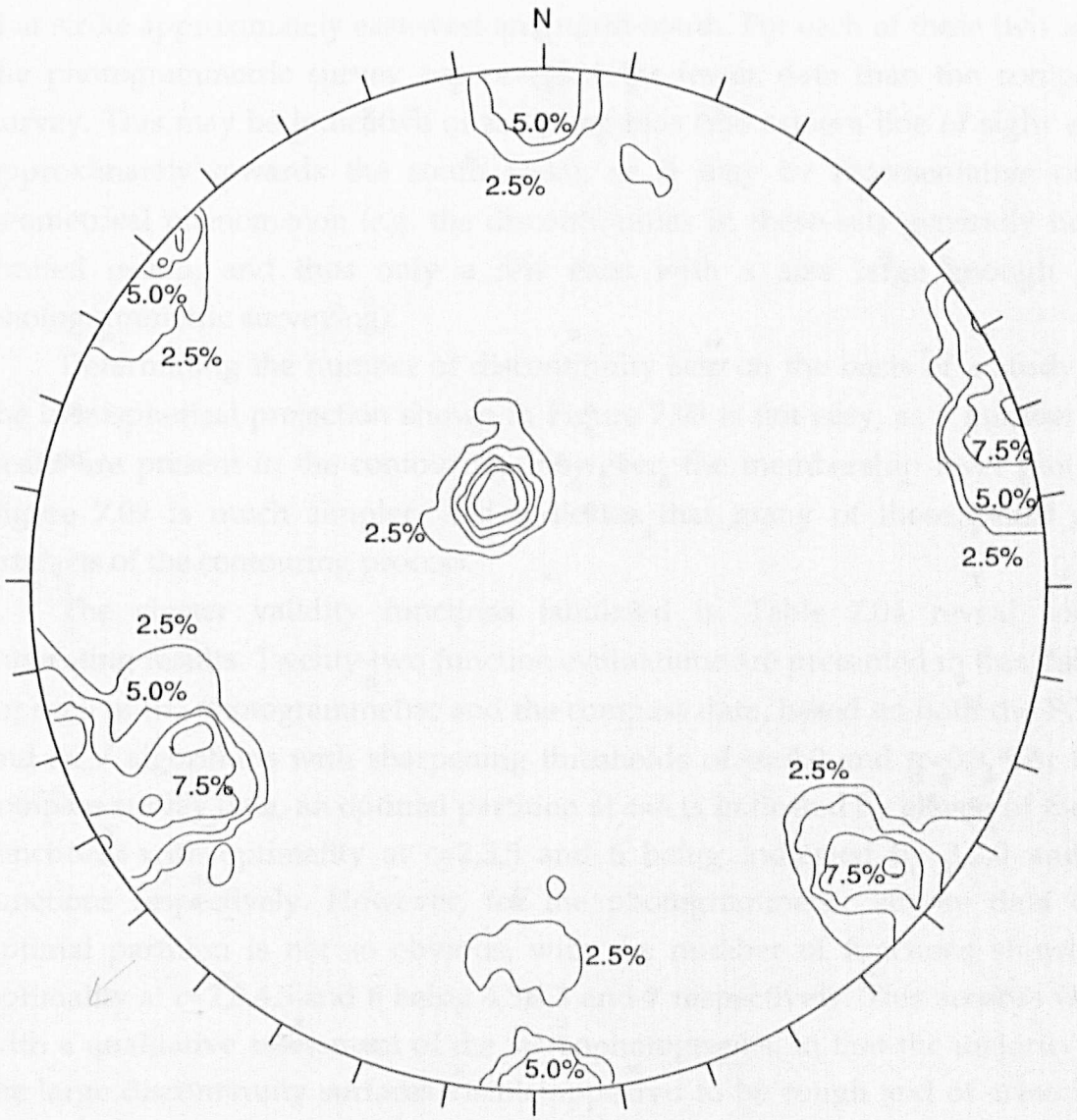


Figure 7.08 Limestone face compass survey: equal angle contoured pole plot.

Table 7.03 Limestone face compass survey: discontinuity orientations.

061,86	313,84	047,73	179,87	149,16	285,89	254,86	177,87	302,35	151,19
181,89	254,88	130,25	053,86	358,70	256,88	191,74	149,22	067,16	015,72
074,72	060,74	307,71	171,42	063,77	126,89	305,17	269,85	197,86	002,76
161,19	337,76	124,20	091,71	068,85	354,55	248,88	303,88	170,20	061,72
310,78	163,25	072,76	311,76	163,31	060,76	135,82	061,90	149,24	312,84

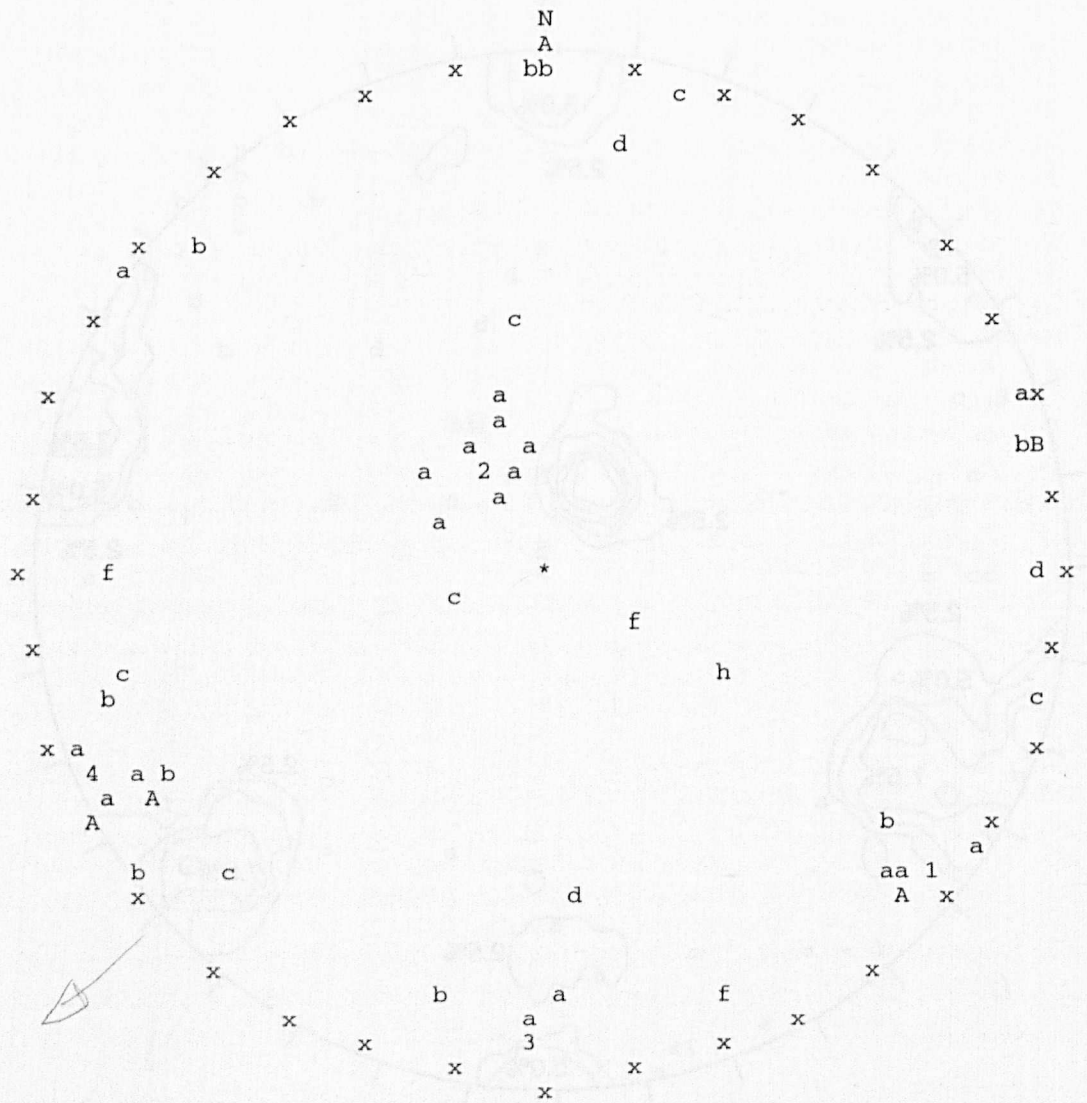


Figure 7.09 Limestone face compass survey: membership level plot.

7.2.3 Discussion

Figure 7.06 shows that the photogrammetric process has allowed sampling of the rock face over its entire extent, with the exception of the region obscured by the small bush. Furthermore, it is clear from the presence of triangles that are long and thin that it has been possible to measure coordinates on discontinuities that are inclined almost parallel to the line of sight.

A comparison of the hemispherical projection plots shown in Figures 7.07 and 7.08 shows that there is reasonable agreement between the photogrammetric and the compass surveys. When the photogrammetric data are analysed with FADD on the basis of there being four clusters, a partition that is comparable to that resulting from a customary contouring analysis being applied to compass data is formed. There are some distinct differences, particularly in the number of data forming the two vertical discontinuity sets that strike approximately east-west and north-south. For each of these two sets, the photogrammetric survey has revealed far fewer data than the compass survey. This may be indicative of sampling bias (the camera line of sight was approximately towards the south west), or it may be representative of a geometrical phenomenon (*e.g.* the discontinuities in these sets generally have limited extent, and thus only a few exist with a size large enough for photogrammetric surveying).

Determining the number of discontinuity sets on the basis of a study of the hemispherical projection shown in Figure 7.08 is not easy, as a number of peaks are present in the contouring. However, the membership level plot of Figure 7.09 is much simpler, and indicates that many of these peaks are artefacts of the contouring process.

The cluster validity functions tabulated in Table 7.04 reveal some interesting results. Twenty-two function evaluations are presented in this Table for each of the photogrammetric and the compass data, based on both the FCM and FCV algorithms with sharpening thresholds of α=0.0 and α=0.9. For the compass survey data, an optimal partition at c=4 is indicated by eleven of these functions, with optimality at c=2,3,5 and 6 being indicated by 3,2,0 and 6 functions respectively. However, for the photogrammetric survey data the optimal partition is not so obvious, with the number of functions showing optimality at c=2,3,4,5 and 6 being 4,5,3,3 and 7 respectively. This accords well with a qualitative assessment of the stereophotographs, in that the majority of the large discontinuity surfaces visible appeared to be rough and of arbitrary shape, and hence not obviously members of distinct sets.

Table 7.04 Limestone face survey: cluster validity functions.

		no. of clusters	partition density	separation index	normalised partition coefficient	modified partition coefficient	partition entropy	partition coefficient
Compass Survey								
FCM	α=0.0	2		0.213	0.287	0.300	0.173	0.643
		3		0.165	0.593	0.408	0.129	0.729
		4		***0.271***	***0.684***	0.482	***0.114***	***0.763***
		5		0.247	0.661	0.448	0.130	0.728
		6		0.078	0.619	***0.501***	0.150	0.683
	α=0.9	2		1.685	-0.404	0.209	***0.011***	0.298
		3		***2.005***	0.308	0.607	0.021	0.539
		4		1.854	***0.422***	***0.662***	0.027	***0.566***
		5		1.186	0.346	0.631	0.022	0.477
		6		1.173	0.350	0.628	0.021	0.459
FCV	α=0.0	2	1.254	0.074	0.198	0.282	0.194	0.599
		3	9.226	0.167	0.611	0.448	0.124	0.741
		4	16.629	***0.271***	***0.666***	0.422	***0.120***	***0.750***
		5	16.825	0.236	0.628	0.460	0.141	0.702
		6	***19.662***	0.178	0.651	***0.541***	0.139	0.709
	α=0.9	2	0.251	***2.100***	-0.518	0.169	***0.009***	0.241
		3	5.643	1.763	0.311	0.601	0.019	***0.541***
		4	9.010	2.089	0.310	0.604	0.019	0.482
		5	7.625	1.849	0.260	0.583	0.016	0.408
		6	***12.144***	0.810	***0.370***	***0.639***	0.023	0.475
Photogrammetric Survey								
FCM	α=0.0	2		0.124	0.348	0.330	***0.159***	***0.674***
		3		0.159	0.476	0.358	0.167	0.651
		4		***0.176***	0.465	0.414	0.191	0.599
		5		0.156	0.480	***0.441***	0.196	0.584
		6		0.156	***0.482***	0.435	0.202	0.568
	α=0.9	2		1.580	-0.448	0.207	0.012	0.276
		3		2.302	-0.003	0.439	0.014	***0.331***
		4		1.777	0.049	0.472	0.017	0.287
		5		2.019	***0.072***	***0.482***	0.011	0.258
		6		***3.116***	0.027	0.457	***0.006***	0.189
FCV	α=0.0	2	3.155	0.124	0.415	0.394	***0.144***	***0.708***
		3	5.350	***0.185***	***0.524***	***0.429***	0.153	0.682
		4	7.150	0.079	0.421	0.356	0.206	0.566
		5	13.263	0.133	0.496	0.377	0.189	0.597
		6	***15.697***	0.133	0.484	0.390	0.199	0.570
	α=0.9	2	1.375	1.087	-0.284	0.284	0.021	0.358
		3	2.556	1.335	0.087	0.498	0.014	***0.392***
		4	1.809	***1.503***	-0.059	0.424	***0.007***	0.206
		5	3.915	1.447	0.072	0.486	0.011	0.258
		6	***4.967***	1.344	***0.130***	***0.518***	0.012	0.275

Optimal partitions shown emboldened and italicised

7.3 Application at a Granodiorite Rock Face

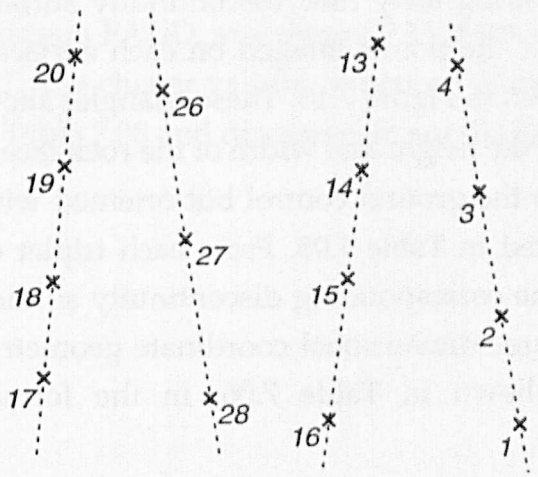
The results of two discontinuity orientation surveys (one photogrammetric, the other using a geological compass) of a rock face exposed in Mountsorrel Quarry, near Leicester, UK (National Grid Reference SK565150) are presented here. The rock mass exposed at this site is formed from a granodiorite of Devonian age, and the quarry face surveyed had a trend of approximately east-west, with the cameras being directed towards the south. The results are presented in the following two sub-Sections.

7.3.1 Photogrammetric survey

An array of four steel cables, carrying a total of 15 control markers, was installed in front of the rock face in the configuration shown in Figure 7.10(a). The reduced relative coordinates of the control markers are given in Figure 7.10(b).

The left and right photographs of the stereopair used in the analysis are shown in Figures 7.11 and 7.12, respectively. These photographs were selected on the basis of maximal number of control markers easily visible on both images, with the result that control markers 3, 4, 13, 15, 16, 18, 20, 26, 27, and 28 were available for the analysis.

In all, one hundred and forty seven point coordinates were taken from the photographs shown in Figures 7.11 and 7.12, and the projections on the left



(a) marker configuration

marker	x (m)	y (m)	z (m)
1	97.1475	61.9322	102.0405
2	97.2372	61.5888	104.4370
3	97.1973	61.0898	106.9373
4	97.1605	60.6247	109.3077
13	99.2284	59.6397	110.1603
14	99.3303	59.8282	107.6042
15	99.4309	60.0164	105.0159
16	99.5411	60.2175	102.2554
17	104.7872	62.4885	100.2308
18	104.8941	61.4483	103.5438
19	105.0166	60.2686	107.3681
20	105.1347	59.1194	111.1381
26	102.1123	60.0816	109.0034
27	102.1120	60.4596	104.5974
28	102.1105	60.8639	100.2491

(b) marker coordinates

Figure 7.10 Granodiorite face photogrammetric survey: control marker configuration and relative coordinates.

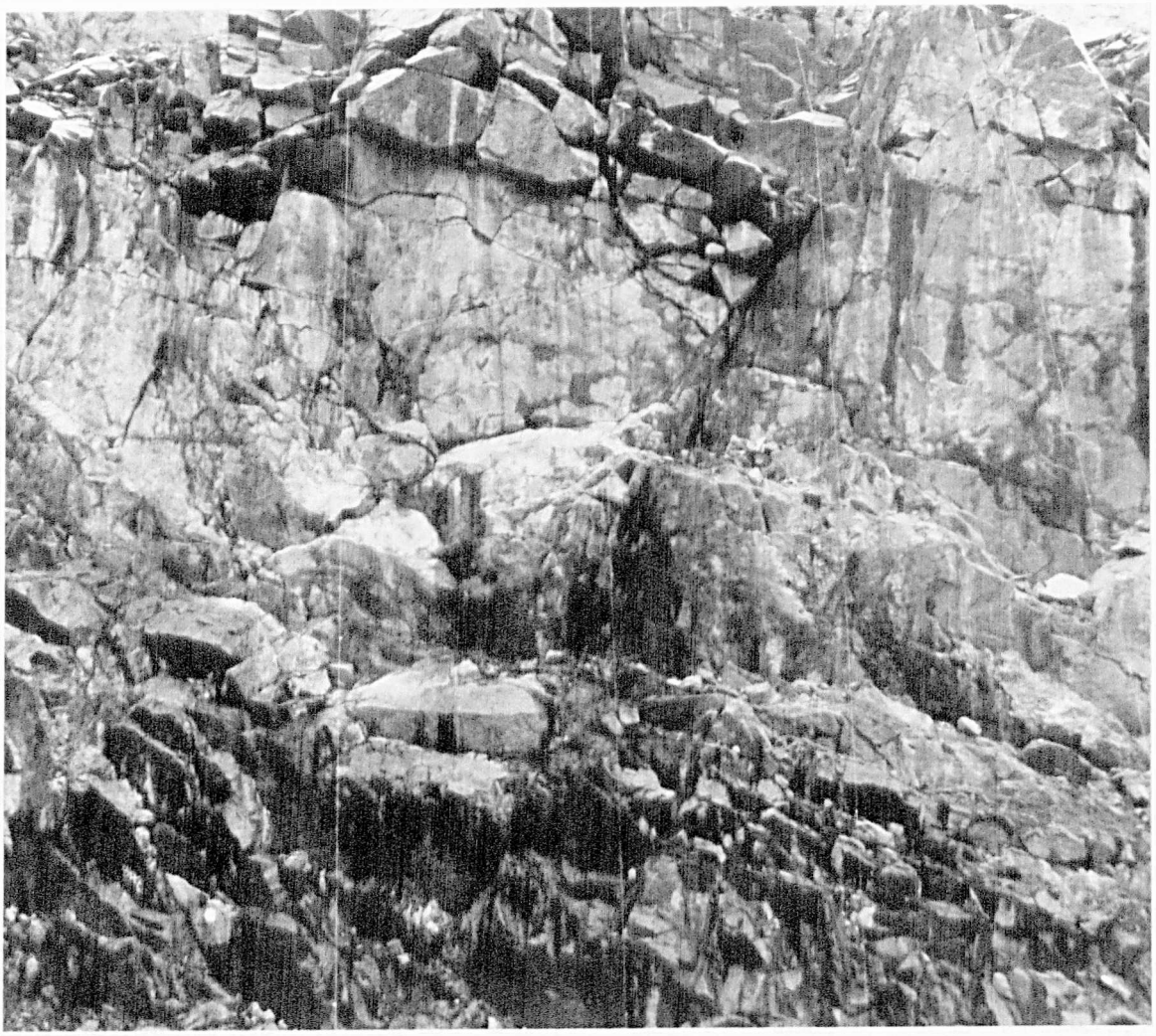


Figure 7.11 Granodiorite face photogrammetric survey: left hand stereophotograph

hand stereophotograph of the corresponding forty nine discontinuity surface triangles (formed by connecting together the points imaged on each surface), together with the control points, are shown in Figure 7.13. These triangles show how the samples were taken from across the height and width of the rock face.

The surface coordinates, relative to the ground control but oriented with respect to magnetic north, are reproduced in Table 7.05. From each triplet of surface coordinates, the orientation of the corresponding discontinuity surface itself was computed using customary three-dimensional coordinate geometry. These discontinuity orientations are shown in Table 7.06, in the format dip direction/dip angle.

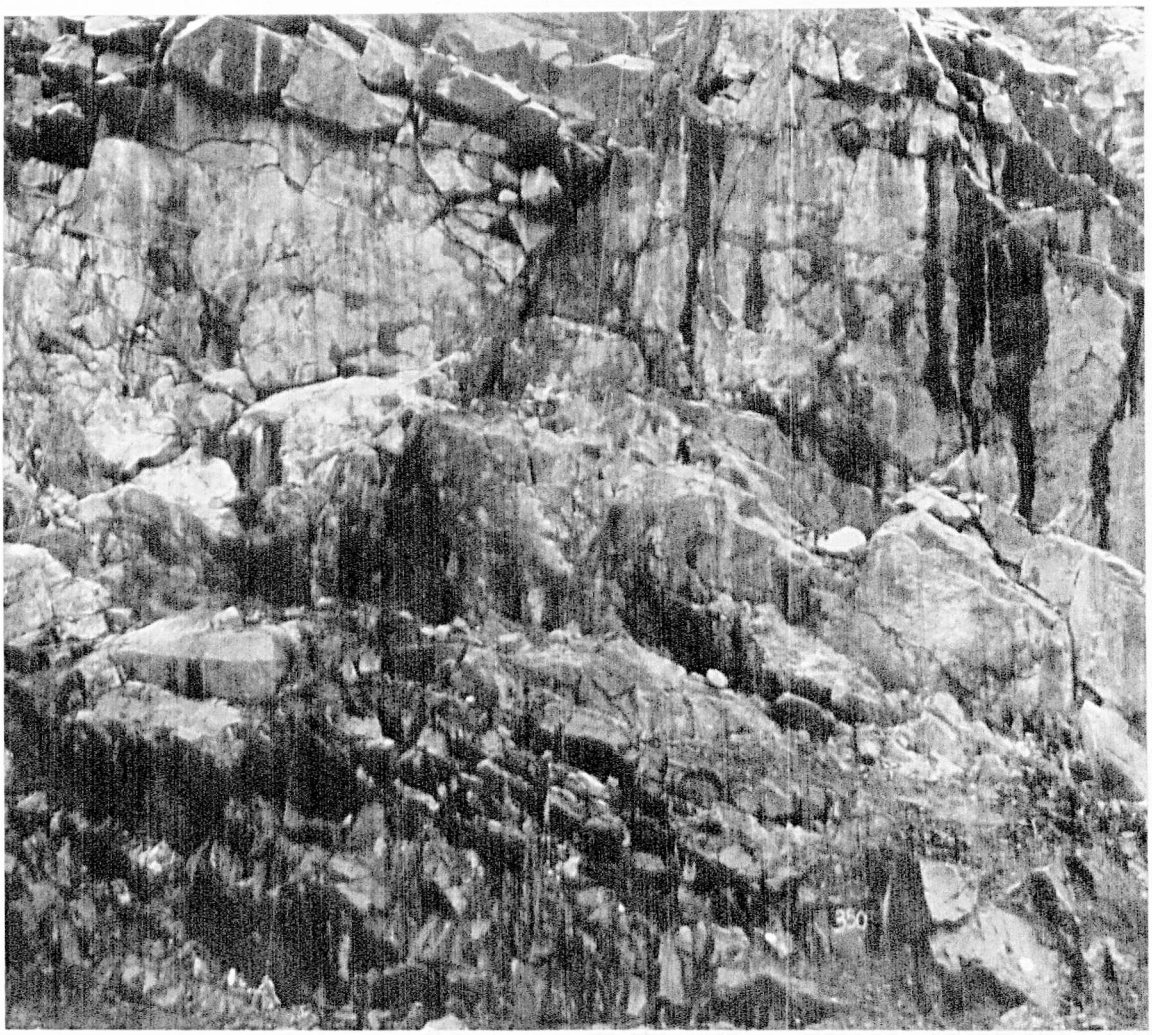


Figure 7.12 Granodiorite face photogrammetric survey: right hand stereophotograph

These orientation data were subsequently processed with the Author's program FADD, and Figure 7.14 shows the resulting cluster assignment plot for c=2. The cluster validity functions corresponding to this analysis are presented in Table 7.08 and discussed in sub-Section 7.3.3

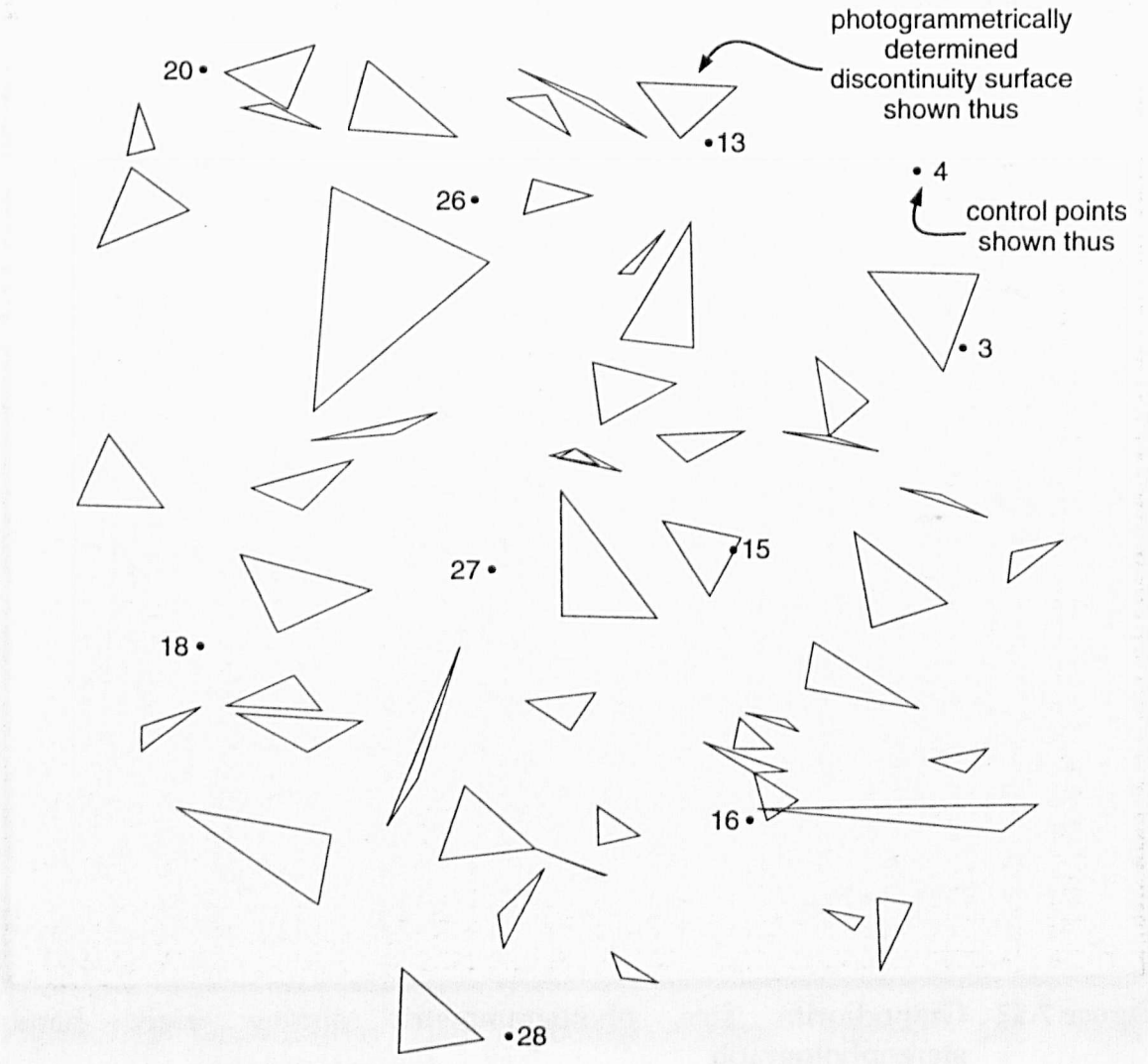


Figure 7.13 Granodiorite face photogrammetric survey: discontinuity triangles and control points.

Table 7.05 Granodiorite face photogrammetric survey: discontinuity surface coordinates.

1:	(104.504, 58.725, 103.511),	(103.768, 58.860, 103.090),	(103.215, 59.005, 103.394)
2:	(104.611, 58.323, 103.677),	(103.625, 58.858, 103.527),	(103.893, 58.387, 104.010)
3:	(104.049, 57.730, 104.660),	(104.504, 58.059, 105.465),	(103.067, 58.498, 104.954)
4:	(103.682, 57.479, 107.121),	(102.761, 58.144, 106.972),	(102.216, 57.940, 107.346)
5:	(103.652, 57.360, 107.560),	(103.564, 58.563, 109.965),	(101.703, 58.719, 108.966)
6:	(105.958, 58.258, 110.273),	(105.235, 58.583, 109.549),	(106.337, 57.860, 109.400)
7:	(104.874, 59.245, 110.984),	(104.138, 59.156, 110.601),	(103.843, 59.410, 111.314)
8:	(104.662, 58.720, 110.878),	(104.327, 59.045, 110.738),	(103.728, 58.933, 110.501)
9:	(103.411, 59.083, 110.411),	(103.197, 59.193, 111.290),	(102.152, 59.166, 110.338)
10:	(101.655, 59.770, 110.473),	(101.122, 59.514, 110.732),	(100.910, 59.683, 110.097)
11:	(100.206, 58.951, 108.763),	(99.933, 58.656, 108.932),	(99.641, 59.053, 109.315)
12:	(100.086, 58.482, 108.140),	(99.251, 58.667, 107.978),	(99.361, 59.169, 109.364)
13:	(100.418, 58.410, 107.862),	(100.298, 58.229, 107.150),	(99.463, 58.598, 107.541)
14:	(101.013, 59.017, 105.994),	(101.007, 58.813, 104.618),	(100.010, 59.055, 104.549)
15:	(101.056, 59.116, 106.347),	(100.393, 59.298, 106.141),	(100.735, 58.671, 106.651)
16:	(101.480, 58.991, 103.638),	(101.039, 59.122, 103.305),	(100.778, 59.218, 103.689)
17:	(105.473, 59.034, 103.048),	(105.498, 58.693, 103.364),	(104.887, 58.636, 103.594)
18:	(105.081, 59.108, 102.473),	(103.650, 58.991, 101.545),	(103.571, 59.460, 102.178)
19:	(102.869, 59.448, 100.903),	(102.853, 59.068, 100.120),	(102.081, 59.428, 100.242)
20:	(102.459, 59.214, 101.967),	(102.239, 59.521, 102.656),	(101.520, 59.418, 102.058)
21:	(100.750, 59.596, 100.800),	(100.833, 59.587, 101.042),	(100.444, 59.786, 100.753)
22:	(99.261, 59.334, 103.418),	(99.292, 59.159, 103.128),	(98.931, 59.367, 103.092)
23:	(98.639, 59.649, 101.451),	(98.317, 59.834, 101.364),	(98.423, 59.818, 101.239)
24:	(97.833, 59.443, 105.452),	(97.686, 59.326, 104.424),	(96.831, 59.399, 104.686)
25:	(98.316, 58.879, 103.846),	(98.295, 59.159, 104.292),	(97.221, 59.261, 103.562)
26:	(97.139, 58.811, 106.681),	(98.210, 58.619, 106.972),	(97.648, 58.638, 106.930)
27:	(97.743, 58.502, 107.146),	(97.840, 58.905, 107.805),	(97.235, 58.894, 107.274)
28:	(97.290, 59.337, 108.718),	(96.369, 59.066, 107.611),	(96.069, 59.573, 108.618)
29:	(100.768, 58.593, 106.732),	(101.134, 59.097, 106.380),	(100.619, 59.185, 106.260)
30:	(99.877, 59.073, 111.284),	(98.800, 59.527, 110.986),	(99.466, 59.305, 110.397)
31:	(99.584, 59.073, 103.209),	(98.740, 59.234, 102.883),	(99.202, 59.472, 102.822)
32:	(99.372, 59.561, 102.664),	(99.221, 59.778, 102.302),	(98.844, 59.701, 102.515)
33:	(102.990, 59.275, 102.292),	(102.679, 59.302, 102.761),	(102.046, 58.273, 104.389)
34:	(105.392, 57.995, 106.014),	(106.413, 57.532, 106.178),	(106.071, 57.897, 106.949)
35:	(105.887, 58.709, 110.864),	(106.004, 58.487, 110.298),	(105.691, 58.305, 110.525)
36:	(99.978, 59.613, 110.169),	(101.470, 59.592, 111.001),	(100.523, 59.342, 110.813)
37:	(98.095, 56.585, 102.883),	(97.019, 60.280, 102.147),	(96.283, 59.702, 102.496)
38:	(98.643, 59.060, 103.462),	(99.197, 59.354, 103.467),	(98.663, 59.445, 103.260)
39:	(96.929, 58.813, 106.228),	(95.936, 58.863, 105.876),	(96.244, 58.444, 106.299)
40:	(99.883, 59.070, 105.646),	(99.138, 59.457, 105.349),	(99.351, 58.925, 104.846)
41:	(99.655, 58.343, 106.990),	(98.826, 58.982, 106.807),	(99.407, 58.631, 106.540)
42:	(95.729, 58.905, 105.450),	(95.413, 59.409, 105.429),	(95.978, 59.176, 105.005)
43:	(97.180, 59.279, 103.010),	(96.910, 59.500, 102.844),	(96.682, 59.586, 103.079)
44:	(101.871, 59.207, 101.443),	(101.939, 59.793, 101.077),	(101.457, 59.519, 101.857)
45:	(104.389, 57.619, 106.430),	(103.792, 57.800, 106.101),	(103.233, 58.030, 106.669)
46:	(98.150, 59.812, 101.554),	(98.262, 59.972, 100.862),	(98.061, 60.356, 101.460)
47:	(100.932, 59.656, 101.776),	(101.415, 59.498, 101.944),	(100.734, 59.458, 101.802)
48:	(100.897, 59.541, 102.464),	(100.486, 59.593, 102.173),	(100.873, 59.422, 102.097)
49:	(101.275, 59.265, 109.764),	(101.351, 59.090, 109.413),	(100.576, 59.293, 109.571)

Table 7.06 Granodiorite face photogrammetric survey: discontinuity orientations.

192.6,86.1	023.3,55.9	209.4,57.6	027.2,41.0	199.1,65.3	212.3,81.3	183.0,71.5
272.8,21.8	184.1,83.9	343.2,70.6	107.0,44.1	196.5,69.7	196.7,73.7	194.2,81.8
352.3,37.4	197.8,88.3	016.9,44.7	200.2,56.9	201.2,65.3	190.2,69.1	213.0,81.2
212.2,65.6	026.4,75.3	182.8,83.1	206.8,61.7	358.8,53.6	210.8,59.0	192.9,67.0
352.5,37.9	021.8,86.0	341.5,37.8	006.2,61.2	068.3,59.3	206.3,77.2	134.5,67.1
337.3,54.5	043.3,20.7	331.3,31.6	337.9,40.0	218.8,60.8	035.6,78.7	056.3,51.2
033.2,77.3	070.9,55.1	198.8,86.3	078.7,78.5	327.5,23.2	200.5,71.9	189.5,65.5

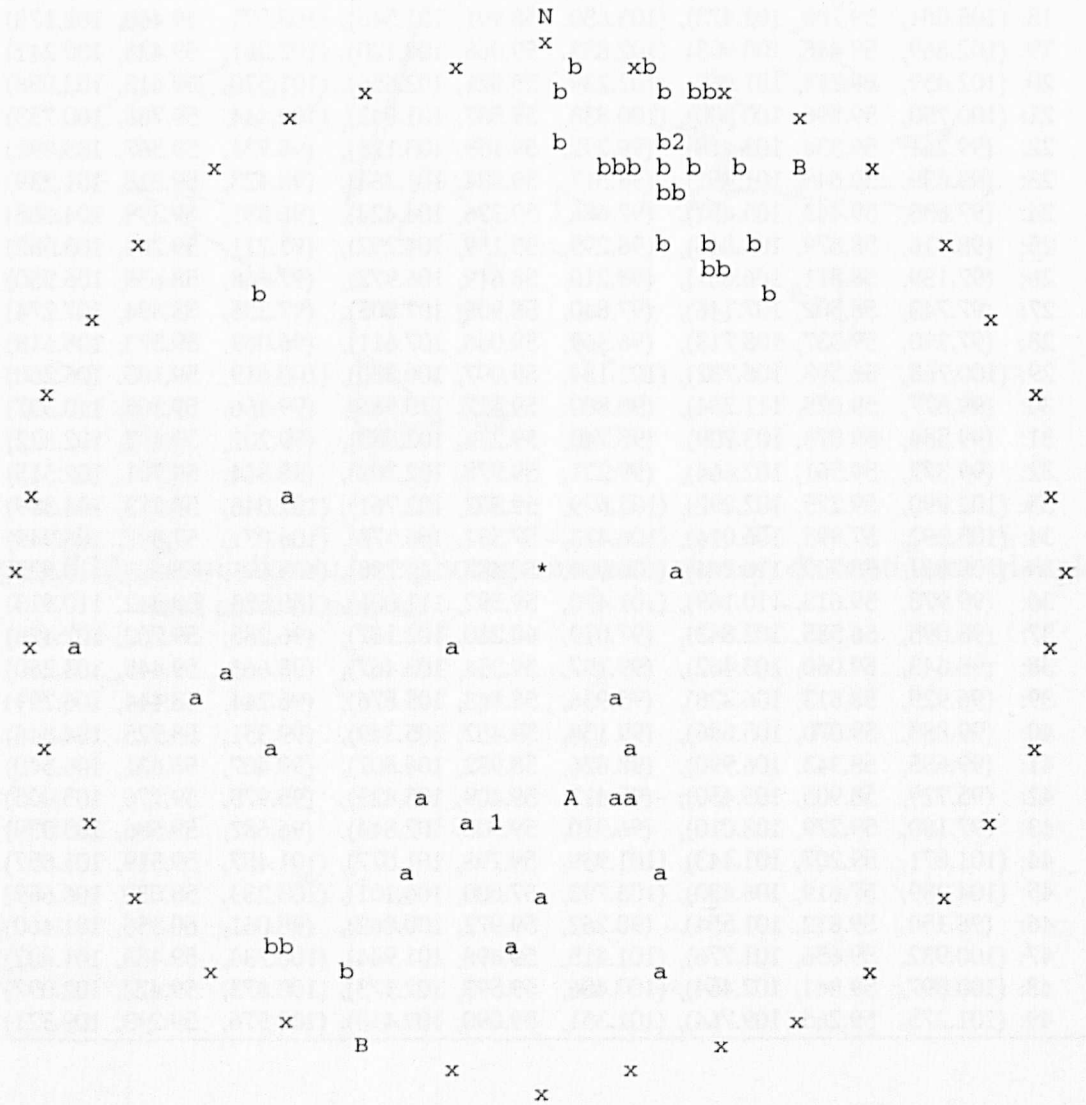


Figure 7.14 Granodiorite face photogrammetric survey: cluster assignment plot.

7.3.2 Compass survey

A total of fifty discontinuity orientation measurements were taken on the rock face in the vicinity of the control wires, at heights up to about 1.5m from ground level and distributed along the length of the exposure. These measurements are reproduced in Table 7.07, and an equal angle lower hemispherical projection of the corresponding pole densities—determined using Schmidt contouring with a 1% counting circle by the program DIPS—is shown in Figure 7.15. The results of a supplementary analysis, conducted using the Author's program FADD, are shown both in Figure 7.16 as a membership level plot for the case c=2, and in Table 7.08 as the magnitudes of the various cluster validity functions applied to the data.

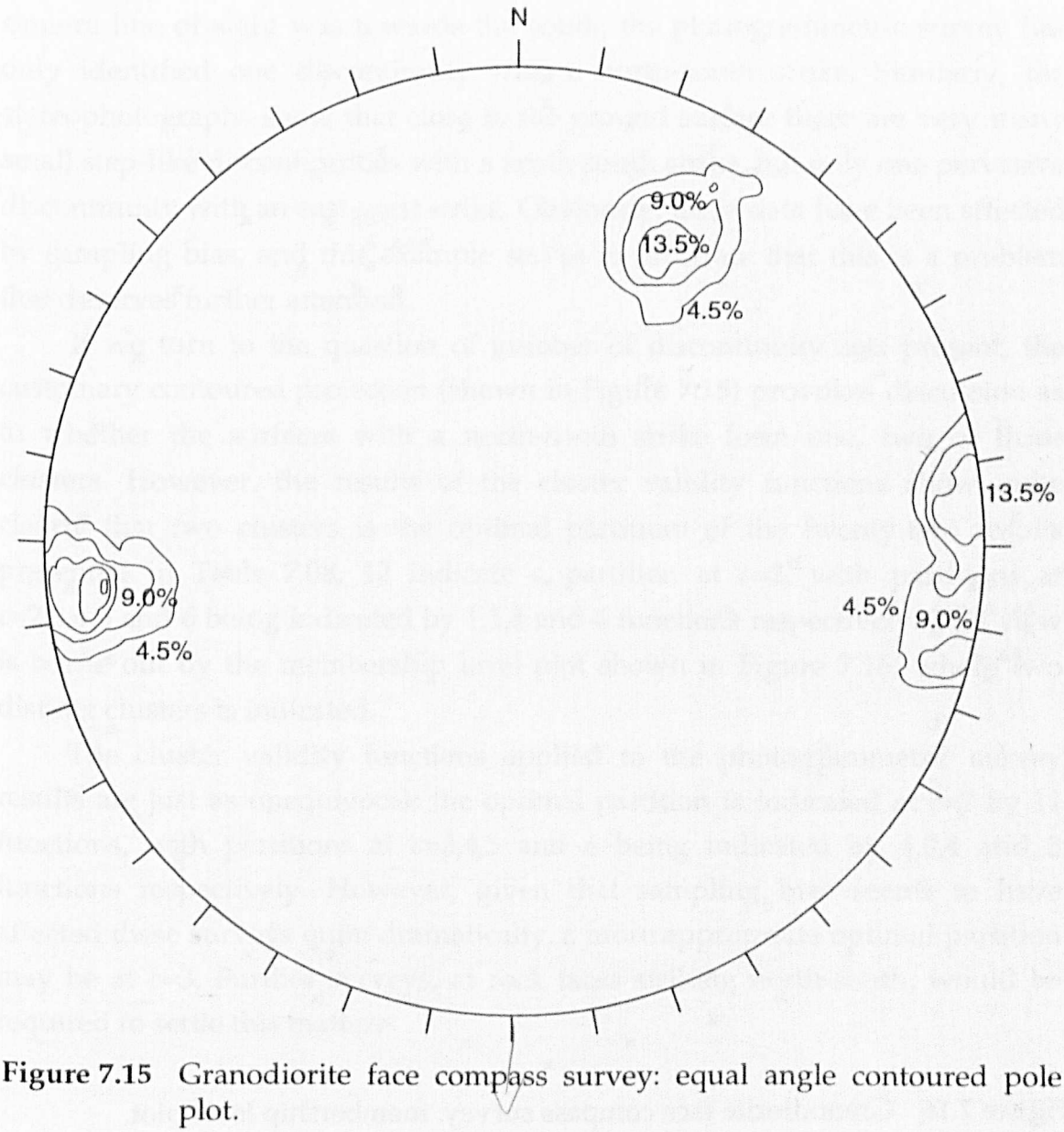


Figure 7.15 Granodiorite face compass survey: equal angle contoured pole plot.

Table 7.07 Granodiorite face compass survey: discontinuity orientations.

322,82	281,82	090,78	212,78	086,82	128,72	270,90	083,90	086,82	030,90
204,70	079,81	207,77	085,76	280,88	290,90	162,67	155,40	100,44	307,88
208,72	330,80	270,82	206,62	079,79	288,86	083,75	055,60	083,70	214,64
084,90	068,76	206,73	080,86	078,32	217,77	153,86	076,83	086,90	085,90
280,88	059,56	202,70	111,74	034,70	200,68	210,66	023,80	081,90	086,88

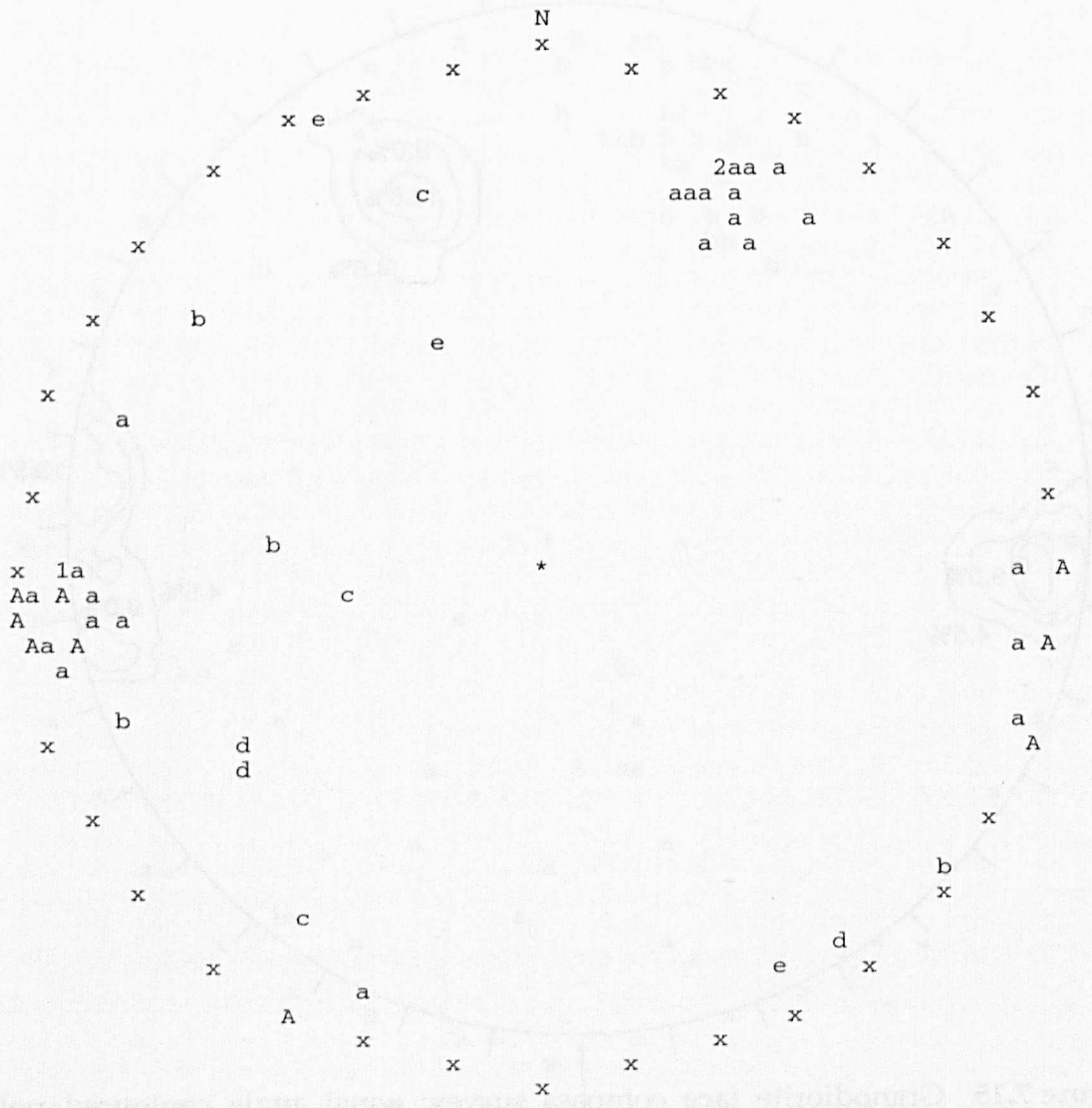


Figure 7.16 Granodiorite face compass survey: membership level plot.

7.3.3 Discussion

The photogrammetric survey has successfully sampled the entire rock face, as the distribution of discontinuity triangles in Figure 7.13 shows. Additionally, the presence of long and thin triangles in this Figure demonstrates that it has been possible to measure coordinates on surfaces that are almost parallel with the line of sight.

If hemispherical projections of the data resulting from both the photogrammetric and the compass surveys are compared, as for example by a study of Figures 7.14 and 7.15, it can be seen that two different views of the structure of the rock mass result: these probably result from sampling bias. In both cases, we can see that the sub-horizontal discontinuity set that dips towards the south-west has been clearly identified. However, because the camera line of sight was towards the south, the photogrammetric survey has only identified one discontinuity with a north-south strike. Similarly, the stereophotographs show that close to the ground surface there are very many small step-like discontinuities with a north-south strike, but only one pervasive discontinuity with an east-west strike. Obviously, these data have been affected by sampling bias, and this example serves to illustrate that this is a problem that deserves further attention.

If we turn to the question of number of discontinuity sets present, the customary contoured projection (shown in Figure 7.15) provokes discussion as to whether the surfaces with a north-south strike form one, two or three clusters. However, the results of the cluster validity functions show quite clearly that two clusters is the optimal partition: of the twenty-two results presented in Table 7.08, 12 indicate a partition at c=2, with partitions at c=2,3,4,5 and 6 being indicated by 1,1,4 and 4 functions respectively. This view is borne out by the membership level plot shown in Figure 7.16, where two distinct clusters is indicated.

The cluster validity functions applied to the photogrammetric survey results are just as unequivocal: the optimal partition is indicated at c=2 by 11 functions, with partitions at c=3,4,5 and 6 being indicated by 4,0,4 and 3 functions respectively. However, given that sampling bias seems to have affected these surveys quite dramatically, a more appropriate optimal partition may be at c=3. Further surveys, at rock faces striking north-south, would be required to settle this matter.

