A Kinematic Numerical Camera Model
for the SPOT-1 Sensor

by

Mark Anthony O'Neill

Submitted for the degree of
Doctor of Philosophy
of the
University of London

Department of Photogrammetry and Surveying
University College London
Gower Street
London WC1E 6BT
United Kingdom

August 1991
Abstract

A novel method for modelling linear push-broom sensors has been developed. A numerical model which incorporates the satellite attitude and position data is used to compute the absolute orientation. This method makes a break with traditional photogrammetric practice, in that instead of using an approach based on collinearity equations, the absolute orientation is computed iteratively using a numerical multi-variable minimisation scheme. All current implementations of the model use the Powell direction-set method, but in principle, any multivariable minimisation scheme could be substituted.

The numerical method has significant advantages over the collinearity approach. The number of ground control points needed to form an accurate model is reduced and the numerical approach offers a superior basis for the development of general purpose multi sensor modelling software.

In order to test these assertions, a numerical model of the SPOT-1 sensor was coded and tested against a pre-existing collinearity based model. Exhaustive tests showed the numerical model, using 3 or fewer ground control points, consistently equaled or bettered the performance of the earlier model, using between 6 and 15 ground control points, on the same test data.

A general purpose sensor modelling system was developed using the code developed for the initial SPOT-1 model. Currently this system supports many rigid linear sensors systems including SPOT-1, SPOT-2, ITIR, MISR, MEOSS and ASAS. Further extensions to the system to enable it to model non-rigid linear sensors such as AVHRR and ATM are planned. Work to enable the system to perform relative orientations for a variety of sensor types is also ongoing.
## Contents

Abstract 2

Contents 3

List of Figures 11

List of Tables 19

Acknowledgements 22

**Thesis Background and Plan** 24

1 Introduction. 24

2 History and Background of the O'Neill-Dowman Camera Model. 24

  2.1 Development of the topographic mapping system under Alvey MMI-137. 25

  2.2 Development of stereo-matchers under Alvey MMI-137. 26

  2.3 Development of camera model under Alvey MMI-137. 28

  2.4 Development of DEM interpolation software under Alvey MMI-137. 29

  2.5 Development of visualisation tools under Alvey MMI-137. 29

3 Post Alvey Development of the Topographic Mapping System. 30

  3.1 Post Alvey development of the stereo matcher. 30

  3.2 Post Alvey development of camera model. 31

  3.3 Post Alvey development of interpolator. 31

  3.4 Post Alvey development of the visualisation software. 31

4 Thesis Plan. 32
2.4.1 A description of the SPOT-1 header data. 61
2.4.2 Setting up the relative camera model. 63
2.4.3 Splining the SPOT-1 header position and velocity data. 63
2.4.4 Computation of approximate satellite reference axes. 63
2.4.5 Computation of the rough attitude matrix. 65
2.4.6 Computation of direction of an arbitrary ray in object space. 68
2.4.7 Computation of position vector of ray in reference space. 68
2.4.8 Computing the space intersection. 69

2.5 The Use of GCPs to Compute the Absolute Attitude. 72
2.5.1 Selection of parameter space. 73
2.5.2 The cost function. 73
2.5.3 Extending the optimised parameter space. 74

2.6 Extension of the Cost Function to Include Conjugate Data. 74
2.6.1 Incorporation of the attitude data. 76

2.7 Extension of the Model to Strips. 77
2.7.1 Extraction of data from individual header data files. 78
2.7.2 Location of tie points to register images within the strip. 79
2.7.3 Automatic calculation of along-track overlap of images within strip. 80

2.8 The Back Transform. 80
2.8.1 Basic transformation algorithm. 80
2.8.2 Increasing the computational efficiency of the algorithm. 81
2.8.3 The effect of blocking on the forward transform. 84
2.8.4 Alternative methods for speeding up the camera model algorithm. 84
   2.8.4.1 Code optimisation. 84
   2.8.4.2 Use of specialist hardware. 86

2.9 The fast O'Neill-Dowman Camera Model. 87

Chapter 3. Initial Implementation: Testing and Results 89
3.1 Introduction. 89
3.2 Testing the Absolute Accuracy of model. 90
   3.2.1 A description of the methods used to assess absolute accuracy. 90
   3.2.2 A discussion of the accuracy of 2-GCP models. 92
   3.2.3 A discussion of the absolute accuracy of 3-GCP models. 101
3.3 Studies of the Dynamics of Camera Model Relaxation. 102
   3.3.1 The ideal relaxation profile. 103
3.3.2 The non-ideal relaxation profile. 103
3.3.3 The relative camera model. 106
3.3.4 An aside: the camera model as a thermodynamic system. 108

3.4 Accuracy of the O'Neill-Dowman Camera Model Without Ground Control 112
3.4.1 Use of zero GCP camera model to detect poorly measured ground control. 114

Chapter 4. O'Neill-Dowman Model: LSL/RAE implementation 117
4.1 Introduction. 117
4.2 Differences Between the Current and Initial Implementations of Model. 118
   4.2.1 Summary of vector parametric ray equation generation in initial implementation. 119
   4.2.2 Ray equation generation in RAE/LSL implementation. 120
   4.2.3 Determination of reference space unit ray direction vector. 120
   4.2.4 Determination of sensor attitude in LSL/RAE implementation. 123
   4.2.5 Determination of the sensor position in RAE/LSL implementation. 126
4.3 The Forward Transform. 127
4.4 The Back Transform. 127
   4.4.1 Fast back transform based on Powell Direction Set method. 127
   4.4.2 Fast back transform algorithm based on Brent method. 128
   4.4.3 A comparison of the throughput of the Powell and Brent relaxation schemes. 132
4.5 Automatic Detection of Poor Ground Control and/or Check Points. 133
   4.5.1 Implementation of automated shift-pruning algorithm. 134
   4.5.2 Examples of the effectiveness of shift-pruning. 136
   4.5.3 Limitations of the shift-pruning algorithm. 136
4.6 Use of Alternative Cost Functions for Absolute Orientation. 137
   4.6.1 Description of the RRSKEW cost function. 137
   4.6.2 A description of the RGCPD cost function. 138
   4.6.3 A comparison of the RRSKEW and RGCPD cost functions. 140
   4.6.4 Comparison of the ALVEY and LSL/RAE implementations of the model. 140

Chapter 5. LSL/RAE Implementation: Testing and Results 143
5.1 Introduction. 143
5.2 Metrics for Testing Camera Model Performance. 144
   5.2.1 Definition of the ARMSE statistic. 144
   5.2.2 Definition of the ARMSPE statistic. 145
5.2.3 Definition of the ARMSHE statistic. 146
5.2.4 Definition of the RRMSE statistic. 146
5.2.5 Definition of the RRMSPE statistic. 147
5.2.6 Definition of the RRMSHE statistic. 147
5.3 Review of Model Accuracy Testing. 148
  5.3.1 Test of absolute error of model. 148
  5.3.2 Stability of the ARMSE as a function of the GCP configuration. 150
  5.3.3 Stability of the strip solution as a function of strip length. 151
5.4 Effect of GCP/Checkpoint Measurement Errors on Model ARMSE. 151
  5.4.1 Effects of GCP observation error on model ARMSE. 151
  5.4.2 Effects of error in check point measurement on ARMSE. 157
5.5 Assessment of the Effect of Sensor Attitude on Model ARMSE. 159
  5.5.1 Effect of error in sensor position on ARMSE. 159
  5.5.2 Effect of error in sensor velocity on ARMSE. 163
  5.5.3 Effect of error in static sensor attitude on ARMSE. 165
  5.5.4 Effect of error in the scene centre time on model RMSE. 167
5.6 Assessment of Accuracy and Speed of Orthoimage Production. 169
  5.6.1 Test of accuracy and speed of back transform. 169
  5.6.2 Definition of the IARMSE statistic. 170
  5.6.3 Use of differential IARMSE statistic as an indicator of model error. 171
  5.6.4 Accuracy of the back transform. 172
  5.6.5 Speed assessment of the fast back transform. 173
  5.6.6 Assessment of orthoimage accuracy. 173
  5.6.7 Assessment of speed of orthoimage production. 175
  5.6.8 Testing of relative error. 178
5.7 Other Accuracy Tests Performed on the Model. 180
  5.7.1 Formation of camera models with a constrained parameter space. 180
  5.7.2 Effect of satellite dynamic attitude on model ARMSE and RRMSE. 181
  5.7.3 Use of shift pruning to reject poor ground control. 182

Chapter 6. Overview of GEODEM Topographic Mapping System 184
6.1 Introduction. 184
6.2 Spot Image Products. 186
6.3 Generation of the Digital Disparity Model [DDM]. 189
  6.3.1 Selection of stereo matching algorithm. 189
6.3.1.1 The Barnard and Thompson algorithm. 190
6.3.1.2 The PMF algorithm. 190
6.3.1.3 The Gruen adaptive least squares correlation algorithm. 191
6.3.1.4 The PRISM algorithm. 191
6.3.1.5 The Ohta and Kanade algorithm. 192
6.3.2 Generation of a DDM using the Otto-Chau algorithm. 193

6.4 Generation of the DEM. 196
6.4.1 Data Required to set up the camera model. 198
6.4.2 The Gugan-Dowman camera model. 200
6.4.3 The O'Neill-Dowman camera model. 201

6.5 Generation of the Gridded DEM [GDEM]. 202

6.6 Uses of SPOT-1 DEM Products. 203

6.7 Conclusion. 208

Chapter 7. Future Research Directions 212
7.1 Introduction. 212
7.2 Finding an Optimal Cost Function and Associated Parameter Set. 212
7.3 The Development of a More Sophisticated Orbital Model. 217
7.3.1 Non-conservation of angular momentum in the current orbit-model. 217
7.3.2 Sensitivity of camera model to orbit segment interpolation. 217
7.3.3 Improving the model of the orbit segment. 218
7.3.3.1 Orbit post processing using conjugate data. 218
7.3.3.2 Post processing using attitude data. 219
7.4 Development of a Multi-Sensor Camera Model System. 220
7.4.1 Scope of multi-sensor camera modelling systems. 220
7.4.2 Extension to non-rigid and active sensor systems. 221
7.4.3 System structure. 223
7.5 Development and Assessment of Camera Model Assisted Stereo Matching 226
7.5.1 Overview of the problems faced by stereo matching algorithms. 226
7.5.2 Generalisation of epipolar algorithms to non epipolar geometries. 232
7.5.3 Adaptation of image space area based algorithms to deal with discontinuities. 234
7.5.4 An Example of a camera constrained stereo matching algorithm. 236
7.5.5 Extension of camera constrained search methods to other stereo matching algorithms. 243
7.5.6 Metrics for inter-image similarity measure. 244
7.5.7 Non camera constrained algorithms for matching large scale imagery.
7.5.7.1 A comparison of the different seeding techniques.
7.5.7.2 The cascade algorithm as a pyramidal stereo matcher.

Chapter 8. Conclusions
8.1 Introduction.
8.2 Accuracy of Camera Modelling System.
8.3 Principal Sources of Error Within Camera Modelling System.
8.3.1 Methods of error removal.
8.3.1.1 Pre processing by stereo matcher.
8.3.1.2 Post processing by multimodal relaxation scheme.

Bibliography

Appendix 1. Results of testing initial implementation
Appendix 2. Comparison of initial model with OEEPE workshop models
Appendix 3. Results of testing LSL/RAE implementation of model

Appendix 4. Software Implementation
A4.1 Introduction.
A4.1.1 Development methodology.
A4.1.2 Choice of implementation language.
A4.1.3 Advantages of using C++ for system development.
A4.1.4 Limitations of the current implementation.
A4.2 Camera Model Implementation Overview.
A4.3 The Top Level of the System.
A4.4 The Top Level Organisation of the Camera Model.
A4.4.1 The Camera Model filter.
A4.4.2 The co-ordinate system interconversion filters.
A4.4.3 The distributed programming support filters.
A4.4.4 The Top Level Organisation of the Camera Model.
A4.5 The Portable UNIX Programming System.
A4.5.1 Introduction. 376
A4.5.2 The utilities library - utilib. 377
A4.5.3 The stochastic programming library - casinolib. 379
A4.5.4 Numerical functions library - nfolib. 380
A4.5.5 The linear programming library - veclib. 381
A4.5.6 The network communications library - netlib. 383
A4.5.7 The HIPL Picture/Header Library - hiplib. 385
A4.5.8 The generic sensor library - glslib. 385

Appendix 5. UNIX man pages for camera model and support filters 389

Appendix 6. A simulation study of the ITIR sensor 432
A6.0 Introduction. 432
A6.1 Methodology. 432
   A6.1.1 Derivation of base ITIR orbit segment and image co-ordinates. 433
   A6.1.2 Formation of SPOT-1 model. 433
   A6.1.3 Generating the sparse digital elevation model. 434
   A6.1.4 Creation of the ITIR orbit segment. 435
   A6.1.5 Simulation of vertical ITIR looks from oblique SPOT-1 looks. 436
A6.2 Checking the Accuracy of the Base ITIR Orbit Segment. 437
A6.3 Creation of Perturbed ITIR Orbit Segments. 438
   A6.3.1 Simulating the effect of pointing angle error. 438
   A6.3.2 The precise meaning of time t. 440
   A6.3.3 Estimation of sensor pointing error from . 440
A6.4 Assessment of Accuracy. 441
A6.5 Simulation of Observation Error. 442
A6.6 Assessment of the Use of Ground Control. 443
A6.7 Assessment of Error in Sensor Position. 443
A6.8 Jitter. 444
A6.9 Results. 444
A6.10 The Effect of Sensor Pointing Error on Observed ARMSE. 445
A6.11 The Effect of 2 GCP's on Reducing Pointing Angle Error. 446
A6.12 Effect of Observation Error on Observed ARMSE. 447
A6.13 Effect of Ground Control on Reducing the Effect of Observation Error. 447
A6.14 Conclusions. 449
A6.15 Recommendations. 449
List of Figures

Figure 1.1. Showing the components of the SPOT-1 satellite system. 37
Figure 1.2. SPOT-1 HRV Instrument [after Cheverel et al, 1981]. 39
Figure 1.3. Schematic of the ground support facilities for SPOT-1. 47
Figure 2.1. Showing the definition of ray-ray skewness and RMS error vectors. 60
Figure 2.2. Showing definition of look angles, PSI_FIRST_X, PSI_FIRST_Y, PSI_LAST_X, and PSI_LAST_Y. 64
Figure 2.3. Showing the geometry of the SPOT-1 sensor. 66
Figure 2.4. Showing the relationship between reference and working attitude for the O'Neill-Dowman SPOT-1 camera model. 67
Figure 2.5. Showing space resection geometry for a two look SPOT-1 stereo system. 71
Figure 2.6. Showing the geometry of the back transform. 82
Figure 2.7. Showing schematic of the attitude blocking concept used the SPOT-1 sensor model. 85
Figure 3.1. Model accuracy: South Yorkshire [29/242] (as a function of GCP line spacing). 93
Figure 3.2. Model accuracy: South Yorkshire [29/242] (as a function of GCP sample spacing). 94

Figure 3.3. Model accuracy: South Yorkshire [29/242] (as a function of GCP vector spacing). 95

Figure 3.4. Model accuracy: SPOT-PEPS Dataset (as a function of line spacing). 96

Figure 3.5. Model accuracy: SPOT-PEPS Dataset (as a function of sample spacing). 97

Figure 3.6. Model accuracy: SPOT-PEPS Dataset (as a function of vector spacing). 98

Figure 3.7. SPOT: Aix en Provence 51/262 near-ideal (relaxation) profile. 104

Figure 3.8. SPOT: Aix en Provence 51/262 non-ideal (relaxation) profile. 105

Figure 3.9. SPOT: Aix en Provence O'Neill-Dowman Camera Model 51/262 ray-ray skewness plot (plan). 110

Figure 3.10. SPOT: Aix en Provence O'Neill-Dowman Camera Model 51/262 ray-ray skewness plot (height). 111

Figure 3.11. Showing a comparison of slope for DEM's which have been generated using a camera model with and without GCP's. 113

Figure 4.1. Showing the geometry of the back transformation. 130

Figure 4.2. Showing back transformation geometry when the line containing the image space point corresponding to the object space point has been found. 131
Figure 4.3  Showing the geometry of the RGCPD cost function. 139

Figure 5.1a.  Showing three ground control points arranged in a long triangle configuration over a four scene strip. 152

Figure 5.1b.  Showing alternative long triangle configuration for a four scene strip. 153

Figure 5.2.  Showing three ground control points arranged in short triangle configuration in a four scene strip. 154

Figure 5.3.  OEEPE strip A: Showing RMS error as a function of strip length: Object space RMS vector error. 155

Figure 5.4.  OEEPE strip A: Effect of error in GCP [image] measurement: Object space RMS vector error. 158

Figure 5.5.  OEEPE strip A: Effect of error in check point [image] measurement: Object space RMS vector error. 160

Figure 5.6.  OEEPE strip A: Effect of error in sensor look positions: Object space RMS vector error. 162

Figure 5.7.  OEEPE strip A: Effect of error in sensor velocity: Object space RMS vector error. 164

Figure 5.8.  OEEPE strip A: Showing the effect of vary satellite attitude: Object space RMS vector error. 166

Figure 5.9.  OEEPE strip A: Showing the effect of varying scene centre time: Object space RMS vector error. 168

Figure 5.10.  10m PAN orthoimage synthesised from SPOT-1 scene SH1870131110005 using OS MCE DEM supplied by National Remote Sensing Centre. 177

Figure 6.1.  SPOT-1 DEM production flowchart. 185
Figure 6.2. Showing SPOT-1 image of Sopka Shivelush Kamchatka, Eastern Siberia, USSR.

Figure 6.3. Showing a typical set of seedpoints generated using the cascade system, using a small section of the SPOT-1 coverage of Death Valley California.

Figure 6.4. Showing typical coverage attained with the Otto-Chau stereo matcher.

Figure 6.5. Showing Kriged DEM generated using Death Valley SPOT-1 image sections.

Figure 6.6. Topographic map of Sopka Shivelush Kamchatka, Lambertian Shaded to emphasise relief.

Figure 6.7. Showing 10m orthoimage of Isle of Wight with MCE DEM from which it was derived.

Figure 6.8. Showing frame from the animated flight over Sopka Shivelush Kamchatka.

Figure 6.9. Showing frame from the animated flight over Ius Chasma, Mars (derived from Viking Orbiter Imagery).

Figure 7.1. Showing camera model organised as interacting group of process-objects.

Figure 7.2. Schematic of a camera geometry refinement system based on a stochastic parameter optimisation scheme.

Figure 7.3. Schematic of the sensor geometry of directionally sensitive RADAR [or SONAR] altimeter system.

Figure 7.4. Showing a dataflow implementation of the O'Neill-Dowman Camera Model.
Figure 7.5.  Showing the concept of disparity domains in a segmented image.  229

Figure 7.6.  Showing the output of the USM stereo matcher for typical urban imagery.  230

Figure 7.7.  Schematic of the USM stereo matcher.  231

Figure 7.8.  Showing the geometry of epipolar line projection using an absolute sensor model.  233

Figure 7.9.  Showing schematic of camera model assisted stereo matching using non epipolar imagery with an epipolar stereo matching algorithm.  235

Figure 7.10.  2D search plane for intra-scanline search.  239

Figure 7.11.  Showing how a relative orientation may be used to extend Ohta and Kanades algorithm to non epipolar imagery.  240

Figure 7.12.  Showing how the minimum cost path tau may be expressed as a function of the zero relief minimum cost path, tau0.  241

Figure 7.13.  Cascading flow chart for conjugate point detection.  247

Figure 7.14.  Showing the output of the Cascade Algorithm operating in autoseeding mode.  249

Figure 7.15.  Showing DDM generated by application of Cascade Algorithm to Bloomsbury Imagery.  250

Figure 7.16.  Showing stereo matched coverage of SIR-B [SAR] image of Mount Shasta.  251
Figure A4.1. Showing the structure of a typical filter within the O'Neill-Dowman Camera Modelling System.

Figure A4.2. Showing the software environment of the O'Neill-Dowman Camera Modelling System.

Figure A4.3. SPOT-1 DEM production flowchart.

Figure A6.1. Showing how the geometry an ITIR vertical look orbit may be synthesised from an oblique SPOT-1 look.

Figure A6.2. Showing the geometry of the ITIR sensor.

Figure A6.3. Showing empirical form of SPOT-1 attitude variograms [after Westin].

Figure A6.4. ITIR sensor simulation [9.0 degrees off nadir]: Ground error as function of yaw variogram sigma.

Figure A6.5. ITIR sensor simulation [9.0 degrees off nadir]: Ground error as function of pitch variogram sigma.

Figure A6.6. ITIR sensor simulation [9.0 degrees off nadir]: Ground error as function of roll variogram sigma.

Figure A6.7. ITIR sensor simulation [4.5 degrees off nadir]: Ground error as function of yaw variogram sigma.

Figure A6.8. ITIR sensor simulation [4.5 degrees off nadir]: Ground error as function of pitch variogram sigma.

Figure A6.9. ITIR sensor simulation [4.5 degrees off nadir]: Ground error as function of roll variogram sigma.

Figure A6.10. ITIR sensor simulation [nadir]: Ground error as a function of yaw variogram sigma.
Figure A6.11. ITIR sensor simulation [nadir]: 464
Ground error as a function of pitch variogram sigma.

Figure A6.12. ITIR sensor simulation [nadir]: 465
Ground error as a function of roll variogram sigma.

Figure A6.13. ITIR sensor simulation [4.5 degrees off nadir]: GCPD cost 466
function; showing the effect of relaxation on predicted roll error.

Figure A6.14. ITIR sensor simulation [4.5 degrees off nadir]: GCPD cost 467
function; showing the effect of relaxation on predicted pitch error.

Figure A6.15. ITIR sensor simulation [4.5 degrees off nadir]: GCPD cost 468
function; showing the effect of relaxation on predicted yaw error.

Figure A6.16. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost 469
function; showing the effect of relaxation on predicted yaw error.

Figure A6.17. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost 470
function; showing the effect of relaxation on predicted yaw error.

Figure A6.18. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost 471
function; showing the effect of relaxation on predicted yaw error.

Figure A6.19. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost 472
function; showing the effect of image observation error [without ground control].

Figure A6.20. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost 473
function; showing the effect of object observation error [without ground control].

Figure A6.21. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost 474
function; showing the effect of image observation error [with ground control].
Figure A6.22. ITIR sensor simulation [4.5 degrees off nadir]: RRS cost function; showing the effect of object observation error [with ground control]. 475

Figure A6.23 ITIR model: Sparse DEM point dataset spatial distribution. 476

Figure A6.24 ITIR simulation [nadir]: Plan error as a function of roll angle. 477

Figure A6.25 ITIR simulation [nadir]: Height error as a function of roll angle. 478

Figure A6.26 ITIR simulation [nadir]: Plan error as a function of pitch angle. 479

Figure A6.27 ITIR simulation [nadir]: Height error as a function of pitch angle. 480

Figure A6.28 ITIR simulation [nadir]: Plan error as a function of yaw angle. 481

Figure A6.29 ITIR simulation [nadir]: Height error as a function of yaw angle. 482

Figure A6.30 ITIR simulation [nadir]: Plan error as a function of yaw, pitch and roll error. 483

Figure A6.31 ITIR simulation [nadir]: Height error as a function of yaw, pitch and roll error. 484
List of Tables

Table 1.1  Orbital parameters for SPOT-1. 45
Table 1.2  Main instrument parameters for each SPOT-1 HRV instrument. 45
Table 3.1  Sample ARMSE statistics for initial implementation of model. 99
Table 3.2  Effect on accuracy of conjugate points and header attitude data. 109
Table 3.3  Showing the effects of shift pruning on the Cyprus and Dorset single scene models. 116
Table 4.1  Showing the effect of automatic pruning on camera model accuracy. 142
Table 4.2  Showing the effect of the cost function on the RMS statistics of the LSL/RAE implementation of the camera model. 142
Table 4.3  A comparison of the RMSE and associated statistics for the Alvey and LSL/RAE implementations of the O'Neill-Dowman Camera Model. 142
Table 5.1  Showing the improvement of the vector ARMSE of forward transformation in the LSL/RAE implementation of software.

Table 5.2  Showing the point transformation rate for the back transform as a function of architecture.

Table 5.3  Showing residuals at check points for the Isle of Wight 10m orthoimage.

Table A1.1  Results of single model accuracy tests.

Table A1.2  Results of OEEPE strip A [50/259-50/262] accuracy tests.

Table A1.3  Results of OEEPE strip B [49/259-49/262] accuracy tests.

Table A1.4  Results of fast O'Neill-Dowman accuracy tests.

Table A2.1  A comparison of the performance of the O'Neill-Dowman Model with other SPOT-1 sensor models for OEEPE strip A.

Table A2.2  A comparison of the performance of the O'Neill-Dowman Model with other SPOT-1 sensor models for OEEPE strip B.

Table A3.1  Table showing the sensor model accuracy for ground control configuration B9.

Table A3.2  Table showing the sensor model accuracy for ground control configuration A15.

Table A3.3  Table showing the sensor model accuracy for Cyprus ground control configuration 1.

Table A3.4  Table showing the sensor model accuracy for Dorset ground control configuration 1.
Table A3.5 Table showing the sensor model accuracy for Oman ground control configuration 1. 336

Table A3.6 Table showing the sensor model accuracy for ground control configuration B9b. 343

Table A3.7 Table showing the sensor model accuracy for ground control configuration A4. 350

Table A3.8 Table showing sensor model accuracy for Wyoming XS ground control configuration 1. 357

Table A6.1 Showing the maximum error radii for parameters subjected to the observation error fuzzing process. 451

Table A6.2 Errors at ground points for attitude errors at 2σ and 3σ levels. 451a
Acknowledgements

I would like to thank my project supervisor Ian Dowman [Department of Photogrammetry and Surveying, University College London] without whose considerable help, this thesis would not have been forthcoming. I would also like to thank Mia Denos for her painstaking reading of the draft manuscript, and for her help with the stereo matching work described in chapter 7. David Gugan has also made many valuable contributions to this work, both as a research associate at UCL and also in his present position, as a consultant to Laserscan. I would also like to thank Tony Cook for his encouragement during the writing-up phase of this thesis.

I would also like to thank my friends at Cambridge University; Mike Roe [Computer Laboratory, University of Cambridge] for many stimulating discussions on optimisation techniques, Nadeem Malik, Chris Kenyon, Mark Manning [Department of Applied Mathematics and Theoretical Physics, University of Cambridge] for their encouragement, stimulating discussion, and provision of additional computing facilities, and Andrew Derbyshire, [Herschel-Smith Laboratory of Medicinal Chemistry, University of Cambridge] for his discussion on the subject of Kalman Filters.

I would like to thank someone who I have never met, but whose indirect influence on this thesis has been enormous, Jawhr Bammi [Case Western Reserve University, Cleveland, Ohio, USA], who supplied me with an effective version of UNIX [GNU-MINIX], an EMACS like editor [MG] and a fully functional optimising compiler [GNU-C version 1.36] for the Atari-ST microcomputer. Without his aid, the writing of this thesis and the testing and development of the algorithms described within it would have been all the harder. I would also like to take this opportunity to thank James Pearson [UCL-PS system manager], for keeping the computing equipment up and running.

I would also like to thank those organisations who donated the data, equipment and expertise without which this thesis would not have been completed. Acorn Computers Ltd supplied me with an R140 workstation. This was used to develop and test a substantial amount of software described in
Acknowledgements

this thesis, and also to typeset the thesis itself using the \texttt{ptroff} text formatter. Various organisations including BP Research, The Ordnance Survey, The Royal Aircraft Establishment and the OEEPE supplied me with SPOT-1 stereo datasets without which the model would have been impossible to test.

Finally, I would like to thank Professor Ian Harley [Head of the Department of Photogrammetry and Surveying at UCL], for providing encouragement and accommodation for the three and a half years I have been working on this thesis.
Thesis Background and Plan

1 Introduction.

This thesis reports work done by the author within the Department of Photogrammetry and Surveying at University College London, into the design of a simple robust kinematic camera modelling system for SPOT-1 satellite sensor and similar linear pushbroom sensor systems. The thesis also describes the work of persons other than the author, in order to place the camera modelling system in its proper context. The area based stereo matcher gruens is principally the work of G.P. Otto and T.K.W. Chau of the Department of Computer Science at UCL. Later extensions to the basic area based stereo matcher have been made by M.J. Zemarly and M. Holden [parallel implementation of stereo matcher on transputer array], the author in collaboration with Mia Denos [coarse to fine stereomatching with texture based autoseeding], and T. Day [C++ implementation of the gruens algorithm]. The interpolator used to generate a gridded DEM from the ungridded camera model output is the work of T. Day. The visualisation tools which are described are principally the work of V. Paramananda with a number of contributions from T. Day and D. Rees. The dynamic visualisation tools evileyeye and genisis are the work of T. Day and D. Rees respectively.

This work described has been accomplished with funding from a number of bodies. Over the time period January 1987 to July 1989, the work was supported by funding from the Alvey Directorate under the Alvey MMI-137 [Real Time 2.5D Vision Systems] project. The Alvey MMI-137 project was a three year project involving a number of collaborators including University College London Department of Photogrammetry and Surveying, University College London Department of Computer Science, Laserscan Laboratories Cambridge, RSRE Malvern and Thorn-EMI Central Research Laboratories, Hayes. The remit of the project was to develop a real time system for the production of dense 2.5D depth maps and associated products from sources as disparate as satellite imagery and the output of close range
After the end of Alvey Project, funding for continued development of the camera model was supported by the Royal Aircraft Establishment and Laserscan Laboratories, who are currently commercialising a system for producing topographic maps from SPOT-1 imagery. This system will be based on the prototype topographic mapping system which was developed under the aegis of the Alvey MMI-137 project.

2 History and Background of the O’Neill-Dowman Camera Model.

2.1 Development of the topographic mapping system under Alvey MMI-137.

The work reported in this thesis was instigated by the Alvey MMI-137 project when it was decided that imagery from the newly launched French SPOT-1 satellite would be project exemplar for the purposes of demonstrating the potential of small and medium scale topographic mapping from space borne sensors. As a result of this decision, three of the project collaborators, the Departments of Photogrammetry and Surveying and Computer Science at University College London and Laserscan Laboratories Cambridge commenced a research program to develop a prototype system to produce topographic map products from SPOT-1 imagery.

The topographic mapping system required the following subsystems:

a) A stereo matching system capable of generating dense digital disparity models from SPOT-1 imagery.

b) A camera model to transform the conjugate points in the digital disparity model to object space.

c) An interpolation scheme which takes the ungridded object space output of the camera model and produces a gridded digital elevation model [DEM].

d) Stereo image visualisation tools using the HIPS image processing system [Landy et al, 1982] operating in the Sunview en-
2.2 Development of stereo-matchers under Alvey MMI-137.

The development of the stereo matching component of the system was undertaken primarily by the Department of Computer Science at UCL. Initially, the PMF stereo ranging algorithm [Pollard et al, 1985], developed by a machine vision research group at Sheffield University was implemented [Chau, 1987a]. However, when used to match SPOT-1 imagery, the PMF algorithm was found to suffer from a number of flaws: the principal problem with PMF was that it requires epipolar imagery: SPOT-1 imagery was shown to be insufficiently epipolar for processing by algorithms such as PMF which require strictly epipolar imagery [Day and Muller, 1989a], unless a resampling scheme is used. In the early stages of the Alvey MMI-137 project it was thought that a digital elevation model [DEM] would be needed to perform epipolar resampling. A resampling scheme using a DEM was in fact developed [O’Neill and Dowman, 1988]. PMF was successfully tested using synthetic epipolar stereo-mates produced using this system indicating that PMF had some potential as a stereo matching algorithm for satellite imagery which was sufficiently epipolar. Subsequently, schemes for using PMF with non-epipolar imagery have been described [Porill et al, 1989], which permit the algorithm to be used with quasi-epipolar imagery. Even if the epipolarity problem could have been solved in the early stages of the Alvey MMI-137 project, there would have still been significant problems with PMF as it is an edge based rather than an area based stereo matching algorithm. The consequence of matching only at edges, located in the imagery using a suitable feature detector, for example the Marr-Hildreth operator [Marr and Hildreth, 1980], is that the sparse rather than a dense 2.5D depth map will be generated.

The requirement that dense depth maps be generated from quasi-epipolar imagery led to the investigation of area based stereo matching algorithms based on patch correlation. Two algorithms were looked at by the Alvey MMI-137 research group. The first algorithm investigated, PRISM [Nishahara, 1984] failed because the correlation process assumes a pink noise distribution which is not present in SPOT-1 imagery. The second
area-based algorithm considered was an extension to the adaptive least squares correlation algorithms proposed by Gruen [Gruen 1985; Gruen and Baltsavias, 1987; Gruen and Baltsavias 1988]. The principal extension to the algorithm was the provision of a sheet growing mechanism, in which the validated matches are used to predict neighbouring stereo correspondences. The prediction mechanism is essential to the operation of the algorithm, as it overcomes the unimodal nature of Gruens' scheme. While the prediction mechanism permits the extended Gruen algorithm to produce dense 2.5D depth maps, it also limits the use of the algorithm to certain imagery types [such as space borne imagery] in which the disparity function is continuous. The sheet growing adaptation of Gruens' algorithm is described by its inventors [Otto and Chau, 1989].

In addition to the principal stereo matching algorithms described above, two other algorithms were also investigated by the project. Barnard and Thompsons algorithm [Barnard and Thompson, 1980] was studied by RSRE Malvern, and coded in Occam on a transputer array. This was done primarily in to gain experience in programming in a MIMD environment. The experiments performed with the transputer implementation of Barnard and Thompson are described by Collins [Collins et al, 1987]. Although the Barnard and Thompson approach cannot produce dense digital elevation models, it can function with non-epipolar imagery. The Alvey MMI-137 project envisaged that the Barnard and Thompson stereo-matcher would be an autoseeding algorithm being used to identify a small number of stereo correspondences. These conjugate points would then be used to seed the area based Otto-Chau algorithm described above, which is capable of producing dense 2.5D digital disparity models [DDM's] from SPOT-1 stereo imagery.

A stereo matching algorithm based on an inter- and intra-scanline correlation suggested by Ohta and Kanade [Ohta and Kanade, 1985] was studied by Thorn-EMI Central Research Laboratories [TECRL]. Like the PMF algorithm, Ohta and Kanade's algorithm requires a strictly epipolar geometry. It is therefore also unsuitable for general image correlation. The TECRL algorithm also proved to be exceedingly slow: it took almost a week of Sun-3 CPU time to process small [240 pixel square] resampled patches extracted from SPOT-1 stereo imagery.
2.3 Development of camera model under Alvey MMI-137.

After the Alvey MMI-137 project had chosen the SPOT-1 system as its exemplar space borne application, the requirement arose for a camera model to transform conjugate points in image space to points in object space in order to form a DEM. Initially, a SPOT-1 camera model developed primarily for hardcopy use with analytical plotters such as the Kern DSR-1 or DSR-11 [Gugan and Dowman, 1988; Gugan, 1987; Gugan, 1988] was adapted for use with a digital photogrammetric workstation. This adaptation of the Gugan-Dowman camera model performed well, but nevertheless a number of shortcomings were identified.

The Gugan-Dowman model predated the launch of the SPOT-1 satellite in 1986, and was developed before detailed information on the sensor was available. As a consequence of this, it does not use telemetry data in order to facilitate setting up a stereo model. Instead, a dynamic adaptation of the classical photogrammetric collinearity equations is used. A consequence of this is that a large number of ground control points are required to compute an accurate absolute stereo model. The O’Neill-Dowman SPOT-1 camera model, which is the major subject of this thesis was developed in order to overcome this problem and provide a simple robust camera model which requires few ground control points. The use of telemetry data from the SPOT-1 satellite giving the position, velocity and attitude of the satellite at regular intervals permitted a model to be developed which retained the level of accuracy achieved by the Gugan-Dowman model whilst using few [typically < 4] ground control points. An additional goal stipulated by the Alvey MMI-137 project was that the camera model be designed in a modular fashion, so that it may form the basis of a comprehensive sensor modelling system.

The first steps toward this goal were taken within the lifetime of the Alvey MMI-137 project, with the addition of a simple idealised linear sensor to the camera modelling system in order to synthesise epipolar stereo-mates from a combination of SPOT-1 imagery and a DEM [O’Neill and Dowman, 1988]. The concept of modularity has proved to be a durable: sensor models which are based on the library of routines developed for the SPOT-1 sensor have been used to build models of the ITIR [Infra red Thermal Imaging Radiometer] sensor [O’Neill and Dowman, 1991], the ASAS sensor and similar
linear rigid sensor systems.

2.4 Development of DEM interpolation software under Alvey MMI-137.

The other major component of a topographic map production system developed during the lifetime of the Alvey MMI-137 project is an interpolator, which is based on the technique of kriging [Delfiner and Delhomme, 1975; Day, 1989]. The purpose of the interpolation algorithm is to take the ungridded DEM produced by the camera model and to remap it to a rectangular gridded DEM [GDEM] which may then be conveniently manipulated.

The kriging algorithm was developed in order to overcome deficiencies in the interpolation software available at the beginning of the Alvey MMI-137 project. Initially, DEM remapping was accomplished using Delaunay Triangulation / bilinear interpolation using the commercial Laserscan package Panacea. The problem with the Panacea package is that the number of points in the ungridded input DEM is limited. The Kriging routines developed under the aegis of Alvey MMI-137 do not have this limitation. In addition, Kriging is capable of giving a statistical measure of quality for each point which is Kriged. If a terrain variogram is available Kriging is more accurate than the Panacea package. If a terrain variogram is unavailable, the accuracy of the Kriging process is of the same order as Panacea.

In addition to the Kriging algorithm, the Alvey MMI-137 project also developed its own Delaunay triangulation/bilinear interpolation scheme capable of processing an arbitrary number of ungridded input points.

2.5 Development of visualisation tools under Alvey MMI-137.

A complete set of dynamic visualisation tools was developed under the aegis of the Alvey MMI-137 project by Paramananda [Paramananda, 1988a, Paramananda, 1988b, Paramananda, 1988c]. This visualisation toolkit was based on two standards which predated the Alvey MMI-137 project: The HIPS picture header standard described by Landy and Cohen [Landy and Cohen, 1982], and the Sunview environment which is the standard windowing environment on Sun Workstations running SunOS 3.0 and SunOS 4.0 operating systems. The basic visualisation toolkit consists of the following
three items:

a) points: a HIPS tool for the manual estimation of feature coordinates with sub-pixel precision on single or paired images.

b) disp: a HIPS tool for general-purpose image display with LUT manipulation.

c) stereo: a HIPS tool for the manual measurement of feature point disparities in digital image pairs.

In addition to the programs described above, a number of other general visualisation tools were developed during the lifetime of the Alvey MMI-137 project. For example, the Gruen_view HIPS tool was developed in order to assess the stereo coverage generated by the Otto-Chau algorithm. Other specialist tools, for example the fly program were developed to facilitate the production of animated movies produced by the evileye renderer [Day, 1988], which uses DEMs and orthoimage as input.

3 Post Alvey Development of the Topographic Mapping System.

After the end of the Alvey Project in July 1989, the further development of the major components of the topographic mapping system was supported by funding from the Royal Aircraft Establishment and Laserscan Laboratories in order to try and produce a commercial topographic mapping system based on the Alvey software.

The period immediately following the end of the Alvey MMI-137 project saw the integration of the stereo matcher, camera model and interpolator into the Geodem System, this is an integrated system which produces DEMs from stereo SPOT-1 imagery.

3.1 Post Alvey development of the stereo matcher.

The stereo matcher has undergone significant changes since the end of the Alvey Project. Improved schemes for parallelisation of the stereo matcher have been devised and implemented on both an Ethernet of Sun Workstations
Thesis Background and Plan

and on a 64 T800 transputer Supernode System [Zemerly et al, 1991]. The stereo matcher has also been extended to deal with multispectral imagery, and multi-image matching and adaptive patch size schemes have been investigated [Upton, 1990]. In addition, the applicability of the stereo matcher to urban imagery has been investigated [Denos, 1989; O’Neill and Denos, 1991] and to human faces [Thomas, 1989].

3.2 Post Alvey development of camera model.

The camera model has also been improved over this time period. The principal improvement has been the provision of a fast, accurate back transform, in which points in object space are projected into the image spaces of one or more of the sensor looks which form the stereo model. The efficient implementation of this operation is required for the production of orthoimages. This is an essential part of any full-scale topographic mapping process. As a result of the fast back-transform implementation, the overall throughput of the model was greatly increased, and correction of software bugs at the same time increased the overall accuracy of the model.

A method of statistically pruning poorly measured ground control and check points, shift-pruning was also introduced into the model at this time. Provision was also added to the model to deal with contiguous strips of SPOT-1 imagery efficiently.

3.3 Post Alvey development of interpolator.

Work was also continued on the kriging interpolation scheme after the end of the Alvey project. The principal areas of development were the investigation of robust methods of estimating the local terrain variogram, in order to improve the accuracy of the interpolation process, and improving the overall computational efficiency of the algorithm.

3.4 Post Alvey development of the visualisation software.

Post Alvey work on the visualisation software has concentrated on improving the reliability and functionality of the code which was developed under the Alvey MMI-137 project. For example, the disp and stereo programs have been recoded to make use of features present in the latest release
of SunOS such as the memory mapping, which enhances program response time and reduces the memory overhead. Custom variants of the visualisation toolkit have also been produced for a number of clients including the German Space Agency [DLR].

The possibility of modifying the visualisation toolkit to support windowing standards other than Sunview has also been considered. For example, the use of the Xview package to port software between Sunview and the increasingly ubiquitous X/open window standard has been investigated.

4 Thesis Plan.

This thesis consists of an introduction and 8 Chapters. The thesis is divided into six sections:

a) Section 1: comprising the introduction, describes the origins and history of the O'Neill-Dowman camera model and the other principal components of the SPOT-1 topographic mapping system developed at UCL. The reason why each of the components was developed under the Alvey MMI-137 project is considered, and the development history of each of the components to August 1990 is reviewed.

b) Section 2: comprising Chapter 1, provides background material about the SPOT-1 mission. The SPOT-1 satellite, its ground control segment, the HRV instruments modelled by the O'Neill-Dowman camera model are all described, together with a review of the mission objectives. This Chapter draws freely on information published in the SPOT-1 user's manual [CNES, 1987], and in the paper describing the SPOT-1 mission [Chevrel et al, 1981], from which the illustrations and tabular material in Chapter 1 are derived.

c) Section 3: comprising Chapters 2, 3, 4 and 5, describe the O'Neill-Dowman camera model algorithm and the tests which have been conducted upon it.
Chapters 2 and 3 describe the algorithm development and testing which was conducted in the period January 1987 - July 1989, under the aegis of the Alvey MMI-137 project. In this period the basic camera model algorithm was defined, coded and tested.

Chapters 4 and 5 describe the algorithm development and testing conducted in the period July 1989 - June 1990 under the aegis of the RAE LPO contracts and the RAE/LSL development subcontract. In this period, a number of amendments were made to the camera model, including the fast back transform scheme, the use of shift-pruning to detect erroneous ground control and check points, and the provision within the model to deal with contiguous strips of SPOT-1 stereo imagery.

d) Section 4, comprising Chapter 6, puts the camera model into context. The role of the camera model within the Geodem topographic mapping system is discussed. The Chapter describes each of the principal components of the Geodem system and proceeds to discuss the ways in which the product DEMs may be used.

e) Section 5, comprising Chapter 7, considers where the research may progress post thesis. The Chapter has two major themes: (1) the development of a generalised camera modelling system which uses the modular system which has been developed to date as a basis and (2) the development of sensor assisted stereo matching systems and their use to produce dense digital disparity models of discontinuous imagery.

f) Section 6, comprising Chapter 8 summarises the conclusions reached.

In addition to the thesis, there are six supporting appendices:
a) Appendix 1 consists of a complete set of results, which were compiled during the testing of the initial [Alvey MMI-137] version of the O'Neill-Dowman camera model.

b) Appendix 2 is a comparison of the O'Neill-Dowman camera model with the models which were assessed at the OEEPE workshop at University College London in October 1989: the ground RMS accuracy of the O'Neill-Dowman model is compared with that of the other SPOT-1 camera models.

c) Appendix 3 consists of a complete set of results, which were compiled during the testing of the LSL/RAE implementation of the O'Neill-Dowman camera model.

d) Appendix 4 describes the modular software system used to fabricate the camera modelling system described in the thesis, including flow-charts of the LSL/RAE implementation of the model.

e) Appendix 5 contains the UNIX "man" pages describing the UNIX-filter implementation of the SPOT-1 camera model and its support filters.

f) Appendix 6 is an abridged account of contract work conducted for the Japanese Geoscience Institute JAPEX, on sensor pointing requirements for the ITIR sensor, which is to be flown as part of the EOS programme. The ITIR sensor model which was developed for JAPEX is based on the generic camera modelling system derived from the O'Neill-Dowman SPOT-1 camera model software.

Endnotes to introduction and thesis plan.

1: Subsequently, the software system derived from the Alvey MMI-137 project, of which the O'Neill-Dowman camera model forms an important component, was awarded the 1990 BCS [British Computer Society] Award for Technical Merit.
Chapter 1
The SPOT-1 Satellite System

1.1 Introduction.

Development of the SPOT [Systeme Pour d'Observation de la Terre] programme was initiated by the French Government in February 1978, the aim being to launch the first satellite in early 1984. Sweden and Belgium subsequently decided to participate in the SPOT programme and have contributed some space and ground based hardware. The following account of the SPOT multimission platform, the SPOT-1 HRV instruments and the supporting ground segment is based on the information given in the paper by Chevrel et al [Chevrel et al, 1981] and the SPOT-1 handbook [CNES, 1986].

The SPOT system has been designed as the forerunner to a series of Earth observation missions to be launched into a low earth orbit from the French developed ARIANE space vehicle, launched from the ESA site at Kourou in French Guiana.

The SPOT system consists of a multimission platform or bus, a mission specific payload and a ground segment.

1.1.1 The SPOT-1 satellite.

The objectives of the SPOT-1 mission were:

a) To experiment on desirable characteristics of future operational remote sensing systems.

b) To create an archive, and make available a worldwide database of remotely sensed imagery for cartographic and Earth resources exploration purposes.
c) To experiment on improving vegetative species discrimination and production forecasting based on frequent image acquisition, and off-nadir viewing.

d) To build up a stereo archive of areas of recognised interest for photointerpretation and planimetric cartography at scales of 1:250,000 and cartographic updating at scales of 1:100,000 and 1:50,000.

e) Qualify the multimission platform and a linear array camera for operation within the design envelope specified by CNES [CNES, 1979].

1.1.2 The SPOT-1 mission specifications.

The objectives of the first SPOT mission given above translate into the following payload design and satellite operational specifications:

a) Complete equatorial coverage capability when the system is used in a fixed nadir-looking operating mode.

b) Rapid access to any point on the surface of the planet. This implies that the facility for observing a designated area more frequently than is possible with fixed-nadir viewing must be designed into the system.

c) Acquisition of stereoscopic pairs over a short time period.

d) High ground resolution.

e) Spectral bands chosen to be good indicators for vegetation status and vegetative species discrimination.

1.2 The Multimission Platform.

The SPOT satellite system has been designed as a modular system. The SPOT satellite adopts an architecture comprised of two main parts, as indicated in Figure 1.1. Firstly, the platform, which carries the mission indepen-
Figure 1.1 Showing the components of the SPOT-1 satellite system.
dent subsystems of the satellite including the attitude and orbit control, power supplies, onboard computer, telemetry and command equipment. The design of the platform is such that it is usable for a number of Earth observation missions. Secondly, the payload which includes the instruments and other mission specific equipment.