Table 7.08 Granodiorite face survey: cluster validity functions.

		no. of clusters	partition density	separation index	normalised partition coefficient	modified partition coefficient	partition entropy	partition coefficient
Compass Survey								
FCM	α=0.0	2		0.102	***0.630***	0.424	***0.090***	***0.815***
		3		0.069	0.607	0.474	0.126	0.738
		4		***0.253***	0.606	0.439	0.141	0.704
		5		0.248	0.604	***0.487***	0.151	0.683
		6		0.152	0.604	0.442	0.154	0.670
	α=0.9	2		1.222	***0.346***	***0.621***	0.023	***0.673***
		3		1.732	0.308	0.596	0.021	0.539
		4		1.875	0.278	0.587	0.021	0.459
		5		***1.937***	0.301	0.605	0.020	0.441
		6		1.129	0.237	0.574	***0.018***	0.364
FCV	α=0.0	2	1.254	0.102	***0.656***	0.413	***0.084***	***0.828***
		3	9.226	0.247	0.607	***0.457***	0.126	0.738
		4	16.629	0.253	0.574	0.444	0.153	0.680
		5	16.825	***0.284***	0.532	0.428	0.176	0.626
		6	***19.662***	0.083	0.547	0.387	0.176	0.622
	α=0.9	2	0.251	1.137	***0.394***	***0.620***	0.021	***0.697***
		3	5.643	1.304	0.293	0.595	0.026	0.529
		4	9.010	1.910	0.206	0.561	0.018	0.404
		5	7.625	***2.119***	0.177	0.540	0.019	0.342
		6	***12.144***	0.923	0.065	0.490	***0.010***	0.221
Photogrammetric Survey								
FCM	α=0.0	2		0.248	0.569	0.406	***0.106***	***0.784***
		3		***0.308***	***0.584***	***0.418***	0.133	0.723
		4		0.113	0.484	0.381	0.183	0.613
		5		0.121	0.467	0.378	0.199	0.574
		6		0.033	0.450	0.362	0.211	0.541
	α=0.9	2		1.032	***0.231***	***0.546***	0.029	***0.616***
		3		1.584	0.151	0.514	0.018	0.434
		4		1.231	0.069	0.477	0.012	0.302
		5		***1.853***	0.036	0.480	***0.008***	0.229
		6		1.476	0.111	0.505	0.013	0.259
FCV	α=0.0	2	6.835	***0.243***	***0.531***	***0.449***	***0.115***	***0.766***
		3	9.238	0.163	0.488	0.359	0.163	0.658
		4	16.010	0.101	0.438	0.389	0.197	0.578
		5	***19.566***	0.058	0.430	0.374	0.210	0.544
		6	16.180	0.032	0.429	0.396	0.218	0.524
	α=0.9	2	4.397	1.186	0.068	0.463	0.029	***0.534***
		3	3.831	***2.093***	0.033	0.456	0.016	0.355
		4	5.132	1.347	0.046	0.471	0.011	0.284
		5	***5.436***	1.303	0.054	0.473	0.011	0.243
		6	4.508	1.936	***0.070***	***0.483***	***0.010***	0.225

Optimal partitions shown emboldened and italicised

7.4 Synopsis

A report on the use of the new techniques for studying rock mass geometry developed by the Author has been given in this Chapter. The presentation has taken the form of a description of the techniques in use, followed by survey and analysis results from two quarry sites in the UK.

Section 7.1 contained a detailed description of the field and office procedures adopted to implement non-metric analytical photogrammetry for the purposes of determining rock mass geometrical data. This description led to the presentation of general guidelines for the installation of photogrammetric control, using a system of thin steel cables carrying markers, followed by a straightforward theodolite survey and subsequent sophisticated network adjustment. The control system and its method of adjustment have been developed by the Author to ensure that high-accuracy photogrammetric control points may reliably and safely be installed at a rock face.

The results of both a photogrammetric and a compass survey of discontinuity orientation at limestone and granodiorite quarries have been presented in Sections 7.2 and 7.3. These results demonstrate the general efficacy of the photogrammetric system and analysis using fuzzy partitioning algorithms. A number of important issues have been discovered as a result of this work, and these are highlighted below.

(i) The use of non-metric photogrammetry, in which photographic prints made from exposures taken with a 35mm camera are analysed with an analytical plotter of appropriate resolution, allows discontinuity orientation to be satisfactorily determined.

(ii) Photogrammetry permits sampling of discontinuity surfaces across the whole extent of a rock face, in contrast to the customary manual method whereby only those surfaces within about 1.5m of the ground will be sampled.

(iii) Discontinuities of all orientations may satisfactorily be surveyed, as the depth of field arising from the use of stereophotographs allows points on surfaces that are almost parallel to the camera line of sight to be coordinated.

(iv) A comparison with customary compass surveys has revealed that sampling bias exists in both methods, with—for either technique—the geometry of the discontinuities themselves playing a large part in determining the form and magnitude of the bias. For example, whilst a

customary compass survey cannot sample those surfaces beyond the reach of the observer, a photogrammetric survey cannot sample surfaces as small in areal extent as can a compass. As a consequence, the use of photogrammetry at rock faces possessing different orientations is indicated as a method of ameliorating the problem.

(v) Orientation analysis using the customary contouring techniques on the hemisphere can lead to difficulties in determining the number of discontinuity sets present. This is in direct contrast to the use of fuzzy partitioning algorithms, where the associated cluster validity functions allow an objective choice of the number of sets to be made.

(vi) A pragmatic approach must be taken when cluster validity functions are used to determine the optimal partition of discontinuity orientation data, as none of the customary functions categorically indicate what the optimum is. Thus, the use of a number of functions is necessary, with sharpening thresholds of both zero and 0.9 being applied. On this basis, the optimal partition may be identified in terms of a majority indication.

(vii) The use of membership level plots, in conjunction with identification of the optimal partition *via* the use of cluster validity functions, allows the extent of clusters to be easily determined in terms of their degree of certainty.

In conclusion, the work of this Chapter has shown how both the photogrammetric systems and the fuzzy partitioning scheme developed by the Author can be used to improve the analysis of rock mass geometry. The fact that there are a number of issues that bear further investigation is recognised, and these—together with a number of others that are related to other aspects of this work—are outlined in the following Chapter.

Part IV
Topics for Further Research & Conclusions

This page intentionally left blank

8 Topics for Further Research

During the course of the work described in the earlier Chapters of this thesis, a number of interesting avenues for further work—with respect to the measurement and analysis of rock mass geometry—have become apparent. These are presented in this Chapter, and have been grouped into three Sections on the basis of their major area of research: development of non-metric photogrammetry; development of fuzzy partitioning algorithms; and the advanced study of rock mass geometry.

8.1 Further Development of Non-Metric Photogrammetry

The work presented here in Chapters 5 and 7 has shown how non-metric analytical photogrammetry may successfully be applied to the measurement of discontinuity geometry. The techniques and equipment used have been found to be acceptable—in terms of accuracy and utility—for the measurement of discontinuity orientation, but in order to successfully extend the application to the other three principal geometrical attributes of spacing, area and roughness, further research is required into the overall behaviour of the photogrammetric system. Seven aspects of the work suggested are briefly described in the following sub-Sections.

8.1.1 Optimisation of the DLT

The formulation of the DLT used in this work follows directly from the original work by Marzan and Karara [94], with the only enhancements being those offered through the incorporation of a singular value decomposition—rather than the assembly of the normal equations—for the solution of the least squares problem. Two other enhancements, more directly related to the formulation of the DLT, have been suggested recently in the literature.

Firstly, Hatze [57] considered the use of non-linear constraints in the solution of the least squares problem as a means of improving the accuracy of

the DLT. The motivation for the inclusion of such a constraint came from the realisation by Hatze that the customary formulation of the DLT does not maintain orthogonality of the transformation matrix between object space coordinates and photograph coordinates. To overcome this deficiency, Hatze derived a non-linear relation among the eleven DLT parameters such that this condition was fulfilled. The numerical implementation of this improved formulation required an iterative solution, but no reference was made to the length of time required for its execution. The improvement in accuracy was, nonetheless, dramatic, with root mean square position errors reported as being about 15% of those achieved with the customary formulation.

More recently, Naftel and Boot [98] reported on an iterative enhancement to the customary DLT formulation. This enhancement stemmed from the realisation that, whenever a non-control point is coordinated, a condition equation based on its measured photograph coordinates and computed object space coordinates can be included in the DLT, albeit with lower weight than an equation formed using a control point. The justification for this comes from the fact that the DLT parameters should be able to fit all of the data in the problem, not just the initial control points. The use of this technique reduced mean root mean square errors to less than 30% of those resulting from the use of the customary DLT formulation, although this was at the expense of requiring 20 or more iterations for each point entered.

How either or both of these enhancements could improve the photogrammetric problem posed by non-metric photography at a rock face is unknown. Whether the improvements would be justified, given the undoubtedly longer program execution times, is also unknown. The real-time nature of the DLT implementation described in Chapter 5 of this work is important, and accuracy improvements accompanied by a reduction in response times may not be acceptable.

8.1.2 Total least squares applied to the solution of the DLT

Customarily, the solution of photogrammetric problems is undertaken on the basis of errors only being present in the elements of the vector of independent variables. Thus, in an overdetermined set of equations

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} = \mathbf{v} \tag{8.01}$$

the classical least squares approach assumes the matrix $\mathbf{A}$ to be without error; hence, all of the errors are associated with the observations contained within

the vector **b**. This is certainly not the case with the DLT for, as (3.15) shows, the elements of the matrix **A** are computed using measurements of the object space coordinates of the control points, or measurements of their photograph coordinates, or both. Consequently, it is fair to say that, in the case of the DLT formulation, errors of equal magnitude are contained in both sides of (8.01).

The solution of such problems requires the method of **total least squares** (TLS). Much valuable work has recently been undertaken on the numerical implementation of this method [137] in conjunction with singular value decomposition, such that it is now a readily applicable and well-understood analytical technique. The application of TLS to the DLT formulation has, however, not yet been undertaken, and so its effect is unknown. It may be that the use of TLS will improve the accuracy of the DLT with only a modest increase in the computational requirements.

8.1.3 Absolute performance of the analytical plotter

The efficacy of the analytical plotter has been demonstrated through its application to the analysis of discontinuity orientation, but its absolute performance characteristics have not been studied. This is a not unimportant point, as completely objective analysis of the errors introduced through the use of this instrument cannot be performed without this information.

Determination of these characteristics is, however, not a trivial operation, given the complexity of the instrument. As a minimum, the positioning characteristics of the analytical plotter can be considered as being primarily a function of the behaviour of both the stepping motors and the couplings between the lead screws and carriages running on the linear bearings. Whilst the behaviour of the coupling can be easily modelled, assuming that the springs within them respond in a linear manner, the same is not true of the stepping motors.

The behaviour of stepping motors when used for accurate positioning has been studied by Acarnley [2], who shows that the positional errors are related not only to the load being positioned and the linear position increment per motor step, but also to the choice of excitation scheme for the motor windings. In the case of the instrument described here in Chapter 5, the excitation scheme can be regarded as being composed of two closed-loop control schemes in parallel: the first being the chopper driver, and the second the positioning control system. The behaviour of such a closed loop system is highly non-linear [16], and so the analysis of its positioning characteristics is not straightforward.

It is likely that an analysis of the positioning characteristics of the analytical plotter will only be possible on the basis of a numerical realisation of the extremely complex mathematical model describing its behaviour.

8.1.4 Automation of the measurement process

A great improvement in the utility of photogrammetry could be achieved if the method could be automated. The prime candidate area for the application of automation is in the measurement of photograph coordinates, as apart from the acquisition of the photography, this is the only part of the process that currently requires direct human intervention. If the identification of homologous points on two stereophotographs could be achieved by machine, then the measurement of photograph coordinates could continue unattended. This would realise great benefits in terms of the quantity of high quality geometrical data available for the analysis of rock mass geometry.

A great deal of research into this problem has been undertaken in the wider field of photogrammetry, and it is reasonable to expect that robust and fast algorithms for 'stereomatching' will soon be available. Once this is so, it will then be a case of adapting the analytical plotter to use this technique, through the introduction of two video cameras to digitise the photographs.

One of the problems associated with the use of video cameras will be the requirement to ensure that photograph coordinates can be obtained with an error of less than 25μm (the threshold identified in Section 5.1 of this work as being adequate for orientation measurements). Currently, non-specialist video cameras are capable of digitising images with a resolution of 1024×1024 pixels; in the case of a photograph displaying a rock face at a scale of 1:80, this represents a region only about 25mm square. Consequently, it will be necessary for the matching algorithm to be embodied in a program that is capable of moving the cameras to other regions of the photographs. This requirement was foreseen during the design stage of the analytical plotter, and has therefore been incorporated into the operation of the closed-loop control system.

8.1.5 Application of digital photogrammetry

The main thrust of modern photogrammetry is in the development of completely digital techniques, whereby images are stored and analysed completely digitally on high-speed computers. The application of such techniques to photogrammetry for rock mass geometry is potentially very useful, although a number of difficulties must be resolved in order to do so.

The previous sub-Section noted how very small pixel sizes will be required if adequate resolution is to be maintained. Digitising a 150mm×150mm photograph at such a scale will result in 36 million pixels, and if each pixel is to be stored by a computer as a single byte, then a file some 34Mbytes in size will result. For stereomatching, a pair of these files is required. Furthermore, this assumes that each pixel can be adequately described by the 256 values possible for an eight-bit byte. If more values are required (as, for example, with 24-bit colour), then much larger files will result.

The problems associated with the manipulation of such large files are currently the subject of research in photogrammetry at large, and developments in this field should be monitored for possible application in rock mechanics.

8.1.6 Further study of the photogrammetric configuration at a rock face

The method, developed as part of this work, of introducing photogrammetric control through the use of coordinated points on steel cables has worked well. However, further work is required in order to determine how, within the constraints generated by the relative positions of a rock face and the camera stations, the cable markers should be disposed in order that optimal photogrammetric solutions may be obtained.

A number of workers have studied this problem in a general sense (see, for example, Cooper [30] and Marzan [93]), and it is well known that the control points for the DLT should be well disposed around the object and as far from coplanar as possible. Unfortunately, neither of these conditions can easily be obtained at a rock face, and so studies should be undertaken to determine what minimum configuration is necessary to maintain the required accuracy of the photogrammetry for the purposes of analysing rock mass geometry.

8.1.7 Simplified methods of introducing photogrammetric control

Although based on the simple concept of markers supported in front of a rock face by tensioned steel cables, the procedures used during this work for introducing photogrammetric control at a rock face require modification in order to become appropriate for routine use. Experience has shown that, whereas installing the cables and taking the photographs took only about 45 minutes, surveying the markers with a theodolite required about 3 hours. This was in part due to the care taken to ensure that a surfeit of high quality coordinated points would be available for subsequent photogrammetric

analysis, but was also due to the inherently slow procedures required when using a theodolite for surveying.

Two methods of improving the speed of the surveying process deserve further evaluation. The first is the use of a modern 'total station' for the surveying process, and the second is to develop techniques that obviate the use of any specialised surveying equipment.

Total stations are extremely sophisticated instruments, combining a theodolite for angle measurement and a laser for distance measurement, and are capable of automatically storing all angle and distance readings on a portable computer attached to the instrument. In order to successfully use such an instrument in the context of this work, the markers on the control cables would need to be covered in material that was reflective to the laser light, but this can easily be achieved. The drawback of these instruments is their current high cost, although this may change in the future.

A scheme that would not require any surveying equipment could be implemented using a combination of a simple cable network, a geological compass for determining orientation, and a sophisticated adjustment procedure similar to that described in Section 5.3 of this work. As an example, four steel cables connected together to form a tetrahedron, and carrying markers at each apex and along each side, could be suspended in front of the rock face using guys from each apex to ensure that all of the cables were in tension. Following determination of the orientations of these cables with a geological compass, and knowing the lengths of the cables and the marker spacings, a survey adjustment would allow determination of the marker coordinates relative both to each other and magnetic North. This information would be sufficient to determine discontinuity coordinates photogrammetrically, such that all but absolute location were known.

It is not known whether such a procedure would be feasible, either in terms of ease of use, marker coordinate accuracy, or stability of the subsequent photogrammetric analysis. It does, however, seem to offer promise as a means of overcoming the slowness associated with theodolite surveys.

8.2 Further Development of Fuzzy Partitioning Algorithms

The work presented here in Chapter 6 has demonstrated the applicability of fuzzy partitioning algorithms to orientation data. These data represent only one of the four principal attributes of discontinuity geometry, and in order to be

able to apply fuzzy algorithms to the complete suite of geometrical attributes, further work is required. Particular areas are identified and briefly described in the following seven sub-Sections.

8.2.1 The use of Mahalanobis angles

In order to identify anisotropic clusters, the program FADD incorporates Mahalanobis distances, with the norm-inducing matrix being computed from the covariance matrix of the data. The discussion on the geometry of orientation data—presented in sub-Section 6.1.1—concluded that the lack of such a distance metric in Riemannian geometry implied that the analysis of such data must be conducted in Euclidean space, if anisotropic clusters are to be identified. The connection between Riemannian and Euclidean geometries—in terms of the angles subtended between two vectors at the coincident origin of these two coordinate systems—was also highlighted in sub-Section 6.1.1. Thus, if anisotropic angle measurement could be incorporated in Euclidean geometry, then this link would allow the analysis to be undertaken in terms of the intrinsically correct Riemannian geometry.

An investigation into the use of **Mahalanobis angles** as a means of implementing such an enhancement is suggested by this reasoning. These angles are defined by Mardia [91] as

$$\cos\theta(\mathbf{x}_1,\mathbf{x}_2,\Sigma)=\frac{\mathbf{x}_1{}^T\Sigma^{-1}\mathbf{x}_2}{d(\mathbf{x}_1,\mathbf{0},\Sigma)\cdot d(\mathbf{x}_2,\mathbf{0},\Sigma)} \tag{8.02}$$

where the angle θ is the Mahalanobis angle subtended at the origin between the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, drawn from a population with covariance matrix Σ, and the operator $d(\mathbf{i},\mathbf{j},\mathbf{A})$ represents the customary Mahalanobis distance between the vectors $\mathbf{i}$ and $\mathbf{j}$, as induced by the matrix $\mathbf{A}$. The method of application of this function is not obvious, as the physical interpretation of the null vectors $\mathbf{0}$ in (8.02) is not clear. Further work may reveal that the vector representing the cluster centroid can be substituted for this vector.

8.2.2 Further study of cluster validity functions

Incorporation of a reliable cluster validity function into a partitioning algorithm is essential, as it is through a comparison of the magnitude of such a function at different partitions of the data that optimal partitions are identified. A number of validity functions have been tested with FADD and, as Figures 6.09 to 6.17

and Table 6.09 show, no one of them can be regarded as truly satisfactory from the point of view of identifying optimal partitions in orientation data. This may be due to the Riemannian nature of orientation data, whereas these functions have been developed for Euclidean data.

At the least, therefore, work is required to develop and identify reliable cluster validity functions that are applicable to Riemannian data. More satisfactory would be the identification of functions that are applicable to data that may be considered as a combination of Riemannian and Euclidean, as complete discontinuity geometrical data are.

8.2.3 Other partitioning algorithms

Recent work by Dave [31] and Dave and Bhaswan [32] has resulted in the fuzzy *c*-shells algorithm, a new clustering technique capable of detecting circular and elliptical ring-like clusters in data. This work has been motivated by the need to identify circular objects in digital images, but may be applicable to the identification of clusters on the surface of the unit sphere.

At first sight, it appears that the *c*-shells algorithm would identify all orientation vectors as being members of the same cluster, as they all possess the same radius relative to the origin of the unit sphere. However, if some other attribute is attached to this data—for example, another geometrical property such as roughness—then it may be possible to interpret this as a modifier to the radius, and hence partition the data accordingly.

The development of a hybrid scheme incorporating fuzzy *c*-shells and fuzzy covariance is intriguing. Should one algorithm be applied before the other, or should they be applied simultaneously in an iterative fashion? Should only certain geometrical attributes be used with each of the schemes (for example, orientation with fuzzy covariance and the remainder with fuzzy *c*-shells)?

8.2.4 Interpolation of partitioning results

One of the difficulties identified with the generation of graphical output from FADD has been that of precisely locating contours of membership values for orientation data. In regions of dense data no problem results, as the membership values applicable to the individual data can be interpolated with confidence. However, in sparsely populated regions this is not the case, for although the existence of zones in which data would possess low membership values can be inferred from the shape of clusters, the membership values they

would take cannot be determined. A geographical analogy to this problem is being able to infer the existence of a deep valley from the terrain, but not knowing how deep it is.

Although this problem can be overcome in the geographical case through the use of simple extrapolation, this technique fails for fuzzy membership values. The reason for this is that such a procedure can only be undertaken by computing the membership value of a dummy datum, but the very introduction of a dummy datum into the partitioning analysis will change the form of the membership matrix.

It is possible that the use of advanced interpolation procedures, applicable to data on the surface of the sphere (see, for example, Wahba [138] and Dierckx [36]), may overcome this problem. Even if such procedures are successful in the case of orientation data, it will be interesting to examine their behaviour with data containing more than three dimensions (i.e. data comprising additional geometrical attributes).

8.2.5 Characterising fuzzy clusters

One of the major uses for the results of a fuzzy partitioning analysis will be as geometrical input to stability analyses of engineered structures in rock. To conveniently use partitioning results in this manner, a method is required that characterises the fuzzy clusters in terms of a surface that represents membership values. This is a similar problem to that of interpolation, as discussed in the previous sub-Section. A number of questions can be raised about the form of the membership value surface. For example, does the application of the Mahalanobis norm result in contours of membership that are elliptical, and do all surfaces have the same cross-sectional shape? These two questions are particularly important, for they will have ramifications for the extent of orientations over which mechanical instability is displayed.

Of equal importance, and connected with the assessment of partition validity, is the problem of determining the 'quality' of individual clusters. An interesting way to accomplish this would be to use the cluster-specific values of components of various cluster validity functions. Examples of suitable candidates are: the magnitude of the partition coefficient, equation (4.42); the determinant of the fuzzy covariance matrix, equation (4.33); and the sum of the cluster's central members, equation (4.59). Being able to rank clusters in this way will assist the interpretation of rock mass geometry, as we can regard these

measures of cluster quality as being representative of the importance of the associated discontinuity set.

8.2.6 Higher dimensional clustering

The efficacy of the customary formulations of the fuzzy *c*-means, fuzzy covariance and fuzzy maximum likelihood estimation algorithms when applied to multi-dimensional data in Euclidean space is well known. For example, Gath and Geva [46] applied the fuzzy maximum likelihood estimation algorithm to good effect with electroencephalogram data in five dimensions. Similarly, in Chapters 6 and 7 of this work, the successful application of these algorithms to orientation data, which is Riemannian in nature, has been demonstrated.

As the discussion presented in Section 1.2 demonstrated, exploiting the interdependence among the four principal geometrical attributes of discontinuities may prove beneficial to the understanding of the geometry of rock masses. However, with discontinuity geometrical data, the attributes of spacing, area and roughness are Euclidean, and orientation is Riemannian. Thus, for the purposes of partitioning algorithms, complete discontinuity geometry may be thought of as five-dimensional: three Euclidean dimensions and two Riemannian dimensions (because orientation is a two-dimensional Riemannian phenomenon).

At present, the behaviour of the algorithms presented in Chapter 6 when faced with data that may be regarded as a hybrid of Euclidean and Riemannian is unknown. Further exploration in this area is essential if partitioning of complete discontinuity geometrical data is to be achieved.

8.2.7 Non-metric methods

The use of non-metric partitioning algorithms was discussed briefly in sub-Section 4.4.5. These methods are appropriate when measures of distance between data are inapplicable, for example when the vectors comprising the data have different physical units attached to their components. This is the case with complete discontinuity geometry data, where coherence does not exist amongst the dimensions of the four principal attributes of spacing, roughness, extent and orientation. As far as this Author has been able to determine, no fundamental studies of the use of non-metric partitioning algorithms in rock mass geometry have been carried out. It may be that their application in this field will be productive.

8.3 Advanced Study of Rock Mass Geometry

The new technique of integrating non-metric photogrammetry and fuzzy partitioning algorithms has been used here to determine orientation of discontinuities. Work on the analysis of the other three principal geometrical attributes of spacing, roughness and extent—either singly or in combination—is still to be performed. In this Section, interesting aspects of this future work are described, together with ramifications for the analysis of rock mass geometry in general. These aspects have been grouped into five sub-Sections: determination of spacing, extent and roughness from photographs; interrelations between extent, orientation and spacing of discontinuity surfaces; the analysis of scanline techniques on rock faces; visualisation techniques for complete rock mass geometry; and enhanced stability analyses.

8.3.1 Determination of spacing, extent and roughness from photographs

The use of photogrammetry for the determination of spacing and extent raises interesting questions of interpretation, the like of which are not usually encountered when using customary manual rock mass surveying techniques. Because a photogrammetric system allows coordination of a number of points on a discontinuity surface, questions arise as to how this data should be analysed, if only single-valued measurements of spacing and extent are required. Whilst it will be possible to fit a plane to such data using least squares techniques, if two adjacent planes are found not to be parallel, then their spacing is not uniquely defined. Similarly, the question of characterising extent, when the coordinates of the perimeter of a discontinuity surface (as exposed at a rock face) are known, will prove difficult to answer. Whilst the customary analysis based on trace length will still be applicable to line traces produced on such surfaces, analysis in the wider context of the areal extent of arbitrary shaped polygons will be the more appropriate. How this analysis should be undertaken is unknown.

Roughness has been determined photogrammetrically in the past, but only through the use of analogue techniques to produce contour maps or linear profiles of discontinuity surfaces. Analytical photogrammetry does not lend itself directly to contouring or profiling, but does permit 'heighting' of discrete points to take place. This is the basis for the production of digital terrain models through photogrammetry, and it will be interesting to attempt such an application for roughness measurement. One method of doing so will be to

coordinate triads of points on a discontinuity surface (ensuring that the triads are of varying sizes), compute the orientation of the normal vector to each triad, and finally apply spherical multivariate analysis to the resulting orientation data. This approach should permit determination of roughness anisotropy and scale effect [56].

8.3.2 Interrelations between extent, orientation and spacing of discontinuity surfaces

The fact that, in a rock mass, the geometrical attributes of extent, orientation and spacing are interrelated was discussed in Section 1.2 and illustrated in Figure 1.02. The integrated system developed here will allow objective analysis of this aspect of rock mass geometry.

Exactly what form the correlation among these three attributes takes is unknown. If, for example, most discontinuities terminate at other discontinuities and are contained in distinct sets, then a 'blocky' rock mass will result, and this should be indicated by some form of high correlation between extent, orientation and spacing. The complementary case—where the extent of discontinuities is not correlated with spacing, and distinct sets are not indicated—probably represents a 'rubbly' rock mass.

8.3.3 The analysis of scanline techniques on rock faces

Although a great deal of measurement of rock mass geometry has been undertaken using scanlines at rock faces, and no doubt much more will be done in the future, extensive field studies of the efficacy of this method with respect to three-dimensional forms of measurement have not been reported. This is probably due to the difficulty of taking complete three-dimensional measurements of a rock mass: drilling boreholes is prohibitively expensive; non-invasive techniques such as geophysics or tomography are either of insufficient resolving power or are not sufficiently developed; and the use of manual measuring techniques at rock faces are too time consuming and inherently dangerous.

The techniques developed as part of this work will allow rapid and thorough three-dimensional measurement and analysis of discontinuity orientation and location at rock faces. It will be possible to use the resulting spatial data to produce linear samples at all locations and orientations on the rock face, such that detailed analysis of the bias inherent in these sampling strategies may be better understood. Of course, the extension to window

sampling is straightforward, and in both cases valuable data for the verification of stochastic modelling schemes for mass geometry will result.

8.3.4 Visualisation techniques for complete rock mass geometry

The most widely used visualisation technique in rock mass geometry—the hemispherical projection—is really only appropriate for orientation data, although it is possible to make use of colour and shading to help portray additional information (such as discontinuity genesis). The measurement and analysis system developed as part of this work will ordinarily be able to work with data representing the complete rock mass geometry and, as was discussed in sub-Section 8.2.6 above, at a minimum this data can be regarded as being five dimensional.

In order to permit full comprehension of the nature of complete rock mass geometry, new and enhanced visualisation techniques will be required, suitable for displaying higher dimensional data. Although some form of enhanced hemispherical projection would seem to be the most acceptable, on the basis of it being a natural progression from the techniques currently in use, further work is required to determine whether or not such an approach is optimal.

8.3.5 Enhanced stability analyses

The use of the computer program FADD to partition orientation data into sets raises interesting questions about how the results may be applied to the further analysis of discontinuity orientation. A simple example concerns both the kinematic and the static analysis of rock block movement, in terms of sliding along planes and intersections between planes.

Customarily, contoured orientation data is used to determine the mean orientation of discontinuities in a particular set, and this orientation used in both kinematic and static analyses. This technique is well known and understood, but its extension to the analysis of sliding along lines of intersection is not. Matheson [95], for example, suggests a method whereby the orientations of intersections are determined vectorially and subsequently contoured, with the contribution made by other geometrical attributes such as extent and spacing recognised and used to 'weight' the resulting contoured plot of intersections. However, the use of FADD—operating with complete rock mass geometrical data—will allow this work to be extended in a novel fashion, with two possible routes for further analysis being immediately apparent.

One method would be to identify those discontinuities with set membership values of, say, more than 0.9 and use these in the computation of intersection orientation. Another method would be to use all discontinuities, regardless of set membership, and then cluster and contour the resulting intersection data. Which of these approaches is the more valid, or whether they will yield similar results, is unknown.

How contours of set membership in orientation data relate to the stability of rock blocks is also unknown. Intuition suggests that these contours could be used to form some form of geometrical factor of safety, in which case the utility of static and—especially—kinematic methods of stability analysis would be much improved. Further work in this direction may prove to be extremely useful to the rock engineering community.

8.4 Synopsis

This Chapter has briefly surveyed a number of possible areas for further research, in terms of extension to the work undertaken as part of this project. These can conveniently be grouped into work associated with the application of non-metric photogrammetry and fuzzy partitioning algorithms to the measurement and analysis of rock mass geometry, and the form of rock mass geometry itself.

Work in the fields of optimising the DLT, and the application of total least squares to the solution of the DLT, should improve the accuracy of the photogrammetric process. In a similar vein, determination of the absolute performance of the analytical plotter will allow more precise positioning and hence coordinate measurement to be achieved. Automation of the measurement process, either through the use of video cameras and stereo matching algorithms, or through the development of digital photogrammetry, will improve the utility of the process. Further study of the photogrammetric configuration at a rock face, including development of simplified methods of introducing photogrammetric control, will allow increased productivity of the field operations.

A number of interesting aspects related to the application of fuzzy partitioning algorithms for the analysis of rock mass geometrical data have been described. The use of Mahalanobis angles as a means of incorporating Riemannian geometry for orientation data, further study of validity functions and the implementation of more sophisticated partitioning algorithms are all

worthy of further investigation. From the point of view of enhanced application of the techniques developed so far, interpolation of membership values resulting from partitioning analysis, the characterisation of fuzzy clusters and the extension to higher dimensional data are all aspects that warrant study. Finally, the development of non-metric methods as a means of analysing data of different dimensions may be particularly useful in the field of rock mass geometry.

The aim of this work has been to develop photogrammetric and mathematical methods for the improved measurement and analysis of rock mass geometry. However, although advanced analysis of rock mass geometry may now be undertaken using these methods, further development of visualisation techniques for complete rock mass geometry is required in order that the results may be more easily understood. Four analyses relating to rock mass geometry have been described: determination of spacing, extent and roughness from photographs; interrelations between extent, orientation and spacing of discontinuity surfaces; and the analysis of scanline techniques on rock faces. Perhaps the most intriguing, and potentially useful for wider application in rock mechanics, is the use of fuzzy membership values in enhanced stability analyses.

Thus, this Chapter has demonstrated how the application of non-metric photogrammetry and fuzzy partitioning algorithms to problems in the measurement and analysis of rock mass geometry has revealed complete new fields of research within the subject area of engineering rock mechanics.

This page intentionally left blank

9 Conclusions

This work has been directed towards developing improved analysis of rock mass geometry through the use of mathematical and photogrammetric methods. Throughout the thesis, discussion has been presented at appropriate points, in order to permit specific conclusions to be drawn and hence allow the theme to develop. Thus, although the main conclusions that can be drawn from the work are specifically related to the analysis of rock mass geometry, the wide-ranging nature of the work means that there are also a number of conclusions that may be appropriate to other fields. However, it is impossible to separate what is and what is not directly relevant to the analysis of rock mass geometry, because this depends as much on the beliefs of the reader as on the conclusions themselves. Consequently, all of the conclusions that can be drawn from this work are given in the following series of numbered points, in the order that they appear in the document.

(i) The geometry of rock masses can be considered in terms of the four principal geometrical parameters associated with discontinuity surfaces, namely: spacing, roughness, extent and orientation. With the current techniques available for the measurement of rock mass geometry, the properties of these principal parameters are not determined directly, but rather in terms of how they are affected by one another. Unfortunately, customary analytical methods are inadequate, in that they can neither properly deal with such measurements, nor allow a rational holistic approach to rock mass characterisation on the basis of all four geometric parameters. Furthermore, most customary techniques for the measurement of rock mass geometry rely on manual methods, and these are slow, prone to sampling bias, and may expose the observer to danger. The requirement for improved methods is thus clearly indicated.

(ii) Given that improved measurement and analytical techniques are required, it is worthwhile to impose four basic constraints, such that they: demonstrate universal applicability; are straightforward in use; use easily

acquired equipment; and offer improved speed and quality of results over customary methods.

(iii) A detailed study of the customary analysis of orientation data through the application of contouring techniques has revealed a number of deficiencies. These include: the use of a step function for determining cluster membership; the implicit use of the Euclidean metric for determination of distance; the use of arbitrarily-sized counting circles; the lack of evidence to confirm that mean orientation coincides with the orientation of a contoured maximum; the application of arbitrary significance thresholds when performing statistical tests to determine cluster existence; and testing for cluster validity in terms of a fit to specific probabilistic distributions. A review of the use of objective functions in the analysis of orientation data has revealed that, they too, suffer from deficiencies with regard to the use of the Euclidean metric, arbitrary significance thresholds and testing against specific probabilistic distributions.

(iv) A thorough review of the field of photogrammetry has identified modern non-metric analytical photogrammetry, subject to suitable development, as being appropriate for use as an improved technique for the acquisition of rock mass geometrical data. In particular, the use of simple 35mm cameras, coupled with a robust system for introducing photogrammetric control, and making use of an instrument of appropriate specification for measuring photograph coordinates, together with the direct linear transform for the solution of the photogrammetric problem, is indicated.

(v) A detailed study of the problem of classifying data through the application of clustering schemes has revealed that: of the various types available, partitioning methods are the most applicable to rock mass geometrical data; fuzzy partitioning algorithms, in which data may be regarded as being members of a number of clusters simultaneously, offer much improved analysis over hard partitioning algorithms, where only single cluster membership is permitted; the incorporation of the Euclidean metric will produce a tendency to identify isotropic clusters, and this may be overcome through the application of Mahalanobis distances, in which the sample covariance matrix is used as the norm-inducing matrix; the fuzzy *c*-means (FCM), fuzzy covariance (FCV) and fuzzy maximum likelihood (FMLE) algorithms offer the prospect of enhanced analysis of rock mass geometrical data; and a number of cluster

validity functions exist that can be used to identify the optimal partition of a data set.

(vi) An original analysis of the errors arising from the photogrammetric determination of discontinuity orientation has been developed by the Author, and is presented in Chapter 5. This analysis considers the geometry of a typical photogrammetric configuration, and has allowed development of both an empirical predictive equation for orientation error, together with the identification of the appropriate fundamental resolution for the measurement of photograph coordinates. As a result, it has been demonstrated that—for a typical configuration of photogrammetry at a rock face—it is possible to determine discontinuity orientation with an error of less than 1°, providing the photograph coordinates of discontinuity surfaces with an effective diameter of 0.5m or greater are measured with a resolution of 25μm or less.

(vii) A novel, high-performance, low-cost analytical plotter has been designed and developed by the Author. This instrument implements sophisticated closed-loop control of photograph position through the application of specialised integrated circuits. The instrument uses commercially available mechanical components, together with custom designed and built electronic circuits, to allow a traditional mirror stereoscope to operate as an analytical plotter in conjunction with a microcomputer. The closed-loop controllers, together with the ancillary interface electronics, have been designed to reside on a single board that installs into an IBM-PC microcomputer. This work has demonstrated the feasibility of constructing high-performance measuring instruments through the use of modern control techniques.

(viii) A novel system of implementing photogrammetric control at a rock face has been developed by the Author. This takes the form of steel cables, each carrying a number of control markers, tensioned in front of a rock face. A straightforward theodolite survey, coupled with a sophisticated least squares adjustment (developed as a suite of spreadsheet analyses by the Author), allows high-precision control points to be installed at a rock face safely and with ease by non-specialist personnel.

(ix) An enhanced form of the direct linear transform for the solution of the photogrammetric problem has been developed by the Author. This incorporates a singular value decomposition for the solution of the least squares problem, and operates in real-time to allow control of the

analytical plotter—through continuous removal of y-parallax from the stereomodel—to take place.

(x) An accuracy test of the complete photogrammetric system has revealed that: only the first lens distortion term should be incorporated into the DLT solution; for the photogrammetric configuration used at a rock face, the solution is insensitive to the variances attached to the object space coordinates of the control points, and their corresponding photograph coordinates; and that root mean square position errors of 25mm are achievable.

(xi) A study of the geometry of orientation data has revealed that the analysis of such data should take place in Riemannian space, but the lack of suitable distance metrics for the assessment of anisotropic distances means that analysis must take place in Euclidean space. Accordingly, the Author has produced FADD, an original computer program that implements the three fuzzy partitioning algorithms of FCM, FCV and FMLE in a sequential manner, and hence provides for an improved method of identifying the optimal partition in orientation data. This program has been designed to correctly account for both the Riemannian nature of orientation data and the existence of antipodal vectors. Additionally, a number of cluster validity functions have been redefined to allow incorporation of a sharpened partition, in order that their discriminatory behaviour may be enhanced.

(xii) A series of exhaustive tests, with both synthetic and natural data sets, has demonstrated the efficacy of FADD to identify the optimal partition in orientation data. These tests have indicated that a fuzzy exponent of m=2 and a sharpening threshold of α=0.9 offer the best operating conditions for FADD. On the basis of tests with the copper porphyry data of Mahtab *et alia* [86], six cluster validity functions have been ranked in order of discriminatory performance for orientation data, as: partition density, separation index, normalised partition coefficient, modified partition coefficient, proportion exponent and partition coefficient.

(xiii) Two novel types of hemispherical projection have been developed by the Author to allow assimilation of the results of FADD. The first of these—cluster membership plots—allows the overall form of the partition to be easily pictured, and the second of these—membership level plots—shows, with striking clarity, the specific form of each cluster, in terms of the distribution of levels of certainty within each cluster. A new

definition of 'discontinuity set' is prompted by these results; one that is based on level of certainty rather than distribution of orientations.

(xiv) The application of the new, integrated techniques for the measurement and analysis of rock mass geometry at two quarry sites within the UK has been reported on. These have revealed the general efficacy of the Author's systems for the introduction of photogrammetric control, the Author's design of analytical plotter, and the Author's implementation of fuzzy partitioning algorithms for the analysis of orientation data. A comparison with customary discontinuity orientation surveys conducted with a geological compass has revealed the improvement offered by photogrammetry in terms of being able to survey the entire rock face. However, the existence of sampling bias has been noted in both techniques, and further work has been recommended in this direction. The use of FADD to analyse the orientation data resulting from the photogrammetric and compass surveys has shown how the question of determining the number of sets present is eased through the application of cluster validity functions.

(xv) Further development of non-metric analytical photogrammetry has been recommended. In particular, the following topics have been identified as being worthy of investigation: optimisation of the DLT through the use of extended formulations; a theoretical study of the geometry of the photogrammetric problem in terms of the configuration present at a rock face; the application of total least squares to the photogrammetric solution; determination of the absolute performance of the analytical plotter, with the aim of optimising its positioning capabilities; automation of the coordination process through the application of video cameras and stereomatching algorithms; the use of digital photogrammetry; and the development of further simplified methods of introducing photogrammetric control.

(xvi) Further development of fuzzy partitioning algorithms is indicated, particularly with respect to the following: the use of Mahalanobis angles to maintain the Riemannian nature of discontinuity data; identification or derivation of cluster validity functions with improved discriminatory powers; the use of other, more specialised, fuzzy partitioning algorithms; the application of sophisticated interpolation algorithms to the partitioning results; the use of closed-forms for the characterisation of

clusters; analysis of data with more dimensions than three; and the application of non-metric methods.

(xvii) Most importantly, the integrated measurement and analysis system developed by the Author will allow advanced study of rock mass geometry. In particular, work in the following fields is prompted: development of methods for the measurement and characterisation of discontinuity spacing, extent and roughness from photographs; investigation into the interrelations between discontinuity extent, orientation, and spacing as a means of characterising rock mass geometry; the synthesis of scanline techniques as a means of studying sampling bias; development of new techniques that will allow visualisation of complete rock mass geometry; and enhanced stability analyses that incorporate the concept of uncertainty in discontinuity orientation.

Finally, it is prudent to bear in mind that other, quite different, techniques may offer yet greater improvements than the integration of non-metric photogrammetry and fuzzy partitioning algorithms. As the philosopher C.S. Peirce (1839-1914) succinctly wrote:

> *Nothing is <u>vital</u> for science; nothing can be. Its accepted propositions, therefore, are but opinions at most; and the whole list is provisional. The scientific man is not in the least wedded to his conclusions. He risks nothing upon them. He stands ready to abandon one or all as soon as experience opposes them.*

Part V
Supporting Material

This page intentionally left blank

10 References

1. Abdel-Aziz, Y.I. and H.M. Karara, 1971. Direct linear transformation from comparator coordinates into object space coordinates in close range photogrammetry. Proc. Symp. Close-Range Photogrammetry, Urbana, Illinois, pp1-18.
2. Acarnley, P.P., 1982. Stepping motors. London: IEE.
3. Adams, L.P., 1981. The use of non-metric cameras in short range photogrammetry. Photogrammetria, 36, pp51-60.
4. Aimone, C.T. and C.H. Dowding, 1983. Fragmentation measurement results for 14 full-scale production blasts: a comparison with a three-dimensional wave code. Proc. 9th Conf. Exp. and Blast. Tech., pp310-333.
5. Allam, M.M., 1978. The estimation of fractures and slope stability of rock faces using analytical photogrammetry. Photogrammetria, 34, pp89-99.
6. Alldredge, J.R. and M.A. Mahtab, 1974. statistical analysis of axial data. Jour. Geol., 82, pp519–524.
7. American Society of Photogrammetry, 1980. Manual of photogrammetry (4th edition). Falls Church, Virginia: American Society of Photogrammetry.
8. Atkinson, K.B. (ed.), 1980. Developments in close range photogrammetry—I. Barking, Essex: Applied Science Publishers Ltd.
9. Attewell, P.B. and I.W. Farmer, 1976. Principles of engineering geology. London: Chapman and Hall.
10. Aydin, A. and J.M. deGraff, 1988. Evolution of polygonal fracture patterns in lava flows. Science, 239, pp471-476.
11. Bandis, S.C., 1993. Rock mass and site characterization. In: Comprehensive rock engineering: principles, practice and projects, J.A. Hudson (ed.). Oxford: Pergamon Press.
12. Bannister, A. and S. Raymond, 1984. Surveying (5th edition). Harlow, Essex: Longman.

13. Barton, C.C. and E. Larsen, 1985. Fractal geometry of two-dimensional fracture networks at Yucca Mountain, Southwestern Nevada. Proc. Int. Symp. Fundamentals of Rock Joints, Bjorkliden, pp77-84.
14. Bayer, U., 1985. Pattern recognition problems in geology and paleontology. Berlin: Springer Verlag.
15. Bezdek, J.C., 1981. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press.
16. Biernson, G., 1988. Principles of feedback control, volume 2: advanced control topics. New York: John Wiley.
17. Blashfield, R.K., M.S. Aldenderfer and L.C. Morey, 1982. Cluster analysis software. In: Handbook of statistics, volume 2. P.R. Krishnaiah and L.N. Kanal (eds.). Amsterdam: North-Holland Publishing Company.
18. Blostein, S.D. and T.S. Huang, 1987. Error analysis in stereo determination of 3-D point positions. IEEE Trans. Pattern Anal. Machine Intell., 9, 6, pp752-765.
19. Bonissone, P.P., 1982. A fuzzy sets based linguistic approach: theory and applications. In: Approximate reasoning in decision analysis, M.M. Gupta and E. Sanchez (eds.). Amsterdam: North-Holland Publishing Company.
20. Bonyun, D. and G. Stevens, 1970. A general purpose computer program to produce geological stereo net diagrams. In: Data processing in biology and geology, Cutbill, J.L. (ed.). London: Academic Press.
21. Brandow, V.D., H.M. Karara, H.H. Damberger and H.H. Krausse, 1976. A non-metric close range photogrammetric system for mapping geologic structures in mines. Photogramm. Eng. and Rem. Sens., 42, 5, pp637-648.
22. Brown, E.T. (ed.), 1981. Rock characterization testing and monitoring—ISRM suggested methods. Oxford: Pergamon Press.
23. Carson, W.W., 1985. An accuracy test of a new stereoscopic instrument. Kart og Plan, 2, pp227-233.
24. Carson, W.W., 1985. Development of an inexpensive image space plotter. Photogrammetric Record, 65, 11, pp525-541.
25. Carson, W.W., 1987. Development of an inexpensive analytical plotter. Photogrammetric Record, 69, 12, pp303-306.
26. Carter, J.W., 1977. Analysis of a simple photographic method proposed for determining size distribution of ore fragments. USBM laboratory report RBM 77-03. Washington: USBM.

27. Charlesworth, H., D. Cruden, J. Ramsden, and Q. Huang, 1989. ORIENT: an interactive FORTRAN 77 program for processing orientations on a microcomputer. Computers and Geosciences, 15, 3, pp275–293.
28. Chaudhuri, B.B. and D.D. Majunder, 1982. On membership in fuzzy sets. In: Approximate reasoning in decision analysis, M.M. Gupta and E. Sanchez (eds.). Amsterdam: North-Holland Publishing Company.
29. Cheeney, R.F., 1983. Statistical methods in geology for field and lab decisions. George Allen & Unwin (Publishers) Ltd., Hemel Hempstead: George Allen & Unwin (Publishers) Ltd.
30. Cooper, M.A.R., 1987. Control surveys in civil engineering. London: Collins.
31. Dave, R.N., 1990. Fuzzy shell-clustering and applications to circle detection in digital images. Int. J. General Systems, 16, pp343-355.
32. Dave, R.N. and K. Bhaswan, 1992. Adaptive fuzzy *c*-shells clustering and detection of ellipses. IEEE Trans. Neural Networks, 3, 5, pp643-662.
33. Demantaras, R.L. and L. Valverde, 1988. New results in fuzzy clustering based on the concept of indistinguishability relation. IEEE Trans. Pattern Anal. Machine Intell., 10, 5, pp754-757.
34. Dershowitz, W.S. and H.H. Einstein, 1988. Characterizing rock joint geometry with joint system models. Rock. Mech. and Rock Eng., 21, pp21-51.
35. DeRycker, K., 1984. A routine method to evaluate the modes of failure and the stability of rock slopes. Unpublished MSc dissertation, Imperial College, University of London.
36. Dierckx, P., 1984. Algorithms for smoothing data on the sphere with tensor product splines. Computing, 32, pp319-342.
37. Dunn, G. and B.S. Everitt, 1982. An introduction to mathematical taxonomy. Cambridge: Cambridge University Press.
38. Engelder, T., 1987. Joints and shear fractures in rock. In: Fracture mechanics of rock, B.K. Atkinson (ed.). London: Academic Press.
39. Everitt, B., 1974. Cluster analysis. London: Heinemann Educational Books Ltd.
40. Faig, W., 1971. Calibration of close-range photogrammetric cameras. Proc. Symp. Close-Range Photogrammetry, Urbana, Illinois, pp111-131.
41. Faig, W., 1975. Photogrammetric equipment systems with non-metric cameras. Proc. Symp. Close-Range Photogrammetric Systems, Champaign, Illinois, pp648-657.

42. Fairhurst, C. and D.H. Lin, 1985. Fuzzy methodology in tunnel support design. In: Research and Engineering Applications in Rock Masses, Proc. 26th US Symp. Rock Mech., E. Ashworth (ed.). Rotterdam: Balkema.
43. Fecker, E. and N. Rengers, 1971. Measurement of large scale roughness of rock planes by means of profilograph and geological compass. I-18, Proc. Symp. Rock Fracture, Nancy.
44. Fisher, N.I., T. Lewis and B.J.J. Embleton, 1987. Statistical analysis of spherical data. Cambridge: Cambridge University Press.
45. Franklin, J.A. and N.H. Maerz, 1986. Digital photo-analysis of rock jointing. Proc. 39th Canadian Geotechnical Conference, pp11-20.
46. Gath, I. and A.B. Geva, 1989. Unsupervised optimal fuzzy clustering. IEEE Trans. Pattern Anal. Machine Intell., 11, 7, pp773-781.
47. Gitman, I. and M.D. Levine, 1970. An algorithm for detecting unimodal fuzzy sets and its application as a clustering technique. IEEE Trans. Comp., C-19, 7, pp583-593.
48. Gleick, J., 1987. Chaos: making a new science. London: Sphere Books Ltd.
49. Golub, G.H. and C.F. van Loan, 1991. Matrix computations (2nd edition). Baltimore: Johns Hopkins University Press.
50. Goodfellow, R.J.F., 1992. The development of objective techniques for the measurment of discontinuity roughness. Unpublished MSc dissertation, Imperial College, University of London.
51. Gozon, J.S., R.R. Britton and J.D. Fado, 1986. Predetermining average fragment size: a case study. Int. Symp. App. Rock Char. Tech. in Mine Design, pp190-195.
52. Grossman, N.F., 1985. The bivariate normal distribution on the tangent plane at the mean attitude. Proc. Int. Symp. Fundamentals of Rock Joints, Björkliden, Sweden.
53. Gruen, A.W., 1985. Data processing methods for amateur photographs. Photogrammetric Record, 65, 11, pp567-579.
54. Gustafson, K.E. and W. Kessel, 1979. Fuzzy clustering with a fuzzy covariance matrix. In: Proc. IEEE-CDC, 2, K.S. Fu (ed.). Piscataway, New Jersey: IEEE Press.
55. Hagan, T.O., 1980. A case for terrestrial photogrammetry in deep mine rock structure studies. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 17, 4, pp191-198.

56. Harrison, J.P. and R.J.F. Goodfellow, 1993. Effect of scale on roughness anisotropy and magnitude. Proc. 2nd Int. Workshop on Scale Effects in Rock Masses. Lisboa, Portugal.
57. Hatze, H., 1988. High precision 3-dimensional photogrammetric calibration and object space reconstruction using a modified DLT approach. J. Biomechanics, 21, 7, pp533-538.
58. Healy, M.J.R., 1986. Matrices for statistics. Oxford: Oxford University Press.
59. Herget, G., 1977. Pit slope manual chapter 2—structural geology. CANMET, CANMET Report 77-41.
60. Hewlett-Packard, 1985. General purpose motion control IC: HCTL-1000. Manufacturers data sheet.
61. Hoek, E. and M. Diederichs, 1989. DIPS users manual. Department of Civil Engineering, University of Toronto.
62. Hudson, J.A. and J.P. Harrison, 1992. A new approach to studying complete rock engineering problems. Quarterly Journal of Engineering Geology, 25, pp93-105.
63. Hudson, J.A. and S.D. Priest, 1979. Discontinuities and rock mass geometry. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 16, 6, pp339-362.
64. Hudson, J.A. and S.D. Priest, 1983. Discontinuity frequency in rock masses. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 20, 2, pp73-89.
65. Kandel, A., 1982. Fuzzy techniques in pattern recognition. New York: John Wiley and Sons.
66. Karara, H.M., 1971. New trends in close-range photogrammetry. Proc. Symp. Close Range Photogrammetry, Urbana, Illinois, pp231-238.
67. Karara, H.M., 1975. Industrial photogrammetry. Proc. Symp. Close-Range Photogrammetric Systems, Champaign, Illinois, pp97-141.
68. Karara, H.M., 1980. Non-metric cameras. In: Developments in close range photogrammetry—1, K.B. Atkinson (ed.). Barking, Essex: Applied Science Publishers Ltd.
69. Karara, H.M., 1989. Non-topographic photogrammetry (2nd edition). Falls Church, Virginia: American Society of Photogrammetry and Remote Sensing.
70. Karzulovic, A. and R.E. Goodman, 1985. Determination of principal joint frequencies. Int. J. Rock Mech. Min. Sci. & Geomech. Abs., Technical note, 22, 6, pp471-473.

71. Kaufmann, A., 1975. Introduction to the theory of fuzzy subsets. New York: Academic Press.
72. Kaufmann, A. and M.M. Gupta, 1985. Introduction to fuzzy arithmetic. New York: Van Nostrand Reinhold.
73. Kondoh, T. and M. Shinji, 1986. Back analysis of assessing for slope stability based on displacement measurements. Proc. Symp. Engineering in Complex Rock Formations, Beijing, pp809-815.
74. Kreyszig, E., 1968. Introduction to differential geometry and Riemannian geometry. Toronto: University of Toronto Press.
75. Kreyszig, E., 1988. Advanced engineering mathematics, 6th edition. New York: John Wiley & Sons.
76. Kulatilake, P.H.S.W., 1988. State-of-the-art in stochastic joint geometry modelling. In: Key Questions in Rock Mechanics, Proc. 29th. US Symp. Rock Mechs., P.A. Cundall, R.L. Sterling, A.M. Starfield (eds.). Rotterdam: Balkema.
77. Kulatilake, P.H.S.W and D.N. Wathugala, 1992. Joint geometry modelling in 3D and validations—a case study. Proc. Conf. Fractured and Jointed Rock Masses. Lake Tahoe, California.
78. Kuo, B.C., 1991. Automatic control systems (6th edition). New Jersey: Prentice-Hall.
79. Kyoya, T., Y. Ichikawa and T. Kawamoto, 1987. Deformation and fracturing process of discontinuous rock masses and damage mechanics. Proc. 6th Int. Cong. Rock Mechs., Montreal, pp1039-1042.
80. Lancaster, D., 1974. TTL Cookbook. Indianapolis, Indiana: Howard W. Sams.
81. Lenz, R.K. and R.Y. Tsai, 1988. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. IEEE Trans. Pattern Anal. Machine Intell., 10, 5, pp713-720.
82. Libert, G. and M. Roubens, 1982. Non-metric fuzzy clustering algorithms and their cluster validity. In: Approximate reasoning in decision analysis, M.M. Gupta and E. Sanchez (eds.). Amsterdam: North-Holland Publishing Company.
83. Lin, Y.M., 1986. Cluster analysis through computer and its application. Proc. Symp. Engineering in Complex Rock Formations, Beijing, pp82-90.
84. Louis, E. and F. Guinea, 1987. The fractal nature of fracture, Europhysics Letters, 3, 8, pp871-877.

85. Maerz, N.H., J.A. Franklin, L. Rothenburg and D.L. Coursen, 1987. Measurement of rock fragmentation by digital photoanalysis. Proc. 6th Cong. Rock Mechs., Montreal, pp687-692.
86. Mahtab, M.A., D.D. Bolstad, J.R. Alldredge and R.J. Shanley, 1972. Analysis of fracture orientations for input to structural models of discontinuous rock. USBM RI7669. Washington: USBM.
87. Mahtab, A., S. Xu, P. Grasso and F.S. Kendorski, 1992. Use of alternative distributions for characterizing joint extent and spacing. Proc. Conf. Fractured and Jointed Rock Masses. Lake Tahoe, California.
88. Mahtab, M.A. and T.M. Yegulalp, 1982. A rejection criterion for definition of clusters in orientation data. In: Issues in Rock Mechanics, Proc. 23rd Symp. Rock Mech., R.E. Goodman and F.E. Heuze (eds.). New York: Society of Mining Engineers of AIME.
89. Mandelbrot, B.B., 1983. The fractal geometry of nature. New York: W.H. Freeman and Co.
90. Mardia, K.V., 1972. Statistics of directional data. London: Academic Press.
91. Mardia, K.V., 1977. Mahalanobis distances and angles. In: Multivariate analysis—IV, P.R. Krishnaiah (ed.). Amsterdam: North-Holland Publishing Company.
92. Mardia, K.V., J.T. Kent and J.M. Bibby, 1979. Multivariate analysis. London: Academic Press.
93. Marzan, G.T., 1975. Optimum configuration of data acquisition in close-range photogrammetry. Proc. Symp. Close-Range Photogrammetric Systems, Champaign, Illinois.
94. Marzan, G.T. and H.M. Karara, 1975. A computer program for direct linear transformation solution of the colinearity condition, and some applications of it. Proc. Symp. Close-Range Photogrammetric Systems, Champaign, Illinois, pp420-476.
95. Matheson, G.D., 1991. The use of field data in the design of rock slopes. In: Rock Mechanics as a Multidiscilpinary Science, Proc. 32nd US Symp. Rock Mechanics, J-C Roegiers (ed.). Rotterdam: Balkema.
96. Miller, S.M., P. C. McWilliam and J.C. Kerkering, 1989. Evaluation of stereo digitizing for measuring rock fracture roughness. In: Rock Mechanics as a Guide for Efficient Utilization of Natural Resources, Proc. 30th US Symp. Rock Mechanics, A.W. Khair (ed.). Rotterdam: Balkema.

97. Milne, D.M., 1988. Suggestions for standardization of rock mass classification. Unpublished MSc dissertation, Imperial College, University of London.
98. Naftel, A.J. and J.C. Boot, 1991. An iterative linear transformation algorithm for solution of the collinearity equations. Photogrammetric Engineering and Remote Sensing, 57, 7, pp913-919.
99. Nguyen, V.U., 1985. Overall evaluation of geotechnical hazard based on fuzzy set theory. Soils and Foundations, 25, 4, pp8-18.
100. Nguyen, V.U., 1985. Some fuzzy set applications in mining geomechanics. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 22, 6, pp369-379.
101. Nguyen, V.U. and E. Ashworth, 1985. Rock mass classification by fuzzy sets. In: Research and Engineering Applications in Rock Masses, Proc. 26th US Symp. Rock Mech., E. Ashworth (ed.). Rotterdam: Balkema.
102. Noren, C.H. and D.D. Porter, 1974. A comparison of theoretical explosive energy and energy measured underwater with measured rock fragmentation. Proc. 3rd Cong. ISRM, 2B, pp1371-1375.
103. Oimoen, D.C., 1987. Evaluation of a tablet digitizer for analytical photogrammetry. Photogrammetric Eng. and Rem. Sens., 53, 6, pp601-603.
104. Petrie, G., 1990. Analogue, analytical and digital photogrammetric systems applied to aerial mapping. In: Engineering Surveying Technology, T.J.M. Kennie and G. Petrie (eds). Glasgow: Blackie & Sons Ltd.
105. Pollard, D.D. and P. Segall, 1987. Theoretical displacements and stress. In: B.K. Atkinson (ed.), Fracture Mechanics of Rock. London: Academic Press.
106. Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, 1992. Numerical recipes in FORTRAN: the art of scientific computing (2nd edition). Cambridge: Cambridge University Press.
107. Price, N.J., 1966. Fault and joint development in brittle and semi-brittle rock. Oxford: Pergamon Press.
108. Priest, S.D., 1985. Hemispherical projection methods in rock mechanics. London: George Allen and Unwin.
109. Priest, S.D., 1993. Discontinuity analysis for rock engineering. London: Chapman and Hall.
110. Priest, S.D. and J.A. Hudson, 1976. Discontinuity spacings in rock. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 13, pp135-148.

111. Priest, S.D. and J.A. Hudson, 1981. Estimation of discontinuity spacing and trace length using scanline surveys. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 18, pp183-197.
112. Rector, R. and G. Alexy, 1980. The 8086 book. Berkeley, California: Osborne/McGraw-Hill.
113. Renshaw, C.E. and D.D. Pollard, 1992. Numerical generation of physically based fracture networks. Proc. Conf. Fractured and Jointed Rock Masses. Lake Tahoe, California.
114. Rogers, D.F. and J.A. Adams, 1990. Mathematical elements for computer graphics (2nd edition). New York: McGraw-Hill.
115. Ross-Brown, D.M., 1973. Aspects of slope design in open pit mining. PhD Thesis, Imperial College, London.
116. Ross-Brown, D.M., 1973. FORTRAN IV programs for determining joint orientations from terrestrial photogrammetric observations. Imperial College Rock Mechanics Research Report 27.
117. Ross-Brown, D.M., 1973. Suggested method for the determination of joint orientations by photogrammetry. Imperial College Rock Mechanics Research Report 24.
118. Ross-Brown, D.M. and K.B. Atkinson, 1972. Terrestrial photogrammetry in open pits: 1—description and use of the phototheodolite in mine surveying. Trans. IMM, A, 81, pp205-213.
119. Ross-Brown, D.M., E.H. Wickens and J.T. Markland, 1973. Terrestrial photogrammetry in open pits: 2—an aid to geological mapping. Trans. IMM, A, 82, pp115-130.
120. Royer, J.P., 1987. Handbook of software and hardware interfacing for IBM PCs. Englewood Cliffs, New Jersey: Prentice-Hall.
121. Sakurai, S. and N. Shimuzu, 1987. Assessment of rock slope stability by fuzzy set theory. Proc. 6th Int. Cong. Rock Mechs, Montreal, pp503-506.
122. Santaló, L.A., 1976. Integral geometry and geometric probability. Reading, Massachusetts: Addison-Wesley.
123. SGS, 1987. Motion control application manual. Manufacturers data sheets.
124. Shanley, R.J. and M.A. Mahtab, 1974. A computer program for clustering data points on the sphere. USBM IC8624. Washington: USBM.
125. Shanley, R.J. and M.A. Mahtab, 1975. FRACTAN: A computer code for analysis of clusters defined on the unit hemisphere. USBM IC8671. Washington: USBM.

126. Shanley, R.J. and M.A. Mahtab, 1976. Delineation and analysis of clusters in orientation data. Math. Geol., 8, 1, pp9–23.
127. Shimuzu, N. and S. Sakurai, 1986. A study on rock mass classification by fuzzy set theory. Proc. JSCE No. 370/III-5, pp225-232.
128. Skjeltorp, A.T. and P. Meakin, 1988. Fracture in microsphere monolayers studied by experiment and computer simulation. Nature, 335, pp424-426.
129. Solla, S.A., 1986. Crack Propagation and Onset of Failure. In: On growth and Form—Fractal and Non-Fractal Patterns in Physics, H.E. Stanley and N. Ostrowsky (eds.). Dordrecht: Martinus Nijhoff Publishers.
130. Späth, H., 1980. Cluster analysis algorithms for data reduction and classification of objects. Chichester: Ellis Horwood Ltd.
131. Sullivan, T.D., 1980. Simple terrestrial photogrammetry. Unpublished MSc dissertation, Imperial College, University of London.
132. Texas Instruments, 1989. The TTL Data Book. Dallas: Texas Instruments.
133. Thomas, A., P. Therme and P. Richard, 1981. A new method for quick measuring and computing parameters on rock jointing. In: Rock mechanics from research to application, Proc. 22nd US Symp. Rock Mechanics, H.H. Einstein (ed.). Cambridge, Massachusetts: Massachusetts Institute of Technology.
134. Turcotte, D.L., 1986. Fractals and fragmentation. J. Geophys. Res., 91, B2, pp1921-1926.
135. Upton, G.J.G. and B. Fingleton, 1989. Spatial data analysis by example—volume 2: categorical and directional data. Chichester: John Wiley & Sons Ltd.
136. vanEvergingen, D.A., J.A.M. vanGool and R.M. Vissers, 1992. QUICKPLOT: a microcomputer-based program for processing of orientation data. Computers and Geosciences, 18, 2/3, pp. 183–287.
137. vanHuffel, S and J. Vandewalle, 1991. The total least squares problem: computational aspects and analysis. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics.
138. Wahba, G., 1981. Spline interpolation and smoothing on the spere. SIAM Journal on Sci. and Stat. Comp., 2, pp.5-16: erratum; 3, pp.385-386.
139. Warburton, P.M., 1980. A stereological interpretation of joint trace data. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 17, 4, pp181-190.
140. Warburton, P.M., 1980. Stereological interpretation of joint trace data: influence of joint shape and implications for geological surveys. Int. J. Rock Mech. Min. Sci. and Geomech. Abs., 17, 6, pp305-316.

141. Weatherburn, C.E., 1938. An introduction to Riemannian geometry and the tensor calculus. Cambridge: Cambridge University Press.
142. Windham, M.P., 1980. Cluster validity for fuzzy clustering algorithms. J. Fuzzy Sets Syst.
143. Wolf, P.R., 1974. Elements of photogrammetry. New York: McGraw-Hill.
144. Zimmermann, H.-J., 1991. Fuzzy set theory and its applications (2nd edition). Dordrecht: Kluwer Academic Publishers.

This page intentionally left blank

Appendix A Analytical Plotter Construction Details

A high-performance, low-cost analytical plotter, operating as an enhancement to a standard mirror stereoscope, has been designed and constructed as part of this work. This instrument, specifically intended for use in photogrammetric procedures associated with the analysis of rock mass geometry, differs in a number of respects from commercially available instruments.

This Appendix primarily contains detailed descriptions of, and associated discussion regarding, all of the elements—both electrical and mechanical—designed for use in the instrument. In order to elucidate a number of aspects associated with the design of the instrument, brief details are presented on the commercially available components used; this includes a functional description of the main control microprocessors. There are eight Sections within this Appendix, and they concern: the general configuration of the analytical plotter; a description of the closed-loop control system; a description of the HCTL-1000 general purpose motion control IC; the design and construction of the ISA-bus interface circuit; the design and construction of the stepping motor driver circuit; a description of miscellaneous electrical issues; programming of the HCTL-1000 controllers; and mechanical aspects.

A.1 General

Figure 3.08(c), in the main text, shows a schematic of an analytical plotter. In the context of the instrument constructed here, an analytical plotter can be thought of as consisting of two major functional elements: a mechanical system that maintains—in their correct overall relative positions—the mirror stereoscope and the two photographs constituting a stereopair, whilst permitting accurate controlled independent movement of the photographs during the coordinate measurement process; and an electrical system comprising a host computer, dedicated microprocessor closed-loop motion controllers and motor drivers for the photograph movement system.

The left and right photographs of the stereopair are each mounted on an assembly of linear bearings, to allow independent movement along two essentially orthogonal axes. Linear motion is provided by leadscrews, fitted with backlash-free nuts and turned by stepping motors. The four axes of motion in the instrument have separate closed-loop control systems, each of which comprises the three main functional units of control microprocessor, motor driver and stepping motor/encoder assembly. The four control microprocessors, together with the necessary interface circuitry, are mounted on a single ISA-bus (i.e. Industry Standard Architecture bus—around which all IBM-compatible personal computers are built) compatible wirewrap board inside the casing of the host computer, which executes the analysis system software. The four motor drivers are mounted on a custom made printed circuit board and, together with a power supply unit and miscellaneous other circuitry, are housed in a separate casing. The user operates the instrument via a joystick, equipped with push-buttons, which is connected to the host computer.

A.2 Closed-Loop Control System

For each of the four axes of motion there is a closed-loop control system, consisting of the elements shown in Figure A.01.

Briefly, these elements are:

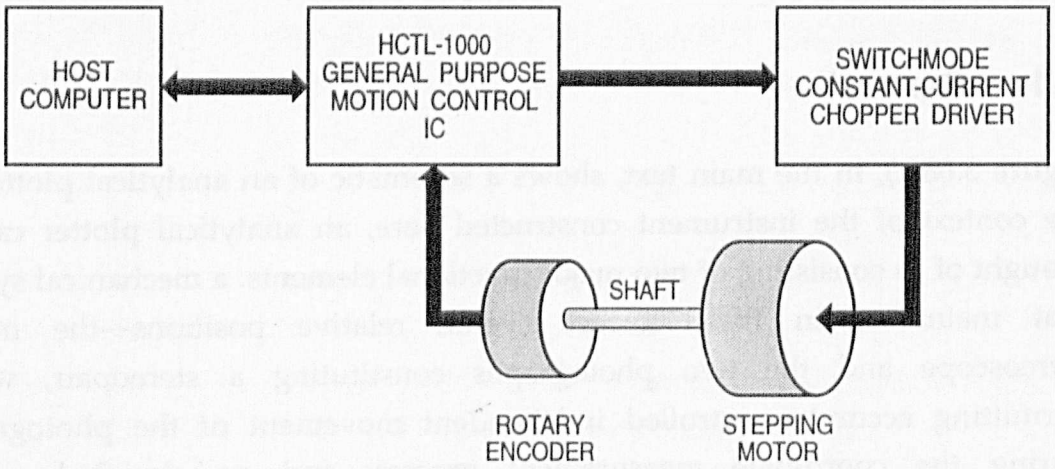


Figure A.01 Block diagram of closed-loop control system.

(i) Host Computer

An Intel 8088-based ISA-bus machine, with software selectable processor clock speeds of 4.77MHz and 8MHz.

A control and analysis program executing on this machine communicates, via assembly language subroutines and interface circuitry, with the HCTL-1000 controller ICs. Following programming to configure the ICs, the only communication necessary is output of either command positions or command velocities and input of actual position.

(ii) HCTL-1000 General Purpose Motion Control IC

A high performance integrated circuit, manufactured by Hewlett-Packard, which performs the time-intensive computation associated with digital closed-loop control.

This device continually compares feedback from the rotary encoder, in terms of actual values of either position or velocity, with the associated command values (velocity is computed from timed position measurements). The resulting output signal to the motor driver consists of motor commutation information (i.e. sign of rotation direction) and energising information (i.e. magnitude).

(iii) Switchmode Constant-Current Chopper Driver

A custom circuit based on two integrated circuits—a L6506 current controller and a L293E push-pull driver—both manufactured by SGS-Thomson.

This circuit performs as a voltage-to-current amplifier, converting the logic signals from the HCTL-1000 into the high currents required for motor operation.

(iv) Stepping Motor/Rotary Encoder Assembly

A size 17, 1.8° step angle, 0.6A/phase, hybrid stepping motor, manufactured by Sanyo-Denki, together with a HEDS5540 three-channel, 500pulse/revolution rotary optical encoder, manufactured by Hewlett-Packard.

These elements are connected by a leadscrew, which is used to convert the rotary motion of the motor into the linear motion required to move the photographs.

Three logic signals are output by the encoder: two trains of count pulses, electrically 90° out of phase (i.e. in quadrature), and an index pulse generated once per mechanical revolution.

A.3 HCTL-1000 General Purpose Motion Control IC

The HCTL-1000 is a general purpose motion controller, available as a 40-pin dual in-line integrated circuit, designed to provide position and velocity control for DC, DC brushless and stepping motors. This Section, describing the operation of, and interfacing to, the device contains condensed and simplified information, compiled from the manufacturer's technical data sheet [60].

The device receives commands from a host processor via an 8-bit bi-directional multiplexed address/data bus, and position feedback is taken from an incremental encoder with quadrature output. The feedback is internally decoded into quadrature counts (thus effectively increasing the resolution of the encoder by a factor of 4), and maintained in a 24-bit counter to keep track of position. Output to the motor is via either a programmable electronic commutator in conjunction with a pulse width modulated (PWM) signal, or an 8-bit control byte, each bit being available on an output pin.

Closed-loop control is maintained through the use of a programmable digital filter, sampled on a discrete sample time basis. Although the internal functions are synchronised to an external clock, the interface operates asynchronously with respect to the closed-loop control functions.

A.3.1 Internal operation

A simplified internal block diagram of the HCTL-1000 is illustrated in Figure A.02, in which only those functional elements relevant to this work are indicated. This diagram shows: at the left, the inputs generated by, and outputs delivered to, the host computer via the ISA-bus interface; at the top and bottom, the inputs from the analytical plotter and rotary encoder, and; at the right, the motor driver control signals.

Associated with most of these functional elements are a number of 8-bit registers, as shown in Figure A.03, which are accessed by the user for configuring the device, issuing movement commands and reading the current state of the device and the motor/encoder assembly.

The HCTL-1000 operates in one of four control modes. These are: position control; proportional velocity control; trapezoidal profile control for point to

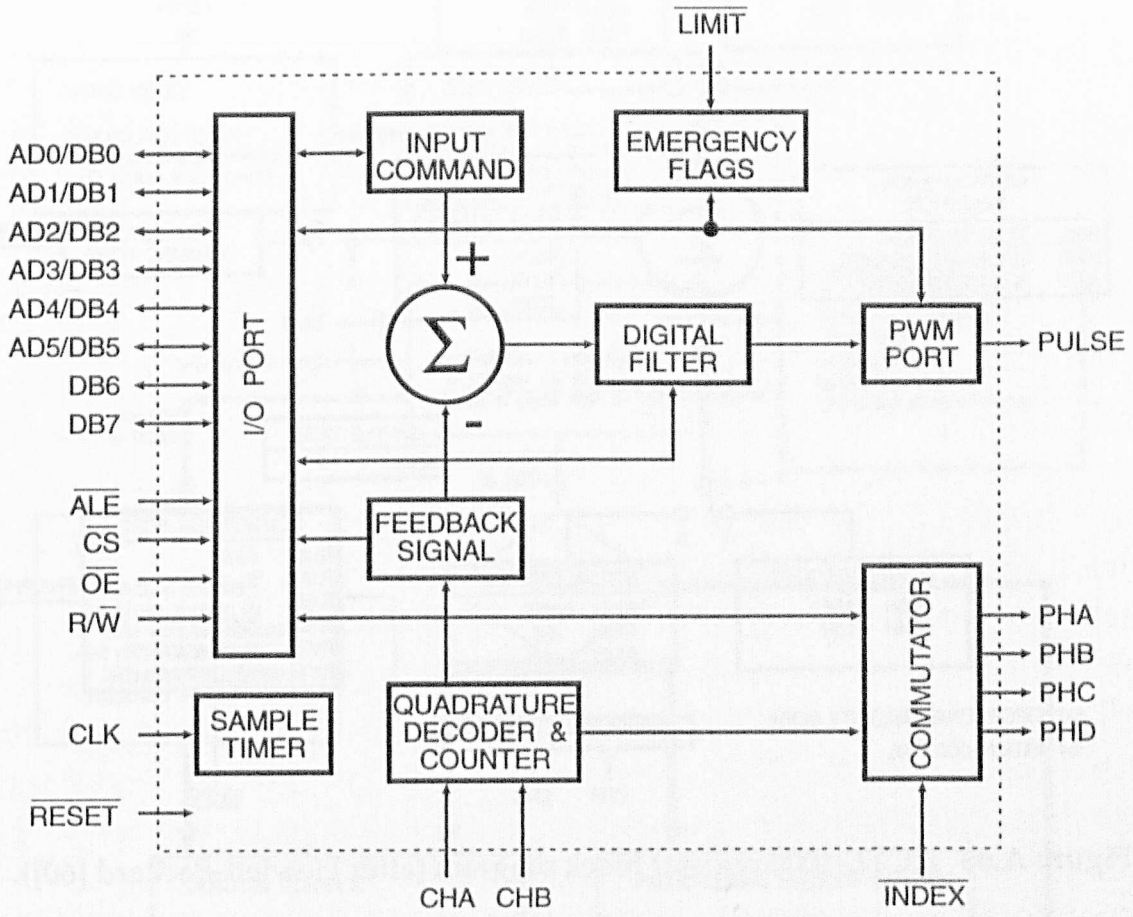


Figure A.02 HCTL-1000 internal block diagram (after Hewlett-Packard [60]).

point moves; and integral velocity control with continuous velocity profiling using linear acceleration. Only the first two of these modes are used in this work. The operation flowchart of the HCTL-1000, shown in Figure A.04, indicates the various routines the device executes, the input required from the user during these routines, and how the device switches from one routine to another.

In all control modes, closed-loop control is performed by the HCTL-1000 through the use of a digital filter, which takes the form

$$D(z) = \frac{K\left(z - \frac{A}{256}\right)}{4\left(z + \frac{B}{256}\right)} \tag{A.01}$$

where $D(z)$=digital domain operator, K=digital filter gain, A=digital filter zero and B=digital filter pole. The device determines the compensation necessary to

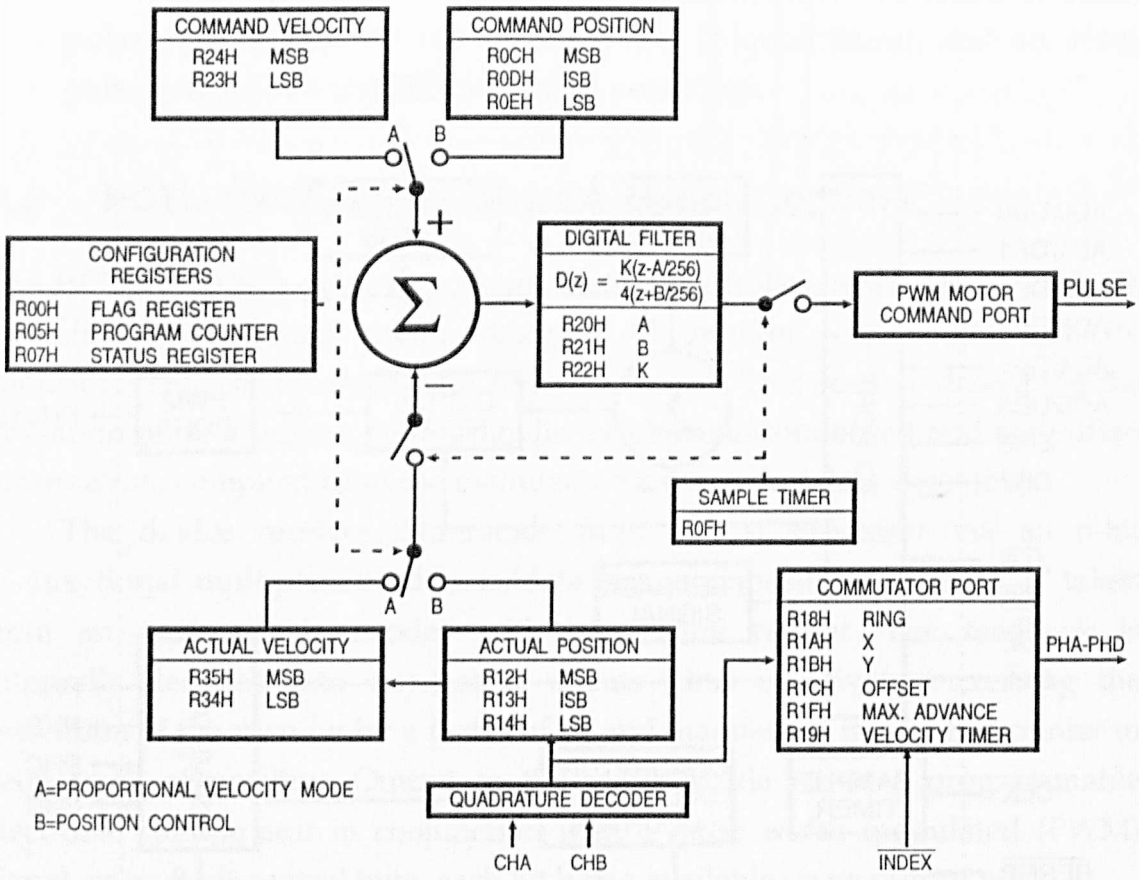


Figure A.03 HCTL-1000 register block diagram (after Hewlett-Packard [60]).

maintain system stability on the basis of samples of the feedback signal taken at a time interval programmed by the user. In position and proportional velocity control modes the filter is implemented as described below.

(i) Position Control

The user specifies a 24-bit scalar word for the position command, which the controller compares to the 24-bit actual position. The difference between these two values is the position error, ΔP, and this is then used in the digital filter to determine the motor command as

$$MC_n = \left(\frac{K}{4}\right)\Delta P_n - \left[\left(\frac{A}{256}\right)\left(\frac{K}{4}\right)\Delta P_{n-1} + \left(\frac{B}{256}\right)MC_{n-1}\right] \qquad \text{(A.02)}$$

where n=current sample time, n-1=previous sample time, and MC=motor command.

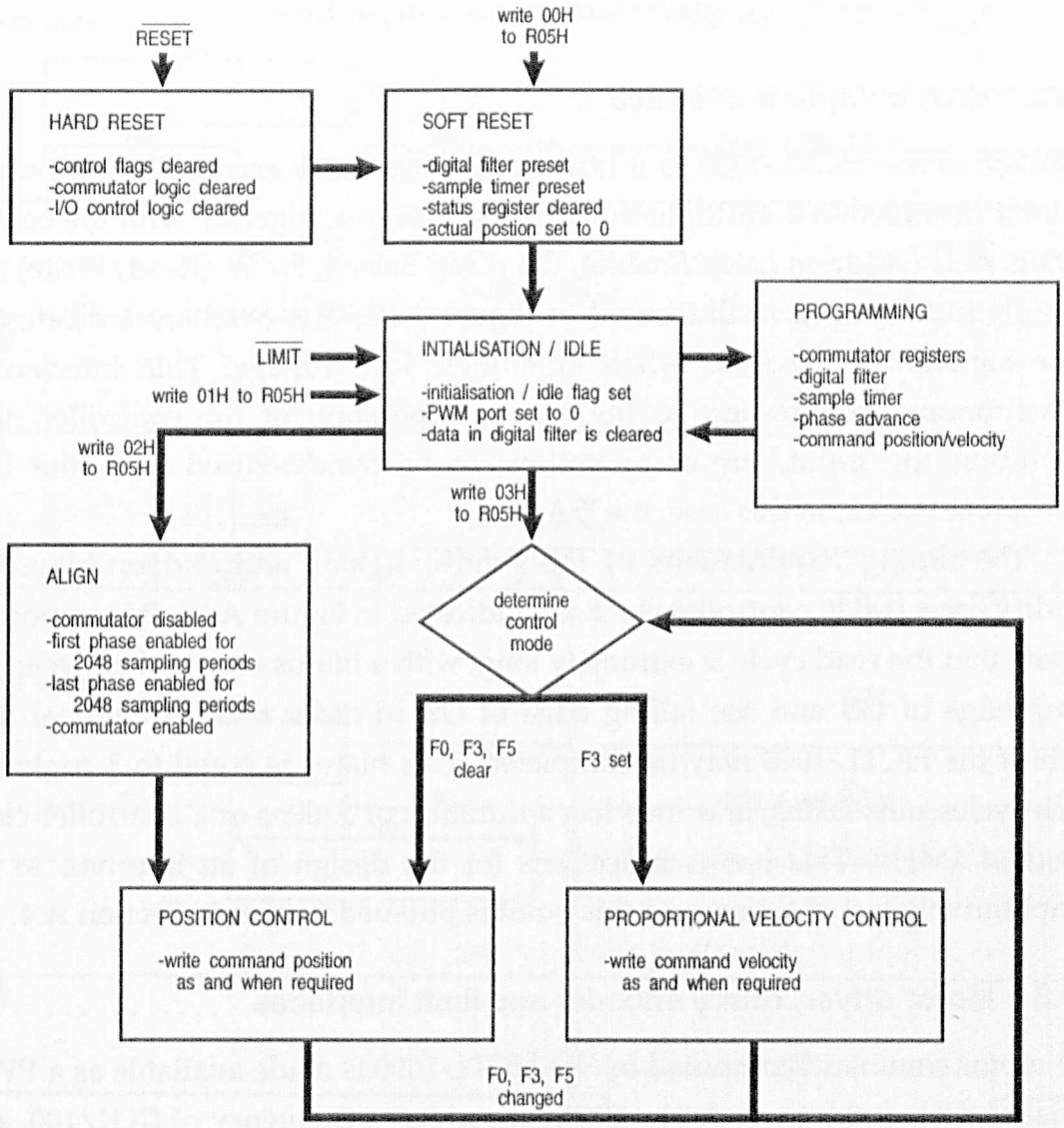


Figure A.04 HCTL-1000 operation flowchart (after Hewlett-Packard [60]).

(ii) Proportional Velocity Control

The user specifies a 16-bit two's complement word for the velocity command, which the controller compares to the 16-bit actual velocity. The difference between these two values is the velocity error, ΔV, and this is then used in the digital filter to determine the motor command:

$$MC_n = \left(\frac{K}{4}\right)\Delta V_n \qquad \text{(A.03)}$$

The controller computes actual velocity from actual position data. The units of velocity are quadrature counts/sample time.

A.3.2 Host computer interface

Interface of the HCTL-1000 to a host microprocessor is accomplished through an 8-bit bi-directional multiplexed address/data bus, together with the control signals $\overline{\text{ALE}}$ (*A*ddress *L*atch *E*nable), $\overline{\text{CS}}$ (*C*hip *S*elect), R/$\overline{\text{W}}$ (*R*ead/*W*rite) and $\overline{\text{OE}}$ (*O*utput *E*nable), as illustrated in Figure A.02. The overbars indicate that these signals are asserted when at a logic low voltage. This interface is asynchronous with respect to the internal operation of the controller, thus permitting the input/output operations to be synchronised with the host microprocessor or, in this case, the ISA-bus.

The timing requirements of the control signals and address/data bus validity for a 1MHz controller clock are indicated in Figure A.05. It is important to note that the read cycle is extremely long, with a hiatus required between the rising edge of $\overline{\text{CS}}$ and the falling edge of $\overline{\text{OE}}$ in order that the internal data fetch of the HCTL-1000 may be completed. This hiatus is equal to 3 controller clock cycles plus 200ns, and thus has a duration of 3200ns at a controller clock speed of 1MHz. This has ramifications for the design of an interface to the comparatively fast ISA-bus, and this point is pursued further in Section A.4.

A.3.3 Motor driver, rotary encoder and limit interfaces

The motor command computed by the HCTL-1000 is made available as a PWM signal and as an 8-bit word. The PWM signal has a frequency of CLK/100, and a variable duty cycle that is resolved into each of the CLK cycles. Thus, the PWM signal can take on all integer values from 0% up to and including 100%.

The commutator port may be programmed by the user to generate logic-level signals for the control of a stepping motor power driver. The commutator port works as a walking-ring counter [80], using quadrature encoder counts as an input clock, and generates the switching sequence on the basis of three programmable parameters—RING, X and Y—together with configuration information on the number of motor phases. RING represents the length of a complete electrical cycle, X the duration a sole motor phase is energised, and Y the duration two adjacent phases are energised. To avoid the difficulty associated with mechanically aligning a high resolution encoder to a motor to an accuracy of better than one encoder quadrature count (e.g. 10.8 minutes of arc with a 500 counts/revolution encoder), electronic fine-tuning is

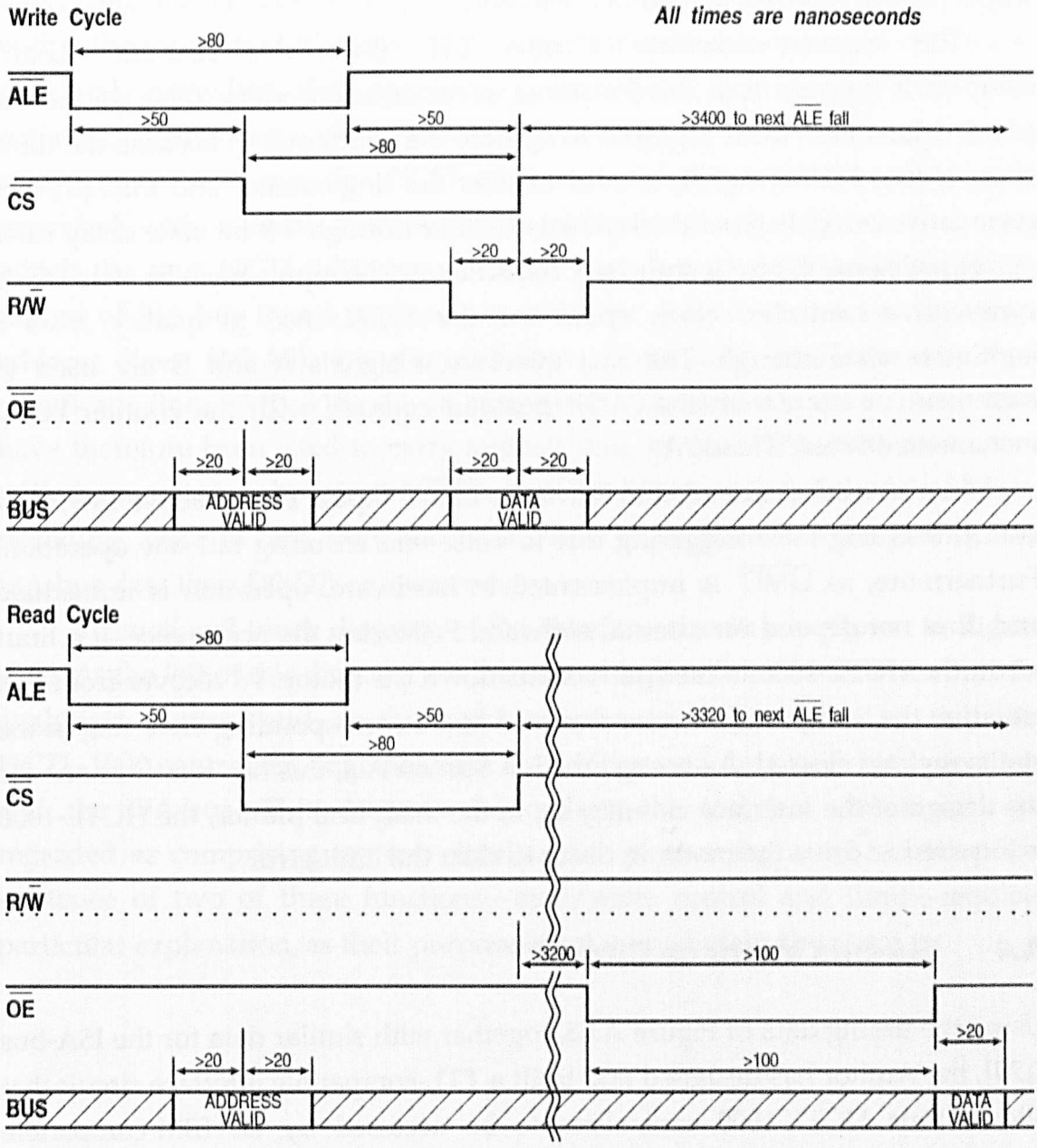


Figure A.05 HCTL-1000 interface timing diagrams (after Hewlett-Packard [60]).

provided through the OFFSET register. This register takes integer data, representing quadrature counts, and adds it to the commutator output, effectively rotating the commutation relative to the motor.

A digital phase advance is implemented in the HCTL-1000, whereby energising of motor phases can be linearly advanced according to the speed of the motor. This enhances high speed operation of a motor, by accounting for the time lag between energising a winding and development of the full

magnetomotive force, thus ensuring a winding is operating optimally with respect to the magnetic circuit in the motor.

The encoder interface accepts TTL (Transistor-Transistor Logic) compatible outputs from 2 or 3 channel incremental encoders, although the use of a 3 channel device is required to operate the commutator, because the third channel (the INDEX signal) is used to reset the ring counter and thus prevent cumulative error. Each of the inputs is filtered through a 3-bit state delay filter to remove noise. Consequently, any transition on the inputs must be stable for 3 consecutive controller clock cycles for the HCTL-1000 to qualify it as a legitimate state change. The two quadrature signals A and B are used to increment or decrement the 24-bit position counter, with the counter being incremented when B leads A.

Limit switches are used to drive the $\overline{\text{LIMIT}}$ input. This is active low, thus both preventing false triggering due to noise and ensuring fail-safe operation. Furthermore, as $\overline{\text{LIMIT}}$ is implemented in hardware, operation is immediate and does not depend on external software. Following the occurrence of a limit error, the HCTL-1000 immediately shuts down the motor. To recover from this situation the limit error must be removed and a corresponding error flag in the status register cleared. As is explained in Section A.4, this has ramifications for the design of the interface circuitry as, in the analytical plotter, the HCTL-1000 is required to drive the motor in order to clear the limit error.

A.4 ISA-bus Interface Circuit

Using the timing data of Figure A.05, together with similar data for the ISA-bus [120], the Author has designed and built a TTL-compatible interface circuit that allows four HCTL-1000 controllers to be accessed by an IBM-compatible microcomputer. Uniquely, this four-channel circuit is contained on a single ISA-bus extension card and hence fits in a single expansion slot of a microcomputer. Consequently, communication with the analytical plotter is simply effected via a single 36-way cable. The interface has been designed to logically reside in the input/output space of the computer and is thus accessed by the host microprocessor's OUT(*port_address, data*) and IN(*data, port_address*) commands [112] for write and read functions, respectively. Computation of *port_address* is performed by an assembly language routine, the code for which is presented in Appendix B.

The circuit has been designed to perform all the functions essential for interfacing the HCTL-1000s to the ISA-bus. These are: uniquely identifying, within the computer I/O space, every register in each device (a total of 256 8-bit registers); providing the capability of writing to and reading from these registers; generating HCTL-1000 control signals; and ensuring timing compatibility between the HCTL-1000 and the ISA-bus. The interface is provided with a local, bi-directional, multiplexed address/data bus, onto which the four HCTL-1000 input/output ports are connected. Although the design of ISA-bus based machines is such that only the ten least significant address lines (A0-A9) are decoded into device addresses, the six most significant lines (A10-A15) do, in fact, carry valid address data. These lines have therefore been used to carry address data to the HCTL-1000 controllers, with the result that the low-order lines are left free for use in interface address decoding. Data to and from the HCTL-1000 controllers is conveyed via the ISA-bus data lines D0-D7, as is customary.

A functional block diagram of the interface circuitry is shown in Figure A.06. At the left of this diagram are the control inputs from the ISA-bus and the analytical plotter, and at the right are the control outputs to the four HCTL-1000 controllers, together with the address and data lines between them and the ISA-bus. This diagram shows how the interface circuit may be regarded as comprising ten sub-circuits, each performing a set function. The existence of two of these functions—read/write control and limit—requires particular explanation, as their purpose is not immediately obvious.

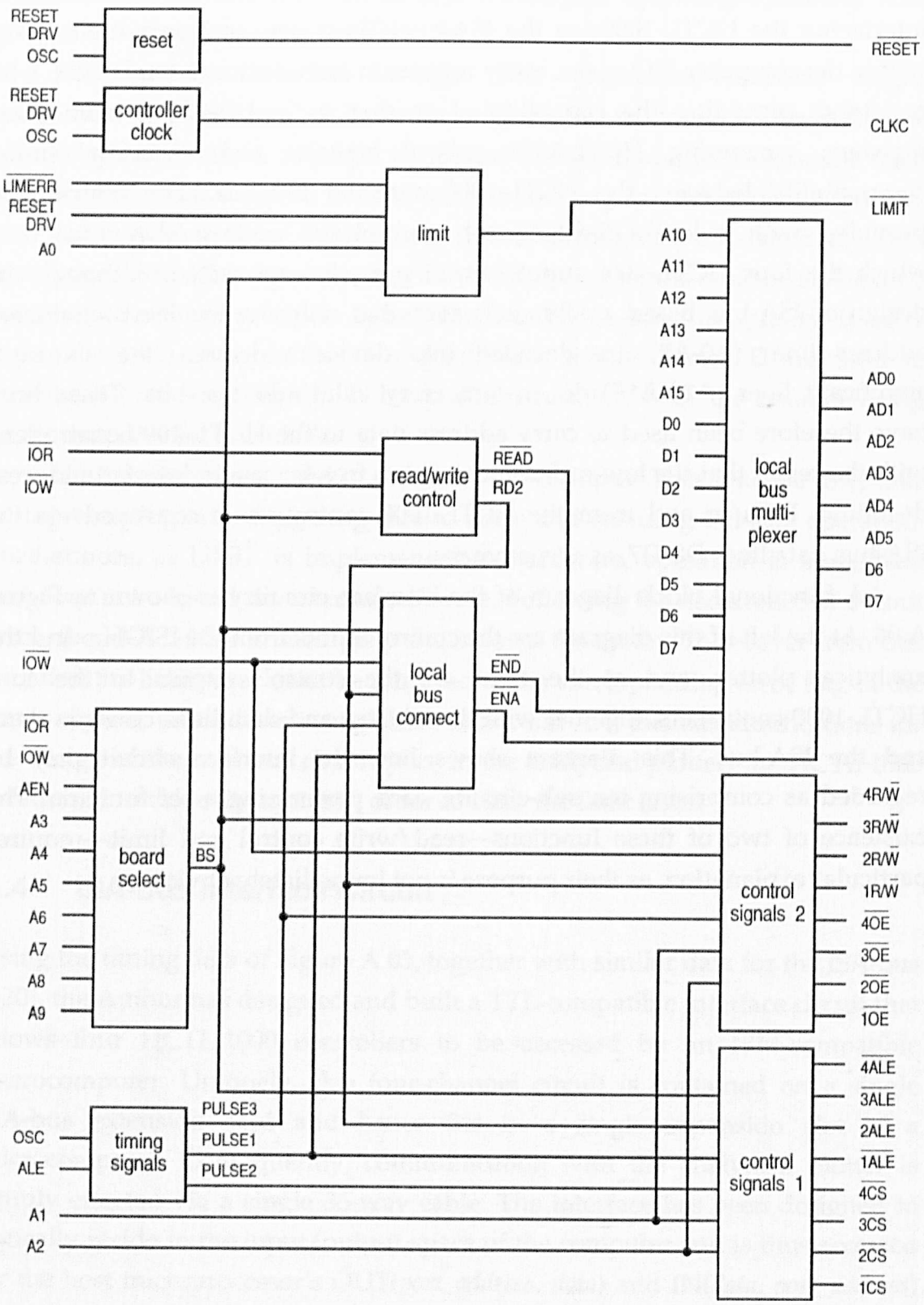


Figure A.06 Interface circuit block diagram.

(i) Read/write control

Implementation of the write function is straightforward, but that of the read function is complicated by the long (up to 3200ns) data fetch time of the HCTL-1000. If the read function is to complete in one cycle of the 4.77MHz ISA-bus, this fetch time can only be accommodated through the insertion of at least 15 wait states of 210ns each. This is far in excess of the permitted maximum of 10 wait states [120], and so in this design the actions either side of the data fetch in the HCTL-1000 read cycle have been separated into two ISA-bus based cycles: 'Initialise Read' and 'Data Read'. Providing at least 15 ISA-bus clock cycles elapse between these read cycles, the data fetch requirements are met. This is not an onerous condition, as the number of clock cycles required by the host processor to execute the instructions necessary to access the device a second time will be in excess of this [112]. Thus, the purpose of the read/write control circuit is to differentiate among the three access functions of Write, Initialise Read and Data Read.

(ii) Limit

The design of the HCTL-1000 is such that, after a limit error has been triggered, the device cannot be used for further control until the mechanical components causing the error have been moved to clear the error. This can be performed manually, but if it is to be performed using the controller itself, an override is required. The purpose of the limit circuit is to provide this override. Whilst the override is active, the motors may be moved under program control (i.e. at this point operator intervention is not possible) to bring them back into limit. Following this, the override is reset and control returned to the operator.

The interface circuit has been constructed on a commercially available wire-wrap board, using HC-TTL [132] devices mounted in IC sockets. Power and Ground lines, together with all decoupling capacitors, have been soldered. In recognition of the fact that the host microcomputer would become obsolete, but that ISA-bus machines would continue to be available, every attempt was made to ensure that each functional sub-circuit could be simply altered to account for faster, or non-standard, implementations of the ISA-bus. Consequently, the design of a number of the sub-circuits is not the simplest (in terms of device count or wiring), but rather the most easily reconfigured. The

implementation of the sub-circuits illustrated in Figure A.06, together with their operation, is now described in detail.

(i) Board Select

The interface must respond only when board-specific input/output is being performed. In order to accomplish this, the following criteria must be fulfilled: the interface must logically reside at a unique address within the address space accessible to the computer; and as the interface is located within the input/output space, it must not respond to either microprocessor-controlled or direct memory access. This has been implemented, using an HC688 8-bit comparator to generate the active-low signal $\overline{\text{BS}}$ (*B*oard *S*elect), as follows: the ISA-bus signal AEN is utilised to identify when DMA (*D*irect *M*emory *A*ccess); the ISA-bus signals of $\overline{\text{IOR}}$ (*I*nput/*O*utput *R*ead) and $\overline{\text{IOW}}$ (*I*nput/*O*utput *W*rite) are ORed to identify when I/O access is in progress; and address lines A3-A9 provide a unique address.

The active state of each of the address lines A3-A9 is switch selectable, to provide the facility for logically locating the interface anywhere within the address space and hence avoiding device contention.