1.3 The SPOT-1 Payload.

To meet the above requirements, the SPOT-1 payload includes two identical High Resolution Visible range instruments [Optical HRV sensors]. In order to facilitate rapid access to any point on the globe and for the acquisition of stereoscopic pairs, these instruments are pointable across the track direction. A schematic diagram of the HRV sensor instrument is shown in Figure 1.2. Data generated by the instruments is either transmitted to ground over a payload specific X-band telemetry link, or stored by two on-board recorders for later recovery by the Mission Control Centre at Toulouse.

The design features of the HRV sensor include:

a) A panchromatic imaging mode with a ground sampling interval [pixel size] of 10 metres.

b) A multispectral imaging mode with a ground sampling interval [pixel size] of 20 metres.

c) A cross-track pointing system [with a maximum pointing angle of 27.5 degrees] to provide an off-nadir viewing capability for the rapid acquisition of stereoscopic images.

d) A bit-rate within 25 megabits/second/channel. This is expected to be compatible with the data-processing capabilities of the SPOT-1 satellite ground-segment over the lifetime of the satellite. The total telemetry rate is 50 megabits per second [with both HRV instruments operating concurrently].

e) 256 grey levels over the full radiometric range [8-bit encoding in the multispectral mode].
Figure 1.2 The SPOT HRV Instrument (after Chevrel et al., 1981)
These constraints have resulted in the selection of a 60 kilometre swath width for the HRV sensors. Because of the high ground resolution, the HRV instruments form images without any mechanical parts [such as scanning mirrors, disk-choppers, or mechanical modulators] being used. Instead, the images are obtained using the pushbroom scanning technique [Boyle and Smith, 1970; NASA, 1972; Thompson, 1979]. In the pushbroom system, each line of the image is electronically scanned by a linear array of detectors, which are located in the focal plane of the instrument. Successive lines of the image are produced as a result of the satellite motion in its orbit. This approach offers two advantages:

a) The exposure time for each ground point is automatically maximised

b) The quasi cylindrical geometry of the instrument ensures excellent photogrammetric quality along the line scan axis of the instrument.

The ground resolution of 20 metres [multispectral imaging mode] over a 60 Km wide swath requires a linear array of 3000 detectors sampled every 3 milliseconds. A ground resolution of 10 metres [panchromatic imaging mode] over a 60 Km swath requires a linear array of 6000 detectors, sampled every 1.5 milliseconds. The HRV instrument meets these requirements using charge coupled device [CCD] elements, which are $26\mu\text{m}$ and $13\mu\text{m}$ square for multispectral and panchromatic images respectively. Because of the design limitations of the HRV instruments, the panchromatic scan line being imaged is 15 Km ahead, on the ground, of the of the corresponding colour scan line.

The amplifier gain for the CCD sensors can be adjusted by ground command to ensure optimal use at all times of the dynamic range of the detectors. In particular, compensation is made for the large variations in the angle of incidence of sunlight on the terrain in each orbit.

1.4 HRV Instrument Optics.
The optical system is illustrated in Figure 1.2, and comprises the following elements:

a) A front end mirror that can be rotated [up to 27.5°] about the roll-axis, permitting selection of target scenes.

b) A folded pseudo-Schmidt telescope, aperture $\frac{f}{3.5}$, focal length 1082 mm with spherical corrector lens.

c) Three dichroic prisms located at the focus for spectral separation.

d) Four beam-splitting prisms, one for each imaged band. This provides precise optical butting of four CCD sub-arrays into a single scan-line.

1.5 The Choice of Spectral Bands.

The choice of the spectral bands used by the SPOT-1 HRV instruments is dictated by both thematic considerations and technical constraints. In the multispectral mode, the major design considerations are:

a) A consistent relationship between the spectral reflectance and vegetational properties [Tucker, 1978; Bunnick, 1978]

b) A good discrimination within vegetated areas and within different soil types.

c) A compatible interpretation of spectral signatures to those obtained from the Landsat D thematic mapper.

d) An improved radiometric sensitivity and resolution for surface water work.

e) At least one spectral band capable of penetrating the water $[H_2O]$ and hydroxyl [OH] bands of the electromagnetic spectrum. This facility allows the instrument to look through cloud.
The choice of bands selected was subject to a number of constraints imposed by the design of the HRV instruments themselves. These include.

a) A sufficiently wide separation of spectral bands to ensure an adequate signal to noise ratio for the required radiometric resolution in all spectral bands.

b) The dichroic beam-splitting system in the HRV instruments requires a 20 nm separation between adjacent bands.

c) The choice of bands adopted must take into account the scattering and absorption characteristics of the atmosphere, [Tanre et al, 1979].

Taking these considerations into account, three bands were selected for the multispectral mode:

A green band [500 to 590 nm] centered around the 550 nm peak in the chlorophyll reflectance curve. This band is on the long wavelength side of the broad attenuation minimum of water [Tyler and Preisendorfer, 1962]. This allows turbidity assessment and bathymetric evaluation to be undertaken in the first 10-20 metres in clear water.

The second band is a red band [610 to 680 nm] similar to channel 5 of the Landsat Multispectral Scanner. This band is intended to provide data on crops, bar oil and rocky surfaces. Atmospheric transmittance within this band is approximately 90% on a fine day. The band is centred on the peak in the chlorophyll absorption curve which is approximately 645 nm.

The third band is in the near infrared [790 to 890 nm]. This band is the one which has the best penetration through the atmosphere. The transmittance of this band is about 95%, given a clear atmosphere and light haze. Vegetation stands out because of its high reflectance, and water surfaces appear very dark [1% reflectance with a high attenuation coefficient]. Vegetal biomass can be evaluated by considering the red and near infrared bands together.
Silicon spectral sensitivity extends to 1100 nm. It was decided not to extend the band beyond 900 nm as this will limit the response modulation effects caused by atmospheric water vapour, and limit the smearing effect of electron diffusion within the detectors.

In order to attain higher ground resolution, a black and white mode [panchromatic mode] was made available. This requires a broader spectral band than the green, red and near infrared bands of the multispectral [XS] imaging system. In order to retain a high capability for texture analysis in support of the XS mode; and a high information content over vegetated areas, the interval 510 nm to 730 nm was chosen for the panchromatic [PAN] band.

1.6 HRV Instrument Pointability.

The HRV instrument has an off-nadir viewing capability which is used for the rapid acquisition of stereoscopic pairs [in both the PAN and XS imaging modes of the instrument]. The line of sight of the HRV instrument can be steered to any of 91 orientations each being 0.6 degrees apart. The central orientations of the two SPOT-1 HRV instruments are offset by 0.163° from nadir. This means that when both HRV's are operated together, their swaths overlap by a nominal 3 Km [nadir].

Access can therefore be programmed to any point target within an off-nadir angle of +/- (0.163 + 27 + 2.065) degrees. This translates on the ground to a band of access 950 +/- 50 Km wide, centered on each satellite track. Selectable scene centres are about 10 Km apart. Perspective and distance degrade the cross-track resolution by increasing the ground sampling interval in the cross-track direction by up to 24% at the outer border of the access band. The higher off-nadir angles also effect the radiometric significance of a scene due to the changing geometry of observation and illumination, and the increased airmass.
1.7 The Satellite Orbit.

The orbital parameters of the SPOT-1 mission have been chosen as a trade-off between many, often conflicting criteria. As for previous remote sensing satellites, the orbit is low [approximately 830 Km] to give a high ground resolution. It is also of low eccentricity [nearly circular], in order to give a nearly constant scale of observation over all areas. The orbit is near-polar, sun-synchronous and phased. This means that the satellite achieves worldwide coverage and that successive images of a given site are obtained under similar conditions of viewing and illumination throughout the year [Brooks, 1977]. A summary of the SPOT-1 mission orbit characteristics are given in table 1.1.

1.8 SPOT-1 Satellite Operating Modes.

The SPOT-1 Satellite has two major operating modes for its two HRV instruments:

a) The *twin mode* in which both instruments operate jointly, with a 0.206 degree overlap. In this mode, the combined swath width of the instruments is 117 Km [nadir] to 150 Km [off-nadir].

b) The *independently pointed mode*, in which the directions and resolutions of the HRV's are unrelated. This mode is intended to be used on request to acquire imagery of a specific sample area, and to reduce the time required to acquire imagery of areas in which cloudy conditions tend to prevail. It is notable that useful data is only attainable when the pointing mirror is stationary. Typically, *two scenes* [each 60 Km x 60 Km] will be lost when the pointing mirror of the HRV instrument is re-positioned.
<table>
<thead>
<tr>
<th>Orbital parameter</th>
<th>Nominal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revolutions/day</td>
<td>$14 + \frac{5}{26}$</td>
</tr>
<tr>
<td>Nodal Period</td>
<td>101.46 min</td>
</tr>
<tr>
<td>Mean altitude (45 deg Northern Lat)</td>
<td>832 km</td>
</tr>
<tr>
<td>Inclination</td>
<td>98.70 deg</td>
</tr>
<tr>
<td>Orbital repeat period</td>
<td>26 days</td>
</tr>
<tr>
<td>Number of tracks</td>
<td>369</td>
</tr>
<tr>
<td>Intertrack distance (equatorial)</td>
<td>108.4 km</td>
</tr>
<tr>
<td>Accessibility pattern (45 deg lat)</td>
<td>1,4,1,4,1,4,1,4,1,4,1,4,1 days</td>
</tr>
<tr>
<td>Mean local solar time at descending node</td>
<td>10:30 AM</td>
</tr>
</tbody>
</table>

Table 1.1 Orbital parameters for SPOT-1

<table>
<thead>
<tr>
<th>Instrument parameter</th>
<th>Colour mode</th>
<th>B/W mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground sampling step (m)</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Array sampling period (m sec)</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>Spectral bands (μm)</td>
<td>xS1 0.50-0.59</td>
<td>p 0.51-0.73</td>
</tr>
<tr>
<td></td>
<td>xS2 0.61-0.68</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xS3 0.79-0.89</td>
<td></td>
</tr>
<tr>
<td>Grey levels</td>
<td>256 (8 bits)</td>
<td>128 (6 bits) or 256 comp. to 6 bits</td>
</tr>
<tr>
<td>Data rate (M bits/sec)</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Swath width (km)</td>
<td>60 (nadir)</td>
<td>60 (nadir)</td>
</tr>
<tr>
<td>Central viewing direction</td>
<td>+0.163 deg</td>
<td>-0.163 deg</td>
</tr>
<tr>
<td>Field steering direction</td>
<td>0 +/- 45 steps 0.6 deg apart</td>
<td></td>
</tr>
<tr>
<td>Outermost observable direction (deg)</td>
<td>+/- 29.23</td>
<td>+/- 475</td>
</tr>
<tr>
<td>Corresponding track distance (km)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.2 Main instrument parameters for each SPOT-1 HRV instrument.
1.9 The Supporting Ground System.

The ground system for SPOT [Figure 1.3], performs three functions:

a) There is a *mission centre* which is responsible for the management of the mission, the scheduling of image acquisition and the processing of the data in accordance with the user requirements.

b) The *ground control segment* performs all satellite management functions, monitors satellite operation, generates commands and establishes the orbit parameters. During the operational phase of the SPOT-1 mission, the Control Centre in Toulouse handles all ground command operations, while the Kourou Station in French Guiana acts as a backup in the event of failure. During the acquisition of imagery, additional ground stations may be used. All command stations are linked to the Mission Control Centre via a high-speed communications network.

c) The *Toulouse image receiving ground segment* receives the payload telemetry data, documents and stores the data and processes it into primary data products.

1.10 Description of SPOT-1 Sensor Geometry.

1.10.1 Overview of pushbroom sensor geometry.

The SPOT-1 sensor is a member of a class of sensors known as *pushbroom sensors*. A pushbroom sensor system consists of a linear array of detectors which is oriented perpendicular to the flight path of the instrument. Successive lines of the image are built up as a result of sensor movement along the flight path. The attitude and position of the sensor varies dynamically during the time period in which imagery is acquired. Thus the sensor geometry is *dynamic*, and the resulting image is composed of \( N_t \) lines with
Figure 1.3 Schematic of the ground support facilities for SPOT-1.
Compared to other sensor systems, pushbroom sensors have a number of advantages:

a) The exposure time for each ground point is maximised,

b) The imaging system contains no mechanical parts, unlike for example thematic mappers. This ensures excellent photogrammetric quality along the pushbroom axis.

c) The use of a linear rigid CCD array gives the combination of high spatial resolution and geometric stability essential for accurate feature identification [Gugan, 1987].

1.10.2 The SPOT-1 sensor geometry.

The SPOT-1 sensor is an example of pushbroom sensor system for across track or side-looking stereoscopy. In the SPOT-1 sensor system, the along-track pointing angle of the sensor is approximately zero [The optical plane of the SPOT-1 is offset by \( \alpha \approx +/-0.16^\circ \) off vertical for the PAN and XS sensors arrays respectively]. The cross-track pointing angle, \( \beta \) is generally non-zero. The precise value of \( \beta \) is dependent on the image which is being acquired.

In cross track stereoscopy, the images which constitute a stereo pair are acquired on successive orbits. In the worst possible cases the constituent images may have been acquired several months apart. This has a number of implications which are further discussed in Chapter 6.

1.10.3 Aberrations in SPOT-1 geometry.

In space borne pushbroom sensor systems such as SPOT-1 there are several geometric effects which must been taken into account in any geometrical model of the sensor:

a) During image acquisition, successive scanlines shift westward as a result of earth rotation. The amount by which successive scanlines shift is a function of the latitude, it being maximal at
b) The line acquisition time is a function of the flying height of the sensor. Therefore height variations will cause scale variations in the along track axis of the acquired imagery.

c) In the case of off-nadir SPOT-1 looks, the line sampling interval increases as a result of the increased distance between the sensor and the target, compared to a nadir view. In the case of extreme off-nadir SPOT-1 views, this leads to a ground pixel size of 12.4m. The nominal nadir ground pixel size is 10.0m.

d) The curvature of the earth deforms the geometry of extreme off-nadir looks.

e) Pitching attitude variations may result in the positions of successive lines being interchanged and/or superimposed. Roll attitude variations cause shifts of the image scanlines in a direction parallel to the pushbroom axis. Yaw attitude variations produce a rotation of the image scanlines.

1.11 An Overview of Current SPOT-1 Sensor Models.

To date, a number of different SPOT-1 sensor models have been developed for Photogrammetric purposes. These include:

a) The Guichard model [Guichard and Pikeroen, 1988; IGN/CNES].

b) The Picht model [Picht, 1989; University of Hannover].

c) The Haas model [Haan, 1989; Politecnico di Milan].

d) The Kratky model [Kratky, 1989; Department of Minerals and Mines, Ottawa].

The SPOT-1 Satellite System

1.11.1 The Guichard model.

Guichard's method is a simple, fast and efficient method of modelling the SPOT-1 sensor. Unfortunately, the publication describing the sensor [Guichard and Pikeroen, 1988] is not very explicit: A great number of complex formulae are presented, but the parameters and coefficients in these formulae are very poorly documented. This is unfortunate as the Guichard model has given a very good account of itself when it has been compared against other SPOT-1 sensor models. Typically the Guichard model give RMS residuals of 9.1m in plan and 5.2m in height for a single SPOT-1 stereo model.

The Guichard model uses the line number within an image as a reference for time variation. Measured image co-ordinates are first corrected for attitude variation. Then the header derived ephemeris is transformed into orbital parameters. Two collinearity equations are introduced for each ground control point used. The resulting model is formed in a similar manner to the Gugan-Dowman model. Apparently, the Guichard model has a minimum of 10 unknown parameters. This implies that a minimum of 5 ground control are required per SPOT-1 stereo-pair, in order to set up a model. However, IGN has presented work at the OEEPE workshop at UCL in which this model was apparently used with only two ground control points. The accuracy of the Guichard model compared to the other models may be explained by IGN/CNES having access to complete SPOT-1 telemetry data, the SPOT-1 header data used by the other centres being in an abridged form.

1.11.2 The Picht model.

The SPOT-1 sensor model developed at Hannover by Picht is part of the BINGO system. BINGO is a general purpose camera modelling system. In addition to the SPOT-1 sensor, the system supports a number of other sensor systems including central perspective aerial photography.
The ground co-ordinate system adopted by the BINGO system is a local Cartesian co-ordinate system. The SPOT model supported by the system uses ephemeris data in order to set up a rough camera model. The sensor positions given in the header data are transformed into an appropriate local co-ordinate system using a set of 6 polynomials, the higher order coefficients of the polynomials being a function of the local co-ordinate system considered. The variation of attitude during the period of image acquisition is modelled by introducing additional high order polynomial coefficients. The BINGO SPOT-1 sensor model is described by Picht [Picht, 1989].

Because SPOT-1 header data is used the BINGO SPOT-1 sensor model is able to work with a small number of ground control points. Typically, the model requires two ground control points in order to form a stereo model giving RMS residual errors 14.0m in plan and 10.0m in height.

1.11.3 The Haan model.

Haan's model is another example of a SPOT-1 sensor model which makes use of header data in order to reduce the amount of ground control which is required to construct a model. The model finds the sensor position in the instrument reference co-ordinate system, at the times when the ground control points were acquired. These sensor positions are estimated from using the scene centre time and the line positions of the ground controls points within the imagery. A rotation matrix is then computed which transform co-ordinates between the geocentric and instrument reference systems. The initial position of the satellite and its attitude parameters are the unknown parameters in the model. Initial estimates of these parameters, which are subsequently refined by the model, are obtained as a function of time via cubic spline interpolation from the SPOT-1 header. The model also uses the HRV sensor pointing angle data given in the header.

Changes in the sensor attitude are correlated within the model to shifts in sensor position. Failing to constrain the attitude in this manner causes instability in the model, which leads to an erroneous model. The model is described by Haan [Haan, 1989]. This model, like Picht's model only requires 2 ground control points in order to set up a stereo model with low RMS residuals.
1.11.4 The Kratky model.

The Kratky model does not make use of SPOT-1 header data. It is able to form a SPOT-1 stereo model using a minimum of 4 ground control points. Like the Guichard and Gugan-Dowman models it uses a collinearity based approach. The Kratky model uses an extended bundle formulation which is applied to ground control points and tie-points. The method solves for 28 unknown parameters when forming the stereo model. The sensor position in the geocentric co-ordinate system is described by a set of three second order polynomials. The sensor roll, pitch and yaw are described by a further set of three quadratic polynomials. This attitude model may be simplified by only considering the zeroth and first order polynomials, thus reducing the number of unknowns to 12. Four additional parameters are introduced in order to compensate for the lack of information about the sensor calibration. The model is soluble with a minimum of four ground control points if linear attitude changes are assumed. With a second order attitude model, five ground control points and two tie-points are required for a solution. Typically RMS residuals using this model are in the order of 21m in plan and 17m in height.

1.11.5 The Westin Model.

Westin's [Westin, 1990] model is yet a further example of a SPOT-1 model which uses the SPOT-1 header data in order to reduce the number of ground control points required to form a stereo model. Typically, this model requires 2-3 ground control points in order to form a stereo model with residual errors of less than 15 metres RMS over a strip of 6 scenes. The model uses the SPOT-1 ephemeris to make an initial estimate of the satellite position. The satellite attitude is computed indirectly from the SPOT-1 header. A set of variograms are computed from the SPOT-1 header; these are then used to compute a further set of functions which describe the attitude variation as a function of time. Because the variograms model general trends in the attitude variation for the SPOT-1 sensor, it is possible to accurately reconstruct long orbit segments using the raw attitude data from one or two SPOT-1 scenes. This means that the model is of potential use for orienting long strips of imagery which may or may not be contiguous. The variogram computa-
The SPOT-1 Satellite System

tion process and associated model have been described by Westin [Westin, 1990].

1.11.6 The Gugan-Dowman model.

The Gugan-Dowman model [Gugan and Dowman, 1987; Gugan, 1987; Gugan, 1988] is one of the earliest SPOT-1 models to be described. The initial design of the model pre-dated the launch of the SPOT-1 sensor on the 22nd of February 1986. The only items which the model requires from the SPOT-1 header are the sensor pointing angles. All of the other parameters required by the model, the along-track sensor offset, the orbit eccentricity, the rate of change of true anomaly, and the ascending node are fixed. The stereo model is set up using a time-dynamic collinearity equation based approach. The time dynamicism has been introduced into the model because both the perspective centre and satellite attitude vary as a function of line number [and time]. The Gugan-Dowman model assumes a smooth elliptical orbit of low eccentricity. Attitude variations are modelled using a set of low-order polynomials.

With linear attitude variation, the model has to solve for 10 unknown parameters. These are the ascending node $\Omega_o$, the orbit inclination $i$, the semi-major orbit-axis $a$. In addition there are six angular parameters, $\omega, \psi, \kappa$, and their associated linear rates of change. With a linear attitude model, at least 5 ground control points are required for a rigorous solution.

With second order attitude variation, a further three parameters, the second order rates of change of $\omega, \psi$ and $\kappa$ are introduced. This 13 parameter model requires at least 7 ground control points for a rigorous solution.

Typically a 5 parameter model gives absolute RMS residuals of 15m in plan and 8m in height. A 7 parameter model gives absolute RMS residuals of 25m in plan and 10m in height.
2.1 Introduction.

The O’Neill-Dowman SPOT-1 camera model has been implemented under the aegis of the Alvey MMI-137 [Real Time 2,5D Vision Systems] project at University College London. The camera model was designed to provide a fast accurate method of transforming SPOT-1 imagery from image space to object space and vice versa, using the minimum amount of ground control to orient the models. Furthermore, the camera model is designed to work accurately with both single SPOT-1 stereo-pairs and strips, which are continuous swaths of SPOT-1 imagery up to several hundred kilometres in length.

2.2 Design Considerations of the O’Neill-Dowman Camera Model.

The three basic design considerations for the O’Neill-Dowman Camera model are:

a) To make use of all available auxiliary information in order to reduce the number of ground control points [GCPs] which are required to form the model.¹

b) Functional simplicity: the model makes use of simple, well understood algorithms when possible to achieve the desired result. In spite of adhering to this approach, the model is quite complex: this is because the satellite telemetry data is of
relatively poor quality. This means that complex relaxation and orbit reconstruction mechanism is required to achieve an accurate model. Accurate orbit parameterisation would permit a more concise algorithm, similar to that which was developed to simulate the Landsat MSS orbital image-sensor geometry, described by Forrest [Forrest, 1981].

c) A modular code structure: this facilitates experimentation with the camera model. Modular code in the form of libraries also permits generally useful routines to be available for other applications.

2.3 An Overview of the O’Neill-Dowman Camera Model in Operation.

Setting up a stereo model using the O’Neill-Dowman camera model is a two stage process:

(a) Setting up a relative model using the SPOT-1 telemetry data.

(b) Orienting the relative model to an absolute co-ordinate system using a small number, typically 2 or 3, GCPs.

2.3.1 Computing the relative model.

In the first stage of the computation, a time variant rough orientation matrix, \( R_\theta(t) \), of the relevant SPOT-1 orbit segment is computed by splining the satellite position and velocity datasets, using a beta cubic spline.

A correction term to the rough attitude matrix is then determined using the header attitude dataset. This is splined to give the time variant matrix, \( R_x(t) \). The relative orientation matrix, \( R_{rel}(t) \), may then be computed:

\[
R_{rel}(t) = R_x(t) \cdot R_\theta(t)
\]  

The computations described above are performed for each camera position or look [scene]. For a stereo-model comprising \( N \) looks, \( N \) time variant orientation matrices must be computed.
Because of the linear geometry of the SPOT-1 camera, the time variant attitude matrix may also be expressed as a function of camera line position:

\[ l = k \cdot t + c \]  
(2.2a)

\[ R_{rel}(l) = R_{rel}(k \cdot t + c) \]  
(2.2b)

Where:

\( k \) and \( c \) are constants.

\( l \) is the line position.

\( t \) is the corresponding acquisition time for line \( l \).

2.3.2 Computing the absolute model.

The second stage of the modelling process is to orientate the model to a known ground [object] co-ordinate system, using \( N \) GCPs. The GCP’s are used to compute a correction to the attitude matrix, \( R_g(t) \). The amended attitude matrix for each look becomes:

\[ R_a(t) = R_g(t) \cdot R_{rel}(t) \]  
(2.3)

Where:

\( R_a(t) \) is the absolute orientation matrix.

It is also sometimes necessary to introduce a constant corrective shift of orbit position, \( \Delta p \), for a given look. The ray unit vector emergent from an image space co-ordinate \([l, s]\), \( \hat{r}(l, s) \) may then be computed via the expression:

\[ \hat{r}(l, s) = R_a(l) \cdot \hat{r}_{ref}(s) \]  
(2.4)

Where:

\( \hat{r}(l, s) \) is the object space ray unit vector emergent from the image co-ordinate \([l, s]\).

\( \hat{r}_{ref}(s) \) is the camera reference space ray unit vector emergent from the image co-ordinate \([l, s]\). This is computed using the nominal camera look angle data given in the SPOT-1 header.
2.3.3 Computing the space intersection.

For a pair of conjugate unit ray vectors \( r_1(l,s) \), and \( r_2(l',s') \), emergent from the left and right looks of a stereo pair respectively, the point of space intersection used by the model is defined to be the mid point of the vector \( \hat{n}_{1,2} \), which is perpendicular to both rays, at their point of closest approach. We will call this position vector \( \hat{s}_{1,2} \). For a system with more than two looks, the space intersection is defined to be the average mid-point which is computed by vector summation of all non-degenerate pairwise combinations of looks.

At this stage, a few further definitions will be of use:

The magnitude of the vector \( \hat{n}_{1,2} \) is defined to be the ray-ray skewness associated with the pair of unit ray vectors \([r_1(l,s), r_2(l',s')]\).

For any GCP or check point, the error vector \( \mathbf{E} \) is defined to be:

\[
\mathbf{E} = \hat{s} - \hat{G}
\]

Where \( \hat{G} \) is the ground position vector of the GCP or check point.

\( \hat{s} \) is the position vector of the corresponding space intersection, which is where the camera model predicts the vector \( \hat{G} \) to be.

2.3.4 The optimisation process.

We are now in a position to see how the matrix \( R_{g}(t) \) which gives the absolute correction to the camera orientation for a given look is computed. This is accomplished using an numerical optimisation technique based upon the Powell direction-set algorithm. [Powell, 1964].
The summed RMS magnitude $|E|$ of a set of $N$ error vectors $E_1^*, E_2^*, \ldots E_N^*$ defined is minimised with respect to an appropriate parameter space using the Powell Direction Set algorithm:

$$C = f(\theta_1, \phi_1, \vec{\Delta p}_1, \theta_2, \phi_2, \vec{\Delta p}_2)$$

(2.6)

Where:

- $C$ is the cost.
- $\theta_1$ is a rotation about the $\hat{v}$ sensor reference axis for look 1.
- $\phi_1$ is a rotation about the $\hat{e}$ sensor reference axis for look 1.
- $\vec{\Delta p}_1$ is a global vector shift applied to the orbit segment of look 1.
- $\theta_2$ is a rotation about the $\hat{v}$ sensor reference axis for look 2.
- $\phi_2$ is a rotation about the $\hat{e}$ sensor reference axis for look 2.
- $\vec{\Delta p}_2$ is a global vector shift applied to the orbit segment of look 2.

The parameter space with respect to which the cost function is to be optimised consists of a set of geometric operations which are simultaneously applied to each look of the camera model. These include:

a) **Uncorrelated** vector shifts of the orbit tracks of each look $[\vec{\Delta p}_1, \vec{\Delta p}_2]$.

b) **Uncorrelated** rotation of the line elements of the camera about their respective perspective centres. Two rotations $[R_b, R_e]$ are required so that a ray emergent from a given perspective centre can explore any point in space.

The optimisation scheme is set up so that additional constraints, for example temporal correction of satellite position may be added without gross changes to the coding of the algorithm.
The model currently uses the Powell direction Set relaxation algorithm: extensive testing has shown this to be the best choice of optimisation algorithms in terms of both computational efficiency and the ease with which the parameter set and its associated cost function may be changed. The camera model has operated in a satisfactory manner with a number of other minimisation algorithms. These include stochastic multimodal optimisation schemes: a relaxation scheme based on the genetic search method, [Davis and Steenstup, 1987; Booker, 1987], and a scheme which uses simulated annealing, [Metropolis et al, 1953; Christofides et al, 1979; Kirkpatrick, 1983]. Multimodal relaxation schemes such as these are very inefficient in computational terms. Analysis of the camera model showed the optimisation problem to be essentially unimodal. The multimodal relaxation schemes were therefore abandoned in favour of the more computationally efficient unimodal schemes.

The camera model has also operated in a satisfactory manner with unimodal algorithms other than the Powell Direction Set method. For example, the downhill simplex minimiser [Nelder and Mead, 1965] has been used experimentally. The method is not used with production versions of the camera modelling system as it is less computationally efficient than Powell’s method.

It is also possible to optimise the relative orientation by introducing the sum of conjugate point ray-ray skewnesses into the parameter space of the relaxation scheme. The conjugate points are obtained from the Otto-Chau stereo matcher output. The ray-ray skewness is shown schematically in Figure 2.1.

2.3.5 The back transformation.

Because the SPOT-1 camera model possesses a linear geometry, with many perspective centres, the back transformation, must be accomplished dynamically via a relaxation scheme. The initial scheme selected minimises the error vector magnitude, $|\vec{E}|$, of a trial ray and a ground point whose corresponding image co-ordinate is required. The image co-ordinates [I',s'] of that ray for which $|\vec{E}|$ is a minimum is taken to be the image co-ordinate corresponding to the ground point. This approach produces an accurate result,
Figure 2.1 Showing the definition of the ray-ray skewness and RMS error vectors.
but it is very slow, because of the large number of floating point operations required to calculate each point.

2.4 A Detailed Description of the O’Neill-Dowman Camera Model.

Having briefly looked at how the model is computed, we shall now look at the individual stages of model formation in greater detail.

2.4.1 A description of the SPOT-1 header data.

In order to produce an accurate camera orientation, the first problem to be solved is the computation of an accurate initial estimate of the SPOT-1 camera position over the time period of image acquisition for each look within the model. This initial phase of the computation uses the SPOT-1 header data to compute initial estimates of the camera position and reference axes over the time periods of interest.

The SPOT-1 header is a condensation of the satellite telemetry data, for a time frame within which the corresponding SPOT-1 image was acquired. The O’Neill-Dowman Camera Model makes use of the following items from the header file:

a) The SCENE_CEN_TIME is the absolute time, accurate to the nearest millisecond, at which the central line [line 3000 PAN mode; line 1500 XS mode], was acquired.

b) The nominal ground pixel size for nadir view: PIXEL_SIZE_X, PIXEL_SIZE_Y, [10m x 10m PAN mode; 20m x 20m XS mode].

c) The SENSOR_IDENT field, which identifies the mode of the HRV instrument when the image was acquired [PAN or XS].

d) The SATELLITE_POSITION dataset. This gives the vector position of the satellite, in a geocentric co-ordinate system. The position of the satellite is given every 60 seconds over a 9
minute time frame, during which the image was acquired. This data is corrected for earth rotation effects.

e) The *INERTIAL VELOCITY* dataset. This gives the velocity vector of the satellite, in the earth centred *geocentric coordinate system*. The velocity of the satellite is given every 60 seconds over a 9 minute time frame within which the image was acquired. This data is corrected for the effects of earth rotation.

f) The *UNIVERSAL TIME* dataset. This gives the absolute time,\(^3\) [yyymmdhhmmss.sss] to the nearest millisecond, at which the corresponding pairs of satellite position vectors and inertial velocity vectors have been acquired.

g) The *ATTITUDE_DATA* dataset. This dataset consists of a set of angular velocities which are obtained by integrating the *rate gyro* output from the SPOT-1 platform. The *angular velocities* are given as *simultaneous rotational components* about the *instantaneous satellite frame of reference*. The satellite ephemeris is given every second, which corresponds to once every 664.89 lines in the case of a panchromatic [PAN] image, or once every 332.45 lines in the case of a multispectral [XS] image. The absolute time when each item of attitude dataset was acquired may be determined using the absolute *scene centre time*, [SCENE_CEN_TIME], and the known delay time between the acquisition of successive lines in the SPOT-1 image [1.504ms PAN; 3.008ms XS].

h) The *PSI X LAST PIXEL*, *PSI X FIRST PIXEL*, *PSI Y LAST PIXEL*, *PSI Y FIRST PIXEL*, give the *nominal* look angles for the first and last CCD elements in the SPOT-1 camera array. These look angles take into account the different offsets of the PAN and XS telescope offsets from the nadir viewing direction. The PAN instrument has an +0.163° offset from nadir while the XS instrument has an offset of −0.163° from the nadir viewing direction. The definition of the
the nominal pointing angles is shown in Figure 2.2.

The format of the SPOT-1 header file is described in full in the
READCCT reference guide [Laserscan, 1987].

2.4.2 Setting up the relative camera model.

In order to use the SPOT-1 header data to set up the relative camera
model, it must first be converted into a form which is amenable to numerical
modelling.

2.4.3 Splining the SPOT-1 header position and velocity data.

The orbit segment of the camera during the image acquisition period is
generated by interpolating the header position data using an appropriate
method. The method which has been chosen for the present model is cubic
spline interpolation using natural boundary conditions: the second deriva­
tives of the splined function are zero at the ends of the data segment. The
method of cubic spline interpolation is described in detail by Press [Press et
al, 1988]. Splining permits the expression of orbit position vector, \( \vec{P} \), as a nu­
merical vector function of the absolute time \( t \):

\[
\vec{P}(t) = \vec{f}(t)
\]  

(2.6)

The velocity data, \( \vec{V} \), is similarly splined using natural boundary condi­
tions to yield a [numerical] vector function of time:

\[
\vec{V}(t) = \vec{f}'(t)
\]  

(2.7)

2.4.4 Computation of approximate satellite reference axes.

The velocity and position vectors may be used to define a third vector,
\( \hat{e}l(t) \), which is parallel to the axis of the pushbroom camera. The vector
triad, \( \hat{e}l(t), \hat{v}(t) \) and \( \hat{e}l(t) \), form a good approximation to the satellite, and
hence, the camera reference axis system at time \( t \), assuming that the satellite
is instantaneously pointing in the direction of the velocity vector \( \vec{V} \), and the
satellite orbit approximates well to a Keplerian orbit:
Figure 2.2 Showing definition of look angles, PSI_FIRST_X, PSI_FIRST_Y, PSI_LAST_X and PSI_LAST_Y.
\[ \hat{\rho}(t) = \text{vunit}(\vec{P}(t)) \] (2.8a)

\[ \hat{\varphi}(t) = \text{vunit}(\vec{V}(t)) \] (2.8b)

\[ \hat{e}_l(t) = \hat{\rho}(t) \times \hat{\varphi}(t) \] (2.8c)

Where:

\text{vunit} signifies the operation of taking the unit vector in the direction of the argument vector.

\( \hat{\rho}(t) \) is a camera attitude reference axis unit vector in the direction of the satellite position vector.

\( \hat{\varphi}(t) \) is a camera attitude reference axis unit vector in the direction of the camera track.

\( \hat{e}_l(t) \) is a camera attitude axis reference unit vector in the direction of the camera pushbroom array.

The geometry of the SPOT-1 camera is shown schematically in Figures 2.3 and 2.4.

2.4.5 Computation of the rough attitude matrix.

Although (2.8) is an approximate expression, it enables an approximate satellite attitude matrix, \( \underline{R}_o(t) \), to be computed from the vector triad \([\hat{\varphi}(t), \hat{e}_l(t), \hat{\rho}(t)]\). This relative orientation matrix \( \underline{R}_o(t) \) transforms the satellite orientation from an internal camera reference space in which the satellite reference axes are parallel to the vectors \([1, 0, 0] (\vec{r}_{\text{ref}}) \), \([0, 1, 0] (\vec{e}_{\text{ref}}) \) and \([0, 0, 1] (\vec{p}_{\text{ref}}) \) to an approximate object space reference axes. The matrix \( \underline{R}_o(t) \) may be formed using the triad \([\hat{\varphi}(t), \hat{e}_l(t), \hat{\rho}(t)]\):

\[ \underline{R}_o(t) = \begin{bmatrix} \hat{\varphi}[1](t) & \hat{e}_l[1](t) & \vec{p}[1](t) \\ \hat{\varphi}[2](t) & \hat{e}_l[2](t) & \vec{p}[2](t) \\ \hat{\varphi}[3](t) & \hat{e}_l[3](t) & \vec{p}[3](t) \end{bmatrix} \] (2.9)

2.4.6 Computation of direction of an arbitrary ray in object space.
Figure 2.3. Showing the geometry of the SPOT-1 sensor.
Orbit segment splined from SPOT-1 header data.

\[
\begin{bmatrix}
X_s \\
Y_s \\
Z_s
\end{bmatrix} = R_o \begin{bmatrix}
X_r \\
Y_r \\
Z_r
\end{bmatrix} + \begin{bmatrix}
Ps_1 \\
Ps_2 \\
Ps_3
\end{bmatrix}
\]

Xr, Yr, Zr: Reference attitude
Xs, Ys, Zs: Working attitude
Ps: Track shift vector (which may result as a consequence of generating an optimal absolute model).

Figure 2.4 Showing the relationship between reference and working attitude for the O'Neill-Dowman SPOT-1 Camera Model.
Transformation of a ray vector, emergent from pixel position \([l,s]\) via the *perspective centre* which corresponds to line \(l\), imaged at time \(t\), may be accomplished by forming the ray in camera reference space, using the nominal look angles, \(\psi_{x1}, \psi_{y1}, \psi_{x2}, \psi_{y2}\), and then transforming this ray to object space co-ordinates:

\[ r^*_o(l,s) = R_\omega(k \cdot t + c) \cdot r_{nf}^*(s) = r^*_o(k \cdot t + c, s) \]  \hspace{1cm} (2.10)

Where:

- \(r_{nf}^*(s)\) is the unit direction vector of a ray in the reference space emergent from image co-ordinate \(s\).
- \(r^*_o(k \cdot t + c, s)\) is the unit vector of the transformed ray in object space expressed as a function of line acquisition time and sample position.
- \(r^*_o(l,s)\) is the unit vector of the transformed ray in object space expressed as a function of line and sample position.
- \(R_\omega(k \cdot t + c)\) is the relative attitude matrix at time \(t\).

Thus, the vector parametric ray equation is given by:

\[ [\vec{P}(t);r^*_o(t,s)] = \vec{P}(t) + \lambda \cdot r^*_o(t,s) \]  \hspace{1cm} (2.11)

Where:

- \(\lambda\) is a scalar constant.
- \([\vec{P}(t);r^*_o(t,s)]\) denotes the vector parametric ray equation.
- \(\vec{P}(t)\) is the position vector of the camera at time \(t\).

2.4.7 Computation of position vector of ray in reference space.

The position of a ray whose image co-ordinates are \([l,s]\) is determined by linear interpolation of the nominal look angles for the first and last CCD elements in the camera array, \(\text{PSI}_X\_\text{LAST}\_\text{PIXEL}, \text{PSI}_X\_\text{FIRST}\_\text{PIXEL}, \text{PSI}_Y\_\text{LAST}\_\text{PIXEL}\) and \(\text{PSI}_Y\_\text{FIRST}\_\text{PIXEL}\). This yields a pair of rotation angles \(\psi_x\) and \(\psi_y\) which correspond to the pixel at image co-ordinate \([l,s]\). The nadir reference ray, \(r_n \hat{\text{ref}} ([0,0,-1])\) is then rotated about the \(X\) ([1,
0, 0]) and Y ([0, 1, 0] axes of the reference axis set, giving the position vector of the ray in reference space:

\[
\psi_x = \psi_{x1} + \frac{(s - 1) \cdot (\psi_{x2} - \psi_{x1})}{n_s}
\]

(2.12)

\[
\psi_y = \psi_{y1} + \frac{(s - 1) \cdot (\psi_{y2} - \psi_{y1})}{n_s}
\]

(2.13)

\[
r_{\text{ref}}(s) = R_x(\psi_y) \cdot R_z(\psi_y) \cdot r_{\text{ref}}
\]

(2.14)

Where:

- \(\psi_{x1}, \psi_{y1}, \psi_{x2},\) and \(\psi_{y2}\) are the nominal look angles for the first and last elements in the SPOT-1 CCD array.
- \(s\) is the sample position.
- \(n_s\) is the number of CCD elements in the SPOT-1 camera array; [6000 PAN; 3000 XS].
- \(R_x(\psi_x)\) and \(R_y(\psi_y)\) are rotations about the X and Y reference axes of \(\psi_x\) and \(\psi_y\) radians respectively:

\[
R_x(\psi_x) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\psi_x) & -\sin(\psi_x) \\
0 & \sin(\psi_x) & \cos(\psi_x)
\end{bmatrix}
\]

(2.15)

\[
R_y(\psi_y) = \begin{bmatrix}
\cos(\psi_y) & 0 & \sin(\psi_y) \\
0 & 1 & 0 \\
-\sin(\psi_y) & 0 & \cos(\psi_y)
\end{bmatrix}
\]

(2.16)

2.4.8 Computing the space intersection.

Equations (2.6) to (2.16) may be used to establish the position and orientation of the satellite over a series of time periods \(\tau_1, \tau_2, \ldots, \tau_t, \ldots, \tau_N\), within which the respective images were acquired for each of the \(N\) looks considered in the space intersection. This yields a set of vector parametric ray equations for each look within the model of the form:
\[
[\vec{P}_1(t_1), \vec{r}_1(t_1); \vec{P}_2(t_2), \vec{r}_2(t_2)]
\]

(2.17)

For the sake of simplicity, we will consider a *minimum* system, consisting of two *looks*, although the formulae given are readily extendible to encompass systems with 3 or more *looks*. The corresponding rays defined by (2.17) are then *intersected* to yield a set of *geocentric ground vectors*, \( \mathcal{S}'(t_1, t_2) \).

The ray intersection is computed by finding the midpoint position of the vector \( \vec{m}(t_1, t_2) \), perpendicular to the ray unit vectors, \( \vec{r}_1(t_1) \) and \( \vec{r}_2(t_2) \), which connects the vector parametric ray equations associated with each look at their point of closest approach. Since this vector is perpendicular to both of the ray direction unit vectors, it is simple to compute a unit vector in the direction of \( \vec{m}(t_1, t_2) \):

\[
\vec{m}(t_1, t_2) = \vec{r}_1(t_1) \times \vec{r}_2(t_2)
\]

(2.18)

The position of the ray intersection is found by solving the set of linear equations, using either Gaussian Elimination [McGregor and Watt, 1986], LU Decomposition [Press et al, 1988], or singular value decomposition [Press et al, 1988]:

\[
\vec{P}_1(t_1) + \Lambda \vec{r}_1(t_1) + \mu \vec{m}(t_1, t_2) = \vec{P}_2(t_2) + \tau \vec{r}_2(t_2)
\]

(2.19)

\[
\mathcal{S}'(t_1, t_2) = \vec{P}_1(t_1) + \Lambda \vec{r}_1(t_1) + \frac{\mu}{2.0} \cdot \vec{m}(t_1, t_2)
\]

(2.20)

Where:

\( \Lambda, \mu \) and \( \tau \) are scalar parameters established by solving (2.20).

\( \mathcal{S}'(t_1, t_2) \) is the *ray intersection* of the ray pair \([\vec{P}_1(t_1), \vec{r}_1(t_1); \vec{P}_2(t_2), \vec{r}_2(t_2)]\). A schematic of the *ray intersection* is shown in Figure 2.5

\( \vec{P}_1(t_1) \) is the position vector of the camera in the orbit segment of look 1.

\( \vec{P}_2(t_2) \) is the position vector of the camera in the orbit segment of look 2.
Ve, el, Ps are the sensor reference axes

S is the position vector of the computed space intersection

O is the origin of the geocentric co-ordinate system

X, Y, Z are the reference axes of the geocentric co-ordinate system

m is the perpendicular between the ray vectors at point of closest approach

Figure 2.5 Showing space resection for a two look SPOT-1 stereo system.
2.5 The Use of GCPs to Compute the Absolute Attitude.

If GCP's are available, these may be used to find the absolute attitude, using the relative attitude calculated using equations (2.6)-(2.20) above as a starting point. The relative camera model may be used to give a set of space intersections which are related to a set of corresponding check points by a linear vector shift:

\[ \mathbf{g}_i = \mathbf{s}_i + \mathbf{e}_i \]  

(2.21)

Where:
- \( \mathbf{s}_i \) is the \( i^{th} \) ground position generated by the camera model.
- \( \mathbf{g}_i \) is the corresponding \( i^{th} \) check point ground position.
- \( \mathbf{e}_i \) is the \( i^{th} \) error or residual vector.

Computation of the absolute orientation is accomplished by minimising a cost function of the form \( C = \sum_{i=1}^{i=N} |E_i|^2 \) which is associated with a set of \( N \) GCPs \( G_1, G_2, . . . , G_i, . . . , G_N \), iteratively with respect to a parameter space consisting of camera position and orientation and any additional parameters which may be introduced as a result of experimentation.

The position and orientation of the camera are known, as functions of \( t \). The associated derivatives with respect to \( t \) are not known, and are not readily computed. Given a stereo conjugate pair \([l, s; l', s']\), experimentation has shown that the set of residual error vectors \( E_{m_1}, . . . , E_{m_i}, . . . , E_{m_N} \) for a given ground control configuration is a global minimum of parameter spaces which consist of camera rotations and shifts. Therefore, the optimal form of relaxation algorithm is a unimodal minimisation scheme which does not require the computation of derivatives. Two numerical relaxation algorithms were identified which fit these criteria:

a) The Downhill Simplex Method [Nelder and Mead, 1965].
b) **The Powell Direction Set Method, [Powell, 1964].**

The Downhill Simplex algorithm is a simple robust algorithm. The method has a geometrical naturalness about it which makes it readily understandable. However, it is not very efficient in terms of the number of function evaluations which it requires to locate a minimum. The method is less computationally efficient than the Powell Direction Set algorithm. Therefore, the Powell Direction Set Algorithm, was adopted as the primary relaxation algorithm for the camera model. The version of the Powell Direction Set algorithm used in the camera model owes much to later workers [Acton, 1970; Brent, 1973; Jacobs, 1977; Press et al, 1988].

If the derivatives were known, or could be calculated at reasonable cost, a faster relaxation scheme based on a **variable metric method**, for example, the Fletcher-Reeves-Polak-Ribiere algorithm [Polak, 1971; Press et al, 1988] could be used.

2.5.1 Selection of parameter space.

In the simplest case, the parameter space to be optimised consists of an **uncorrelated** set of differential rotation angles \([\delta \psi_{x1}, \delta \psi_{y1}; \delta \psi_{x2}, \delta \psi_{y2}]\) about the X and Y reference axes of the camera for each look respectively.

2.5.2 The [RRSKEW] cost function.

The relaxation algorithm finds a set of rotation angles, \([\delta \psi_{x1o}, \delta \psi_{y1o}; \delta \psi_{x2o}, \delta \psi_{y2o}]\) which minimise a unimodal cost function of the form:

\[
\text{cost}^2 = \frac{\sum_{i=1}^{i=N_g} |G_i^* - G_i| ^ 2}{N_g} \tag{2.22}
\]

Where:

- \(\text{cost}\) is the scalar cost,
- \(G_i^*\) is the \(i^{th}\) GCP,
$S_i$ is the corresponding $i^{th}$ space intersection.

$N_g$ is the number of GCPs.