(ii) Timing Signals

An HC164 shift register in combination with a D-type flip-flop generates the three fundamental timing signals PULSE1, PULSE2 and PULSE3. Feedback from the flip-flop provides the shift register with an input signal exactly three OSC cycles in duration. The ISA-bus signal of ALE is inverted and used to restart the sequence by clearing the shift register/flip-flop combination, thus ensuring the pulses, and hence interface operation, are synchronised with the ISA-bus input/output cycles.

(iii) Read/Write Control

A D-type flip-flop with feedback, controlled by $\overline{\text{IOR}}$, $\overline{\text{IOW}}$ and $\overline{\text{BS}}$, generates the signals $\overline{\text{READ}}$ and RD2. These, together with $\overline{\text{IOW}}$, are used to differentiate among the three cycles of Write, Initialise Read and Data Read.

Each write to the board clears the flip-flop, thus clearing $\overline{\text{READ}}$ and RD2. At the end of the first read cycle following this, the flip-flop is

clocked when $\overline{\text{IOR}}$ rises, and the feedback loop causes both $\overline{\text{READ}}$ and RD2 to be asserted. These signals remain in this state (providing a write to the board does not take place) until the end of the next read cycle, when the flip-flop is again clocked when $\overline{\text{IOR}}$ rises. At this point, the feedback loop causes $\overline{\text{READ}}$ and RD2 to be cleared.

(iv) Local Bus Connect

An HC352 dual 8-line to 1-line inverting data selector is used as a programmable combinatorial logic block [80] to generate the two local bus connect signals $\overline{\text{END}}$ (i.e. *EN*able *D*ata) and $\overline{\text{ENA}}$ (i.e. *EN*able *A*ddress). $\overline{\text{BS}}$ enables the devices, with $\overline{\text{IOW}}$, PULSE1, PULSE3 and RD2 generating the connect signals according to the following truth tables:

SIGNAL	PULSE3	$\overline{\text{IOW}}$	RD2	$\overline{\text{END}}$
PIN	C*x*	A	B	OUT
C0	0	0	0	1
C1	Q	0	1	$\overline{\text{Q}}$
C2	Q	1	0	$\overline{\text{Q}}$
C3	0	1	1	1

PULSE1	$\overline{\text{IOW}}$	RD2	$\overline{\text{ENA}}$
C*x*	A	B	OUT
Q	0	0	$\overline{\text{Q}}$
0	0	1	1
Q	1	0	$\overline{\text{Q}}$
0	1	1	1

Q in the C*x* column of these tables indicates a signal that may take on either of the logic states 0 and 1, and $\overline{\text{Q}}$ in the OUT column indicates that the complement of this signal is made available as the output.

(v) Control Signals 1

An HC352 dual 8-line to 1-line inverting data selector is used as a programmable combinatorial logic block to generate the two HCTL-1000 control signals $\overline{\text{ALE}}$ and $\overline{\text{CS}}$. $\overline{\text{BS}}$ enables the devices, with $\overline{\text{IOW}}$, PULSE1, PULSE2 and RD2 generating the signals according to the following truth tables:

SIGNAL	PULSE1	$\overline{IOW}$	RD2	$\overline{ALE}$	PULSE2	$\overline{IOW}$	RD2	$\overline{CS}$
PIN	C*x*	A	B	OUT	C*x*	A	B	OUT
C0	Q	0	0	$\overline{Q}$	Q	0	0	$\overline{Q}$
C1	0	0	1	1	0	0	1	1
C2	Q	1	0	$\overline{Q}$	Q	1	0	$\overline{Q}$
C3	0	1	1	1	0	1	1	1

An HC139 dual 2-to-4 line multiplexer under the control of ISA-bus lines A1 and A2 then decodes the relevant controller number.

(vi) Control Signals 2

An HC138 3-to-8 line multiplexer generates a generic signal that is subsequently decoded by RD2 into the HCTL-1000 control signals $R/\overline{W}$ and $\overline{OE}$, depending on whether RD2=0 or RD2=1, respectively. Additional decoding to determine the relevant controller number is provided by the ISA-bus lines A1 and A2. In effect, the internal circuitry of the HC138, together with the HC02, computes the enable as $EN = \overline{BS} \cdot PULSE3 \cdot (\overline{IOW} + RD2)$.

(vii) Local Bus Multiplexer

An HC541 uni-directional octal buffer and an HC245 bi-directional octal bus transceiver multiplex the ISA-bus address lines of A10-A15 and data lines of D0-D7 respectively onto the local eight-bit bus, to which the four HCTL-1000 controllers are connected. Additionally, the HC245 drives the ISA-bus data lines D0-D7 during the Data Read cycle.

(viii) Limit

The active-low signal $\overline{LIMERR}$ from the sterecomparator, which indicates a limit error on one or more of the axes, is passed to the HCTL-1000 controllers via an override circuit based on a D-type flip-flop. The override is operated by clearing A0 and writing to the interface, and subsequently reset by asserting A0 for all succeeding reads and writes.

(ix) Reset

The active-low $\overline{\text{RESET}}$ signal required for a hard-reset of the HCTL-1000 is generated by a 555 timer, working as a monostable source to produce a pulse of 3 second duration. The circuit is triggered by either a pulse produced from the ISA-bus RESET DRV signal or the push-button mounted on the board itself. RESET DRV is generated by the host computer at power-up and when a reset of the host itself is performed.

(x) Controller Clock

The ISA-bus 14.31MHz OSC signal is divided, firstly by an HC161 synchronous 4-bit binary counter, and secondly by a D-type flip-flop, to give a 50% duty cycle clock signal of a rate suitable for the HCTL-1000 controllers. The terminal count of the HC161 is switch selectable, allowing the following controller clock speeds to be chosen:

L4	L2	L1	count	HC161 o/p (MHz)	HC74 o/p (MHz)
0	0	0	8	1.79	0.89
0	0	1	7	2.04	1.02
0	1	0	6	2.39	1.19
0	1	1	5	2.86	1.43
1	0	0	4	3.58	1.79
1	0	1	3	4.77	2.39
1	1	0	2	7.16	3.58

As the HCTL-1000 is designed to operate at clock speeds in the range of 1MHz to 2MHz, only terminal counts in excess of 3 are usable.

A wiring schematic of the interface board circuit is shown in Figures A.07 and A.08. The sub-circuits referred to above are indicated by dashed lines around the relevant circuit elements. Constructional details such as decoupling capacitors, power and ground lines and connectors are not shown in these diagrams. The timing diagrams in Figure A.09 illustrate how the interface requirements of the ISA-bus and the HCTL-1000 are met using this circuit.

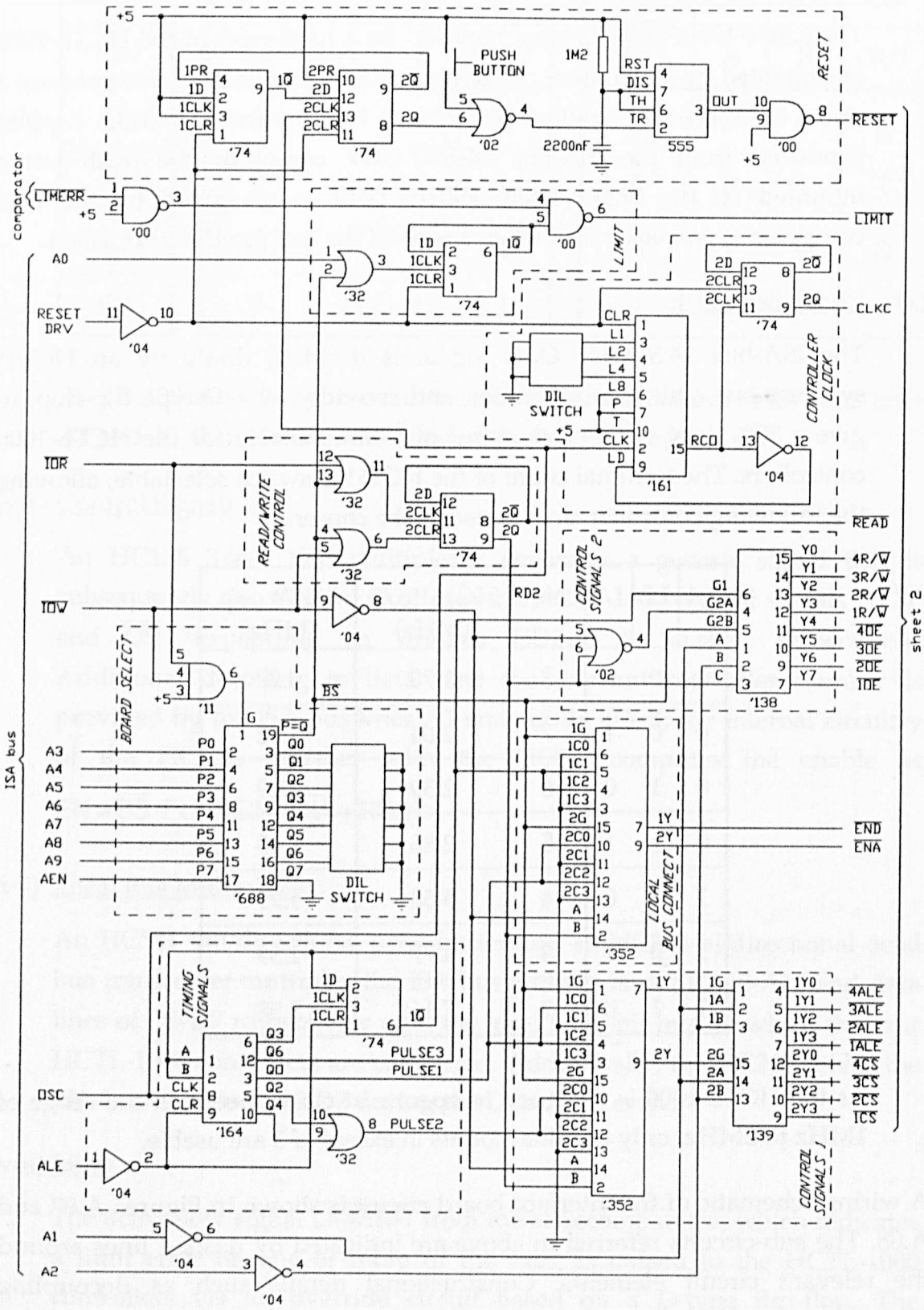


Figure A.07 Interface board schematic: sheet 1 of 2.

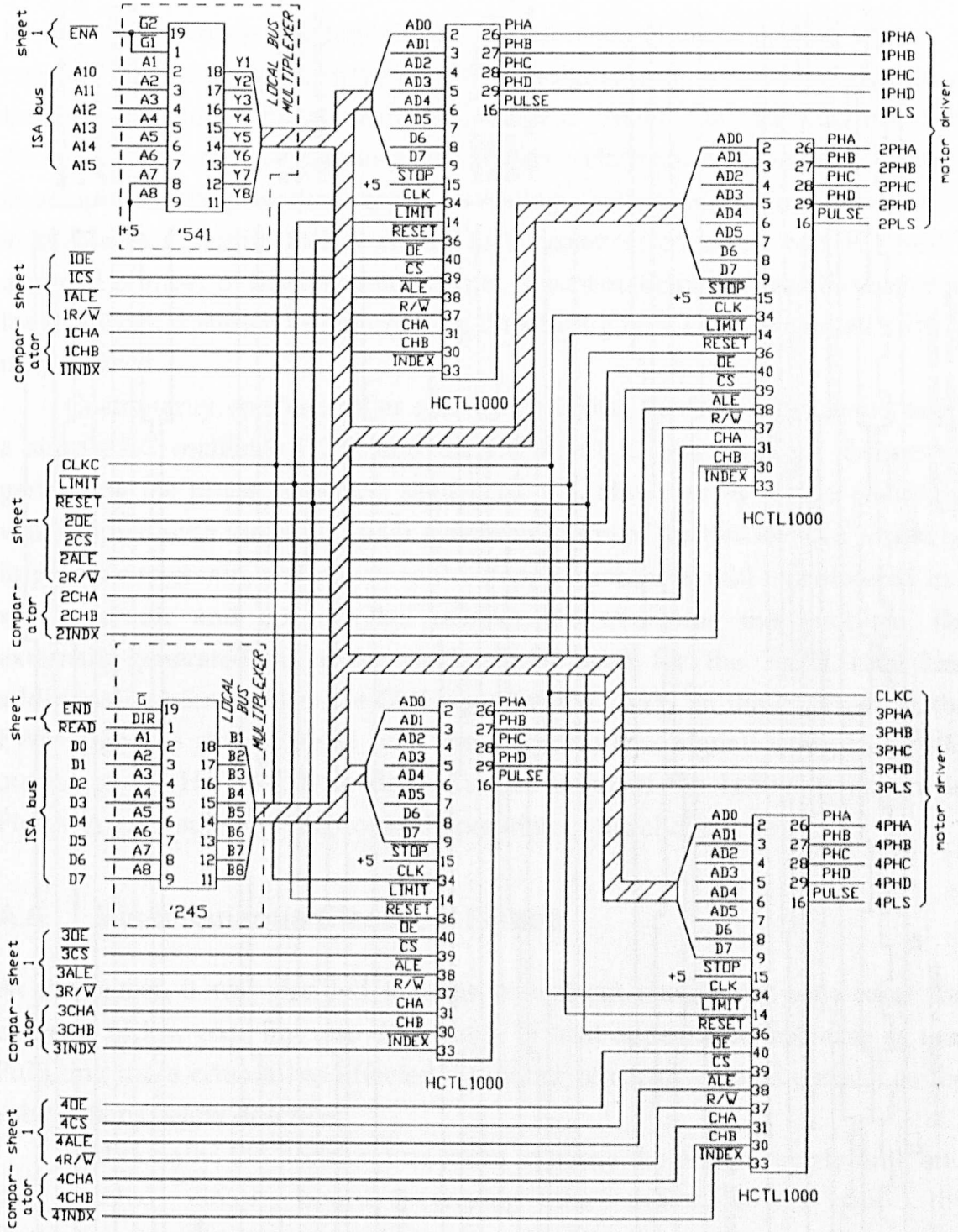


Figure A.08 Interface board schematic: sheet 2 of 2.

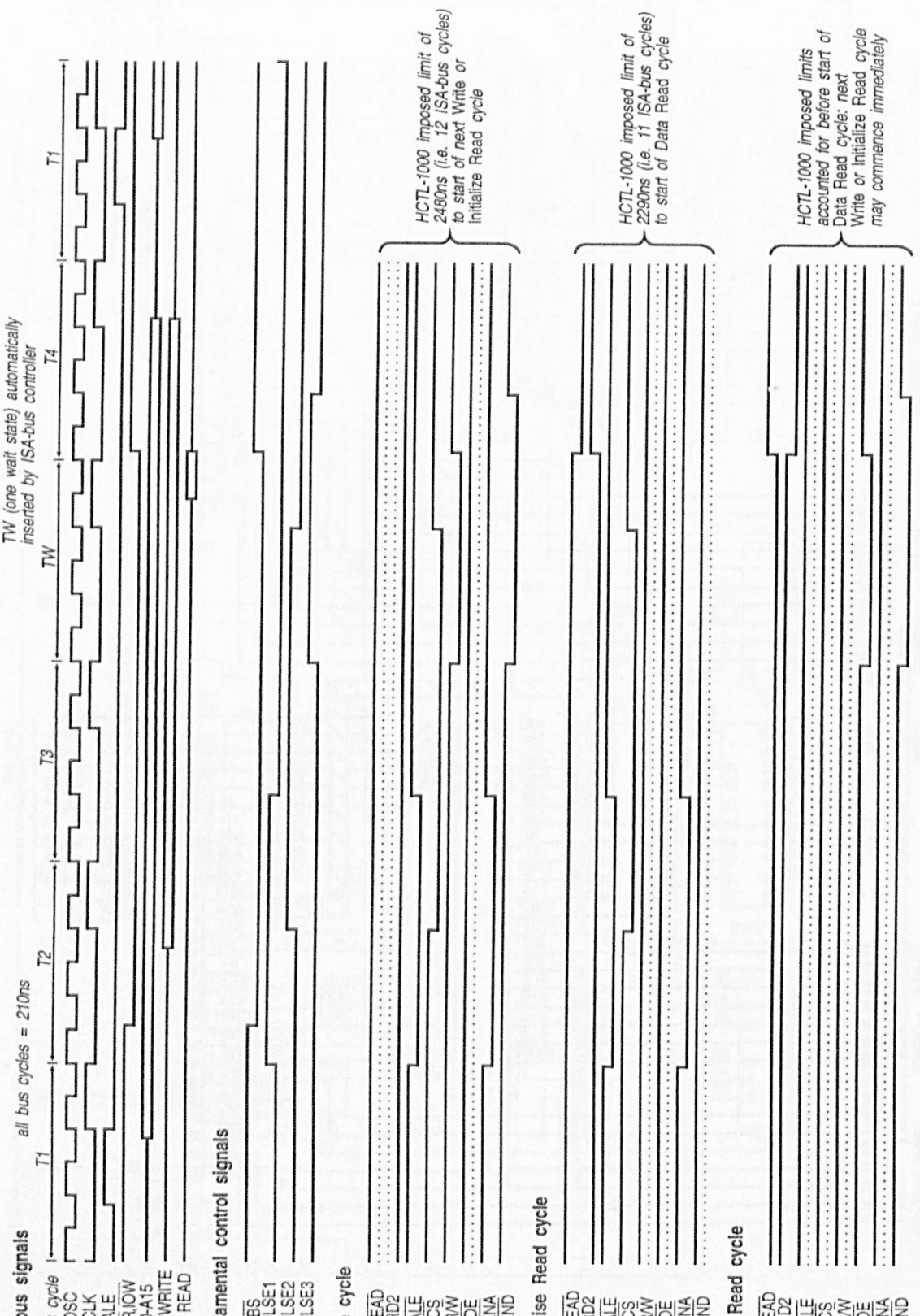


Figure A.09 Interface circuit timing diagrams.

A.5 Stepping Motor Switchmode Constant-Current Chopper Driver

In order to increase the high-speed performance of the stepping motors, a constant-current chopper drive has been designed and constructed. Drives of this sort are more efficient, and dissipate less power, than the common L/R drive [2], but require sophisticated control electronics. However, modern developments in integrated circuit technology have resulted in devices being available that permit such a circuit to be constructed using two ICs and a minimal number of discrete components. A schematic of the circuit developed for this work is shown in Figure A.10, illustrating how such integrated circuits may be used.

Customarily, in a controller such as the L6506, the CLK input is driven by a simple RC oscillator [123], and the EN input is derived from the system generating the phase switching sequences [60]. However, if such a technique was adopted with the HCTL-1000, synchronisation of the two devices would be impossible to obtain, and power to the stepping motor would be delivered in a very sporadic and uncontrolled manner. To overcome this problem, the externally-generated controller clock signal used for the HCTL-1000 has, additionally, been used as the CLK input for the L6506. In the HCTL-1000, the PWM signal is synchronised with the external clock signal; using the PWM output of the HCTL-1000 to drive the EN input of the L6506, as shown in Figure A.11, results in synchronised operation of the chopper circuit.

A.6 Miscellaneous Electrical Issues

At the outset, it was decided that the equipment should not only meet the criterion of low-cost, but also the criteria of neat appearance and ease of use. Fulfilling these criteria has affected a number of constructional details, as the sub-Sections below describe.

Additionally, the decision was taken to house the power supply unit, and the printed circuit boards carrying the motor driver and the floating spot brightness control/repeater circuits, in a separate accessory cabinet. This reduces the connections that are the responsibility of the user to the three linking the mains supply, host microcomputer and analytical plotter to the accessory cabinet. The cabinet itself is a housing for an IBM-compatible

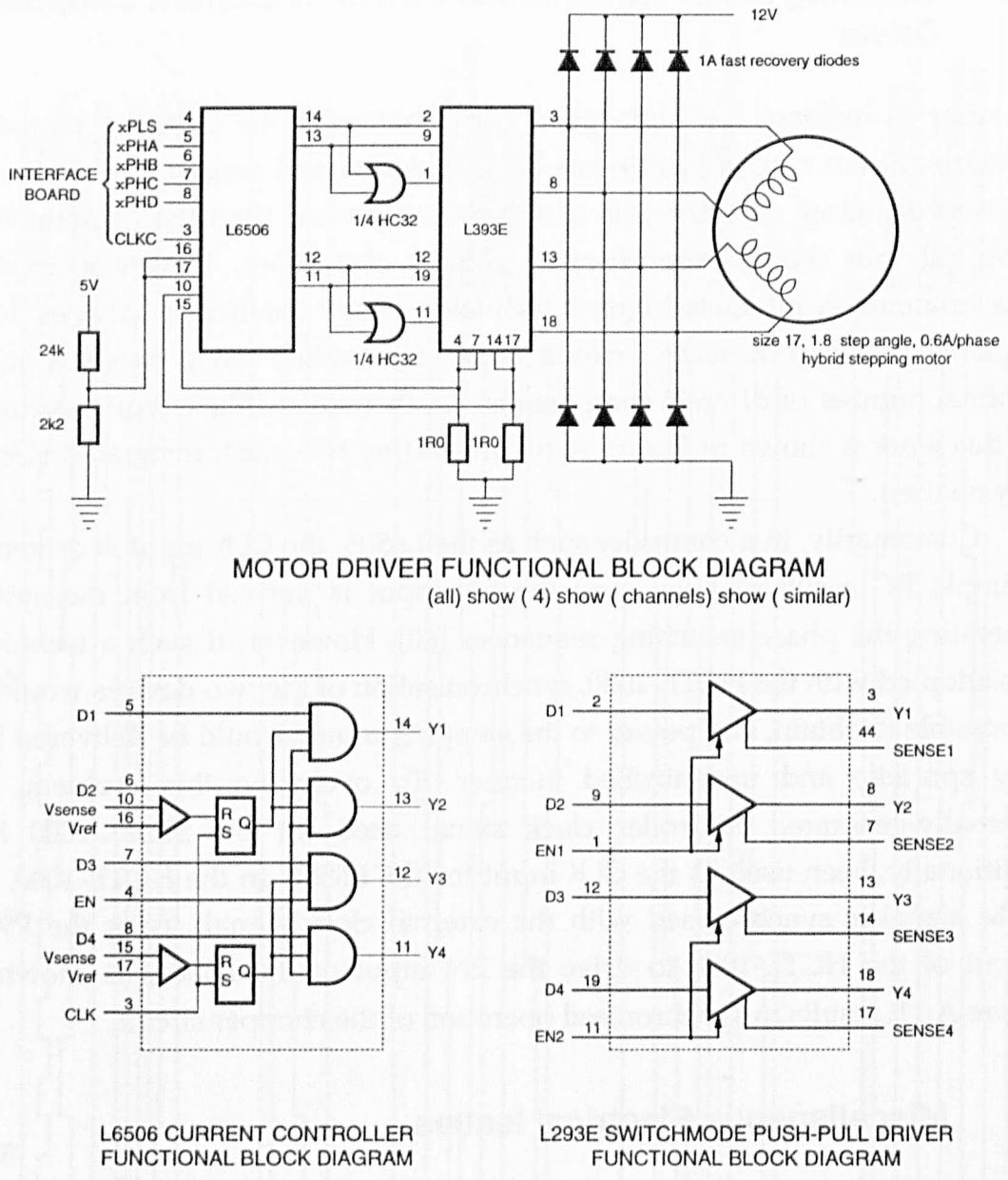


Figure A.10 Schematic of stepping motor switchmode constant-current chopper driver.

microcomputer, and is shown—complete with the enclosed circuits—in Figure A.12.

A.6.1 Optical limit switches

To protect the analytical plotter from mechanical damage resulting from movement of the photographs either beyond the limits of leadscrew travel or such that they would collide, a system of limit switches has been installed. To remove the problem of contact bounce associated with mechanical switches in logic circuits, TTL-compatible opto-switches, type HOA2001 manufactured by

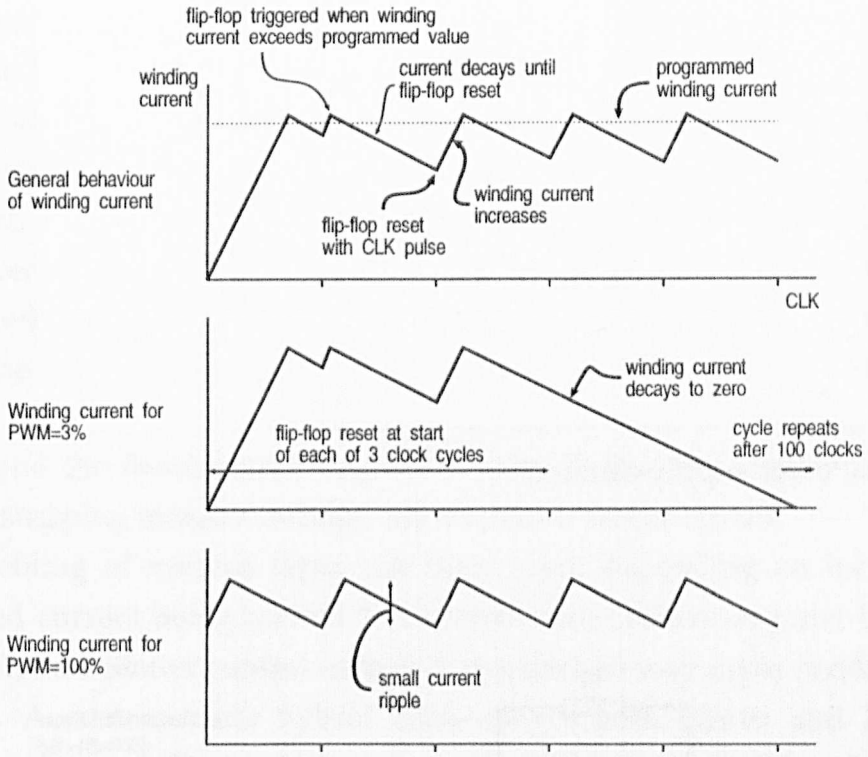


Figure A.11 Constant-current, pulse width modulated control of stepping motor windings.

Honeywell, were chosen. The slotted switches are operated by passing an opaque vane through the slot in the body of the switch, with the output signal being low when asserted. This arrangement is fail safe; dirt obscuring a switch, a malfunction or disconnection will trigger a signal interpreted as a limit error. Each switch was mounted on a small custom-made printed circuit board, to simplify mounting, power supply and connection details. A photograph of a limit switch in place on the analytical plotter, showing the arrangement adopted, has previously been given in Figure 5.17.

A.6.2 Floating spot brightness control

The floating spot brightness control originally fitted to the stereoscope was implemented using a mains input step-down transformer and a high-power wire-wound variable resistance to form a variable voltage controller. This arrangement proved unsuitable for this work, as the chosen power supply unit could provide only 5V and 12V supplies. Consequently, a current limited voltage controller, based on the SGS-Thomson L200, 2.0A, adjustable voltage regulator was constructed. The circuit diagram is shown in Figure A.13.

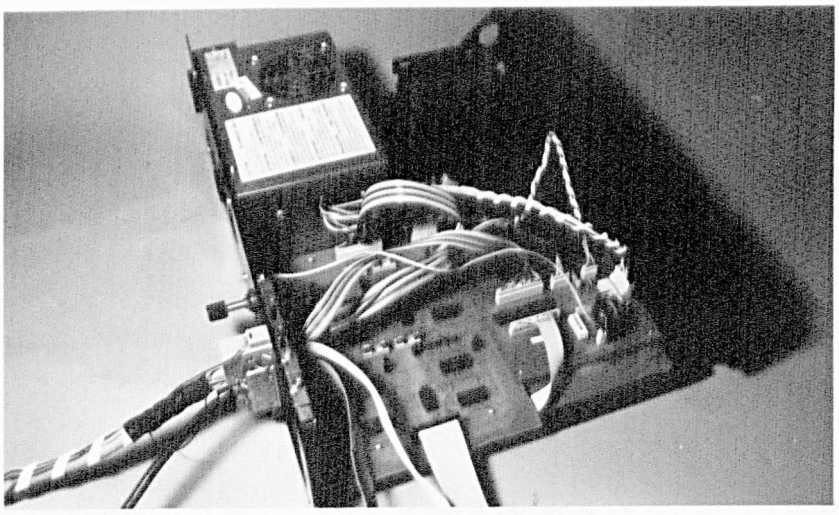


Figure A.12 Power supply cabinet.

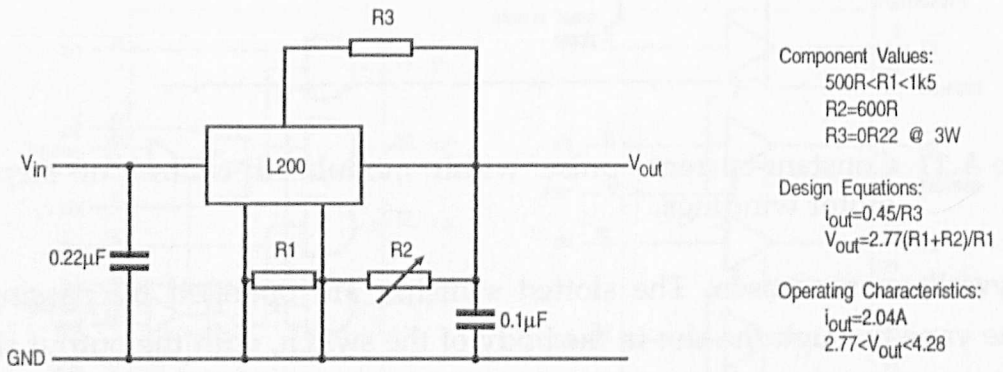


Figure A.13 L200-based floating spot brightness control.

A.6.3 Repeater circuit

Transmission line effects limit the length of cable over which TTL signals can be transmitted to about 3m. The total length of cable between components on the analytical plotter, and the HCTL-1000 controllers in the host computer, is in excess of this. To overcome this problem, a repeater circuit, consisting simply of two HC541 octal line drivers, has been installed on the same printed circuit board as the brightness control.

A.6.4 Power supply, cabling and connectors

The number of high-speed logic circuits in the equipment necessitates a regulated power supply with low output voltage ripple. Furthermore, the power supply should be in keeping with the philosophy of a low-cost instrument based on readily available components. Consequently, an industry standard 200W switch-mode power supply unit, as fitted in IBM-compatible microcomputers, was used. These units are able to supply 5V at 6A and 12V at 4A, together with low-current -5V and -12V supplies and a Power Good signal. Here, the 5V supply powers the logic elements associated with the motor driver circuit, the limit switches, the analytical plotter terminal board, the repeater board and the floating spot brightness control. The 12V supply is used solely for the stepping motor windings, via the motor driver circuit.

Cabling of various types has been used, depending on location, signal type and current being carried. Between the host microcomputer interface and the analytical plotter control cabinet a shielded 36-way cable carries logic-level signals. A custom-made hybrid cable carries both power and logic signals between the control cabinet and the analytical plotter. In this cable, a 20-way double screened (i.e. foil and braid) cable, is surrounded by a 22-way ribbon cable, which comprises 14×0.13mm stranded conductors. These carry, respectively, logic signals from the rotary encoders and the limit switches, and high-current supplies to the stepping motor windings, the floating spot lamps and the limit switches and the rotary encoders. Given the criterion of limiting the analytical plotter connections to one, this configuration was adopted in order to ensure that the noise-susceptible logic signals were isolated from the noisy power lines, and yet the whole was contained within a single cable. On

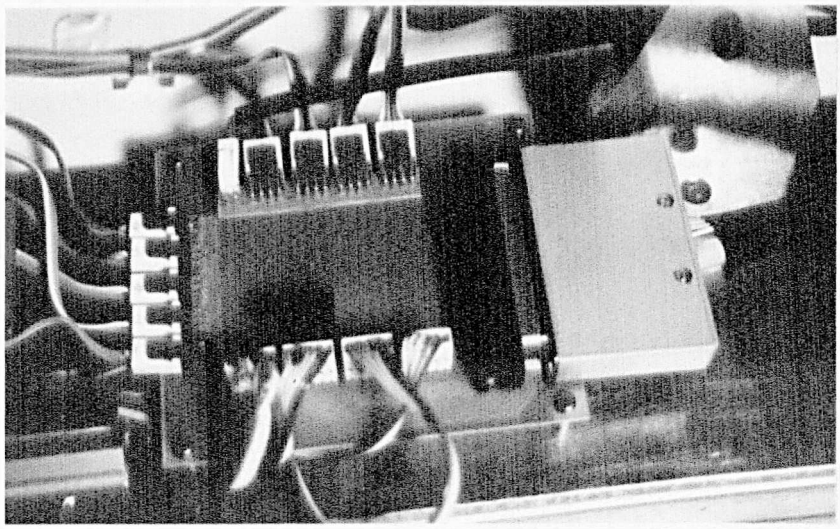


Figure A.14 Analytical plotter terminal board.

the analytical plotter itself, ribbon cables of varying width, and comprising 14×0.13mm stranded conductors, connect the various components to the terminal board. Figure A.14 shows how the terminal board interfaces the various ribbon cables to the main hybrid cable that leads to the power supply cabinet.

The variety of components and circuits in the instrument has led to a variety of connectors being used, although the number has been kept to a minimum. In the cables linking the host microcomputer to the control cabinet, and the analytical plotter to the control cabinet, the choice of connector was lead by the need to prevent incorrect connection of the cables. Thus, a 37-way D-type plug and a 36-way Centronics plug form the former cable, a 37-way D-type socket and a 37-way D-type plug form the latter. On the analytical plotter, 2.54mm pitch KK connectors are used exclusively, as they are easily used with ribbon cable and permit compact terminations to be made on the electrical components (e.g. the custom-made connector units on the stepping motors and limit switches, and the manufacturer's design on the rotary encoders). IDC (insulation displacement connectors) and KK connectors are used on the motor driver and brightness control/repeater boards, for logic and current signals, respectively.

A.7 HCTL-1000 Programming

In order to properly configure the HCTL-1000 controllers, they must be programmed by loading certain registers. In the implementation described here, registers associated with the commutator port and the digital filter must be preset.

Figure A.15 demonstrates graphically the possible values for the commutator registers RING, X and Y, with the components used in this work. With a 500 counts/revolution encoder and a 1.8° stepping motor, a RING of 40 is defined. By changing the values of X and Y, as shown, it is possible to generate 2-phase on, 1-phase on and half-step driving modes. Although the driver will normally be used in half-step mode, the other modes are available for test and development purposes.

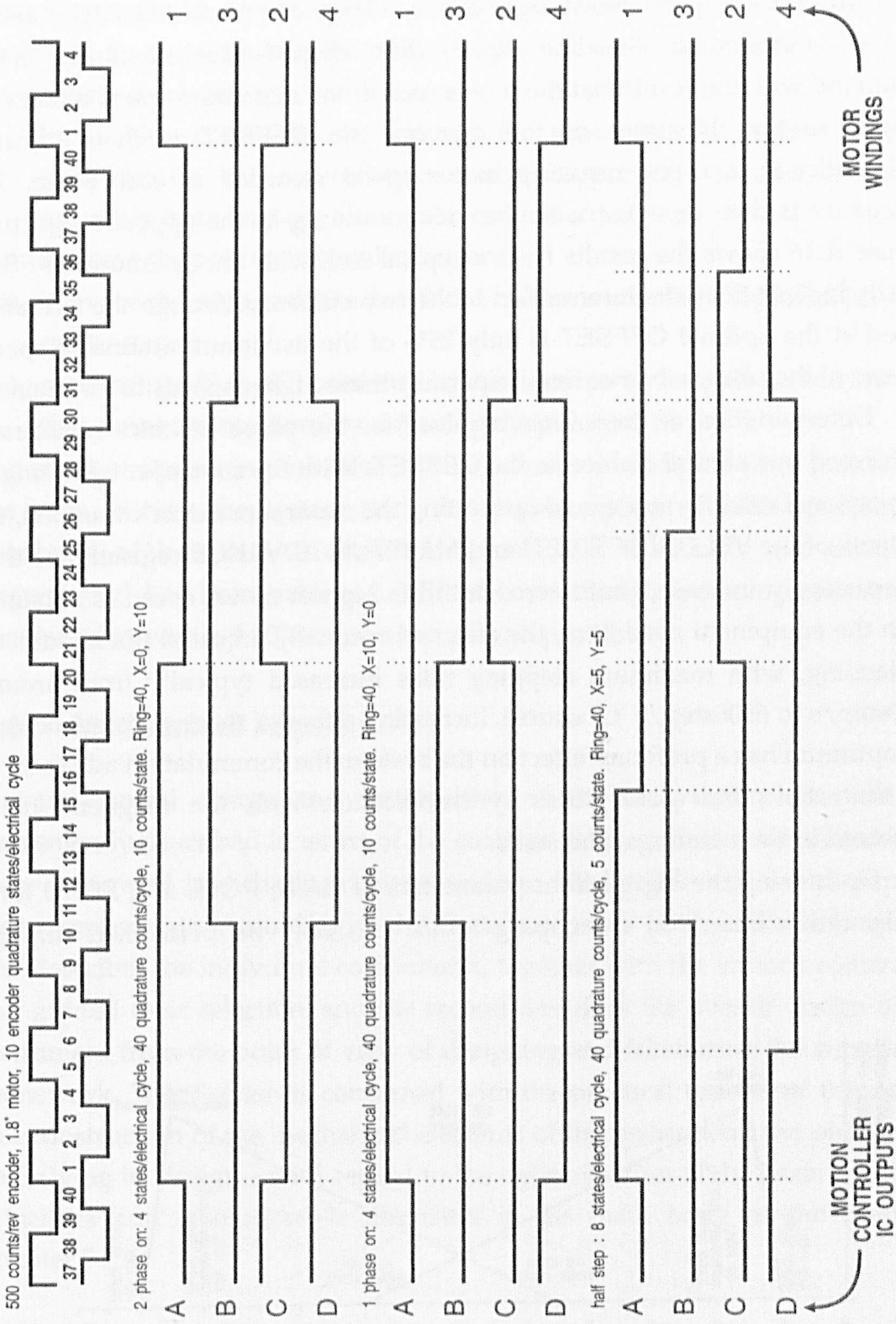


Figure A.15 Stepping motor driving sequences using HCTL-1000 commutator port.

The required value of the OFFSET is dependent on the assembly of the motor/encoder assembly, and as such must be determined experimentally for each motor. In proportional velocity mode, a large command velocity is issued to the controller, so that the digital filter outputs the maximum motor command with the result that the motor runs at the maximum speed attainable, subject to the dynamics of the system. The OFFSET is then adjusted systematically, and the resulting motor speed recorded at each stage. The procedure is then repeated with the motor running in the opposite direction. Figure A.16 shows the results from a typical test, with the optimum OFFSET clearly indicated by the intersection of the two curves. Although the operating speed at the optimal OFFSET is only 25% of the maximum attainable speed, the use of the phase advance feature permits these higher speeds to be attained.

Determination of the values required for the phase advance registers is performed in a similar fashion to the OFFSET. With the controller operating in proportional velocity mode, and outputting the maximum motor command, the contents of the VELOCITY TIMER and MAXIMUM ADVANCE registers are then systematically increased from zero, until the highest motor speed is obtained. With the equipment used here, the effect of optimally adjusted phase advance is startling, with maximum stepping rates increased typically from around 1700step/s to 6800step/s. Of course, increasing either of these registers beyond the optimum has a profound effect on the system: the commutation advances so far that the motor rotor loses synchronisation with the magnetic fields produced in the windings, and stalls.

Optimising the digital filter parameters of GAIN, POLE and ZERO for a configuration based on a stepping motor can only be performed through

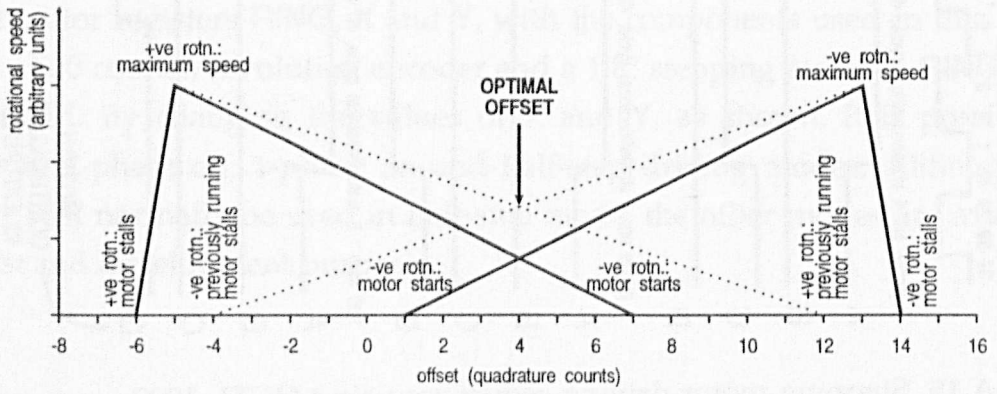


Figure A.16 Typical OFFSET optimisation curves.

experimentation. The analytical transfer function techniques available in control theory [16, 78] are not applicable to this system, as the response of the stepping motor to the non-linear PWM signal (see Figure A.11) cannot be linearised by mapping into the complex plane. The procedure thus adopted to optimise these parameters is as follows:

(i) in position control mode, increase the gain until the system becomes unstable, i.e. the controller is unable to prevent the motor/encoder assembly from hunting around its command position;
(ii) increase the pole until stability returns;
(iii) increase the zero until the deadband becomes unacceptably large (say, ten motor steps);
(iv) repeat steps (i) to (iii) until any further increase in the gain causes hunting that cannot be removed by adjusting the pole and zero.

The effect of optimising the offset, phase advance and digital filter parameters has been to tune the system such that a peak speed of 6800step/s, and a deadband of less than 10 quadrature counts (i.e. 21.6mm/s and 0.006mm, respectively) has been achieved. With this deadband, a photograph will be positioned to within 0.003mm of its command position.

A.8 Mechanical Aspects

The mechanical design and construction of the analytical plotter can be conveniently described in terms of the commercially available components that were integrated into the instrument, and the framework required to support these components. Accordingly, this Section consists of two sub-Sections: the first describes the individual components, together with the various constraints that guided their selection; and the second describes the overall design of the instrument from the point of view of designing and fabricating the supporting framework. This Section is concerned with the practical aspects of the design and construction of the mechanical elements of the instrument; the philosophy underlying its design—with regard to the determination of the basic operating principles and resolution—is discussed in the main body of the thesis at Chapter 5.

A.8.1 Components

In order to produce an analytical plotter, mechanical components that are capable of moving and positioning the two photographs with a high degree of

accuracy are required. Rather than attempt to fabricate such specialised items, it is most convenient to identify commercially available components for these tasks. Consequently, this sub-Section presents descriptions, in the following Divisions, of the: stepping motors and associated couplings; linear bearings; leadscrews, drivenuts and mounting bearings; and a novel coupling between the leadscrews and the carriages running on the linear bearings.

Motors and couplings

Three types of stepping motors are available on the market: variable reluctance, permanent magnet and hybrid. Acarnley [2] provides a thorough assessment of the relative advantages and disadvantages of the three types and, on the basis of this, hybrid motors were identified as appropriate for the analytical plotter due to their small step angles, high power output to size ratio and excellent dynamic performance. The particular motors used were size 17 (i.e. 42mm overall dimension of stator), 4 phase, 0.6A/phase, 1.8° step angle stepping motors, manufactured by Sanyo Denki, Japan. These particular phase currents were chosen to comply with the output capacity of the power supply unit.

The general philosophy of designing for minimal machining tolerances meant that the motors were required to be connected to the leadscrews through couplers capable of absorbing any resulting lack of fit. The simplicity of the design meant that this could be manifested as a combination of both non-straightness and non-parallelism between the axes of the motor shaft and the leadscrew. At the same time as coping with these constructional tolerances, these couplings would be required to not induce any backlash. Of the many available devices on the market, it was felt that standard Oldham couplings fitted with a resilient acetal connecting element would best fulfil these criteria, as well as providing for easy assembly and disassembly of the motors, and hence these were adopted.

Linear bearings

The critical requirements of the linear motion system adopted in the analytical plotter were high linear tolerances in a compact component. Although the low load and speed characteristics of the instrument meant that neither of these requirements was particularly onerous in isolation, few commercially available linear bearings could match both of them. Finally, a novel system composed of 12mm diameter hardened and ground steel shafts supported in a light, but rigid, alloy extrusion was identified. These components, distributed in the UK

by Unimatic Engineers Limited, London, are specified with a parallelism and straightness of better than 0.01mm per 1000mm. The moving parts of this system are formed from elements containing two recirculating tracks of ball bearings, as shown in Figure A.17. These elements are provided with plain mounting holes, such that they may be simply affixed to suitable carriages.

Two variants of these bearings were available, in which either one or two steel shafts were mounted within the alloy extrusion, as shown in the schematic diagrams of Figure A.17. This allowed optimisation of the configuration of the instrument, as is discussed in sub-Section A.8.2.

Leadscrews, drivenuts and bearings

To achieve the necessary small linear step distance, leadscrews were adopted as a means of positioning the elements of the analytical plotter. To ensure that no large errors would be induced by the threads of these components, precision ground leadscrews with high lead accuracy were indicated. Of the components readily available in London, those manufactured by Precision Industrial Components Corporation, Connecticut, USA, and distributed by Unimatic Engineers, London, were found to be most suitable. Hence, ¼-20 (i.e. 0.25" diameter, 0.05" pitch) Acme-thread leadscrews, ground to a lead accuracy of

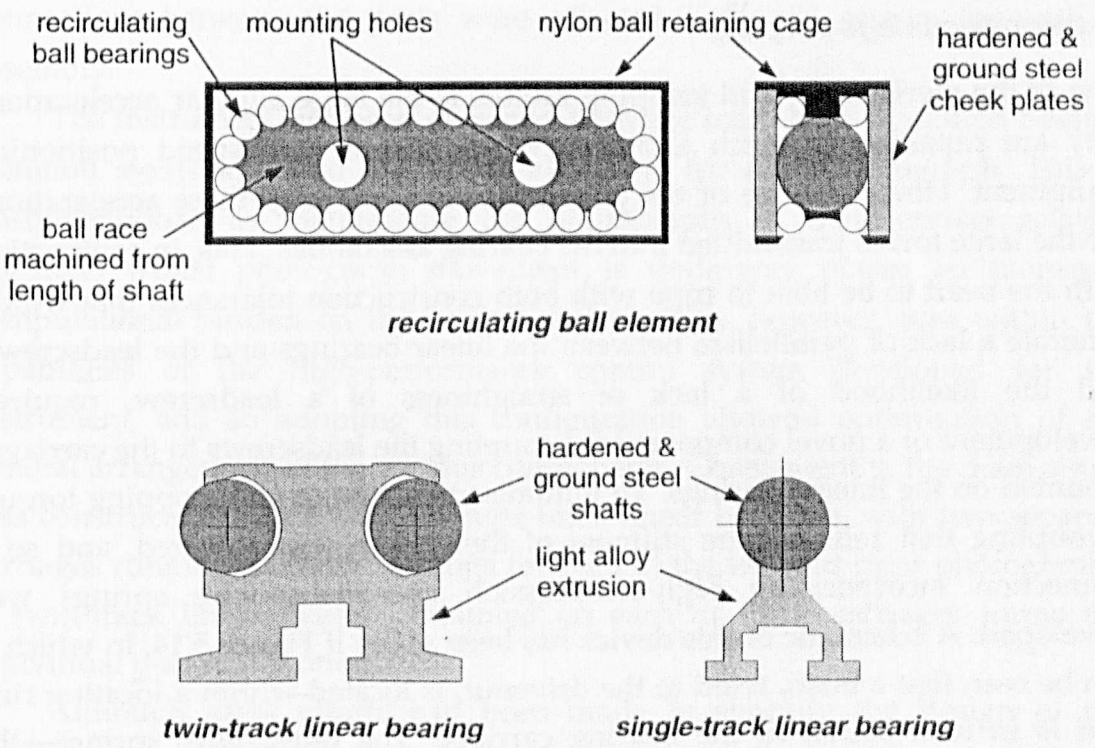


Figure A.17 Recirculating ball elements and single- and twin-track linear bearings.

±0.003"/ft and ±0.00025"/in were adopted. In conjunction with stepping motors capable of 400 discrete steps per revolution, this thread indicates a fundamental resolution for the analytical plotter of 0.003mm.

In order to achieve this small resolution, care was required in the choice of drivenuts and mountings for the leadscrews. Fortunately, the manufacturer of the leadscrews also produced suitable anti-backlash drivenuts, machined from acetal and impregnated with Teflon and silicone, and so these were fitted. The mounting of the leadscrew was undertaken using miniature needle-roller bearings held in plummer blocks. By machining shoulders onto the leadscrews, and fastening the plummer blocks onto the base plate of the analytical plotter, an arrangement was created whereby the bearings could be held tight against the shoulders to prevent axial play of the leadscrew. This arrangement has been found to be eminently satisfactory, in terms of producing low-friction running and secure mounting.

The shoulders machined onto the ends of the leadscrews have been used to affix both the Oldham couplings and the rotary encoders. Although it is customary to mount a rotary encoder directly onto the motor it is monitoring, it was felt that, as leadscrew rather than motor rotation was the important motion in this instrument, mounting the encoders onto the leadscrews was appropriate.

Leadscrew/carriage coupling

One of the merits of hybrid stepping motors is the large angular accelerations they are capable of, which allows development of high speed positioning equipment. However, one of the penalties associated with these accelerations are the large forces transmitted into the bearing assemblies. This, in conjunction with the need to be able to cope with both construction tolerances that would generate a lack of parallelism between the linear bearings and the leadscrews, and the likelihood of a lack of straightness of a leadscrew, required development of a novel component for coupling the leadscrews to the carriages mounted on the linear bearings. To minimise the starting and stopping torque, a coupling that reduced the stiffness of the system was required, and so a connection incorporating high-compression rate music-wire springs was developed. A schematic of this device has been given if Figure 5.14, in which it can be seen that a bush, fixed to the drivenut, is located within a locating ring that is in turn bolted to the bearing carriage. The music-wire springs—the

compression rates of which are 2.61N/mm—are held within the locating ring, and act upon the bush.

This device is successful in reducing system stiffness, and the springs ensure that the relative positions of the locating ring and the bush are maintained whenever the system is stationary. As the only requirement for coordination is when the system is stationary, the presence of the springs does not adversely affect the positioning accuracy of the instrument.