The optimal parameter set, $[\delta \psi_{x1}, \delta \psi_{y1}, \delta \psi_{x2}, \delta \psi_{y2}]$ found by the Powell Direction Set algorithm are used to correct the pointing angles $\psi_{x1}$, $\psi_{y1}$, $\psi_{x2}$ and $\psi_{y2}$, which are derived from the SPOT-1 header in equations (2.13) and (2.14). This produces a modified transform between camera reference space and object space:

\[
R_{x1}' = R_{x1}(\psi_{x1} + \delta \psi_{x1}) \tag{2.23}
\]

\[
R_{y1}' = R_{y1}(\psi_{y1} + \delta \psi_{y1}) \tag{2.24}
\]

\[
R_{x2}' = R_{x2}(\psi_{x2} + \delta \psi_{x2}) \tag{2.25}
\]

\[
R_{y2}' = R_{y2}(\psi_{y2} + \delta \psi_{y2}) \tag{2.26}
\]

**2.5.3 Extending the optimised parameter space.**

In addition to rotational parameters described, an uncorrelated pair of [constant] position shift parameters for each orbit $\Delta \beta_1, \Delta \beta_2$ may also be introduced. The parameter set to be optimised in this case is: $[R_{x1}, R_{y1}, \Delta \beta_1, R_{x2}, R_{y2}, \Delta \beta_2]$.

**2.6 Extension of the Cost Function to Include Conjugate Data.**

The cost function for the minimisation process may be readily extended to include a relative component in addition to the absolute component described above. The cost function then becomes:

\[
C = W_{\text{abs}} \cdot C_{\text{abs}} + W_{\text{rel}} \cdot C_{\text{rel}} \tag{2.27}
\]

where:

$W_{\text{rel}}$ and $W_{\text{abs}}$ are the weighting factors for the relative and absolute contributions to the cost function.
$C_{abs}$, the absolute component of the cost function is defined in (2.22).

$C_{rel}$, the relative part of the cost function is defined:

$$C_{rel}^2 = \frac{\sum_{i=1}^{i=N_c} |\mathbf{m}_i|^2}{N_g}$$  \hspace{1cm} (2.28)

Where:

$|\mathbf{m}_i|$ is the ray-ray skewness of the $i^{th}$ pair of conjugate rays whose unit direction vectors are $\mathbf{r}_{1i}$, $\mathbf{r}_{2i}$, emergent from camera looks 1 and 2 respectively. In general, for $N$ looks, this function will be the mean of the sum of the squares of the magnitudes of all non-degenerate combinations of conjugate ray skewness vectors.

The conjugate points used by the cost function (2.22) may be obtained from the output of a stereo matching algorithm, for example the Otto-Chau stereo matcher [Otto and Chau, 1989]. The weighting factors are introduced into the cost function so that irrespective of the number of conjugate points or GCPs used in setting up the model, the total weights of the of the conjugate and GCP terms within the cost function is constant irrespective of the number of points in the conjugate and GCP datasets respectively. This avoids a combination of the linear geometry of the SPOT-1 camera, and large numbers of conjugate points giving rise to spurious orientations. This problem is further discussed in Chapter 3, which investigates the accuracy of the O’Neill-Dowman Camera Model.

It is clear that there is still work to be done investigating what constitutes an optimal parameter set, and cost function. The cost functions and parameter sets given in equations (2.22) and (2.28) work adequately, but they are almost certainly sub-optimal. Could a more accurate result be obtained using more sophisticated cost function and parameter set combinations? These considerations are addressed in Chapter 7, which discusses Future Research Directions.
2.6.1 Incorporation of the attitude data.

The model described above achieves an absolute RMS error of 12 to 15 metres, over a single 60km x 60km SPOT-1 stereo-pair, using 2 GCPs [See Appendix 1]. The associated RMS ray-ray skewness over a single SPOT-1 stereo pair, which is a measure of the accuracy of the relative orientation, is about 20 metres. Inclusion of the attitude data from the SPOT-1 header in conjunction with conjugate points, leads to a reduction in the error observed in the relative orientation, without adversely effecting the absolute error. The absolute model error is in fact made slightly worse [typically of the order of a metre], but this is more than offset by the improvement to the relative camera model [about 6 to 12 metres]. The reason for the increase in the absolute error may be errors in the measurement of the GCPs becoming more apparent when attitude data is incorporated into the model.

In order to incorporate the attitude data, it is splined in a similar manner to the position and velocity data. This permits the instantaneous line element rotation vector at time \( t \) to be expressed as a vector function of time:

\[
R_s(t) = \int f(\theta(t), \tau(t), \psi(t)) \, dt
\]  

(2.29)

Where:

- \( R_s(t) \) is the correction to the attitude matrix, \( R_0(t) \) at time \( t \).
- \( [\theta(t), \tau(t), \psi(t)] \) are corresponding ephemeris velocity components about the satellite reference axes, at time \( t \).
- \( \theta(t), \tau(t) \) and \( \psi(t) \), may be splined from the SPOT-1 header. The perturbation matrix \( R_s(t) \), hence the corrected attitude matrix, \( R_a(t) \) is computed by integrating the angular velocities over period \( \tau \) and then using the Euler formulation [Thompson, 1969], to form the perturbation matrix \( R_s(t) \):

\[
\tau = t - t_s
\]  

(2.30a)

\[
R_a(t) = \int_{t_s}^{t} \delta R_s(t) \, dt \cdot R_s(t)
\]  

(2.30b)
Where:

- $R_a(t)$ is the corrected camera attitude matrix.

- $\delta \hat{\mathbf{R}}_a(t)$ is an approximation to the angular velocity matrix over a small time step $\delta t$, with respect to the instantaneous camera reference axes.

- $t_s$ is the absolute time at which imaging of the scene commenced.

- $t$ is the time at which the line for which the ephemeris correction is being computed was acquired.

The object space ray direction vector, corrected with header attitude data thus becomes:

$$ \hat{\mathbf{r}}(l,s) = R_a(t) \cdot \hat{\mathbf{r}}_{\text{ref}}(s) $$

(2.31)

Where:

- $\hat{\mathbf{r}}(l,s)$ is the object space direction vector for the ray emergent from the pixel co-ordinate $[l,s]$ in the image plane.

- $\hat{\mathbf{r}}_{\text{ref}}(s)$ is the reference space direction vector for the ray emergent from pixel co-ordinate $[s]$ on the camera array.

2.7 Extension of the Model to Strips.

The camera model has been successfully extended to process strips of continuous imagery. There are two problems which had to be overcome in order to extend the model to continuous strips:

a) Transformation of the data supplied in the header files to a form which is suitable for processing a strip of imagery.

b) The location of tie points to register the images within the strip to each other.
2.7.1 Extraction of data from individual header data files.

Producing a pair of header files which can be used to set up a strip camera model, from a series of single model headers is relatively straightforward. Given that the strip consists of less than 60 contiguous image pairs, the satellite velocity and position information supplied in first image header may be used for the rest of the strip. Furthermore, the camera pointing angles, $\psi_{x1}$, $\psi_{y1}$, $\psi_{x2}$ and $\psi_{y2}$ supplied in the first image header may be used with the rest of the strip: by definition, a strip cannot be continuous if the attitude of the pointing mirror in the HRV instrument is changed, as this will cause a loss in the continuity of image acquisition.

The only data which changes significantly along a given strip of images therefore, is the camera attitude. If this is to be used in the orientation calculations, the attitude data from each individual image header within the strip must be catenated to produce an attitude dataset which is applicable to the strip.

2.7.2 Location of tie points to register images within the strip.

In an automatic camera modelling system, it is desirable for the ground transformation to proceed with as little operator intervention as possible: the need to locate tie points forms a potentially serious barrier to the automatic processing of strips of data, which are composed of a sets of continuous single images.

The problem arises because of the arbitrary way in which SPOT Image extract single scenes from the continuous swath of imagery transmitted from the SPOT-1 satellite. In general, the images do not abut. This means that the mean time between scene centre times is less than 9.024 seconds [which is the time between scene centres for the case of an ideal strip, in which all images are perfectly abutted]. In a real strip, a correction has to be made for the overlap of individual images forming the strip. Traditionally, this process has been accomplished by observing each of the images in the strip in turn, and locating a feature which is common in a pair of adjacent images. From a knowledge of the co-ordinates of this feature, it possible to compute the image overlap $\Delta l$, between the adjacent images $a$ and $b$.
\[
\Delta l_{ab} = 6000 - (l_a - l_b)
\] (2.32)

Where:

- \(l_a\) is the line position of a feature in image \(a\).
- \(l_b\) is the line position of the same feature in image \(b\): \(\tau_a < \tau_b\), where \(\tau_a\) is the scene centre time of image \(a\), and \(\tau_b\) is the scene centre time for image \(b\).

Because of operator error, in general \(N_{Ut}\) features will be observed in each of the images. In this case (2.32) becomes:

\[
\Delta l_{ab} = \sum_{i=1}^{N_{Ut}} \frac{6000 - (l_{ai} - l_{bi})}{N_{Ut}}
\] (2.33)

where:

- \(N_{Ut}\) is the number of tie points.

2.7.3 Automatic calculation of along-track overlap of images within strip.

It is possible to avoid having to adjust the strip manually. The scene centre time is available to the nearest millisecond for each individual scene in the strip. Furthermore, the nominal acquisition time for a single line of SPOT-1 imagery is known: [1.504ms PAN, 3.008ms XS]. Since the number of lines in a SPOT-1 single scene image is also known [6000 lines PAN; 3000 lines XS], it is possible to adjust the strip automatically:

\[
\Delta l_{ab} = \frac{(\tau_b - \tau_a) - t_{image}}{t_{line}}
\] (2.34)

Where:

- \(\Delta l_{ab}\) is the number of lines between the scene centres of two abutting scenes [6000 lines for two perfectly abutted PAN scenes; 3000 lines for two perfectly abutted XS scenes].
- \(\tau_a\) and \(\tau_b\) are the scene centre times of images \(a\) and \(b\) respectively.
\( t_{\text{image}} \) is the time between the scene centres of two perfectly abutted scenes [9.024 s for SPOT-1].

\( t_{\text{line}} \) is the time required to acquire one line of imagery, [SPOT-1: 1.504 ms SPOT-1 PAN, 3.008 ms SPOT-1 XS].

Experiments which have been conducted have show this automatic procedure to be perfectly adequate: a scene centre time accurate to the nearest millisecond permits the automatic strip adjustment scheme to find the overlap between abutting images to an accuracy of +/- 0.5 lines.

2.8 The Back Transform.

2.8.1 Basic transformation algorithm.

The calculation of the back transform is accomplished in the following manner: although the camera attitude matrix has been expressed as a function of the absolute time \( t \), it may equally well be expressed as a function of image line co-ordinate, \([l]\). The camera position in space, \( F(t) \), may also be expressed as a function of the line co-ordinate as implied by (2.2):

\[
R_a(l) = R_a(k \cdot t + c) \tag{2.35}
\]

\[
F(l) = F(k \cdot t + c) \tag{2.36}
\]

The shortest perpendicular vector, \( E \), between a point on the ground, \( \mathcal{G} \) and the ray emanating from the image co-ordinate \([l,s]\) may be expressed as a function of the image co-ordinate \([l,s]\):

\[
\hat{r}(l,s) = R_a(l) \cdot r_{\text{ref}}(s) \tag{2.37}
\]

Where:

\( \hat{r}(l,s) \) is the ray direction vector for the ray emergent from image co-ordinate \([l,s]\) in object space. \( r_{\text{ref}}(s) \) is the corresponding ray direction vector in camera reference space.
\[ \mathbf{u}(l, \mathcal{O}) = \mathcal{O} - \mathcal{F}(l) \]  

\[ \theta(l, s, \mathcal{G}) = \cos^{-1}(\mathbf{u}(l, \mathcal{O}) \cdot \mathbf{f}(l, s)) \]  

\[ E(l, s, \mathcal{G}) = |\mathbf{u}(l, \mathcal{O})| \cdot \sin(\theta(l, s, \mathcal{G})) \]

Where:

\( \mathbf{u}(l, \mathcal{O}) \) is a vector joining the perspective centre associated with image line \( l \) to the GCP \( \mathcal{O} \).

\( \mathbf{f}(l, s) \) is the unit direction vector of the ray emergent from pixel position \([l, s]\).

\( \theta(l, s, \mathcal{G}) \) is the angle between the the vectors \( \mathbf{u}(l, \mathcal{O}) \) and \( \mathbf{f}(l, s) \).

Thus, as shown in Figure 2.6, it is possible to express the vector \( E \) as a function of the image co-ordinate, \([l, s]\). The back transformation, of the object space vector \( \mathcal{G} \), may be found by minimising \( |E(l, s, \mathcal{G})| \) with respect to line and sample: the image point \([l', s']\) for which the error vector magnitude, \( |E(l, s, \mathcal{G})| \) is a global minimum of \( E(l, s, \mathcal{G}) \), is defined to be the back transform of the object space vector, \( \mathcal{G} \). This is the co-ordinate within the image space of a given look which corresponds to the point \( \mathcal{G} \) on the ground. The relaxation process is accomplished using the same Powell Direction Set technique used to establish the absolute camera model. The parameter which is to be optimised in this case consists of the image space co-ordinate, \([l, s]\); the cost function is the magnitude of the corresponding error vector \( |E(l, s, \mathcal{G})| \).

2.8.2 Increasing the computational efficiency of the algorithm.

The problem with the basic back transformation algorithm described above is that it not computationally efficient. Even using RISC technology [Sun-4/60 SPARCstation], the algorithm takes of order 2 seconds to transform a point between object space and image space. For a practical system therefore, this basic algorithm is clearly unacceptable. Stereo matching a typical SPOT-1 stereo image pair produces \( \sim 800,000 \) matched points. The
$P[l]$ is the position vector of line element 1

$r[l,s]$ is the ray emergent from sample $s$ of line 1

$u[l,G]$ is the vector joining the perspective centre of line element 1 to the object point. When $r[l,s]$ and $u[l,G]$ are parallel, the image point $[l,s]$ is the back transform of the object space point $G$.

Figure 2.6 Showing the Geometry of the Back Transform
current algorithm would require 18.5 days to generate the corresponding orthoimage. This assumes that the associated image re-mapping process takes a negligible time to execute!

The assumption implicit in the basic algorithm is that the camera attitude is changing on a time scale of the order of the line acquisition rate or faster. Investigation of integrated rate data, derived from the SPOT-1 header indicated that the camera attitude only changes significantly over a much larger time scale, of the order of that required to acquire 30-50 image lines. Furthermore, profiling has indicated that the bottlenecks in the basic algorithm are in the splining and matrix manipulation computations required to set up the camera attitude matrix. In the basic back transformation algorithm, the attitude matrices are re-computed for each point on the image plane. Given that the change in the attitude matrix is negligible over small blocks of image lines, it is clearly very inefficient to recalculate it for every image point within such a block. The computation of the vector parametric ray equation requires \( 4 \) matrix multiplications. Matrix multiplication of a pair of \( N \times N \) matrices requires \( 2N^3 - N^2 \) floating point operations. Multiplication of an \( N \) component vector by an \( N \times N \) matrix requires \( 2N^2 - N \) floating point operations. Computation of the attitude matrices only once per block therefore results in a vast decrease in the number of floating point operations in the back transform. For example, for a SPOT-1 PAN image, assuming a block-length of 5 lines, the number of potential floating point operations per block is reduced by 36000. Unfortunately, it is impossible to achieve the increased throughput of 4-5 orders of magnitude which is predicted by naive accounting! This is because certain floating point operations, for example the computation of the nominal look angles, \( \psi_x \) and \( \psi_y \), still have to be computed for each pixel co-ordinate within a block. Also the cost function must still be computed at each pixel position. This requires at least 10 floating point operations. Nevertheless, even taking these operations into account, far fewer floating point operations are required if blocking is employed. Taking all factors into account, blocking should result in a speed increase. The associated loss in camera model accuracy, expressed in terms of absolute RMS error, is insignificant \([< 1m]\). Thus the time required to produce the orthoimage for a typical SPOT-1 scene is reduced from the 18.5+ days required in the case of the naive back transformation algorithm to under 3 hours assuming a Sun
The beneficial effects of blocking on performance are not restricted to the object-space to image-space transform. Since the forward transform requires the same attitude matrices as the back-transform, its performance will also be substantially improved by blocking. Since the forward transform is non-iterative, the increase in computational throughput will be much closer to the 3-4 orders of magnitude naively predicted for the blocked back-transform. This will put the forward-transform into a performance envelope in which the computational bottlenecks are not caused by the algorithm itself, but by the efficiency of the supporting operating system, in particular its data input/output [I/O] efficiency.

2.8.4 Alternative methods for speeding up the camera model algorithm.

The method of blocking described above is one approach which may be used to substantially increase the performance of the O’Neill-Dowman Camera Model Algorithm. The chief objection to this sort of approach is the fact that the increase in performance is bought at the expense of reducing the overall accuracy of the model.6 There are several alternatives to the blocking approach which do not have this undesirable side effect. These include:

a) Writing code for time-critical operations in such a manner that the throughput is maximised.

b) Use of specialist hardware to perform time critical operations such as matrix-matrix multiplication, matrix-vector multiplication and the solution of linear systems of equations.

2.8.4.1 Code optimisation.
**Vs, el, Ps:** sensor reference axes.

**r:** Unit vector in direction of ray.

Pixel size 10m on ground (nominal).

swath cut by line element as it sweeps along track

block of lines for which attitude matrices are held constant. All rays emerging from this region have the same attitude.

Figure 2.7. Showing schematic of the attitude blocking concept used by the SPOT-1 sensor model
The principal objection with the first approach cited above is one of return. Although it is possible to increase the speed of critical matrix operations [at the loss of some generality] by careful coding, the performance increases which are achievable are small [< 10% increase in throughput] unless the critical routines are *hand-coded* in assembler.

Implementation optimisation of time-critical parts of the algorithm, in particular the *spling*, *matrix-multiplication* and *linear equation solution* may be considered to squeeze extra performance out of the camera modelling system. This may be desirable for compute intensive applications such as orthoimage production. Since the matrices which occur in spatial problems such as geometric camera modelling are generally small, the use of *hardwired methods* may be effective. For example, the solution of sets of linear equations may be solved by explicitly computing the inverse of the design matrix via its determinant, using expansion by row or column [Press et al, 1988]. This may be more efficient than a general method, for example, the Gauss-Jordan or LU decomposition methods [Press et al, 1988].

2.8.4.2 Use of specialist hardware.

The second approach which makes use of specialist hardware is not within the scope of the present work as this is essentially a *software study*. Semi-conductor manufactures such as Intel and Motorola produce a range of specialist VLSI devices such as digital signal processors [DSP’s]. Some of these devices may be used as a hardware vector and matrix arithmetic unit, operating as a *co-processor* within a computer system, in a similar manner to a conventional scalar floating point unit [FPU]. Development of a custom vector and matrix arithmetic unit is also a possibility using *application specific integrated circuit* [ASIC] technology. Since many time-critical operations within the camera model, and indeed within the DEM production system, may be efficiently vectorised, an alternative hardware solution is afforded by microprocessor devices incorporating *vector arithmetic units* such as the Intel INS80860 series [Hayes, 1989].
2.9 The Fast O'Neill-Dowman Camera Model.

Experimentation has shown that the space intersection $S^*$ generated by the rough attitude matrices for a pair of looks, $R_{o1}$, $R_{o2}$ is related to the corresponding ground control point, $G^*_i$ by a vector shift, $E^*$, to a good approximation. Thus, it is possible to set up a fast camera model which replaces numerical minimisation process, which is compute intensive, by a vector shift $E^*_ae^*$:

$$E^*_ae^* = \frac{\sum_{i=1}^{N_g} S^*_i - G^*_i}{N_g}$$  \hspace{1cm} (2.41)

Where:

$N_g$ is the number of GCPs.

$S^*_i$ is the $i^{th}$ space intersection vector, corresponding to the $i^{th}$ ground control vector. The space intersection is generated by the rough orientation matrix, derived from the SPOT-1 header. The rough attitude matrix may be corrected if required using the satellite attitude.

$G^*_i$ is the corresponding $i^{th}$ ground control vector.

The fast version of the camera model achieved results, as indicated in Appendix 1, which are for the case cited, almost as good as those achieved by relaxation. The fast version of the camera model may also be used to identify and remove rogue ground control and check points via the method of shift-pruning [rejection of erroneous ground control and check points], which is described in Chapter 4.

Endnotes to Chapter 2

1: A GCP is a point whose position is accurately determined on the ground and within N images which are used in computing a stereo model. The GCPs are used to orient a relative camera model, thus making it absolute.
2: A check point is a point whose position is accurately determined on the ground and within N images which are used to set up a stereo model. A set of check points is used to test the accuracy to which an absolute camera model has been computed.

3: This absolute time is in fact GMT according to CNES.

4: This was not found to be the case for the LSL/RAE implementation of the camera model described in chapter 4. Possible reasons for this discrepancy are discussed in chapters 4 and 5.

5: Profile is a UNIX tool which may be used to assess the computational efficiency of a program. The profiler gives sufficient information for a programmer to decide how the computational efficiency of the program may be increased.

6: This is pedantic. Tests have shown that blocking algorithm worsens the absolute RMS error by 0.25m-0.75m if the number of line in each block is small [between 5 and 15 lines per block].
Chapter 3
Testing and Results:
Initial Implementation

3.1 Introduction.

The preliminary assessment of the accuracy of the O’Neill-Dowman SPOT-1 camera model was accomplished using the SPOT-PEPS test dataset. This dataset consists of two single scene SPOT-1 stereo-pairs of the Marseille and Aix en Provence regions of the South of France. These datasets possess a number of geographical features which are advantageous when testing a camera model algorithm. These include:

(a) Coastal features: the Marseille SPOT-1 stereo pair contains a significant amount of coastline extending from just east of Marseille almost as far west as the Rhone delta.

(b) Both SPOT-1 stereo pairs contain a great deal of rugged, rapidly changing relief. Such features are a tough test of both the stereo matcher and the camera model.

In addition to the SPOT-PEPS test imagery, the model has also been tested using additional single scene SPOT-1 stereo pairs including:

(a) South Yorkshire [scene 29/242].

(b) The Isle of Wight [scene 30/248].

(c) Dorset [scene 28/247].

(d) Western Cyprus [scene 112/280].

In addition to the tests which have been conducted using single SPOT-1 stereo pairs, the initial implementation of the model has also been tested us-
ing two continuous strips of test imagery supplied under the aegis of the OEEPE. The two OEEPE strips each consist of 4 pairs of abutting 60km x 60km SPOT-1 stereo image pairs [49/259-49/262, 50/259-50/262], acquired over the South of France. A catalogue of control measured by IGN is provided for each. The two OEEPE test datasets are intended to be used as a standard in order to test the triangulation capabilities of competing SPOT-1 sensor models. To date, seven centres have participated in the OEEPE test scheme:

1) The Canadian Centre for Mapping, Ottawa, Canada.
2) Institut fur Photogrammetrie, Universitat Hannover, West Germany.
3) Institut Geographique National, France.
4) Politecnico Milano, Milano, Italy.
5) Department of Geographic Information, Queensland, Australia.
6) Department of Photogrammetry and Surveying, University College London, United Kingdom.
7) Laserscan Laboratories, Cambridge, United Kingdom.

In addition to analysing the raw RMS residual errors for single SPOT-1 stereo pairs and multiscene strips, tests were also conducted to assess the stability of the camera model with respect to the distribution of the GCP’s over the imagery. Further experiments were conducted in order to establish how camera model absolute and relative RMS errors and the dynamic relaxation profile of the optimisation process may vary as a function of the accuracy with which the ground control has been observed.

3.2 Testing the Absolute Accuracy of the O’Neill-Dowman SPOT-1 Camera Model.

3.2.1 A description of the methods used to assess absolute accuracy.
In order to assess absolute accuracy, a modified version of the camera model, adapted specifically for the purpose of quality assessment was developed. This modified model takes a small set of between 10 and 100 ground control/check points whose ground and image co-ordinates have been independently determined to a high level of confidence, and forms a series of models using all non degenerate combinations of 2 and 3 points from the GCP set. The remaining points are used as check points, in order assess the accuracy of the model formed. This prescription is repeated using successive sets of 2 or 3 ground points until all such combinations of points have been processed. This quality assessment (QA) model produces two outputs:

Firstly, for each model formed, the absolute check point RMS error (ARMSE) is computed using the RRSKEW cost function described in Section 2.5.2:

\[
ARMSE^2 = \frac{\sum_{i=1}^{N_{chk}} |S_i^* - G_i^*|^2}{N_{chk}}
\]

Where:

- \(N_{chk}\) is the number of check points.
- \(S_i^*\) is the \(i^{th}\) space intersection, which is the ground co-ordinate predicted by the camera model.
- \(G_i^*\) is the \(i^{th}\) corresponding check point ground co-ordinate.

Secondly, the residual vectors are computed for each check point in the model. These error vectors are given in a form which may be assessed graphically by using either the Sunview based Vec program [O’Neill, 1988b], or the Uniras [1989] based Vector_plot program [Lewis and O’Neill, 1989]. The residual vectors are defined by the following expression:

\[
E_i^* = S_i^* - G_i^*
\]

Where:

- \(S_i^*\) is the \(i^{th}\) space intersection co-ordinate generated by the camera model.
- \(G_i^*\) is the corresponding \(i^{th}\) check point ground co-ordinate.
\( \mathbf{E}_i \) is the \( i^{th} \) residual vector.

The \textit{absolute accuracy} tests were performed using both 2 and 3 GCP's for the Aix en Provence, South Yorkshire and Cyprus single scene SPOT-1 stereo-pairs. In addition, a number of further models were formed using selected ground control distributions for the other single scene stereo-pairs and for the two OEEPE strips. The results of this testing, consisting of vector plots of the individual absolute and relative check point error vectors and tables showing the associated absolute and relative RMS error statistics, are given in Appendix 1. In addition, a summary of these results is given in tables 3.1a, 3.1b, 3.1c and 3.1d.

### 3.2.2 A discussion of the accuracy of 2-GCP models.

Analysis of the results given in Appendix 1 shows that in the case of a two control point model, the accuracy of model formed is critically sensitive to the \textit{sample separation} of the GCP's. This is shown graphically in Figures 3.1, 3.2 and 3.3 in the case of South Yorkshire stereo pair and Figures 3.4, 3.5 and 3.6 in the case of Aix en Provence stereo pair. These Figures show the overall ARMSE associated with a set of camera models, sorted by line separation, sample separation and vector separation. The vector separation is defined to be the magnitude of the \textit{image co-ordinate difference vector} of the control points which were used to form the model:

\[
\mathbf{d}_l = \mathbf{l}_{\text{gcp1}} - \mathbf{l}_{\text{gcp2}}
\]

Where:

\( \mathbf{d}_l \) is the \textit{image difference vector}

\( \mathbf{l}_{\text{gcp1}} \) is the image position co-ordinate vector [line, sample] of the first GCP.

\( \mathbf{l}_{\text{gcp2}} \) is the image position co-ordinate vector [line, sample] of the second GCP.

The poor correlation between the line separation of the GCP's used to form a model and the resulting ARMSE implies that the accuracy of a camera model is not \textit{simply} correlated to the linear separation of the GCP's. Conversely, there is a high degree of correlation between the sample separation of the GCP's and the accuracy of the model formed. This implies that
Figure 3.1

Model accuracy: South Yorkshire [29/242]
Model accuracy: South Yorkshire [29/242]

Sample spacing

Model RMS residual
Model accuracy: South Yorkshire [29/242]

Vector [line and sample] spacing
Figure 3.4

Model accuracy: SPOT-PEPS dataset

Line spacing

Model RMS residual
Model accuracy: SPOT-PEPS dataset

Model RMS residual

Sample spacing
Model accuracy: SPOT-PEPS dataset
Table 3.1 Sample ARMSE statistics for initial implementation of model.

<table>
<thead>
<tr>
<th>GCP's</th>
<th>worst</th>
<th>average</th>
<th>best</th>
<th>no. of models</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>21.00</td>
<td>18.50</td>
<td>16.80</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>15.52</td>
<td>15.30</td>
<td>14.78</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>13.78</td>
<td>-</td>
<td>13.16</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCP's</td>
<td>worst</td>
<td>average</td>
<td>best</td>
<td>no. of models</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td>2</td>
<td>25.20</td>
<td>20.95</td>
<td>16.59</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>20.48</td>
<td>18.30</td>
<td>17.34</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>15.50</td>
<td>-</td>
<td>15.23</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCP's</td>
<td>worst</td>
<td>average</td>
<td>best</td>
<td>no. of models</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td>2</td>
<td>39.04</td>
<td>29.94</td>
<td>25.68</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>22.35</td>
<td>20.10</td>
<td>17.13</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>18.04</td>
<td>-</td>
<td>17.98</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCP's</td>
<td>worst</td>
<td>average</td>
<td>best</td>
<td>no. of models</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td>1</td>
<td>50.31</td>
<td>39.50</td>
<td>35.63</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>35.39</td>
<td>-</td>
<td>26.41</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>19.44</td>
<td>17.99</td>
<td>16.76</td>
<td>6</td>
</tr>
</tbody>
</table>
the accuracy of the camera model is highly correlated to the sample separation of the GCP's: models in which the control points have a wide separation in sample space are more likely to form a stable model possessing a low ARMSE. Conversely, models in which the control points are close together in sample space are more likely to be unstable and have a high ARMSE.

Looking at Figures 3.1-3.6 it is clear that some choices of GCP's form an adequate camera model despite the fact that they are not widely separated in sample space. It is the accuracy with which the ground control is observed which has the greatest effect on the accuracy of the model formed. The larger the observation error, the higher the corresponding ARMSE. The sample separation of the control points may be seen as a gain factor in the expression giving the ARMSE in terms of observation error:

\[ \text{ARMSE} = \frac{f(E_{\text{obs}})}{\delta s} \]  

Where:

- \( \delta s \) is the separation of the GCP's in sample space.
- \( E_{\text{obs}} \) is the observation error.

Nominally, an experienced operator can measure a GCP in SPOT-1 imagery to around 5 metres on the ground, and identify the corresponding features in the imagery to an accuracy of about 0.3 pixels [Peacegood, 1989]. As implied in equation (3.4) a set of ideally measured GCP's in which observation error is minimal will yield an accurate camera model even when these points are close together in sample space. The cause of these errors may be explained in the following manner: When a model is set up using two GCP's only two of the three rotational degrees of freedom within the model are fixed. Therefore, a poorly observed pair of GCP's will give rise to spurious rotations about this third unfixed axis.\(^1\) If the points are close together in sample space the spurious rotation will be about the roll axis of the satellite. The effect of rotation about the roll axis appears to be far more appreciable than the effects of rotation about the pitch and/or yaw axes of the satellite. Rotation about unfixed axes occur with poorly observed ground control points, which are not widely separated in sample space. It has been estimated independently [CNES, 1987] that the SPOT-1 platform is more sensitive to rotation about the roll axis than it is to rotations about the pitch.
and yaw axes. This may explain the anomaly which has been observed experimentally. Typically, if the ground control is measured by a competent operator, a sample space separation \( \geq 1500 \) pixels will be required in order to form an adequate model with an ARMSE \( \leq 15.0 \) metres.

In the case of two control point models, even if the control is well observed, there is a marked tendency for the models formed to be rotated compared to the ideal zero error model about an axis which is defined by the GCP pair. This is a systematic effect in which the size of error vector tends to increase with increasing distance from the axis defined by the ground control point pair. In the case of the single scene models, the magnitude of this systematic error is still within acceptable bounds, even at the periphery of the model. For models of extended strips, this is not the case; typical 2 control point ARMSE statistics for the two OEEPE strips were found to be at least 10 metres higher than the corresponding ARMSE statistics associated with single scene models extracted from the strips.

3.2.3 A discussion of the absolute accuracy of 3-GCP models.

Many of the problems described in the previous section can of course be overcome by using three rather than two GCP’s. If this is done, all the rotational degrees of freedom are fixed, with the result that the models which are formed become much more stable with respect to the distribution of ground control. For example, a much greater percentage of the models formed in the 3-GCP quality assessment using the Aix en Provence SPOT-1 stereo pair, have an ARMSE \( \leq 20 \) metres [81.47\%] than for the corresponding 2-GCP study [57.35\%]. Although the 3-GCP models are less sensitive to the distribution of the ground control and the accuracy with which it is measured, the ARMSE of these models appears to be more highly correlated to the linear separation of the ground control.\(^2\) The most stable 3-GCP control configuration was found to be the long triangle. In this configuration, there is a large separation \( \geq 2500 \) pixels in both the line and sample directions. Some typical long triangle configurations can be seen in the control configurations overlaid on the vector plots which are given in Appendix 1. Although the long triangle configuration can reduce the systematic error for single scene SPOT-1 stereo-pairs, it really comes into its own when dealing with continuous strips of data. In these cases, the effect of unfixed degrees of freedom, especially
Testing and Results: Initial Implementation

those perpendicular to the satellite track, are large. With the ground control configured as a long triangle, the O'Neill-Dowman camera model performs well when compared with the other camera models which were presented at the OEEPE workshop held at University College London in September 1989. A comparison of the O'Neill-Dowman model with the camera models presented by other organisations participating in the OEEPE workshop is given in Appendix 2. It is interesting to note that the two models developed at University College London, the O'Neill-Dowman and Gugan-Dowman models were the most consistent models [in terms of RMS statistics] to be tested using the OEEPE test strips. It is also noteworthy that the O'Neill-Dowman camera model is significantly less complex than many of the other camera models presented at the OEEPE workshop.

In practice the O'Neill-Dowman camera model will often give satisfactory results even using ground control of indifferent quality. For example, a model has been formed using a SPOT-1 single scene stereo pair of the Big Horn Basin in Wyoming, USA. In this case the ground control, extracted in part from 1:50000 maps of the area was of low accuracy. In addition, the maximum sample space separation of the GCP's was <1500 pixels, with the distribution of ground control skewed to one side of the imagery. Despite the poor control, a model was set up with an ARMSE of <34 metres.

3.3 Studies of the Dynamics of Camera Model Relaxation.

The effects of GCP observation error and distribution on the accuracy of the camera model have been further investigated by looking at the dynamic relaxation profile of the camera model. The dynamic relaxation profile is defined as that set of N ARMSE statistics which are associated with the N iterations of the Powell Direction Set required to locate a global minimum. Thus, in this test, the ARMSE associated with a model given by equation (3.1) is generated for each iteration of the Powell Direction Set. The form of the dynamic relaxation profile has been observed in the case of both 2-GCP and 3-GCP models of the Aix en Provence and South Yorkshire single scene SPOT-1 models. Two distinct forms of relaxation profile have been observed. The relaxation profile is observed as a 2 dimensional plot of ARMSE against Direction Set iteration number. Thus, the relaxation profile
charts the dynamic progress of the absolute orientation of the sensor model to a given set of ground control points. A description of the two forms of relaxation profile observed are given in sections 3.3.1 and 3.3.2.

3.3.1 The Ideal Relaxation Profile.

If the ground control used to orient the model is satisfactory, sample spacing ≥2500 pixels, GCP location error <0.3 pixels in image space and <5 metres on the ground, an ideal relaxation profile similar to that shown in Figure 3.7 is observed. The ARMSE for the model at each successive iteration of the direction set is a decreasing function of the iteration count: the gradient $\frac{\partial \text{ARMSE}}{\partial i}$ of the function is always negative. The residual error observed at the end of the relaxation process is due to observation error in the ground control and checkpoints and error in the satellite position, velocity and attitude datasets extracted from the SPOT-1 header. Although there was insufficient time for experimentation, it is likely that the greater part of the error is due to observation error in the ground control and checkpoint datasets. The effect of observation error on ARMSE is further discussed in Chapter 5 and Appendix 7 which present the results of quantitative studies of observation error for the SPOT-1 and ITIR [Intermediate Thermal Infra-red Radiometer] sensors respectively.

3.3.2 The Non-Ideal Relaxation Profile.

An example of the second type of relaxation profile which has been observed is shown in Figure 3.8. In this case, observation of the ground control is poor, [sample spacing of control points <2500 pixels and/or ground control observation error >0.5 pixels in image space and >10 metres in on the ground]. In the case of this type of profile, a phenomenon which we call over relaxation occurs. Over relaxation may be detected by the presence of extrema in the relaxation profile. A profile of this nature is associated with ground control points and/or checkpoints which contain significant observation error and/or have sample space co-ordinates which are too close together.
SPOT: Aix en Provence 51/262 near-ideal profile
SPOT: Aix en Provence 51/262 non-ideal profile

RMS residual vs. Iteration no.
3.3.3 The relative camera model.

In addition to looking at the absolute accuracy of the model, a study has also been made of the effect of using conjugate points and header attitude data in order to enhance the accuracy of the camera model. It was concluded that great care has to be exercised in the use of conjugate points.\(^3\) The ray-ray skewness statistic [RRMSE] which is a measure of the relative accuracy of the camera model is defined by the expression:

\[
RRMSE^2 = \frac{\sum_{i=1}^{N_{cng}} |m_i|^2}{N_{cng}}
\]  

(3.4)

Where:

\(N_{cng}\) is the number of conjugate points.

\(RRMSE\) is the root mean square ray-ray skewness for \(N_{cng}\) conjugate points which are well distributed over the model.

\(m_i\) is the ray-ray skewness vector for the \(i^{th}\) conjugate point.

The implication of the studies conducted to date is that attempting to extend the relaxation scheme using conjugate points and minimising their RRMSE is not likely to improve either the relative or absolute accuracy of the model. As a consequence of the weak, linear SPOT-1 geometry, the RRMSE function is multimodal and therefore possesses a large, potentially infinite number of minima. This gives rise to a large manifold of degenerate relaxation endpoints. These solutions within this manifold are indistinguishable from one another, and they all possess the property of reducing the RRMSE without improving the overall accuracy of the camera model. Therefore, any cost function which uses conjugate points must use additional information to break the degeneracy of the RRMSE function, which would effectively make it unimodal, and amenable to relaxation via the Powell Direction Set method.\(^4\) A simple cost function which has improved the relative model in a number of cases is given below:

\[
C = \sum_{i=1}^{N_{cng}} w_{aba} \cdot \text{skew}_{i,aba} + \sum_{j=1}^{N_{cng}} w_{rel} \cdot \text{skew}_{j,rel}
\]  

(3.5)
Testing and Results: Initial Implementation

Where:

- $C$ is the cost function. This is implicitly a function of the parameter set $\delta\psi_{x1}, \delta\psi_{y1}, \delta\rho_{x1}, \delta\psi_{x2}, \delta\psi_{y2}, \delta\rho_{x2}$.

- $w_{\text{abs}}$ and $w_{\text{rel}}$ are the weighting factors of the absolute and relative components of the cost function respectively. In order for the model to be stable $w_{\text{abs}}$ must be larger than $w_{\text{rel}}$.

- $\text{skew}_{i,\text{abs}}$ and $\text{skew}_{j,\text{rel}}$ are the magnitudes of the the $i^{th}$ absolute and the $j^{th}$ relatives [ray-ray skewness] residuals respectively.

- $N_{\text{gcp}}$ is the number of GCP's.

- $N_{\text{cnj}}$ is the number of conjugate points.

In this cost function, the multimodal nature of the ray-ray skewness term is offset by introducing an absolute ARMSE term, This limits instabilities observed when conjugate points are used alone, or the conjugate term is dominant.

The cost function given in equation (3.5) can produce disappointing results: the ARMSE attained using conjugate points for the South Yorkshire model [12.82 metres] is slightly worse than the result attained if only GCP's are used [12.81 metres]. The accuracy of the corresponding RRMSE also shows no improvement: [22.27 metres with conjugate points, 20.43 metres without conjugate points]. This may be attributed to the following factors:

a) The weighting factors used in (3.5); $w_{\text{rel}} = 0.6$, $w_{\text{abs}} = 0.4$ are likely to be non optimal.

b) The SPOT-1 camera is modelled as if the image plane is a rigid surface. In reality, the attitude of the sensor changes over the time period in which the image was acquired. It is likely therefore, that more satisfactory results may be obtained if the image plane were modelled as a non-rigid surface using an appropriate method, for example, Fourier series or Tshebyshev polynomials.
Testing and Results: Initial Implementation

c) Failing to use the header attitude data supplied in the header to correct the sensor attitude matrix function.

Modeling the sensor image plane as a non rigid sheet and/or correcting the sensor attitude using the header data would break the degeneracy of the solution manifold produced when conjugate points are introduced into the cost function. In effect, the wrinkling of the image planes may force the emergence of a global minimum in the RRMSE function, especially if the header attitude data is incorporated. This symmetry breaking gives rise to a noticeable increase in the accuracy of the relative orientation which is attainable using conjugate points. With as few as 15 well distributed conjugate points, the RRMSE models formed using the Aix en Provence SPOT-PEPS stereo-pair was reduced by 5.2 metres from 21 metres without attitude data or conjugate points, to 15.98 metres with attitude data and conjugate points. However, the ARMSE of the model was made slightly worse [to 13.5 metres from 12.89 metres]. A possible cause of the increase in the absolute error term is that the enhanced relative model may highlight deficiencies in the ground control.

Using the header attitude data alone also resulted in an enhanced relative model [RRMSE is improved to 17.25 metres from 21 metres], but the ARMSE is again degraded [to 13.51 metres from 12.89 metres]. The results of experimenting with conjugate points are presented in table 32. The method of incorporating the header data into the model has been presented in Chapter 2.

3.3.4 An Aside: The Camera Model as a Thermodynamic System.

Looking at a typical vector plot of a well distributed set of ray-ray skewness vectors [Figure 3.9 and 3.10] it is noticeable that these vectors still exhibit a great deal of correlation at the endpoint of the relaxation process. This implies that information about the satellite attitude present in the conjugate points has not been fully utilised. If this were the case, the residuals would be completely uncorrelated, indicating that the relaxation process had extracted all of the information about the sensor attitude from the conjugate point distribution: put in another way, the Shannon-Jaynes informational entropy [Shannon 1948; Jaynes 1980, Jaynes 1983], of the residual dataset is
a) Showing the effect of conjugate points on the accuracy of the South Yorkshire model using the cost function given in equation 3.1.

<table>
<thead>
<tr>
<th>ARMSE (m)</th>
<th>RRMSE (m)</th>
<th>GCP's</th>
<th>Number of Checkpoints</th>
<th>Conjugate points</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.81</td>
<td>20.43</td>
<td>2</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>12.82</td>
<td>22.27</td>
<td>2</td>
<td>16</td>
<td>100</td>
</tr>
</tbody>
</table>

b) Showing the effect of a combination of conjugate points and header attitude data on the accuracy of the Aix en Provence SPOT-PEPS model.

<table>
<thead>
<tr>
<th>ARMSE (m)</th>
<th>RRMSE (m)</th>
<th>GCP's</th>
<th>Number of Checkpoints</th>
<th>Conjugate points</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.89</td>
<td>21.00</td>
<td>2</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>13.50</td>
<td>15.98</td>
<td>2</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

c) Showing the effect of the header attitude data on the accuracy of the Aix en Provence SPOT-PEPS model.

<table>
<thead>
<tr>
<th>ARMSE (m)</th>
<th>RRMSE (m)</th>
<th>GCP's</th>
<th>Number of Checkpoints</th>
<th>Conjugate points</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.89</td>
<td>21.00</td>
<td>2</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>13.51</td>
<td>17.25</td>
<td>2</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.2 Effect on Accuracy of Conjugate Points and Header Attitude Data.
SPOT: Aix en Provence O’Neill-Dowman Camera Model
51/262 ray-ray skewness plot

northing (false origin: 117000)

metres

eastings (false origin: 812000)

key
○cpt
SPOT: Aix en Provence O’Neill-Dowman Camera Model
51/262 ray-ray skewness plot

Figure 3.10

key
O cpt

metres
35.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0

northing (false origin: 124000)
easting (false origin: 814000)
maximised in those cases for which the choice of cost function and parameter space allow the relaxation to extract all useful information about the sensor attitude from the conjugate point distribution. Conceptually, the relaxation process can be viewed as an information pump which transfers information [order] from the conjugate point dataset to the camera attitude matrices. As the relaxation process proceeds, the correlation of the ray-ray skewness [and absolute residual] vectors diminishes as the camera model position and attitude becomes increasingly refined. This thermodynamic isomorphism may be taken further: while the directional correlation of the residual vectors corresponds to an entropy term, the RMS sum of the vector magnitudes is clearly the corresponding energy term.

The isomorphisms which have been described in the previous sections may be equally well applied to the process of optimising the absolute camera model using ground control rather than conjugate points, or to mixed optimisation in which both ground control and conjugate points play a part.

3.4 Accuracy of the O’Neill-Dowman Camera Model Without Ground Control.

The accuracy of the O’Neill-Dowman model in the absence of ground control has also been investigated. The output of a test which compares the residual error vectors produced by a 2-GCP model and a 0-GCP model of Aix en Provence, when compared to a DEM derived from aerial photography of the region, is shown in Figure 3.11. Note the similarity of the two error-images. This indicates that to first order, the only difference between the 2-GCP and 0-GCP models is a shift of origin: non-linear effects such as shear and distortion are minimal. Thus, applications which only require slope data, for example, dynamic visualisation or hydrological studies, a computationally cheap zero GCP model is adequate. DEM’s may also be constructed for areas, for example Sopka Shiveluch Kamchatka in Eastern Siberia for which ground control is unavailable. This situation may arise either for political reasons, or simply because the area of interest is not amenable to ground survey. If superior header data is supplied in later SPOT missions, it may be feasible to devise a Fast SPOT Camera Model, which does not require ground control for those applications requiring only moderate accuracies.
figure 3.11. Showing a comparison of slope for DEM's which have been generated using a camera model with and without GCP's.

a) 3 GCP model.

b) zero GCP model.
3.4.1 Use of Zero GCP Camera Model to Detect Poorly Measured Ground Control.

The zero GCP camera model may be used to weed out poorly observed ground control or check points from datasets used to set up and test the camera model. Given the sensitivity of the O’Neill-Dowman model to ground control, failure to remove rogue ground control may result in either a very poor model being formed, if the rogue point is one of the GCP’s used to set up the model, or an anomalous test result if the rogue point is in the set of check points.

A technique, shift pruning, has been developed to remove rogue GCP’s. The shift vectors:

\[ S_{\text{shift}} = S_i^0 - G_i \]  

Where:
- \( S_{\text{shift}} \) is the \( i^{th} \) shift vector.
- \( S_i^0 \) is the \( i^{th} \) space intersection for the zero GCP model.
- \( G_i \) is the corresponding position in object space of the \( i^{th} \) GCP.

The mean shift vector is then computed for all \( N_{\text{gcp}} \) ground control and check points:

\[ \text{Shift}_{av} = \frac{\sum_{i=1}^{i=N_{\text{gcp}}} |S_{\text{shift}}|}{N_{\text{gcp}}} \]  

Where:
- \( \text{Shift}_{av} \) is the mean shift magnitude.

The variance of the set of shift vectors is then computed: any GCP’s whose shift vector magnitude differs from the mean magnitude by greater than 2 \( \sigma \) are discarded. Although this process is at present accomplished by hand, there is no reason why it should not be automated, enabling the camera model to automatically discard dubious ground control and/or test points prior to setting up a model.
The results of the shift pruning technique can be quite spectacular. As an example, we shall look at a set of ground control/check points which were derived manually from 1:10000 mapping, in order to set up a single scene SPOT-1 model of Cyprus. Prior to shift pruning, a typical O’Neill-Dowman camera model formed and checked using points from this dataset possessed an absolute RMS plan error of 80.68 metres, and an absolute RMS height error of 14.88 metres. Following shift pruning technique this residual error was reduced to 11.11 metres in plan, and 13.22 metres in height. A set of manually measured control and check points used to set up a single scene SPOT-1 camera model of Dorset also showed improvements; from 12.92 metres in plan and 23.26 metres in height before shift pruning, to 9.54 metres in plan and 14.50 metres in height respectively after shift pruning. The results of the shift pruning experiments are shown in table 3.3.

Endnotes to Chapter 3

1: Errors in the position and attitude data supplied in the SPOT-1 header may also contribute to these errors.

2: A further study has shown that the 2-GCP models are also correlated to the separation of the GCP’s in line space, but generally the random effect of observation error will tend to mask this correlation effect in all but the most accurately observed 2-GCP models.

3: A conjugate point is a point whose co-ordinates have been determined in both images in the stereo pair. Generally, conjugate data is obtained using an appropriate stereo matching algorithm, for example the Otto-Chau algorithm. In many cases, the introduction of conjugate points in the relaxation process actually makes the model worse! Similar effects were noted by Gugan [Gugan, 1988] when he tested his model.

4: A similar problem to this was encountered when designing the cascade algorithm which is a stochastic autoseder for the Otto-Chau stereo matcher. In this case the degeneracy of a multimodal manifold was broken by applying a unimodal minimiser coarse to fine. Perhaps a similar approach could be applied within the O’Neill-Dowman camera model.

5: This difference is not significant given the much larger sample considered when several hundred conjugate points are used.

6: This is essentially the same as imposing a confidence limit on the standard deviation $\sigma$ of the GCP/checkpoint dataset.
<table>
<thead>
<tr>
<th>Model</th>
<th>ARMSE Vector (m)</th>
<th>ARMSE Plan (m)</th>
<th>ARMSE Height (m)</th>
<th>GCP's</th>
<th>Check Points</th>
<th>Shift Pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyprus [112/280]</td>
<td>82.04</td>
<td>80.68</td>
<td>14.88</td>
<td>2</td>
<td>15</td>
<td>no</td>
</tr>
<tr>
<td>Cyprus [112/280]</td>
<td>17.27</td>
<td>11.11</td>
<td>13.22</td>
<td>2</td>
<td>15</td>
<td>yes</td>
</tr>
<tr>
<td>Dorset [281/247]</td>
<td>26.60</td>
<td>12.92</td>
<td>23.26</td>
<td>2</td>
<td>21</td>
<td>no</td>
</tr>
<tr>
<td>Dorset [281/247]</td>
<td>17.35</td>
<td>9.54</td>
<td>14.50</td>
<td>2</td>
<td>21</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 3.3 Showing the Effects of Shift Pruning on the Cyprus and Dorset Single Scene Models.
Chapter 4
O’Neill-Dowman Model:
LSL/RAE implementation

4.1 Introduction.

The major problem with the initial implementation of the O’Neill-Dowman camera model, which was developed under the aegis of the Alvey MMI-137 project, lies in the algorithmic inefficiencies and consequent poor throughput of its back transform. The function of the back transform is to take points in object space and to transform them to a camera position or look in image space. In a production environment, the back transform is required for the important practical task of generating orthoimages. An orthoimage is an image in which the effects of relief and sensor distortion have been removed. As stated in Chapter 2, the initial implementation of the camera model uses general but computationally inefficient algorithm based on the Powell Direction Set relaxation scheme.

In a production environment in which topographic maps and associated products are routinely produced, it is important to maximise the throughput of the back transform. The generation of an orthoimage from a complete single SPOT-1 stereo model, covering a ground area of 60Km x 60km typically requires of the order of 1,000,000 points to be back transformed from the SPOT-1 digital elevation model into the image space of one of the camera positions or looks. Typically, the vertical look, if available, is used since this image generally contains the least distortion due to earth curvature, and relief effects.

The implementation of the back transform in the initial version of the camera model requires about 1.5 seconds on a Sun 4/60 SPARC station to
transform a single point from object space to image space. The implication of this is that the production of a complete single-scene SPOT-1 orthoimage would require of the order of 17 days CPU time on a Sun SPARC station 4/60.

In order to rectify these faults, a prototype production camera model has been implemented, which makes use of tabulation and interpolation techniques to compute items such as the sensor position and attitude matrices which are repeatedly required in both the back and forward transformation processes. The principal design criteria for the LSL/RAE implementation of the O'Neill-Dowman camera model were that the forward transform be at least as fast as that of the initial implementation of the model, and that the throughput of the back transform be of the same order as the forward transform. On an unloaded Sun SPARC station 4/60, the initial implementation of the model requires about 60 CPU minutes to transform of the order of 1,000,000 conjugate points from image to ground space.

The back transform of the RAE/LSL implementation of the camera model, which uses a new algorithm, tabulates and interpolates those quantities which are expensive to compute. It was found to take approximately 138 minutes for a 30 metre SPOT-1 DEM, covering 60km x 60km, and containing in excess of 1,000,000 points to be transformed into the image space of a single camera position [look]. The extensive tabulation within the RAE/LSL implementation of the camera model has also led to an improved performance for the forward transform. On average, this has been found to be 15-20% faster than that of the initial implementation of the model. Thus, the major criterion specified for the LSL/RAE production prototype camera model has been satisfied.

4.2 Differences Between the Current and Initial Implementations of Model.

The algorithms which are used to initially form a new camera model, using the SPOT-1 header data, in conjunction with ground control data, remain unchanged from those described in Chapter 2. The principal changes to the model lie in the way that the camera orientation parameters are subse-
quently used to perform both the forward transform [in which a set of corresponding points in image space are transformed to ground space] and the back transform [which performs the reverse transformation, taking points in object space and transforming them into a corresponding set of points in one or more of the camera positions or looks in image space].

4.2.1 Summary of vector parametric ray equation generation in initial implementation.

In the initial implementation of the model, the sensor orientation matrix and position vector are re-computed at each pixel position in order to find the vector parametric ray equation of the emergent ray. This computation is described in detail in Section 4 of Chapter 2. In order to put the modifications made in the LSL/RAE implementation of the camera model into context, a brief description of this computation is given below.

The nominal camera pointing angles PSI_FIRST_X, PSI_LAST_X, PSI_FIRST_Y, PSI_LAST_Y are first used to establish the nominal direction of the ray in camera reference space using equations (2.10)-(2.12). This nominal ray direction is then corrected, if necessary, using the attitude parameters computed during the absolute orientation of the model to the ground control using equations (2.19)-(2.22). The ray is then transformed to object space using the sensor attitude matrix $R_0(t)$. The sensor attitude matrix is computed as a function of acquisition time, and hence of line position. The matrix $R_0(t)$ is computed by finding unit vectors in the directions of the sensor position vector ($\hat{p}(t)$), the pushbroom vector ($\hat{e}(t)$), and the velocity vector ($\hat{v}(t)$). The vectors $\hat{p}(t)$ and $\hat{v}(t)$ are found by interpolating the tabulated orbit position and satellite velocity using cubic spline interpolation, and then forming the appropriate unit vectors. $\hat{e}(t)$ is then formed by assuming that $\hat{p}(t)$ and $\hat{v}(t)$ are perpendicular, and form two of the orthogonal satellite reference axes. $\hat{e}(t)$ is then the vector product of these two vectors. This process for finding the approximate set of sensor reference axes closely follows that outlined in the Spot User Handbook, [CNES, 1987]. The derivation of these reference axes is described formally in equations (2.6)-(2.8). The derivation of an approximation to the sensor attitude matrix $R_0(t)$ from $\hat{p}(t)$, $\hat{e}(t)$ and $\hat{v}(t)$ is given in equation (2.9).
4.2.2 Ray equation generation in RAE/LSL implementation.

In the initial implementation of the camera model, the process described above is repeated N times for each set of N corresponding rays space-intersected in the case of the forward transform. In the case of the back transform, several [10-20], ray position vectors within the back transform look are computed as part of an iterative process which finds the image coordinate corresponding to a given ground position. This process is described formally by equations (2.35)-(2.40). The computation of the sensor attitude matrices and position vectors from scratch for each ray intersection is a very inefficient strategy which wastes a great deal of CPU time. In the case of the back transform, it leads to processing times which are of the order of weeks for single SPOT-1 orthoimages covering 60km x 60km of terrain.

In the RAE/LSL implementation of the camera model, care has been taken to ensure that quantities which have a large computational overhead, such as the sensor attitude and position matrices are computed only as often as necessary. In the case of the generation of the vector parametric ray equations, the computational throughput been considerably enhanced by using a scheme of tabulation and interpolation, which is a cheap computational strategy, wherever possible. The following items which are used in the generation of object space vector parametric ray equations are pre-tabulated in the RAE/LSL implementation of the camera model:

a) The unit direction vector of the ray in camera reference space.

b) The sensor attitude expressed as a matrix function of line position, hence acquisition time.

c) The sensor position expressed as a vector function of line position, hence acquisition time.

4.2.3 Determination of reference space ray direction unit vector.

In the RAE/LSL implementation of the camera model, the emergent ray direction is tabulated as a function of sample position in a reference co-
ordinate system whose origin is at the sensor perspective centre for a given line. Direct computation of the unit reference ray direction vector [equations (2.10)-(2.12)] is expensive. The principal reason for this is that the computation of the pair of rotation matrices is required to transform from the reference ray direction [This is the direction of a ray emergent from sample 3000 in the case of a PAN image, or from sample 1500 in the case of an XS image] to the reference space ray direction for a ray emergent from some sample $s$ [where $0 \leq s < 6000$ PAN; $0 \leq s < 3000$ XS]. Forming the rotation matrix is computationally expensive because of the difficulty of computing the trigonometric matrix elements within it. An account of the difficulties of implementing efficient strategies for the computation of trigonometric and other non analytical mathematical functions has been given by Fish [Fish, 1987], in his description of PML, the Portable Mathematical Library which is one of libraries supplied with the GNU C compiler and which has been used to implement the work described in this thesis.

In order to reduce the overhead of computing the unit ray direction vector in reference space, in the RAE/LSL implementation of the camera model, it is tabulated prior to any transformations taking place. Subsequently, the reference space unit direction vector for some arbitrary sample $s$, $\hat{r}(s)_{ref}$ may be computed from this table using linear interpolation.

Naively, since the nominal look angles $\psi_x$ and $\psi_y$ are generated from the linear expressions (2.10) and (2.11) one may expect to be able to compute the reference ray position, without any loss of accuracy, by linearly interpolating the unit ray direction vectors emergent from the first and last sample positions:

$$\hat{r}(s)_{ref} = \hat{r}_{last} \cdot \frac{s}{N_x} + \hat{r}_{first} \cdot \left( 1 - \frac{s}{N_x} \right)$$  \hspace{1cm} (4.1)

Where:

$s$ is the sample position.

$\hat{r}(s)_{ref}$ is the unit direction vector in camera reference space of the ray emergent from sample $s$, 

121
\( \hat{r}_{\text{first}} \) is the unit direction vector in camera reference space of the ray emergent the first pixel [sample 1],

\( \hat{r}_{\text{last}} \) is the unit direction vector in camera reference space for the ray emergent from the last pixel [sample 6000 PAN; sample 3000 XS],

\( N_s \) is the number of samples [6000 PAN; 3000 XS].

Unfortunately, because \( \hat{r}_{\text{first}} \) and \( \hat{r}_{\text{last}} \) are vectors rather scalar quantities, equation (4.1) is implicitly non-linear. Equation (4.1) correctly predicts the direction of the unit ray direction vector emergent from sample \( s \), but the magnitude is incorrect. The error in the magnitude will be at its most extreme towards the centre of the image [sample 3000 PAN; sample 1500 XS]. There are two ways in which this may be rectified:

a) Compute the reference space ray vector using equation (4.1). This vector may then be normalised, creating the required reference space unit direction vector. The problem with this approach is that the normalisation process will have to be repeated for every set of image points transformed, in the case of the forward transform, or, worse still, several times for each ground point transformed, in the case of the back transform. The renormalisation process requires the computation of a unit vector. This in turn requires the computation of a square root which, like the trigonometric functions requires a significant computational overhead to evaluate.

b) Tabulate the unit direction vector in reference space ensuring that there are sufficient entries in the table to reduce the interpolation error within acceptable bounds. As the sample step length \( \delta s \) is decreased below a critical value the inaccuracy in the vector interpolation process described above is reduced to an acceptable level. Empirically, a sample step length \( \delta s \) of 60 samples for a PAN model, or 30 samples for an XS model was found to yield an absolute RMS error of the same order of accuracy as that of the initial implementation of the model.
For a table of arbitrary length, (4.1) becomes:

\[ \hat{r}(s)_{ref} = \hat{r}_j_{ref} \cdot \frac{s \% \delta s}{\delta s} + \hat{r}_i_{ref} \cdot (1 - \frac{s \% \delta s}{\delta s}) \]  

(4.2)

Where:

- \( s \) is the sample position,
- \( \delta s \) is the sample step length,
- \( \hat{r}(s)_{ref} \) is the unit direction vector in reference space for the ray emergent from sample \( s \),
- \( \hat{r}_i_{ref} \) and \( \hat{r}_j_{ref} \) are the two entries in the table which are closest to the desired entry \([i < j]\),
- \( \% \) is the modulo operator.

4.2.4 Determination of sensor attitude in LSL/RAE implementation.

The computation of the sensor attitude matrix for each object space vector parametric ray equation required, is the single biggest factor contributing to the slowness of the back transform in the initial implementation of the O'Neill-Dowman Camera Model. In the RAE/LSL implementation of the model, the sensor attitude is tabulated as time variant, hence acquisition time variant matrix, prior to any transformation operations.

In the initial implementation of the model, the computation of an object space vector parametric ray equation proceeds via two attitude transformation matrices, the elements of which are functions of line acquisition time \( t \):

a) The rough attitude matrix \( R_g(t) \) which is computed from the orbit data read from the SPOT-1 header.

b) An implicit attitude matrix \( R_c \) which corrects to the header derived sensor orientation matrix \( R_c(t) \). This matrix maps the zero GCP camera model to a ground truth, which is defined by a set of user supplied ground control points. In the initial implementation of the camera model, this perturbation matrix was further subdivided into a pair of rotation matrices:
\[ R_s = R(\psi_x) \cdot R(\psi_y) \] (4.3)

Where:

- \( R_s \) is the sensor attitude correction matrix determined by the optimisation process.
- \( R(\psi_x) \) and \( R(\psi_y) \) are a pair of rotation matrices about the \( \hat{v}(t) \) and \( \hat{e}(t) \) axes in camera reference space respectively. The size of the corresponding rotations \( \psi_x \) and \( \psi_y \) are determined by the absolute orientation process. Note that each of the camera looks is relaxed separately.

In order to tabulate the sensor attitude function it is convenient to combine all transformations into one overall transformation matrix \( R_s(t) \):

\[ R_s(t) = R_s(t) \cdot R_s \] (4.4)

A numerical function, which describes the sensor attitude as a function of time \( t \), may be constructed by sampling the sensor attitude, matrix \( R_s(t) \), at \( N_{opt} \) times \( t_1, \ldots, t_i, \ldots, t_{N_{opt}} \) over the period of image acquisition. The optimum number of entries in the table, \( N_{opt} \), was determined empirically by forming a series of models in which the number of entries in the blocking table, \( N \), was iteratively adjusted until the absolute RMSE of the model formed was as good as the corresponding absolute RMSE produced by the initial implementation of the model.

For a single scene SPOT-1 stereo pairs, values of \( N \geq 10 \) produced satisfactory models in which the absolute RMSE is about 1-3 metres better than the corresponding absolute RMSE for the initial implementation of the model. Values of \( N < 10 \) were found to give rise to models in which the absolute RMSE was slightly worse than that of the corresponding absolute RMSE produced by the initial model: an \( N \) of 2 for example, gives rise to a model whose absolute RMSE is between 1.5 and 3.5 metres worse than the initial implementation of the model. The precise difference appears to be dependent on the ground control configuration used. In the case of a strip, the number of entries in the table must be increased by a factor \( M_s \), where \( M_s \) is the number of models in the strip:
\[ N_s = M_s \cdot N_{opt} \quad (4.5) \]

Where:

- \( M_s \) is the number of entries in the attitude function table for the strip.
- \( N_{opt} \) is the optimal number of attitude table entries for a single scene stereo model.

The overall attitude matrix for an arbitrary line position \( l \) is computed by a linear interpolation using the two attitude table elements \( i \) and \( j \) \( [i < j] \), which bracket the desired line position \( l \):

\[ i = \text{floor} \left( \frac{l}{\delta l} \right) \quad (4.6a) \]

\[ j = \text{floor} \left( \frac{l}{\delta l} \right) + 1 \quad (4.6b) \]

\[ a_{r,c,l} = a_{r,c,i} \cdot \frac{l \% \delta l}{\delta l} + a_{r,c,j} \cdot (1 - \frac{l \% \delta l}{\delta l}) \quad (4.6c) \]

Where:

- The operation \( \text{floor} (x) \) means find the biggest integer less than \( x \).
- \( \% \) is the modulo operator.
- \( l \) is the line.
- \( \delta l \) is the line step.
- \( a_{r,c,l} \) is the \( r,c^{th} \) element of the desired overall orientation matrix for line \( l \),
- \( a_{r,c,i} \) is the \( r,c^{th} \) element of the \( i^{th} \) entry in the attitude table,
- \( a_{r,c,j} \) is the \( r,c^{th} \) element of the \( j^{th} \) entry in the attitude table,
- \( r,c \) are the row and column indices respectively of the attitude matrix element \( a \).
4.2.5 Determination of the sensor position in RAE/LSL implementation.

The mechanism required to set up a numerical function from which the sensor position may be interpolated is almost identical to that outlined above for the sensor attitude. The only difference is that the components being interpolated are now those of a vector rather than a matrix:

\[
i = \text{floor} \left( \frac{l}{\delta l} \right)
\]

\[
j = \text{floor} \left( \frac{l}{\delta l} \right) + 1
\]

\[
a_{r,j} = a_{r,i} \cdot \frac{1\%\delta l}{\delta l} + a_{r,j} \cdot (1 - \frac{1\%\delta l}{\delta l})
\]

Where:

The operation \( \text{floor}(x) \) means find the biggest integer less than \( x \).

\% is the modulo operator.

\( l \) is the line.

\( \delta l \) is the line step.

\( a_{r,i} \) is the \( r^{th} \) component of the desired sensor position vector,

\( a_{r,j} \) is the \( r^{th} \) component of the \( i^{th} \) entry in the position vector table,

\( a_{r,j} \) is the \( r^{th} \) component of the \( j^{th} \) entry in the position vector table,

\( r \) is the row index of component \( a \) within position vector.

The initial tabulation of the satellite position is accomplished by splining the satellite position vector data read in from the SPOT-1 header in a similar manner to the initial implementation. In the LSL/RAE implementation of the SPOT-1 camera model, the sensor attitude and position blocking functions use the linear interpolation scheme described in equations (4.6) and (4.7) in
order to compute arbitrary sensor attitude matrices and position vectors from stored tables of values. Computationally, it is nearly as cheap to use a cubic spline interpolation scheme. In the case of the SPOT-1 satellite, the sensor attitude and position do not change rapidly enough to justify the use of cubic spline interpolation. If however, the underlying generic linear sensor model were to be used to model aerial push-brooms or similar platforms, which experience large changes in attitude and position, the use of a cubic spline rather than a linear interpolation scheme may lead to a more accurate determination of arbitrary sensor attitude matrices and position vectors from sets of tabulated values.

4.3 The Forward Transform.

The only change which has been made to the forward transform in the RAE/LSL implementation of the camera model is in the method by which the ray equations are obtained. This has been described in detail in section 4.2. Once the ray equations have been computed, the method of obtaining the space intersection is the same as that used for the initial implementation of the model, which is described in section 2.4.8 of Chapter 2.

4.4 The Back Transform.

4.4.1 Fast back transform based on Powell Direction Set method.

In order to attain a reasonable throughput under production conditions, the final implementation of the back transform for the LSL/RAE variant of the camera model differs appreciably from the method which was advocated for the initial implementation of the model. This initial implementation, which is described in section 2.8 of Chapter 2, used a Powell Direction Set relaxation scheme in order to find the optimum image space correspondence for a given ground point. The basis of the method is to use the Powell minimiser to search through [2 dimensional] image space until the ray which projects to the ground point with minimum residual error is found. The line and sample co-ordinates in image space of this ray are by definition the back
transform of the ground point.

The principal disadvantage of this method is that it can be slow. Although it is numerically very stable, the Direction Set method typically requires some 10-20 iterations in order to converge. In the initial implementation of the model, as a consequence of having to compute the vector parametric ray equation from scratch at each iteration, the resulting back transform was exceedingly slow: about 1-1.5 seconds of CPU time was required per point transformed on a Sun 4/60 SPARC station. The use of pre-tabulated numerical functions to generate the vector parametric ray equations reduces the CPU time required to transform 1 point using the Powell Direction Set method, to 0.001 seconds. This performance is quite respectable: a 30 metre SPOT-1 DEM could be transformed to image space in about 2.8 hours, and the generality offered by the 2-D Powell back transform algorithm is retained: this algorithm can be applied without loss of accuracy to quasi-linear camera geometries such as AVHRR, TIMS or ATM, and helical geometries such as Landsat TM, in addition to rigid linear geometries such as SPOT-1.

4.4.2 Fast back transform algorithm based on Brent method.

The fact that the geometry of the SPOT-1 sensor is rigidly linear may be used to reduce the dimensionality of the minimisation process thereby increasing throughput. The multidimensional Powell relaxation scheme may be replaced by a linear relaxation scheme of greater computational efficiency, for example, the Brent optimiser [Brent, 1973].

The reduction in the dimensionality of the problem is accomplished in the following manner: consider a vector \( T(l) \) which joins a ground point whose back transform is sought to the perspective centre \( p(l) \) associated with line \( l \) within a given look:

\[
T(l) = p(l) - \vec{C}
\]  

(4.8)

Where:

\( \vec{C} \) is the vector position of a ground point whose back transform is sought,
\( \vec{r}(l) \) is a vector joining the ground point position vector \( \vec{G} \) to the perspective centre position vector \( \vec{P}(l) \), of an arbitrary line, \( l \), in the selected image in which the back transform of the point \( \vec{G} \) is sought.

The geometry corresponding to the situation where the image line, \( l \) is arbitrary is shown in Figure 4.1. In Figure 4.2 the corresponding geometry is shown for the case where the line \( l \) is the image line which contains the image co-ordinate which is the back transform of the ground point \( \vec{G} \). In this case, because the SPOT-1 sensor is a rigid pushbroom, the vector \( \vec{r}(l) \) is constrained to lie in a plane in object space containing all emergent ray vectors from the perspective centre \( \vec{P}(l) \). Thus, for the special case where the line \( l \) contains the back transform of the ground point \( \vec{G} \):

\[
\vec{f}_{\text{first}}(l_b) \times \vec{f}_{\text{last}}(l_b) \cdot \vec{r}(l_b) - |\vec{r}(l_b)| = 0
\]

(4.9)

Where:

- \( \vec{f}_{\text{first}}, \vec{f}_{\text{last}} \) are unit vectors in the directions of the first [line 0] and last [line 6000 PAN; line 3000 XS] rays emergent from the SPOT-1 sensor in object space.

- \( \vec{r}(l_b) \) is the vector in the direction of the line which joins the ground point vector \( \vec{G} \) to the perspective centre vector \( \vec{P}(l_b) \).

- \( l_b \) is the image line which contains the image co-ordinate which is the back transform of the ground point \( \vec{G} \).

The Brent algorithm, like the Powell Direction Set Method, unimodally relaxes a scalar cost function. In the case of Brent’s method, this is a scalar cost function of a single independent variable. In the general case, where the line \( l \) does not contain the back transform of the ground point \( \vec{G} \), equation (4.9) will return a non-zero value which is a unimodal function of the line, \( l \). Equation (4.9) therefore fulfills the criteria required by a cost function to be used in conjunction with Brent’s method to find the line \( l \) within the image which contains the back transform of the ground point \( \vec{G} \). The Brent method locates this line by iteratively altering the line position \( l \) and computing the subsequent change in the cost function, given by (4.9), until it finds the line \( l \) such that the associated cost function is minimised.
Vs, el, Ps are the sensor reference axes

$I$ is a vector joining the perspective centre of an arbitrary line within the image to a object point. The line contains the back transform of this point if the vector $I$ is parallel to the object space ray plane.

Figure 4.1 Showing the geometry of the back transformation.
$v_s, e_l, p_s$ are the sensor reference axes

1 is a vector joining the point in object space to the perspective centre of the SPOT-1 line segment which contains the back transformed point.

**Figure 4.2** Showing back transform geometry when the line containing the image space point corresponding to the object space point has been found.
Having located the line \( l_b \) which contains the back transform of the ground point, the rigid linear nature of the SPOT-1 pushbroom permits the corresponding sample position, \( s_b \), to be found geometrically:

\[
 s_b = \frac{\theta_b}{\theta_{\text{first, last}}} \tag{4.10}
\]

Where:

- \( s_b \) is the sample co-ordinate of the back transformed point.
- \( \theta_{\text{first, last}} \) is the angle between the first and last rays in image space.
- \( \theta_b \) is defined below:

\[
 \theta_b = \cos^{-1}(\hat{r}_{\text{first}} \cdot \hat{l}_b) \tag{4.11}
\]

Where:

- \( \hat{r}_{\text{first}} \) is the unit vector in the direction of the first ray, emergent from the first sample.
- \( \hat{l}_b \) is a unit vector in the direction of the vector \( \hat{l}_b' \), which joins the perspective centre associated with line \( l_b \) to the ground point \( \vec{C} \).

4.4.3 A comparison of the throughput of the Powell and Brent relaxation schemes.

The reduction in the dimensionality of the minimisation process reduces the number of iterations required to locate the back transform of a ground point \( \vec{C} \). If the two dimensional Powell based algorithm requires \( N \) steps to find the back transform of a ground point, typically the Brent based scheme described above would find a solution in about \( \sqrt{N} \) iterations. Naively, one may expect Powell to be as efficient as Brent, if it is performing a one dimensional minimisation, rather than the two dimensional minimisation specified in the initial implementation of the camera model. This is not the case. The functionality of the Powell Direction Set method which enable it to perform multidimensional minimisations incur an additional computational over-
head, which makes it inherently slower than the less complex Brent scheme. Consequently the Brent method has been adopted as the back transform relaxation scheme in the LSL/RAE implementation of the O’Neill-Dowman camera model.

Empirically, the measured throughput of the back-transform algorithm using the Brent minimisation scheme was found to be 15-20% greater than that of the Powell Direction Set back-transform relaxation scheme. Thus, the Brent based back transformation algorithm implemented in the LSL/RAE variant of the camera model is capable of transforming a standard 1,000,000 point, 60km x 60km, 30 metre SPOT-1 DEM from ground space to image space in 2.31 hours on a Sun SPARC station 4/60: the initial two dimensional Powell variant of the algorithm using pre-tabulated sensor attitude matrices and position vectors requires 2.8 hours to transform the same standard dataset, and a one dimensional variant of the Powell relaxation scheme requires an intermediate time period of 2.65 hours on the same hardware. The difference between the fastest variant of the Powell based schemes, and the Brent based method (~30 minutes), is sufficient to justify the use of the Brent based algorithm for sensors with rigid linear geometries such as SPOT-1. However, the Powell based schemes are sufficiently fast to be used in those situations where the greater generality of multi-dimensional minimisation scheme is required: This will certainly be the case if the underlying generic linear sensor model is adapted to model quasi-linear sensor geometries such as AVHRR or ATM, or helical sensor geometries such as Landsat TM.

4.5 Automatic Detection of Poor Ground Control and/or Check Points.

When testing the initial Alvey MMI-137 implementation of the camera model a manual technique shift-pruning was used in order to remove rogue ground control and check points which, if retained, would give rise to a poor model. The requirement for a filter within the camera model which automatically rejects poor ground control and/or checkpoints was became apparent when the prototype UCL GEODEM system, was first used to generate digital elevation models from complete 60km x 60km SPOT-1 scenes, under the
aegis of the UCL-RAE LPO contracts. Later, a further contract placed by the Ordnance Survey, to produce a complete SPOT-1 DEM of Oman exposed further problems in the camera modelling component of the GEODEM system which would not have occurred had an automatic pruning algorithm been implemented.

The underlying problem in all of the cases cited above is the same. There has been an observation error in one or more of the ground control and/or checkpoints. This error manifests itself in the formation of a poor model if one or more of the ground control points input are in error or, an apparently poor model if one or more of the checkpoints input are in error.

In the RAE/LSL implementation of the model, the shift-pruning technique described in section 2.9 of Chapter 2 has been enhanced, and implemented as a selectable filter which pre-processes the ground control and checkpoint datasets. It automatically detects and rejects poorly observed ground control and/or check points by statistical evaluation.

4.5.1 Implementation of automated shift-pruning algorithm.

The automatic pruning algorithm is conceptually very simple. The mean magnitude of the vectors \( \mathbf{s}_i \) joining the zero GCP camera model space intersections for both the ground control and check points to the corresponding measured positions is computed using the following expression:

\[
\mathbf{s}_i = \mathbf{z}_i - \mathbf{m}_i 
\]  

(4.12)

Where:

\( \mathbf{s}_i \) is the \( i^{th} \) shift vector,

\( \mathbf{z}_i \) is the space intersection of the \( i^{th} \) GCP/checkpoint formed by the zero GCP camera model,

\( \mathbf{m}_i \) is the corresponding \( i^{th} \) measured GCP/checkpoint position in ground space.

\[
msvm = \frac{\sum_{i=1}^{i=N} |s_i|}{N_{gc}} 
\]  

(4.13)
Where:

\( msvm \) is the mean shift vector magnitude,

\( N_{gc} \) is the number of points in the GCP/checkpoint dataset to be pruned.

The mean shift vector magnitude may be computed and used to find the average deviation \( [avmd] \) of each individual vector in the dataset. The \( avmd \) is the difference between the magnitude of a given vector and the \( msvm \) for the GCP/checkpoint dataset:

\[
avmd_i = \text{abs} (msvm - |s|^i) \quad (4.14)
\]

Where:

The operation \( \text{abs} (x) \) means take the absolute value of \( x \).

\( avmd_i \) is the average deviation in magnitude for the \( i^{th} \) vector in the GCP/checkpoint dataset,

\( msvm \) is the mean vector magnitude as computed in equation (4.13).

Empirically, it has been discovered that the dominant error term [by an order of magnitude] in the zero GCP SPOT-1 camera model is translational. This implies that the average deviation in the shift vector magnitude for well observed ground control and checkpoints should be low. Experimentally, the ARMSE for well observed ground control and check point datasets combinations was found to be of the order of 6-12 metres. If a poorly observed point is introduced into such a dataset, its \( avmd \) will be large compared to that of the other points. Artificially induced errors of 25-50 metres in ground position and/or 2-5 pixels in image position both produced \( avmd \)'s which were well outside the 2 \( \sigma \) criterion which is used by the pruning algorithm to test whether a given ground control or checkpoint is statistically acceptable.

If many of the points in a ground control and check point dataset are in error, the pruning algorithm is no longer able to identify which of the ground control and check points within the dataset are in error. The successful detec-
tion of erroneous points is dependent on the number of such points in a
given dataset being low compared to the number of accurately measured
points. If many of the ground control and check points are in error the sta-
tistical distribution of \( \text{avmd} \)'s will be very broad. This will in turn give rise to
a large standard deviation, \( \sigma \). Empirically, it can be shown that values of \( \sigma >
10 \) metres are generally associated with distributions in which many of the
ground control and/or check points are poorly observed. In this situation the
pruning algorithm informs the user that the ground control and checkpoint
dataset is too poor to proceed with the formation of the model.

4.5.2 Examples of the effectiveness of shift-pruning.

A number of examples, which illustrate the effectiveness of the automat-
ic GCP/checkpoint pruning technique are given in table 4.1.

4.5.3 Limitations of the shift-pruning algorithm.

The automatic pruning algorithm which has been described is restricted
to camera geometries such as SPOT-1, in which the dominant error term in
the zero GCP camera model is translational. In theory, the technique could
be extended to other models in which errors due to effects such as rotation,
shear, and scaling are significant, but in these cases, the simple translational
mapping which relates the SPOT-1 zero GCP model to the measured ground
control points must be supplanted by a more sophisticated mapping, for ex-
ample an affine transformation, which is capable of absorbing the effects of
the additional transformations. The method also requires ground control and
check point datasets which contain a sufficient number of points to permit
the statistical methods described above to work. Empirical studies have
shown that the method is capable of detecting individual poorly observed
points when the density of such points does not exceed 1 erroneous point for
approximately every 4-5 accurately observed points. The implication of this
is, given a small dataset, containing for example 3 ground control points, one
of which is in error, the method would unable to detect which of the points
was in error, although the spread of the dataset, characterised by a large
value of \( \sigma \) would enable it to mark the dataset as poor.
4.6 Use of Alternative Cost Functions for Absolute Orientation.

In the initial implementation of the O’Neill-Dowman camera model, the cost function used computed the RMS error between a set of space intersections predicted by the current iteration of the Powell Direction Set relaxation scheme and the corresponding positions of an independently observed set of ground control points as described in section 2.5 of Chapter 2. The RAE/LSL implementation of the O’Neill-Dowman camera model allows the user to select either this space intersection cost function [RRSKEW], or an alternative cost function [RGCPD] which orients each look in a stereo model to the ground control independently. The primary motivation for introducing this new cost function was to improve the accuracy of zero-GCP models, which have to suffice when ground control is unavailable. The SPOT-1 header for each look contains an estimate of the plan position of the central and corner pixels of each scene imaged in the geographical co-ordinate system. This data is given to an accuracy of one second of arc. Theoretically, this means that a zero-GCP model with an ARMSPE of between 30 and 40 metres is possible without any ground control, if this data is utilised. Since the geographical positions of the corner and centre points are unique for each of the looks within a SPOT-1 stereo model, a cost function which is based on ray-ray intersection will be unable to utilise this data. Therefore, in the RAE/LSL implementation of the model, a new cost function was developed which relaxes the relative model by finding the shortest RMS perpendicular distances between a bundle of rays emergent from a set of ground control point image co-ordinates and their corresponding ground positions. This cost function possesses the desired property that it can orient each look within the stereo model independently of any other, and can therefore make use of the scene centre and corner point data supplied in the SPOT-1 header.

4.6.1 Description of the RRSKEW cost function.

The RRSKEW cost function is described by the following expression:

\[
\text{cost} = \left( \frac{1}{N_9} \right) \sum_{i=1}^{N_9} |S_i^* - G_i|^2
\]  

(4.15)
Where:

- \( \text{cost} \) is the scalar cost,
- \( G^i \) is the \( i^{th} \) GCP,
- \( S^i \) is the corresponding \( i^{th} \) space intersection.
- \( N_g \) is the number of GCPs.

\( \text{cost} \) is the cost function. This function is implicitly a function of the rotational parameters \( \delta\psi_x \) [about sensor axis \( \hat{v} \)], \( \delta\psi_y \) [about sensor axis \( \hat{e}l \)] and the orbit segment shift \( \Delta \rho_l \).

4.6.2 Description of the RGCPD cost function.

The RGCPD cost function is described by the following expression:

\[
\text{cost}_l = \frac{\sum_{i=1}^{N_{\text{gcp}}} |p_{i,x,l}|^2}{N_{\text{gcp}}} \tag{4.16}
\]

Where:

- \( \text{cost}_l \) is the contribution to the overall cost function of look \( l \).
- This function is implicitly a function of the rotational parameters \( \delta\psi_x \) [about sensor axis \( \hat{v} \)], \( \delta\psi_y \) [about sensor axis \( \hat{e}l \)] and the orbit segment shift \( \Delta \rho_l \).
- \( p_{i,x,l} \) is the shortest distance between the ray associated with the \( i^{th} \) ground control point, emergent from image co-ordinate \([l,s]\) and the position of the ground control point on the ground.
- \( N_{\text{gcp}} \) is the number of ground control points used to form the model.

Because each look is oriented independently, the RGCPD cost function is able to make use of the scene centre and corner points which are supplied in the SPOT-1 header. In order to do this, the co-ordinates obtained from the SPOT-1 header must be converted from the geographical co-ordinate system to the geocentric co-ordinate system. The scene centre and corner point data
Figure 4.3 Showing the Geometry of the RGCPD Cost Function
supplied in the SPOT-1 header give an estimate of plan position only. Therefore, in order to use this data in the formation of a camera model an estimate of the height of the scene corner and centre points must be made. In the present implementation of the model, these points are assumed to lie at the height which is predicted by the zero order camera model. Other schemes have also been tried, for example, assuming that the scene centre and corner points all lie at mean sea level, but all the other schemes tried to date have resulted in the formation of a less accurate camera model than the simple scheme which is described above.  

4.6.3 A comparison of the RRSKEW and RGCPD cost functions.

A comparison of the effectiveness of the RRSKEW and RGCPD cost functions is shown in table 4.2. This test was conducted without the inclusion of scene centre or corner points in the case of the RGCPD relaxation. As it can be seen, the RGCPD relaxation consistently forms better camera models than the original RRSKEW cost function, giving an improvement of the order of between 0.5 and 1.5 metres depending the ground control configuration used.  

4.6.4 Comparison of the ALVEY and LSL/RAE Implementations of the Model.

In table 4.3 a comparison is drawn between the accuracy of the original Alvey implementation of the O’Neill-Dowman camera model and the faster production implementation developed under the aegis of the RAE/LSL sub-contract. Due to the correction of a number of [minor] bugs in the initial implementation the overall accuracy of a typical model, formed using the new implementation has been improved by between 2 and 5 metres.

Endnotes for Chapter 4

1: The generation of an orthoimage is essential if accurate maps are to be generated from remotely sensed imagery.
2: It is in fact the I/O capability of the operating system which now limits throughput in the case of the forward transform.

3: In the case of these contracts, ground control was observed using inexperienced operators. As a result without pruning, the GCP/checkpoint dataset supplied gave rise to very poor sensor models.

4: The acronyms RRSKEW and RGCPD are: RRSKEW: ray-ray skewness (cost function). RGCPD: Ray - ground control point difference (cost function).

5: Latitude, longitude, height.

6: The ARMSPE is defined in equation (5.2).

7: Height data from maps could be used to estimate the height at the scene centre and corner points. However, if we are trying to develop a fully automated sensor model, this sort of approach is cheating!

8: Tests conducted under the aegis of the ITIR contract [Appendix 6] imply that the RGCPD solution is numerically less stable than the RRSKEW solution under certain conditions.
<table>
<thead>
<tr>
<th>Model</th>
<th>Pruned</th>
<th>ARMSE [metres]</th>
<th>ARMSPE [metres]</th>
<th>ARMSHE [metres]</th>
<th>GCPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dorset [28/247]</td>
<td>no</td>
<td>26.6</td>
<td>12.92</td>
<td>23.26</td>
<td>2</td>
</tr>
<tr>
<td>Dorset [28/247]</td>
<td>yes</td>
<td>17.35</td>
<td>9.54</td>
<td>14.50</td>
<td>2</td>
</tr>
<tr>
<td>Cyprus [112/280]</td>
<td>no</td>
<td>82.04</td>
<td>80.68</td>
<td>14.88</td>
<td>2</td>
</tr>
<tr>
<td>Cyprus [112/280]</td>
<td>yes</td>
<td>17.27</td>
<td>11.11</td>
<td>13.22</td>
<td>2</td>
</tr>
<tr>
<td>Oman [164/316]</td>
<td>no</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Oman [164/316]</td>
<td>yes</td>
<td>9.61</td>
<td>8.36</td>
<td>5.51</td>
<td>2</td>
</tr>
</tbody>
</table>

In the case of the unpruned Oman dataset, there was sufficient ambiguity in the supplied control to cause the pruner to abort processing.

Table 4.1 Showing the effect of automatic pruning on camera model accuracy

<table>
<thead>
<tr>
<th>Cost Function</th>
<th>GCP config</th>
<th>ARMSE [metres]</th>
<th>ARMSPE [metres]</th>
<th>ARMSHE [metres]</th>
<th>GCPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRSKEW</td>
<td>B.7</td>
<td>15.62</td>
<td>13.68</td>
<td>7.49</td>
<td>3</td>
</tr>
<tr>
<td>RRSKEW</td>
<td>B.9</td>
<td>15.51</td>
<td>13.98</td>
<td>6.71</td>
<td>3</td>
</tr>
<tr>
<td>RRSKEW</td>
<td>B.16</td>
<td>15.58</td>
<td>13.65</td>
<td>7.47</td>
<td>3</td>
</tr>
<tr>
<td>RGCPD</td>
<td>B.7</td>
<td>15.57</td>
<td>13.17</td>
<td>8.25</td>
<td>3</td>
</tr>
<tr>
<td>RGCPD</td>
<td>B.9</td>
<td>14.16</td>
<td>12.58</td>
<td>6.51</td>
<td>3</td>
</tr>
<tr>
<td>RGCPD</td>
<td>B.16</td>
<td>14.84</td>
<td>12.73</td>
<td>7.57</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4.2 Showing the effect of the cost function on the RMS statistics of the RAE/LSL implementation of the camera model

<table>
<thead>
<tr>
<th>Model</th>
<th>OEEPE strip</th>
<th>ARMSE [metres]</th>
<th>ARMSPE [metres]</th>
<th>ARMSHE [metres]</th>
<th>GCPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alvey</td>
<td>A</td>
<td>17.10</td>
<td>13.47</td>
<td>10.52</td>
<td>3</td>
</tr>
<tr>
<td>RAE/LSL</td>
<td>A</td>
<td>16.13</td>
<td>12.83</td>
<td>9.96</td>
<td>3</td>
</tr>
<tr>
<td>Alvey</td>
<td>B</td>
<td>17.36</td>
<td>15.46</td>
<td>7.90</td>
<td>3</td>
</tr>
<tr>
<td>RAE/LSL</td>
<td>B</td>
<td>15.52</td>
<td>13.98</td>
<td>6.72</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4.3 A comparison of ARMSE and associated statistics for Alvey and RAE/LSL implementations of the O'Neill-Dowman Camera Model
Chapter 5
LSL/RAE Implementation: Testing and Results

5.1 Introduction.

The production implementation of the O’Neill-Dowman camera model, which was developed under the aegis of the Laserscan Laboratories/RAE UCL subcontract, has been subjected to accuracy tests similar to those conducted on the initial variant of the model, described in Chapter 3. Because the current implementation of the model is intended to be the basis of a commercial system, an assessment of the throughput of the model has also been made. In a production environment, it is essential that the model is able to transform large quantities of data between image space and object space as rapidly as possible. The principal advances which have been made in the production implementation of the code are:

a) The provision of a back transform which is fast enough to produce SPOT-1 orthoimages in the order of hours rather than days.¹

b) The ability to deal with multispectral XS imagery.

Many of the tests which have been applied to the LSL/RAE production variant of the camera model code have been concerned with the testing of these new features. In addition, the speed and accuracy of orthoimage production has also been investigated as orthoimage production is an important practical application for a camera model in a topographic mapping environment.
5.2 Metrics for Testing Camera Model Performance.

The overall accuracy of the model was tested in a similar manner to the initial implementation: the camera modelling software was run for a number of SPOT-1 single stereo pairs and two [OEEPE] test strips each consisting of four abutting stereo pairs. The following statistics were computed for each of the models formed:

a) The Absolute Root Mean Square Error [ARMSE] statistic: this is a measure of the absolute accuracy of the camera model formed. The smaller the value of the ARMSE, the better the absolute accuracy of the camera model.\(^2\) The ARMSE statistic is subdivided into a plan error statistic, the Absolute Root Mean Square Plan Error or ARMSPE and a height error statistic the Absolute Root Mean Square Height Error or ARMSHE. The ARMSPE and ARMSHE are only meaningful measures of the camera model accuracy if the object space co-ordinates and their associated errors have been expressed in an appropriate local vertical co-ordinate system.\(^3\)

b) The Relative or skew Root Mean Square Error [RRMSE] statistic: this is a measure of the mean ray-ray skewness of the model, which is directly related to the accuracy of relative orientation of the camera model. The smaller the RRMSE, the better the accuracy of the relative orientation. The RRMSE statistic may be subdivided, like the ARMSE statistic into plan and height sub-components. These are the Relative Root Mean Square Plan Error or RRMSPE, and the Relative Root Mean Square Height Error, or RRMSHE respectively. The RRMSPE and RRMSHE statistics are only meaningful as measures of the accuracy of the relative orientation if the object space co-ordinates produced by the camera model are expressed in an appropriate local vertical co-ordinate system.

5.2.1 Definition of the ARMSE statistic.
The ARMSE statistic is defined by the expression:

\[
\text{ARMSE}^2 = \frac{\sum_{i=1}^{i=N_{chk}} |S_i^* - G_i^*|^2}{N_{chk}}
\]  

(5.1)

Where:

ARMSE is the Absolute Root Mean Square error for the model.

\( S_i^* \) is the \( i^{th} \) space intersection, which is the \( i^{th} \) ground position predicted by the camera model.

\( G_i^* \) is the corresponding \( i^{th} \) measured ground position.

\( N_{chk} \) is the total number of checkpoints considered.

5.2.2 Definition of the ARMSPE statistic.

The ARMSPE statistic is defined by the expression:

\[
\text{ARMSPE}^2 = \frac{\sum_{i=1}^{i=N_{chk}} |S_{ip}^* - G_{ip}^*|^2}{N_{chk}}
\]  

(5.2)

Where:

ARMSPE is the Absolute Root Mean Square plan error for the model.

\( S_{ip}^* \) is a vector containing the plan components of the \( i^{th} \) space intersection \( S_i^* \), expressed in an appropriate local vertical coordinate system.

\( G_{ip}^* \) is a vector containing the corresponding plan components of the \( i^{th} \) measured ground position \( G_i^* \), expressed in a suitable local vertical coordinate system.

\( N_{chk} \) is the total number of checkpoints considered.
5.2.3 Definition of the ARMSHE statistic.

The ARMSHE statistic is defined by the expression:

\[
ARMSHE^2 = \frac{\sum_{i=1}^{i=N_{chk}} |S_{ih} - G_{ih}|^2}{N_{chk}}
\]  

(5.3)

Where:

- ARMSPE is the Absolute Root Mean Square height error for the model.
- \(S_{ih}\) is the height component of the \(i^{th}\) space intersection, \(S_i\) expressed in an appropriate local vertical co-ordinate system.
- \(G_{ih}\) is the corresponding scalar height component of the \(i^{th}\) measured ground position \(G_i^h\), expressed in a suitable local vertical co-ordinate system.
- \(N_{chk}\) is the total number of checkpoints considered.

5.2.4 Definition of the RRMSE statistic.

The RRMSE statistic is defined by the expression:

\[
RRMSE^2 = \frac{\sum_{i=1}^{i=N_{chk}} |\vec{m}_i|^2}{N_{chk}}
\]  

(5.4)

Where:

- RRMSE is the Relative Root Mean Square error for the model: this is a measure of the accuracy of the relative orientation of the model.
- \(\vec{m}_i\) is the \(i^{th}\) ray-ray skewness vector.
- \(N_{chk}\) is the total number of checkpoints considered.
5.2.5 Definition of the RRMSPE statistic.

The RRMSPE statistic is defined by the expression:

\[
RRMSPE^2 = \frac{\sum_{i=1}^{N_{chk}} |m_{ip}|^2}{N_{chk}}
\] (5.5)

Where:

RRMSPE is the Relative Root Mean Square plan error for the model.

\(m_{ip}\) is a vector containing the plan components of the \(i^{th}\) ray-ray skewness vector \(m_i\), expressed in an appropriate local vertical co-ordinate system.

\(N_{chk}\) is the total number of checkpoints considered.

5.2.6 Definition of the RRMSHE statistic.

The RRMSHE statistic is defined by the expression:

\[
RRMSHE^2 = \frac{\sum_{i=1}^{N_{chk}} |m_{ih}|^2}{N_{chk}}
\] (5.6)

Where:

RRMSHE is the Relative Root Mean Square height error for the model.

\(m_{ih}\) is the height component of the \(i^{th}\) ray-ray skewness vector, expressed in an appropriate local vertical co-ordinate system.

\(N_{chk}\) is the total number of checkpoints considered.