A.8.2 Fabrication and construction

The analytical plotter can be thought of as an accurate assembly of the various mechanical components described in the previous sub-Section, together with the mirror stereoscope. In order to maintain these components in their correct positions, a framework fabricated from simple parts machined from aluminium alloy was designed. To reduce the complexity of the instrument, the four axes of motion were designed as identical assemblies of stepping motors, leadscrews, encoder, and couplings.

The decision was taken to design the framework such that all of the component parts were bolted together, in order that high tolerance construction would be avoided and hence only those machining procedures that are normally available in an engineering workshop could be utilised. The connections between the parts were effected with cap screws for ease of assembly.

The instrument has been designed with the main axes of motion being a common x-axis and two separate y-axes, one for each photograph. This is unconventional, as maintaining the photographs in their correct relative positions whilst photograph movement is underway places an increased computational burden on the control system. This, however, was within the capabilities of the high-performance control system developed for the instrument, and so adopting this configuration allowed optimisation of the general arrangement of the various components. Consequently, the main x-axis was constructed from a pair of single-track linear bearings, with two separate carriages running on these bearings for each of the left and right photographs. A twin-track linear bearing mounted on each of these carriages forms the individual y-axes of motion.

Although great efforts had been made to simplify the design of the individual parts of the instrument, fabrication was still a complex job, and so a series of some eleven engineering drawings was produced by the Author to

ensure faultless construction and assembly. Three of these drawings are reproduced here as Figures A.18, A.19 and A.20. These show, respectively, the following: a general arrangement of the base plate, on which the x-axis linear bearings, stepping motors and leadscrews, together with the supports for the mirror stereoscope, are mounted; one of the two upper carriage plates, showing a y-axis linear bearing, stepping motor and leadscrew; and a sectioned general arrangement through both the lower and upper bearing assemblies. A photograph of the completed instrument has already been presented in Figure 5.20.

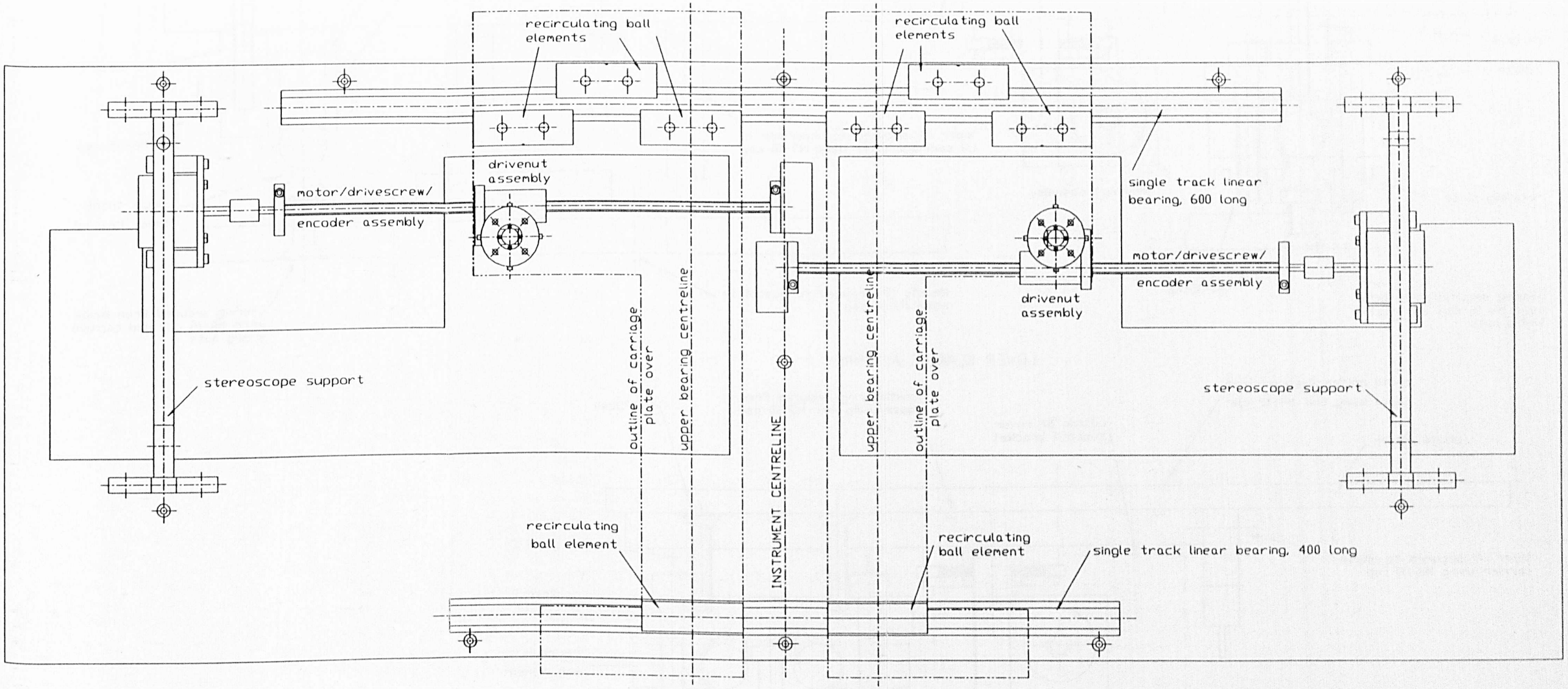


Figure A.18 Analytical plotter: base plate general arrangement

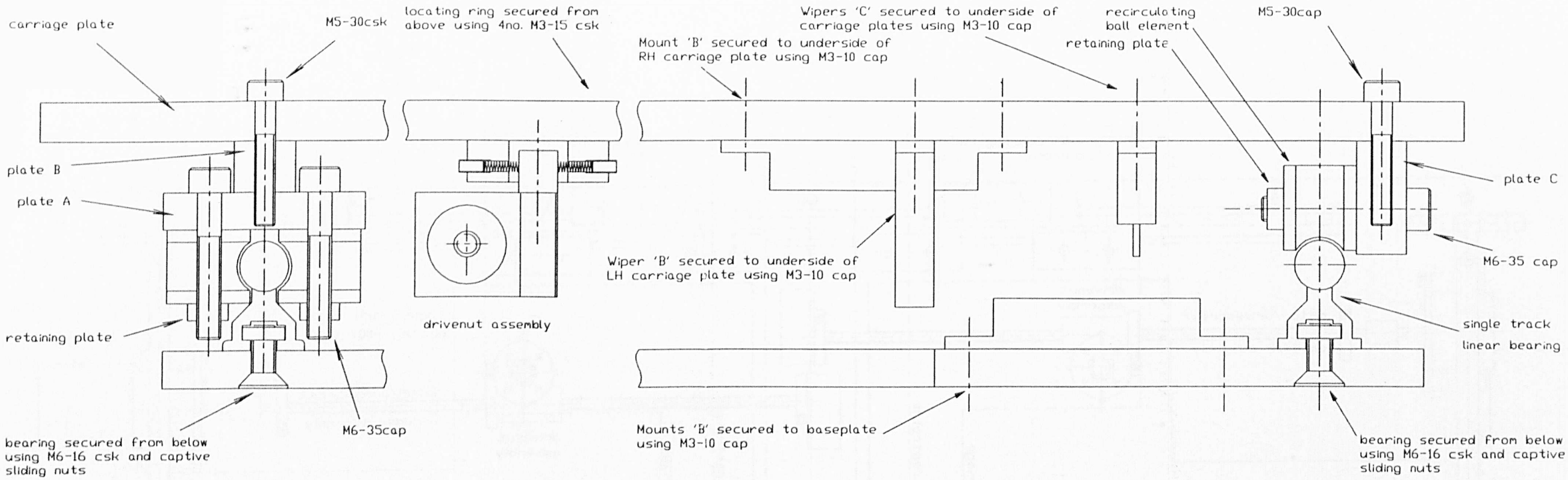


Figure A.19 Analytical plotter: sectioned general arrangements

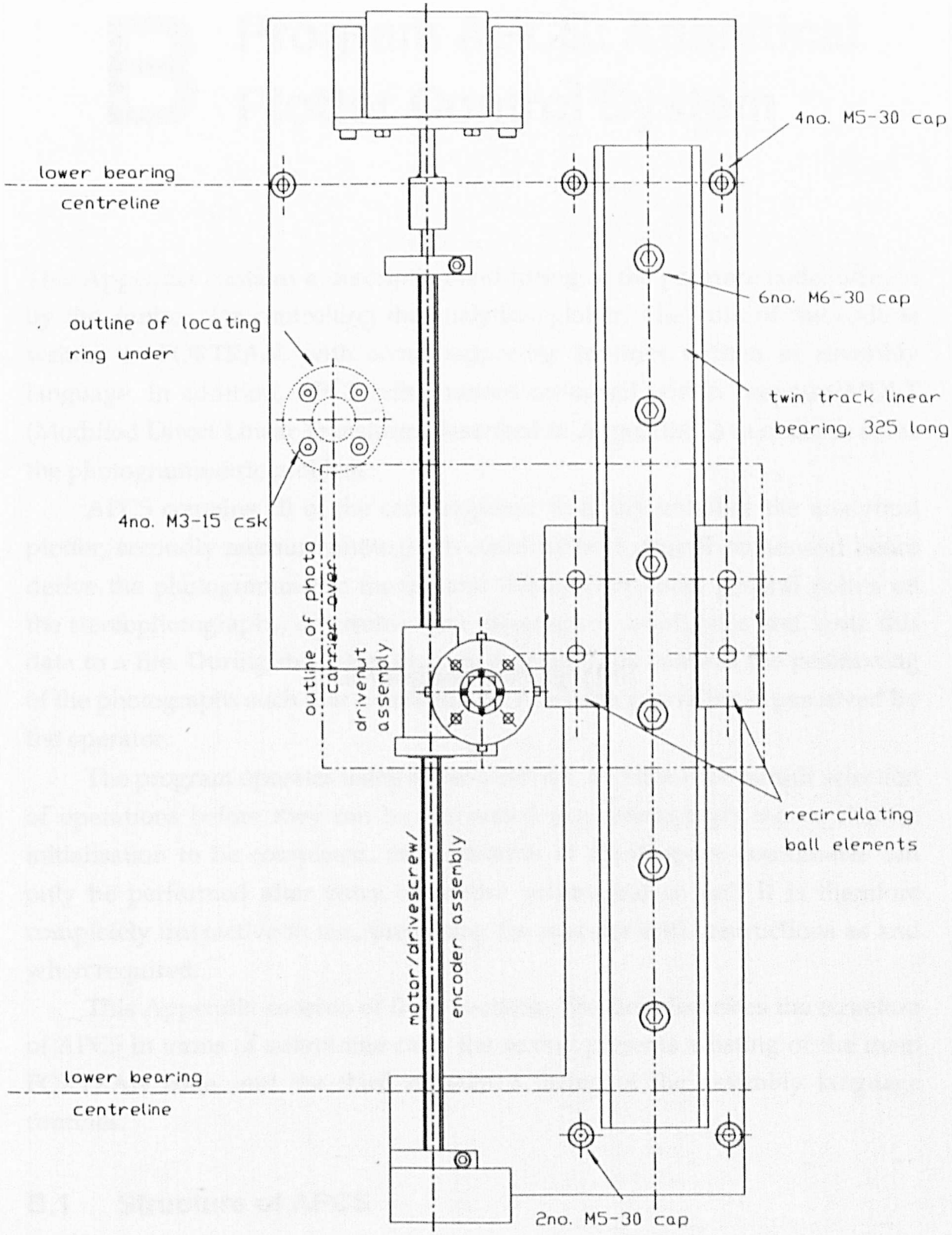


Figure A.20 Analytical plotter: general arrangement of left carriage plate

This page intentionally left blank

Appendix B Program APCS: Analytical Plotter Control System

This Appendix contains a description and listing of the program code, written by the Author, for controlling the analytical plotter. The bulk of the code is written in FORTRAN, with some supporting routines written in assembly language. In addition, APCS calls routines contained within program MDLT (Modified Direct Linear Transform, described in Appendix C) in order to solve the photogrammetric problem.

APCS contains all of the code required to firstly initialise the analytical plotter, secondly measure photograph coordinates of control points and hence derive the photogrammetric model, and thirdly coordinate general points on the stereophotographs, determine their object space coordinates and write this data to a file. During this latter phase it automatically controls the positioning of the photographs such that a stereomodel free from y-parallax is perceived by the operator.

The program operates using a menu system, and does not permit selection of operations before they can be performed (e.g. photograph moves require initialisation to be completed, determination of object space coordinates can only be performed after entry of control points, and so on). It is therefore completely interactive in use, presenting the operator with instructions as and when required.

This Appendix consists of three Sections. The first describes the structure of APCS in terms of subroutine calls, the second presents a listing of the main FORTRAN code, and the third contains a listing of the assembly language routines.

B.1 Structure of APCS

In order to produce a system that is straightforward to use, APCS has been written such that a series of menus is presented to the user. Each of the menu choices represents a particular subroutine, with the result that the subroutines

are arranged in a classic tree-like pattern. The subroutines close to the trunk of the tree do no more than prompt the user for a menu selection; those at the ends of the branches perform the actual work.

The complexity of APCS can be judged from the fact that it consists of 27 FORTRAN subroutines, 2 FORTRAN functions and 14 assembly language subroutines. In order to display the structure of the program, a tabular format has been adopted for the tree of subroutine calls, and this is illustrated Tables B.01 and B.02. The first of these is concerned with the initialisation of the instrument, and the second with the measurement of photograph coordinates. A brief description of the function of the main subroutines is now given.

Table B.01 Structure of program APCS: sheet 1.

APCS	**CLS**				
	MENU	**CLRBLK**[3]			
		SETCUR[3]			
		CHAT[3]			
		USERIN	**JSTICK**		
			KCODE		
		WAIT[2]			
	INIT	MENU...			
		JOYINI	MENU...		
			MSGBOX[2]	**SETCUR**[2]	
			USERIN[2]...		
			BEEP		
			SETCUR[2]		
			WAIT[2]		
			CLRBLK[2]		
		CONINI	MENU...		
			CHGPAR	MENU...	
				CLRBLK	
				MSGBOX...	
				GETPAR	**SETCUR**[2]
					KCODE
			DEVINI	MSGBOX[2]...	
				BYTOUT[20]	
				WAIT[2]	
			CLRBLK[2]		
			MSGBOX...		
			USERIN...		
			WAIT		
		CMPINI	MENU...		
			CLRBLK[2]		
			LIMITS	MENU...	
				CLRBLK	
			ALIGN	MENU...	
				CLRBLK[2]	
				BYTOUT[7]	
				MSGBOX...	
				KCODE	
				SETCUR	
contd. on next sheet		**CLRBLK**			

A subroutine name followed by an ellipsis (i.e. ...) means that a complete definition of the subroutine is given earlier in the table.
A number in brackets (i.e. [6]) represents the number of calls made to that subroutine by the calling subroutine.
Names in **BOLD** represent assembly language subroutines, names in *ITALIC* represent subroutines within program MDLT.

Table B.02 Structure of program APCS: sheet 2.

APCS *(contd.)*	MOVEP	MENU...					
		CLRBLK[2]					
		MCOORD	MENU...				
			CLRBLK				
			MCNTRL	MENU...			
				CLRBLK			
				MSGBOX...			
				USERIN...			
				SYNCH	*YPARX*[2]		
					XVEL[3]		
					YVEL[3]		
					INTVEL[4]	VELCTY	**BYTIN**[2]
							B2INT4
							INT4B3
							BYTOUT
						BYTOUT	
					MSGBOX[4]...		
					POSCTL[6]	POSITN[2]	**BYTIN**[6]
							B3INT4[2]
							INT4B3...
							BYTOUT[3]
						BYTOUT	
					POSITN[10]...		
					USERIN[4]...		
					VELCTY[8]...		
					BEEP[3]		
					CLRBLK		
					SETCTL	MSGBOX...	
						SETCUR[2]	
						CLRBLK	
						BEEP	
					WAIT		
			MOBJCT	MENU...			
				CLRBLK[3]			
				MSGBOX[3]...			
				PHOTO			
				CTLCHK			
				USERIN[2]...			
				SYNCH...	OBJCRD	MSGBOX[2]...	
						SETCUR[2]	
						CLRBLK	
						BEEP	
						OBJECT	
						USERIN...	
		INTVEL[2]...					
		MSGBOX...					
		USERIN...					
		YVEL...					
		XVEL...					
		VELCTY[4]...					
		POSCTL[2]...					
		BEEP					
	QUIT	MENU...					
		BYTOUT					
		WAIT[2]					
		SETCUR[2]					
		CLS					
		CLRBLK					

A subroutine name followed by an ellipsis (i.e. ...) means that a complete definition of the subroutine is given earlier in the table.
A number in brackets (i.e. [6]) represents the number of calls made to that subroutine by the calling subroutine.
Names in **BOLD** represent assembly language subroutines, names in *ITALIC* represent subroutines within program MDLT.

MENU The fundamental subroutine that handles writing to the screen and analysing the user's response.

INIT Controls initialisation of the joystick (in terms of calibrating its movement, through a call to JOYINI), the closed-loop motion controller (by programming the HCTL-1000 controller circuits, through a call to CONINI), and the movement of the analytical plotter (by determining the limits of movement, and allowing adjustment of the rotary encoders relative to the motors, through a call to CMPINI).

MOVEP The principal subroutine for controlling movement of the photographs, both during the phase when control points are being coordinated (through subroutine MCNTRL), and when general object space coordinates are being determined (through subroutine MOBJCT).

MOBJCT Controls measurement of object space coordinates, including calls to the MDLT subroutines PHOTO and CTLCHK which are responsible, respectively, for computation of the DLT parameters and a check of the fit of the photogrammetric model to the control point data.

SYNCH The subroutine that controls all photograph movement when both photographs are moving simultaneously. This subroutine continually reads command velocity information from the joystick (subroutine JSTICK), determines both the motor velocities for the left hand photograph (subroutine VELCTY) and the required position for the right hand photograph (subroutine POSITN) to maintain the stereomodel, and writes these commands to the motion controllers (subroutine BYTOUT).

SETCTL Called by SYNCH whilst the coordination of control points is underway (i.e. SYNCH has been called from MCNTRL), to record both the photograph coordinates and (*via* keyboard input from the user) the object space coordinates of the control points.

OBJCRD Called by SYNCH whilst the determination of general object space coordinates is underway (i.e. SYNCH has been called from MOBJCT). OBJCRD both records the photograph coordinates and computes the corresponding object space coordinates of a point (through a call to the MDLT subroutine OBJECT) and then writes this information, together with a user-supplied numeric identifier, to a file.

B.2 Listing of program APCS

A complete listing of APCS is given here. Although APCS has been written in FORTRAN 77, some common extensions to the language that have been included by the manufacturer of the compiler have been used, particularly where they either simplify the code or allow for greater efficiency in execution. The extensions used in APCS are:

(i) the data types `LOGICAL*1`, `INTEGER*1` and `INTEGER*2` have been used, where appropriate, to both conserve memory and simplify interfacing with assembly language procedures;

(ii) `DO WHILE....END DO` loops have been used where a sequence of instructions is to be executed continuously until input from the user is received;

(iii) `INCLUDE` files, referenced in the listing of APCS as, for example, `INCLUDE 'STATUS.BLK'`, and which are automatically inserted in the code at compile time, have been used for all extensively used `COMMON` declarations in order to reduce the bulk of the code;

(iv) comments, beginning with an exclamation mark, following and on the same line as, program statements.

The complete set of `INCLUDE` files required for APCS follows.

```
C Include File: STATUS.BLK
      LOGICAL*1
     +     STKOK,CNTOK,CMPOK,CTLOK,DLTOK
      COMMON/STATUS/
     +     STKOK,CNTOK,CMPOK,CTLOK,DLTOK

C Include File IODAT.BLK
      INTEGER
     +     IDEV,ODEV
      COMMON/CONFIG/
     +     IDEV,ODEV

C Include File CTLDAT.BLK
      INTEGER
     +     NPAR,NPHOT,NUMCTL,MAXCTL,CTLID
      REAL
     +     CTLCRD,STDERR,CPCRD,PCVAR,SCALE
      DIMENSION
     +     CTLID(24),CTLCRD(3,24),STDERR(3),CPCRD(2,24,2),
     +     PCVAR(2,2)
      COMMON
     +     /CTLDAT/NPAR,NUMCTL,NPHOT,MAXCTL,CTLID,CTLCRD,STDERR,CPCRD,
     +             PCVAR,SCALE
```

```
C Include File: HCTL.BLK
      INTEGER*2
     +    BADD,SMPLTR,COMRNG,COMVTR,COMMX,COMMY,OFFSET,MAXADV,
     +    FTZERO,FTPOLE,FTGAIN,ACCLN
      DIMENSION
     +    OFFSET(4)
      COMMON
     +    /HCTL/BADD,SMPLTR,COMRNG,COMVTR,COMMX,COMMY,OFFSET,MAXADV,
     +          FTZERO,FTPOLE,FTGAIN,ACCLN

C Include File: BOXDAT.BLK
      CHARACTER
     +    TOPLIN*80,MIDLIN*80,BOTLIN*80
      COMMON
     +    /BOX/TOPLIN,MIDLIN,BOTLIN
```

The code for APCS now follows. Each procedure (subroutine or function) is separated from adjacent procedures by a row of asterisks across the page. The procedures are given in row-major order corresponding to Tables B.01 and B.02, i.e. APCS, then MENU, then USERIN, then JSTICK and so on.

```
       PROGRAM APCS
C
C     Analytical Plotter Control System
C
C Type Declarations: Common Blocks
      INCLUDE 'STATUS.BLK'
      INCLUDE 'IODAT.BLK'
      INCLUDE 'CTLDAT.BLK'
      INCLUDE 'HCTL.BLK'
      INCLUDE 'BOXDAT.BLK'
C Type Declarations: Local Variables
      INTEGER*2
     +    MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE,L1,L2
      LOGICAL*1
     +    ALWAYS,DRAWN
      CHARACTER
     +    MNUTXT*60,CHCTXT*25,BANNER*80,B1*1,B2*1,B4*1,B5*1,
     +    TCHAR*1,MCHAR*1,BCHAR*1
      DIMENSION
     +    MNUTXT(2),CHCTXT(3),BANNER(5),B1(80),B2(80),B4(80),B5(80),
     +    TCHAR(80),MCHAR(80),BCHAR(80)
      EQUIVALENCE
     +    (TOPLIN,TCHAR), (MIDLIN,MCHAR), (BOTLIN,BCHAR),
     +    (BANNER(1),B1(1)), (BANNER(2),B2(1)),
     +    (BANNER(4),B4(1)), (BANNER(5),B5(1))
C Data Initialization
      DATA                                          ! variables in BOXDAT.BLK
     +    TCHAR/'+', 78*'-', '+'/,
     +    MCHAR/'|', 78*' ', '|'/,
     +    BCHAR/'+', 78*'-', '+'/
      DATA                                          ! variables in HCTL.BLK
     +    BADD/#300/, SMPLTR/50/, COMRNG/40/, COMVTR/100/,
     +    COMMX/10/, COMMY/0/, OFFSET(1)/4/, OFFSET(2)/9/,
     +    OFFSET(3)/-1/, OFFSET(4)/7/, MAXADV/10/, FTZERO/170/,
     +    FTPOLE/120/, FTGAIN/190/, ACCLN/1280/
      DATA                                          ! variables in IODAT.BLK
     +    IDEV/7/,ODEV/8/
      DATA                                          ! variables in CTLDAT.BLK
     +    CORRCT/1/,NPHOT/2/,MAXCTL/24/,SCALE/1574.803/,
                                          ! SCALE converts quad counts to mm
     +    (STDERR(L1),L1=1,3)/3*0.001/,
     +    ((PCVAR(L2,L1),L1=1,2),L2=1,2)/4*1.905E-3/
      DATA                                          ! variables in STATUS.BLK
     +    STKOK/.FALSE./,CNTOK/.FALSE./,CMPOK/.FALSE./,DLTOK/.FALSE./
      DATA                                          ! local variables
     +    B1/'#', 78*'=', '#'/,
```

```
     +    B2/'"', 78*' ', '"'/,
     +    B4/'"', 78*' ', '"'/,
     +    B5/'#', 78*'=', '#'/,
     +    BANNER(3)/'"     A N A L Y T I C A L   P L O T T E R   C O N
     +T R O L   P R O G R A M     "'/,
     +    ALWAYS/.TRUE./,MNUROW/6/,TXTNUM/2/,MAXCHC/3/,CHCWID/25/,
     +    MNUTXT(1)/
     +'    Use button on joystick base to cycle through functions, '/,
     +    MNUTXT(2)/
     +'        button on joystick to select.                       '/,
     +    CHCTXT(1)/' Initialize          '/,
     +    CHCTXT(2)/' Move photos         '/,
     +    CHCTXT(3)/' Quit                '/
C Executable Code
      NPAR = 11 + MOD(CORRCT,10) + 2*INT(CORRCT/10)
      OPEN(UNIT=ODEV,
     +     ACCESS='APPEND',FILE='STERCOM.TXT',FORM='FORMATTED',
     +     MODE='WRITE',SHARE='DENYWR',STATUS='UNKNOWN',BLOCKSIZE=512)
      CALL CLS
      WRITE(ODEV,*)'DLT results from analytical plotter control program'
      WRITE(*,'(1X,5A\)') (BANNER(L1),L1=1,5)
      DRAWN=.FALSE.
      CHOICE=3
      DO WHILE (ALWAYS)
          CALL MENU
     +       (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
          DRAWN=.TRUE.
          IF (CHOICE.EQ.1)THEN
              CALL INIT
          ELSE IF (CHOICE.EQ.2) THEN
              CALL MOVEP
          ELSE
              CALL QUIT
          ENDIF
      END DO
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE MENU
     +    (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
C
C     write MENU on screen:
C         writes box, enters menu titles, toggles menu choices, returns
C         with choice number
C     Input:
C         MNUROW = row on which menu box is to start
C         MNUTXT = text entered into menu box
C         TXTNUM = number of rows of text
C         MAXCHC = maximum no. of choices allowed
C         CHCWID = width of each choice box
C         CHCTXT = character array of choice text
C         DRAWN  = .TRUE. means this menu is already on the screen
C     Return:
C         CHOICE = number of choice
C
C Type Declarations: Formal Arguments
      INTEGER*2
     +    MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE
      CHARACTER
     +    MNUTXT*(*),CHCTXT*(*)
      LOGICAL*1
     +    DRAWN
      DIMENSION
     +    MNUTXT(TXTNUM),CHCTXT(MAXCHC)
C Type Declarations: Common Blocks
      INCLUDE 'BOXDAT.BLK'
C Type Declarations: Local Variables
      INTEGER*2
     +    CHCROW,TXTROW,COL,L1,XSTICK,YSTICK
      LOGICAL*1
     +    ALWAYS,SELECT,NXTCHC,RSPNSE
      PARAMETER
```

```
     +    (ALWAYS=.TRUE.)
C Statement Function: define start column of choice-text blocks
      COL(CHOICE)=INT2(CHCWID*(CHOICE-1)+3)
C Executable Code
      TXTROW=MNUROW+1
      CHCROW=MNUROW+TXTNUM+2
      CALL CLRBLK(INT2(MNUROW+TXTNUM+5),1,25,80) ! clear from this menu down
      IF (.NOT.DRAWN) THEN
          CALL CLRBLK(MNUROW,1,25,80)            ! clear from previous menu down
          CALL SETCUR(MNUROW,1)          ! write menu box
          WRITE(*,'(10(A\))') TOPLIN, (MIDLIN, L1=1,TXTNUM+3),BOTLIN
          DO 1000 L1=1,TXTNUM
              CALL SETCUR(INT2(MNUROW+L1),3)
              WRITE(*,'(\A\)') MNUTXT(L1)          ! write menu text
 1000     CONTINUE
          CALL SETCUR(CHCROW,3)                  ! write choice text
          WRITE(*,'(20(A\))') (CHCTXT(L1), L1=1,MAXCHC)
          CALL CHAT(CHCROW,COL(CHOICE),CHCWID,3)
      END IF
      DO WHILE(ALWAYS)
          RSPNSE=.FALSE.
          DO WHILE(.NOT.RSPNSE)
              CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
          END DO
          IF (NXTCHC) THEN                                  ! new choice
              CALL CHAT(CHCROW,COL(CHOICE),CHCWID,1)        ! screen to normal
              CHOICE=1+MOD(CHOICE,MAXCHC)
              CALL CHAT(CHCROW,COL(CHOICE),CHCWID,3)        ! screen to reverse
          ELSE IF (SELECT) THEN                             ! choice selected
              CALL CLRBLK(INT2(MNUROW+TXTNUM+5),1,25,80)
              CALL WAIT(3)
              RETURN
          ENDIF
          CALL WAIT(3)
      END DO
      END
* * * *
```

```
      LOGICAL*1
     +    ALWAYS,DRAWN
      CHARACTER
     + MNUTXT*24,CHCTXT*19
      DIMENSION
     +    MNUTXT(1),CHCTXT(4)
C Executable Code
      ALWAYS=.TRUE.
      MNUROW=13
      TXTNUM=1
      MAXCHC=4
      CHCWID=19
      MNUTXT(1)=' Initialization choices:'
      CHCTXT(1)=' Joystick          '
      CHCTXT(2)=' Controller        '
      CHCTXT(3)=' Stereocomparator  '
      CHCTXT(4)=' Previous menu     '
      DRAWN=.FALSE.
      CHOICE=4
      DO WHILE (ALWAYS)
          CALL MENU
     +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
          IF (CHOICE.EQ.1)THEN
              CALL JOYINI
          ELSE IF (CHOICE.EQ.2) THEN
              CALL CONINI
          ELSE IF (CHOICE.EQ.3) THEN
              CALL CMPINI
          ELSE
              CALL CLRBLK(MNUROW,1,24,80)
              RETURN
          ENDIF
      END DO
      END
* * * *
```

```
     +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
           DRAWN=.TRUE.
           IF (CHOICE.EQ.1)THEN                    ! initialize joystick
               DO 1000 L1=0,4
                   IF(L1.EQ.0)THEN
                       STKMSG = ' Move stick to centre position'//
     +                           ' and press a button...'
                   ELSE
                       WRITE(STKMSG,'(A,I3,A)')
     +                 ' Move stick to ',L1*3,
     +                 ' o''clock and press a button...'
                   ENDIF
                   CALL MSGBOX(STKMSG,1)
                   RSPNSE = .FALSE.
                   DO WHILE(.NOT.RSPNSE)
                       CALL USERIN
     +                     (XPOS(L1),YPOS(L1),SELECT,NXTCHC,RSPNSE)
                   END DO
                   CALL BEEP
                   CALL SETCUR(23,60)
                   WRITE(*,'(\''Wait...''\)')
                   CALL WAIT(5)
                   CALL SETCUR(23,60)
                   WRITE(*,'(\''        ''\)')
    1000           CONTINUE
                   XMAX=XPOS(1)
                   XMIN=XPOS(3)
                   XCEN=XPOS(0)
                   YMAX=YPOS(4)
                   YMIN=YPOS(2)
                   YCEN=YPOS(0)
                   STKMSG ='Joystick initialized. '//
     +                     'Strike any key/button to continue...'
                   CALL MSGBOX(STKMSG,1)
                   RSPNSE=.FALSE.
                   DO WHILE(.NOT.RSPNSE)
                       CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
                   END DO
                   CALL WAIT(5)
                   CALL CLRBLK(22,1,24,80)
                   STKOK=.TRUE.
               ELSE
                   CALL CLRBLK(MNUROW,1,25,80)
                   RETURN
               ENDIF
           END DO
           END
* * * *
```

```
      DO 2000 L1=1,TXTNUM
          CALL SETCUR(INT2(23-TXTNUM+L1),3)
          WRITE(*,'(\A\)') TEXT(L1)
 2000 CONTINUE
      RETURN
      END
* * * *
```

```
     +    MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE,NEWVAR,MAXVAL,MINVAL
      LOGICAL*1
     +    ALWAYS,DRAWN
      CHARACTER
     +    MNUTXT*71,CHCTXT*5,PARTXT*20,MSGTXT*60
      DIMENSION
     +    MNUTXT(3),CHCTXT(15)
      PARAMETER
     +    (ALWAYS=.TRUE.)
      DATA
     +    MNUTXT(1)/'|Main|  Commutator  |      Offsets      |Max.|Digit
     +al Filter|Max.|Vel.|'/,
     +    MNUTXT(2)/'|Tmr.|Ring  X    Y  |  1    2    3    4 |Adv.|Zero
     +Pole Gain|Accn|Tmr.|'/,
     +    MNUTXT(3)/'+----+----+----+----+----+----+----+----+----+----+
     +----+----+----+----+'/,
     +    CHCTXT(15)/' done '/
C Executable Code
      DRAWN=.FALSE.
      CHCWID=5
      MAXCHC= 15
      CHOICE=15
      MNUROW=13
      TXTNUM=3
      DO WHILE (ALWAYS)
          WRITE(CHCTXT( 1),100) SMPLTR
          WRITE(CHCTXT( 2),100) COMRNG
          WRITE(CHCTXT( 3),100) COMMX
          WRITE(CHCTXT( 4),100) COMMY
          WRITE(CHCTXT( 5),100) OFFSET(1)
          WRITE(CHCTXT( 6),100) OFFSET(2)
          WRITE(CHCTXT( 7),100) OFFSET(3)
          WRITE(CHCTXT( 8),100) OFFSET(4)
          WRITE(CHCTXT( 9),100) MAXADV
          WRITE(CHCTXT(10),100) FTZERO
          WRITE(CHCTXT(11),100) FTPOLE
          WRITE(CHCTXT(12),100) FTGAIN
          WRITE(CHCTXT(13),100) ACCLN
          WRITE(CHCTXT(14),100) COMVTR
  100     FORMAT(I5)
          CALL MENU
     +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
          IF (CHOICE.EQ.15) THEN              ! finished
              CALL CLRBLK(MNUROW,1,24,80)
              RETURN
          ELSE
              IF (CHOICE.EQ.1) THEN                   ! sample timer
                  PARTXT='sample timer:'
                  NEWVAR= SMPLTR
                  MAXVAL= 255
                  MINVAL= 0
              ELSEIF (CHOICE.EQ.2) THEN               ! ring
                  PARTXT='commutator ring:'
                  NEWVAR= COMRNG
                  MAXVAL= 127
                  MINVAL= 0
              ELSEIF (CHOICE.EQ.3) THEN               ! x
                  PARTXT='commutator X:'
                  NEWVAR= COMMX
                  MAXVAL= 127
                  MINVAL= 0
              ELSEIF (CHOICE.EQ.4) THEN               ! y
                  PARTXT='commutator Y:'
                  NEWVAR= COMMY
                  MAXVAL= 127
                  MINVAL= 0
              ELSEIF (CHOICE.EQ.5) THEN               ! offset 1
                  PARTXT='motor 1 offset:'
                  NEWVAR= OFFSET(1)
                  MAXVAL=  127
                  MINVAL= -128
```

```
            ELSEIF (CHOICE.EQ.6) THEN                ! offset 2
                PARTXT='motor 2 offset:'
                NEWVAR= OFFSET(2)
                MAXVAL=  127
                MINVAL= -128
            ELSEIF (CHOICE.EQ.7) THEN                ! offset 3
                PARTXT='motor 3 offset:'
                NEWVAR= OFFSET(3)
                MAXVAL=  127
                MINVAL= -128
            ELSEIF (CHOICE.EQ.8) THEN                ! offset 4
                PARTXT='motor 4 offset:'
                NEWVAR= OFFSET(4)
                MAXVAL=  127
                MINVAL= -128
            ELSEIF (CHOICE.EQ. 9) THEN               ! max advance
                PARTXT='maximum advance:'
                NEWVAR= MAXADV
                MAXVAL= 127
                MINVAL= 0
            ELSEIF (CHOICE.EQ.10) THEN               ! zero
                PARTXT='filter zero:'
                NEWVAR= FTZERO
                MAXVAL= 255
                MINVAL= 0
            ELSEIF (CHOICE.EQ.11) THEN               ! pole
                PARTXT='filter pole:'
                NEWVAR= FTPOLE
                MAXVAL= 255
                MINVAL= 0
            ELSEIF (CHOICE.EQ.12) THEN               ! gain
                PARTXT='filter gain:'
                NEWVAR= FTGAIN
                MAXVAL= 255
                MINVAL= 0
            ELSEIF (CHOICE.EQ.13) THEN               ! max accln
                PARTXT='maximum accleration:'
                NEWVAR= ACCLN
                MAXVAL= 32767
                MINVAL= 0
            ELSEIF (CHOICE.EQ.14) THEN                ! velocity timer
                PARTXT='velocity timer:'
                NEWVAR= COMVTR
                MAXVAL= 255
                MINVAL= 0
            ENDIF
            MSGTXT = ' Use _/_ to change '//PARTXT
            CALL MSGBOX(MSGTXT,1)
            CALL GETPAR(NEWVAR,MAXVAL,MINVAL)
            IF (CHOICE.EQ. 1) SMPLTR=   NEWVAR
            IF (CHOICE.EQ. 2) COMRNG=   NEWVAR
            IF (CHOICE.EQ. 3) COMMX =   NEWVAR
            IF (CHOICE.EQ. 4) COMMY =   NEWVAR
            IF (CHOICE.EQ. 5) OFFSET(1)=NEWVAR
            IF (CHOICE.EQ. 6) OFFSET(2)=NEWVAR
            IF (CHOICE.EQ. 7) OFFSET(3)=NEWVAR
            IF (CHOICE.EQ. 8) OFFSET(4)=NEWVAR
            IF (CHOICE.EQ. 9) MAXADV=   NEWVAR
            IF (CHOICE.EQ.10) FTZERO=   NEWVAR
            IF (CHOICE.EQ.11) FTPOLE=   NEWVAR
            IF (CHOICE.EQ.12) FTGAIN=   NEWVAR
            IF (CHOICE.EQ.13) ACCLN =   NEWVAR
            IF (CHOICE.EQ.14) COMVTR=   NEWVAR
        ENDIF
    ENDDO
    END
* * * *
```

```
C Type Declarations: Formal Arguments
      INTEGER*2
     +    NEWVAR,MAXVAL,MINVAL
C Type Declarations: Local Variables
      INTEGER*2
     +    TEMP,CODE
      LOGICAL*1
     +    ALWAYS
      PARAMETER
     +    (ALWAYS=.TRUE.)
C Executable Code
      TEMP= NEWVAR
      CALL SETCUR(23,70)
      WRITE(*,'(\I5\)')TEMP
      DO WHILE (ALWAYS)
          CODE= 0
          DO WHILE (CODE.EQ.0)
              CALL KCODE(CODE)
          ENDDO
          IF ((CODE.EQ.-72) .OR. (CODE.EQ.-80)) THEN
              TEMP = TEMP + SIGN(1,76+CODE)
              TEMP= MAX(MINVAL,TEMP)
              TEMP= MIN(MAXVAL,TEMP)
              CALL SETCUR(23,70)
              WRITE(*,'(\I5\)')TEMP
          ELSEIF (CODE.EQ.13) THEN
              NEWVAR= TEMP
              RETURN
          ELSEIF (CODE.EQ.27) THEN
              RETURN
          ENDIF
      ENDDO
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE DEVINI
C
C     DEVice INItialisation
C
C Type Declarations: Common Blocks
      INCLUDE 'HCTL.BLK'
C Type Declarations: Local Variables
      REAL
     +    ALNTME
      INTEGER*1
     +    CNUM,SMPTR1,CMRNG1,CMVTR1,COMMX1,COMMY1,OFFST1,
     +    MXADV1,FTZRO1,FTPLE1,FLTGN1,ACCLN1
      LOGICAL*1
     +    ALWAYS,CLRLIM
      CHARACTER
     +    MSGTXT*60
      DIMENSION
     +    OFFST1(8),ACCLN1(2)
      EQUIVALENCE
     +    (SMPLTR,SMPTR1),(COMRNG,CMRNG1),(COMVTR,CMVTR1),
     +    (COMMX,COMMX1),(COMMY,COMMY1),(OFFSET(1),OFFST1(1)),
     +    (MAXADV,MXADV1),(FTZERO,FTZRO1),(FTPOLE,FTPLE1),
     +    (FTGAIN,FLTGN1),(ACCLN, ACCLN1)
      PARAMETER
     +    (ALWAYS=.TRUE.)
      DATA
     +    MSGTXT/'Initializing channel'/
C Executable Code
      CLRLIM = .FALSE.
      DO 1000 CNUM=1,4
          WRITE(MSGTXT(22:23),'(I2)') CNUM
          CALL MSGBOX(MSGTXT,1)
          CALL BYTOUT(#00,BADD,#05,CNUM,CLRLIM)
          CALL WAIT(10)
 1000 CONTINUE
      DO 2000 CNUM=1,4                    ! program all devices
          CALL BYTOUT(SMPTR1,BADD,#0F,CNUM,CLRLIM)     ! sample timer=150
```

```
          CALL BYTOUT(#02,BADD,#07,CNUM,CLRLIM)     ! quad, 4ph, pwm inhibit off
          CALL BYTOUT(CMRNG1,BADD,#18,CNUM,CLRLIM)     ! ring=40
          CALL BYTOUT(COMMX1,BADD,#1A,CNUM,CLRLIM)     ! x=10
          CALL BYTOUT(COMMY1,BADD,#1B,CNUM,CLRLIM)     ! y=0
          CALL BYTOUT(OFFST1(2*CNUM-1),BADD,#1C,CNUM,CLRLIM)     ! offset
          CALL BYTOUT(FLTGN1,BADD,#22,CNUM,CLRLIM)     ! gain
          CALL BYTOUT(FTPLE1,BADD,#21,CNUM,CLRLIM)     ! pole
          CALL BYTOUT(FTZRO1,BADD,#20,CNUM,CLRLIM)     ! zero
          CALL BYTOUT(CMVTR1,BADD,#19,CNUM,CLRLIM)     ! comm. vel. timer
          CALL BYTOUT(MXADV1,BADD,#1F,CNUM,CLRLIM)     ! max advance
          CALL BYTOUT(#00,BADD,#3C,CNUM,CLRLIM)     ! command velocity
          CALL BYTOUT(#00,BADD,#0C,CNUM,CLRLIM)     ! command pos MSB
          CALL BYTOUT(#00,BADD,#0D,CNUM,CLRLIM)     ! command pos ISB
          CALL BYTOUT(#00,BADD,#0E,CNUM,CLRLIM)     ! command pos LSB
          CALL BYTOUT(ACCLN1(1),BADD,#26,CNUM,CLRLIM)     ! accln LSB=00h
          CALL BYTOUT(ACCLN1(2),BADD,#27,CNUM,CLRLIM)     ! accln MSB=05h
          CALL BYTOUT(#00,BADD,#00,CNUM,CLRLIM)     ! position control
          CALL BYTOUT(#02,BADD,#05,CNUM,CLRLIM)     ! align
 2000 CONTINUE
      CALL MSGBOX('Waiting for align to complete',1)
      ALNTME = REAL(4096*16*(SMPLTR+1))/1.0E6     ! align time (sec)
      CALL WAIT(INT2(ALNTME*10.0))                ! 1/10ths sec wait
      RETURN
      END
* * * *
```

```
     +          (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
           IF (CHOICE.EQ.MAXCHC)THEN
               CALL CLRBLK(MNUROW,1,24,80)
               RETURN
           ELSE IF (CHOICE.EQ.1) THEN
               CALL LIMITS
           ELSE IF (CHOICE.EQ.2) THEN
               CALL ALIGN
           ELSE IF (CHOICE.EQ.3) THEN
               CONTINUE
           ELSE
               CALL CLRBLK(MNUROW,1,24,80)
               RETURN
           ENDIF
       END DO
       END
* * * *
```

```
C                       direction= -1 or +1
                        SPEED=MAXVEL*DIR
                        CALL SETCUR(24,5)
                        WRITE(*,'(\A,I3\)')' Working on',DIR*L1
                        CALL VELCTY(SPEED,CNUM,VELWRT)
                        LIMOK=.TRUE.
                        CODE=0
                        DO WHILE ( LIMOK )
                            CALL BYTIN(N,BADD,#07,CNUM,CLRLIM)
                            N=N.AND.2#10000000
                            IF(N.EQ.0)LIMOK=.FALSE.
                            CALL KCODE(CODE)
                            IF (CODE.NE.0) THEN
                                SPEED=0
                                DO 3 L3=1,4
                                    CALL VELCTY(SPEED,L3,VELWRT)
                                    CALL BYTOUT(#03,BADD,#05,L3,CLRLIM)
    3                           CONTINUE
                                RETURN
                            END IF
                        END DO
C                       limit has been reached. Read position:
                        CALL BYTIN(C,BADD,#14,CNUM,CLRLIM)
                        CALL BYTIN(B,BADD,#13,CNUM,CLRLIM)
                        CALL BYTIN(A,BADD,#12,CNUM,CLRLIM)
                        CALL B3INT4(LIMIT(L2,L1),A,B,C)
C                       write to sample timer register with CLRLIM asserted
C                       to over-ride limit switch
                        CALL BYTOUT(INT1(255),BADD,#0F,CNUM,TOGGLE)
C                       write to all status and program counter registers
C                       to reset controllers
                        DO 4 L3=1,4
                            CALL BYTOUT(#00,BADD,#07,L3,CLRLIM)
                            CALL BYTOUT(#03,BADD,#05,L3,CLRLIM)
    4                   CONTINUE
C                       set speed for reverse direction
                        CALL VELCTY(-SPEED,CNUM,VELWRT)
                        CALL BYTOUT(#0B,BADD,#00,CNUM,CLRLIM)
                        CALL BYTOUT(#03,BADD,#05,CNUM,CLRLIM)
                        CALL WAIT(20)
C                       stop movement after 2 seconds
                        CALL VELCTY(INT4(0),CNUM,VELWRT)
C                       write to sample timer register with CLRLIM asserted
C                       to reset limit switch
                        CALL BYTOUT(INT1(255),BADD,#0F,CNUM,TOGGLE)
    2               CONTINUE
    1           CONTINUE
                CALL CLREND(24,5)
                CALL SETCUR(22,5)
                DO 5 L1=1,2
                    WRITE(*,'(4I15)') (LIMIT(L1,L2), L2=1,4)
    5           CONTINUE
                PAUSE
C                CALL SETCUR(22,1)
C                WRITE(*,'(\3A\)') TOPLIN,MIDLIN,BOTLIN
            ELSE
                CALL CLRBLK(MNUROW,1,24,80)
                RETURN
            ENDIF
        END DO
        END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE ALIGN
C
C     energise last motor phases to allow alignment of encoders
C
C Type Declarations: Common Blocks
      INCLUDE 'STATUS.BLK'
C Type Declarations: Local Variables
      INTEGER*1
     +    CNUM,OFFSET
```

```
      INTEGER*2
     +     MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE,CODE,
     +     BADD
      LOGICAL*1
     +     ALWAYS,CLRLIM,DRAWN,TOGGLE,VELWRT
      CHARACTER
     +     MNUTXT*60,CHCTXT*24,TEXT*60
      DIMENSION
     +     MNUTXT(4),CHCTXT(3)
C Executable Code
      BADD=#300                    !
      CLRLIM=.FALSE.               !
      VELWRT=.FALSE.               !
      TOGGLE=.TRUE.                !
      ALWAYS=.TRUE.
      MNUROW=13
      CHCWID=24
      TXTNUM=4
      MNUTXT(1)=
     +'Align motor/encoder assemblies:                                '
      MNUTXT(2)=
     +'The last phase of all motors will be energised, so that        '
      MNUTXT(3)=
     +'the encoders may be aligned such that their index signals      '
      MNUTXT(4)=
     +'are active.                                                    '
      MAXCHC=2
      CHCTXT(1)=' Perform alignment      '
      CHCTXT(2)=' Previous menu          '
      TEXT='Windings energised. Use ↑/↓ to change offset, '//
     +     'any other key to quit...'
      DRAWN=.FALSE.
      CHOICE=MAXCHC
      DO WHILE (ALWAYS)
          CALL MENU
     +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
          DRAWN=.TRUE.
          IF (CHOICE.EQ.MAXCHC)THEN
              CALL CLRBLK(MNUROW,1,24,80)
              RETURN
          ELSE                                    ! align...
              DO 1000 CNUM=1,4
                  CALL BYTOUT(#01,BADD,#05,CNUM,CLRLIM)     ! idle
                  CALL BYTOUT(#64,BADD,#09,CNUM,CLRLIM)     ! max pwm
                  CALL BYTOUT(#0C,BADD,#00,CNUM,CLRLIM)     ! hold commutator
                  CALL BYTOUT(#00,BADD,#1C,CNUM,CLRLIM)     ! offset=00
1000          CONTINUE
              CALL MSGBOX(TEXT,1)
              CODE=0
              OFFSET=0
              DO WHILE (CODE.EQ.0)
                  CALL KCODE(CODE)
                  IF (CODE.EQ.-72 .OR. CODE.EQ.-80) THEN
                      OFFSET = OFFSET + (CODE+76)/4
                      CALL SETCUR(23,76)
                      WRITE(*,'(\I3\)') OFFSET
                      DO 2000 CNUM=1,4
                          CALL BYTOUT(OFFSET,BADD,#1C,CNUM,CLRLIM)
2000                  CONTINUE
                      CODE= 0
                  ENDIF
              END DO
              DO 3000 CNUM=1,4
                  CALL BYTOUT(#00,BADD,#09,CNUM,CLRLIM)     ! clear pwm
                  CALL BYTOUT(#03,BADD,#05,CNUM,CLRLIM)     ! enter control mode
3000          CONTINUE
              CALL CLRBLK(22,1,24,80)
          ENDIF
      END DO
      END
* * * *
```

```
      SUBROUTINE MOVEP
C
C     MOVE Photographs
C
C Type Declarations: Common Blocks
      INCLUDE 'STATUS.BLK'
C Type Declarations: Local Variables
      INTEGER*1
     +    XAXIS,YAXIS,SPDSGN
      INTEGER*2
     +    MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE,XSTICK,YSTICK
      INTEGER*4
     +    XVEL,YVEL,XSPEED,YSPEED
      LOGICAL*1
     +    ALWAYS,DRAWN,CLRLIM,VELWRT,SELECT,NXTCHC,RSPNSE
      CHARACTER
     + MNUTXT*70,CHCTXT*18,TEXT*76
      DIMENSION
     +    MNUTXT(3),CHCTXT(4),TEXT(2)
      PARAMETER
     +    (VELWRT=.FALSE., ALWAYS=.TRUE.)
      DATA
     +    TEXT(1)/'Press a button or strike any key to terminate'/,
     +    TEXT(2)/'movement and return to menu...'/
C Executable Code
      CLRLIM=.FALSE.
      MNUROW=13
      CHCWID=18
      MNUTXT(1)=
     +'Move photographs:                                          '
      IF (STKOK.AND.CNTOK) THEN                  ! joystick & controller OK
          CHCTXT(1)=' Left photo only  '
          CHCTXT(2)=' Right photo only '
          CHCTXT(3)=' Measure coords   '
          CHCTXT(4)=' Previous menu    '
          TXTNUM=1
          MAXCHC=4
      ELSE
          CHCTXT(1)=' Previous menu '
          TXTNUM=3
          MAXCHC=1
          IF (STKOK) THEN                        ! only joystick initialised
              MNUTXT(2)=
     +        'This cannot be performed until the controller has been'
              MNUTXT(3)=
     +        'initialized.'
          ELSE IF (CNTOK) THEN                   ! only controller initialised
              MNUTXT(2)=
     +        'This cannot be performed until the joystick has been'
              MNUTXT(3)=
     +        'initialized.'
          ELSE                                   ! nothing initialised
              MNUTXT(2)=
     +        'This cannot be performed until both the joystick and the'
              MNUTXT(3)=
     +        'controller have been initialized.'
          ENDIF
      ENDIF
      DRAWN=.FALSE.                              ! menu not yet one screen
      CHOICE=MAXCHC
      DO WHILE (ALWAYS)
          CALL MENU
     +        (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
          DRAWN=.TRUE.                           ! menu now on screen
          IF (CHOICE.EQ.MAXCHC)THEN
              CALL CLRBLK(MNUROW,1,24,80)
              RETURN
          ELSE IF (CHOICE.EQ.3) THEN             ! move synchronised
              CALL MCOORD
              DRAWN = .FALSE.
          ELSE                                   ! move L or R
```

```
                XAXIS = 3-CHOICE                    ! x = 2 (L) or 1 (R)
                YAXIS = XAXIS+2                     ! y = 4 (L) or 3 (R)
                CALL INTVEL(XAXIS)
                CALL INTVEL(YAXIS)
                IF (XAXIS.EQ.1) THEN
                    SPDSGN= -1
                ELSE
                    SPDSGN= 1
                ENDIF
                CALL MSGBOX(TEXT,2)
                RSPNSE = .FALSE.
                DO WHILE(.NOT.RSPNSE)
                    CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
                    YSPEED=YVEL(YSTICK)
                    XSPEED= SPDSGN*XVEL(XSTICK)
                    CALL VELCTY(XSPEED,XAXIS,VELWRT)
                    CALL VELCTY(YSPEED,YAXIS,VELWRT)
                END DO                              ! move finished
                CALL VELCTY(0,XAXIS,VELWRT)         ! velocity = 0
                CALL VELCTY(0,YAXIS,VELWRT)
                CALL POSCTL(XAXIS)                  ! switch to position control
                CALL POSCTL(YAXIS)
                CALL BEEP
                CALL CLRBLK(INT2(MNUROW+TXTNUM+5),1,25,80)
            ENDIF
        END DO
        END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        SUBROUTINE MCOORD
C
C       Measure COORDinates
C
C Type Declarations: Local Variables
        INTEGER*2
       +    MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE
        LOGICAL*1
       +    ALWAYS,DRAWN
        CHARACTER
       +    MNUTXT*35,CHCTXT*20
        DIMENSION
       +    MNUTXT(1),CHCTXT(3)
        PARAMETER
       +    (ALWAYS=.TRUE.)
        DATA
       +    MNUTXT(1)/' Measure coordinates:'/,
       +    CHCTXT(1)/'   Control points   '/,
       +    CHCTXT(2)/'    Object points   '/,
       +    CHCTXT(3)/'    Previous menu   '/
C Executable Code
        MNUROW=13
        TXTNUM=1
        MAXCHC=3
        CHCWID=20
        DO WHILE (ALWAYS)
            DRAWN=.FALSE.
            CHOICE=3
            CALL MENU
      +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,CHOICE)
            IF (CHOICE.EQ.MAXCHC) THEN              ! quit
                CALL CLRBLK(MNUROW,1,25,80)
                RETURN
            ELSE IF (CHOICE.EQ.1) THEN              ! enter control points
                CALL MCNTRL
            ELSEIF (CHOICE.EQ.2) THEN               ! measure object points
                CALL MOBJCT
            ENDIF
        END DO
        END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        SUBROUTINE MCNTRL
C Type Declarations: Common Blocks
```

```
      INCLUDE 'STATUS.BLK'
      INCLUDE 'CTLDAT.BLK'
C Type Declarations: Local Variables
      INTEGER*2
     +    MNUROW,TXTNUM,MAXCHC,CHCWID,CHOICE,XSTICK,YSTICK
      LOGICAL*1
     +    ALWAYS,DRAWN,SELECT,NXTCHC,RSPNSE,MSRCTL
      CHARACTER
     +    MNUTXT*15,CHCTXT*20,OVRCHK*76
      DIMENSION
     +    MNUTXT(1),CHCTXT(3),OVRCHK(2)
      PARAMETER
     +    (ALWAYS=.TRUE.)
      DATA
     +    MNUTXT(1)/'Control Points:'/,
     +    CHCTXT(1)/'New series of points'/,
     +    CHCTXT(2)/'  Add more points   '/,
     +    CHCTXT(3)/'   Previous menu    '/,
     +    OVRCHK(1)/'Restarting will ERASE all previously entered contro
     +l points.'/,
     +    OVRCHK(2)/'Press stick button or Enter to do this, any other k
     +ey to return to menu...'/
C Executable Code
      MNUROW=13
      TXTNUM=1
      MAXCHC=3
      CHCWID=20
      DRAWN=.FALSE.
      CHOICE=3
      DO WHILE (ALWAYS)
          CALL MENU
     +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,
     +          CHOICE)
          IF (CHOICE.EQ.MAXCHC) THEN
              CALL CLRBLK(MNUROW,1,25,80)
              RETURN
          ELSE
              MSRCTL = .FALSE.
              IF (CHOICE.EQ.1) THEN                ! start afresh...
                  CALL MSGBOX(OVRCHK,2)
                  RSPNSE = .FALSE.
                  DO WHILE(.NOT.RSPNSE)
                      CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
                  ENDDO
                  IF (SELECT) THEN                    ! OK to start afresh
                      MSRCTL=.TRUE.
                      CTLOK = .FALSE.
                      DLTOK = .FALSE.
                      NUMCTL= 0
                      NUMOBJ= 0
                  ENDIF
              ELSEIF (CHOICE.EQ.2) THEN                ! add to control points
                  DLTOK = .FALSE.
                  MSRCTL = .TRUE.
              ENDIF
              IF (MSRCTL) CALL SYNCH(MSRCTL)
          ENDIF
      ENDDO
      RETURN
      END
* * * *
```

```
C Type Declarations: Local Variables
      INTEGER*1
     +    L1
      INTEGER*2
     +    XSTICK,YSTICK
      INTEGER
     +    XPOSL,YPOSL,XPOSR,YPOSR,XPARX,YDIFF,
     +    XSPEED,YSPEED,XVEL,YVEL,PCOORD,ACTPOS,COMPOS,POSERR,THRESH
      REAL
     +    YPARX,XPL,XPR,YPL
      LOGICAL*1
     +    ALWAYS,POSRD,POSWRT,VELRD,VELWRT,SELECT,NXTCHC,RSPNSE,CRDFLG,
     +    OK,ENGAGE
      CHARACTER
     +    TEXTS*70,TEXTF*70,TEXTC*70,ERRTXT*70,
     +    NTEXT*70,DTEXT*70,ETEXT*70
      DIMENSION
     +    TEXTS(3),TEXTF(3),TEXTC(3),PCOORD(4)
      PARAMETER
     +    (ALWAYS=.TRUE.,
     +    POSRD=.TRUE., POSWRT=.FALSE., VELRD=.TRUE., VELWRT=.FALSE.)
      EXTERNAL
     +    YPARX,XVEL,YVEL
C Data Initialization
      DATA
     +    CRDFLG/.FALSE./,THRESH/10/,
     +    TEXTS(1)/
     +    'Coordinate measurement:'/,
     +    TEXTS(2)/
     +    'Use base button/any key to return to previous menu,'/,
     +    TEXTS(3)/
     +    '    stick button to adjust height of floating spot...'/,
     +    TEXTF(1)/
     +    ' Floating  ¿       stick button to record coordinates,'/,
     +    NTEXT/
     +    '   spot    Ã  use: base button or'/,
     +    DTEXT/
     +    '   spot    Ã  use: base button to DISENGAGE y-parallax contro
     +l,'/,
     +    ETEXT/
     +    '   spot    Ã  use: base button to ENGAGE y-parallax control,'
     +/,
     +    TEXTF(3)/
     +    'adjustment Ù       any key for photograph moves...'/,
     +    TEXTC(1)/
     +    'Coords taken, waiting to return to photograph moves,'/,
     +    TEXTC(2)/
     +    'Use base button/any key to DISCARD these coordinates,'/,
     +    TEXTC(3)/
     +    '    stick button to SAVE these coordinates...'/,
     +    ERRTXT/
     +    'Channel x is out of tolerance: correct and strike a key to co
     +ntinue...'/
C Executable Code
      CALL INTVEL(2)                              ! switch channels 2 & 4 to
      CALL INTVEL(4)                              ! integral velocity mode
      DO WHILE (ALWAYS)
          CALL MSGBOX(TEXTS,3)                    ! sync. moves message
          CALL POSCTL(1)                          ! switch channels 1 & 3 to
          CALL POSCTL(3)                          ! position control
          CALL POSITN(COMPOS,XPOSR,1,POSRD)
          CALL POSITN(COMPOS,XPOSL,2,POSRD)
          CALL POSITN(COMPOS,YPOSR,3,POSRD)
          CALL POSITN(COMPOS,YPOSL,4,POSRD)
          XPARX = XPOSL+XPOSR                     ! current x-parallax
          YDIFF = YPOSL-YPOSR                     ! current y-parallax
          RSPNSE = .FALSE.
          DO WHILE(.NOT.RSPNSE)
              CALL POSITN(COMPOS,XPOSL,2,POSRD)   !
              CALL POSITN(COMPOS,YPOSL,4,POSRD)   ! read xL, yL, xR
              CALL POSITN(COMPOS,XPOSR,1,POSRD)   !
```

```
    IF (DLTOK) THEN
        XPL =  XPOSL/SCALE
        YPL =  YPOSL/SCALE
        XPR = -XPOSR/SCALE              ! sign change
        YPOSR = YPARX(XPL,YPL,XPR,CRDFLG)*SCALE  ! yR to remove y-parallax
    ELSE
        YPOSR = YPOSL-YDIFF             ! keep previous y-parallax
    ENDIF
    XPOSR = XPARX-XPOSL                 ! xR to maintain x-parallax
    CALL POSITN(XPOSR,ACTPOS,1,POSWRT)
    CALL POSITN(YPOSR,ACTPOS,3,POSWRT)
    CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)! read com. vel.
    XSPEED = XVEL(XSTICK)
    YSPEED = YVEL(YSTICK)
    CALL VELCTY(XSPEED,2,VELWRT)
    CALL VELCTY(YSPEED,4,VELWRT)
END DO
CALL VELCTY(0,2,VELWRT)                 ! bring L photo to rest
CALL VELCTY(0,4,VELWRT)                 !
CALL BEEP
IF (SELECT) THEN                        ! adjust spot height
    CALL POSITN(COMPOS,XPOSL,2,POSRD)   !
    CALL POSITN(COMPOS,YPOSL,4,POSRD)   ! read xL & yL
    XPL =  XPOSL/SCALE
    YPL =  YPOSL/SCALE
    CALL INTVEL(1)                      ! switch R photo to
    IF (.NOT.DLTOK) THEN
        CALL INTVEL(3)                  ! integral velocity mode
        TEXTF(2) = NTEXT                ! y-parallax fixed
        ENGAGE = .FALSE.
    ELSE
        CALL POSCTL(3)                  ! position control
        TEXTF(2) = DTEXT                ! y-parallax engaged
        ENGAGE = .TRUE.
    ENDIF
    CALL MSGBOX(TEXTF,3)
    OK = .TRUE.
    DO WHILE (OK)
        CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)! read comvel
        IF (RSPNSE) THEN
            IF (DLTOK) THEN
                IF (NXTCHC) THEN
                    ENGAGE = .NOT.ENGAGE    ! toggle
                    IF (ENGAGE) THEN
                        CALL POSCTL(3)      ! position control
                        TEXTF(2) = DTEXT    ! use DTEXT
                    ELSE
                        CALL INTVEL(3)      ! integral velocity mode
                        TEXTF(2) = ETEXT    ! use ETEXT
                    ENDIF
                    CALL MSGBOX(TEXTF,3)    ! new message box
                ELSE
                    OK = .FALSE.            ! prepare to quit
                ENDIF
            ELSE
                OK = .FALSE.                ! prepare to quit
            ENDIF
        ENDIF
        XSPEED = -XVEL(XSTICK)
        YSPEED = YVEL(YSTICK)
        CALL VELCTY(XSPEED,1,VELWRT)
        CALL POSITN(COMPOS,XPOSR,1,POSRD)   ! read xR
        IF (ENGAGE) THEN
            XPR = -XPOSR/SCALE              ! sign change
            YPOSR = YPARX(XPL,YPL,XPR,CRDFLG)*SCALE
                                            ! yR to remove y-parallax
            CALL POSITN(YPOSR,ACTPOS,3,POSWRT)
        ELSE
            CALL VELCTY(YSPEED,3,VELWRT)
        ENDIF
    END DO
```

```
            CALL VELCTY(0,1,VELWRT)                    ! ensure R photo is at rest
            IF (.NOT.DLTOK) CALL VELCTY(0,3,VELWRT) !
            CALL BEEP
            IF (SELECT) THEN                          ! coords are reqd
                DO 3000 L1=1,4
                    POSERR = 2*THRESH
                    DO WHILE (POSERR.GT.THRESH)
                        CALL POSITN(COMPOS,ACTPOS,L1,POSRD)
                        POSERR = ABS(ACTPOS)-ABS(COMPOS)
                        IF (POSERR.LE.THRESH) THEN
                            PCOORD(L1)=ACTPOS ! record photocoord
                            EXIT
                        ELSE
                            WRITE(ERRTXT(8:9),'(I2)') L1
                            CALL BEEP
                            CALL MSGBOX(ERRTXT,1)
                            RSPNSE = .FALSE.
                            DO WHILE (.NOT.RSPNSE)
                                CALL USERIN(XSTICK,YSTICK,SELECT,
     +                                                    NXTCHC,RSPNSE)
                            END DO
                        ENDIF
                    END DO
 3000               CONTINUE
                    CALL MSGBOX(TEXTC,3)
                    RSPNSE = .FALSE.
                    DO WHILE (.NOT.RSPNSE)
                        CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
                    END DO
                    IF (SELECT) THEN
                        CALL CLRBLK(20,1,25,80)
                        IF (MSRCTL) THEN
                            CALL SETCTL(PCOORD)
                        ELSE
                            CALL OBJCRD(PCOORD)
                        ENDIF
                    ENDIF
                ENDIF                                 ! coordinates finished
            ELSE                                      ! moves finished
                CALL POSCTL(1)                        ! switch all channels to
                CALL POSCTL(2)                        ! position control
                CALL POSCTL(3)                        !
                CALL POSCTL(4)                        !
                RETURN                                ! moves finished
            ENDIF
            CALL WAIT(5)
        END DO
        END
* * * *
```

```
      END
* * * *
```

```
     +    A,B,C
      LOGICAL*1
     +    CLRLIM
C Executable Code
      CLRLIM=.FALSE.
      IF (RDWRT) THEN                              ! read the velocity
          CALL BYTIN(B,BADD,#35,CNUM,CLRLIM)
          CALL BYTIN(C,BADD,#34,CNUM,CLRLIM)
          CALL B2INT4(VEL,B,C)
      ELSE                                         ! write the velocity
          CALL INT4B3(VEL,A,B,C)
          CALL BYTOUT(C,BADD,#3C,CNUM,CLRLIM)
      ENDIF
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE B2INT4(N,MSB,LSB)
C
C     Convert 2x8bit integers to 32bit integer
C
C      Input:
C              MSB = most significant byte
C              LSB = least significant byte
C      Output:
C               N= integer
C
C Type Declarations: Formal Arguments
      INTEGER*1
     +    MSB,LSB
      INTEGER*4
     +    N
C Type Declarations: Local Variables
      INTEGER*1
     +    I1
      INTEGER*2
     +    I2
      DIMENSION
     +    I1(2)
      EQUIVALENCE (I2,I1(1))
C Executable Code
      I1(2)=MSB
      I1(1)=LSB
      N= INT4(I2)
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE INT4B3(N,A,B,C)
C
C     Convert 32bit integer to 3x8bit integers
C
C      Input:
C               N=signed integer
C      Output:
C             A = most significant bit
C             B = intermediate significant bit
C             C = least significant bit
C
C Type Declarations: Formal Arguments
      INTEGER*1
     +    A,B,C
      INTEGER*4
     +    N
C Type Declarations: Local Variables
      INTEGER*1
     +    I1
      INTEGER*4
     +    I4
      DIMENSION
     +    I1(4)
      EQUIVALENCE (I4,I1(1))
C Executable Code
```

```
      I4=N
      A=I1(3)
      B=I1(2)
      C=I1(1)
      RETURN
      END
* * * *
```

```
            CALL INT4B3(COMPOS,A,B,C)
            CALL BYTOUT(A,BADD,#0C,CNUM,CLRLIM)
            CALL BYTOUT(B,BADD,#0D,CNUM,CLRLIM)
            CALL BYTOUT(C,BADD,#0E,CNUM,CLRLIM)
         ENDIF
         RETURN
         END
* * * *
```

```
      WRITE(TEXTI1(45:46),'(I2)') NUMCTL
      IOS= 1
      DO WHILE (IOS.NE.0)
          CALL MSGBOX(TEXTI,2)
          CALL SETCUR(22,65)
          READ(*,'(A)') CRDTXT
          CALL SETCUR (23,1)
          WRITE(*,'(\'' ''\)')
          IF (CRDTXT(1:1).EQ.'X' .OR. CRDTXT(1:1).EQ.'x') THEN
              NUMCTL= NUMCTL - 1
              CALL CLRBLK(21,1,25,80)
              RETURN
          ENDIF
          READ(CRDTXT,'(I6)',IOSTAT=IOS) CTLID(NUMCTL)
          IF (IOS.NE.0) CALL BEEP
      END DO
      DO 1000 L1=1,3
          IOS= 1
          WRITE(TEXTC1(7:7),'(A1)') CNAME(L1)
          WRITE(TEXTC1(43:44),'(I2)') CTLID(NUMCTL)
          DO WHILE (IOS.NE.0)
              CALL MSGBOX(TEXTC,2)
              CALL SETCUR(22,65)
              READ(*,'(A)') CRDTXT
              CALL SETCUR (23,1)
              WRITE(*,'(\'' ''\)')
              IF (CRDTXT(1:1).EQ.'X' .OR. CRDTXT(1:1).EQ.'x') THEN
                  NUMCTL= NUMCTL - 1
                  CALL CLRBLK(21,1,25,80)
                  RETURN
              ENDIF
              READ(CRDTXT,'(F10.3)',IOSTAT=IOS) CTLCRD(L1,NUMCTL)
              IF (IOS.NE.0) CALL BEEP
          END DO
 1000 CONTINUE
      CPCRD(1,NUMCTL,1) =  REAL(PHTCRD(2))/SCALE
      CPCRD(2,NUMCTL,1) =  REAL(PHTCRD(4))/SCALE
      CPCRD(1,NUMCTL,2) = -REAL(PHTCRD(1))/SCALE     ! sign change
      CPCRD(2,NUMCTL,2) =  REAL(PHTCRD(3))/SCALE
      NEQNS= NUMCTL*2
      IF (NEQNS.GT.NPAR) THEN
          CTLOK = .TRUE.
      ENDIF
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE OBJCRD(PHTCRD)
C
C     OBJect CooRDinates
C
C Type Declarations: Formal Arguments
      INTEGER
     +    PHTCRD
      DIMENSION
     +    PHTCRD(4)
C Type Declarations: Common Blocks
      INCLUDE 'CTLDAT.BLK'
      INCLUDE 'IODAT.BLK'
C Type Declarations: Local Variables
      REAL
     +    PCOORD,OBJTSC,VRNCE
      INTEGER*2
     +    IOS,OBJID,XSTICK,YSTICK,L1
      LOGICAL*1
     +    SELECT,NXTCHC,RSPNSE
      CHARACTER
     +    CRDTXT*10,TEXTC*60,TEXTC1*60,TEXTO*72,TEXTO2*72
      DIMENSION
     +    PCOORD(2,2),OBJTSC(3),VRNCE(4),TEXTC(2),TEXTO(3)
      EQUIVALENCE
     +    (TEXTC1,TEXTC), (TEXTO2,TEXTO(2))
```

```
      DATA
     +    TEXTC(1)/'Enter numeric identifier for observed point **:'/,
     +    TEXTC(2)/'(format required is iiiiii, but X quits)'/,
     +    TEXTO(1)/'Object point  ID      x         y         z      var
     +(x)  var(y)  var(z)'/,
     +    TEXTO(3)/'Stick button/Enter SAVES to file, anything else DISC
     +ARDS...'/
C Executable Code
      NUMOBJ= NUMOBJ+1
      WRITE(TEXTC1(45:46),'(I2)') NUMOBJ
      IOS= 1
      DO WHILE (IOS.NE.0)
          CALL MSGBOX(TEXTC,2)
          CALL SETCUR(22,65)
          READ(*,'(A)') CRDTXT
          CALL SETCUR (23,1)
          WRITE(*,'(\'' ''\)')
          IF (CRDTXT(1:1).EQ.'X' .OR. CRDTXT(1:1).EQ.'x') THEN
              NUMOBJ= NUMOBJ - 1
              CALL CLRBLK(21,1,25,80)
              RETURN
          ENDIF
          READ(CRDTXT,'(I6)',IOSTAT=IOS) OBJID
          IF (IOS.NE.0) CALL BEEP
      END DO
      PCOORD(1,1) =  REAL(PHTCRD(2))/SCALE
      PCOORD(2,1) =  REAL(PHTCRD(4))/SCALE
      PCOORD(1,2) = -REAL(PHTCRD(1))/SCALE      ! sign change
      PCOORD(2,2) =  REAL(PHTCRD(3))/SCALE
      CALL OBJECT(PCOORD,NPHOT,OBJTSC,VRNCE)
      WRITE(TEXTO2,100) NUMOBJ, OBJID, (OBJTSC(L1),L1=1,3),
     +                  (VRNCE(L1),L1=1,3)
  100 FORMAT(I8,I9,3F10.3,3F8.4)
      CALL MSGBOX(TEXTO,3)
      RSPNSE = .FALSE.
      DO WHILE (.NOT.RSPNSE)
          CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
      ENDDO
      IF (SELECT) THEN
          WRITE(ODEV,'(A)') TEXTO2
      ENDIF
      RETURN
      END
* * * *
```

```
     +    CHCTXT(3)/'        Test       '/,
     +    CHCTXT(4)/'   Previous menu   '/,
     +    CTLERR(1)/'Insufficient control points have been entered.'/,
     +    CTLERR(2)/'Press any key/button to continue...'/
     +    DLTERR(1)/'DLT parameters have not yet been computed.'/,
     +    DLTERR(2)/'Press any key/button to continue...'/,
     +    DLTMSG/'Computing DLT parameters. This will take a while - ple
     +ase wait...'/
      DATA
     +    (TEST1(I),I=1,8)/15,27,18,28,16, 3,13,20/,
     +    ((TEST2(I,J),I=1,3),J=1,8)/
     +         99.431,  60.016, 105.016,   102.112,  60.460, 104.597,
     +        104.894,  61.448, 103.544,   102.110,  60.864, 100.249,
     +         99.541,  60.217, 102.255,    97.197,  61.090, 106.937,
     +         99.228,  59.640, 110.160,   105.135,  59.119, 111.138/,
     +    (TEST3(1,I,1),I=1,8)/
     +        .247, 32.532, 70.253, 29.703,
     +      -2.303,-29.298,  4.255, 70.441/,
     +    (TEST3(2,I,1),I=1,8)/
     +       -.359, -2.319,-11.286,-62.788,
     +     -34.761, 26.339, 52.949, 62.664/,
     +    (TEST3(1,I,2),I=1,8)/
     +       -.007, 35.684, 80.313, 43.080,
     +       3.181,-30.658, -6.172, 59.128/,
     +    (TEST3(2,I,2),I=1,8)/
     +        .048, -2.003,-11.036,-62.070,
     +     -35.030, 26.469, 54.787, 65.179/
C Executable Code
      MNUROW=13
      TXTNUM=1
      MAXCHC=4
      CHCWID=19
      DRAWN=.FALSE.
      CHOICE=MAXCHC
      DO WHILE (ALWAYS)
          CALL MENU
     +         (MNUROW,MNUTXT,TXTNUM,MAXCHC,CHCWID,CHCTXT,DRAWN,
     +          CHOICE)
          IF (CHOICE.EQ.MAXCHC) THEN              ! quit
              CALL CLRBLK(MNUROW,1,25,80)
              RETURN
          ELSE IF (CHOICE.EQ.1) THEN              ! compute DLT parameters
              IF (CTLOK) THEN
                  CALL MSGBOX(DLTMSG,1)
                  WRITE(ODEV,*)'Computation of DLT parameters...'
                  WRITE(ODEV,100) NPAR,NUMCTL,(STDERR(I),I=1,3)
  100             FORMAT(/I4,I11,3F11.3)
                  DO J=1,NUMCTL
                      WRITE(ODEV,200) CTLID(J),(CTLCRD(I,J),I=1,3),
     +                                ((CPCRD(I,J,K),I=1,2),K=1,2)
  200                 FORMAT(I4,7F11.3)
                  END DO
                  CALL PHOTO
     +                    (CORRCT,NUMCTL,NPHOT,CTLID,CTLCRD,STDERR,
     +                                         CPCRD,PCVAR,MAXCTL,ODEV)
                  CALL CTLCHK
                  WRITE(ODEV,*)'New DLT values in use from hereon.'
                  DLTOK = .TRUE.
              ELSE
                  CALL MSGBOX(CTLERR,2)           ! insufficient control
                  RSPNSE = .FALSE..
                  DO WHILE (.NOT.RSPNSE)
                      CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
                  ENDDO
                  CALL CLRBLK(MNUROW,1,25,80)
                  RETURN
              ENDIF
          ELSEIF (CHOICE.EQ.2) THEN               ! measure object points
              IF (DLTOK) THEN
                  MSRCTL = .FALSE.
                  CALL SYNCH(MSRCTL)
```

```
            ELSE
                CALL MSGBOX(DLTERR,2)          ! DLT parameters not computed
                RSPNSE = .FALSE.
                DO WHILE (.NOT.RSPNSE)
                    CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
                ENDDO
                CALL CLRBLK(MNUROW,1,25,80)
            ENDIF
        ELSEIF (CHOICE.EQ.3) THEN              ! write test control data
            CTLOK = .TRUE.
            NUMCTL= 8
            DO J=1,8
                CTLID(J)= TEST1(J)
                DO I=1,3
                    CTLCRD(I,J) = TEST2(I,J)
                ENDDO
                DO K=1,2
                    DO I=1,2
                        CPCRD(I,J,K) = TEST3(I,J,K)
                    ENDDO
                ENDDO
            ENDDO
            CALL MSGBOX('Test data written.',1)
            RSPNSE = .FALSE.
            DO WHILE (.NOT.RSPNSE)
                CALL USERIN(XSTICK,YSTICK,SELECT,NXTCHC,RSPNSE)
            ENDDO
            CALL CLRBLK(MNUROW,1,25,80)
        ENDIF
    ENDDO
    RETURN
    END
* * * *
```

```
      CALL SETCUR(1,1)
      CALL CLS
      STOP' '
   ELSE
      CALL CLRBLK(MNUROW,1,24,80)
      RETURN
   ENDIF
   END
```

B.3 Supporting Assembly Language Routines

Although the main control program, APCS, is written in FORTRAN, there are a number of supporting routines that, because they need to access either the operating system or the machine input/output directly, have been written in assembly language. These routines may be grouped as follows:

(i) Keyboard handling routines, that extend the keyboard input provided by FORTRAN;

(ii) Routines for allowing joystick input;

(iii) Screen writing routines, that allow the cursor to be set to specific positions, clear all or part of the screen, and change the display attributes of areas of the screen;

(iv) Input/output routines for the motion control interface board, including changing the ISA-bus speed; and

(v) Timing and sound generation routines.

These routines have all been written in 'standard' Intel-8086 assembly language [116], and will therefore compile with any assembler written for this device.

Because these routines are all required to interface to FORTRAN, it has been necessary to include both a prologue and an epilogue in each routine such that the state of the processor is saved, and then subsequently reinstated, before and after the code in the routine is executed. These sections of code are specific to the particular FORTRAN compiler adopted, and so have been identified by comments in the routines. The compiler used in this work was Microsoft FORTRAN 5.1; other compilers will probably require different code.

The code follows, in the order of the numbered points above, with the boundaries between routines indicated by a line of asterisks across the page.

```
;+------------------------------------------------------------------------+
;|                                                                        |
;| KCODE:  return a Key CODE                                              |
;| Usage:  CALL KCODE(CODE). Parameter is INTEGER*2                       |
;|                           CODE= ASCII of key pressed if positive,      |
;|                                 extended code of key pressed if negative
;|                                                                        |
;+------------------------------------------------------------------------+
    TITLE     keycodes.asm:  Intercept keyboard scan codes
    NAME      keycodes
KEYCODE_TEXT  SEGMENT  BYTE PUBLIC 'CODE'
KEYCODE_TEXT  ENDS
_DATA         SEGMENT  WORD PUBLIC 'DATA'
_DATA         ENDS
CONST         SEGMENT  WORD PUBLIC 'CONST'
CONST         ENDS
_BSS          SEGMENT  WORD PUBLIC 'BSS'
_BSS          ENDS
DGROUP   GROUP     CONST,     _BSS,      _DATA
    ASSUME    CS: KEYCODE_TEXT,  DS: DGROUP,     SS: DGROUP,     ES: DGROUP
_DATA         SEGMENT             ;
```

```
_DATA           ENDS                    ;
_BSS            SEGMENT                 ;
_BSS            ENDS                    ;
KEYCODE_TEXT    SEGMENT                 ;
    PUBLIC      KCODE                   ; declare all labels as PUBLIC for
                                        ; easier debugging
KCODE           PROC FAR
    push        bp                      ; save calling program's BP
    mov         bp,sp                   ; set base pointer to stack pointer
    mov         ah,0BH                  ;
    int         21H                     ; Check for a keystroke
    cmp         al,0FFH                 ; check the returned status
    je          KEY_WAITING             ; if al=FFh, key is waiting
NO_KEY:
    xor         ax,ax                   ; clear ax
    jmp         FINISH
KEY_WAITING:
    mov         ah,08H                  ;
    int         21H                     ; read the character
    cmp         al,00                   ; is al=0?
    je          EXCODE                  ; if so, extended code has been read
    xor         ah,ah                   ; otherwise clear ah
    jmp         FINISH
EXCODE:
    int         21H                     ; read the extended code
    xor         ah,ah                   ; clear ah
    neg         ax                      ; and make the code negative
FINISH:
    les         bx,DWORD PTR [bp+06d];
    mov         WORD PTR es:[bx],ax  ; load stack with parameter
    pop         bp                      ; restore calling program's BP
    ret         04d                     ; clear parameters from stack
KCODE           ENDP                    ;
KEYCODE_TEXT    ENDS                    ; code segment finished
END
;* * * *
```

```
    out     dx,al                   ; send arbitrary byte to port
    mov     di,0000                 ; initialize counters in di and si
    mov     si,0000                 ;
LOOP01:
    in        al,dx                 ; read the port
    test      al,01                 ; if al=1 then bit 0 is equal to 1
    je        JUMP01                ; if so, skip the counter
    inc       di                    ; increment x-axis counter
JUMP01:
    test      al,02                 ; if al=2 then bit 1 is equal to 1
    je        JUMP02                ; if so, skip the counter
    inc       si                    ; increment y-axis counter
JUMP02:
    test      al,03                 ; if al=3 then bit 0 and bit 1 equal 1
    jne       LOOP01                ; if not true, do loop again
    mov       cl,04                 ;
    shr       al,cl                 ; shift 4 bits right: put button data in al
    and       ax,0003               ; zero all bits other than 0 and 1
    les       bx,DWORD PTR [bp+0Eh]     ; 1st parameter:
    mov       WORD PTR es:[bx],di       ;     x-axis
    les       bx,DWORD PTR [bp+0Ah]     ; 2nd parameter:
    mov       WORD PTR es:[bx],si       ;     y-axis
    les       bx,DWORD PTR [bp+06h]     ; 3rd parameter:
    mov       WORD PTR es:[bx],ax       ;     button data
    pop       si                    ;
    pop       di                    ;
    mov       sp,bp                 ; remove local variable space
    pop       bp                    ; restore calling program's bp
    ret       0CH                   ; clear parameters from stack:
                                    ;     calculated as (4*n) for n parameters
JSTICK    ENDP                      ;
JSTICK_TEXT  ENDS                   ; code segment finished
END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;+-------------------------------------------------------------------------------+
;| CLS: CLear Screen.                                                            |
;| Usage: CALL CLS   (no parameters)                                             |
;|                                                                               |
;| SETCUR: SET CURsor to absolute position                                       |
;| Usage:  CALL SETCUR(ROW,COL). Both parameters are INTEGER*2                   |
;|                                                                               |
;| CLREND: CLeaR to END of line                                                  |
;| Usage:  CALL CLREND(ROW,COL). Both parameters are INTEGER*2                   |
;|                                                                               |
;| CLRBLK  CLeaR BLocK                                                           |
;| Usage:  CALL CLRBLK(BEGROW,BEGCOL,ENDROW,ENDCOL).                             |
;|                                  All parameters are INTEGER*2                 |
;|                                                                               |
;| CHAT:   CHange ATtribute                                                      |
;| Usage:  CALL CHAT(ROW,COL,CHARS,CODE). All parameters are INTEGER*2           |
;|                                        CHARS= no. of chars to operate on      |
;|                                        CODE= 1= normal                        |
;|                                              2= intense                       |
;|                                              3= reverse                       |
;|                                              4= blinking normal               |
;|                                              5= blinking intense              |
;|                                              6= blinking reverse              |
;+-------------------------------------------------------------------------------+
    TITLE     SCREEN.ASM: Various screen manipulation subroutines
    NAME      SCREEN
SCREEN_TEXT   SEGMENT  BYTE PUBLIC 'CODE'
SCREEN_TEXT   ENDS
_DATA         SEGMENT  WORD PUBLIC 'DATA'
_DATA         ENDS
CONST         SEGMENT  WORD PUBLIC 'CONST'
CONST         ENDS
_BSS          SEGMENT  WORD PUBLIC 'BSS'
_BSS          ENDS
DGROUP   GROUP     CONST,     _BSS,      _DATA
    ASSUME    CS: SCREEN_TEXT,    DS: DGROUP,      SS: DGROUP,     ES: DGROUP
_DATA         SEGMENT             ; All initialized local data resides in _DATA
```

```
stk_siz     dw   0020H          ; local stack will be 32bytes long
ATTRIBS     db   007H           ; 1: normal
            db   00FH           ; 2: intense
            db   070H           ; 3: reverse
            db   087H           ; 4: blinking normal
            db   08FH           ; 5: blinking intense
            db   0F0H           ; 6: blinking reverse
_DATA       ENDS                ;
_BSS        SEGMENT             ;
    old_sp  dw 01H DUP (?)      ;
    row     dw 01H DUP (?)      ;
    col     dw 01H DUP (?)      ;
    chars   dw 01H DUP (?)      ;
    attrib  dw 01H DUP (?)      ;
    old_cur dw 01h dup (?)      ;
    begrow  equ  row            ; redefine the CHAT parameter names for
    begcol  equ  col            ; use in CLRBLK
    endrow  equ  chars          ;
    endcol  equ  attrib         ;
_BSS        ENDS                ;
SCREEN_TEXT SEGMENT             ;
    PUBLIC  CLS,SETCUR,CLREND,CHAT,CLRBLK ; declare all labels as PUBLIC
                                          ; for easier debugging
CLS         PROC FAR
    mov     ax,0600H            ; ah=scroll ax=clear screen
    mov     bh,07H              ; attribute for a fill
    xor     cx,cx               ; coords of top left
    mov     dx,1950H            ; coords of bottom right
    int     10H                 ; clear the screen
    xor     bh,bh               ; set to page 0 (mono screen)
    xor     dx,dx               ; coords of top left
    mov     ah,02H              ; function to set cursor
    int     10H                 ; set the cursor
    ret
CLS     ENDP
SCREEN  PROC    FAR
SETCUR  LABEL   FAR
    mov     dx,00H              ; code for SETCUR is 0
    jmp     MAIN
CLREND  LABEL   FAR
    mov     dx,01H              ; code for CLREND is +1
    jmp     MAIN
CHAT    LABEL   FAR
    mov     dx,02H              ; code for CHAT is +2
    jmp     MAIN
CLRBLK  LABEL   FAR
    mov     dx,03H              ; code for CLRBLK is +3
MAIN:
    push    bp                  ; |
    mov     bp,sp               ; |
    mov     old_sp,sp           ; | MS-Fortran standard prologue
    sub     sp,stk_siz          ; |
    push    si                  ; |
    push    di                  ; |
    mov     si,dx               ; now put subroutine code into si
    mov     di,00H              ; additional offset of 1st and 2nd params
                                ;  when only 2 params are being loaded
    cmp     si,1                ; If we are executing CLREND or SETCUR,
    jle     LOAD_2              ;  then load only two params
LOAD_4:                         ; If we are executing CHAT or CLRBLK.
    mov     di,08D              ; additional offset of 1st and 2nd params
                                ;  when 4 params are being loaded
    les     bx,DWORD PTR [bp+06d]; load es:bx with address of attrib/endcol
    mov     ax,WORD PTR es:[bx]  ; load reg with parameter:
    mov     attrib,ax
    les     bx,DWORD PTR [bp+10d]; load es:bx with address of chars/endrow
    mov     ax,WORD PTR es:[bx]  ; load reg with parameter:
    mov     chars,ax
LOAD_2:
    les     bx,DWORD PTR [bp+di+06d]; load es:bx with address of col/begcol
    mov     ax,WORD PTR es:[bx]     ; load reg with parameter:
```

```
    mov     col,ax
    les     bx,DWORD PTR [bp+di+10d]; load es:bx with address of row/begrow
    mov     ax,WORD PTR es:[bx]     ; load reg with parameter:
    mov     row,ax
    cmp     si,3                    ; are we using clrblk?
    jne     DET_MODE                ; if not, jump to det_mode
    mov     ax,begrow               ; otherwise ensure begrow is within range
    mov     cx,025D
    call    MODULO
    mov     al,dl
    cbw
    inc     ax
    mov     begrow,ax
    mov     ax,begcol               ; ensure begcol is within range
    mov     cx,080D
    call    MODULO
    mov     al,dl
    cbw
    inc     ax
    mov     begcol,ax
    mov     ax,endrow               ; ensure endrow is within range
    mov     cx,025D
    call    MODULO
    mov     al,dl
    cbw
    inc     ax
    mov     endrow,ax
    mov     ax,endcol               ; ensure endcol is within range
    mov     cx,080D
    call    MODULO
    mov     al,dl
    cbw
    inc     ax
    mov     endcol,ax
ROW_CHG:
    mov     ax,begrow
    mov     bx,endrow
    cmp     ax,bx
    jle     COL_CHG
    xchg    ax,bx
    mov     endrow,ax
    mov     begrow,bx
COL_CHG:
    mov     ax,begcol
    mov     bx,endcol
    cmp     ax,bx
    jle     DET_MODE
    xchg    ax,bx
    mov     endcol,ax
    mov     begcol,bx
DET_MODE:
    mov     ah,0FH                  ; determine graphics mode:
    int     10H                     ; page number in bh
    mov     ah,03H                  ; locate cursor:
    int     10H                     ; DH=row DL=column
    mov     old_cur,dx              ; save cursor position
    mov     ax,row                  ; row # in ax
    mov     cx,0019H                ; max no of rows
    call    MODULO                  ; put mod. in dx
    mov     bl,dl                   ; mov mod. to bl
    mov     ax,col                  ; col # in ax
    mov     cx,050H                 ; max no of columns
    call    MODULO                  ; put mod. in dx
    mov     dh,bl                   ; assemble row:col coords in dh:dl
    mov     ah,02H                  ;
    int     10H                     ; and then position cursor
    cmp     si,00H                  ;
    jne     NXT_CODE                ; if this is not SETCUR, do some more
    jmp     FINISHED                ; if SI=0, SETCUR is finished
NXT_CODE:
    cmp     si,2
```

```
    je      CHANGE              ; if SI=2, CHANGE attributes
    cmp     si,3
    je      CLEAR               ; if si=3, CLEAR block
                                ; otherwise, erase to the end:
    mov     cx,0050H            ; length of a row
    sub     cl,dl               ; number of chars to erase
    mov     al,20H              ; write a space
    mov     bl,07H              ; with normal attribute
    mov     ah,09H              ; using function 9
    int     10H                 ; of int 10
    jmp     FINISHED
CHANGE:
    mov     ax,attrib           ;
    mov     cx,06H              ; 6 different attributes
    call    MODULO              ; ensure attrib is in range
    lea     bp,attribs          ; address of attribute data
    add     bp,dx               ; offset of required byte
    mov     bl,ds:[bp]          ; move it into bl
    mov     ax,chars            ;
    mov     di,bx               ; save the attribute byte
    mov     cx,050H             ; max no of chars
    call    MODULO              ;
    mov     bp,dx               ; save no.of chars to change
    inc     bp                  ; this is because of DEC later
    mov     ah,03H              ; use function 3 of
    int     10H                 ; int 10 to locate cursor
LOOP1:
    mov     ah,02H              ; use function 2 of
    int     10H                 ; int 10 to position cursor
    mov     ah,08H              ; use function 8 of
    int     10H                 ; int 10 to get char & attrib
    mov     bx,di               ; get the new attribute byte
    mov     cx,01H              ; write this character once
    mov     ah,09H              ; using function 9
    int     10H                 ; of int 109
    inc     dl                  ; increment the column,
    dec     bp                  ; decrement the overall counter
    jnz     LOOP1               ; more to do? Go and do it...
    mov     dx,old_cur          ; find where the cursor was...
    mov     ah,09H              ; and use function 9
    int     10H                 ; of int 10 to replace it
    jmp     FINISHED
CLEAR:                          ; jump here to clear a block
    mov     di,endrow
    sub     di,begrow           ; no. of rows to clear in di
    inc     di
    mov     cx,endcol           ;
    sub     cx,begcol
    inc     cx                  ; no. cols to clear in cx
WRT_SPC:
    mov     al,20H              ; write spaces
    mov     bl,07H              ; with normal attribute
    mov     ah,09H              ; use function 9
    int     10H                 ; of int 10 to write cx spaces
    inc     dh                  ; increase row no.
    mov     ah,02H              ; and then
    int     10H                 ; position cursor
    dec     di                  ; decrease counter
    jnz     WRT_SPC             ; carry on
    mov     dx,old_cur          ; find where the cursor was...
    mov     ah,02H              ; and use function 2
    int     10H                 ; of int 10 to replace it
FINISHED:
    mov     dx,si               ; move subroutine code into dx
    pop     di                  ; |
    pop     si                  ; | MS-Fortran standard epilogue
    mov     sp,old_sp           ; |
    pop     bp                  ; |
    cmp     dx,01D              ; what subroutine was called?
    jle     RETF_2              ; if <=1 only 2 parameters
            ret 16D             ; pop 4 parameters
```

```
RETF_2:         ret 08D                 ; pop 2 parameters
SCREEN    ENDP
SUBS      PROC      NEAR
          PUBLIC    MODULO,
MODULO    LABEL     NEAR
     cmp        ax,01H                  ; is ax >= 1?
     jns        OKAY                    ; if yes, ok
     sub        ax,01H                  ; if no, subtract 1 and
     neg        ax                      ; change sign
OKAY:
     sub        ax,01H                  ; subtract 1 from ax
     cwd                                ; convert ax to dx:ax
     idiv       cx                      ; divide by cx (max value), leaving remainder
                                        ; in dx
     ret
SUBS      ENDP
SCREEN_TEXT    ENDS                     ; code segment finished
END
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;    PORT Input-Output operations
;    Called from FORTRAN as:
;        call BYTOUT(N,Badd,REG,CNUM,clRLIM)
;        call  BYTIN(N,Badd,REG,CNUM,clRLIM)
;    where
;                N = byte to output/input                          INTEGER*1
;             Badd = board address                                 INTEGER*2
;              REG = register number                               INTEGER*1
;             CNUM = channel number                                INTEGER*1
;           clRLIM = limit reset .TRUE. = 1 = override             LOGICal*1
;                               .FalSE.= 0 = normal operation
     TITLE      portio.asm
     NAME       portio
PORT_IO_TEXT  SEGMENT  BYTE PUBLIC 'CODE'
PORT_IO_TEXT  ENDS
_DATA         SEGMENT  WORD PUBLIC 'DATA'
_DATA         ENDS
CONST         SEGMENT  WORD PUBLIC 'CONST'
CONST         ENDS
_BSS          SEGMENT  WORD PUBLIC 'BSS'
_BSS          ENDS
DGROUP   GROUP     CONST,    _BSS,     _DATA
    ASSUME     CS: PORT_IO_TEXT,      DS: DGROUP,     SS: DGROUP,     ES: DGROUP
_DATA         SEGMENT                ; All initialized local data
_DATA         ENDS                   ; resides in _DATA.
_BSS          SEGMENT                ; All uninitialized local data resides in
_BSS          ENDS                   ; _BSS.   uldata1 is integer*2
PORT_IO_TEXT  SEGMENT                ; code segment start
PUBLIC    BYTOUT,BYTIN               ; declare all labels as PUBLIC for
                                     ; easier debugging
EXTRN     SPD_4:FAR,SPD_9:FAR        ; declare external procedures
PORT_IO       PROC FAR               ;
BYTOUT:
     push    bp                      ; save calling program's bp
     mov     bp,sp                   ; set base pointer to stack pointer
     sub     sp,04H                  ; reserve 04 bytes for local stack.
     mov     ah,0                    ; ah=0 means output
     les     bx,DWORD PTR [bp+22d]; load es:bx with address of N
     mov     al,BYTE PTR es:[bx]   ; load al with N
     jmp     MAIN
BYTIN:
     push    bp                      ; save calling program's bp
     mov     bp,sp                   ; set base pointer to stack pointer
     sub     sp,04H                  ; reserve 04 bytes for local stack.
     mov     ah,1                    ; ah=1 means input
MAIN:                                ; COMPUTE '88 PORT addRESS OF HCTL REGISTER
     les     bx,DWORD PTR [bp+18d]; load es:bx with address of Badd
     mov     dx,WORD PTR es:[bx]   ; load dx with Badd:
                                     ; Badd occupies bits 3-9 of dx
     shr     dx,1
     shr     dx,1                    ; shift so that Badd occupies bits 1-7 of dx
     les     bx,DWORD PTR [bp+14d]; load es:bx with address of REG
```

```
        mov     dh,BYTE PTR es:[bx]  ; load dh with REG:
                                     ; REG occupies bits 8-13 of dx
        shl     dx,1                 ;
        shl     dx,1                 ; REG & Badd occupy bits 10-15 & 3-9 of dx
        les     bx,DWORD PTR [bp+10d]; load es:bx with address of CNUM
        mov     cl,BYTE PTR es:[bx]  ; load cl with CNUM
        and     cl,00000111b         ; clear all but bits 0-2
        sub     cl,01                ; cl=CNUM-1, occupying bits 0-1
        shl     cl,01                ; CNUM-1 occupies bits 1-2 of cx
        or      dl,cl                ; transfer CNUM-1 to dx
        les     bx,DWORD PTR [bp+06d]; load es:bx with address of clRLIM
        mov     cl,BYTE PTR es:[bx]  ; load cl with clRLIM
        not     cl                   ; negate cl (board is active low, variable
                                     ;            is active true)
        and     cl,01                ; clear all but bit 0
        or      dl,cl                ; set bit 0 of dl to reflect clRLIM
        cmp     ah,01                ; test for operation type
        je      READ                 ; ah=1 for input
        out     dx,al                ; output N
        jmp     FINISH
READ:
        push    dx
        call    SPD_4                ; switch processor speed
        pop     dx
        mov     ah,10H               ; set counter
        in      al,dx                ; start HCTL read cycle
LOOP01:                              ; |
        dec     ah                   ; | delay loop
        jnz     LOOP01               ; |
        in      al,dx                ; complete HCTL read cycle
        push    ax
        call    SPD_9                ; switch processor speed
        pop     ax
        les     bx,DWORD PTR [bp+22d]; load es:bx with address of N
        mov     BYTE PTR es:[bx],al  ; load return stack with N
FINISH:
        mov     sp,bp                ; remove space reserved for local stack
        pop     bp                   ; restore calling program's bp
        ret     20D                  ; clear parameters from stack & return
PORT_IO         ENDP
PORT_IO_TEXT    ENDS                 ; code segment finished
END
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;   MS-FORTRAN compatible subroutine:
;
;   SPD_CHG: Routine to switch Elonex PC88C processor speed.
;   Called as:
;       SPD_4 - switch to 4.77MHz
;       SPD_8 - switch to 8MHz
;       SPD_9 - switch to 9MHz
;
;   Destroys ax and dx.
;
    TITLE       spd_chg.asm
    NAME        spd_chg
SPD_CHG_TEXT    SEGMENT  BYTE PUBLIC 'CODE'
SPD_CHG_TEXT    ENDS
_DATA           SEGMENT  WORD PUBLIC 'DATA'
_DATA           ENDS
CONST           SEGMENT  WORD PUBLIC 'CONST'
CONST           ENDS
_BSS            SEGMENT  WORD PUBLIC 'BSS'
_BSS            ENDS
DGROUP    GROUP CONST, _BSS, _DATA
ASSUME    CS: SPD_CHG_TEXT, DS: DGROUP, SS: DGROUP, ES: DGROUP
_DATA           SEGMENT              ; All initialized local data
_DATA           ENDS                 ; resides in _DATA.
_BSS            SEGMENT              ; All uninitialized local data resides in
_BSS            ENDS                 ; _BSS.
SPD_CHG_TEXT    SEGMENT              ; code segment start
    PUBLIC      SPD_4,SPD_8,SPD_9    ; declare all labels as PUBLIC for
```

```
                                        ; easier debugging
SPD_CHG         PROC FAR                ;
SPD_4:
    mov         ah,00H
    jmp         main
SPD_8:
    mov         ah,40H
    jmp         MAIN
SPD_9:
    mov         ah,80H
MAIN:
    in          al,61H
    or          al,04H
    out         61H,al
    and         al,02
    or          ah,al
    mov         al,ah
    out         63H,al
    mov         dx,03ffH
    out         dx,al
    ret
SPD_CHG         ENDP                    ;
SPD_CHG_TEXT    ENDS                    ; code segment finished
END
;* * * *
```

```
WAIT_TEXT     ENDS                  ; code segment finished
END
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;+-------------------------------------------------------------------------------+
;|                                                                               |
;| BEEPER: make sounds using the beeper                                          |
;|                                                                               |
;| Usage:  call BEEP (no parameters)                                             |
;| Usage:  call BOOP (no parameters)                                             |
;| Usage:  call TONE (on,off,loops)                                              |
;|                                                                               |
;+-------------------------------------------------------------------------------+
    TITLE     beeper.asm: sound a beep from MS-Fortran
    NAME      beeper
BEEPER_TEXT   SEGMENT  BYTE PUBLIC 'CODE'
BEEPER_TEXT   ENDS
_DATA         SEGMENT  WORD PUBLIC 'DATA'
_DATA         ENDS
CONST         SEGMENT  WORD PUBLIC 'CONST'
CONST         ENDS
_BSS          SEGMENT  WORD PUBLIC 'BSS'
_BSS          ENDS
DGROUP   GROUP     CONST,     _BSS,      _DATA
    ASSUME    CS: BEEPER_TEXT,     DS: DGROUP,     SS: DGROUP,     ES: DGROUP
_DATA         SEGMENT
_DATA         ENDS
_BSS          SEGMENT
     on_time  dw 01h dup (?)
     off_time dw 01h dup (?)
_BSS          ENDS
BEEPER_TEXT   SEGMENT
    PUBLIC    BEEP,BOOP,TONE      ; declare all labels as PUBLIC for
                                  ; easier debugging
BEEP          PROC FAR            ;
    mov  dl,07H                   ; character to write is BEL:
    mov  ah,02H                   ; function 2 of
    int  21H                      ; int21 writes to the screen
    ret                           ;
BEEP          ENDP                ;
BOOP          PROC FAR            ;
    mov  dx,30D
    in   al,61H
    and  al,0feH
LOOP1:
    or   al,2
    out  61H,al
    mov  cx,17500D
CYCLE1:
    loop CYCLE1
    and  al,0fdH
    out  61H,al
    mov  cx,350D
CYCLE2:
    loop CYCLE2
    dec  dx
    jnz  LOOP1
    ret
BOOP          ENDP                ;
TONE          PROC FAR            ;
    push      bp                  ; save calling program's bp
    mov       bp,sp               ; set base pointer to stack pointer
    les       bx,DWORD PTR [bp+14d]; load es:bx with address of 1st parameter
    mov       ax,WORD PTR es:[bx]  ; load reg with parameter:
    mov       on_time,ax          ; save on_time counter
    les       bx,DWORD PTR [bp+10d]; load es:bx with address of 2nd parameter
    mov       ax,WORD PTR es:[bx]  ; load reg with parameter:
    mov       off_time,ax         ; save off_time counter
    les       bx,DWORD PTR [bp+06d]; load es:bx with address of 3rd parameter
    mov       dx,WORD PTR es:[bx]  ; load reg with tot_time counter
    in        al,61H
    and       al,0feH
```

```
LOOP2:
    or          al,2
    out         61H,al
    mov         cx,on_time
CYCLE3:
    loop        CYCLE3
    and         al,0fdH
    out         61H,al
    mov         cx,off_time
CYCLE4:
    loop        CYCLE4
    dec         dx
    jnz         LOOP2
    pop         bp                  ; restore calling program's bp
    ret         12D                 ; clear 3 parameters from stack
TONE            ENDP                ;
BEEPER_TEXT     ENDS                ; code segment finished
END
```

Appendix C Program MDLT: Modified Direct Linear Transformation

This Appendix contains a description and listing of the program code used for solving the photogrammetric problem by means of the Direct Linear Transform (DLT). This code is an extensively modified version of the program published by Marzan and Karara [98]. The modifications have been made by the current Author in order that the DLT may be used both with control points of poor spatial distribution and for real-time control of an analytical plotter.