It has already been stated that the plan and height subcomponents of the ARMSE and the RRMSE must be expressed in an appropriate local co-ordinate system. The reason for this is that if the terms plan error and height error are to be meaningful, a co-ordinate system in which the height error is perpendicular to the local geoid, and the plan error is parallel to the
local geoid must be selected. Local co-ordinate systems such as NG [National Grid] in the United Kingdom, LZ3 [Lambert Zone-3] in France, or UTM [Universal Transverse Mercator], which is used in many parts of the World at middle latitudes are examples of appropriate local vertical systems. The ARMSE and RRMSE statistics are derived from vector magnitude errors and are therefore not restricted to local vertical co-ordinate systems. These statistics are therefore meaningful measures of camera model accuracy in any co-ordinate system, including the geocentric co-ordinate system with which the camera model transformations are performed.

5.3 Review of Model Accuracy Testing.

The camera model accuracy testing was conducted principally using the OEEPE test data set. The RRSKEW cost function described in Section 2.5.2 was used in all the tests described in this chapter. The OEEPE test data set consists of two strips, each of which contain 4 abutting single scene SPOT-1 stereo images. These strips are both located in the Aix en Provence region of the South of France. They extend from the coastal district surrounding the Mediterranean port of Marseille to the foothills of the Alps. Additional supporting accuracy data was obtained by forming models using a number of single SPOT-1 PAN stereo scenes. These included South Yorkshire [OS 1:50000 sheet 111], The Isle of Wight and Solent [OS 1:50000 sheet 196], South Dorset, The Sultanate of Oman, and the Black Hills, Dakota. In addition, in order to test the ability of the model to handle multispectral XS imagery, a model was formed using XS stereo images of the Bighorn Basin in Wyoming.

5.3.1 Test of absolute error of model.

Due to the correction of coding errors made in the initial implementation of the model the test results attained using the LSL/RAE implementation of the model were superior to those attained using the initial implementation of the model. The difference in ARMSE between the two implementations is summarised in Table 5.1. These [strip] models were all formed using 3 ground control points arranged in the long triangle configuration described.
<table>
<thead>
<tr>
<th>software</th>
<th>model</th>
<th>GCPs</th>
<th>Checkpoints</th>
<th>ARMSE(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial</td>
<td>OEEPE A</td>
<td>3</td>
<td>104</td>
<td>17.1</td>
</tr>
<tr>
<td>LSL/RAE</td>
<td>OEEPE A</td>
<td>3</td>
<td>104</td>
<td>15.5</td>
</tr>
<tr>
<td>initial</td>
<td>OEEPE B</td>
<td>3</td>
<td>98</td>
<td>16.76</td>
</tr>
<tr>
<td>LSL/RAE</td>
<td>OEEPE B</td>
<td>3</td>
<td>98</td>
<td>16.1</td>
</tr>
</tbody>
</table>

Table 5.1. Showing the improvement in the vector ARMSE of forward transform for LSL/RAE implementation of software.
in Chapter 3. The ARMSE and consequently the ARMSPE and ARMSHE of single scene stereo models also showed significant improvement when formed using the LSL/RAE variant of the camera model. In particular, in the case of the PAN OMAN model a sub pixel ARMSE statistic of 9.8 metres was achieved with this variant of the model.5

Vector plots showing the absolute plan and height error at each of the check points used for model evaluation were produced for each of the models formed. Vector plots for each of the test models formed appear in Appendix 3 together with tabulations of the corresponding ARMSE, ARMSPE and ARMSHE statistics.

The individual vectors in the absolute vector plots shown in Appendix 3 tend to be scattered in random directions. This indicates that the relaxation process used to compute the absolute model has proceeded as far as it can. As we have already discussed in Chapter 3, in terms of information theory, the free energy of the dataset which is related to the sum of the magnitudes of the check point space intersection error vectors is at a global minimum. Further minimisation is not possible because the of the high entropy of the residual vectors which are all pointing in random directions.6

5.3.2 Stability of the ARMSE as a function of the GCP configuration.

When testing the initial implementation of the O’Neill-Dowman camera model, a series of experiments were conducted in order to establish the sensitivity of the ARMSE statistic to the distribution of ground control. These experiments implied that the model was generally unstable if the ground control configuration did not possess at least 1 pair of GCPs for which the sample space separation was greater than ~1500 pixels. This set of experiments also showed that given the simple orbit model and geoid used in the O’Neill-Dowman model, adequate models of continuous strips containing 3 or more scenes could not generally be formed using less than 3 ground control points. As a consequence of this, the long triangle configuration discussed in Chapter 3, was used for the formation of all subsequent strip test models.
The problem with the long triangle configuration is the practical inconvenience of acquiring the necessary ground control: Depending on the precise form of the long triangle, GCPs must be observed either at both ends of the strip as shown in Figure 5.1a, or at the start, middle and end of the strip as shown in Figure 5.1b. In many practical situations, ground control may be available only for a single model within the strip. In order to evaluate the performance of the camera model under these conditions, models were formed using OEEPE strip A in which the ground control was configured as a short triangle in which all the ground control points were contained within one scene, as indicated in Figure 5.2. As indicated in Appendix 3, all the strip models formed gave an ARMSE statistic of between 17 and 24 metres. An ARMSE of this order of magnitude implies that the observation of ground control in a single scene within a strip may yield models which are sufficiently accurate for the production of small scale [1:100000 or 1:200000] topographic maps.

5.3.3 Stability of the strip solution as a function of strip length.

The stability of the ARMSE and associated statistics as a function of strip length has also been investigated. In order to accomplish this, the ARMSE, ARMSPE, ARMSHE, RRMSSE, RRMSPE and RRMSHE statistics were computed for sub-strip models which contained the equivalent of 1, 2 and 3 single 60km x 60km abutting SPOT-1 scenes. In each case the ground control was configured as a long triangle of 3 GCPs. The results of these tests indicate that all the absolute model statistics remain approximately constant as more abutting scenes are added to the strip. A plots showing the variation of the ARMSE as a function of the number of scenes in strip A is shown in Figure 5.3.

5.4 Effect of GCP/Checkpoint Measurement Errors on Model ARMSE.

5.4.1 Effects of GCP observation error on model ARMSE.
Figure 5.1.a  Showing three ground control points arranged in a long triangle configuration over a four scene strip
Figure 5.1.b Showing alternative long triangle configuration for a four scene strip
Figure 5.2. Showing three ground control points arranged in a short triangle configuration in a four scene strip.
OOEPE strip A: Showing RMS error as a function of strip length
Object space RMS vector error

Figure 5.3
When the initial implementation of the O'Neill-Dowman camera model was tested, it was suspected that a large proportion of the observed ARMSE could be due to errors in the observation of GCP and check points, image space observations being particularly prone to error. In order to test this assertion, a special test version of the LSL/RAE implementation of the camera model was produced. This model was equipped with routines which introduce random error into a set of selected items of data which include the image and ground co-ordinates of the GCPs and check points. The random error is introduced into these items of data via a fuzzing process. This fuzzing process uses a random number generator to add a measure of uncertainty to selected items of data such as the image co-ordinates of a GCP in the manner indicated in equation (5.10):

\[ l_s' = \rho_u \cdot (\text{random} - 0.5) + l_s \]  

(5.7)

Where:

- \( l_s' \) is the line or sample co-ordinate subjected to random error.
- \( l_s \) is the observed line or sample co-ordinate.
- \( \rho_u \) is the diameter of maximum uncertainty which is permitted in the line and sample co-ordinates: Thus the maximum random error which is sometimes referred to as the error radius or the muffin tin radius is \( \frac{\rho_u}{2} \).

random is a random number generator returning a pseudo random deviate in the range \( 0.0 \leq \text{random} < 1.0 \).

The fuzzing function given in (5.10) produces a random deviate in the range \( -\frac{\rho_u}{2} \leq \text{random} < \frac{\rho_u}{2} \), thus the mean uncertainty associated with a given error radius \( \rho_u \) is \( \frac{\rho_u}{4} \), assuming that the random number generator produces a white noise distribution of error.

The following procedure was adopted to produce the plots shown in the figures below. In order to achieve some measure of statistical significance, between 5 and 10 fuzzed models were produced at each selected uncertainty
radius $\rho_{u\ GCP}$. The range of the variation in the uncertainty radius was chosen to extend over the 0.3-0.6 pixel zone in which image space observation error made by experienced photogrammetric operators is likely to lie [Peacegood, 1989]. In order to reduce the effects of noise in the ensuing analysis, synthetic zero ARMSE datasets derived from the GCP/checkpoint datasets of the OEEPE strips were used in order to analyse the effect of GCP and check point observation error on the accuracy of the model formed. These datasets were also used to investigate other aspects of sensor error which are described in the succeeding sections.

The synthetic test datasets were produced in the following manner: The ground control and check points for the two OEEPE strips was back transformed into image space using the best model parameters for each strip respectively. The back transformed image co-ordinate data was then combined with the original ground co-ordinate data to form an ideal ground control point/check point dataset in which the ARMSE and related error measures were minimal [ARMSE < 3.0 metres for strip A; ARMSE < 10.0 metres for strip B].

The effect of introducing random error into GCP image co-ordinates on the corresponding ARMSE statistic is shown in Figure 5.4 for OEEPE strip A. As one might expect, a line of linearly regressed best fit, which expresses the ARMSE as a linear function of the mean ground control point error $\frac{\rho_{u}}{2}$ increases as the error radius is increased. An experienced photogrammetric operator may be expected to locate ground control in PAN SPOT-1 imagery to an accuracy of between 0.3 and 0.6 pixels [Peacegood, 1989]. This level of error will introduce an extra 6 to 9 metres of error into the ARMSE statistic of a typical 4 scene strip.

5.4.2 Effects of error in check point measurement on ARMSE.

The procedure adopted for looking at the effects of check point observation error on the ARMSE statistic is identical to that adopted in order to assess the effect of GCP error with the exception that check points rather than ground control points are subjected to the fuzzing process prior to model formation. The resulting plots which show the observed ARMSE statistic as
OEEPE strip A: Effect of error in GCP [image] measurement
Object space RMS vector error

![Graph showing the relationship between mean GCP error and RMSE.](image)
a function of check point error radius $\rho_{chk}$ are shown in Figure 5.5 for OEEPE strip A. A line of best fit has been fitted to the data using \textit{linear least squares regression}. The result of this experiment indicate that the \textit{observed} ARMSE increases as a function of the observation error in the check points. In this case it is important to note the degraded ARMSE is not the \textit{true} model ARMSE: it is quite feasible to have a \textit{good} model which appears to be \textit{poor} because the check point dataset which is used to assess its accuracy is itself inaccurate. The positions of the check points in image space can be measured by an experienced photogrammetric operator to an accuracy of between 0.3 and 0.6 pixels [Peacegood, 1989]. The \textit{apparent} contribution to the ARMSE of a model, which uses a checkpoint dataset with observation errors of this magnitude will be of the order of 8 to 12 metres.

5.5 Assessment of the Effect of Sensor Attitude on Model ARMSE.

An assessment has also been made of the effect of error in the following items of telemetry data read from the SPOT-1 header:

a) Sensor position.

b) Sensor velocity.

c) Static sensor attitude.\textsuperscript{10}

d) Scene centre time.

5.5.1 Effect of error in sensor position on ARMSE.

Error may be introduced into the sensor position vector by using a pseudo random number generator to generate a random vector $v_{err}^i$:

$$v_{err}^i = \rho_{pos} \cdot (\text{random} - 0.5) \quad (5.8)$$

Where:

$v_{err}^i$ is the $i^{th}$ component of the random error vector.

159
stripa.fuzz.imchk2.pstats
OEEPE strip A: Effect of error in check pt [image] measurement
Object space RMS vector error

Figure 5.5

RMSE / metres

mean check pt error / metres
\( \rho_{\text{pos}} \) is the satellite position vector error radius.

\( \rho_{\text{pos}} \) is the muffin tin radius of uncertainty in the component of the satellite position vector \( \nu_{\text{err} i} \).

\textit{random} is a pseudo random number generator producing a white noise distribution of deviates in the range \( 0.0 \leq \text{random} < 1.0 \).

A set of random error vectors generated using (5.8) above may then be added to the 9 satellite position vectors obtained from the SPOT-1 headers corresponding to each sensor look to yield a pair of orbit position segments which are subject to a random error:

\[ \vec{p}'_i = \vec{p}_i + \nu_{\text{err} i} \]  \hspace{1cm} (5.9)

Where:

\( \vec{p}'_i \) is the \( i^{th} \) vector in the set of [9] vectors describing the sensor orbit position which have been subjected to random error.

\( \vec{p}_i \) is the \( i^{th} \) position vector in the orbit segment.

\( \nu_{\text{err} i} \) is the \( i^{th} \) randomly generated error vector.

A number of camera models were then generated with different error radii, \( \rho_{\text{pos}} \). In order to achieve some measure of statistical significance a minimum of 5 camera models were formed for each selected error radius \( \rho_{\text{pos}} \). Figure 5.6 shows the model ARMSE as a function of the error radius \( \rho_{\text{pos}} \) for OEEPE strip A.

An error in the sensor position of tens of metres is readily absorbed by the Powell relaxation process used by the absolute orientation. Errors in the sensor position only begin to have a noticeable effect on the ARMSE, when they are large enough to effect the accuracy with which the sensor reference axes are computed. This only occurs when the errors are of the order of hundreds of metres.
OEEPE strip A: Effect of error in sensor look positions
Object space RMS vector error
5.5.2 Effect of error in sensor velocity on ARMSE.

The methodology used for determining the effect of sensor velocity error on the model ARMSE is very similar to that described above. A series of random error vectors are created using the formula given in (5.10). These random error vectors are then added to the 9 velocity vectors read from each of the SPOT-1 headers to yield a pair of orbit velocity segments which are subject to random error:

\[ v_{err \ i} = \rho_{vel} \cdot (\text{random} - 0.5) \]  

(5.10)

Where:

- \( v_{err \ i} \) is the \( i^{th} \) component of the random error vector.
- \( \rho_{vel} \) is the satellite velocity vector uncertainty radius.
- \( \rho_{vel} \) is the error radius for uncertainty in the component of the velocity vector, \( v_{err \ i} \).
- \( \text{random} \) is a pseudo random number generator producing a white noise distribution of deviates in the range \( 0.0 \leq \text{random} < 1.0 \).

\[ \bar{v}_{i} = v_{i} + v_{err \ i} \]  

(5.11)

Where:

- \( v_{err \ i} \) is the \( i^{th} \) random error vector.
- \( v_{err \ i} \) is the \( i^{th} \) velocity vector obtained from the SPOT-1 header.
- \( v_{i} \) is the \( i^{th} \) velocity vector subject to random error.

In order to achieve a measure of statistical significance a minimum of 5-10 camera models were formed for each error radius considered. Figure 5.7 shows the model ARMSE as a function of the mean error in sensor velocity \( \rho_{vel} \). The model appears to be more sensitive to error in sensor velocity than it is to error in the sensor position. With mean error spanning a range of between 0 and 50 metres per second the observed ARMSE, and hence model error, rises rapidly as the error radius \( \rho_{vel} \) is increased. This is not surprising,
OEEPE strip A: Effect of error in sensor velocity
Object space RMS vector error

Figure 5.7

RMSE / metres
mean error in sensor velocity / metres per second
as an error of tens of metres introduces sufficient error into the sensor velocity, whose nominal magnitude is 6 kilometres per second to perturb the sensor reference axes computation, which assumes that the initial attitude of the pushbroom for some line $i$ is perpendicular to the velocity vector computed at line $i$. This assumption will clearly be in error if the error in the satellite velocity vector is large compared to its nominal magnitude.

5.5.3 Effect of error in static sensor attitude on ARMSE.

The effect of error in static sensor attitude on the model ARMSE was investigated by introducing random error into the sensor pointing angles $\text{PSI\_FIRST\_X}$, $\text{PSI\_LAST\_X}$, $\text{PSI\_FIRST\_Y}$ and $\text{PSI\_LAST\_Y}$ for each look which are obtained from the SPOT-1 headers:

$$\theta' = \rho_\theta \cdot (\text{random} - 0.5) + \theta$$

(5.12)

Where:

$\theta'$ is a static sensor pointing angle [PSI\_FIRST\_X, PSI\_LAST\_X, PSI\_FIRST\_Y or PSI\_LAST\_Y] subjected to random error.

$\theta$ is a static sensor pointing angle [PSI\_FIRST\_X, PSI\_LAST\_X, PSI\_FIRST\_Y and PSI\_LAST\_Y] obtained from the SPOT-1 header.

$\rho_\theta$ is the error radius for the sensor pointing angle uncertainty, $\theta$.

$\text{random}$ is a random number generator with a white noise distribution of random deviates over the range $0.0 \leq \text{random} < 1.0$.

The static sensor attitude, which is the nominal pointing direction for the sensor must not be confused with the satellite attitude, which gives the dynamical variation in the sensor pointing angles for the time period $\tau$ over which the image is acquired.

In order to achieve statistical significance a minimum of 5-10 camera models were formed for each error radius considered. Figure 5.8 shows the model ARMSE as a function of the log of the error radius of the static sen-
OEEPE strip A: Showing effect of varying satellite attitude
Object space RMS vector error

Figure 5.8

RMSE / metres

log mean error in attitude [\(rx, ry, rz\)] / radians
sor pointing angle $\rho_p$ for OEEPE strip A.\textsuperscript{11}

If the mean error in the pointing angle is less than 0.001 radians, the effect of error on the ARMSE is minimal. At the 0.001 radian limit, the ARMSE lies in the range 10-100 metres. Above the 0.001 radian limit the ARMSE increases exponentially; a mean error in the pointing angle of 0.01 radians yield an ARMSE in the range 400 to 900 metres.

5.5.4 Effect of error in the scene centre time on model RMSE.

The effect of error in the scene centre time was investigated by reading the scene centre time for each sensor look from the appropriate SPOT-1 header and then adding a random error term:

\begin{equation}
\text{set' = set} \cdot (\text{random} - 0.5) + \text{set}
\end{equation}

Where:

- $\text{set'}$ is the scene centre time subjected to random error.
- $\text{set}$ is the scene centre time obtained from the SPOT-1 header accurate to a millisecond.
- $\text{random}$ is a random number generator which generates a white noise distribution of random deviates in the range $0.0 \leq \text{random} < 1.0$.
- $\text{set}$ is the error radius for uncertainty in the scene centre time, $\text{set'}$.

In order to achieve statistical significance a minimum of 5-10 camera models were formed for each chosen error radius. Figure 5.9 show the model ARMSE as a function of the log of the scene centre time error radius $\rho_{\text{set}}$ time for OEEPE strip A.

The introduction of error into the scene centre time has very little effect on the ARMSE of the models formed: For a range of scene centre time errors between $1.0 \times 10^{-6}$ and 1.0 seconds the gradient of the line of best fit is nearly zero. This result is not surprising as error in the scene centre time is essentially a linear shift. The linear cost function used by the Powell relaxation scheme in order to orient the absolute model is therefore very efficient.
OEEPE strip A: Showing effect of varying scene centre time
Object space RMS vector error

Figure 5.9
at compensating for such effects.

5.6 Assessment of Accuracy and Speed of Orthoimage Production.

In addition to transforming stereo matched data from image co-ordinates to ground co-ordinates one of the most important functions of a camera model is the production of orthoimages. An orthoimage is an image in which the distortion due to the effects of sensor attitude, sensor geometry and terrain relief have been removed.

The test of orthoimage accuracy was conducted using the 50 metre MCE DEM of the Isle of Wight in conjunction with SPOT-1 imagery of the Isle of Wight and Solent area. These items were supplied under the aegis of the RAE-LPO contracts. The fast back-transform software was used in conjunction with the warping program praw [Day, 1988e] to produce an orthoimage from the right image of the stereo-pair [scene identifier S1H1870131110005]. Initially, it was hoped that ground control whose ground co-ordinates had been determined using the Global Positioning System [GPS] would be used to form the model. Unfortunately, the features chosen for GPS acquisition were not readily identifiable in the imagery. This meant that these GCPs could not be used to form a reliable camera model. Therefore, an existing GCP/checkpoint dataset of the area which had been used for the RAE-LPO work [ARMSE 22.5 metres] had to be substituted.

5.6.1 Test of accuracy and speed of back transform.

One of the principal features added in the LSL/RAE implementation of the camera model is an efficient back transform which is capable of high throughput in a production environment. This takes points in object space and finds the corresponding image co-ordinates in one or more of the sensor positions or looks. The accuracy of the back transform may be tested in a similar manner to the forward transform by computing an appropriate [absolute] RMS error statistic using the checkpoint dataset.
The method used to compute this RMS statistic is almost identical to that used to compute the object space ARMSE given in equation (5.1). The basis of the method is to measure the position of \( N_{c_kh} \) check points in image space. These image points may be measured using hardcopy images on an analytical plotter such as the Kern DSR1. Alternatively, image co-ordinates may be measured directly from the digital image data using either the stereo, disp, or points programs developed at UCL [Paramananda, 1988b], which run under Sunview on Sun-3 and Sun-4 workstations, or a commercial digital photogrammetric workstation implementation such as the \( r^2 S \) system.

5.6.2 Definition of the IARMSE statistic.

Formally the image absolute root mean square error [IARMSE] may be defined:

\[
IARMSE^2 = \frac{\sum_{i=1}^{i=N_{c_kh}} | p_{pred_i} |^2 - | p_{chk_i} |^2}{N_{c_kh}}
\]  

(5.14)

Where:

- IARMSE is the absolute root mean square error in image space.
- \( N_{c_kh} \) is the number of checkpoints in the test dataset.
- \( p_{pred_i} \) is the co-ordinate of the \( i^{th} \) check point in image space predicted by the model.
- \( p_{chk_i} \) is the observed position of the \( i^{th} \) check point in image space.

In addition to the IARMSE which is a scalar statistic measure of the overall back transform, the local variation of the back transform residuals may be investigated by plotting the local error vector, \([\delta \text{line}_{c_kh}, \delta \text{sample}_{c_kh}]\) at each check point co-ordinate within image space, \([\text{line}_{c_kh}, \text{sample}_{c_kh}]\).

Vector plots of the back transform vector residuals together with their corresponding IARMSE statistics have been produced for both the left and right looks of the two OEEPE test strips. In addition, back transform vector
residual plots and corresponding IARMSE statistics were also produced for the South Yorkshire, Isle of Wight, Dorset, Oman and Wyoming [XS] single scene models. These results are given in Appendix 3.

5.6.3 Use of differential IARMSE statistic as an indicator of model error.

During the course of testing the back transform it was noticed that there is a correlation between a high differential IARMSE and the formation of a poor camera model, with a correspondingly large forward transform ARMSE. The differential IARMSE, $\delta$IARMSE is defined by the expression:

$$\delta\text{IARMSE} = \text{IARMSE}_l - \text{IARMSE}_r,$$  \hspace{1cm} (5.15)\]

Where:

- $\delta\text{IARMSE}$ is the differential IARMSE,
- $\text{IARMSE}_l$ is the left image IARMSE,
- $\text{IARMSE}_r$ is the right image IARMSE.

Although models which are grossly in error [ARMSE > 50 metres] may be readily detected by the possession of a large IARMSE statistic for both the left and right images, the differential expression appears to be a good indicator of poor models whose ARMSE lies between 25 and 50 metres. In the case of such models either or both of the individual IARMSE statistics for the left and right images may be in themselves satisfactory [< 2 pixels], but such models will typically yield a $\delta\text{IARMSE} > 0.75$ pixels [for good models the $\delta\text{IARMSE}$ statistic is generally less than 0.5 pixels].

As well as being a potential metric of absolute model accuracy, statistics related to the differential IARMSE may also be used to predict appropriate weights for the space intersection. The present implementation of the model assumes that the point of space intersection for a ray-pair is located at the mid-point of their common skewness vector, $\tilde{m}_i$. This implicitly assumes an equal error weighting for each ray in the conjugate pair. Given a pair of IARMSE metrics $R_l$ and $R_r$, where $R_l > R_r$, a more realistic set of weights which could be used in a subsequent iteration of the absolute orientation process is given by the expression:
\[ W_t = 1.0 - \frac{R_t}{R_t + R_r} \]  
(5.16)

\[ W_r = 1.0 - \frac{R_r}{R_t + R_r} \]  
(5.17)

Where:
- \( R_t \) is the IARMSE for the logical left look.
- \( R_r \) is the IARMSE for the logical right look.
- \( W_t \) is the weight for the logical left look.
- \( W_r \) is the weight for the logical right look.

This idea is extensible to systems containing more than two looks.

5.6.4 Accuracy of the back transform.

The back transform error is an absolute error. It is not surprising therefore that the vector plots show the same high entropy random scatter which was observed in the case of the space intersection vector plots for the forward transform which are also absolute residuals. The back transform IARMSE was found to be of the same order as the corresponding forward transform ARMSE, given that the nominal SPOT-1 size for nadir viewing is 10 metres. Typically, good models formed using the OEEPE strip A yield ARMSE statistics of about 17 metres. The corresponding IARMSE, which is an average of the RMS statistics for the corresponding left and right looks in the stereo model is about 1.6 pixels corresponding to an equivalent nominal nadir ground error of \(~16\) metres. A similarly close correspondence exists between the averaged IARMSE of the back transform, and the ARMSE of the forward transform for the models formed using the OEEPE strip B. In this case, the ARMSE for good models is about 16 metres. The corresponding average IARMSE is about \(1.55\) pixels, which is equivalent to a nominal nadir ground error of \(15.5\) metres. Similar correspondences between the averaged back transform IARMSE statistic and the forward transform ARMSE statistic were noted in the case of each of the single scene stereo models that were formed.
5.6.5 Speed assessment of the fast back transform.

The speed of the back transform was assessed by using the UNIX profile command to determine the CPU time required to transform 1,000,000 points from object space to image space. The transformation of this number of object points is required in order to generate an orthoimage of a 60km x 60km SPOT-1 scene in conjunction a suitable warping function, for example the HIPS filter praw [Day, 1988e]. The time required to transform 1,000,000 points from image space to ground space was found to be approximately 120 minutes on a Sun 4/60 SPARC station [25Mhz SPARC RISC CPU + Weitek FPU + 8M RAM]. On a Sun-3 system [20 Mhz 68020 CISC CPU + 68881 FPU + 8M RAM] it takes 950+ minutes - which is about 8 to 9 times as long as for the SPARC station. Throughput on the Acorn R140 system [8Mhz ARM-2 RISC CPU + Weitek FPU + 4M RAM] was found to be 2.5 times as fast as a Sun-3: the time required to transform a 1,000,000 point test dataset from ground space to image space was found to be about 400 minutes.

The relative transformation speeds for these three processors, and in addition, an estimate of the throughput using the 30Mhz ARM-3 RISC processor are shown in table 5.2.

5.6.6 Assessment of orthoimage accuracy.

The checkpoints which had been used to compute the camera model ARMSE were located in the [10 metre] orthoimage. With a knowledge of the position of the bottom left hand [south west] corner of the orthoimage and the length of its sides in the National Grid system, the observations of the checkpoint co-ordinates in pixel co-ordinates within the orthoimage may be expressed in the National Grid [NG] co-ordinates:

\[ E_{ng} = E_{n wg} + \frac{s_{chk}}{s_{sl}} \cdot \delta E_{ng} \]  

(5.18)

\[ N_{ng} = N_{n wg} + \frac{l_{chk}}{l_{sl}} \cdot \delta N_{ng} \]  

(5.19)
<table>
<thead>
<tr>
<th>Processor (type and speed)</th>
<th>FPU</th>
<th>memory size (real) (virtual)</th>
<th>transform rate (pts per sec)</th>
<th>cost / MIP (pounds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPARC (25Mhz)</td>
<td>Weltek</td>
<td>16M 36M</td>
<td>112</td>
<td>533 749</td>
</tr>
<tr>
<td>ARM2 (8Mhz)</td>
<td>Weltek</td>
<td>4M 12.3M</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>ARM3 (30Mhz)</td>
<td>on chip FPU</td>
<td>8M 28M</td>
<td>105</td>
<td>192</td>
</tr>
<tr>
<td>MC68020 (20Mhz)</td>
<td>MC68861</td>
<td>8M 28M</td>
<td>14</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 5.2. Showing the point transformation rate for the back transform as a function of architecture. The cost per MIP for each architecture is quoted in pounds for each architecture at 1990 list prices.
Where:

\[ [x_{ckk}, y_{ckk}] \] are the co-ordinates of a check point in the orthoimage expressed in pixel co-ordinates [line, sample].

\[ [l_i, s_i] \] are the side lengths of the image expressed in pixel co-ordinates.

\[ [N_{ng}, E_{ng}] \] are the co-ordinates of a check-point expressed in the National Grid co-ordinates.

\[ [N_{swng}, E_{swng}] \] is the position of the SW corner of the orthoimage expressed in National Grid co-ordinates.

\[ [SN_{ng}, SE_{ng}] \] are the lengths of the sides of the orthoimage in the National Grid System.

The resulting National Grid co-ordinates for the checkpoints may then be compared with corresponding points located within the Ordnance Survey 1:50000 map sheet 196 [Solent and Isle of Wight]. The results of this comparison for the Isle of Wight orthoimage are shown in table 5.3. The results attained are very promising given that the camera model used to produce the orthoimage was mediocre. The orthoimage is shown in Figure 5.10. A visual quality assessment shows the orthoimage to be well registered to the MCE DEM. The apparently poor quality of the orthoimage is due to a loss of resolution which is unavoidable when it is printed on a single bit-plane hard copy device such as a monochrome laser printer.

5.6.7 Assessment of speed of orthoimage production.

The throughput of the orthoimage production process was measured by using the UNIX C shell builtin command time to compute the CPU time used in orthoimage production. Full command profiling could not be used in this case, because the orthoimage production software consists of a UNIX pipeline which is an aggregate of several command binaries, rather than a single command binary. Orthoimage production timings were taken using a 25Mhz Sun 4/60 SPARC station equipped with a Weitek FPU and 16 megabytes of RAM. The creation of the orthoimage of the Isle of Wight shown in Figure 5.10, requiring the transformation of some 96000 points from ground
<table>
<thead>
<tr>
<th>ft</th>
<th>E(NG)</th>
<th>N(NG)</th>
<th>Ht</th>
<th>dE(NG)</th>
<th>dN(NG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>461385.0</td>
<td>90705.0</td>
<td>42.0</td>
<td>5.3</td>
<td>-2.9</td>
</tr>
<tr>
<td>4</td>
<td>451167.0</td>
<td>3605.0</td>
<td>39.0</td>
<td>20.5</td>
<td>20.4</td>
</tr>
<tr>
<td>6</td>
<td>443530.0</td>
<td>90187.0</td>
<td>19.0</td>
<td>19.7</td>
<td>-28.7</td>
</tr>
<tr>
<td>8</td>
<td>455900.0</td>
<td>83445.0</td>
<td>54.0</td>
<td>5.6</td>
<td>6.20</td>
</tr>
<tr>
<td>9</td>
<td>438575.0</td>
<td>86372.0</td>
<td>63.0</td>
<td>4.0</td>
<td>4.4</td>
</tr>
<tr>
<td>10</td>
<td>452050.0</td>
<td>80620.0</td>
<td>46.0</td>
<td>-1.0</td>
<td>-1.2</td>
</tr>
<tr>
<td>11</td>
<td>445542.0</td>
<td>82192.0</td>
<td>67.0</td>
<td>10.5</td>
<td>7.5</td>
</tr>
<tr>
<td>27</td>
<td>434288.0</td>
<td>88407.0</td>
<td>16.0</td>
<td>-14.2</td>
<td>26.2</td>
</tr>
<tr>
<td>28</td>
<td>433564.0</td>
<td>88065.0</td>
<td>32.0</td>
<td>17.2</td>
<td>20.3</td>
</tr>
<tr>
<td>29</td>
<td>433639.0</td>
<td>86541.0</td>
<td>17.0</td>
<td>9.0</td>
<td>26.6</td>
</tr>
</tbody>
</table>

NB: all units are in metres

Table 5.3. Showing residuals at checkpoints for the Isle of Wight 10m orthoimage.
Figure 5.10. 10m PAN orthoimage synthesized from SPOT-1 scene SH1870131110005 using OS MCE DEM supplied by National Remote Sensing Centre.
co-ordinates to image co-ordinates took about 20 minutes of CPU time using this hardware configuration. This timing is likely to be very imprecise, as the time command gives the time used by the system on behalf of the user processes [the orthoimage pipeline], in addition to the time used by user processes themselves. With moderate system loading, the system overhead may be as much as 50% of the quoted CPU time.

5.6.8 Testing of relative error.

Relative accuracy statistics were also computed in the cases of all the camera models cited above. In the case of all the models formed, the RRMSE statistics attained were very similar to those attained using the initial implementation of the model, that is a RRMSE of between 8 and 15 metres. The RRMSE statistics attained in the case of the strip models tend to be larger than those attained from single scene stereo models. Vector plots showing the relative plan and height error at each of the points used to check the models formed are given in Appendix 3, together with the tabulation of corresponding RRMSE, RRMSPE and RRMSHE statistics.

The directions of the individual vectors in the relative accuracy plots tend to be highly correlated. This implies that there are imperfections in the relative orientation, which result in it being unable to find a good estimate of the global minimum RRMSE. The most probable explanation for this is that the simple parameter space which is used in the relaxation process does not consist of the optimum parameter set for a relative orientation, with the result that the cost function used in the relaxation process is not a unimodal function of the ray-ray skewness: the provision of an accurate relative camera model has proved to be a difficult goal to attain. Experiments which were performed with the original implementation of the camera model showed that inclusion of conjugate point data tends to increase the accuracy of the relative camera model - but at the expense of the absolute orientation. This effect is thought to be due to the weak geometry of the linear SPOT-1 sensor: the RAE/LSL implementation of the camera model does not show any notable advances over its predecessor in this respect.
The fact the residual vector directions are correlated at least at a local level, implies that there is an algorithm yet to be discovered, which is capable of forming a superior relative orientation. Two possible candidate schemes are outlined below. Both of these methods make use of conjugate data points which have been derived by stereo matching the associated imagery using the Otto-Chau stereo matcher [Otto and Chau, 1989].

a) An orbit post-processing scheme in which the locally correlated residual ray-ray skewness error is used to preprocess the orbit segment of one or more of the stereo looks in order to synthesise a model in which the ray-ray intersection error has been removed.

b) Precalculation of the relative attitude: in this case a set of features is identified in a reference image. An affine transformation is then used to correlate this set of features to the same set of features which has been observed in another image. The parameters of the affine transform may then be used to establish the relative orientation of the two images [O'Neill, 1991b]. This approach may be extended in the case of pushbroom imagery in order to compute the instantaneous attitude as a function of line position \( l \). This data may then be used to augment the attitude data derived from the SPOT-1 header.

In the case of both the schemes outlined above a composite camera model may be constructed which uses the Powell Direction Set algorithm to orient the model in association with additional orbit segment pre-processing or post-processing steps. In the case of method (a), the optimum form of composite model is likely to be an iterative scheme in which the absolute orientation by Powell Direction Set relaxation is followed by an orbit post-processing phase in which the reference orbit segment is adjusted to remove the residual skewness at the check point space intersections. This post-processed orbit segment is then used as an input orbit segment for a further relaxation phase which is used to re-orient the post processed model to the GCPs. The relaxation/post-processing cycle is iterated until self consistency is achieved in both the absolute and relative RMS statistics.
In the case of the relative attitude precalculation scheme, the relative attitude data is read in by the camera model and used to *pre-process* the sensor position and attitude prior to the absolute orientation phase.

Both of the schemes outlined above are capable of introducing *constrained plasticity* into the camera model. Alternative approaches which introduce additional terms related to skewness into the parameter space of the relaxation process results in an *unstable* relaxation in which *unconstrained plasticity* is introduced into the model which drives it towards false minima. As previously noted, the introduction of such unconstrained plasticity tends to result in a grossly distorted model, exhibiting a low RRMSE but a high ARMSE, which bears little resemblance to reality. The latter of the two methods outlined above, which is based on work due to Devereux [Devereux, 1989], is the more general. As well as providing a relative orientation for the SPOT-1 sensor, the method may be used to compute the relative orientation for other, quite unrelated sensors. In fact, the precalculation scheme may be thought of as a *virtual* relative camera model, which can provide both *homogeneous* relative orientations between two *looks* taken using sensors of the same type and *heterogeneous* relative orientations between a pair of *looks* taken by sensors of different type. The latter class of relative orientation was being researched under the aegis of the Leverhulme Trust project at UCL with a view to synthesising DEMs from imagery acquired using heterogeneous sensors.

### 5.7 Other Accuracy Tests Performed on the Model.

In addition to the major tests which have been described in the preceding sections, a number of other tests have been conducted using the RAE/LSL implementation of the O’Neill-Dowman Camera Model. A summary of these tests is given below:

#### 5.7.1 Formation of camera models with a constrained parameter space.

A number of camera models were formed in which an upper limit was set on the rotation parameters \([\delta \psi_{x1}, \delta \psi_{y1}, \delta \psi_{x2}, \delta \psi_{y2}]\) and translation parame-
ters $\{\Delta p_1, \Delta p_2\}$ in the parameter space of the optimisation algorithm. This was accomplished via a modified RRSKEW cost function which penalises large values being assigned to any of these parameters. The reason for this set of tests was to determine whether it was possible to attain a better ARMSE statistic for models in which the sensor geometry is constrained to be close to that of the physical sensor when the imagery was acquired. Models formed without any constraint on the relaxation space parameters tend to give rise to models in which the sensor attitude and position deviate from physically realistic values. Unfortunately, the ARMSE statistics of models formed using a constrained parameter space tended to be larger than the ARMSE statistics of the corresponding unconstrained models by a factor of between 2 and 6 metres. The most probable reason for this is absolute errors in the orbit segment derived from the SPOT-1 headers.

5.7.2 Effect of satellite dynamic attitude on model ARMSE and RRMSE.

In addition to supplying nominal sensor pointing data and orbit position and velocity data over the time period of image acquisition the SPOT-1 header also provides a satellite attitude dataset which describes the dynamical variation in the sensor pointing angles for the time period over which the corresponding image was acquired. In theory, the use of attitude data should lead to improvements in the accuracy of both the relative and absolute orientations. In practice the inclusion of satellite ephemeris data was found to worsen the model ARMSE by between 0.5 and 1.5 metres. The relative model, characterised by the RRMSE was also found to worsen by an amount varying between 0.5 and 2.5 metres. The cause of these effects is likely to be the absolute error in the orbit segment read from the SPOT-1 header which causes a mismatching of the attitude and orbit data recorded in the SPOT-1 header. In the initial implementation of the model, the use of attitude data was found to improve the relative orientation to the slight detriment of the absolute orientation. This effect may be explained by the presence of program bugs in the initial version of the camera model which were discovered and corrected during the implementation of the LSL/RAE variant of the model: the [few] tests which have been conducted to date tend to verify this state of affairs, as do the results reported by Gugan [Gugan, 1988] and other workers modelling the SPOT-1 sensor. However, a more exhaustive exami-
nation of the use of attitude data by the O'Neill-Dowman camera model is justified.

5.7.3 Use of shift pruning to reject poor ground control.

In section 4 of Chapter 3, it was suggested that poor GCPs may be removed by considering the average deviation of the magnitude of a set of shift vectors which connect the zero-GCP camera model space intersections [which is where the zero-GCP camera model predicts the GCP/checkpoints to be] to the corresponding GCP/checkpoints themselves. The nature of the SPOT-1 geometry is such that the transform between the zero-GCP space intersections and the corresponding GCP/checkpoints approximates well to a linear shift. This in turn implies that the average deviation of the GCP/checkpoint shift vector dataset should be low. If it is not, the inference is that something is wrong with the camera model.

The utility of this shift-pruning technique to detect rogue GCP/checkpoints was tested by artificially adding random error to GCP/checkpoints which previously formed a good model, and observing whether the erroneous points were detected via the technique. The results of the test were very encouraging; artificially introduced error points which lay outside a 2 σ distribution were readily detected and rejected.

The method also permits complete GCP/checkpoint datasets to be discarded as erroneous if σ for the dataset is above some user determined limit [Empirically σ values above about 12 metres were found to be associated with GCP/checkpoint datasets containing many erroneous points].

A detailed description of the automated pruning algorithm, together with a number of examples of its use, is given in Chapter 4.

Endnotes to Chapter 5.

1: This back transform works in conjunction with a suitable de-warping filter, for example the praw filter developed by Day [Day, 1988e] in order to produce the orthoimage.
We are assuming here that the checkpoints are accurate! Literally, a low ARMSE indicates a good fit between the checkpoint dataset and the corresponding space intersections predicted by the camera model. Of course, this checkpoint dataset may be in error.

Map projections such as UTM [Universal Transverse Mercator], NG [Ordnance Survey National Grid] and LZ3 [Lambert Zone 3] have been used in the current series of tests as local vertical systems.

Which one is selected depends on where in the World the test area is located.

In the case of this model the ground co-ordinates of the GCPs were located using GPS [Global Positioning System] and the corresponding image co-ordinates were observed by a trained photogrammetrist at the National Remote Sensing Centre using a GEMS system.

How far the minimisation process can proceed is related to the linearity of the cost function: A cost function which permits non-linear variation of the camera look angles as a function of the image line will give lower residuals as higher order attitude variations of the sensor platform can be modelled. The problem is that the derivation of such a cost function frequently requires more information about the dynamics of the sensor than is readily available.

This is because in the case of strip models the effect of drift in the computed orbit segments is greater than it is for the single scene stereo models because the orbit segments used in strip computations are longer giving errors a greater chance of accumulating.

Unfortunately, work performed under the aegis of the Leverhulme Trust Project at UCL has shown that these "general" methods only work in a satisfactory manner in situations where disparities are small and the motions of the sensor platform are predictable.

This is a typical conjugate point count when a pair 6000 x 6000 pixel SPOT-1 images are stereo matched, and hence is typical of the point count within a 60km x 60km SPOT-1 DEM. In order to form an orthoimage therefore, this number of points must be transformed from object space to image space.

The static sensor attitude, which gives the nominal pointing direction of the SPOT-1 sensor, is computed from SPOT-1 header fields PSI_FIRST_PIXEL_X, PSI_LAST_PIXEL_X, PSI_FIRST_PIXEL_Y, and PSI_LAST_PIXEL_Y. It should not be confused with the epheméris which gives the dynamical variation of the SPOT-1 sensor pointing direction over the time period in which a given SPOT-1 scene is imaged.

A log plot was required because of the rate at which ARMSE statistic increased in size as the error radius of the pointing angle is increased.

Yes - but it might if we constrained the skewness term in the cost function defined in (2.8) and (3.5) to be the projection of the skewness vector $\mathbf{m}_{l,l'}$, associated with line $l$ in look 0 and line $l'$ in look 1 in the direction of a vector $\mathbf{B}_{l,l'}$ which joins the two corresponding perspective centres $\mathbf{P}_l$ and $\mathbf{P}_{l'}$. 

Chapter 6
Overview of GEODEM Topographic Mapping System

6.1 Introduction.

A system has been developed at University College London arising out of the Alvey MMI-137 project and LSL/RAE follow on contracts which produces 2.5D topographic maps from SPOT-1 and other forms of satellite imagery, for example, AVHRR, KFA1000 and certain types of aerial photography. The material described in this chapter is the work of more than one person. In addition to the author, important contributions have been made by:


b) T. Day, H. Thomas: Implementation of interpolators based on Kriging and Delaunay triangulation respectively.

c) V. Paramananda: image display tools.


In this Chapter, the components of this topographic mapping system, and the role of the SPOT-1 sensor model within it are described. The digital elevation model [DEM] production process is discussed with particular reference to SPOT-1 imagery, but the techniques discussed are general and may be applied to other forms of imagery. Only the sensor modelling part of the DEM production process is dependent on the type of imagery. A flow chart which shows the processes required to derive a SPOT-1 DEM from homologous imagery and associated auxiliary data is shown in Figure 6.1. A proto-
Figure 6.1 SPOT-1 DEM production flowchart.
type system based on this flow chart has been used to produce 60km x 60km DEM's of complete SPOT-1 scenes of the British Isles [South Yorkshire, Hampshire & Isle of Wight], the South of France [Aix en Provence and Marseille] and also other parts of the world including Wyoming, North and South Dakota, Oman and Sierra Leone. The task of producing a DEM may be divided into four parts:

a) Input and decoding of the data [SPOT-1 header data, SPOT-1 stereo images].

b) Application of an automated stereo matcher to generate a digital disparity model [DDM]. The DDM consists of a large number of pairs of conjugate points.²

c) Setting up a model of the sensor used to acquire the imagery. The model is computed using a small set of ground control points. The ground accuracy of the model may then be tested with respect to a larger set of check points. Once the sensor model is formed to the desired degree of accuracy, it is used to transform the DDM, to produce an ungridded digital elevation model [DEM]. [The development of an accurate sensor model for the SPOT-1 satellite is the major theme of this thesis.]

d) Use of an interpolation algorithm to remap the ungridded DEM to a gridded digital elevation model [GDEM].

6.2 Spot Image Products.

The imagery is supplied by SPOT Image in a standard binary interleaved [BIL] format on 6400 BPI magnetic tapes. Each tape contains data pertaining to a single image which covers a ground area of 60km x 60km assuming nadir viewing. For off-nadir viewing, panoramic effects mean that the scene width is stretched in the cross-track direction up to 81 km. It is possible, by prior arrangement with SPOT Image to acquire strips of imagery which contain several 60km x 60km scenes. Assuming nadir viewing, a strip will cover an area with a swath width of 60km over a small segment
of the SPOT-1 satellite orbit. The SPOT-1 image product, supplied may be in either panchromatic, [PA] or multispectral, [XS] format. The image is extracted from the tape and transformed into a format suitable for processing using the HIPS library of image-processing tools [Landy and Cohen, 1982] using the all_spot filter developed at UCL.

In addition to specifying whether the imagery is to be supplied in single image or in strip form, the level of pre-processing which is to be done by SPOT Image prior to shipping must also be specified. SPOT Image provides imagery which is preprocessed to the following standards:

(a) Level 1A: The image is corrected for radiometric effects without any geometric pre-processing.

b) Level 1B: The image is corrected for radiometric effects. In addition, it is preprocessed geometrically to remove distortion due to earth rotation and earth curvature. The geometric preprocessing includes a resampling in the cross track direction in order to ensure that the pixel size in the corrected imagery is the nominal nadir pixel size of 10 m; because of the panoramic effect and earth curvature, the pixels become distorted: in the case of extreme off nadir viewing, [27°], the pixel width in the cross track direction is increased from 10m to 12.4m.

c) Level 2: Imagery is preprocessed to level 1B standard and then resampled into an appropriate local ground co-ordinate system, for example Lambert Conformal Conic, [LCC] or Universal Transverse Mercator, [UTM].

The complete range of SPOT Image products are described in volume 2 of The SPOT User's Handbook, [CNES, 1987].

For the purpose of accurate geometric mapping it is essential that the imagery used by the camera model is pre-processed as little as possible. Therefore level 1A imagery, is the SPOT Image product which is most suitable for topographic mapping purposes. Level 1B imagery may also be used as long as key information in the accompanying SPOT header, such as sen-
sor position and velocity, and sensor pointing data are modified to remove the changes which are introduced by level 1B preprocessing.

A typical SPOT-1 stereo image processed to the SPOT Image level 1A standard is shown in Figure 6.2 [Sopka Shivelush Kamchatka, Eastern Siberia]. Such images may be interactively displayed using disp, which is a general purpose HIPS based display tool which has been developed at UCL under the aegis of the Alvey MMI-137 project.

6.3 Generation of the Digital Disparity Model [DDM].

Once the image and header data are decoded, the next stage in producing a DEM is the generation of the digital disparity model [DDM]. This is accomplished by taking points in the left reference image and then locating the corresponding points in the right image of the SPOT-1 stereo-pair, using an automatic stereo matching technique.