This Appendix consists of two Sections. The first describes the structure of MDLT in terms of subroutine calls, and the second presents a listing of the FORTRAN code.

C.1 Structure of MDLT

Although MDLT has been designed to be executed from within program APCS (presented in Appendix B), it has been written as a standalone program. In this project, it has been found convenient to compile all of the code representing MDLT into a single library that can then be easily linked into APCS, or any other driving program.

The use of the DLT to solve the photogrammetric problem can customarily be divided into two parts: firstly, solution of the DLT equations and evaluation of the DLT coefficients; and, secondly, use of these coefficients to determine the object space coordinates corresponding to points imaged on the stereophotographs. The only connection between these two parts is shared data; the DLT coefficients and the interior orientation parameters associated with the two cameras. Consequently, MDLT has been written such that these procedures are totally separate, with the necessary data being shared between them by means of COMMON blocks. This has the effect of reducing the number of formal arguments required for the various subroutines in the program.

In this work, the novel use of the DLT is to achieve—in real-time—a stereomodel free from y-parallax. Consequently, there is in this instance a third

aspect to the photogrammetric problem, that of computing the y-parallax present at any point in the steromodel. As with the computation of object space coordinates, this only requires access to the DLT coefficients and the interior orientation parameters of the two photographs, and once again COMMON blocks represent the most efficient means of providing this access.

The structure of MDLT is indicated in Table C.01. The three separate photogrammetric procedures, described above, are indicated by the first three independent blocks of subroutines, with the fourth block representing a supplementary procedure designed to check to fit of the photogrammetric model to the control point data. All of these procedures share data through the use of COMMON blocks, the contents of which are described in the following Section.

A brief description of the main subroutines follows.

PHOTO The main subroutine for controlling solution of the DLT equations for all of the photographs currently in use (which, in the case of APCS, is two).

PHOTWT Determine the weights to be applied to each of the equations used in the DLT solution.

DPARM Determine, for each observation equation, the parameters associated with the DLT coefficients.

BFUNCS Called by SVDFIT to build the matrix that is subsequently decomposed. Here, BFUNCS simply transfers the parameters

Table C.01 Structure of program MDLT.

PHOTO	PHOTWT	
	DPARM	
	SVDFIT	BFUNCS[2]
		SVDCMP
		SVBKSB
	ORIENT[2]	
	SVDVAR	

OBJECT	CCOORD[3]	
	OBJTWT	
	OPARM	CCOORD[2]...
	SVDFIT...	
	SVDVAR...	

YPARX

CTLCHK	OBJECT...

A subroutine name followed by an ellipsis (i.e. ...) means that a complete definition of the subroutine is given earlier in the table. A number in brackets (i.e. [6]) represents the number of calls made to that subroutine by the calling subroutine.

determined by either DPARM or OPARM into a work array used by SVDFIT.

ORIENT Compute the interior and exterior orientation parameters for all of the photographs currently in use.

OBJECT Compute the object space coordinates of a point using measured photograph coordinates and the appropriate DLT coefficients.

CCOORD Apply lens distortion corrections to measured photograph coordinates.

OBJTWT Determine the weights to be applied to each of the equations used in the solution of object space coordinates.

OPARM Determine, for each observation equation, the parameters associated with the x, y and z components of the object space coordinates.

CTLCHK Repeatedly calls OBJECT, with the photograph coordinates of the control points, to determine the error between measured and computed object space coordinates.

Four of the subroutines—SVDFIT, SVDCMP, SVBKSB and SVDVAR—have been obtained from the literature [110] because of their specialist nature. Respectively, these routines perform the linear least squares fit on the basis of a singular value decomposition, implement the SVD, determine the parameters corresponding to the SVD and determine the covariance matrix corresponding to the best-fit parameters.

C.2 Listing of program MDLT

A complete listing of MDLT is given here. Although MDLT has been written in FORTRAN 77, some common extensions to the language that have been included by the manufacturer of the compiler have been used, particularly where they either simplify the code or allow for greater efficiency in execution. The extensions used in MDLT are:

(i) the data types `LOGICAL*1`, `INTEGER*1` and `INTEGER*2` have been used, where appropriate, to conserve memory and improve execution speed;

(ii) `INCLUDE` files, referenced in the listing of MDLT as, for example, `INCLUDE 'STATUS.BLK'`, and which are automatically inserted in the code at compile time, have been used for all extensively used `COMMON` declarations in order to reduce the bulk of the code;

(iii) comments, beginning with an exclamation mark, following and on the same line as, program statements.

The complete set of INCLUDE files required for MDLT follows.

```
C Include File DLT.PRM
      INTEGER
     +    DLTK1,DLTK2,DLTK3,DLTK4                                  ! K1=no. photos
      PARAMETER                                                    ! K2=no. DLT params
     +    ( DLTK1=2, DLTK2=20, DLTK3=50, DLTK4=2*DLTK3 )           ! K3=max no. data

C Include File DLTCRD.BLK
      INTEGER
     +    NPHOT,NUMCTL,CTLID
      DOUBLE PRECISION
     +    CONTRL,SX,SY,SZ,CPC,VRNCE
      COMMON
     +    /DLTCRD/
     +    NPHOT,NUMCTL,CTLID(DLTK3),CONTRL(3,DLTK3),CPC(2,DLTK3,DLTK1),
     +    VRNCE(2,DLTK1),SX,SY,SZ

C Include File TRNSFM.BLK
      INTEGER
     +    CORRCT,STERM,ATERM,IP
      DOUBLE PRECISION
     +    D,INOR,EXOR
      COMMON
     +    /TRNSFM/
     +    CORRCT,STERM,ATERM,IP,
     +    D(DLTK2,DLTK1),INOR(3,DLTK1),EXOR(6,DLTK1)

C Include File FITDAT.BLK
      DOUBLE PRECISION
     +    PARM(DLTK2,DLTK4)
      COMMON
     +    /FITDAT/PARM

C Include File IODAT.BLK
      INTEGER
     +    IDEV,ODEV
      COMMON
     +    /IODAT/IDEV,ODEV

C Include File CVRMAT.BLK
      DOUBLE PRECISION
     +    CVM
      COMMON/CVRNCE/
     +    CVM(DLTK2,DLTK2)

C Include File OBJDAT.BLK
      DOUBLE PRECISION
     +    OPC(2,DLTK1)
      COMMON
     +    /OBJDAT/OPC
```

The code for MDLT now follows. Each procedure (subroutine or function) is separated from adjacent procedures by a row of asterisks across the page. The procedures are generally given in row-major order corresponding to Table C.01, i.e. PHOTO, then PHOTWT, then DPARM, and so on, with the exception of the supporting routines SVDFIT SVDCMP, SVBKSB and SVDVAR which are listed last.

```
      SUBROUTINE PHOTO(CN,NC,NP,ID,SC,SE,PC,PV,MC,OD)
C
C     PHOTO computes the DLT parameters and interior and exterior
C     orientation for all photographs.
C
C     CN = code for type of coordinate corrections required: two digit number
C          xy with y=no. of symmetric terms and x=no. of asymmetric terms
C     NC = number of control points
C     NP = number of photographs
C     ID = vector of control point i.d.
C     SC = array (i,j) of space coordinates: i=x,y,z; j=point
C     SE = vector of standard errors in control point space coordinates
C     PC = array (i,j,k) of photograph coordinates: i=x,y; j=point, k=photo
C     PV = array (i,j) of photocoord variances: i=x,y; j=photo
C     MC = max. no. control points (for dimensioning PC)
C     OD = output device
C
C Type Declarations: Formal Arguments
      INTEGER
     +    CN,NC,NP,ID,MC,OD
      REAL
     +    SC,SE,PC,PV
      DIMENSION
     +    ID(*),SC(3,*),SE(3),PC(2,MC,*),PV(2,*)
C Type Declarations: Common Blocks
      INCLUDE 'DLT.PRM'
      INCLUDE 'DLTCRD.BLK'
      INCLUDE 'TRNSFM.BLK'
      INCLUDE 'FITDAT.BLK'
      INCLUDE 'IODAT.BLK'
      INCLUDE 'CVRMAT.BLK'
C Type Declarations: Local Variables
      INTEGER*2
     +    AXIS,PNUM,TCOUNT
      DOUBLE PRECISION
     +    DOF,SDEV,DTEMP,CHI2,VARUW,X,Y,OLDC2,TOL,INFNTY,CNUM
      DOUBLE PRECISION
     +    U,V,W
      INTEGER
     +    PHNUM,NEQNS,I,J,K,NPASS,ERR,ND
      DIMENSION
     +    SDEV(DLTK4,DLTK1),DTEMP(DLTK2),VARUW(DLTK1),CNUM(DLTK1),
     +    X(DLTK4),Y(DLTK4),U(DLTK4,DLTK2),V(DLTK2,DLTK2),W(DLTK2)
      CHARACTER*5 INHDG(3),EXHDG(6)
      DATA
     +    (INHDG(i),i=1,3)/'   Xp','   Yp','    c'/,
     +    (EXHDG(i),i=1,6)
     +        /'omega',' phi','kappa','    X','    Y','    Z'/,
     +    TOL/0.0005/,INFNTY/9.99D+099/
C Executable Code
      CORRCT = CN                                  ! transfer formal arguments
      NUMCTL = NC                                  ! to COMMON variables
      NPHOT = NP
      ODEV = OD
      DO 1000 I=1,NUMCTL
          CTLID(I) = ID(I)
 1000 CONTINUE
      DO 2000 J=1,NUMCTL
          DO 2100 I=1,3
              CONTRL(I,J) = SC(I,J)
 2100     CONTINUE
 2000 CONTINUE
      SX = SE(1)
      SY = SE(2)
      SZ = SE(3)
      DO 3000 K=1,NPHOT
          DO 3100 J=1,NUMCTL
              DO 3110 I=1,2
                  CPC(I,J,K) = PC(I,J,K)
 3110         CONTINUE
 3100     CONTINUE
```

```
3000 CONTINUE
     DO 4000 J=1,NPHOT
         DO 4100 I=1,2
             VRNCE(I,J) = PV(I,J)
4100     CONTINUE
4000 CONTINUE
     WRITE(ODEV,'(A//A)')
    +    ' PHOTOGRAMMETRIC DATA:',' Control parameters:'
     WRITE(ODEV,'(4(A,I3))') ' Correction code:',CORRCT,
    +                        ' Number of control points:',NUMCTL
     WRITE(ODEV,'(A)')
    +    ' Control space coordinates:'
     DO J=1,NUMCTL
         WRITE(ODEV,'(2X,I4,3(2X,F8.3))') CTLID(J),(CONTRL(I,J),I=1,3)
     ENDDO
     WRITE(ODEV,'(A)') ' Standard errors of space coordinates:'
     WRITE(ODEV,'(3(2X,F7.4))')SX,SY,SZ
     WRITE(ODEV,'(A)')' Photograph coordinates:'
     DO J=1,NUMCTL
         WRITE(ODEV,'(2X,I4,4(2X,F8.3))')
    +        CTLID(J),((CPC(I,J,K),I=1,2),K=1,2)
     ENDDO
     WRITE(ODEV,'(A)') ' Variances of photograph coordinates:'
     WRITE(ODEV,'(2(12X,A))') 'Photo 1','   Photo 2'
     DO I=1,2
         WRITE(ODEV,'(2(1X,F21.6))') (VRNCE(I,J),J=1,2)
     ENDDO
     STERM = MOD(CORRCT,10)                      ! photo analysis starts here
     ATERM = INT(CORRCT/10)
     IP = 11 + STERM + 2*ATERM                   ! no. of DLT parameters
     NEQNS=2*NUMCTL
     DOF = NEQNS - IP                            ! DOF=N(data)-N(parameters)
     DO  6000 PHNUM=1,NPHOT                      ! analyse all photos
         IF (IP.GT.11) THEN
             INOR(1,PHNUM)=0.0                   ! ensure comparator coords of
             INOR(2,PHNUM)=0.0                   ! principal point and higher
             DO 6100 I=12,IP                     ! DLT parameters are all set
                 D(I,PHNUM)=0.0                  ! to zero, as they will be used
6100         CONTINUE                            ! in the first pass of SVDFIT
         ENDIF
         DO 6200 I=1,NEQNS
             AXIS = 2-MOD(I,2)
             PNUM = (1+I)/2
             X(I) = DBLE(I)                      ! initialize vectors of indep
             Y(I) = CPC(AXIS,PNUM,PHNUM)         ! and dep variables
6200     CONTINUE
         OLDC2 = 1000.0
         IF (ATERM.GT.0) THEN
             TCOUNT = 2
         ELSE
             TCOUNT = 30
         ENDIF
         DO 6300 NPASS=1,TCOUNT
             CALL PHOTWT(NPASS,NEQNS,PHNUM,SDEV(1,PHNUM)) ! calc sdevs
             CALL DPARM(NPASS,NEQNS,PHNUM)       ! calc eqn parameters
             IF (NPASS.EQ.1) THEN
                 ND = 11
             ELSE
                 ND = IP
             ENDIF
             CALL SVDFIT                         ! least squares soln.
    +            (X,Y,SDEV(1,PHNUM),NEQNS,DTEMP,! chi2 (chi-squared) equals
    +             ND,U,V,W,DLTK4,DLTK2,CHI2,ERR)! sum of residuals squared
             DO 6310 I=1,ND                      ! update D to include latest
                 D(I,PHNUM) = DTEMP(I)           ! computed values from SVDFIT
6310         CONTINUE
             IF (IP.GT.11) THEN                  ! lens distortion corrections
                 I=1                             ! required on next pass, so
                 CALL ORIENT(PHNUM,I)            ! compute camera orientations
             ENDIF
             IF (ABS(OLDC2-CHI2).LT.TOL) THEN
```

```
                EXIT
            ELSE
                OLDC2 = CHI2
            ENDIF
 6300       CONTINUE
            CALL SVDVAR(V,ND,DLTK2,W,CVM,DLTK2)     ! compute var-covar matrix...
            VARUW(PHNUM) = CHI2/DOF
            DO 6400 J=1,IP                          ! ...and scale throughout by
                DO 6410 I=1,IP                      ! variance of unit weight
                    CVM(I,J) = CVM(I,J)*VARUW(PHNUM)
 6410           CONTINUE
                SDEV(J,PHNUM) = SQRT(CVM(J,J))
 6400       CONTINUE
            IF (W(IP).NE.0) THEN
                CNUM(PHNUM) = W(1)/W(IP)            ! compute condition number
            ELSE
                CNUM(PHNUM) = INFNTY
            ENDIF
            NPASS=2                                 ! compute interior and exterior
            CALL ORIENT(PHNUM,NPASS)                ! orientations of camera
 6000   CONTINUE
        WRITE(ODEV,10)                              ! print title
        WRITE(ODEV,'('' '')')
        WRITE(ODEV,'(10X\)')
        DO I=1,2
            WRITE(ODEV,20) I
        ENDDO
        WRITE(ODEV,'('' '')')
        WRITE(ODEV,30)
       +    ('DLT parameters   standard deviations    ',I=1,NPHOT)
        WRITE(ODEV,'('' '')')
        DO 8000 I = 1,IP                            ! print the DLT parameters
            DO 8100 J=1,2
                WRITE(ODEV,40) I,D(I,J),SDEV(I,J)
 8100       CONTINUE
            WRITE(ODEV,'('' '')')
 8000   CONTINUE
        WRITE(ODEV,'('' '')')
        WRITE(ODEV,50) (SQRT(VARUW(I)),I=1,NPHOT)
        WRITE(ODEV,'('' '')')
        WRITE(ODEV,60) (CNUM(I),I=1,NPHOT)
        WRITE(ODEV,'('' '')')
        WRITE(ODEV,'('' Interior Orientation:'')')
        DO I=1,3
            WRITE(ODEV,70) INHDG(I),(INOR(I,J),J=1,2)
        END DO
        WRITE(ODEV,'('' Exterior Orientation:'')')
        DO I=1,6
            WRITE(ODEV,70) EXHDG(I),(EXOR(I,J),J=1,2)
        END DO
        RETURN
   10   FORMAT(                                                              /
       +    ' Computed values of DLT parameters and standard deviations'/)
   20   FORMAT(10X,'photograph',I3,'                    '\)
   30   FORMAT(1X,4(A\))
   40   FORMAT(1X,I2,2X,E11.5,4X,E11.5,'          '\)
   50   FORMAT(' Standard errors of unit weight:'/,4(20X,F11.5,9X))
   60   FORMAT(' Condition numbers:'/,4(20X,E11.5,9X))
   70   FORMAT(1X,A,4(16X,F9.4,15X))
        END
* * * *
```

```
      DIMENSION
     +    SDEV(*)
C Type Declarations: Common Blocks
      INCLUDE 'DLT.PRM'
      INCLUDE 'DLTCRD.BLK'
      INCLUDE 'TRNSFM.BLK'
C Type Declarations: Local Variables
      INTEGER
     +    I,J,AXIS,INC,PNUM
      DOUBLE PRECISION
     +    COORD,VAR,A,B
      DIMENSION
     +    B(3)
C Executable Code
      IF (NPASS.EQ.1) THEN
          DO 1000 I=1,NEQNS
              SDEV(I)=1.0                          ! set std. devs. to unity
 1000     CONTINUE
      ELSE                                         ! compute new weights using
          DO 2000 I=1,NEQNS                        ! DLT parameters from previous
              AXIS = 2 - MOD(I,2)                  ! pass
              INC = 4*(AXIS-1)
              PNUM = (I+1)/2
              COORD=CPC(AXIS,PNUM,PHNUM)           ! retrieve photo coord
              VAR = 0.0                            ! reset variance
              IF (AXIS.EQ.1) THEN
                  A = 1.0                          ! this is a new point, so
                  DO 2100 J = 1 , 3                ! compute factor A:
                      A = A + D(J+8,PHNUM)*CONTRL(J,PNUM)
 2100             CONTINUE                         ! A = 1 + xL9 + yL10 + zL10
              ENDIF
              DO 2200 J=1,3                                  ! evaluate
                  B(J) = D(J+8,PHNUM)*COORD-D(J+INC,PHNUM)   ! components of
 2200         CONTINUE                                       ! weight function
              VAR = (B(1)*SX)**2 + (B(2)*SY)**2 + (B(3)*SZ)**2
              VAR = VAR/(A*A) + VRNCE(AXIS,PHNUM)      ! VAR=variance associated
              SDEV(I)=SQRT(VAR)                        ! with condition equation
 2000     CONTINUE                                     ! (eq16 of M&K)
      ENDIF
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE DPARM(P,N,F)
C
C     DPARM evaluates the basis functions for fitting the DLT parameters
C
C     P = pass number
C     N = no of equations to be built
C     F = scalar integer: photo number
C
C Type Declarations: Formal Arguments
      INTEGER
     +    P,N,F
C Type Declarations: Common Blocks
      INCLUDE 'DLT.PRM'
      INCLUDE 'DLTCRD.BLK'
      INCLUDE 'TRNSFM.BLK'
      INCLUDE 'FITDAT.BLK'
      INCLUDE 'IODAT.BLK'
C Type Declarations: Local Variables
      INTEGER
     +    I,J,PNUM,AXIS,INC
      DOUBLE PRECISION
     +    COORD,XDIST,YDIST,DIST,XDSQRD,YDSQRD,R2,R4,R6,R9,A
C Executable Code
      DO 1000 I=1,N                                ! equation no.
          PNUM = (I+1)/2                           ! point no.
          AXIS = 1 + MOD(I+1,2)
          INC = 4*(AXIS-1)                         ! inc(x)=0, inc(y)=4
          COORD = CPC(AXIS,PNUM,F)                 ! retrieve photo coord
          DO 1100 J=1,8
```

```
            PARM(J,I)=0.0                          ! reset first 8 coefficients
 1100     CONTINUE
          DO 1200 J=1,3                            ! build Q(1)-Q(3) [axis=1] or
              PARM(J+INC,I) = CONTRL(J,PNUM)       ! Q(5)-Q(7) [axis=2], together
              PARM(J+8,I) = -CONTRL(J,PNUM)*COORD! with Q(9)-Q(11)
 1200     CONTINUE
          PARM(4+INC,I) = 1.0                      ! Q(4) or Q(8)
          IF (IP.GT.11 .AND. P.GT.1) THEN          ! corrections required
              XDIST = CPC(1,PNUM,F) - INOR(1,F)
              YDIST = CPC(2,PNUM,F) - INOR(2,F)
              IF (AXIS.EQ.1)THEN
                  DIST=XDIST
              ELSE
                  DIST=YDIST
              ENDIF
              XDSQRD = XDIST**2
              YDSQRD = YDIST**2
              R2 = XDSQRD + YDSQRD
              R4 = R2*R2
              R6 = R2*R4
              A=1.0
              DO 1300 J=1,3
                  A = A + D(J+8,F)*CONTRL(J,PNUM)
 1300         CONTINUE
              IF (STERM.GE.1) THEN             ! symmetric corrections
                  PARM(12,I) = -DIST*R2*A
                  IF (STERM.GE.2) THEN
                      PARM(13,I) = -DIST*R4*A
                      IF(STERM.GE.3) THEN
                          PARM(14,I) = -DIST*R6*A
                      ENDIF
                  ENDIF
              ENDIF
              IF (ATERM.GT.0) THEN             ! asymmetric corrections
                  R9 = 2.0*XDIST*YDIST
                  IF (AXIS.EQ.1) THEN
                      PARM(12+STERM,I) = -(2.0*XDSQRD + R2)*A
                      PARM(13+STERM,I) = -R9*A
                  ELSE
                      PARM(12+STERM,I) = -R9*A
                      PARM(13+STERM,I) = -(2.0*YDSQRD + R2)*A
                  ENDIF
              ENDIF
          ENDIF
 1000 CONTINUE
      RETURN
      END
* * * *
```

```
C Executable Code
      J = INT(X)
      DO 1000 I=1,M
         COEFF(I) = PARM(I,J)
 1000 CONTINUE
      RETURN
      END
* * * *
```

```
      END
* * * *
```

```
                                                ! to the sum of the residuals
                                                ! squared
          IF (ABS(OLDC2-CHI2).LT.TOL) THEN
              EXIT
          ELSE
              OLDC2 = CHI2
          ENDIF
 4000 CONTINUE
      CALL SVDVAR(V,3,2*DLTK1,W,CVMOBJ,3)       ! compute var-covar matrix
      VARUW = CHI2/DOF                          ! and scale throughout by
      VR(4) = 0.0                               ! variance of unit weight
      DO 5000 J=1,3
          SC(J) = OBJTSC(J)
          DO 5100 I=1,3
              CVMOBJ(I,J) = CVMOBJ(I,J)*VARUW
 5100     CONTINUE
          VR(J) = CVMOBJ(J,J)                   ! variance for this direction
          VR(4) = VR(4) + CVMOBJ(J,J)           ! variance of position
 5000 CONTINUE
      RETURN
      END
* * * *
```

```
                ASYM= D(12+STERM,PHNUM)*R7 + D(13+STERM,PHNUM)*R9
            ELSE
                R8 = 2.0*YDSQRD + RSQRD
                ASYM= D(12+STERM,PHNUM)*R9 + D(13+STERM,PHNUM)*R8
            ENDIF
        ENDIF
        CCOORD = OPC(PAXIS,PHNUM) + (DIST*SYMCOR+ASYM)
    ENDIF
    RETURN
    END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
    SUBROUTINE OBJTWT(NPASS,NEQNS,OBJTSC,SDEV,ERROR)
C
C     OBJTWT computes the standard deviations to be applied to each
C     of the independent equations used in the solution of object
C     space coordinates.
C
C Type Declarations: Formal Arguments
      INTEGER
     +    NPASS,NEQNS
      LOGICAL*1
     +    ERROR
      DOUBLE PRECISION
     +    OBJTSC,SDEV
      DIMENSION
     +    OBJTSC(*),SDEV(*)
C Type Declarations: Common Blocks
      INCLUDE 'DLT.PRM'
      INCLUDE 'DLTCRD.BLK'
      INCLUDE 'TRNSFM.BLK'
      INCLUDE 'OBJDAT.BLK'
      INCLUDE 'CVRMAT.BLK'
C Type Declarations: Local Variables
      INTEGER
     +    AXIS,INC,PHNUM,I,J,K,L
      DOUBLE PRECISION
     +    COORD,VAR,A,A2,B,TEMP,
     +    XDIST,YDIST,DIST,XDSQRD,YDSQRD,R2,R4,R6,R9
      DIMENSION
     +    B(DLTK2)
C Executable Code
      ERROR=.FALSE.
      IF (NPASS.EQ.1) THEN                          ! first pass, so sdev=1.0
          DO 1000 I=1,NEQNS
              SDEV(I)=1.0
 1000     CONTINUE
      ELSE                                          ! compute new weights using
          DO 2000 I=1,NEQNS                         ! OBJTSC and var/covar matrix
              AXIS = 2 - MOD(I,2)                   ! from last pass of DLT soln.
              PHNUM = (I+1)/2
              INC = 4*(AXIS-1)
              COORD=OPC(AXIS,PHNUM)                 ! retrieve photo coord
              IF (AXIS.EQ.1) THEN
                  A = 1.0                           ! this is a new photo, so
                  DO 2100 J = 1 , 3                 ! compute factor A^2:
                      A = A + D(J+8,PHNUM)*OBJTSC(J)
 2100                 CONTINUE                       ! A = 1 + xL9 + yL10 + zL11
                  A2 = A*A
              ENDIF
              DO 2200 J=1,8                         ! reset first 8 DLT params.
                  B(J) = 0.0
 2200         CONTINUE
              DO 2300 J=1,3                         ! build non-zero elements of
                  K = J + INC                       ! condition equation by
                  L = J + 8                         ! looping over x,y and z:
                  B(K) = -OBJTSC(J)                 ! 1,2,3 [x] or 5,6,7 [y]
                  B(L) = OBJTSC(J)*COORD            ! 9,10,11
 2300         CONTINUE
              B(4+INC) = -1.0                       ! 4 [x] or 8 [y]
              IF (IP.GT.11) THEN                    ! corrections required
                  XDIST = OPC(1,PHNUM) - INOR(1,PHNUM)
```

```
                YDIST = OPC(2,PHNUM) - INOR(2,PHNUM)
                IF (AXIS.EQ.1)THEN
                    DIST=XDIST
                ELSE
                    DIST=YDIST
                ENDIF
                XDSQRD = XDIST**2
                YDSQRD = YDIST**2
                R2 = XDSQRD + YDSQRD
                R4 = R2*R2
                R6 = R2*R4
                IF (STERM.GE.1) THEN            ! symmetric corrections
                    B(12) = DIST*R2*A
                    IF (STERM.GE.2) THEN
                        B(13) = DIST*R4*A
                        IF(STERM.GE.3) THEN
                            B(14) = DIST*R6*A
                        ENDIF
                    ENDIF
                ENDIF
                IF (ATERM.GT.0) THEN            ! asymmetric corrections
                    R9 = 2.0*XDIST*YDIST
                    IF (AXIS.EQ.1) THEN
                        B(12+STERM) = (2.0*XDSQRD + R2)*A
                        B(13+STERM) = R9*A
                    ELSE
                        B(12+STERM) = R9*A
                        B(13+STERM) = (2.0*YDSQRD + R2)*A
                    ENDIF
                ENDIF
            ENDIF
            VAR = 0.0
            DO 2400 J=1,IP
                TEMP = 0.0                      ! build 1st term of
                DO 2410 K=1,IP                  ! eqn29 in M&K. CVM
                    TEMP = TEMP + B(K)*CVM(K,J)! is var/covar matrix
 2410           CONTINUE                        ! from DLT parameters
                VAR = VAR + TEMP*B(J)           ! calculation
 2400       CONTINUE
            VAR = VAR + A2*VRNCE(AXIS,PHNUM)    ! add 2nd term
            IF (VAR.LT.0.0) THEN
                ERROR=.TRUE.
                RETURN
            ELSE
                ERROR=.FALSE.
                SDEV(I) = SQRT(VAR)
            ENDIF
 2000     CONTINUE                              ! next equation
      ENDIF
      RETURN
      END
* * * *
```

```
     +     CCOORD
      EXTERNAL
     +     CCOORD
C Executable Code
      DO 1000 I=1,N*2                              ! build all equations
          AXIS = 2-MOD(I,2)
          FNUM = (1+I)/2
          INC = 4*(AXIS-1)
          DO 1100 J=1,3                            ! build elements Q(1)-Q(3)...
              K = J + INC                          ! using D(1)-D(3) or D(5)-D(7)
              L = J + 8                            ! ...and D(9)-D(11)
              PARM(J,I) = D(K,FNUM) - D(L,FNUM)*CCOORD(AXIS,FNUM)
 1100     CONTINUE
 1000 CONTINUE
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      REAL FUNCTION YPARX(XL,YL,XR,COORD)
C
C     compute Y-coord to maintain PARallaX free stereoscopic model
C
C Type Declarations: Formal Arguments
      REAL
     +     XL,YL,XR
      LOGICAL*1
     +     COORD
C Type Declarations: Common Blocks
      INCLUDE 'DLT.PRM'
      INCLUDE 'TRNSFM.BLK'
      INCLUDE 'OBJDAT.BLK'
C Type Declarations: Local Variables
      INTEGER*2
     +     L1,LOOP
      REAL
     +     A11,A21,A31,A12,A22,A32,A13,A23,A33,B1,B2,B3,
     +     C1,C2,C3,C4,C5,C6,DET,KX,KY,KZ,COR
      DOUBLE PRECISION
     +     CCOORD
      EXTERNAL
     +     CCOORD
C Executable Code
      IF (COORD) THEN
          LOOP = 10
          OPC(1,1) = DBLE(XL)
          OPC(2,1) = DBLE(YL)
          OPC(1,2) = DBLE(XR)
          XL = REAL(CCOORD(1,1))
          YL = REAL(CCOORD(2,1))
      ELSE
          LOOP = 1
      ENDIF
      DO L1=1,LOOP
          IF (L1.GT.1) THEN
              OPC(2,2) = DBLE(YPARX)
              XR = REAL(CCOORD(1,2))
              COR = REAL(CCOORD(2,2)) - YPARX
          ENDIF
          A11 = D(1,1)-D( 9,1)*XL
          A21 = D(5,1)-D( 9,1)*YL
          A31 = D(1,2)-D( 9,2)*XR
          A12 = D(2,1)-D(10,1)*XL
          A22 = D(6,1)-D(10,1)*YL
          A32 = D(2,2)-D(10,2)*XR
          A13 = D(3,1)-D(11,1)*XL
          A23 = D(7,1)-D(11,1)*YL
          A33 = D(3,2)-D(11,2)*XR
          B1  = XL-D(4,1)
          B2  = YL-D(8,1)
          B3  = XR-D(4,2)
          C1  = A22*A33-A32*A23
          C2  = A31*A23-A21*A33
```

```
            C3   = A21*A32-A31*A22
            C4   = B2*A33-B3*A23
            C5   = B2*A32-B3*A22
            C6   = B3*A21-B2*A31
            DET  = A11*C1+A12*C2+A13*C3
            KX   = B1*C1-A12*C4+A13*C5
            KY   = B1*C2+A11*C4+A13*C6
            KZ   = B1*C3-A12*C6-A11*C5
            YPARX = (KX*D(5,2) + KY*D(6,2) + KZ*D(7,2) + DET*D(8,2)) /
     +              (KX*D(9,2) + KY*D(10,2) + KZ*D(11,2) + DET)
            IF (COORD) THEN
                YPARX = YPARX + COR
            ENDIF
        ENDDO
        RETURN
        END
* * * *
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE SVDFIT(X,Y,SIG,NDATA,A,MA,U,V,W,MP,NP,CHISQ,ERR)
C
C     Given a set of NDATA points X(I) ,Y(I) with individual standard
C     deviations SIG(I), use Chi-2 minimization to determine the MA
C     coefficients A of the fitting function y = SUM(A.AFUNC(x)).
C     The fitting equations are solved using singular value
C     decomposition of the NDATA by MA matrix. Arrays U,V,W
C     provide workspace on input, on output they define the singular
C     value decomposition, and can be used to obtain the covariance
C     matrix. MP,NP are the physical dimensions of the matrices U,V,W,
C     as indicated below. It is necessary that MP>NDATA, NP>MA. The
C     program returns values for the MA fit parameters A and CHISQ.
C     The user supplies a subroutine BFUNCS(X,AFUNC,MA) that returns the
C     MA basis functions evaluated at x =X in the array AFUNC.
C
C Type Declarations: Formal arguments
      INTEGER
     +    NDATA,MA,MP,NP,ERR
      DOUBLE PRECISION
     +    X,Y,SIG,A,U,V,W,CHISQ
      DIMENSION
     +    X(NDATA),Y(NDATA),SIG(NDATA),A(MA),U(MP,NP),V(NP,NP),W(NP)
C Type Declarations:  Local variables
      INTEGER
     +    MMAX,NMAX,I,J
      DOUBLE PRECISION
     +    TOL,B,AFUNC,TMP,WMAX,THRESH,SUM
      PARAMETER(NMAX=50,MMAX=20,TOL=1.E-9)        ! max. expected NDATA and MA
      DIMENSION
     +    B(NMAX),AFUNC(MMAX)
C Executable Code
      DO 1000 I=1,NDATA                          ! accumulate fitting matrix coeffs
          CALL BFUNCS(X(I),AFUNC,MA)
          TMP=1.0/SIG(I)
          DO 1100 J=1,MA
              U(I,J)=AFUNC(J)*TMP
 1100     CONTINUE
          B(I)=Y(I)*TMP
 1000 CONTINUE
      CALL SVDCMP(U,NDATA,MA,MP,NP,W,V,ERR)       ! singular value decomposition
      IF (ERR.NE.0) RETURN
      WMAX=0.0
      DO 2000 J=1, MA                             ! edit singular values,
          IF (W(J).GT.WMAX) WMAX=W(J)             ! using TOL from parameter
 2000 CONTINUE                                    ! statement between here...
      THRESH=TOL*WMAX                             !
      DO 3000 J=1,MA                              !
          IF (W(J).LT.THRESH) W(J)=0.0            !
 3000 CONTINUE                                    ! ...and here.
      CALL SVBKSB(U,W,V,NDATA,MA,MP,NP,B,A)
      CHISQ=0.0                                   ! evaluate chi-square
      DO 4000 I=1,NDATA
          CALL BFUNCS(X(I),AFUNC,MA)
          SUM=0.0
          DO 4100 J=1,MA
              SUM=SUM+A(J)*AFUNC(J)
 4100     CONTINUE
          CHISQ=CHISQ+((Y(I)-SUM)/SIG(I))**2
 4000 CONTINUE
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE SVDCMP(A,M,N,MP,NP,W,V,ERR)
C
C     Given a matrix A, with logical dimensions M by N and physical
C     dimensions MP by NP, this routine computes its singular value
C     decomposition, A = U.W.Tr(V). The matrix U replaces A on output.
C     The diagonal matrix of singular values W is output as a vector W.
C     The matrix V (not the transpose Tr(V)) is output as V. M must be
C     greater or equal to N; if it is smaller, then A should be filled
```

```
C     up to square with zero rows.
C
C Type Declarations: Formal arguments
      INTEGER
     +    M,N,MP,NP,ERR
      DOUBLE PRECISION
     +    A,W,V
      DIMENSION
     +    A(MP,NP),W(NP),V(NP,NP)
C Type Declarations: Local variables
      INTEGER
     +    NMAX,MMAX,I,J,K,L,ITS,NM,JJ
      DOUBLE PRECISION
     +    RV1,G,SCALE,ANORM,S,F,H,C,X,Y,Z
      LOGICAL*1
     +    FLAG
      PARAMETER (MMAX=50,NMAX=20)       ! max. no. of parameters=NMAX
      DIMENSION
     +    RV1(NMAX)
C Executable Code
      ERR=0
      IF (M.LT.N) ERR=1
      IF (M.GT.MMAX) ERR=2
      IF (N.GT.NMAX) ERR=3
      IF (ERR.NE.0) RETURN
      G=0.0                    ! Start Householder reduction to bidiagonal form
      SCALE=0.0
      ANORM=0.0
      DO 1000 I=1,N
          L=I+1
          RV1(I)=SCALE*G
          G=0.0
          S=0.0
          SCALE=0.0
          IF (I.LE.M) THEN
              DO 1100 K=I,M
                  SCALE=SCALE+ABS(A(K,I))
 1100         CONTINUE
              IF (SCALE.NE.0.0) THEN
                  DO 1200 K=I,M
                      A(K,I)=A(K,I)/SCALE
                      S=S+A(K,I)*A(K,I)
 1200             CONTINUE
                  F=A(I,I)
                  G=-SIGN(SQRT(S),F)
                  H=F*G-S
                  A(I,I)=F-G
                  IF (I.NE.N) THEN
                      DO 1300 J=L,N
                          S=0.0
                          DO 1310 K=I,M
                              S=S+A(K,I)*A(K,J)
 1310                     CONTINUE
                          F=S/H
                          DO 1320 K=I,M
                              A(K,J)=A(K,J)+F*A(K,I)
 1320                     CONTINUE
 1300                 CONTINUE
                  ENDIF
                  DO 1400 K=I,M
                      A(K,I)=A(K,I)*SCALE
 1400             CONTINUE
              ENDIF
          ENDIF
          W(I)=SCALE*G
          G=0.0
          S=0.0
          SCALE=0.0
          IF ( (I.LE.M) .AND. (I.NE.N) ) THEN
              DO 1500 K=L,N
                  SCALE=SCALE+ABS(A(I,K))
```

```
1500          CONTINUE
              IF (SCALE.NE.0.0) THEN
                  DO 1600 K=L,N
                      A(I,K)=A(I,K)/SCALE
                      S=S+A(I,K)*A(I,K)
1600              CONTINUE
                  F=A(I,L)
                  G=-SIGN(SQRT(S),F)
                  H=F*G-S
                  A(I,L)=F-G
                  DO 1700 K=L,N
                      RV1(K)=A(I,K)/H
1700              CONTINUE
                  IF (I.NE.M) THEN
                      DO 1800 J=L,M
                          S=0.0
                          DO 1810 K=L,N
                              S=S+A(J,K)*A(I,K)
1810                      CONTINUE
                          DO 1820 K=L,N
                              A(J,K)=A(J,K)+S*RV1(K)
1820                      CONTINUE
1800                  CONTINUE
                  ENDIF
                  DO 1900 K=L,N
                      A(I,K)=A(I,K)*SCALE
1900              CONTINUE
              ENDIF
          ENDIF
          ANORM=MAX( ANORM,(ABS(W(I))+ABS(RV1(I))) )
1000  CONTINUE
      DO 2000 I=N,1,-1                  ! Accumulate right-hand transformations
          IF (I.LT.N) THEN
              IF (G.NE.0.0) THEN
                  DO 2100 J=L,N
                      V(J,I)=(A(I,J)/A(I,L))/G    ! double division
2100              CONTINUE                        ! avoids underflow
                  DO 2200 J=L,N
                      S=0.0
                      DO 2210 K=L,N
                          S=S+A(I,K)*V(K,J)
2210                  CONTINUE
                      DO 2220 K=L,N
                          V(K,J)=V(K,J)+S*V(K,I)
2220                  CONTINUE
2200              CONTINUE
              ENDIF
              DO 2300 J=L,N
                  V(I,J)=0.0
                  V(J,I)=0.0
2300          CONTINUE
          ENDIF
          V(I,I)=1.0
          G=RV1(I)
          L=I
2000  CONTINUE
      DO 3000 I=N,1,-1                  ! accumulate left-hand transformations
          L=I+1
          G=W(I)
          IF (I.LT.N) THEN
              DO 3100 J=L,N
                  A(I,J)=0.0
3100          CONTINUE
          ENDIF
          IF (G.NE.0.0) THEN
              G=1.0/G
              IF (I.NE.N) THEN
                  DO 3200 J=L,N
                      S=0.0
                      DO 3210 K=L,M
                          S=S+A(K,I)*A(K,J)
```

```
 3210                    CONTINUE
                         F=(S/A(I,I))*G
                         DO 3220 K=I,M
                             A(K,J)=A(K,J)+F*A(K,I)
 3220                    CONTINUE
 3200                CONTINUE
                  ENDIF
                  DO 3300 J=I,M
                      A(J,I)=A(J,I)*G
 3300             CONTINUE
               ELSE
                  DO 3400 J=I,M
                      A(J,I)=0.0
 3400             CONTINUE
               ENDIF
               A(I,I)=A(I,I)+1.0
 3000 CONTINUE
C      Diagonalization of the bidiagonal form
       DO 4000 K=N,1,-1                          ! loop over singular values
          DO 4100 ITS=1,30                       ! loop over allowed iterations
             FLAG = .FALSE.
             DO 4110 L=K,1,-1                    ! test for splitting
                NM=L-1                           ! RV1(1) is always 0
                IF ((ABS(RV1(L))+ANORM).EQ.ANORM) THEN
                   EXIT
                ELSEIF ((ABS(W(NM))+ANORM).EQ.ANORM) THEN
                   FLAG = .TRUE.
                   EXIT
                ENDIF
 4110           CONTINUE
             IF (FLAG) THEN
                C=0.0                            ! cancel RV1(L) if L>1
                S=1.0
                DO 4120 I=L,K
                   F=S*RV1(I)
                   RV1(I)=C*RV1(I)
                   IF ((ABS(F)+ANORM).EQ.ANORM) EXIT
                   G=W(I)
                   H=SQRT(F*F+G*G)
                   W(I)=H
                   H=1.0/H
                   C= (G*H)
                   S=-(F*H)
                   DO 4121 J=1,M
                      Y=A(J,NM)
                      Z=A(J,I)
                      A(J,NM)= (Y*C)+(Z*S)
                      A(J,I)=  (Z*C)-(Y*S)
 4121                 CONTINUE
 4120              CONTINUE
             ENDIF
             Z=W(K)
             IF (L.EQ.K) THEN                    ! convergence
                IF (Z.LT.0.0) THEN               ! singular value made non -ve
                   W(K)=-Z
                   DO 4130 J=1,N
                      V(J,K)=-V(J,K)
 4130                 CONTINUE
                ENDIF
                EXIT
             ENDIF
             IF (ITS.EQ.30) THEN
                ERR = -1
                RETURN
             ENDIF
             X=W(L)                              ! shift from bottom 2x2 minor
             NM=K-1
             Y=W(NM)
             G=RV1(NM)
             H=RV1(K)
             F=((Y-Z)*(Y+Z)+(G-H)*(G+H))/(2.0*H*Y)
```

```
              G=SQRT(F*F+1.0)
              F=((X-Z)*(X+Z)+H*((Y/(F+SIGN(G,F)))-H))/X
              C=1.0                                  ! next QR transformation
              S=1.0
              DO 4140 J=L, NM
                  I=J+1
                  G=RV1(I)
                  Y=W(I)
                  H=S*G
                  G=C*G
                  Z=SQRT(F*F+H*H)
                  RV1(J)=Z
                  C=F/Z
                  S=H/Z
                  F= (X*C)+(G*S)
                  G= (G*C)-(X*S)
                  H=Y*S
                  Y=Y*C
                  DO 4141 JJ=1,N
                      X=V(JJ,J)
                      Z=V(JJ,I)
                      V(JJ,J)= (X*C)+(Z*S)
                      V(JJ,I)= (Z*C)-(X*S)
 4141             CONTINUE
                  Z=SQRT(F*F+H*H)
                  W(J)=Z                             ! arbitrary rotation if Z=0
                  IF (Z.NE.0.0) THEN
                      Z=1.0/Z
                      C=F*Z
                      S=H*Z
                  ENDIF
                  F= (C*G)+(S*Y)
                  X= (C*Y)-(S*G)
                  DO 4142 JJ=1,M
                      Y=A(JJ,J)
                      Z=A(JJ,I)
                      A(JJ,J)= (Y*C)+(Z*S)
                      A(JJ,I)= (Z*C)-(Y*S)
 4142             CONTINUE
 4140         CONTINUE
              RV1(L)=0.0
              RV1(K)=F
              W(K)=X
 4100     CONTINUE
 4000 CONTINUE
      RETURN
      END
* * * *
```

```
     +    S,TMP
      PARAMETER (NMAX=100)                          ! max anticipated value of N
      DIMENSION
     +    TMP(NMAX)
C Executable Code
      DO 1000 J=1,N
          S=0.0
          IF(W(J).NE.0.0)THEN                       ! nonzero result only if wj is nonzero
              DO 1100 I=1,M
                  S=S+U(I,J)*B(I)
 1100         CONTINUE
              S=S/W(J)                              ! this is the divide by wj
          ENDIF
          TMP(J)=S
 1000 CONTINUE
      DO 2000 J=1,N                                 ! matrix multiply by V to get answer
          S=0.0
          DO 2100 JJ=1,N
              S=S+V(J,JJ)*TMP(JJ)
 2100     CONTINUE
          X(J)=S
 2000 CONTINUE
      RETURN
      END
* * * *
```

Appendix D Program FADD: Fuzzy Analysis of Discontinuity Data

As part of this work a computer program, which applies the fuzzy *c*-means, fuzzy covariance and fuzzy maximum likelihood estimation algorithms, together with a number of cluster validity functions, to the problem of identifying clusters in orientation data has been written by the Author.

Currently, FADD (Fuzzy Analysis of Discontinuity Data) is designed to execute on microcomputers running MS-DOS or a similar operating system. The version included in this Appendix is analytically identical to that used to generate the results presented in Chapters 6 and 7. Operationally, it is much simplified, with all of the user interaction functionality (runtime setting of file names, checking of input data, analysis and output options) removed to save space.