6.3.1 Selection of stereo matching algorithm.

Although conceptually simple, designing an efficient algorithm capable of performing this task automatically has taken 25 man years of effort. Several stereo matching algorithms were targeted as being possible candidate algorithms by the Alvey MMI-137 project. These included:

a) The Barnard and Thompson algorithm [Barnard and Thompson, 1980].

b) The Pollard Mayhew Frisby Algorithm [PMF] [Pollard et al, 1985].

c) The Gruen adaptive least squares correlation algorithm [ALSC] [Gruen 1985; Gruen and Baltsavias, 1987; Gruen and Baltsavias, 1988].

d) The Practical Real Time Stereo Matcher [PRISM], [Nishahara, 1984].
Figure 6.2
Sopka Shivelush, Kamchatka, USSR
SPOT-1 satellite scene of volcanoes in the Soviet far east.
The scene was acquired on the morning of 2 November 1987 with illumination from the south-east.
Two such scenes, taken with different view angles on different dates, form a stereo-pair.
Data supplied to UCL by Dr David Pieri, Jet Propulsion Laboratory, as part of a
joint USA-USSR volcano monitoring project.
(C) 1987 CNES
6.3.1.1 The Barnard and Thompson algorithm.

This algorithm matches sparse networks of image features which are located using a suitable feature detector, for example the Moravec or Foerstner operators [Moravec, 1977; Foerstner et al., 1987]. Typically, this algorithm matches about 0.1% of all possible matches, making it a poor candidate for generating a dense DDM. However, application of the algorithm does not require any knowledge of the sensor; it can therefore be used to compute a 2D image relative orientation which can then be used to resample the imagery, enabling a second pass epipolar, or quasi-epipolar stereomatcher such as PMF or PRISM to generate a DDM of greater density. In addition to providing a mechanism for computing 2-D relative sensor orientations, the Barnard and Thompson algorithm may also be used to provide seeding data for unimodal area based stereo matching algorithms such as the Otto-Chau algorithm. The Barnard and Thompson algorithm is slow; a later algorithm, the cascade algorithm [Denos and O'Neill, 1991a, 1991b] which utilises the Gruen ALSC technique in conjunction with a pyramidal processing harness may be used to generate sparse DDM's for seedpoint or 2-D relative sensor orientation with greater computational efficiency.

6.3.1.2 The PMF algorithm.

This algorithm was developed under the aegis of the Alvey programme in the Department of Psychology at Sheffield University by Pollard et al. Like the Barnard and Thompson algorithm, PMF is a feature based technique, although its input consists of edge rather than discrete point features. These edge features are extracted using the Marr-Hildreth edge detector [Marr and Hildreth, 1980]. Stereo matching is achieved by finding the feature in the right hand image which possesses the minimum disparity gradient relative to the left hand image feature which is to be matched. In order to limit the number of edges in the candidate set within the right image, disparity window and disparity gradient limit constraints are imposed. Given that linear image segmentation is performed to sub-pixel acuity prior to matching, the PMF algorithm is typically capable of finding about 10% of possible matches.
stereo matches. Its chief weaknesses are its requirement for imagery that contains strong linear features and which is also epipolar. However, given that the Barnard and Thompson or cascade algorithms may be used to compute appropriate resampling parameters, PMF shows promise as a disparity domain boundary detector for urban and other forms of large scale imagery.

6.3.1.3 The Gruen adaptive least squares correlation algorithm.

The Gruen ALSC algorithm finds stereo conjugates using patch correlation. A square patch in the left hand [reference] image is correlated to a patch in the right hand image which has been distorted via an affine transformation. The transformation parameters are varied by iterative application of the least squares technique until the residual difference between the patches is minimised. The method is highly accurate with match accuracies of < 0.1 pixel have been quoted [Gruen, 1985]. It works particularly well with satellite acquired terrain imagery, and other forms of highly textured imagery. The chief weaknesses of the method are the computational cost, and the fact that the ALSC method is unimodal and therefore only capable of seeking local as opposed to global minima. The latter restriction has been overcome by Otto and Chau [Otto and Chau, 1989] who proposed a sheet growing mechanism in which a successful match is used to predict further matches in its local vicinity. The resulting algorithm is typically able to match in excess 90% of all potential conjugate points. If used within a coarse to fine harness such as the cheops or cascade algorithms [Denos and O’Neill, 1991a, 1991b], the stereo coverage of the algorithm may be further enhanced, especially in the case of large scale imagery containing discontinuities and occlusions.

6.3.1.4 The PRISM algorithm.

The Practical Real Time Stereo Matcher [PRISM] is a feature based stereo matcher. A $\Delta G$ operator is used to convolve the input imagery in such a manner that pixels within areas of the image with a positive grey level gradient change are assigned the value +1, and those within any region of the input imagery in which the grey level gradient change is negative are assigned the value -1. Pixels on zero-crossings [edges] are assigned the value
zero. The resulting image is a pattern of the values +1, -1 and 0. It is called a rep image by Nishahara. The left [reference] and right rep images are then patch-correlated. Unlike the Gruen ALSC algorithm only translation and scaling parameters are used in the correlation process. This implies that the images must be approximately epipolar if correlation is to be accomplished with confidence. The PRISM algorithm matches in a coarse to fine manner. Initial matches are made using rep images derived from averaged grey level images which have been reduced by a factor $2^m$, where $m$ is an arbitrary constant. Successful matches are then used to seed matching at finer resolution until, ultimately, the imagery is matched at the scale of the input imagery.

Nishahara's original specification for the PRISM algorithm is able to deal with pink noise in the input imagery, and is capable of working at video refresh rates. As a consequence of using patch-correlation, the algorithm is capable of generating a denser DDM than PMF. The primary weakness of the algorithm is that it requires quasi epipolar imagery. This means that in general imagery will have to be resampled before the PRISM algorithm can be applied.

6.3.1.5 The Ohta and Kanade algorithm.

The Ohta and Kanade algorithm [Ohta and Kanade, 1985] is a further example of a feature based stereo matching algorithm. Like PMF, the features used are edges, but unlike PMF, the algorithm is potentially capable of generating a dense DDM provided there are sufficient correlatable edge-features in the stereo-imagery. The Ohta and Kanade algorithm uses a dynamic programming technique to compute a piecewise linear mapping function $\eta_e$, which correlates pixels in the left image to the right image. The edgels are used as reference points when computing $\eta_e$. The algorithm requires input imagery to be epipolar, as $\eta$ is a linear function. The algorithm may be adapted as described by O'Neill and Denos, [O'Neill and Denos, 1991] to use grey level [terrain] imagery directly, provided that it is epipolar. The main weakness of the Ohta and Kanade approach is that like PMF and PRISM, it requires imagery to be epipolar.
6.3.2 Generation of a DDM using the Otto-Chau algorithm.

The algorithm which was eventually adopted by the Alvey MMI-137 consortium after exhaustive trials [Day and Muller, 1989a; Day and Muller 1989b; Day and Muller, 1988c; Otto, 1986] was the Otto-Chau algorithm. This algorithm combines the basic Gruen ALSC algorithm with a predictive sheet growing mechanism. This enables the Otto-Chau algorithm to overcome the unimodal limitations of the Gruen ALSC algorithm. Essentially, the sheet growing mechanism makes the approximation that the disparity gradient of the imagery which is being matched is gentle. For satellite imagery, this approximation holds. The algorithm has been used successfully to match SPOT-1, AVHRR and Russian KFA1000 imagery. At larger scale, as for example in the case of urban imagery, this disparity gradient limit approximation breaks down. Work by Denos [Denos, 1989] has shown that shadows, occlusions and discontinuities in large scale imagery cause the Otto-Chau algorithm to blunder significantly. Many of these problems may be overcome by the cheops or cascade algorithms which apply the Otto-Chau algorithm via a harness which automatically builds and then matches a coarse to fine pyramid of imagery. In addition, the constraint module of the underlying Gruen ALSC has been enhanced to cope with the type of blunders which are likely to lead to ALSC matcher failure at large scale. In particular, the following enhancements have been implemented:

- Improved thresholding for the parameters of the Gruen ALSC shaping matrix in order to stop the right hand image patch collapsing to a line or singularity in regions of high disparity gradient.

- The addition of a local epipolarity constraint: the directions of the predicted and refined disparity vectors for a match should be the same as, within any local area, all disparity vectors should be almost parallel. This constraint is particularly useful for removing the erroneous matches which occur on the boundaries of disparity domains.

- The use of edge cues: blundering at disparity domain boundaries within large scale imagery can be inhibited if the algorithm uses edge cues to prevent predicted matches which lie on disparity
domain boundaries being put into the *priority queue* of the sheet growing mechanism.

- The facility to use alternative minimisation schemes to adaptive least squares for patch correlation. In particular, the *Powell Direction Set* method has been used as an alternative to the Gruen ALSC technique, in order to investigate the effect of including higher order terms in the Taylor expansion of the matrix elements of the shaping matrix. The inclusion of higher order terms may enhance the performance of the algorithm in regions where the disparity gradient is rapidly changing.

The generation of a DDM using the Otto-Chau algorithm is conceptually simple:

a) Firstly, a small number of *corresponding seedpoints* [typically < 4] are located within the stereo image pair. This is accomplished digitally using the photogrammetric workstation programs, disp, points, or stereo which have been developed by Paramananda as part of the Alvey MMI-137 *image display toolkit* [Paramananda 1988a; Paramananda, 1988b; Paramananda, 1988c]. Alternatively, the seedpoints may be extracted automatically using the *autoseed* software developed initially by Chau [Chau, 1988c], and later extended under the aegis of Leverhulme Trust project [Allison et al, 1991]. Alternatively, the *cascade* algorithm may be used. A typical set of seedpoints generated by the *cascade* algorithm together with the stereo coverage which was generated using them is shown in Figure 6.3.

b) The images and the seedpoints are then fed into the Otto-Chau stereo matcher. A typical 60km x 60km SPOT-1 image will be matched [at a 6 pixel correlation grid] in approximately 900 CPU minutes on a Silicon Graphics 4G/330 system equipped with 64M of RAM. With a 16 node Parsys Supernode, an image of similar size is matched in approximately 240 cumulative CPU minutes, using a *geometrically* parallel form of the
Figure 6.3 Showing a typical set of seedpoints generated using the Cascade system using a small section of the SPOT-1 stereo coverage of Death Valley California.
Otto-Chau algorithm [Zemerly et al, 1991]. In each case, the product of the stereo matching is a dense DDM of corresponding image points.

In practice, the situation is often more complex than this. The Otto-Chau algorithm requires fine-tuning in order to match a SPOT-1 stereo-pair in a satisfactory fashion [no large holes in the sheet of stereo correspondences, and no blundering]. This fine tuning is accomplished via a large, non-orthogonal parameter set. Typical factors which effect the performance of the stereo matching process include:

a) The time delay between the acquisition of the images. In a cross-track stereo-system such as SPOT-1, the images in a stereo-pair may be acquired at different dates, often months apart. This means that the texture in the corresponding images of the stereo pair may be very different, making image correlation a formidable problem.

b) Cloud cover: If one or both the images are partially occluded by cloud, matching is not possible in the areas of the stereo-pair effected.

The coverage attained by the Otto-Chau algorithm when matching a typical SPOT-1 stereo pair is shown in Figure 6.4 [Death Valley, California USA].

A description of the practicalities of stereo matching using the Otto-Chau stereo matching algorithm is given by Day [Day, 1988d]. Dynamic display of stereo coverage during the matching process is possible using the HIPS tool Gruen_view which is part of the Alvey MMI-137 Image Display Toolkit, [Day, 1988a].

6.4 Generation of the DEM.

Once stereo matching is completed, the next stage in the process of DEM production is the generation of an ungridded digital elevation model or DEM. This requires an accurate camera model to be formed. This is used to
Figure 6.4 Showing typical coverage attained by the Otto-Chau stereo matcher. This stereo matched coverage was generated using the automatically generated seed points shown in figure 6.3.

a) Left image segment.

b) Right image segment.

c) Area not stereo matched [left image].

d) Area stereo matched [left image].
transform the 2D co-ordinates within the DDM to corresponding 3D ground co-ordinates within the DEM.

Two SPOT-1 sensor models have been used in the DEM production process:

a) The Gugan-Dowman Camera Model.

b) The O'Neil-Dowman Camera Model.

6.4.1 Data Required to set up the camera model.

In addition to imagery, standard level 1A or 1B SPOT Image tapes contain header and auxiliary header data. This header data is converted into a format which may be read by the camera model using the readcct program developed under the aegis of the Alvey MMI-137 project by Laserscan Laboratories. Unfortunately, the readcct program is hardware-dependent and can only be executed on the VAX family of computers under the VMS operating system. The programs for the rest of the DEM production process are implemented as UNIX filters, which will run under any modern variant of the UNIX operating system. The decoding of SPOT-1 headers is at present a non-trivial process, requiring the transfer of header data between VMS and UNIX systems. These difficulties have been documented by Day [Day, 1988c]. Efficiency and rationalisation demand that the functionality of the readcct program be built into a UNIX filter, permitting the complete topographic map production system to be run in a single homogeneous environment.5

In order to set up an accurate absolute camera model, ground control points [GCP's] are required to register the camera model to ground truth. If the accuracy of the model is assessed, the ground and image positions of additional check points will need to be determined. A great deal of the early work of the Alvey MMI-137 project was directed at the problem of obtaining accurate ground control and check points [Muller, 1987a; Muller 1987b; Morris et al, 1987]. This work highlighted three possible source of data:
a) The use of GPS [Global Position System]: in order to make use of GPS, a surveyor must go into the field, and place one or more GPS beacons. The position of these beacons on the ground is then located by a GPS Satellite. The GPS system is potentially capable of locating ground control points on the ground to better than 5m in plan and height. Despite this, the use of GPS is often not a preferred method of acquiring ground control points, as it is expensive. Also, practical experience with GPS has shown that there is potential difficulty in identifying the positions of the GPS points in the imagery at the level of accuracy required. Location of control and check points within the imagery was initially by manual measurement [Peacegood, 1989], accomplished using the interactive HIPS tool, stereo that the estimate of the position of control features within the imagery may be improved [by 0.3-0.5 pixels in image space corresponding to 3m to 5m on the ground] by applying the Gruen adaptive least squares [ALSC] algorithm to manually measured image points in order to obtain a refined estimate of feature position.

b) The use of maps: in principle it is possible to measure ground control points, in a local co-ordinate system, using existing mapping of an area for which a SPOT-1 DEM is to be produced. The use of existing maps requires that the operator locates map features which can also be located on the SPOT-1 imagery. For this reason, the heights of features which have high visibility both on the maps and within the imagery often have to be estimated. Morris [Morris et al, 1987] has developed software to accomplish this by sectioning contour data. In a number of tests conducted using this software, acceptable accuracies [< 10 m in plan and height] were attained using 1:25,000 or larger scale mapping.

c) The third method involves the use of aerial photography in analogue/analytical photogrammetric instruments, with the model being set up in the correct absolute orientation. As
mentioned above in (b), the operator must ensure that features located in the aerial photography are also identifiable in the SPOT-1 imagery.

Tests which have been conducted with the O'Neill-Dowman camera model [see appendices 1 and 2] have shown that in favourable conditions it is possible to set up a reasonably accurate model [RMSE < 800m] without any ground control. This Figure may be further improved by taking into account the projected positions of the image corner points [latitude, longitude] which are supplied in the SPOT-1 header. According to the SPOT-1 User Manual, [CNES, 1987], the plan error in these points varies between 35m and 100m. Use of these corner points may yield models with RMS errors of less than 100m in the absence of ground control.

6.4.2 The Gugan-Dowman camera model.

The Gugan-Dowman Camera Model [Gugan and Dowman, 1987; Gugan, 1987; Gugan, 1988] is a SPOT-1 camera model developed at UCL under NERC funding. The development of the model preceded the launch of SPOT-1 in February 1986. The model was initially used with hardcopy imagery on analytical plotters such as the Kern DSR-1 and DSR-11 instruments. Subsequently, the model was also ported to the I²S and Sun environments for use with digital imagery. Commercial systems which are based on the Gugan-Dowman model have been developed by both Kern and GEMS of Cambridge.

The model uses a time dependent form of the collinearity equations [Burnside, 1979]. These are used to establish the exterior orientation of central perspective aerial camera models. The time dependency of the equations used in the SPOT-1 model is a consequence of the dynamical nature of the SPOT-1 sensor. The Gugan-Dowman model allows for gentle undulation of the image plane during image acquisition via a low-order polynomial correction which is used to vary the ephemeris. The orbit segment is calculated without any external sources of data other than the ground control points, and the angle of the SPOT-1 HRV instrument pointing mirror. Later the Gugan-Dowman model was modified to make use of the SPOT-1 header data although the subsequent effect on model accuracy was small. The failure
to use attitude data means that the model requires a large number of ground control points. About 6 to 10 points per stereo pair are required in order to set up a good model [RMS ground error < 15m]. The minimum number of ground control points required depends upon the number of unknown parameters in the model. In order to attain performances comparable to the O'Neill-Dowman Camera Model using 2 ground control points, a 7 or 10 parameter Gugan-Dowman model is required. Such models require a minimum of 5 and 4 ground control points respectively in order to set them up. Work by Neto [Neto, 1989] has shown that 6-15 ground control points are required in order to form a good model for strips of imagery. By comparison, the O'Neill-Dowman Camera Model requires 2-4 ground control points in order to set up a good strip-model. The design and implementation of the Gugan-Dowman Camera Model is described by Gugan, [Gugan, 1988].

6.4.3 The O'Neill-Dowman camera model.

In view of the large number of ground control points required by the Gugan-Dowman Camera Model, the Alvey MMI-137 consortium decided to develop its own SPOT-1 camera model. This model is radically different to the Gugan-Dowman model, in that it derives much of the data required to set up a perspective view from the associated SPOT-1 header. Fewer GCP's are therefore required than if the traditional collinearity approach were used. Using 2-3 GCP's, the O'Neill-Dowman approach is capable of producing a model with an ground RMSE of less than 12m [vector] [8m in plan, 5m in height]. For strips of imagery a ground RMSE of between 14m and 18m [13m in plan, 10m in height] may be attained. Extrapolation studies have indicated that accuracies of this order may be maintained over strips containing many [10 + scenes]. This performance compares very well with other SPOT-1 model which have been assessed under the aegis of the OEEPE SPOT triangulation workshop held at University College London in September 1989. It is of note that many of these models are far more complex the the O'Neill-Dowman model. In particular, many models employ considerably more complex methodologies for computing both the SPOT satellite orbit and local variations of the geoid.
The DEM is generated by taking the DDM generated by the stereo matcher and feeding through the O'Neill-Dowman or Gugan-Dowman Camera Models, both of which are implemented as UNIX filters. Prior to transforming the DDM, the camera model is set up and checked using the set of accurately measured ground-control and check points. In the case of the O'Neill-Dowman model, a test mode is provided in which the check points and their corresponding space intersection points are written to file. After conversion to an appropriate local co-ordinate system, the RMSE may be computed using the UNIX filter, rms [O'Neill, 1990]. Alternatively, the vector residuals themselves may be viewed interactively using the interactive HIPS tool vec [O'Neill, 1989b], or statically using the uniras based vector_plot filter [Lewis and O'Neill, 1990]. When the operator is satisfied that a model of sufficient accuracy has been computed, the DDM can then be transformed to ground.

Both camera models work in the earth centred [geocentric] co-ordinate system. It is more convenient to have the DEM in a local system such as NG [National Grid], UTM [Universal Transverse Mercator], LZ3 [Lambert Zone 3] or the geographical co-ordinate systems. For this reason the geocentric output of the camera model is processed by a filter such as gcutm, gcLz3, or gcgeo , which performs the conversion to an appropriate local co-ordinate system. A description of the camera modelling part of the DEM production process is given by O'Neill [O'Neill, 1989a].

6.5 Generation of the Gridded DEM [GDEM].

The final phase in the generation of a DEM is the resampling of the ungridded raw output of the camera model to a regular grid, thus generating a gridded digital elevation model [GDEM]. Initially, this process was accomplished using Delaunay Triangulation followed by bilinear interpolation using the Laserscan package Panacea running under VAX VMS. Due to the complexities of transferring data between VMS and UNIX and the fact that Panacea is limited to <500000 input points, the Alvey MMI-137 project developed its own routine to produce gridded DEM's from raw camera model output. In keeping with the rest of the DEM production system, this
tool is implemented as a pipelined UNIX filter. Currently two interpolation filters are available:

a) Delaunay: This routine is an implementation of the Delaunay Triangulation algorithm for the UNIX environment. To date it has been used in conjunction with a simple bi-linear interpolation scheme for stereo medical applications, for example, the modelling of the 2.5D structure of human faces prior to orthodontal or cosmetic surgery [Thomas, 1989].

b) krig_some: This is an implementation of the Kriging algorithm [Delfiner and Delhomme, 1975; Day, 1990b] as a UNIX filter. Kriging is a statistical process originally developed in the mining industry for assessing ore reserves by J.B. Krige. Like the Panacea package, the Kriging process takes irregular input and transforms it to gridded output. However, Kriging is also capable of giving a statistical quality measure for each point Kriged. If no terrain variogram is available, the accuracy of the Kriging process is of the same order as Delaunay Triangulation. If a variogram is available, the Kriging process is capable of far higher accuracies than the Delaunay based approach. This seems like a chicken and egg situation as in order to generate a terrain variogram or semi-variogram, a DEM is required: it may be possible to design an iterative scheme, in which a variogram generated using a rough DEM is then used to refine that DEM in a self consistent manner.

An example GDEM produced by kriging is shown in Figure 6.5 [Death Valley California USA].

6.6 Uses of SPOT-1 DEM Products.

The gridded SPOT-1 GDEM may be used for a wide range of applications including:
Figure 6.5. Showing Kriged DEM generated using Death Valley SPOT-1 image sections.
a) Geomorphological studies: the GDEM yields a wealth of geomorphological information. In particular, the GDEM provides much information about the hydrology and slope of the area covered. Timely monitoring of geomorphology is also of use to organisations such as oil companies to assist their exploration efforts, and to environmental groups analysing for example, the effect of rising mean sea level due to the global warming [the Greenhouse Effect].

b) Intervisibility studies: because the SPOT-1 GDEM contains relief information, it may be used, in conjunction with suitable visualisation tools, to assess the environmental impact of building projects. Bodies such as the National Grid Company use intervisibility studies to assess the impact of their activities on the environment.

c) Perhaps the most important use of automated topographic mapping is the provision of maps for those parts of the world which are currently unmapped. Organisations such as the Ordnance Survey are currently showing interest in the potential of automated mapping techniques. At present the research and development department of the Ordnance Survey are assessing a number of automated mapping systems including the GEODEM system described here, and also the DCCS [Digital Correlation Comparison System] which has been developed for automatic aerial triangulation. An example of a topographic map of Sopka Shivelush Kamchatka which has been Lambertian shaded to emphasise relief is shown in Figure 6.6. In Figure 6.7, an orthoimage of the Isle of Wight is illustrated with the GDEM from which it was derived.

d) Topographic products derived from SPOT imagery may also form an important component of flight simulator systems in the future. At present many flight simulator systems generate the view from the aircraft cockpit using terrain objects which have been pre-defined by the manufacturers of the simulator.
Figure 6.6
Sopka Shiveluch, Kamchatka, USSR
Hill-shaded relief image of automatically-produced digital elevation model (DEM) with illumination from south-east.
Processing performed by the GEODEM SPOT-DEM Bureau Service at University College London using the UCL-Alvey research software.
DEM (C) 1989 UCL
Figure 6.7. Showing Isle of Wight orthoimage and MCE DEM:

(a) MCE DEM of Isle of Wight (C) Crown Copyright.

(b) 50m PAN SPOT-1 orthoimage of Isle of Wight synthesized using MCE DEM.
The use of DEM's coupled with advanced visualisation techniques such as fractal interpolation [Rees and Muller, 1990], may lead to more realistic environments for flight simulator systems.

e) SPOT-1 derived GDEM's may also be used to produce animations which may be used to assist scientific studies, as an educational aid, or for commercial purposes, such as advertisement. Movies produced using the GEODEM system have recently been featured by both ITV news, and the BBC program "Tomorrow's World" in which a flyover of the Sopka Shivelush Kamchatka region of Eastern Siberia was featured. A frame from the Kamchatka movie is shown in Figure 6.8.

f) Work is currently ongoing under the aegis of the SERC funded EXODUS project at UCL to investigate the systematic use of the Geodem System for Planetary Mapping applications. Figure 6.9 presents an early result of this work, a frame from the visualisation of the Ius Chasma region of Mars, which was obtained using the evileye rendering system [Day, 1988b; Day 1990a], using a GDEM derived from Viking Imagery, which was created using a DDM matched using the Otto-Chau algorithm.

6.7 Conclusion.

A system for the extraction of dense digital elevation models from a number of differing forms and scales of imagery has been described, using the SPOT-1 system as a specimen case. The system described forms the basis of a production system for the production of topographic maps from a wide variety of different types of imagery. These include SPOT-1, AVHRR and KFA1000 satellite imagery, large scale imagery, and close range images of human faces. The accuracy of the system is good: for example, DEM's derived from SPOT-1 stereo imagery using the O'Neill-Dowman sensor model typically exhibit an absolute RMS errors of less than 12m over a 60km x 60km area of terrain. The results attained with other space borne
Figure 6.8
Sopka Shiveluch, Kamchatka, USSR
Visualisation of SPOT-1 satellite image overlaid on automatically-derived digital elevation model (DEM).
DEM produced at UCL using Otto-Chau stereomatcher by T Day
Visualisation (C) 1989 UCL
SPOT image (C) 1987 CNES
sensors have proved equally satisfactory. In the case of the large scale and close range regimes, the system is still experimental, but it is expected that the results attained in this regime will ultimately be as good as those attained using space borne imagery.

Endnotes to Chapter 6

1: At present, the production system can only cope with aerial stereo pairs for which the effects of shadows, discontinuities and occlusions are small.

2: A typical 60km x 60km SPOT-1 DEM will contain more than 800,000 conjugate points.

3: A harness is a program which controls the activities of another program or programs.

4: It appears that matching at video refresh rate was accomplished using custom hardware based on the Motorola 68010 or 68020 processor.

5: A UNIX filter has now been written in UCL-PS to accomplish this task.
Figure 6.9
Ius Chasma, Mars
Visualisation of automatically-derived terrain overlaid by Viking Orbiter image.
(C) UCL 1989
Chapter 7
Future Research Directions

7.1 Introduction.

The studies which have been conducted to date using the O’Neill-Dowman camera model have highlighted a number of avenues which could be investigated in the future. These include:

a) Studies to investigate the optimal cost function and associated parameter set for the SPOT-1 camera model.

b) The development of more sophisticated orbit models which take account of the physics of the orbiting satellite.

c) Development of a multi-sensor camera modelling system.

d) Development and assessment of camera model assisted stereo matching techniques.

7.2 Finding an Optimal Cost Function and Associated Parameter Set.

The cost function and associated parameter set which are used by the O’Neill-Dowman Camera Model at present, have evolved via the manual analysis of a large number of camera models. In this manner, it was possible to find a combination of cost function and parameter set which works reasonably well; yielding camera models which are better than an arbitrary error-bound [Within the Alvey MMI-137 project, a model was deemed acceptable if the ground RMS error associated with 15 or more well distributed checkpoints was less than 15 metres].
With this arbitrary approach it is totally impossible to assess whether the models formed are in any sense optimal with regard to the choice of cost function and parameter set. In principle, it is possible to make this assessment by observing a large number of camera models which have been set up using a combination of parameter set and cost function which are drawn from a pool of possible parameters sets and cost functions. The problem of finding an optimal camera model is, like the problem of setting up an individual camera model, an exercise in constrained optimisation. In this case a parameter set/cost function combination is sought, for which the observed RMS check-point error across a number of different SPOT-1 camera models is minimised. In this context different has a dual meaning:

a) The use of more than one SPOT-1 stereo-pair to test a given parameter set or cost function. This will eliminate the effect of idiosyncrasies which are peculiar to a given stereo model.

b) The use of different control configurations, chosen from a well distributed pool of ground control, to set up multiple models for each of the stereo-pairs considered in (a). Multiple models have to be computed in order to attain a measure of statistical significance. If only one model were computed per stereo-pair, the effect of operator-error in the measurement of the control and check points may be significant.

The process outlined above represents a methodical extension of the rather haphazard scheme which was used to establish the present parameter set and cost function of the O'Neill-Dowman camera model, which is adequate, but almost certainly sub optimal. In view of the potentially enormous size of the parameter space, it is clear that the manual approach, used to find sub-optimal camera models is no longer feasible. An automatic method of performing the optimisation must be sought. Figures 7.1 and 7.2 show how the existing O'Neill-Dowman Camera model may be adapted to facilitate the parameter set and cost function optimisation described above.

The changes to the existing model are twofold:
Figure 7.1 Showing camera model organised as an interacting group of process-objects.
Description of boundary conditions

Stochastic optimiser:
Genetic algorithm or
Simulated Annealer

Optimised parameter set

Camera model data

Camera model(s)

trial parameter set

parameter set cost

Figure 7.2 Schematic of a camera geometry refinement system based on a stochastic parameter optimisation scheme.
a) The division of the existing camera model into three modules: A transformation module, which transforms points between image and object space; an optimisation module; and a cost function module. In a UNIX environment, the three modules run as co-operating, but independent processes, exchanging data via both files or other communications mechanisms, for example named pipes [System V named FIFO]. The advantage of this multi-process variant of the camera model is that it enables the experimenter to alter the cost function and its parameter set without altering any other component in the system.

b) The addition of a camera model creation module.

The functions of this module include:

1) Choice of the parameter set/cost function combination.

2) The initialisation and execution of a statistically significant set of camera models, for each parameter set/cost function combination selected from the parameter set/cost function pool.

3) Recording the best parameter set/cost function combination attained to date. It is likely that the choice of subsequent parameter set/cost function combinations will take into account past parameter set/cost function combinations which have resulted in a low RMS check-point error, taking into account all models and control-point configurations tested.

Because the search space is large and, unlike that of the individual camera model, possibly multimodal, the optimum relaxation strategy is likely to be based upon simulated annealing, genetic search or a variant thereof; [Booker, 1987; Davis and Steenstup, 1987]. Despite being optimal in terms of the time required to locate a global minimum in a multimodal function, the genetic search method, especially sexual variants of the algorithm are complex to implement. Although it is less efficient computationally, simulated annealing, [Metropolis et al, 1953; Christofides et al, 1979; Kirkpatrick,
1983] is certainly a simpler algorithm to implement.

7.3 The Development of a More Sophisticated Orbital Model.

7.3.1 Non-conservation of angular momentum in the current orbit-model.

At present the O'Neill-Dowman Camera Model uses a very simple form of orbital model consisting of position and velocity vectors which have simply been splined from the data tables supplied in the SPOT-1 header data. Although this simple approach gives a model of the orbit which is usable, work by Paramananda and Otto [under the aegis of the Alvey MMI-137 project] brings the physical credibility of this approach into question. In an ideal orbit, the orbiting body conserves angular momentum. Paramananda and Otto found that orbits which were splined in the manner described above showed a significant deviation from the ideal case, in which angular momentum is conserved. Smaller deviations from the ideal case are to be expected, as energy is put into the system, for example, when small attitude correcting retro-rockets are fired, or when the attitude of the pointing mirrors on the HRV instruments is changed.¹

7.3.2 Sensitivity of camera model to orbit segment interpolation.

The work done by Paramananda and Otto on the physical realism of orbital segments created directly from the SPOT-1 orbital segment suggests that accuracy may be enhanced by building an orbital model which is more realistic. The sensitivity of the camera model to the accuracy of the orbital segment is further supported by work done by O'Neill. In this case, the effect on camera model accuracy of approximating the orbital segment as a straight line over the 9.024 seconds during which the SPOT-1 image is acquired was investigated. Naively, as this represents only 0.0013 of the SPOT-1 satellite's orbit [assuming \( r_e \) is 6400 Km and the operating altitude of the SPOT-1 satellite is 840 Km], one might expect the linear approximation to yield a model whose accuracy is of the same order as one which uses splined orbital segments. The results of the tests conducted imply otherwise: models which yield RMS checkpoint error less than 15.0 metres using
splined orbital segments yield RMS checkpoint errors of greater than 500 metres using a linear approximation to orbit segments.

7.3.3 Improving the model of the orbit segment.

Both pieces of work described above imply that the existing camera model may be improved by developing an enhanced orbit model which takes into account the physical processes to which the satellite is subjected. A possible path for developing an enhanced model is to apply the technique of post processing to the existing orbit model. We will examine a number of schemes by which this post-processing may be accomplished.

7.3.3.1 Orbit post processing using conjugate data.

The first scheme uses conjugate data, derived from Otto-Chau Stereo matcher output. In Chapter 4, the unsuccessful application of this conjugate data to refining the attitude of the SPOT-1 sensor was described. In the present scheme, the same conjugate data would be used to refine the position rather than the orientation of the sensor.

The local ray-ray skewness is zeroed at each conjugate point by adjusting the position of the corresponding orbit point, for each iteration of the relaxation algorithm described in Chapter 2:

\[
\Delta \vec{p}_L = \frac{1}{2} \cdot \vec{m}_i
\]  

(7.1)

\[
\Delta \vec{p}_N = \frac{-1}{2} \cdot \vec{m}_i
\]  

(7.2)

Where:

\( \Delta \vec{p}_L \) is the local orbit position adjustment vector for the logical left camera position in the vicinity of the \( i^{th} \) check point.

\( \Delta \vec{p}_N \) is the local orbit position adjustment vector for the logical right camera position in the vicinity of the \( i^{th} \) check point.
Future Research Directions

\( \vec{m}_i \) is the ray-ray skewness vector for the \( i^{th} \) check point.

The adjusted orbit sensor positions, found by applying (7.1) and (7.2) at each check point may then resplined: the modified spline may then be used to generate the sensor positions used by the next iteration of the orientation algorithm. Although the beta-cubic spline used by the current version of the camera modeler may be used in the resplining process, the tensioned spline software developed by Morris [Morris et al, 1987] may prove a superior alternative.

If the position of the sensor is adjusted, the direction and magnitude of the velocity vectors must be adjusted also [in order that the orbit model remains physically credible]. The correction to the velocity vector may be found by differentiating the sensor position correction function, \( \Delta \vec{r}(t) \) using an appropriate numerical method, for example Milne’s method [Milne, 1949].

7.3.3.2 Post processing using attitude data.

In addition to using conjugate data, post processing of the orbit segment may also be accomplished using the header attitude data. As we shall see, in order to use the attitude data effectively a detailed dynamical model of the orbiting satellite is required. This is the case because the orbital adjustment is calculated from a knowledge of the coupling between the translational and rotational motions of the satellite. These coupling coefficients are a function of frequency; the lower the frequency of a rotational motion, the greater is the probability that it will couple to one or more of the translational degrees of freedom of the satellite. The coupling coefficients form a characteristic fingerprint of a given mechanical system, such as the SPOT-1 satellite: once established it is possible, theoretically at least, to compute the orbit segment position and velocity correction functions \( \Delta \vec{r}(t) \) and \( \Delta \vec{v}(t) \) from the attitude rates, given that the coupling coefficients are known. Since there is no readily available data which describes rotational-translational couplings for the SPOT satellite, the elucidation of the coupling coefficients would have to be accomplished using trial and error: using an appropriate cost function which takes account of coupling effects, an optimal set of coupling parameters for SPOT may be found as a consequence of the parameter set/cost optimisation calculation described in section 7.2.
Once a complete model of the SPOT-1 sensor has been built up, it may be possible to further address the error problem using sophisticated digital filtering techniques, such as Kalman filtering, which require an accurate model of the process to be available as a precursor to their use.

7.4 Development of a Multi-Sensor Camera Model System.

7.4.1 Scope of multi-sensor camera modelling systems.

The work described above has concentrated on the development of an efficient, robust sensor model for the SPOT-1 sensor. The ultimate aim of the present research is to produce a multi-sensor camera modelling system. Such a system would support a number of satellite sensor models, including SPOT, LANDSAT and NOAA. In addition, it would also support a simple central perspective camera model, enabling large-scale aerial photography to be processed by the system. The primary objective of the system would be the production of digital elevation models [DEM's]. One of the most desirable properties of a camera modelling system is that of synergy. The system as a whole would be able to produce digital elevation models which could not be produced by any single component of the system. Consider the following examples.

The SPOT-1 satellite is very good at producing small-scale digital elevation models, which could be used to great effect in topographic maps. On the other hand, because it has a ground resolution of 10 m PAN and 20 m XS, it is not ideal for producing digital elevation models of urban areas. Large-scale aerial photography is capable of producing accurate digital elevation models of urban areas, but it cannot easily cover the area routinely imaged by SPOT-1. Given that the camera modelling system can register one DEM to another, a symbiosis is apparent: outside urban areas [where there is little interesting detail of human interest on a sub 10 m scale] use the SPOT-1 generated DEM; in the urban areas [where there is detail of human interest on a sub 10 m scale] register the DEM generated by the aerial photography to the SPOT-1 DEM and use that instead. Thus, using both SPOT-1 and aerial photography, a DEM which contains an accurate topographic model of both
towns and the countryside which surrounds them may be built up. Using
SPOT-1 or aerial photography alone, one would be left with just the country­
side, or the towns respectively! This symbiosis of scales is not just restricted
to the case cited above: the same situation, on a different scale, would result
if a DEM generated from NOAA AVHRR [1 km pixel size], were merged
with SPOT-1 data.

A camera modelling system could also be used for simulation. For ex­
ample, given a DEM, which has been generated by a given sensor, the image
seen by a totally different form of sensor could be synthesised. Work has al­
ready been done in this area [O'Neill and Dowman, 1988], in order to pro­
vide synthetic data to test epipolar stereo matching algorithms.

The sensor assisted stereo matching techniques described in detail in
section 7.5 provide yet another example of the sort of application which
could be undertaken using a camera modelling system.2

7.4.2 Extension to non-rigid and active sensor systems.

The extension of the general purpose camera model system to non-rigid
systems such as AVHRR or ATM with their inherent non-linearities is the
biggest challenge in developing a general purpose sensor modelling system.
A great deal of thought has recently been devoted to this problem. The most
promising approach would be to adopt a linearise and solve strategy to this
problem. The mechanics of this involve finding a function \( \gamma(\text{line}, \text{sample}) \)
which transforms a non-linear input image space into a linear input image
space which may be processed by the linear sensor models which have been
developed to date.

The provision of a sensor model for active sensor systems such as SAR
[Synthetic Aperture Radar] is another area which is currently under appraisal.
Two models are required for SAR systems. The first model, a stereo model
using the range equations effectively reduces [for sensors such as SIR-B or
Seasat] to a linear push-broom model. Given that the effect of undesirable
geometric effects such as layover and foreshortening are small, [which is the
case for space borne SAR instruments], the sensor modelling system
developed to date could be used with little modification. A model of this
Future Research Directions

type has already been suggested by Curlander [Curlander, 1982].

The second model is a monoscopic model, which is for use with directionally sensitive radar altimeter systems. The essential geometry of a system of this type is shown in Figure 7.3. It can be seen that given the time of flight data, and the pointing angle \( \theta \) of the rays emergent from the sensor, the direct production of a digital elevation model is possible, given the sensor attitude and its associated orbit segment.

7.4.3 System structure.

The design considerations which have been applied to the O'Neill-Dowman SPOT sensor model; functional simplicity and modular implementation are even more important in the case of a general purpose camera modelling system. In order to prevent the wheel being re-invented, as much of the code as possible should be common to all sensor models. The library system which is described in Appendix 4 represents a useful starting point, which can be built upon, in this respect. A common data standard between all components using the system is also essential.

As well as having a modular code structure, the camera modelling system must also be easy to use, as the personnel using the system on a day to day basis will not be computer scientists. However, these users may well wish to set up their own custom sensor models, in order to simulate the imagery produced by a given sensor, or even to facilitate the design and testing of sensor which does not yet exist. In order to be flexible, modular, interactive and easy to use, the design of the camera modelling system is radically different from that of a stand alone sensor model. In the UNIX environment for example, sensor models are best implemented as a process-pipelines rather than single processes. An implementation of the SPOT-1 sensor model, incorporating a conjugate driven orbit post-processing scheme is shown in Figure 7.4. As it can be seen, the model consists of interacting atom processes, which carry out basic functions, common to all sensors which may be modelled by the system. The power of this approach lies in the fact that the user can rapidly define new sensor models in terms of network of interacting atomic processes. There is no need to compile any code, as the functionality of any sensor model can be expressed in terms of an interacting subset of
Figure 7.3 Schematic of the sensor geometry of a directionally sensitive radar [or sonar] altimeter system.
Showing an dataflow implementation of the SPOT-1 sensor model. A number of systems, notably the BAE ICES system, offer support for data flow implementations of this type.

**Figure 7.4** Showing a dataflow implementation of the O’Neill-Dowman Camera Model.
the atomic processes provided by the camera modelling system. All the po-
tential user of the system requires is a working knowledge of UNIX and one of its command-shells, for example, sh or csh. In addition to being easy to use for inexperienced users, the process-network approach also forms the basis of a powerful tool for experimentation with sensor models.

7.5 Development and Assessment of Camera Model Assisted Stereo Matching.

In the following section, the use of sensor models to assist stereo matching is considered. In addition, to place the sensor assisted matching in its proper context, other recent non-sensor based stereo work is also described. Much of this work has been applied to large scale imagery. This is because it is only at this scale, that effects such as discontinuities and occlusions which may be overcome using advanced stereo matching techniques such as sensor assisted matching, are apparent.

7.5.1 Overview of the problems faced by stereo matching algorithms.

The limitations of current image based stereo matching systems have been demonstrated in the recent work on the stereo matching of urban im-
agery by Denos [Denos, 1989]. In this work, a number of conventional image based stereo matching algorithms were used to try to match imagery con-
taining occlusions, discontinuities and shadows. Two classes of image based stereo matcher were considered. These classes may be conveniently labelled as feature based stereo matchers, typified by algorithms such as PMF [Pollard et al, 1985], and the area or texture based stereo matchers, typified by algorithms such as the Otto-Chau algorithm [Otto and Chau, 1989]. In the course of the investigation, both classes of algorithm were found to be defective in the matching of urban areas. The principal problem with the feature based algorithms such as PMF is that they are only able to match features [in the case of PMF edges]. This means that at best such an algo-

rithm is only able to give a partial stick model of the urban area. This raises two objections.
1) A great deal of useful data is being discarded. In a typical case an image containing 160,000 pixels may only yield ~6000-7000 matched pixels if it is stereo matched using PMF.

2) It is at best computationally expensive, and at worst impossible to infer the full wire frame model of the urban scene from the output of a typical PMF run. Systems which infer such information heuristically from the output of the PMF or similar algorithms do exist, an example being the TINA system [Porill et al, 1988], which was developed at Sheffield University. Systems such as this have, in general, been developed for pick and place applications in which the input imagery is significantly less complex than the large scale imagery of urban scenes. It is also of note that the orienting of such systems is inherently less complicated than is the case for aerial photography.

The second class of stereo matcher, the area based stereo matcher typified by the area based Otto-Chau algorithm, matches patches of texture which have been extracted from the stereo pair, rather than features which have been pre-extracted from the imagery. In principle, one would expect this approach to yield a significantly better result than the feature based algorithms. In practice, this is not found to be the case. The reason for this lies in the design philosophy of algorithms such as the Otto-Chau algorithm. Although the area based matching technique has proved useful for a wide range of applications ranging from the SPOT-1 imagery [for which the algorithm was initially designed under the aegis of the Alvey MMI-137 project] [Muller et al, 1988a] to human faces [Thomas, 1989], all the imagery which it has been successful in matching has possessed continuous disparity functions. The Otto-Chau algorithm requires a continuous disparity function for two reasons:

1) The prediction mechanism used in the sheet growing assumes that implicitly the disparity function is continuous.
2) The underlying adaptive least squares correlation technique, due to Gruen and Baltsavias [Gruen, 1985; Gruen and Baltsavias, 1987; Gruen and Baltsavias 1988], uses an affine transformation to distort the right hand image patch prior to correlating it with the left hand reference patch. The use of an affine transformation, implicitly limits the correlation scheme to that subset of objects whose disparity function is both continuous and smooth.

Work by Denos and O'Neill has shown that in scenes exhibiting a high proportion of discontinuities, the assumptions which are implicit in the Otto Chau algorithm break down. Such images exhibit a disparity function which is broken up into a number of distinct disparity domains, as illustrated in Figure 7.5. In such an image, depending on the exact parameters which have been supplied to the Otto-Chau algorithm, the sheet growth is either inhibited at the domain walls or the sheet continues to grow past the domain wall with significant blundering occurring, as illustrated in Figure 7.6. It appears that the significant factors in determining the behaviour of the Otto-Chau algorithm at domain walls are the patch radius and S threshold parameters, supplied to the algorithm. The patch radius parameter controls the size of the environmental support patches about a point which is to be matched, and the S threshold parameter controls the maximum rotation which may be applied to the non-reference image patch relative to the reference image prior to correlation. Much time has been invested in determining an optimum parameter set for the Otto-Chau algorithm, which will enable it to match a scene using seed points derived from another [feature based] stereo matcher such as PMF. The logic here is to supply at least one seedpoint per each disparity domain. In theory, this should enable the Otto-Chau algorithm to match a significant proportion of each disparity domain. In practice, this first pass urban stereo matcher [USM], shown in Figure 7.7, has not been as successful as one might hope. Firstly, the PMF algorithm which was used to generate the seed points for Otto-Chau tends to match linear features which lie along the domain walls. It has been suggested that for a segmented image, the domain walls represent a highly non-optimal starting point for sheet growing. Experiments which have been performed by Denos and O'Neill tend to support this view. Secondly, experiment has also shown that it is
Figure 7.5. Showing the concept of disparity domains in a segmented image.
Figure 7.6 Showing the output of the USM stereo matcher for typical urban imagery. Note the sheet growth overhanging vertices. Subsequent analysis of parallax has shown much of this overhanging sheet to be inaccurately matched. Figures (a) and (b) are the images to be matched (left and right respectively), figures (c) and (d) are coverage images showing what areas the images have been matched (c) and not matched (d).
Figure 7.7 Schematic of the USM stereomatcher.
Future Research Directions

exceedingly difficult to get the Otto-Chau stereo matcher to match successfully in areas where the disparity domains are small in comparison to the patch size, \( \text{patchsize} = 2r + 1 \), where \( r \) is the patch radius. With this regime, the sheet growing mechanism may \textit{jump} the domain boundary, with the result that regions of the image are matched inappropriately. In this situation, the tendency for the sheet to jump narrow disparity domains may be reduced by decreasing the patch size. However, because of the \textit{non orthogonal} nature of the Otto-Chau parameter space, it was found that reducing the patch size below a critical limit \( r \approx 8 \) for urban imagery, led to significant blundering due to distortional effect.

7.5.2 Generalisation of epipolar algorithms to non epipolar geometries.

The image-matching experiments which have been done to date imply that stereo matching using direct \textit{image space} based matching techniques are feasible for texture based images in which the \textit{disparity function} is continuous. As the number of discontinuities in the disparity function grows, both the complexity of the matching scheme and the computational expense of performing the matching grow to an unacceptable limit. There are \textit{two} approaches which may yield practical \textit{general} solutions to this problem:

a) The use of a camera model to restrict the class of permissible solutions.

b) The use of appropriate image based cues, for example edge-points generated by a suitable edge-detection filter such as Marr-Hildreth [Marr and Hildreth, 1980], or Canny [Canny, 1986], to detect those regions within the image space in which the disparity function is likely to become discontinuous.

If the camera model is known, it is possible to \textit{remap} the stereo image pair into epipolar form as illustrated in Figure 7.8. Many algorithms which are intrinsically good at matching images which contain occlusions and discontinuities, for example the Pollard Mayhew Fribsy algorithm - PMF [Pollard et al., 1985], the Dynamic Programming Technique [Ohta and Kanade, 1985] and the PRISM algorithm [Nishahara, 1984] are all effectively restricted to epipolar or quasi-epipolar geometries. With the provision of a
P1, P2 are the position vectors of the perspective centres of sensors 1 and 2.
Vs, e1, Ps are the sensor reference axes.