D.1 Operation of FADD

The functional operation of FADD is indicated via the flowchart of Figure D.1. Flow of control is primarily top to bottom, secondarily left to right. Detailed operational descriptions of each of the modules forming FADD may be found in the code at the end of this Appendix. The functions they perform are briefly outlined in the following paragraphs.

FADD main module, containing array and common block declarations.

PRELIM setting of filenames, output specification, analysis options, reading and checking of data file.

ANALYS controls main analysis, including initialisation of fuzzy membership array.

OBSDCS computes direction cosines of the input orientation data.

LOPLOT low resolution hemispherical projection plot using ASCII symbols.

COORD computes page coordinates of observations.

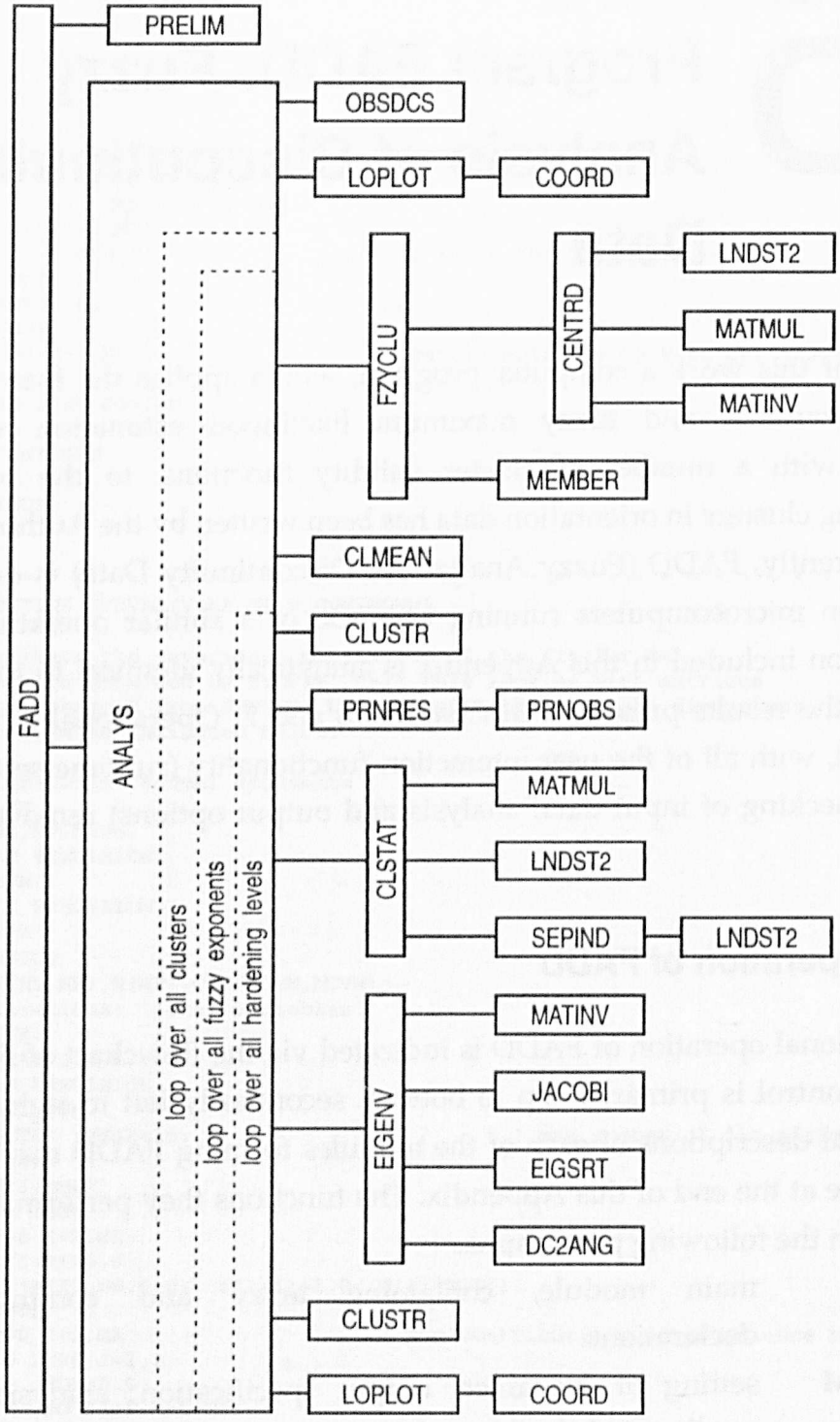


Figure D.1 Flowchart of program FADD.

FZYCLU	cycles through iterative FCM, FCV and FMLE clustering analyses.
CENTRD	computes cluster centroids.
LNDST2	computes square of norm-induced distance.
MATMUL	matrix multiplication.
MATINV	matrix inversion.
MEMBER	computes membership values for all data.
CLMEAN	transfer centroids to main orientation data array as both components and orientations.
CLUSTR	assigns data to clusters based on generalised hardening.
PRNRES	produce the data assignment output report.
PRNOBS	print observations assigned to a given cluster.
CLSTAT	compute cluster statistics.
SEPIND	compute separation index.
EIGENV	compute eigenvalues, eigenvectors and corresponding orientations for both hard and fuzzy scatter matrices.
JACOBI	eigenvalues and eigenvectors by Jacobi transformations.
EIGSRT	sort eigenvalues into descending order.
DC2ANG	transform direction cosines into angular orientations.

D.2 Input and Output Files

FADD requires an input file containing the lines title, number of data, and repeated data lines in the form of azimuth/dip, as ASCII text. An example FADD data file is:

```
Example data file for FADD
4
012,34
123,45
234,56
345,67
```

This must be saved as the file FADD.DAT. If FADD is unable to either find or access the file, it terminates with an appropriate error message.

Ouput from FADD is sent to a file FADD.OUT. This output is verbose, consisting of a listing of the data and a hemispherical projection, and then for

each combination of number of clusters, fuzzy exponent and sharpening level, the following:

a listing of the termination variables used in the iteration process;
a listing of the data assigned to each cluster;
cluster validity measures;
hard and fuzzy scatter matrices (as appropriate), together with their eigenvalues, eigenvectors and associated orientations; and
a hemispherical projection plot indicating the data assigned to the clusters at different hardening levels.

D.3 Listing of Program FADD

A complete listing for FADD is given here. Although FADD has been written in FORTRAN 77, some common extensions to the language that have been included by the manufacturer of the compiler have been used, particularly where they simplify the code. The extensions used in FADD are:

(i) the data types `LOGICAL*1` and `INTEGER*2` have been used where appropriate to conserve memory;

(ii) `DO WHILE....END DO` loops have been included when iterations are required;

(iii) the intrinsic function `INDEX(`*`charA`*`,`*`charB`*`)`, which returns the location of substring *`charA`* within string *`charB`*;

(iv) `INCLUDE` files, referenced in the listing of FADD as, for example, `INCLUDE 'USEFUL.FOR'`, and which are automatically inserted in the code at compile time, have been used for all `COMMON` declarations in order to reduce the bulk of the code;

(v) comments, beginning with an exclamation mark, following and on the same line as, program statements.

The complete set of `INCLUDE` files required for FADD follows.

```
C Include file PARAMETR.FOR
      INTEGER
     +          K1,K2,K3
      PARAMETER
     +     (K1=1000,                          ! max. no. feature vectors
     +      K2=6,                             ! max. no. feature dimensions
     +      K3=20)                            ! max. no. clusters
C Include file CLSTYP.FOR
      LOGICAL*1
     +     FCM,FCV,FMLE
      COMMON
     +     /CLSTYP/FCM,FCV,FMLE
C Include file CLUDAT.FOR
      REAL
```

```
     +     SSP,FCOVAR,DETF,APPROB
      DIMENSION
     +     SSP(K2,K2,K3),FCOVAR(K2,K2,K3),DETF(K3),APPROB(K3)
      COMMON
     +     /CLUDAT/SSP,FCOVAR,DETF,APPROB
C Include File: DEVICE.FOR
      INTEGER*2
     +     DEVICE
      COMMON
     +     /DEVICE/DEVICE(4)
C Include file FZYDAT.FOR
      REAL
     +     IDENT,FEXP,FEXPI
      DIMENSION
     +     IDENT(K2,K2)
      COMMON
     +     / FZYDAT/IDENT,FEXP,FEXPI
C Include File: MAXSIZ.FOR
      INTEGER
     +     MAXOBS,MAXSCN
      COMMON/MAXSIZ/
     +     MAXOBS,MAXSCN
C Include File: MESAGE.FOR
      CHARACTER
     +     PLOTYP*7,PROTYP*5,HEMTYP*5
      COMMON/MESAGE/
     +     PLOTYP,PROTYP,HEMTYP
C Include file PAGE.FOR
      CHARACTER
     +     PAGE*1
      COMMON
     +     /PAGE/PAGE(160,90)
C Include File: PAGSIZ.FOR
      REAL
     +     COLPIN,LINPIN,COLWID,LINEHT
      INTEGER
     +     MAXCOL,MAXROW,MIDCOL,MIDROW
      COMMON/PAGSIZ/MAXCOL,MAXROW,COLPIN,LINPIN,
     +              COLWID,MIDCOL,LINEHT,MIDROW
C Include File: PROJ.FOR
      REAL
     +     RADIUS
      LOGICAL*1
     +     FLGOUT,FLGPLT,FLGPRO,FLGHEM
      COMMON/  PROJ/
     +     FLGOUT,FLGPLT,FLGPRO,FLGHEM,RADIUS
C Include file TXTBLK.FOR
      CHARACTER
     +     TEXT*68
      COMMON
     +     /TXTBLK/TEXT
C Include File: USEFUL.FOR
      REAL
     +     DEGRAD,RADEG,PI
      COMMON
     +     /USEFUL/DEGRAD,RADEG,PI
```

The code for FADD follows. Each procedure (subroutine or function) is separated from its neighbour by a line of asterisks, and all procedures follow the same structure. But for subroutines to compute matrix inverses, eigenvalues and eigenvectors, this code is complete and may be executed as is. These subroutines have not been included as they are readily available in subroutine libraries or in print (see, for example, Press, et al. [102]).

```
      PROGRAM FADD
C
C     Fuzzy Analysis of Directional Data
C
C     J.P. Harrison, Imperial College
C
      INCLUDE 'PARAMETR.FOR'
C Type Declarations: Common Blocks
      REAL
     +    A
      INTEGER*2
     +    B
      COMMON/ ARRAY/
     +    A(K1,2:7),B(K1,2)
      INCLUDE 'MAXSIZ.FOR'
      INCLUDE 'PAGE.FOR'
      INCLUDE 'PAGSIZ.FOR'
      INCLUDE 'MESAGE.FOR'
      INCLUDE 'PROJ.FOR'
      INCLUDE 'DEVICE.FOR'
      INCLUDE 'USEFUL.FOR'
C Type Declarations: Local Variables
      INTEGER
     +    TOTOBS
C Executable Code
      MAXOBS=K1
      MAXSCN=K4
      WRITE(*,100)
  100 FORMAT(                                                          /
     + '                    FADD:  Fuzzy Analysis of Directional Data'  /
     + '                      Research and Development Version 1992'    /
     +                                                                  /
     + '                                    J.P. Harrison'              /
     +                                                                  /
     + ' FADD partitions directional data using the three fuzzy objecti'
     +                                                    've function'/
     + ' algorithms: fuzzy c-means; fuzzy covariance; fuzzy maximum lik'
     +                                                       'elihood.'/
     + ' Data input is via an ASCII file of azimuth,dip angle data pair'
     +                                                  's. Either planar'/
     + ' (i.e. dip direction/dip angle) or axial data may be partitione'
     +                                                            'd.'/
     + ' Results are output to an ASCII file, and optionally may be:'   /
     + '              i) hemispherical projection plots;'               /
     + '             ii) lists of observations forming clusters;'       /
     + '            iii) descriptive statistics of cluster orientation,'
     +                                                        ' and;'/
     + '             iv) descriptive statistics of partition validity.')
      CALL PRELIM(A,B,K1,TOTOBS)
      CALL ANALYS(A,B,K1,TOTOBS)
      STOP' Analysis complete'
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE PRELIM(A,B,NP,TOTOBS)
C
C PERFORMS VARIOUS PRELIMINARY OPERATIONS:
C     Setting i/o device numbers;
C     Opening input & output files;
C     Setting o/p specification;
C     Reading data file;
C     Setting analysis parameters
C
C Type Declarations: Formal Parameters
      REAL
     +    A
      INTEGER*2
     +    B
      INTEGER
     +    NP,TOTOBS
      DIMENSION
     +    A(NP,*),B(NP,*)
```

```
C Type Declarations: Common Blocks
      INCLUDE 'DEVICE.FOR'
      INCLUDE 'USEFUL.FOR'
      INCLUDE 'PROJ.FOR'
      INCLUDE 'PAGSIZ.FOR'
      INCLUDE 'MESAGE.FOR'
C Type Declarations: Local Variables
      REAL
     +    C1,C2
      INTEGER
     +    BORDER,IOS,L1
      LOGICAL*1
     +    ECHO
      CHARACTER
     +    TITLE*80
C Executable Code
      DEVICE(1) = 5                                  !DOS standard input
      DEVICE(2) = 7                                  !user named input
      DEVICE(3) = 6                                  !DOS standard output
      DEVICE(4) = 8                                  !user named output
      PI = 4.0*ATAN(1.0)
      DEGRAD = PI/180.0
      RADEG = 180.0/PI
      RADIUS=90.
      LINPIN=6.
      COLPIN=10.
      BORDER=1
      COLWID=25.4/COLPIN
      LINEHT=25.4/LINPIN
      C1=RADIUS/COLWID
      C2=RADIUS/LINEHT
      MIDCOL=INT(C1)+BORDER
      MIDROW=INT(C2)+BORDER
      MAXCOL=BORDER+2+INT(2.* C1)
      MAXROW=BORDER+2+INT(2.* C2)
C                      FLGOUT       FLGPRO       FLGPLT    FLGHEM
C          .TRUE.    plot required equal angle   normals   lower
C          .FALSE.      no plot    equal area     dips     upper
C
      FLGOUT=.TRUE.
      FLGPRO=.TRUE.
      FLGPLT=.TRUE.
      FLGHEM=.TRUE.
      PROTYP='area '
      HEMTYP='upper'
      PLOTYP='dips    '
      OPEN(DEVICE(2),
     +    FILE='FADD.DAT',IOSTAT=IOS,FORM='FORMATTED',STATUS='OLD')
      IF (IOS.NE.0) STOP 'FADD.DAT not found'
      OPEN (DEVICE(4),FILE='FADD.OUT',IOSTAT=IOS,STATUS='UNKNOWN')
      READ(DEVICE(2),'(A)',IOSTAT=IOS) TITLE
      IF (IOS.NE.0) STOP 'Unable to read FADD.DAT'
      DO 1000 L1=3,4
          WRITE(DEVICE(L1),*) TITLE                  ! print to file, echo to screen
 1000 CONTINUE
      READ(DEVICE(2),*) TOTOBS                       ! no. of observations
      DO 2000 L1=1,TOTOBS
              READ(DEVICE(2),*) A(L1,2),A(L1,3)
              B(L1,1)=L1
              B(L1,2)=0
 2000 CONTINUE
      ECHO=.TRUE.
      CALL PRNOBS(A,NP,TOTOBS,B,2,0,ECHO)
      RETURN
      END
* * * *
```

```
C Type Declarations: Formal Parameters
      REAL
     +    A
      INTEGER*2
     +    B
      INTEGER
     +    NP,N
      DIMENSION
     +    A(NP,*),B(NP,*)
C Type Declarations: Common Blocks
      INCLUDE 'TXTBLK.FOR'
      INCLUDE 'CLUDAT.FOR'
      INCLUDE 'DEVICE.FOR'
      INCLUDE 'FZYDAT.FOR'
      INCLUDE 'CLSTYP.FOR'
C Type Declarations: Local Variables
      INTEGER
     +    K,L0,L1,L2,M,NI
      REAL
     +    U,X,V,ALPHA
      LOGICAL*1
     +    OK,PLTCEN,PLTDAT
      CHARACTER
     +    LEGEND*19
      DIMENSION
     +    U(K3,K1),X(K2,K1),V(K2,K3),ALPHA(6)
      PARAMETER
     +    (PLTDAT=.TRUE.)
      DATA
     +     ALPHA(1)/0.0/, ALPHA(2)/0.5/, ALPHA(3)/0.6/,
     +     ALPHA(4)/0.7/, ALPHA(5)/0.8/, ALPHA(6)/0.9/
C Executable code
      M = 3                                          ! dimension of cluster space
      CALL STATUS(' ')
      CALL STATUS('Calculating direction cosines...')
      DO 1000 L1=1,N
          CALL OBSDCS(A(L1,4),A(L1,5),A(L1,6),
     +                          (180.+A(L1,2)),(90.-A(L1,3)))
 1000 CONTINUE
      CALL STATUS('Plotting all data...')
      PLTCEN = .FALSE.
      CALL LOPLOT(A,B,NP,N,U,K3,2,0,0,PLTDAT,PLTCEN,'all data')
      CALL STATUS('                        Finished.')
C     Copy direction cosines to array X,
C     write sequential i.d. into col1 of B for subsequent unsorting
      DO 2000 L1=1,N
          DO 2100 L2=1,M
              X(L2,L1)=A(L1,L2+3)
              B(L1,1)=L1
 2100     CONTINUE
 2000 CONTINUE
C     initialize identity matrix
      DO 3000 L1=1,M
          DO 3100 L2=1,M
              IDENT(L2,L1)=0.0
 3100     CONTINUE
          IDENT(L1,L1)=1.0
 3000 CONTINUE
      DO 4000 L0=2,5                                 ! fuzzy exponent: 2 thro 5
          FEXP=REAL(L0)
          FEXPI=1.0/(FEXP-1.0)
          DO 4100 K=2,6                              ! clusters: 2 thro 6
              DO 4110 L1=1,N                         ! initialize memberships,
                  DO 4111 L2=1,K                     !
                      U(L2,L1)=0.0                   !
 4111                 CONTINUE                       ! ensuring
                  U(1+MOD(L1,K),L1)=1.0              ! all rows of U are different
 4110             CONTINUE
              PLTCEN = .TRUE.
              DO 4120 L1=1,3
                  WRITE(TEXT,'(A)') 'Evaluating      ,     clusters:'
```

```
                IF (L1.EQ.1) THEN
                    FCM=.TRUE.
                    FCV=.FALSE.
                    WRITE(TEXT(12:15),'(A)') ' FCM'
                    LEGEND = 'FCM clustered data '
                ELSE IF (L1.EQ.2) THEN
                    FCM=.FALSE.
                    FCV=.TRUE.
                    WRITE(TEXT(12:15),'(A)')' FCV'
                    LEGEND = 'FCV clustered data '
                ELSE
                    FCM=.FALSE.
                    FCV=.FALSE.
                    WRITE(TEXT(12:15),'(A)')'FMLE'
                    LEGEND = 'FMLE clustered data'
                ENDIF
                FMLE=.NOT.(FCM.OR.FCV)
                WRITE(TEXT(18:19),'(I2)')K
C               perform the clustering:
                CALL FZYCLU(U,X,V,K,M,N,K3,K2,K1,OK,NI)
C               write cluster means into A and B
                CALL CLMEAN(V,A,B,NP,N,K,M,K3,K2)
                DO 4121 L2=1,6                    ! analyse a-hard partitions
                    WRITE(DEVICE(4),100) ALPHA(L2)
  100               FORMAT('  Alpha-hard =',F4.1)
                    CALL STATUS('Clustering...')
                    CALL CLUSTR (U,B,ALPHA(L2),N,K,K1,K3)
                    CALL STATUS('Printing results...')
                    CALL PRNRES(A,B,NP,N,K)
                    IF (OK) THEN
                        CALL STATUS
     +                      ('Computing cluster statistics....')
                        CALL CLSTAT(U,X,V,B,N,M,K,NP,K2,K3,NI)
                        CALL EIGENV(K,M)
                    ELSE
                        WRITE(DEVICE(4),200)
  200                   FORMAT(//21('*')//
     +                      ' Negative determinant'//21('*')//)
                    END IF
 4121             CONTINUE
                CALL CLUSTR (U,B,ALPHA(1),N,K,K1,K3)
                CALL LOPLOT
     +              (A,B,NP,N,U,K3,2,1,K,PLTDAT,PLTCEN,LEGEND)
 4120         CONTINUE
 4100     CONTINUE
 4000 CONTINUE
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE OBSDCS (L,M,N,AZ,DIP)
C
C CALCULATE THE DIRECTION COSINES OF AN OBSERVATION
C
C Type Declarations: Formal Parameters
      REAL
     +    L,M,N,AZ,DIP
C Type Declarations: Common Blocks
      INCLUDE 'USEFUL.FOR'
C Type Declarations: Local Variables
      REAL
     +    ALPHA,BETA,COSB
C Executable Code
      ALPHA = AZ*DEGRAD
      BETA = DIP*DEGRAD
      COSB=COS(BETA)
      L=SIN(ALPHA)*COSB
      M=COS(ALPHA)*COSB
      N=SIN(BETA)
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
      SUBROUTINE CLSTAT(U,X,V,B,N,M,K,NP,MP,KP,NI)
C
C     CLuster STATistics
C
C          U(k,n) = fuzzy membership array
C          X(m,n) = feature vector array
C          V(m,k) = cluster centroid array
C               N = number of feature vectors
C               K = number of clusters
C               M = number of feature dimensions
C
C          Input: U,X,V,N,M,K,NP,MP,KP
C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*     METHODOLOGY
*
*     loop over all clusters
*         loop over all data
*             add to partition coefficient
*             add to partition entropy
*             comp. squared Mahalonobis distance
*             if distance < 1 then
*                 add to cluster partition density
*                 add to global partition density
*                 add to SSP matrix for the cluster
*             end if
*         end loop
*         comp. Si/det(Fi), add to avg. partition density
*         add det(Fi) to FHVOL
*     end loop
*     comp. avg. partition density
*     comp. partition density
*     comp. partition coefficient and entropy
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE 'PARAMETR.PRM'
C Type Declarations: Formal Arguments
      REAL
     +    U,X,V
      INTEGER
     +    N,K,M,NP,KP,MP,NI
      INTEGER*2
     +    B
      DIMENSION
     +    U(KP,NP),X(MP,NP),V(MP,KP),B(NP,*)
C Type Declarations: Common Blocks
      INCLUDE 'CLUDAT.BLK'
      INCLUDE 'DEVICE.BLK'
      INCLUDE 'CLSTYP.BLK'
C Type Declarations: Local Variables
      INTEGER
     +    K5
      PARAMETER
     +    (K5=K2*K2)
      INTEGER
     +    L1,L2,L3,C
      REAL
     +    PARCO,NPARCO,PAREN,PEX,SC,SI,MPC,FHVOL,PD,DPA,
     +    IDENT,RTDETF,DIST,SENSE,THISU,MINU,CX,TEMP,CENMEM,
     +    CLURAD,LNDST2,DST2,PROEXP
      DIMENSION
     +    CX(K2),TEMP(K2,K2),CENMEM(K3),SI(0:K3),IDENT(K2,K2),CLURAD(K2)
      EXTERNAL
     +    MATMUL,LNDST2,SEPIND,PROEXP
      DATA
     +    ((IDENT(L1,L2),L1=1,K2),L2=1,K2)/K5*0.0/,
     +    (IDENT(L1,L1),L1=1,K2)/K2*1.0/
C Executable Code
      PARCO = 0.0                                ! partition coeff
      PAREN = 0.0                                ! partition entropy
```

```
      SC = 0.0                                    ! separation coefficient
      MPC = 0.0                                   ! modified parco
      FHVOL = 0.0                                 ! fuzzy hypervolume
      PD = 0.0                                    ! partition density
      DPA = 0.0                                   ! avg. partition density
      MINU = 1.0
      DO 1000 L1=1,K                        ! loop over all clusters
          CENMEM(L1) = 0.0                        ! S for this cluster = zero
          CLURAD(L1) = 0.0                        ! cluster radius = 0
          DO 1100 L2=1,M
              DO 1200 L3=1,M
                  SSP(L3,L2,L1)=0.0               ! SSP matrix for the cluster = zero
 1200         CONTINUE
 1100     CONTINUE
 1000 CONTINUE
      DO 2000 L1=1,N                              ! loop over all data
          C=B(L1,2)                               ! this datum's cluster
          IF (C.NE.0) THEN
              THISU = ABS(U(C,L1))
              SENSE = SIGN(1.0,U(C,L1))
              DO L2 = 1,M
                  CX = SENSE*X(L2,L1)
              ENDDO
              DST2 = LNDST2(CX,V(1,C),IDENT,K2,M)
              DST2 = MIN(DST2,4.0-DST2)
              CLURAD(C) = MAX(CLURAD(C),THISU*THISU*DST2)
              MPC = MPC + THISU
              MINU = MIN(MINU,THISU)
              PARCO=PARCO+THISU**2
              IF (THISU.NE.0.0) THEN
                  PAREN=PAREN+THISU*LOG(THISU)
              ENDIF
              CALL MATMUL(X(1,L1),X(1,L1),TEMP,M,1,M,K2,1,K2)
C                 to form X.trans(X) as TEMP (M x M matrix), and add all
C                 elements of it to the SSP matrix for the cluster. Sign
C                 transfer of X not needed as all elements of SSP are
C                 effectively to power 2.
              DO 2100 L2=1,M
                  DO 2110 L3=1,M
                      SSP(L3,L2,C)=SSP(L3,L2,C)+TEMP(L3,L2)
 2110             CONTINUE
 2100         CONTINUE
              IF (FCV.OR.FMLE) THEN
                  DO 2200 L2=1,K
C                     loop over all mathematical clusters
                      SENSE=SIGN(1.0,U(L2,L1))
C                     determine components of X by looping over all
C                     dimensions, taking sense of X into account
                      DO 2210 L3=1,M
                          CX(L3)=SENSE*X(L3,L1)
 2210                 CONTINUE
C                     determine squared Mahalanobis distance
                      DIST=LNDST2(CX,V(1,L2),FCOVAR(1,1,L2),MP,M)
                      IF (DIST.LT.1.0) THEN
C                         the squared Mahalanobis distance shows this datum to
C                         be within one sdev of the mean of this cluster, so
C                         add this membership to this cluster partition density
                          CENMEM(L2)=CENMEM(L2)+ABS(U(L2,L1))
C                         and add it to global partition density
                          PD=PD+ABS(U(L2,L1))
                      END IF
 2200             CONTINUE
              END IF
          ENDIF
 2000 CONTINUE
      DO L1=1,K
          CLURAD(L1) = SQRT(CLURAD(L1))           ! finish cluster radius calc.
      ENDDO
      IF (FCM.OR.FCV) THEN
          CALL STATUS('Computing proportion exponent....')
          PEX = PROEXP(U,B,N,K,NP,KP)
```

```
      ENDIF
      IF (FCV.OR.FMLE) THEN
          DO 3000 L1=1,K
              RTDETF=SQRT(DETF(L1))              ! compute Si/det(Fi) and add
              DPA=DPA+CENMEM(L1)/RTDETF          ! to average partition density
              FHVOL=FHVOL+RTDETF                 ! and add det(Fi) to FHVOL
 3000     CONTINUE
          DPA=DPA/K                              ! avg partition density
          PD=PD/FHVOL                            ! partition density
      END IF
      PARCO=PARCO/N                              ! partition coefficient
      NPARCO = (PARCO*REAL(K)-1.0)/(REAL(K)-1.0) ! normalised partition coeff.
      PAREN=-PAREN/N                             ! partition entropy
      MPC = (MPC/REAL(N) + MINU)/2.0             ! modified partition coeff.
      MPC = (MPC*REAL(K)-1.0)/(REAL(K)-1.0)
      CALL STATUS('Computing separation index....')
      CALL SEPIND(X,B,SI,K,M,N,MP,NP)
      CALL STATUS('Computing separation coefficient....')
      DO L1=1,K-1
          DO L2=L1+1,K
              DST2 = LNDST2(V(1,L1),V(1,L2),IDENT,K2,M)
              DST2 = MIN(DST2,4.0-DST2)          ! ensure acute angle is used
              SC = MAX(SC,(CLURAD(L1)+CLURAD(L2))/SQRT(DST2))
          ENDDO
      ENDDO
      SC = 1.0-SC
C     write the results to the file
      WRITE(DEVICE(4),10)
      IF (FCM) THEN
          WRITE(DEVICE(4),20) K,NI,PARCO,NPARCO,PAREN,PEX,SC,SI(0),MPC
      ELSE
          WRITE(DEVICE(4),20)
     +        K,NI,PARCO,NPARCO,PAREN,PEX,SC,SI(0),MPC,FHVOL,PD,DPA
      ENDIF
      WRITE(DEVICE(4),30) (SI(L1), L1=1,K)
      RETURN
   10 FORMAT(
     +   /'  C its      pc       npc       pe       pex       sc       si',
     +    '       mpc       fhv       pd       apd')
   20 FORMAT(1X,I2,1X,I3,10(1X,F8.3))
   30 FORMAT(' Cluster separation indices:'/,10(1X,F7.3))
      END
* * * *
```

```
     +        L1,INT(A(N+L1,2)),INT(A(N+L1,3))    ! print the centroid
 1000 CONTINUE
      CALL PRNOBS(A,NP,N,B,2,0,ECHO)              ! print unassigned data
      RETURN
  100 FORMAT(' Number of clusters=',I3)
  200 FORMAT(' Centroid ',I3,':     ',I4,'/',I2)
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE PRNOBS(A,NP,NOBS,B,N1,N2,FLGECH)
C
C PRINTS THE AZIMUTH/DIP OF THOSE OBS WHOSE VALUE IN COL N1 IS N2
C AND IF FLGECH IS TRUE, WILL ECHO TO THE SCREEN
C
C Type Declarations: Formal Parameters
      REAL
     +    A
      INTEGER*2
     +    B
      INTEGER
     +    NP,NOBS,N1,N2
      LOGICAL*1
     +    FLGECH
      DIMENSION
     +    A(NP,*),B(NP,*)
C Type Declarations: Common Blocks
      INCLUDE 'DEVICE.FOR'
C Type Declarations: Local Variables
      INTEGER
     +    K,N,LODEV,L1,L2,L3,C,D
      DIMENSION
     +    C(10),D(10)
C Executable Code
      K=0
      N=0
      IF(FLGECH)THEN
          LODEV=3
      ELSE
          LODEV=4
      ENDIF
      DO 1000 L1=LODEV,4
          IF (N2.EQ.0) THEN
              WRITE(DEVICE(L1),100)               ! title for unclustered data
          ELSE
              WRITE(DEVICE(L1),200) N2            ! title for clustered data
          ENDIF
 1000 CONTINUE
      DO 2000 L1=1,NOBS
          IF (B(L1,N1).EQ.N2) THEN
              K = K+1
              N = N+1
              C(K)=INT(A(L1,2))
              D(K)=INT(A(L1,3))
              IF (K.EQ.10) THEN
                  K=0
                  DO 2100 L2=LODEV,4
                      WRITE(DEVICE(L2),300) (C(L3),D(L3),L3=1,10)
 2100             CONTINUE
              ENDIF
          ENDIF
 2000 CONTINUE
      DO 3000 L1=LODEV,4
          IF (K.GT.0) THEN
              WRITE(DEVICE(L1),300) (C(L2),D(L2),L2=1,K)
          ENDIF
          WRITE(DEVICE(L1),400) N
 3000 CONTINUE
      RETURN
  100 FORMAT(/
     +  ' Unclustered data:    Format is dip direction/dip angle')
  200 FORMAT(/
     +  ' Data in cluster ',I2,':  Format is dip direction/dip angle')
```

```
  300 FORMAT(10(I4,'/',I2))
  400 FORMAT(' Number of data =',I4)
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUBROUTINE LOPLOT
     +     (A,B,NP,NOBS,U,KP,N1,N2,N3,PLTDAT,PLTCEN,TEXT)
C
C     low resolution plot using the printer of data whose value in column N1
C     of B equals N2 thro' N3. The string TEXT is added as a title,
C     if PLTDAT is true the data are plotted and if PLTCEN is true the
C     centroids are plotted.
C Type Declarations: Formal Parameters
      REAL
     +     A,U
      INTEGER*2
     +     B
      INTEGER
     +     NP,KP,NOBS,N1,N2,N3
      LOGICAL*1
     +     PLTDAT,PLTCEN
      CHARACTER
     +     TEXT*(*)
      DIMENSION
     +     A(NP,*),B(NP,*),U(KP,*)
C Type Declarations: Common Blocks
      INCLUDE 'DEVICE.FOR'
      INCLUDE 'PAGE.FOR'
      INCLUDE 'MESAGE.FOR'
      INCLUDE 'PAGSIZ.FOR'
      INCLUDE 'USEFUL.FOR'
      INCLUDE 'PROJ.FOR'
C Type Declarations: Local Variables
      REAL
     +     X,Y,C1
      INTEGER
     +     L1,L2,K1,K2,COL,ROW,BEG,END,CLULO,CLUHI,C,LEVEL
      CHARACTER
     +     PLOT*8,UCASE*11,LCASE*11,UC*1,LC*1
      DIMENSION
     +     UC(0:10),LC(0:10)
      EQUIVALENCE
     +     (UCASE,UC(0)), (LCASE,LC(0))
      DATA
     +     UCASE/'#JIHGFEDCBA'/, LCASE/'+jihgfedcba'/
      COL(X)=INT(X/COLWID)+MIDCOL                ! Integer statement functions
      ROW(Y)=INT(Y/LINEHT)+MIDROW                ! for plot calcs
C Executable Code
      IF(.NOT.FLGOUT) RETURN                     ! because no plot required
      DO 1000 L1=1,MAXROW
          DO 1100 L2=1,MAXCOL
              PAGE(L2,L1)=' '                    ! blank the page
 1100     CONTINUE
 1000 CONTINUE
      DO 2000 L1=10,360,10                       ! add boundary, north & centre
          C1=DEGRAD*L1
          K1=COL(RADIUS*SIN(C1))
          K2=ROW(RADIUS*COS(C1))
          PAGE(K1,K2)='x'
 2000 CONTINUE
      PAGE(K1,K2+1)='N'
      K2=ROW(0.)
      PAGE(K1,K2)='*'
      IF (FLGPLT) THEN                           ! assemble the title...
          PLOT=' normals'
      ELSE
          PLOT='s        '
      ENDIF
      WRITE(DEVICE(4),100)PROTYP,HEMTYP,PLOT,TEXT! ... and print it
  100 FORMAT(/
     +    '          equal ',A,' ',A,' hemispherica',
     +                      'l projection of observation',A/
```

```
     +    '                              for ',A)
          N2 = ABS(N2)
          N3 = ABS(N3)
          CLULO = MIN(N2,N3)
          CLUHI = MAX(N2,N3)
          IF (.NOT.PLTCEN) THEN                          ! centroids not reqd
              BEG = 1
              END = NOBS
          ELSE                                           ! centroids reqd
              END = NOBS + CLUHI
              IF (.NOT.PLTDAT) THEN                      ! centroids only reqd
                  BEG = NOBS + CLULO
              ENDIF
          ENDIF
          DO 3000 L1=BEG,END                             ! loop over data & centroids
              C = B(L1,N1)
              IF (C.GE.CLULO .AND. C.LE.CLUHI) THEN
                  IF (C.EQ.0) THEN
                      LEVEL = 0
                  ELSE
                      LEVEL = MIN(10,1+INT(10.0*ABS(U(C,L1))))
                  ENDIF
                  CALL COORD(X,Y,A,NP,L1)
                  K1=COL(X)
                  K2=ROW(Y)
                  IF (PAGE(K1,K2).EQ.' ') THEN
                      PAGE(K1,K2)=LC(LEVEL)
                  ELSE
                      PAGE(K1,K2)=UC(LEVEL)
                  ENDIF
                  IF (L1 .GT. NOBS) THEN                 ! these are centroids
                      PAGE(K1,K2)=CHAR(48+MOD(L1-NOBS,10))
                  ENDIF
              ENDIF
     3000 CONTINUE
          DO 4000 L1=MAXROW,1,-1                         ! print the assembled page
              WRITE(DEVICE(4),*)(PAGE(L2,L1),L2=1,MAXCOL)
     4000 CONTINUE
          WRITE(DEVICE(4),200) COLPIN,LINPIN             ! print the circularity text
      200 FORMAT(/
         +  ' Projection will be circular if printed at ',F4.1,
         +                      ' chars/inch and ',F4.1,' lines/inch'/)
          RETURN
          END
* * * *
```

```
                MEMB=ABS(U(L2,L1))
                IF (MEMB.GT.MAXVAL) THEN            ! biggest so far?
                    MAXVAL=MEMB
                    CLUNUM=L2
                ENDIF
 1100       CONTINUE
            IF (MAXVAL.LT.ALPHA) CLUNUM=0           ! test against alpha-level
            B(L1,2)=CLUNUM
 1000   CONTINUE
        RETURN
        END
* * * *
```

```
C Type Declarations: Local Variables
      REAL
     +    ALPHA,BETA
      INTEGER
     +    L1,L2
      LOGICAL*1
     +    PLANE
C Executable Code
      PLANE=.TRUE.            ! set the plane/normal switch for DC2ANG
      DO 1000 L1=1,K                              ! copy all cluster centroids
          DO 1100 L2=1,M                          ! copy all components
              A(L1+N,L2+3)=V(L2,L1)
 1100     CONTINUE
          CALL DC2ANG(ALPHA,BETA,V(1,L1),V(2,L1),V(3,L1),PLANE)
          A(L1+N,2)=ALPHA
          A(L1+N,3)=BETA
          B(L1+N,2)=INT2(L1)                      ! set cluster number in B
 1000 CONTINUE
      RETURN
      END
* * * *
```

```
            WRITE(DEVICE(4),
     +              '(///'' Cluster fuzzy scatter matrices '',
     +                  ''eigenvalues and eigenvectors:'')')
          ELSE
            MATTYP='Fuzzy covariance matrix'
            WRITE(DEVICE(4),
     +              '(///'' Cluster fuzzy covariance matrices '',
     +                  ''eigenvalues and eigenvectors:'')')
          ENDIF
          DO 1100 L2=1,K
C           loop over all clusters
            IF (L1.EQ.1) THEN
C             assemble SSP matrix for eigensolving
              DO 1110 L3=1,M
                DO 1111 L4=1,M
                  A(L4,L3)=SSP(L4,L3,L2)
 1111           CONTINUE
 1110         CONTINUE
            ELSE
C             find inverse of inverse of fuzzy scatter/covariance
C             matrix for eigensolving
              DO 1120 L3=1,M
                DO 1121 L4=1,M
                  TEMP(L4,L3)=FCOVAR(L4,L3,L2)
 1121           CONTINUE
 1120         CONTINUE
              CALL MATINV(TEMP,A,M,K2,DET,OK)
            ENDIF
            DO 1130 L3=1,3
              LINE=SPACES
              DO 1131 L4=1,3
                B=L4*10-9
                E=B+8
                WRITE(LINE(B:E),'(1P,G9.2)') A(L3,L4)
 1131           CONTINUE
              BLOCK(L3)=LINE
 1130       CONTINUE
            CALL JACOBI(A,M,K2,EVALUE,EVECTR,NROT,OK)
            IF (OK) THEN
              CALL EIGSRT(EVALUE,EVECTR,M,K2)
              DO 1140 L3=1,3
                LINE=BLOCK(L3)
                WRITE(LINE(33:41),'(1P,G9.2)') EVALUE(L3)
                DO 1141 L4=1,3
                  B=L4*10+34
                  E=B+8
                  WRITE(LINE(B:E),'(1P,G9.2)') EVECTR(L4,L3)
 1141           CONTINUE
                CALL DC2ANG(ALPHA,BETA,EVECTR(1,L3),
     +                EVECTR(2,L3),EVECTR(3,L3),PLANE)
                WRITE(LINE(75:80),'(I3,''/'',I2)')
     +                INT(ALPHA),INT(BETA)
                BLOCK(L3)=LINE
 1140         CONTINUE
            ELSE
              LINE=BLOCK(2)
              WRITE(LINE(38:58),'(''EIGENSYSTEM INSOLUBLE'')')
              BLOCK(2)=LINE
            ENDIF
            WRITE(DEVICE(4),100)L2,MATTYP
            WRITE(DEVICE(4),'(3(A/),/)') (BLOCK(L3),L3=1,3)
 1100     CONTINUE
 1000   CONTINUE
        RETURN
  100   FORMAT(
     +    /5X,'Cluster ',I2,10X,A,                                    /
     +    12X,'Matrix',14X,'Eigenvalues',9X,'Eigenvectors',11X,'Dirn.' /
     +    '+',27('-'),'+   +-------+  +',27('-'),'+  +----+'          )
        END
* * * *
```

```
C
C     Write a status message on the screen
C
C          Input: TEXT
C
C Type Declarations: Formal Parameters
      CHARACTER
     +    TEXT*(*)
C Type Declarations: Common Blocks
      INCLUDE 'DEVICE.FOR'
C Executable Code
      WRITE(DEVICE(3),'(6X,A)') TEXT
      RETURN
      END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      REAL FUNCTION SEPIND(X,B,NP,N,M,MP)
C
C     SEParation INDex
C
C          X(m,n) = feature vector array
C               N = number of feature vectors
C               M = number of feature dimensions
C
C          Input: X,N,M,NP,MP,KP
C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*     METHODOLOGY
*
*          maxdia = 0, mindis = huge
*          loop over all data
*              read assigned cluster
*              read vector
*              loop over remaining data
*                  compute distance from vector in store
*                  read assigned cluster
*                  if this is the same as the working cluster
*                      if this distance > maxdia, maxdia = this distance
*                  else
*                      if this distance < mindis, mindis = this distance
*                  endif
*              next datum
*          next datum
*          separation index = mindis/maxdia
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE 'PARAMETR.FOR'
C Type Declarations: Formal Parameters
      REAL
     +    X
      INTEGER
     +    NP,N,M,MP
      INTEGER*2
     +    B
      DIMENSION
     +    X(MP,*),B(NP,*)
C Type Declarations: Common Blocks
      INCLUDE 'FZYDAT.FOR'
C Type Declarations: Local Variables
      INTEGER
     +    L1,L2,C1,C2
      REAL
     +    MAXDIA,MINDIS,DST2,LNDST2,DSTNCE,TINY,HUGE
      EXTERNAL
     +    LNDST2
      PARAMETER
     +    (TINY=1.0E-9, HUGE=1.0E38)
C Executable Code
      MAXDIA = 0.0                                  ! max diam = zero
      MINDIS = 1.0E6                                ! min dist = huge
      DO 1000 L1=1,N-1                              ! loop over all but last datum
```

```
            C1=B(L1,2)                                  ! read assigned cluster
            IF (C1.NE.0) THEN
                DO 1100 L2=L1,N                         ! loop over remaining data
                    C2=B(L2,2)                          ! read assigned cluster
                    IF (C2.NE.0) THEN
                        DST2 = LNDST2(X(1,L1),X(1,L2),IDENT,K2,M)
                        DST2 = MIN(DST2,4.0-DST2)  ! ensure acute angle is used
                        DSTNCE = SQRT(DST2)
                        IF (C1.EQ.C2) THEN
                            MAXDIA = MAX(MAXDIA,DSTNCE)
                        ELSE
                            MINDIS = MIN(MINDIS,DSTNCE)
                        ENDIF
                    ENDIF
 1100           CONTINUE
            ENDIF
 1000 CONTINUE
      IF (MAXDIA.GT.TINY) THEN
          SEPIND = MINDIS/MAXDIA
      ELSE
          SEPIND = HUGE
      ENDIF
      RETURN
      END
* * * *
```

```
      INTEGER
     +    NP,N1
      DIMENSION A(NP,*)
C Type Declarations: Common Blocks
      INCLUDE 'PROJ.FOR'
C Type Declarations: Local Variables
      REAL
     +    L,M,N,C1,C2
C Executable Code
      L=A(N1,4)
      M=A(N1,5)
      N=A(N1,6)
      IF(.NOT.FLGPLT) THEN
C         calculate d.c.'s of obs, but
          IF(N.EQ.1.)THEN
C             this is a horizontal line, so
              CALL OBSDCS(L,M,N,A(N1,2),A(N1,3))
C             to calc the d.c.'s from the angles
          ELSE
C             calc the d.c.'s from those of the normal:
              C1=SQRT(1.-N*N)
              C2=N/C1
              L=-L*C2
              M=-M*C2
              N=C1
          ENDIF
      ENDIF
      IF(.NOT.FLGHEM) THEN                          ! upper hemispher
          L=-L
          M=-M
      ENDIF
      C1=1.+N                                       ! scale factor
      IF(FLGPRO) C1=SQRT(C1)                        ! convert to equal area
      X=L*(RADIUS/C1)                               ! x & y
      Y=M*(RADIUS/C1)                               ! coords
      RETURN
      END
* * * *
```

```
     +    COMP(K2),OLDV(K2),TEMP(K2,K2),TMPCOV(K2,K2)
      EXTERNAL
     +    LNDST2,STATUS,MATMUL,MATINV
C Executable Code
      CHGCEN = 0.0
      ERRFLG=.FALSE.
      DO 1000 L1=1,K
C         loop over all clusters
          WRITE(TEXT(44:),'('', cluster '',I2)')L1
          CALL STATUS(TEXT)
          DENOM=0.0
C         initialize 1st moment to zero
          DO 1100 L2=1,M
              COMP(L2)=0.0
 1100     CONTINUE
C         loop over all data
          DO 1200 L2=1,N
C             compute fuzzy membership, -ve if U is -ve
              MEMB=SIGN(ABS(U(L1,L2))**FEXP,U(L1,L2))
C             add to running denominator total
              DENOM=DENOM+ABS(MEMB)
C             loop over all dimensions
              DO 1210 L3=1,M
C                 compute 1st moment, add to total: if MEMB is -ve then
C                 the reverse direction of X is effectively taken
                  COMP(L3)=COMP(L3)+X(L3,L2)*MEMB
 1210         CONTINUE
 1200     CONTINUE
          MAGV=0.0                                  ! initialize vector magnitude
          DO 1300 L2=1,M                            ! loop over all dimensions
              OLDV(L2)=V(L2,L1)                     ! save component of V
              V(L2,L1)=COMP(L2)/DENOM               ! new component of V
              MAGV=MAGV+V(L2,L1)*V(L2,L1)           ! build magnitude of V
 1300     CONTINUE
          MAGV=SQRT(MAGV)
          DO 1400 L2=1,M
              V(L2,L1)=V(L2,L1)/MAGV                ! scale V so mag(V)=1.0
 1400     CONTINUE
          DSTNCE =SQRT(LNDST2(V(1,L1),OLDV,IDENT,MP,M))
          CHGCEN = MAX(CHGCEN,DSTNCE)
          WRITE(DEVICE(4),10) L1,DSTNCE
   10     FORMAT(' Cluster',I3,': centroid change of distance =',F10.5)
          IF(.NOT.FCM) THEN
C             the method requires computation of the scatter/covariance matrix,
C             so zero the TeMPorary COVariance matrix
              DO 1500 L2=1,M
                  DO 1510 L3=1,M
                      TMPCOV(L3,L2)=0.0
 1510             CONTINUE
 1500         CONTINUE
C             and compute either the new fuzzy scatter matrix or the fuzzy
C             covariance matrix:
              DO 1600 L2=1,N
C             loop over all data
C                 determine the sense of X and the multiplier for
C                 the fuzzy scatter matrix
                  SENSE=SIGN(1.0,U(L1,L2))
                  SCALE=ABS(U(L1,L2))
C                 determine the multiplier for the fuzzy covariance matrix
                  IF (FMLE) SCALE=SCALE/DENOM
                  DO 1610 L3=1,M
C                     loop over all dimensions to get the components of X-V
                      COMP(L3)=SENSE*X(L3,L2)-V(L3,L1)
 1610             CONTINUE
                  CALL MATMUL(COMP,COMP,TEMP,M,int(1),M,K2,int(1),K2)
C                 to form (X-V).trans(X-V) as TEMP (M x M matrix).
                  DO 1620 L3=1,M
                      DO 1621 L4=1,M
                          TMPCOV(L4,L3)=TMPCOV(L4,L3)+TEMP(L4,L3)*SCALE
 1621                 CONTINUE
 1620             CONTINUE
```

```
 1600          CONTINUE
C              invert TMPCOV into FCOVAR, compute det(F) as well
               CALL MATINV(TMPCOV,FCOVAR(1,1,L1),M,K2,DETF(L1),OK)
               IF (.NOT.OK) STOP' Singular scatter/covariance matrix'
C              check to see if det(F) is negative
               IF (DETF(L1).LT.0.0) ERRFLG=.TRUE.
               IF (FCV) THEN
C                  scale inv(SFi) to form the distance weighting matrix
                   SCALE=DETF(L1)**(1.0/M)
                   DO 1700 L3=1,M
                       DO 1710 L4=1,M
                           FCOVAR(L4,L3,L1)=FCOVAR(L4,L3,L1)*SCALE
 1710                  CONTINUE
 1700              CONTINUE
               ELSE IF (FMLE) THEN
C                  compute the a priori probability using DENOM
                   APPROB(L1)=DENOM/N
               END IF
           END IF
 1000 CONTINUE
      OK=.NOT.ERRFLG
      RETURN
      END
* * * *
```

```
C                this is FMLE, so compute using exponential Mahalanobis
C                distance (norm is fuzzy covariance matrix)
                 D2INV = LNDST2(CX,V(1,L2),FCOVAR(1,1,L2),MP,M)
                 D2INV = (APPROB(L2)/SQRT(DETF(L2))) * EXP(-D2INV/2.0)
               END IF
C              save U for later, compute new U taking SENSE into account
               OLDU(L2)=ABS(U(L2,L1))
               U(L2,L1)=SENSE*(D2INV**FEXPI)
C              compute denominator, ready to determine memberships
               DENOM=DENOM+ABS(U(L2,L1))
 1100        CONTINUE
C            divide numerators by DENOM to determine memberships
             SCALE=1.0/DENOM
             DO 1200 L2=1,K
                 U(L2,L1)=U(L2,L1)*SCALE
                 CHANGE = ABS(U(L2,L1)) - OLDU(L2)
                 CHGMEM=MAX(CHGMEM,ABS(CHANGE))
 1200        CONTINUE
 1000 CONTINUE
      WRITE(DEVICE(4),'('' Maximum change in U ='',F10.4)')CHGMEM
      RETURN
      END
* * * *
```