Figure 7.8. Showing the geometry of epipolar line projection using an absolute sensor model. Note that a simple shift is shown here for clarity. In general, [B - B] is the projection of [A - A] in the image plane of sensor 2 which may be both rotated and shifted relative to sensor 1.
camera model, it is possible to efficiently resample the image to a quasi-epipolar form *before* the stereo matching algorithm proper is applied. The resampling phase uses the camera model to project the image of a scan line [a-a] in the reference image into the non-reference image. The line [b-b] in the non-reference image is then the quasi-epipolar mate of the line [a-a]. Resampling along each line [b-b] will produce an image pair which is effectively epipolar. The precise form of the camera model is dependent upon the type of sensor used to acquire the imagery. The use of the camera model allows algorithms which are potentially very good at matching scenes which contain occlusions and discontinuities to be applied to imagery whose geometry is non-epipolar. The provision of robust stereo algorithms which are capable of working with non-epipolar imagery is of particular importance in photogrammetry and remote sensing; large scale imagery which has been acquired by airborne sensors is almost always non-epipolar, and flights over urban areas are likely to result in imagery which contains a large number of discontinuities and occlusions.

It is notable that the algorithms described above are much less CPU intensive than the Otto-Chau algorithm. This is because the invocation of the epipolar constraint effectively reduces the dimensionality of the search space. It is likely therefore, that non-epipolar versions of these algorithms, employing a camera driven resampler will also be computationally efficient, compared to a purely image based technique, given that it is possible to implement the resampling stage efficiently. A schematic of the camera-resampled stereo matching system is shown in Figure 7.9.

7.5.3 Adaptation of image space area based algorithms to deal with discontinuities.

It is also possible to adapt image-space texture/area matchers such as the Otto-Chau algorithm to deal effectively with discontinuous imagery. Recent work, has indicated that the adoption of the pyramidal processing scheme which is described in Section 7.5.7, in conjunction with an edge map to mark the disparity domain boundaries, permits the basic texture/area technique to be extended to discontinuous imagery.
Figure 7.9 Showing schematic of camera model assisted stereo matching using non-epipolar imagery with an epipolar stereo matching algorithm
To date standard edge detection techniques such as the Marr-Hildreth and Canny edge detection filters have been used to infer the boundaries of the disparity domains. It is likely that better results will be obtained if there is less ambiguity in the edge maps. Experiments with the both the PMF algorithm and filters which use edge detection in conjunction with region growing support show that these methods produce superior edge-maps to an edge detection technique applied in isolation.

7.5.4 An example of a camera constrained stereo matching algorithm.

In the preceding section, the utility of a sensor model in building a robust efficient stereo matching system, capable of dealing with non-epipolar imagery was discussed. In this section we shall discuss how a specimen algorithm due to Ohta and Kanade may be extended to produce dense disparity maps from non-epipolar imagery.

In any area based stereo matching system, the principal problem is to find some mapping function $\eta$, such that a pixel in a reference image is unambiguously mapped into another image space, with which it shares overlapping object space coverage:

$$P^*_r = \eta(P^*_i) \quad (7.3)$$

Where:

$P^*_r$ is the pixel in the non reference image corresponding to pixel $P_i$ in the reference image.

In the case of a general area based stereo matching problem, the computation of $\eta$ is difficult due to the effects of occlusion within the object space. In such circumstances $\eta$ has no simple analytical form. In the following section, a knowledge of the relative camera model was used to build a warping vector $F^*_u$ which can be considered to be the pixel-pixel mapping function, $\eta$, expressed in object space. For a real world stereo matching system, the estimation of $\eta$ can often be accomplished much more efficiently in image space, using the camera model to constrain the search. This is especially true for linear sensors such as SPOT-1 for which the mapping between image space and object space is complex [O'Neill and Dowman, 1989b; O'Neill
and Dowman, 1991]. Using the camera model to locate the quasi-epipolar line in the non-reference image, effectively reduces \( \eta \) to a one dimensional function, as any parallax between the reference and non-reference images must lie along this line. In order to compute \( \eta \), the texture in the non-reference image must be resampled along the epipolar line. \( \eta \) will now take the form:

\[
\Delta x = \eta(x)
\]  

(7.4)

Where:

\( \Delta x \) is the parallax for some pixel at a position \( x \) along the epipolar line in the reference image.

If the texture to be matched does not contain any occlusions and is thus a single disparity domain, \( \eta \) is well approximated by a set of polynomials, \( p_i(x) \), such that:

\[
\Delta x = \sum_{i=0}^{i=N} p_i(x)
\]  

(7.5)

Equation (7.5) above may be combined with the correlation equation in order to find a set of polynomial coefficients which results in the optimum correlation of texture between the reference and non-reference images.

\[
C_i = \beta(P_i^r) - \beta(P_i^t)
\]  

(7.6)

Where:

\( P_i^r \) is the position vector corresponding to pixel \( P_i \) in the reference image.

\( \beta \) signifies the operation: take the brightness at the current pixel position.

\( C_i \) is the subcomponent of the cost function due to the pixel vector pair \( [P_i^r, P_i^t] \).

In order to deal adequately with the problems of occlusions, shadows and discontinuities, the best approach to the computation of \( \eta \) is afforded by a modification of the stereo matching algorithm described by Ohta and
Kanade [Ohta & Kanade, 1985]. In Ohta and Kanade’s algorithm, a dynamic programming approach is used to find the optimal path through a [two dimensional] cost space as indicated in Figure 7.10. The computational cost used by Ohta and Kanade is defined in terms of the similarity between the intensities of pixels in regions in the corresponding left and right scan-lines, which are the projections of the appropriate segment of the path in cost space along the two scan-line axes. The nodes which demarcate the linear segments into which the path is subdivided lie on discontinuities in intensity which have been detected in the input imagery via the application of an appropriate edge detection algorithm:

\[
\tau = \sum_{i=1}^{N-1} l_i \quad (7.7)
\]

Where:
- \(\tau\) is the path through cost space,
- \(N\) is the number of nodes in the path \(\tau\),
- \(l_i\) is the linear segment between the \(i^{th}\) and \((i+1)^{th}\) nodes in the path.

The restriction of the algorithm to epipolar imagery is a mechanism to reduce the dimensionality of the cost space: if disparity is permitted in both the line and sample directions, the cost space necessarily extends over three dimensions. As indicated in Figure 7.11, a relative orientation may be used to locate a 2 dimensional plane within the 3 dimensional cost space, within which the path of minimum cost must lie. This plane intersects the non-reference image along a quasi-epipolar line [b-b].

The minimum cost path for any stereo image pair, \(\tau_r\), may be expressed as a function of the minimum cost path in the absence of relief, \(\tau_0\) as illustrated in Figure 7.12:

\[
\tau_r = f(\tau_0) \quad (7.8)
\]

The function defined in equation (7.4) has the property that it is always quasi-continuous even if the corresponding imagery contains discontinuities.
Figure 7.10 2D search plane for intra-scanline search. Intensity profiles are shown along each axis. The horizontal axis corresponds to the left scanline and the vertical one corresponds to the right scanline. Vertical and horizontal lines are the edge positions, and path selection is done at their intersections. (aft. [Ohta and Kanade, 1985])
Figure 7.11. Showing how a relative orientation may be used to extend Ohta and Kanade's algorithm to non-epipolar imagery.
Figure 7.12. Showing how the minimum cost path, \( f \), may be expressed as a function of the zero relief minimum cost path.
A discontinuity in the imagery may be readily detected as it will cause either:

a) A disparity reversal, in which the relative disparity vector between the $i^{th}$ and $i+1^{th}$ pixel in the non-reference image becomes locally reversed. [supercritical case].

b) A zero relative disparity condition, in which the relative disparity between the $i^{th}$ pixel and the $i+1^{th}$ pixel in the non-reference image is zero. [critical case].

This observation leads to a simple rule: All critical or supercritical matchings are to be discarded, as they either lie on discontinuities [critical matches] or within zones of occlusion [supercritical case].

The fact that $\tau$ may be quasi-continuous implies that the dynamic programming approach may be used to produce dense disparity digital elevation models [DDEM's] for imagery which contains a large number of edgels which are unambiguously located in both images.

This continuity condition also allows the dynamic programming approach to be extended to imagery in which edge based features are more sparse. In this case, the linear segments which form the function $\tau$ may be replaced by a sum of Tshebyshev polynomials in order to yield a continuous function $\tau_c$:

$$
\tau_c = \sum_{j=1}^{j=M} a_j \cdot t(\tau_0)_{i,j}
$$

(7.9)

Where:

$\tau_c$ is the continuous path through cost space,

$N$ is the number of nodes in the path,

$M$ is the number of Tshebyshev polynomials describing the $i^{th}$ segment in the path.

$a_j$ is the coefficient of the $j^{th}$ Tshebyshev polynomial, $t(\tau_0)_{i,j}$. 
The optimal set of Tshebyshev polynomials within each path segment may be found by using an appropriate optimisation method. If the imagery is presented to the algorithm in a coarse to fine pyramidal fashion, unimodal minimisation techniques such as adaptive least squares, or the Powell Direction set method may be used. Recent work with the purely image based Otto-Chau algorithm has shown empirically, that a coarse to fine pyramid of images may be readily globally minimised by unimodal minimisation techniques. This technique is almost certainly more efficient in computational terms than its obvious alternative, using a multimodal minimiser such as a genetic algorithm or simulated annealer on fixed-scale imagery. Hybrid fixed scale techniques, using a multimodal minimiser to guess an initial solution, and a unimodal minimiser to refine the guess have not yet been assessed.

The function $\tau_e$ which describes the optimal mapping between the two, potentially discontinuous intensity spaces is found by finding an optimal set of Tshebyshev coefficients $a_j$ using an appropriate optimisation method. The use of a continuous function $\tau_e$ to describe the optimal mapping between the intensity spaces of the left and right images obviates the requirement for edge-ordering which is of course required if the path of minimum cost is segmented on the basis of edges detected within the imagery.

7.5.5 Extension of camera constrained search methods to other stereo matching algorithms.

The method of using a camera model to constrain image space searching along quasi epipolar lines is not restricted to Ohta and Kanade's algorithm. This algorithm has been chosen for a detailed discussion of the method as it is capable of producing dense disparity maps.

With the use of a camera model, other algorithms initially limited to epipolar imagery could also be extended to work with non-epipolar imagery. A notable example is PMF: the inclusion of a camera model derived relative orientation within the PMF algorithm would permit the [non-epipolar] non-reference image to be resampled along quasi-epipolar lines, thus permitting the PMF algorithm to operate with such imagery.
7.5.6 Metrics for inter-image similarity measure.

Although the approach described above could use the simple similarity measure given in equation (7.4), the preferred measure is that used in the original Ohta and Kanade algorithm:

\[ m = \frac{1}{2} \left( \frac{1}{k} \sum_{i=1}^{k} a_i + \frac{1}{l} \sum_{j=1}^{l} b_j \right) \]  

\[ \sigma^2 = \frac{1}{2} \left( \frac{1}{k} \sum_{i=1}^{k} (a_i - m)^2 + \frac{1}{l} \sum_{j=1}^{l} (b_j - m)^2 \right) \]  

Where:
- \( m \) is the mean of all pixel values in the intervals in the left and right images mapped by the cost space function \( \tau \) or \( \delta \),
- \( k \) is number of pixels in the left image,
- \( l \) is the number of pixel in the right image,
- \( a_i \) is the grey level value of the \( i^{th} \) pixel in the left image,
- \( b_j \) is the grey level value of the \( j^{th} \) pixel in the right image,
- \( \sigma^2 \) is the variance of all pixel values in the intervals in the left and right images mapped by the cost space function \( \tau \) or \( \delta \).

The cost, \( C_p \), of the primitive path \( \tau \) or \( \delta \) is then defined:

\[ C_p = \sigma^2 \sqrt{k^2 + l^2} \]  

7.5.7 Non camera constrained algorithms for matching large scale imagery.

Camera constrained stereo matching algorithms suffer from the drawback that sufficient data to form a camera model must be available before the matching process can proceed. One of the chief attractions of area based algorithms which do not require epipolar constraints, is that stereo matching can proceed in the absence of a camera model. We will now look briefly at a number of ways in which the area based Otto-Chau algorithm may be adapted in order to perform better with large scale imagery.
The optimum way of adapting sheet/area based matching algorithms to deal with imagery containing discontinuities, is to incorporate an edge map generated by running a suitable operator, for example the Marr-Hildreth operator, into the process. The edge map is then used by the Otto-Chau algorithm to identify potential disparity domains. This permits the algorithm to limit sheet growth at the walls of the disparity domain; blundering at disparity domain walls is a major source of error when unmodified area based algorithms are used to match imagery which contains discontinuities. One limitation of this approach is that it requires at least one seed point per potential disparity domain. This does not present a problem, as a number of autoseeding schemes have been developed for area based stereo matchers including:

1) The use of the Foerstner [Foerstner et al, 1987] or Moravec [Moravec, 1977] operators to detect interest points in the stereo imagery. Subsequently these points are sorted into conjugate pairs using the Gruen algorithm. A 2-D relative orientation is desirable in order to limit the size of the disparity window in the right image which it has to search for features which match to a given point in the left image. A scheme of this nature has been implemented at UCL by Allison [Allison et al, 1991].

2) The use of a top-down pyramidal autoseeding technique. The use of a technique of this type extends the pull in range of correlators which use unimodal minimisations schemes, such as Gruens' algorithm, by decomposing a global minimum into a succession of local minima, each at a different scale. By applying the unimodal minimiser recursively at each scale of size, we can effectively force it to perform a global optimisation. It is notable that the number of operations required to locate a global minimum in this manner is probably less than the number of steps which would be required by stochastic global optimiser such as a genetic algorithm or a simulated annealer. A pyramidal autoseeder [and stereo matcher], cascade, which uses coarse to fine optimisation has been developed by O’Neill and Denos at UCL.
In the case of small images, it is also possible to generate a set of seeds manually using a digital photogrammetric workstation package.

7.5.7.1 A comparison of the different seeding techniques.

The Foerstner or Moravec operators tend to detect points which are located on edges or corners within the imagery, whereas the pyramidal matching scheme tends to produce points which are evenly distributed over the imagery. The large scale image matching tests described by Denos [Denos, 1989] showed that for segmented images consisting of several disparity domains, the Otto-Chau algorithm matches optimally if the centre rather than the edges of the disparity domains are seeded. The implication of this is that the cascade algorithm is the preferred method of seeding, as it rapidly generates a large number of good seeds [a seed is deemed to be good if the Otto-Chau algorithm is able to grow from it] which are well distributed over the image. Manual measurement of seedpoints is much slower and less accurate, but is still potentially capable of generating a well distributed set of seeds. Allison’s techniques which are based on feature detectors such as the Foerstner and Moravec operators are capable of accurate seedpoint detection. It is less generally useful than the cascade algorithm for two reasons. Firstly, the seeds are generated on disparity domain boundaries. They are therefore unlikely to be optimal for seeding the Otto-Chau algorithm because of the reasons given above. Secondly, compared to the cascade algorithm, Allison’s approach is painfully slow. Typically, the method takes 2 CPU minutes to generate a good seedpoint. In the case of segmented imagery, we may well require 100 or more seedpoints in order to ensure that there is at least one seedpoint per disparity domain. Allison’s methods would typically require 2.7 CPU hours to generate this number of seeds. This does not compare favourably with the 5-7 CPU minutes taken by the cascade algorithm to generate a seed dataset of similar size.

7.5.7.2 The cascade algorithm as a pyramidal stereo matcher.

A further advantage of the cascade algorithm is that it is able to function as a pyramidal stereo matcher in addition to being an autoseeding algorithm. The flowchart of the cascade algorithm shown in Figure 7.13 indi-
Fig 7.13. Cascading Flowchart for Conjugate Point Detection
cates how this may be accomplished. The output of the \textit{cascade} algorithm applied to stereo aerial photography of the Bloomsbury area of Central London, operating as a simple autoseeder is shown in Figure 7.14.\textsuperscript{3} In Figure 7.15, the DDM generated by the \textit{cascade} algorithm applied to the Bloomsbury aerial photography, operating as a \textit{combined} autoseeder and pyramidal stereo-matcher is shown. Because the imagery has been matched on many scales, pyramidally generated DDM's will often be significantly denser than those generated at fixed scale of size. This is particularly true in the case of large scale imagery containing discontinuities and occlusions such as the Bloomsbury Imagery. For some forms of imagery, the pyramidal technique may be the only practical way of generating a dense DDM. For example, Denos [Denos, 1991a] has recently used pyramidally autoseed and stereo matching methods to generate dense DDM’s from noisy SIR-B SAR imagery of Mount Shasta, USA. As we can see in Figure 7.16, application of pyramidal the techniques yields a dense DDM. The application of the Otto-Chau algorithm at a fixed size scale did not give sufficient coverage to produce a figure.

\textit{Endnotes to Chapter 7}

1: Work on a predictive model of the ITIR sensor has confirmed the view that the attitude data is of very limited use in improving the accuracy of SPOT-1 camera model. This assertion is supported by earlier work by Gugan, and also from a comparison of the accuracy of the Westin SPOT-1 camera model [Westin, 1990], which uses the attitude data, and variants of the O’Neill-Dowman camera model which do not.

2: A modular general purpose camera modelling system has now been developed using the SPOT-1 sensor model which is the subject of this thesis as a basis, although it is compilation, rather than process module based. The use of this general purpose system to build a predictive model of the ITIR along-track sensor system is discussed in Appendix 6.

3: The Bloomsbury aerial photography is of scale 1:5000. The run numbers of the photographs are 518754 and 518755 for the left and right images respectively. The photographs were recorded on Agfapan 25 professional film by a camera using a Wild Universal Aviogon lens [with central perspective]. The principal distance is 153.52mm, and the aperture setting is \( \frac{f}{4} \)
Figure 7.14. Showing the output of the cascade algorithm operating in autoseeding mode:

(a) Left image.
(b) Right image.
(c) Raw conjugate point output.
(d) Conjugate point output filtered by ECOBE global epipolar constraint filtering.
Figure 7.15. Showing DDM generated by application of Cascade Algorithm to Bloomsbury Imagery:

(a) Left image.
(b) Right image.
(c) Area not stereo matched.
(d) Area stereo matched.
Figure 7.16. Showing stereo matched coverage of SIR-B image of Mount Shasta:

(a) Left image.

(b) Right image.

(c) Area not stereo matched.

(d) Area stereo matched.
Chapter 8
Conclusions.

8.1 Introduction.

A general purpose sensor modelling system has been developed, based on a novel numerical orientation scheme. Currently, the sensor modelling system is capable of supporting models of a wide variety of rigid linear sensors. Careful attention has been paid to the software engineering aspects of the system: This has permitted new sensors to be modelled by the system with minimal effort. Sensor models exhibiting both cross-track and along track stereoscopy are supported. The former class of sensor is represented by the SPOT-1 and SPOT-2 sensor models, while the latter class of sensor is represented by the ITIR, MISR, MEOSS and ASAS sensor models. At the time of writing, support for both central-perspective and non-rigid linear sensors systems such as AVHRR and ATM is also planned.

8.2 Accuracy of modelling system.

Extensive accuracy tests have been performed with both the SPOT-1 and ITIR sensor models. These tests have shown that the novel absolute orientation system is potentially capable of high accuracies. With only two ground control points, the SPOT-1 sensor modeller is capable of forming models with a sub-pixel ARMSE [< 10m] under favourable conditions. With strips of up to 4 scenes, the SPOT-1 modeller consistently produced ARMSE's of between 14 and 17 metres: The performance of the model compared very favourably with other SPOT-1 sensor models whose accuracy was examined at the OEEPE SPOT-1 workshop held at UCL in September 1989. The ITIR modeller, using simulated data, as the EOS mission carrying the ITIR sensor is yet to be flown exhibited a performance envelope which was similar to the SPOT-1 sensor model.
In both cases, the numerical relaxation scheme developed within the framework of this thesis proved highly effective in orienting the models using just 2 or 3 ground control points. The relaxation scheme has in fact proved to be more effective than initially thought possible. This is because the non-linear attitude and position perturbations experienced by space borne linear-pushbroom sensor system are small, and trial cost functions used by the relaxation scheme were essentially good at removing linear errors. If the attitude and positional errors become larger, it is necessary to incorporate the ephemeris into the cost function in order to take account of the subsequent warping induced in the image plane. It is clear that this approach will have to be adopted in order to form accurate models for aerial sensors such as ASAS, MEOSS or CEASAR, in which the attitude variation is much more extreme. Currently, methods of elucidating attitude data from 3D relative orientations of airborne sensor systems is being studied by the Leverhulme Trust Project at UCL.

It was possible to show that the performance of both the SPOT-1 and ITIR sensor models is limited by the accuracy with which ground control points can be measured in image space.

8.3 Principal sources of error.

Work by Peacegood [Peacegood, 1989] suggested that under optimum conditions, a GCP can be located in a SPOT-1 PAN image to an accuracy of about a third of a pixel [which corresponds nominally to a 3-6 metre error on the ground]. The effects of this measurement error was simulated for both the SPOT-1 and ITIR sensors. These simulations showed that a high proportion of the observed ARMSE could be attributed to the highly non-linear observation errors made by the operator when observing the GCPs. If the GCP measurement error could be removed, it is possible that the ARMSE would be reduced to between 3 and 5 metres for the SPOT-1 sensor and between 3 and 8 metres for the ITIR sensor. Experience with SPOT-1 stereo imagery of The Isle of Wight and South Hampshire, and Oman suggest that a smaller but still significant source of non-linearities may be removed by using GPS [Global Positioning System] to measure the ground position of GCP and
8.3.1 Methods of removing error

There are two methods which could be used to limit operator error in both ground control and check points used by the sensor modelling system.

8.3.1.1 Pre-processing by stereo matcher.

The optimal method would be to use an appropriate stereo matching algorithm, for example the Gruen ALSC technique to refine points which have been measured by the operator. This method has already been used within University College with some success; although the effects of such refinement has yet to be assessed quantitatively, it is likely that pre-processing using Gruen's method could reduce measurement errors to better then 0.1 pixels, which corresponds to a ground error of less than 1 metre. This will bring the ARMSE of real camera models close to the low [3-8 metre] values suggested by simulation.

8.3.1.2 Post processing by multimodal relaxation scheme.

The second method of reducing the effect of GCP and checkpoint measurement error results from a detailed analysis of the relaxation curve characteristic of both the SPOT-1 and ITIR sensor models. The form of these relaxation curves as a function of random GCP measurement error are shown in Appendix 6. Examination of these curves shows that the unimodality of the ARMSE function breaks down when the ARMSE is less than 5 metres. This is caused because the ARMSE function contains non linearities which become apparent at this scale of size. The probable cause of this effect is the inability of the linear relaxation cost function to deal with small scale non linear perturbations in the sensor position and attitude. Observation errors made by the operator may also play a role in these small scale non linear errors. It is possible that the effects of these errors may be significantly reduced by adaptively switching to a multimodal relaxation scheme when the ARMSE is of the order of 5 metres. The problem with this approach is that multimodal minimisation schemes - such as simulated annealers and genetic algorithms are computationally expensive and inefficient. Therefore, it is
better to remove observation error before sensor modelling commences and to provide a relaxation cost function which capable of absorbing non-linear effects to the desired level of accuracy.

**Endnotes for chapter 8**

1: In order to form a camera model with a sub-pixel ARMSE the position of the GCP in object space must be located using GPS. The corresponding image space positions of the GCP point must be refined after they have been measured. The Gruen ALSC technique may be used to perform this conjugate point refinement.

2: The Leverhulme Trust Project is concerned with developing a prototype system for extracting topographic maps and associated products from a variety of airborne sensors in order to facilitate timely monitoring of environmental change.

3: But we still cannot be sure that the Gruen ALSC algorithm is locating the same feature in image space that we have measured in object space.

4: Observation error will introduce subsidiary minimina into the ARMSE function. With a function of this kind, a unimodal relaxation scheme such as the Powell Direction Set may well seek the wrong minimum, leading to the behaviour observed in appendix 6.
Bibliography

[Acton, 1970]

[Allison et al, 1991]

[Barnard and Thompson, 1980]

[Boddington, 1991]

[Booker, 1987]

[Bourne, 1983]

[Bowyer and Woodwark, 1983]

[Boyle and Smith, 1970]
[Brooks, 1977]

[Brent, 1973]

[Bunnick, 1978]

[Burnside, 1979]

[Canny, 1986]

[CNES, 1977]
Centre National d'Etudes Spatiales [CNES], *Systeme Probatoire d'Observation de la Terre [SPOT]*, Rapport de Phase A, 5 Volumes.

[Chevrel et al, 1981]

[CNES, 1979]
Centre National d'Etudes Spatiales [CNES], *Plateforme multimmision SPOT*, Manuel de l'utilisateur.
[CNES, 1987]

[Collins et al, 1987]

[Chau, 1987a]
Chau T.K.W., Specification and implementation of the PMF stereo ranging algorithm, Alvey MMI-137 internal report M01S1CS [part], Department of Computer Science, University College London.

[Chau 1987b]
Chau T.K.W., Implementation notes on the PMF2 stereo algorithm, Alvey MMI-137 internal report M01S1CS [part], Department of Computer Science, University College London.

[Chau, 1988a]
Chau T.K.W., Design and implementation notes for UCL-CS version of Gruen’s stereo matching algorithm, Alvey MMI-137 internal report M21S1CS [part], Department of Computer Science, University College London.

[Chau, 1988b]
Chau T.K.W., 1988, Design and implementation notes for UCL-CS region growing version of Gruen’s stereo matching algorithm, Alvey MMI-137 internal report M21S1CS [part], Department of Computer Science, University College London.

[Chau, 1988c]
Chau T.K.W., Implementation notes for automatic seed point generation and disparity estimation, Alvey MMI-137
Bibliography

internal report M21S1CS [part], Department of Computer Science, University College London.

[Christofides et al, 1979]

[Curlander, 1982]

[Davis and Steenstrup, 1987]

[Day, 1988a]
Day T., *Gruen_view UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

[Day, 1988b]
Day T., *evileye UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

[Day, 1988c]
Day T., *How_to_Readheader UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

[Day, 1988d]
*How_to_Gruens UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

259
[Day, 1988e]
Day T., *prow UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

[Day, 1989]
Day T., *Marseilles the Movie [Mission to Earth: Marseilles]*, Alvey MMI-137 internal report M36D1PS, Department of Photogrammetry and Surveying, University College London.

[Day, 1990a]
Day T., *Renderer for whole SPOT scenes for 2.5D visualisation* Alvey MMI-137 internal report M36S1PS, Department of Photogrammetry and Surveying, University College London.

[Day, 1990b]

[Day and Muller, 1989a]
Day T. and J-P Muller, *Quality assessment of the Otto-Chau stereo-matcher for SPOT-1 imagery*, Alvey MMI-137 internal reports M22R1PS and M35S1PS, Department of Photogrammetry and Surveying, University College London

[Day and Muller, 1989b]

[Day and Muller, 1989c]
[Delfiner and Delhomme, 1975]

[Denos, 1989]

[Denos, 1991a]

[Denos, 1991b]

[Denos and O’Neill, 1991a]

[Denos Mia and M.A. O’Neill]

[Devereux, 1989]

[Dowman, 1987]
Dowman I.J., *The accuracy of SPOT-1 orientation data*, Alvey MMI-137 internal report M28R1PS [part], Department of Photogrammetry and Surveying, University College London.
[Feldman, 1979]

[Foerstner et al, 1987]

[Forrest, 1981]

[Fish, 1987]

[Gruen, 1985]

[Gruen and Baltsavias, 1987]

[Gruen and Baltsavias, 1988]

[Gugan, 1987]
[Gugan, 1988]

[Gugan and Dowman, 1988]

[Guichard and Pikeroen, 1988]

[Haan, 1989]
Haan A., *Triangulation with SPOT Data at the Istituto di Topographia, Fotogrammetria e Geofisica: Methods and Results*, OEEPE Workshop on Block Triangulation of SPOT Data, Department of Photogrammetry and Surveying, University College London, September 1989.

[Hayes 1989]
*The Intel 80860 Microprocessor - A Cray on a chip?* Byte 1989

[Hoffman, 1988]

[Jacobs, 1977]

[Jaynes, 1980]
[Jaynes, 1983]

[Kirkpatrick, 1983]

[Kernighan and Ritchie, 1978]

[Kernighan and Ritchie, 1988]

[Kratky, 1989]

[Landy and Cohen, 1982]

[Lewis and O'Neill, 1990]
Lewis P. and M.A. O'Neill, *vector_plot UNIX man page*, Alvey MMI-137 Software Documentation, Department of Photogrammetry and Surveying, University College London.

[Lippman, 1989]

[McGregor and Watt, 1986]
[Marr and Hildreth, 1980]

[Metcalfe and Boggs, 1976]

[Metropolis et al, 1953]

[Milne, 1949]

[Moravec, 1977]

[Morris et al, 1988]

[Mikhail, 1991]

[Mullender et al, 1990]
[Muller, 1987a]
Muller J-P., *Summary of GCF's available for absolute orientation of SPOT-1*, Alvey MMI-137 internal report M13D1PS, M13D2PS, M13R1PS, M28R1PS [part], Department of Photogrammetry and Surveying, University College London.

[Muller, 1987b]
Muller J-P., *A strategy for assessing the utility and accuracy of map-based 3D ground control features [3GCF's] for the absolute orientation of SPOT-1 imagery*, Alvey MMI-137 internal report M28R1PS [part], Department of Photogrammetry and Surveying, University College London.

[Muller et al, 1988a]

[Muller et al, 1988b]

[Muller et al, 1990a]

[Muller et al, 1990b]
[Muller et al, 1990c]

[Nelder and Mead, 1965]

[Neto, 1989]

[NASA, 1972]
National Aeronautics and Space Administration [NASA], Advanced scanners and imaging systems for earth observation, sp-335.

[NASA, 1979]
National Aeronautics and Space Administration [NASA], Preliminary Stereosat Mission Description JPL 720-33.

[Nishahara, 1984]

[O’Neill, 1987]

[O’Neill, 1988a]

[O’Neill, 1988b]
O’Neill M.A, GPROC - an Integrated System for the Processing of Numerical Data, Software - Practice and Experi-
Bibliography


[O’Neill, 1988c]

[O’Neill, 1989a]
O’Neill M.A., How to Camera Model UNIX man page, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

[O’Neill, 1989b]

[O’Neill, 1990]

[O’Neill, 1991a]

[O’Neill and Dowman, 1988]

[O’Neill and Dowman, 1989a]
O’Neill M.A. and I.J. Dowman, A new SPOT-1 Camera Model which uses SPOT-1 header data and a minimum number of ground control points, Alvey MMI-137 internal report M28S3PS [part], Department of Photogrammetry and Surveying, University College London.
[O'Neill and Dowman, 1989b]

[O'Neill and Dowman, 1989c]

[O'Neill, 1991b]

[O'Neill and Denos, 1991]

[O'Neill and Dowman, 1991]

[Ohta and Kanade, 1985]

[Ordnance Survey, 1983]
Ordnance Survey Information, *Transverse Mercator Projection*, Published by The Ordnance Survey, 1983.
Bibliography

[Otto, 1986]

[Otto and Chau, 1989]

[Paramananda, 1988a]
Paramananda V., *disp UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying.

[Paramananda, 1988b]
Paramananda V., *points UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying.

[Paramananda, 1988c]
Paramananda V., *stereo UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, 1988

[Peacegood, 1989]
Peacegood G., *Measuring GCP's - How accurate is it?* Alvey MMI-137 internal reports M22D2PS, M28R1PS, Department of Photogrammetry and Surveying, University College London.

[Picht, 1989]
*Processing of SPOT Image Blocks with Program BINGO*, OEEPE Workshop on Block Triangulation of SPOT Data, Department of Photogrammetry and Surveying, University College London, September 1989
[Pitkin, 1987]
Pitkin M., *Observation of the Monte Sainte Victoire Alvey grid DEM*, Alvey MMI-137 internal report M10D5PS, Department of Photogrammetry and Surveying, University College London.

[Polak, 1971]

[Polland et al, 1985]

[Porill et al, 1988]

[Powell, 1964]

[Press et al, 1988]

[Laserscan, 1987]

[Shannon, 1948]
[Snyder, 1987]

[Stroustrup, 1987]

[Tanenbaum, 1987]

[Tanre, 1979]

[Thomas, 1989]

[Thompson, 1979]

[Thompson, 1969]

[Tucker, 1978]
[Tyler and Preisendorfer, 1962]

[Rees and Muller, 1990]

[Richards, 1988]
Richards S., *the fly UNIX man page*, Alvey MMI-137 software documentation, Department of Photogrammetry and Surveying, University College London.

[Upton, 1990]

[Veillet, 1989]
Veillet I., *Background to Test on Triangulation of SPOT Data*, OEEPE Workshop on Block Triangulation of SPOT Data, Department of Photogrammetry and Surveying, September, 1989.

[Westin, 1990]

[Zemerly et al, 1991]
Appendix 1
Results of testing initial implementation
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta p_1, \delta \theta_2, \delta \phi_2, \delta p_2)\)

<table>
<thead>
<tr>
<th>Model</th>
<th>GCP Source</th>
<th>Co-ordinate System</th>
<th>RMS Plan Error (m)</th>
<th>RMS Height Error (m)</th>
<th>RMS Vector Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>South Yorkshire</td>
<td>Gugan</td>
<td>NG</td>
<td>12.55</td>
<td>9.68</td>
<td>12.60</td>
</tr>
<tr>
<td>Dorset</td>
<td>Odell</td>
<td>NG</td>
<td>26.60</td>
<td>23.26</td>
<td>23.26</td>
</tr>
<tr>
<td>Dorset</td>
<td>Odell [filtered]</td>
<td>NG</td>
<td>17.35</td>
<td>14.50</td>
<td>14.50</td>
</tr>
<tr>
<td>Cyprus</td>
<td>Gugan</td>
<td>UTM</td>
<td>24.44</td>
<td>9.54</td>
<td>22.51</td>
</tr>
<tr>
<td>Cyprus</td>
<td>Odell</td>
<td>UTM</td>
<td>82.04</td>
<td>14.88</td>
<td>13.22</td>
</tr>
<tr>
<td>Cyprus</td>
<td>Odell [filtered]</td>
<td>UTM</td>
<td>17.27</td>
<td>11.11</td>
<td>11.11</td>
</tr>
</tbody>
</table>

Table A1.1 Results of single model accuracy tests.
SPOT: South Yorkshire O'Neill-Dowman Camera Model

29/242 17/04/1987

RRSKEW cost function, parameter space is (80^2, 8^<t>, 8^/7)

key

- orn
- cpt

metres

0.0
6.0
12.0
18.0
24.0
30.0
36.0
42.0

eastings (false origin: 436000)

nortings (false origin: 322000)
SPOT: South Yorkshire O'Neill-Dowman Camera Model
29/242 17/04/1987HEIGHT

key
O orn
• cpt

RRSKEW cost function, parameter space is (80, 84, 81, 82, 84, 87)

northing (false origin: 324000)
easting (false origin: 441000)

0 5000 10000 15000 20000 25000 30000 35000 40000 45000 50000 55000 60000 65000 70000

0 5000 10000 15000 20000 25000 30000 35000 40000 45000 50000 55000 60000 65000 70000

0 6.0 12.0 18.0 24.0 30.0 36.0 42.0

277
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$

SPOT: Dorset O’Neill-Dowman Camera Model
28/247 17/04/1987 not shift pruned C.Helie HT)

SPOT: Dorset O’Neill-Dowman Camera Model
28/247 17/04/1987 not shift pruned C.Helie HT)

279
SPOT: Dorset O’Neill-Dowman Camera Model
28/247 17/04/1987 shift pruned CPiA

RSKuE cost function, parameter space is (ε, δ, μ, ν, θ).

key

- orn
- cpt

metres

- 35.0
- 30.0
- 25.0
- 20.0
- 15.0
- 10.0
- 5.0
- 0.0

northerings (false origin: 75000)
eastings (false origin: 324000)
SPOT: Dorset O’Neill-Dowman Camera Model
28/247 17/04/1987 shift pruned [HEIGHT]

RRSKSW cost function, parameter space is (69, 64, 67, 69, 69, 69)

northings (false origin: 45000)
eastings (false origin: 337000)

key
○ orn
● cpt

metres
56.0
48.0
40.0
32.0
24.0
16.0
8.0
0.0
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
SPOT: Cyprus O'Neill-Dowman Camera Model
Odell points - no shift Pruning

Key:
- orn
- cpt

Metres:
- 385.0
- 330.0
- 275.0
- 220.0
- 165.0
- 110.0
- 55.0
- 0.0

Northing (false origin: 3478000)
Eastings (false origin: 286000)
SPOT: Cyprus O’Neill-Dowman Camera Model
Odell points - no shift Pruning

RRSKEW cost function, parameter space is (80i, 5<|>i, 8p[, 802 » S < t> 2, 8 p l)
SPOT: Cyprus O’Neill-Dowman Camera Model
Odell points - with shift Pruning

eastings (false origin: 409000)
northings (false origin: 3804000)

metres

key

- cpt

- orn

RRSKEW cost function, parameter space is (θθ, φφ, αα, ββ, δδ, εε, ζζ, ηη)
SPOT: Cyprus O’Neill-Dowman Camera Model
Odell points - with shift pruning

RRSKEW cost function, parameter space is

(key

cpt

orn

metres

56.0

48.0

40.0

32.0

24.0

16.0

8.0

0.0

nortings (false origin: 3803000)
eastings (false origin: 431000)
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta p_1, \delta \theta_2, \delta \phi_2, \delta p_2)\)

<table>
<thead>
<tr>
<th>Ground Control Configuration</th>
<th>RMS Vector Error (m)</th>
<th>RMS Plan Error (m)</th>
<th>RMS Height Error (m)</th>
<th>Co-ordinate System</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>34.07 [102 points]</td>
<td>31.83</td>
<td>12.14</td>
<td>LZ3</td>
</tr>
<tr>
<td>A4</td>
<td>24.47 [93 points]</td>
<td>21.58</td>
<td>11.44</td>
<td>LZ3</td>
</tr>
<tr>
<td>A11</td>
<td>17.14 [94 points]</td>
<td>13.77</td>
<td>10.20</td>
<td>LZ3</td>
</tr>
<tr>
<td>A16</td>
<td>17.10 [93 points]</td>
<td>13.47</td>
<td>10.52</td>
<td>LZ3</td>
</tr>
</tbody>
</table>

Table A1.2 Results of OEEPE strip A [50/259-50/262] accuracy tests.
OEEPE strip A 50.259 - 50.262
Residuals, configuration 3 [PLAN]

key
- cpt
O orn

metres
- 154.0
- 132.0
- 110.0
- 88.0
- 66.0
- 44.0
- 22.0
- 0.0

norrhings (false origin: 90000)
eastings (false origin: 784000)

RRSKEW cost function, parameter space is: (89, 84, 97, 89, 84, 97)
OEEPE strip A 50.259 - 50.262
Residuals, configuration 3

key
- cpt
- orn

metres
- 161.0
- 138.0
- 115.0
- 92.0
- 69.0
- 46.0
- 23.0
- 0.0

eastings (false origin: 809000)
northings (false origin: 99000)
OEEPE strip A 50.259 - 50.262
Residuals, configuration 4 \text{HEIGHT}

RRSKEW cost function, parameter space is (50i,5\theta i,8p[, S02 ,6 \theta 2 ,5 /? 5

key
- cpt
- orn

metres
\begin{align*}
\text{154.0} \\
\text{132.0} \\
\text{110.0} \\
\text{88.0} \\
\text{66.0} \\
\text{44.0} \\
\text{22.0} \\
\text{0.0}
\end{align*}

northings (false origin: 1020000)
eastings (false origin: 808000)
OEEPE strip A 50.259 - 50.262
Residuals, configuration 11

RRSKEW cost function, parameter space is (80, 90, 89, 87, 89)

key
- cpt

orn

metres
133.0
114.0
95.0
76.0
57.0
38.0
19.0
0.0

northings (false origin: 1040000)
eastings (false origin: 7880000)
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1^2, \delta \theta_2, \delta \phi_2, \delta \rho_2^2)\)

<table>
<thead>
<tr>
<th>Ground Control Configuration</th>
<th>RMS Vector Error(m)</th>
<th>RMS Plan Error(m)</th>
<th>RMS Height Error(m)</th>
<th>Co-ordinate System</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>26.41 [130 points]</td>
<td>21.84</td>
<td>14.84</td>
<td>LZ3</td>
</tr>
<tr>
<td>B6</td>
<td>17.57 [130 points]</td>
<td>15.32</td>
<td>8.61</td>
<td>LZ3</td>
</tr>
<tr>
<td>B7</td>
<td>19.45 [130 points]</td>
<td>16.08</td>
<td>10.93</td>
<td>LZ3</td>
</tr>
<tr>
<td>B9</td>
<td>17.36 [120 points]</td>
<td>15.46</td>
<td>7.90</td>
<td>LZ3</td>
</tr>
</tbody>
</table>

Table A1.3 Results of OEEPE strip B [49/259-49/262] accuracy tests.
OEEPE strip B 49.259 - 49.262
residuals configuration 1 [PLAN]

RRSKewing cost function, parameter space is (89°, 66°, 89°, 82°, 82°, 82°)

key
- cpt
- orn

metres
- 357.0
- 306.0
- 255.0
- 204.0
- 153.0
- 102.0
- 51.0
- 0.0

northing (false origin: 95000)
easting (false origin: 722000)
OEEPE strip B 49.259 - 49.262
residuals configuration 1

RRSKEW cost function, parameter space is (50, 50) ≤ x, y ≤ 2

eastings (false origin: 741000)
northings (false origin: 91000)

key
• cpt
○ orn

metres
- 357.0
- 306.0
- 255.0
- 204.0
- 153.0
- 102.0
- 51.0
- 0.0
OEEPE strip B 49.259 - 49.262
residuals configuration 6c

RRSK EW cost function, parameter space is (501,5$,5! ,5pi, 5e2 ,5$2 ,5pl)

key
- cpt
- orn

metres
- 336.0
- 288.0
- 240.0
- 192.0
- 144.0
- 96.0
- 48.0
- 0.0

eastings (false origin: 741000)
northings (false origin: 102000)
OEEPE strip B 49.259 - 49.262
residuals configuration 7

RRSEKW cost function, parameter space is (801, 6 < J > 1, 5pt *)

northing (false origin: 98000)

eastings (false origin: 738000)

key
- cpt
O orn

metres
357.0
306.0
255.0
204.0
153.0
102.0
51.0
0.0

eastings (false origin: 738000)
OEEPE strip B 49.259 - 49.262
residuals configuration 7

eastings (false origin: 741000)
northings (false origin: 102000)

key
• cpt
○ orn

RRSKEW cost function, parameter space is (0.9, 0.9, 0.9, 0.9)

metres
336.0
288.0
240.0
192.0
144.0
96.0
48.0
0.0

eastings (false origin: 741000)
OEEPE strip B
configuration9, residuals [PLAN]

RRSEKW cost function, Parameter space is (x,y,z)

key
• cpt
○ orn

metres
357.0
306.0
255.0
204.0
153.0
102.0
51.0
0.0

northings (false origin: 91000)
eastings (false origin: 739000)
OEEPE strip B
configuration 9, residuals

RRSEW cost function, parameter space is (S_0, &h, &pi, &d_2, &e_1, &p_1)

key
- cpt
- orn

metres
- 336.0
- 288.0
- 240.0
- 192.0
- 144.0
- 96.0
- 48.0
- 0.0

northing (false origin: 101000)
easting (false origin: 741000)
Table A1.4 Results of Fast O'Neill-Dowman accuracy tests.

<table>
<thead>
<tr>
<th>Model</th>
<th>RMS Vector Error(m)</th>
<th>RMS Plan Error(m)</th>
<th>RMS Height Error(m)</th>
<th>Co-ordinate System</th>
</tr>
</thead>
<tbody>
<tr>
<td>South Yorkshire</td>
<td>13.96 (16 points)</td>
<td>12.14</td>
<td>6.90</td>
<td>NG</td>
</tr>
</tbody>
</table>
Appendix 2
Comparison of initial model with OEEPE workshop models
### Table A2.1. A comparison of the performance of the O'Neill-Dowman model with other SPOT-1 sensor models for OEEPE strip A.

<table>
<thead>
<tr>
<th>Software</th>
<th>GCP's</th>
<th>Check Points</th>
<th>Plan(m)</th>
<th>Height(m)</th>
<th>Vector(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGN</td>
<td>4</td>
<td>101</td>
<td>9.1</td>
<td>5.2</td>
<td>10.5</td>
</tr>
<tr>
<td>Ottawa</td>
<td>4</td>
<td>78</td>
<td>17.2</td>
<td>21.5</td>
<td>27.5</td>
</tr>
<tr>
<td>Hannover</td>
<td>2</td>
<td>88</td>
<td>14.5</td>
<td>8.9</td>
<td>17.0</td>
</tr>
<tr>
<td>Brisbane</td>
<td>2</td>
<td>106</td>
<td>14.2</td>
<td>10.8</td>
<td>17.8</td>
</tr>
<tr>
<td>Gugan</td>
<td>10</td>
<td>106</td>
<td>16.1</td>
<td>7.3</td>
<td>17.7</td>
</tr>
<tr>
<td>O'Neill</td>
<td>3</td>
<td>106</td>
<td>13.8</td>
<td>10.2</td>
<td>17.1</td>
</tr>
</tbody>
</table>

### Table A2.2. A comparison of the performance of the O'Neill-Dowman model with other SPOT-1 sensor models for OEEPE strip B.

<table>
<thead>
<tr>
<th>Software</th>
<th>GCPs</th>
<th>Check Points</th>
<th>Plan(m)</th>
<th>Height(m)</th>
<th>Vector(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGN</td>
<td>3</td>
<td>101</td>
<td>20.8</td>
<td>6.6</td>
<td>21.8</td>
</tr>
<tr>
<td>Ottawa</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hannover</td>
<td>3</td>
<td>124</td>
<td>17.6</td>
<td>5.7</td>
<td>18.5</td>
</tr>
<tr>
<td>Brisbane</td>
<td>3</td>
<td>122</td>
<td>15.6</td>
<td>3.8</td>
<td>16.1</td>
</tr>
<tr>
<td>Gugan</td>
<td>6</td>
<td>135</td>
<td>14.1</td>
<td>8.9</td>
<td>16.6</td>
</tr>
<tr>
<td>O'Neill</td>
<td>3</td>
<td>120</td>
<td>14.8</td>
<td>7.7</td>
<td>16.7</td>
</tr>
</tbody>
</table>

306
Appendix 3
Results of testing LSL/RAE implementation of model
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta p_1, \delta \theta_2, \delta \phi_2, \delta p_2)\)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute [98 vectors]</td>
<td>15.51</td>
<td>13.99</td>
<td>6.72</td>
</tr>
<tr>
<td>Skew [98 vectors]</td>
<td>9.64</td>
<td>7.04</td>
<td>6.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space [98 vectors]</td>
<td>1.78</td>
<td>1.61</td>
</tr>
</tbody>
</table>

A 3.1 Table showing sensor model accuracy for ground control configuration B9.
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta p_1, \delta \theta_2, \delta \phi_2, \delta p_2)$
RRSKEW cost function, parameter space is \((\delta \phi_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
fid: skew.vec.9.plot

OEEPE strip B SPOT1 PAN camera model residuals [plan]

Relative ray-ray skew RMSE

RRSEKFW cost function, parameter space is \((a_0, a_1, a_2, a_3)\).
fid: skew.vec.9.plot
OEEPE strip B SPOT1 PAN camera model residuals [ht]
Relative ray-ray skew RMSE
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$
fid: lim.vec.9.plot

OEEPE strip B SPOT1 PAN camera model residuals [plan]
left image

RRSKEW cost function, parameter space is \( \mathcal{O} \), \( \mathcal{P} \), \( \mathcal{S} \), \( \mathcal{V} \), \( \mathcal{X} \), \( \mathcal{Y} \), \( \mathcal{Z} \).
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi, \delta p^1_1, \delta \theta_2, \delta \phi_2, \delta p^2_2)\)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute [104 vectors]</td>
<td>16.14</td>
<td>12.83</td>
<td>9.97</td>
</tr>
<tr>
<td>Skew [104 vectors]</td>
<td>12.97</td>
<td>9.57</td>
<td>8.72</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space [104 vectors]</td>
<td>1.59</td>
<td>1.74</td>
</tr>
</tbody>
</table>

A 3.2 Table showing sensor model accuracy for ground control configuration A15.
fid: nmodel.4scenes.tests/abs.vec.15.plot
OEEPE strip A SPOT1 PAN camera model residuals [plan]

Absolute RMSE

RRSKEW cost function, parameter space is $802.5^2 \times 8$.5

key
O era
- cpt

metres

316
Absolute RMSE

RRSKEW cost function, parameter space is (a1, a2, b1, b2, b3, b4)
Relative ray-ray skew RMSE

fid: nmodel.4scenes.tests/skew.vec.l5.plot

OEEPE strip A SPOT1 PAN camera model residuals [plan]
Relative ray-ray skew RMSE
Relative ray-ray skew RMSE

Key:
- O: original
- ▲: corrected

Parameter space is (80°, 84°, 88°, 92°, 96°, 100°, 104°).
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
RRSKEW cost function, parameter space is $\left( \delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2 \right)$

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute [18 vectors]</td>
<td>23.65</td>
<td>20.26</td>
<td>10.14</td>
</tr>
<tr>
<td>Skew [18 vectors]</td>
<td>8.07</td>
<td>4.02</td>
<td>5.79</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space [18 vectors]</td>
<td>1.93</td>
<td>2.09</td>
</tr>
</tbody>
</table>

A 3.3 Table showing sensor model accuracy for Cyprus ground control configuration1.
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho^+_1, \delta \theta_2, \delta \phi_2, \delta \rho^+_2)$
fid: abs.vec.1.plot
Cyprus SPOT1 PAN camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is $(\theta_1, \theta_2, \theta_3, \theta_4)$
fid: skew.vec.1.plot
Cyprus SPOT1 PAN camera model residuals [plan]
Relative ray-ray skew RMSE

![Graph showing relative ray-ray skew RMSE with annotations and data points.]

**RRSKEW cost function, parameter space is:**

(68°, 58°, 57°, 88°, 68°, 57°)
Cyprus SPOT1 PAN camera model residuals [plan]
Relative ray-ray skew RMSE

fid: skew.vec.1.plot

RRSKEW cost function, parameter space is (\(\theta_1, \theta_2, \phi_1, \phi_2, \phi_3, \phi_4\))

key

- CRM
- CPE
Cyprus SPOT1 PAN camera model residuals [plan]
right image

RRSKEW cost function, parameter space is \((\theta_y, \phi_y, \theta^\gamma, \phi^\gamma, \delta)\)
fid: lim.vec.1.plot
Cyprus SPOT1 PAN camera model residuals [plan]
left image

RRSKEW cost function, parameter space is $(60^2, 60^2, 5^2, 6^0, 8^0, 8^0)$
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \theta_2, \delta \phi_1, \delta \phi_2, \delta \rho_1, \delta \rho_2)\)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skew [21 vectors]</td>
<td>7.20</td>
<td>5.74</td>
<td>4.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space</td>
<td>1.41</td>
<td>1.08</td>
</tr>
</tbody>
</table>

A 3.4 Table showing sensor model accuracy for Dorset ground control configuration.
fid: abs.vec.1.plot
Dorset SPOT1 PAN camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is \((86, 89, 87, 88, 84, 89, 72)\)
fid: abs.vec.1.plot
Dorset SPOT1 PAN camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is (66, 84, 517, 89, 59)
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
fid: rim.vec.1.plot
Dorset SPOT1 PAN camera model residuals [plan]
right image

RRSEW cost function, parameter space is $[56, 5, 5, 17, 85, 80, 80, 80, 82]$. 

key
- o rm
- c pt
fid: lim.vec.1.plot
Dorset SPOT1 PAN camera model residuals [plan]
left image

RRSKEW cost function, parameter space is (\theta_1, \theta_2, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5)
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>9.12</td>
<td>8.36</td>
<td>5.51</td>
</tr>
<tr>
<td>[13 vectors]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skew</td>
<td>10.27</td>
<td>3.13</td>
<td>8.89</td>
</tr>
<tr>
<td>[13 vectors]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space</td>
<td>0.82</td>
<td>0.99</td>
</tr>
<tr>
<td>[13 vectors]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A 3.5 Table showing sensor model accuracy for Oman ground control configuration.
fid: abs.vec.1.plot
Oman SPOT1 PAN camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is \((0, \theta_4, \theta_7), (0, \theta_3, \theta_7)\)
fid: abs.vec.1.plot
Oman SPOT1 PAN camera model residuals [ht]
Absolute RMSE
fid: skew.vec.1.plot
Oman SPOT1 PAN camera model residuals [plan]
Relative ray-ray skew RMSE

-1090000.0
-1080000.0
-1070000.0
-1060000.0
-1050000.0
-1040000.0
-1030000.0
-1020000.0
-1010000.0
-1000000.0
-990000.0
-980000.0
-970000.0
-960000.0
-950000.0
-940000.0
-930000.0
-920000.0
-910000.0
-900000.0
-890000.0
-880000.0
-870000.0
-860000.0
-850000.0
-840000.0
-830000.0
-820000.0
-810000.0
-800000.0
-790000.0
-780000.0
-770000.0
-760000.0
-750000.0
-740000.0
-730000.0
-720000.0
-710000.0
-700000.0
-690000.0
-680000.0
-670000.0
-660000.0
-650000.0
-640000.0
-630000.0
-620000.0
-610000.0
-600000.0
-590000.0
-580000.0
-570000.0
-560000.0
-550000.0
-540000.0
-530000.0
-520000.0
-510000.0
-500000.0
-490000.0
-480000.0
-470000.0
-460000.0
-450000.0
-440000.0
-430000.0
-420000.0
-410000.0
-400000.0
-390000.0
-380000.0
-370000.0
-360000.0
-350000.0
-340000.0
-330000.0
-320000.0
-310000.0
-300000.0
-290000.0
-280000.0
-270000.0
-260000.0
-250000.0
-240000.0
-230000.0
-220000.0
-210000.0
-200000.0
-190000.0
-180000.0
-170000.0
-160000.0
-150000.0
-140000.0
-130000.0
-120000.0
-110000.0
-100000.0
-90000.0
-80000.0
-70000.0
-60000.0
-50000.0
-40000.0
-30000.0
-20000.0
-10000.0
0

 eastings UTM

339
fid: skew.vec.1.plot
Oman SPOT1 PAN camera model residuals [ht]
Relative ray-ray skew RMSE
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta p_1^e, \delta \theta_2, \delta \phi_2, \delta p_2^e)$.
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>14.17</td>
<td>12.58</td>
<td>6.51</td>
</tr>
<tr>
<td>[98 vectors]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skew</td>
<td>8.40</td>
<td>6.12</td>
<td>5.66</td>
</tr>
<tr>
<td>[98 vectors]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space</td>
<td>1.57</td>
<td>1.59</td>
</tr>
<tr>
<td>[98 vectors]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A Table showing sensor model accuracy for ground control configuration B9b.
fid: nmodel.nrelax.4scenes.vam.tests/abs.vec.9.plot
OEEPE strip B SPOT1 PAN camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is (69,64,17,51,50)
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \gamma_1, \delta \theta_2, \delta \phi_2, \delta \gamma_2)$.
Relative ray-ray skew RMSE

RRSKEW cost function, parameter space is $(\theta_0^{\phi}, \phi, \delta, \gamma_1, \theta_2^{\phi}, \delta_2)$
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta \rho_2, \delta \theta_2, \delta \phi_2, \delta \rho_2)\)
RRSKEW cost function, parameter space is (501, 5 < |p| < 1, 502^2/7)

fid: nmodel.nrelax.4scenes.vam.tests/rim.vec.9.plot
OEEPE strip B SPOT1 PAN camera model residuals [plan]
right image
fid: nmodel.nrelax.4scenes.vam.tests/lim.vec.9.plot

OEEPE strip B SPOT1 PAN camera model residuals [plan]
left image

RRSKEW cost function, parameter space is (89,90)
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute [104 vectors]</td>
<td>17.35</td>
<td>15.06</td>
<td>8.77</td>
</tr>
<tr>
<td>Skew [104 vectors]</td>
<td>12.96</td>
<td>9.51</td>
<td>8.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space</td>
<td>1.84</td>
<td>1.69</td>
</tr>
</tbody>
</table>

Table showing sensor model accuracy for ground control configuration A4.
fid: nmodel.nrelax.4scenes.1orn.tests/abs.vec.4.plot
OEEPE strip A SPOT1 PAN camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is \((\delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7)\)
Absolute RMSE

RRSKEW cost function, parameter space is \((86, 86, 86, 89.2, 89.2, 97.7)\)
fid: nmodel.nrelax.4scenes.1orn.tests/skew.vec.4.plot
OEEPE strip A SPOT1 PAN camera model residuals [plan]
Relative ray-ray skew RMSE

RRSKEW cost function, parameter space is (9λ, 0.84, 0.2, 0.8, 0.8, 0.84, 0.2)
fid: nmodel.nrelax.4scenes.1orn.tests/skew.vec.4.plot
OEEPE strip A SPOT1 PAN camera model residuals [ht]
Relative ray-ray skew RMSE
RRSKEW cost function, parameter space is \((\delta \theta_1, \delta \phi_1, \delta p_1, \delta \theta_2, \delta \phi_2, \delta p_2)\)
fid: nmodel.nrelax.4scenes.1orn.tests/lim.vec.4.plot

OEEPE strip A SPOT1 PAN camera model residuals [plan]

left image
RRSKEW cost function, parameter space is \((\delta\theta_1, \delta\phi_1, \delta\rho_1, \delta\theta_2, \delta\phi_2, \delta\rho_2)\)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Vector Error(m)</th>
<th>Plan Error(m)</th>
<th>Height Error(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute [12 vectors]</td>
<td>51.57</td>
<td>29.13</td>
<td>35.55</td>
</tr>
<tr>
<td>Skew [12 vectors]</td>
<td>102.27</td>
<td>73.77</td>
<td>69.35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Left Image Vector(pixels)</th>
<th>Right Image Vector(pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Space [12 vectors]</td>
<td>2.57</td>
<td>3.35</td>
</tr>
</tbody>
</table>

A 3.8 Table showing sensor model accuracy for Wyoming XS ground control configuration 1.
fid: abs.vec.1.plot
Wyoming SPOT1 XS camera model residuals [plan]
Absolute RMSE

RRSKEW cost function, parameter space is (98, 84, 97, 82, 86, 92)
RRSKEW cost function, parameter space is $(\delta \theta, \delta \phi, \delta p_1, \delta \theta_2, \delta \phi_2, \delta p_2)$
fid: skew.vec.1.plot
Wyoming SPOT1 XS camera model residuals [plan]
Relative ray-ray skew RMSE
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$
RRSKEW cost function, parameter space is $(\delta \theta_1, \delta \phi_1, \delta \rho_1, \delta \theta_2, \delta \phi_2, \delta \rho_2)$.
Wyoming SPOT1 XS camera model residuals [plan]
left image

RRSKEW cost function, parameter space is (88,80,87,88,82,80,92,97)
Appendix 4
Software Implementation

A4.1 Introduction.

It is unusual in a thesis of this nature to enter into lengthy discussion about the nature of the software implementation. Traditionally, academic research projects have concentrated on researching and developing algorithms, with the subsequent software implementation being dismissed as an uninteresting, though necessary technical task. This philosophy is very dangerous as its implicit assertion is that software implementation is a task which may be accomplished without much thought or planning. This has led to a very unsatisfactory situation in which excellent algorithms developed under the aegis of university research projects have been rendered virtually useless by poor or indifferent software implementations.

For large and complex systems, such as the O'Neill-Dowman Camera Model, and indeed the UCL Geodem system as a whole, it is clear that the tasks of algorithm identification, development and software implementation are equal partners. Carrying this analysis a little further, it is clear that algorithm selection and software implementation are not isolated tasks. The efficiency of a given algorithm is critically dependent on the programming language in which it is coded, and indeed on the hardware on which it is run.

A4.1.1 Development methodology.

The implementation of the O'Neill-Dowman Camera Model system is a compromise. Although attention has been given to algorithmic efficiency, effort has also been expended in making the code easy to port between different data-processing systems. Since the Camera Modeling System is complex, the code has been implemented in adherence to strict software engineering techniques. These include:
a) Subdivision of the code into small, easily readable subroutines.
b) Use of object code libraries.
c) Use of intelligent compilation techniques [Feldman, 1979]
d) Use of comments within the program sources in order to make the program sources self-documenting.

The only major deviation from established software engineering practice was the failure to use an adequate revision control system such as the UNIX SCCS [Source code Control System] or the MINIX RCS [Revision Control System]. The use of such a system would have reduced the time required to develop the code as it enables the programmer to back-track to previous versions of a program when bugs are inadvertently introduced into the current version of the program.

A4.1.2 Choice of implementation language.

The Alvey MMI-137 project standardised on the C programming language. The Camera Model, developed under the aegis of this project is coded in C. Initially, the programs where coded in the dialect of C defined in the first edition of *The C Programming Language* [Kernighan and Ritchie, 1978]. Later, when suitable compilers became available, the system was re-coded to the ANSI standard of the C language. From the software engineering point of view, the ANSI C language represents a considerable advance over the original C implementation. In particular, the language promotes the use a strict typing mechanism for function and procedure arguments which is similar to that used by the Pascal programming language. The ANSI version of the C programming language is described by Kernighan and Ritchie [Kernighan and Ritchie, 1988].

The decision by the Alvey MMI-137 consortium to standardise on the C programming language was a result of two influences:

a) One of the consortiums partners, the Department of Computer Science at UCL already had considerable experience with the C programming language because of its standardisation on the C/UNIX environment for its computer systems.
b) The SERC IT initiative at the time of the start of the Alvey MMI-137 project in July 1986 supported the use of C/UNIX based workstations such as the ICL-PERQ and later the Sun series of workstations.

Implementation of scientific software within the C/UNIX environment was possesses many advantages compared to traditional scientific environments, which are based on FORTRAN. These include:

a) The support for modular compilation and object code libraries in C is excellent.

b) Because C is a block structured language, it is an efficient vehicle for the production of clear readable code. Although it is not nearly so good in this respect as Pascal or Modula-2 because rules enforced by these languages are left to the discretion of the programmer in C.

c) The UNIX part of the C/UNIX environment provides abstractions such as pipes and sockets, which are easily accessible from C, and which facilitate the development of distributed systems. The easy development of distributed systems becomes an important considerations in the case of applications like the SPOT-1 Camera Model, where the processing load is high.

The disadvantages of using C for scientific programming arise largely because of its relatively recent arrival to the scientific programming arena. Compared to FORTRAN, C has little built-in support for abstractions used by scientific programmers. In the case of the Camera Model for example, all the mathematical support libraries had to be explicitly programmed. At the beginning of the Alvey MMI-137 project, there was also concern about the efficiency of the object code produced by the then current C-compilers. With the introduction of efficient, optimising compilers such as the Free Software Foundations GNU series of C compilers, these concerns have now evaporated. However, the efficient vectorisation or parallelisation of C code, which would permit near optimal matrix operations to be programmed on machines such as the Intel INS80860, the Cray X-MP supercomputer series,
and other forms of vector processors or MIMD/SIMD arrays is still an issue. Efficient parallel/vectorising FORTRAN systems have been in existence for the ICL DAP and Cray machines respectively for a number of years.

A4.1.3 Advantages of using C++ for system development.

The object oriented extension to C, C++ [Stroustrup, 1987, Lippman, 1989] overcomes many problems which arise when using standard C. For example, the operator overloading abstraction provides an efficient mechanism for the provision of a consistent interface to specialist mathematical libraries. C++ supports software engineering methodologies better than C, as it provides strong type checking mechanisms, data hiding, and generic datatype [class] abstractions. Reports indicate that programmer productivity using C++ is greater than that achieved using C. Experience with C++ within the Department of Photogrammetry and Surveying at UCL implies that recoding of the SPOT-1 Camera Model in this language would result in a desirable enhancement to the software engineering standard of the model.

One of the original goals of the Alvey MMI-137 project was the development of a generic Camera Modeling System, which transparently incorporated a whole series of camera models including SPOT-1, Landsat Thematic Mapper, and also various aerial photographic camera models. One approach to the construction of such a system is to provide a set of atomic functions at process level, and build dataflow models which consist of pipelines of atomic processes which achieve the desired functionality. Systems of this sort already exist, often with iconic interfaces [Boddington, 1991]. While this approach has the advantage of being interactive, it is not well suited to small single processor systems where the overheads of scheduling multiple communicating processes may be prohibitively high. For these systems, the alternative methodology of the software-ic becomes attractive. C++ with its support for data-hiding, and object orientation would be an attractive vehicle with which to implement systems of this type.

A4.1.4 Limitations of the current implementation.
Despite the attempts which have been made to make the Camera Model both portable and efficient, the efficiency of the current implementation is system dependent. Because the production of digital elevation models involves the transformation of large quantities of data, the Camera Modeler has been implemented so that the code may be readily ported to coarse grained parallel MIMD architectures, such as the Parsys Supernode. The system is therefore written as a set of loosely coupled tasks which execute as separate processes on a multitasking or multiprocessing system, and communicate by file like FIFO abstractions called pipes. This limits efficient implementation of the Camera Modeling system to machines which run the UNIX operating system, or a derivative such as XENIX, ULTRIX, IDRIS or HELIOS. Implementation experiments have shown that ports to machines running radically different operating systems such as the IBM-3084 [running TSO/Phoenix] are possible, but at some detriment to the overall performance of the system, as disk-based interprocess communication has to be substituted for kernel-based interprocess pipes on such systems.

A4.2 Camera Model Implementation Overview.

The O’Neill-Dowman Camera Model is currently implemented in the ANSI dialect of the C programming language on a number of different computer systems. The program is highly modular, and in order to facilitate porting to new machines, the program is based on a set of support libraries, The Portable UNIX Programming System, [PUPS] which was implemented prior to the camera model as part of the GPROC system [O’Neill, 1988b].

The camera model is implemented as a quaternary structure, consisting of four distinct layers. Taking a top down approach these layers are:

- The UNIX filter or process layer. This layer consists of the Camera Model and its support filters.
- The subroutines which are specific to the Camera Model.
Software Implementation

- Low level utilities provided by the [UNIX] operating system.

Schematics of the structure of the Camera Model and its support filters are shown in Figure A4.1 and A4.2. The role of the SPOT-1 Camera Model and its support filters in the automated DEM production process is shown in Figure A4.3. We will look at each layer of the software implementation in turn:

A4.3 The Top Level of the System.

The top level of the system consists of the SPOT-1 Camera Model and its supporting UNIX filters. A UNIX filter is a program running on a UNIX system, generally written in C. The filter reads input data from the standard input stream which is generally connected to the user console, but which may be redirected to either a file or a pipe via the UNIX command interpreter, the shell. The filter writes output data to the data stream standard output, like the standard input stream, the standard output stream is generally connected to the user console although it too may be redirected to a file or pipe by the shell. A full description of the UNIX terms described above may be found in elementary texts describing the UNIX operating system, for example *The UNIX System*, [Bourne, 1983].

A4.4 The Top Level Organisation of the Camera Model.

The top level of the Camera Modelling system consists of the following filters:

A4.4.1 The Camera Model filter.

This filter sets up a SPOT-1 Camera Model using four or less ground control points, and the SPOT-1 header files [CNES, 1987] corresponding to two or more SPOT-1 looks. The program uses this information to compute either the space intersection of a pair of conjugate points in image space, or alternatively to back project a point in object space into the image space of one of the camera positions or looks.
Generic sensor model libraries

Intermediate level executive containing high level function calls which are written in terms of the lower level libraries of the C scientific programming environment, and the low level C and UNIX libraries.

PORTABLE UNIX PROGRAMMING SYSTEM [PUPS]

Lower level routines which support the following operations:
- Linear programming
- Numerical analysis
- Stochastic programming
- Utility functions
- Image processing (HIPS) functions
- RPC based interprocess communications
- Window based graphic tools

UNIX system interface

UNIX and C libraries: Low level bindings to operating system and C [ANSI] supported utility functions

Figure A.4.1 Showing the structure of a typical filter within the O'Neill-Dowman Camera Modelling system.
Figure A4.2 Showing the software environment of the O'Neill-Dowman Camera Modelling system.
A4.4.2 The co-ordinate system interconversion filters.

The following set of filters were implemented in the UNIX environment by the author to support the Camera Model by providing a standard set of transformations between the geocentric co-ordinate system used by the Camera Model, and an appropriate local co-ordinate system. A description of the algorithms used is given by Snyder [Snyder, 1987]. In addition a set of filters to convert between pixel and plate image co-ordinates has been coded.

The filters are documented in the on-line UNIX man page system. These man pages are reproduced in Appendix 5.

`gclz3/lz3gc:`

This pair of filters transforms points in object space between the geocentric co-ordinate system to the Lambert Zone 3, [French Lambert] local co-ordinate system. These filters are based on earlier code written by Gugan [Gugan, 1987].

`gcgeo/geogc:`

This pair of filters transforms points in object space between the geocentric co-ordinate system and the geodetic co-ordinate system [latitude, longitude, height]. These filters are based on earlier code written by Gugan [Gugan, 1987].

`gcutm/utmgc:`

This pair of filters transforms points in object space between the geocentric co-ordinate system and either the UTM [Universal Transverse Mercator] co-ordinate system, or the NG [National Grid] which is a modified UTM co-ordinate system [Ordnance Survey, 1983]. These filters are based on earlier code written by Gugan [Gugan, 1987].
pixplate/platepix:

This pair of filters transforms image points between the plate or fiducial image co-ordinate system and the pixel image co-ordinate system. These filters are based on earlier code written by Gugan, [Gugan, 1987].

gproc/vec:

These sunview filters are used to interactively view output from the Camera Model. They were developed to look at test mode output from the Camera Modeller which is used to assess the quality of a given camera model. The vec and gproc filters are described in detail by O'Neill [O'Neill, 1988b; O'Neill, 1989b].

A4.4.3 The distributed programming support filters.

The UCL Geodem system is a large system. Given the limitations of current CPU technologies, it is necessary to distribute applications such as the Stereo Matcher, the Camera Modeller and the DEM interpolator [Kriger] over a network of compute servers. The parallelisation scheme adopted for all the components of the UCL Geodem system is essentially coarse grained. This means that whole programs are allocated to given processors on the network. In the alternative fine grained parallelisation scheme, the parallelism is inherent in the programming language, with the result that different portions of a given program may run transparently on different compute servers. This situation is typified by applications written in the Occam programming language and run on transputer arrays. Another example of a programming language which permits fine grain parallelisation is the 3M Parallel C compiler, which again runs on transputer based hardware.

Although simpler than fine grained schemes, there are many problems which must be solved before schemes employing coarse grained parallelism can operate efficiently. These include:

a) Resource management: The load of work to be done must be balanced across all available compute servers if the application is to perform optimally. Mechanisms must be provided which
monitor the current load on the network, and then \textit{balance} the current process load so that the loading of any one compute server does not become unacceptably high. In high data-rate applications, such as the UCL Geodem system, mechanisms to monitor disc usage which respond appropriately when the target disc server is full are also desirable.

b) Extended process control: A mechanism to kill processes on machine \(a\), from a remote machine \(b\).

c) Fault tolerance: In a multiprocessor environment, a mechanism which makes the system tolerant to faults [such as a compute server going down] is desirable. This fault avoidance mechanism can be either built into the component programs [by making them restartable] or a function of the network [work that was being done by the defunct server is redistributed to another server].

The \textit{Virtual Machine Shell}, \texttt{vsh} provides much of the functionality required for coarse grained parallelism within MIMD environments. Specifically, \texttt{vsh} provides the following services:

- A mechanism to transparently execute processes on the compute-server which currently has the lowest loading [\texttt{vsh} considers memory utilisation, process loading and swap efficiency when computing the loading].

- A mechanism for executing heterogeneous and homogeneous process farms. A heterogeneous farm consists of a number of \textit{different} processes running in parallel; a homogeneous farm consists of several instances of \textit{the same} process running in parallel.

- A mechanism for building cross machine pipelines. In a cross machine pipeline, the processes which constitute the pipeline are executing on different compute servers.
A4.5 The Portable UNIX Programming System.

A4.5.1 Introduction.

The Portable UNIX Programming System [PUPS] consists of a set of subroutine libraries which were developed to facilitate the modular programming of scientific and engineering applications such as the O'Neill-Dowman Camera Model.

The Portable UNIX Programming System has assisted in porting the O'Neill-Dowman Camera Model to a number of different computer systems, by providing a standard machine independent environment. To date the O'Neill-Dowman Camera Model has been ported to the following platforms:

- Sun-3 workstation; ANSI [GNU C 1.39] running under Sun OS 4.0 [based on BSD 4.3 UNIX and AT&T System V UNIX].
- Sun-4 workstation; ANSI [GNU C 1.39] running under Sun OS 4.0.
- Atari 1040STF personal computer; ANSI [GNU C 1.36] C running under MINIX 1.5 [Tanenbaum, 1987]. The functionality of MINIX is based on that of AT&T System 7 UNIX.
- IBM 3084Q mainframe; Norcroft [ANSI] C running under MVS/TSO/Phoenix-3.
- Acorn R140 workstation; Norcroft [ANSI] C running under RISCIX 1.15 which is a port of BSD 4.3 UNIX.
- Silicon Graphics 3G/440 workstation; ANSI [GNU C 1.39] running under AT&T System V UNIX.
- Cray Research Cray X-MP/28 Supercomputer; ANSI [Cray C] running under Unicos, which is a port of AT&T system V UNIX.

There are currently 7 support libraries in the PUPS environment. Single line descriptions of all the subroutines provided are given below.
A4.5.2 The utilities library - utilib.

The utilities library provides a number of general utility functions which will be useful in almost any application. In addition, the library provides a comprehensive mechanism for decoding items from the command tail of a UNIX filter.

Functionality of the utilities library.

The standard command tail decoding functions.

1. std_init: search for standard items in command tail.
2. ch_dec: extract char from command tail.
3. str_dec: extract string from command tail.
4. i_dec: extract integer from the command tail.
5. fp_dec: extract floating point number from the command tail.
6. copy_tail: copy command tail to user specified storage area.
7. argfile: insert a set of arguments from a file into command tail.
8. t_arg_errs: check that all arguments read in on the command tail have been successfully parsed and list those which have not.
9. help: display relevant man page for current application.

File utility functions.

1. xfopen: Check for the existence of a level 2 file, open it if it exists, otherwise terminate calling process, printing a suitable error message.
2. xopen: Check for the existence of a level 1 file. Open it if it exists, otherwise terminate calling process, printing a suitable error message.
3. strp_cmmnts: Strip comment lines prepended by a user defined
character token from a level 2 file.
4. lock: Get discretionary file lock.
6. unlock: Release discretionary file lock.
7. flock: Get discretionary stream lock.
8. funlock: Release discretionary stream lock.

Standard string functions.

1. strncmps: Test for existence of substring.
2. strext: Extract substring.
3. strccpy: Copy string checking for null argument pointer in the case of the "from" string.
4. strhat: Divide a string into head and tail substrings at selected demarcation character.

Memory allocation functions.

1. xmalloc: Allocate a sequence of bytes, checking that memory was actually allocated.
2. xcalloc: Allocate a sequence of elements of a given type, checking that the elements were actually allocated.
3. xrealloc: Reallocate a sequence of bytes to a pre-existing datastructure, checking that memory could actually be allocated.
4. xfree: Release memory, checking that memory had been previously allocated.

General purpose utility functions.

1. error: Standard error handler.
2. actoi: Convert character to integer.
3. ieven: Test for even integer.
4. iodd: Test for odd integer.
5. chsign: Return sign of character.
6. pause: Pause routine [used in debugging].
7. imax: Find maximum of pair of integers.
8. fmax: Find maximum of pair of floats.
9. imin: Find minimum of pair of integers.
10. fmin: Find minimum of a pair of floats
11. isign: Find sign of integer.
12. fsign: Find sign of float.
13. sqr: Pascal compatible squaring routine.
14. round: Pascal compatible rounding function.
15. apsin: Approximate sine using Taylor series.
17. execls: overlay current process with a command which is given in the form of a string.
18. new_thread: create a new thread of control within an application.

A4.5.3 The stochastic programming library - casinolib.

The stochastic function library, casinolib provides a number of functions for generating random, Gaussian, gamma, binomial, Poisson and user defined distributions. The random number generators provided are superior to those supplied as part of the standard UNIX library: The period to all intents and purposes is practically infinite, and there [ought] to be no sensible correlations between successive elements in the sequence.

The stochastic programming library was implemented to support stochastic minimisation using genetic search and simulated annealing.

Functionality of the stochastic programming library.

1. ran1: Return precision random deviate [Linear congruential method].
2. ran2: Return reduced precision random deviate [Linear congruential method].
3. ran3: Return random deviate using Knuths' method
4. gasdev: Return Gaussian deviate (using Box-Muller transform).
5. gasdev: Return Gaussian deviate (using central limit theorem).
6. gammln: Find log of gamma function.
7. gamdev: Return gamma deviates (using substitution method).
8. poidev: Return Poisson deviate (using substitution method).
9. bnldev: Return binomial deviates (using substitution method).
10. numdev: Return user defined deviates (using substitution method).

A4.5.4 Numerical functions library - nfolib.

The numerical functions library supplies a number of useful routines which operate on numerical functions. Routines are provided to perform least squares regression, interpolate, integrate, differentiate and find the extrema of numerical functions.

Functionality of the numerical functions library.

1. least_squares: Generate least squares fitting parameters. Currently supports linear, inverse, power, logarithmic and exponential regressions.
2. spline: Cubic spline fit parameterisation function.
4. linmin: Linear minimisation support routine for the Powell minimiser.
5. mnbrak: Routine to bracket a parabolic minimum bracketer.
7. anneal: Multivariate annealing minimiser.
8. lsq_fgen: Generate numerical function point using parameters generated by the least squares function.
9. lint: Linearly interpolate numerical function.
10. splint: Interpolate numerical function using cubic spline.
11. trapez: Integrate numerical function using the trapezium rule.
15. golden: Perform a linear minimisation using the Golden Search method.

A4.5.5 The linear programming library - veclib.

The linear programming library is a fairly complete support library for linear programming written in C. Currently, the library supports an extensive set of operations on both vectors and matrices, including several different methods for solving systems of linear equations, matrix multiplication, scalar products, vector products etc. The linear programming library was implemented to provide a standard environment to support the geometric aspects of the O’Neill-Dowman Camera Model.

Functionality of the linear programming library.

Standard vector operations.

1. vtomac: Convert vector to column of matrix.
2. vinv: Invert the sign of a vectors components.
3. vadd: Add a pair of vectors.
4. vsub: Subtract a pair of vectors.
5. vcross: Subtract a pair of vectors.
6. vunit: Find the unit vector in the direction of a given vector.
7. vsign: Return the component sign vector of a given vector.
8. vscaln: Multiply vector by scalar.
9. vscald: Divide vector by scalar.
10. vplanp: Return a pair of vectors which are mutually perpendicular.
ular to each other and to the argument vector.

11. vurnd: Return a unit vector whose direction is random.
12. vrotx: Rotate vector about X axis of space.
13. vroty: Rotate vector about Y axis of space.
14. vrotxz: Rotate vector about Z axis of space.
15. vifota: Rotate vector about an arbitrary axis.
16. vmult: Find the product of a pair of vectors.
17. vpointv: Find the shortest vector between a line in the direction of the argument vector and a given point in space.
18. vlinesv: Find the shortest vector between a pair of lines in the direction of the argument vectors.
19. vlinesv: Return the vector which is the mean vector between a series of line-pairs.
20. vLU_solve: Solve a set of linear equations using the LU decomposition method.
21. vGE_solve: Solve a set of linear equations using Gaussian Elimination.
23. vmag: Return the magnitude of a vector.
24. vdot: Return the scalar product of a pair of vectors.
25. vang: Return the smallest angle between a pair of vectors.
26. vpar: Return the Boolean TRUE if the vectors are parallel.
27. veq: Return the Boolean TRUE if the vectors are identical.
28. vsizes: Return the Boolean TRUE if the vectors have the same number of dimensions.
29. vread: Read a vector from a level 2 file.
30. vass: Assign values to the components of a vector.
31. vprint: Print the components of a vector on console [debugging aid].
32. vwrite: Write vector to level 2 file.

Standard matrix operations.
1. msquare: Return the Boolean TRUE if matrix is square.
2. msym: Return the Boolean TRUE if matrix is symmetric.
3. meq: Return the Boolean TRUE if a pair of matrices are identical.
4. mass: Assign components of matrix.
5. mzero: Initialise matrix as the zero matrix.
6. mident: Initialise matrix as the identity matrix.
7. mprint: Print the components of matrix on console [debugging aid].
8. mdet2: Find determinant of 2x2 matrix by expansion of minors.
9. mdet3: Find determinant of 3x3 matrix by expansion of minors.
10. mmadd: Find the sum of a pair of matrices.
11. mmsub: Find the difference of a pair of matrices.
12. mmmult: Find the product of a pair of matrices.
13. mscaln: Multiply matrix by scalar.
14. meuler: Compute Euler matrix.
15. mbint: Linearly interpolate matrix function.
16. minv: Invert matrix using LU decomposition.
17. madj3: Find the adjoint of a 3x3 matrix.

A4.5.6 The network communications library - netlib.

The Network Library provides a set of routines which facilitate distributed programming of applications over a number of nominated compute and disk servers connected by a suitable network and associated communications protocol, for example the Ethernet [Metcalfe and Boggs, 1976] running either the TCP/IP or Amoeba [Mullender et al, 1990]. The system provides a number of handles for coarse grained parallelisation of applications such as the Camera Modeling system. The programming model adopted by the library is that of the co-operating process cluster, [O'Neil, 1987] in which a task to be accomplished by the system is carried out by a number of co-operating processes which may be located on different compute servers on the network.
Functionality of Network Library.

1. **get_best_server**: Dynamically select the network server which currently has the lowest loading.
2. **get_server_kstats**: Get the kernel statistics [load average, memory utilisation, process load] for a set of networked processors.
3. **copen**: Execute a pipeline of commands returning a *file descriptor* to the end of the pipeline. The descriptor may be either read, write or read/write. This routine is an extension of the system command processor which permits a process to interact with an executing pipeline of commands.
4. **fcopen**: This routine is functionally identical with copen with the exception that a *stream* is returned to the calling routine rather than a file descriptor.
5. **ch__create**: Create a pair of *named pipes* which will be associated with a given interprocess communication channel.
6. **ch_open**: Open an interprocess communication *channel*. An interprocess communication channel consists of two pipes, thus permitting bi-directional data exchanges between a pair of co-operating processes.
7. **ch_close**: Close a communication channel between a pair of co-operating processes.
8. **ch_destroy**: Delete the *named pipes* associated with a given interprocess communications channel.
9. **ch_ack**: Acknowledge receipt of data; essentially a write-handshake.
10. **ch_read**: Read data from channel using handshaking.
11. **ch_write**: Write data to channel using handshaking.
12. **mkfifo**: Make a named pipe [system V named FIFO].
13. **putpkt**: Send packet of data down pipe.
14. **getpkt**: Get packet of data from pipe.
A4.5.7 The HIPL Picture/Header Library - hiplib.

The HIPL Picture/Header Library provides ANSI-C support for the HIPL Picture/Header standard [Landy and Cohen, 1982]. It consists of a set of routines for the manipulation of image data encoded to the HIPS standard.

Functionality of the HIPL Picture/Header Library.

1. hp_addr: Determine the address in memory of a pixel within a HIPS image frame.
2. hp_frame_skip: Skip a frame of image data when reading from a HIPS file.
3. hp_upd_desc: Update the HIPS Header descriptor string.
4. hp_set_desc: Initialise the HIPS Header descriptor string.
5. hp_frd_row: Read in a row of pixel values from a HIPS image file.
6. hp_frw_row: Write a row of pixel values to a HIPS image file.
7. hp_rd_header: Read HIPS header data from standard input.
8. hp_frd_hdr: Read HIPS header data from a file.
9. hp_wr_hdr: Write HIPS header data to standard output.
10. hp_fwr_hdr: Write HIPS header data to a file.
11. hp_init_hdr: Initialise HIPS header data structure.
12. hp_pread: Read data from a pipe, blocking until all data has been read.
13. hp_rd_hdr_info: Read HIPS header into an application returning the header data structure, the pixel format, and the number of rows and columns in the image data frames.
14. hp_wr_hdr_info: Write HIPS header data structure to file, automatically updating the descriptor field of the header.
15. hp_pixel_size: Return the pixel-size in bytes.

A4.5.8 The generic sensor library - glslib.

The generic sensor library is composed of a number of functions which may be used to build arbitrary geometric camera models. Although the library was derived from the earlier Sensorlib Library written to facilitate
development of the O'Neill-Dowman SPOT-1 Camera Model, all SPOT-1 dependencies were subsequently removed, creating the generic geometric camera model library glslib. The functionality of glslib is sufficient to cope with any type of rigid linear sensor model, be it either a cross track push-broom [such as SPOT-1], an along track push-broom [such as MEOSS or ITIR], or a central perspective model [most types of 'traditional’ aerial survey cameras]. The library routines are capable of supporting up to 32 camera positions [or looks ].

The functionality of the library is currently being extended to enable it to deal with more complex non-rigid sensors such as AVHRR or Landsat-TM, and synthetic aperture radars.

Performance monitor functions.

1. Powell_loop_mon: Loop monitor for Powell Direction Set minimiser
2. Powell_exit_mon: Exit monitor for Powell Direction Set Minimiser
3. ground_test: Compute object space residuals from supplied checkpoints
4. image_test: Compute image space residuals from supplied checkpoints

Data transformation functions.

1. to_image: Transform data from object space to the image space of a selected camera position or look.
2. to_ground: Transform conjugate points in the image spaces of N camera positions or looks to object space.

Powell Direction Set and Brent linear minimiser support functions.
Software Implementation

1. get_pptrs: Get pointers into attitude array
2. form_model: Form model using Powell Direction set minimiser
3. perturb_ray: Compute cost function for camera model setup
4. scalar_cost: Compute scalar cost from residual vectors
5. perturb_isp: Cost function for back transform
6. add_relax_att: Add relaxation perturbation to attitude matrix

Sensor orbit segment reconstructions functions.

1. get_vs: Get inertial velocity of satellite
2. get_psp: Get position of satellite from header
3. get_lat: Get line acquisition time
4. get_sensor_r_axes: Get sensor reference axes
5. orbit_parameters: Form all required orbit parameters using satellite velocity and position data read from SPOT-1 header.
6. add_attitude: Take account of the satellite ephemeris on satellite attitude
7. get_fl_rays: Get first and last rays in ray plane

Geometrical ray modeling support functions.

1. get_ray_pd: Get object space parametric ray equation
2. fray: Form ray emergent from rigid camera
3. formim: Find image plane position within rigid camera of entrant ray

Utility functions.

1. alloc_att_table: Allocate sensor attitude table
2. free_att_tables: Free attitude table
3. make_att_tables: Set up attitude tables for main transforms
4. make_satt_tables: Set up attitude tables for initialisation phase
5. get_attitude: Get pointer into attitude tables
6. get_normal: Get ray plane normal
7. read_attitude: Read attitude data
8. write_attitude: Write attitude data to file
9. get_imsc: Get scene centre time
10. adjust_for_strip: Take account of input which is in the form of a contiguous strip of imagery

Data input routines.

1. rd_gcpts: read in ground control points.
2. rd_sscps: read in scene centre and corner points
3. rd_cngps: read in conjugate data points
4. rd_chkpts: read in camera model check points

Endnotes to Appendix 4

1: An efficient vectorising C compiler is now available for the Cray X-MP and Y-MP supercomputer series under the UNIX system V like operating system, UNICOS.
Appendix 5

UNIX man pages for camera model and support filters
NAME
How to camera model - Notes on the camera modeller.

INTRODUCTION
This document should contain enough information to enable the novice camera modeller to get started. It assumes that you know some UNIX and at least a little about the camera modelling program.

PRELIMINARIES
Before running the camera model proper, the ground control data must be converted into the correct format. That is, the ground co-ordinates expressed in the geocentric co-ordinate system, and the image co-ordinates expressed as pixel [line and sample] co-ordinates.

The ground control data must also be in the correct format for the filter and other camera modeller filters:

\[ \text{ft} \_ \text{code} \ \text{line} \_ \text{ll} \ \text{sample} \_ \text{ll} \ \text{line} \_ \text{lr} \ \text{sample} \_ \text{lr} \ \text{X} \ \text{Y} \ \text{Z} \ldots \text{rest of line} \ldots \]

Where:

- \text{ft} \_ \text{code} \ is an integer feature code which identifies the ground control point.
- \text{line} \_ \text{ll} \ is the logical left line co-ordinate [expressed in pixels], \text{sample} \_ \text{ll} \ is the logical left sample co-ordinate [expressed in pixels].
- \text{line} \_ \text{lr} \ is the logical right line co-ordinate [expressed in pixels], \text{sample} \_ \text{lr} \ is the logical right sample co-ordinate [expressed in pixels].
- \text{X} \ is the X geocentric ground co-ordinate [expressed in metres].
- \text{Y} \ is the Y geocentric ground co-ordinate [expressed in metres].
- \text{Z} \ is the Z geocentric ground co-ordinate [expressed in metres].

When the data is in the correct format it may be transformed to pixel by applying the \text{platepix} filter [this assumes that the data is supplied in fiducial co-ordinates]:

\[ \text{platepix} \ -\text{nskip} <\text{gcps.orn} \ >\text{out}; \ \text{mv} \out \text{gcps.orn} \]

The command line given ensures that the final output file has the same name as the initial input file.

If the ground data is expressed in either the UTM, OSGB or French Lambert co-ordinate systems, it must be converted to geocentrics using the appropriate filter:

\[ \text{lz3gc} \ -\text{gcp} <\text{gcps.orn} \ >\text{out}; \ \text{mv} \out \text{gcps.orn} \quad [\text{French Lambert}] \]
\[ \text{utmgc} \ -\text{gcp} \ -\text{projf} \text{ gcp.eli} <\text{gcps.orn} \ >\text{out}; \ \text{mv} \out \text{gcps.orn} \quad [\text{UTM}] \]
\[ \text{utmgc} \ -\text{osgb} \ -\text{gcp} <\text{gcps.orn} \ >\text{out}; \ \text{mv} \out \text{gcps.orn} \quad [\text{OSGB}] \]

In the case of the UTM transformation, a projection file must be supplied, see \text{utmgc(l)}, \text{lz3gc(l)}, and \text{geogc(l)} for further details of these transformation filters.

RUNNING THE SPACE INTERSECTION ROUTINE.
Once the ground control data is in a suitable format, the space intersection routine, \text{spotlm} may be used to either transform ground points to image space, or to transform image points to ground space. The camera modeller program may operate in either test or transform mode, and in addition the camera model may be established using a combination of user selectable algorithms. Full details of how the camera model is set up is given in \text{spotlm(l)}, so a couple of illustrative examples only, will be given here.

To typically transform from image space to ground space, use the shell script:

\[ \text{zcat} \text{ gruen\_output} \ | \text{ cat} \ -n \ | \text{ awk} \ '\{(print }$1,$3,$2,$5,$4,$11,$12)\}' \ | \text{ spotlm} \ -\text{argf} \ \text{toground\_agf} \ | \text{ gcutm} \ -\text{osgb} \ | \text{ compress} \ >\text{ground\_cords} \]
The file toground.agf gives additional parameters which are used by the spotlm filter, typically:

# Tell the camera model which way to transform data -toground
# Produce a log of camera modeller O/P -verbose
# Tell the modeller where the starting scene is -startscene -10
# Tell the modeller how many scenes there are in the strip -nscenes 20
# Produce an attitude file which containing orbit segment data -wr_attitude oman.att
# Set tolerance for Powell relaxation scheme -relax_tol 0.00001
# Set up pruning limit for control and check points in standard deviation units -prune 4.0
# Use ground control point absolute difference cost function -gcpd_cost_function
# Use angular perturbation of look angles -la_correction
# Use orbit shift perturbations -orbit_correction
# Test file data: first file is where to put the test O/P -test check oman.chk
# the second is a list of checkpoints -test check oman.chk
# These are the header files - the one associated with the leftmost look is given first -hfiles oman.lh oman.rh
# This is the orientation file of measured ground control points -gcpf oman.orn

The meaning of the flags specified in toground.agf are given in spotlm(1).

The example script given assumes both the input image co-ordinates and the output ground co-ordinates are in compressed format.

To transform from ground space to image space, typically use the shell script:

zcat ground_co_ords | umgc -osgb | spotlm -argf toimage.agf | compress >image_cords

The file toimage.agf gives additional parameters which are used by the spotlm filter, typically:
# Tell the camera model which way to transform data -toimage

# Produce a log of camera modeller O/P
-verbose

# Tell the modeller where the starting scene is
-startscene -10

# Tell the modeller how many scenes there are in the strip
-nscenes 20

# Produce an attitude file which containing orbit segment data
-wr_attitude oman.att

# Set tolerance for Powell relaxation scheme
-relax_tol 0.00001

# Set up pruning limit for control and check points in standard
deviation units
-prune 4.0

# Use ground control point absolute difference cost function
-gcpd_cost_function

# Use angular perturbation of look angles
-la_correction

# Use orbit shift perturbations
-orbit_correction

# Test file data: first file is where to put the test O/P
# the second is a list of checkpoints -test check oman.chk

# These are the header files - the one associated with the
# leftmost look is given first
-hfiles oman.lh oman.rh

# This is the orientation file of measured ground control points
-gcpf oman.or

The meaning of the flags specified in toground.agf are given in spotlm(1).
The example script given assumes both the input ground co-ordinates and the output image co-ordinates are in compressed format.

SEE ALSO

COPYRIGHT
(C) 1988 University College London, Gower Street, London WC1E 6BT UK
[Alvey MMI-137, Real-time 2.5D Vision Systems]
AUTHOR
M.A. O’Neill.

SEE ALSO
spot1m(1),lz3gc(1),gclz3(1),utmgc(1),gcutm(1), pixplate(1),platepix(1)
NAME
spotlm – Advanced O’Neill-Dowman SPOT-1 camera model.

SYNOPSIS

< Ascii_list_in

> Ascii_list_out

DESCRIPTION
spotlm is a model of the SPOT-1 sensor. It provides a mechanism for transforming points between a geocentric object space and SPOT-1 image space. SPOT-1 is one of a family of rigid linear sensor models presently supported by the O’Neill-Dowman sensor supported by the O’Neill-Dowman sensor modelling system.

OPTIONS
-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The program then exits.

-version displays the program version number.

-usage displays the commands tail options which are accepted by this program.

-slots displays the pups(3) library dependancies for this program.

-nice scheduling_level determines the niceness at which the program is scheduled. In keeping with all UNIX user processes. The scheduling_level must be a cardinal number between 0 and 20, the default niceness for spotl is 4.

-argf argument_file tells the program to take its command line arguments from the specified argument_file. The data format of the argument file is:

# ... optional comment line ...
-argflag [1] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

# ... optional comment line ...

argument_parameter [1,n]

-verbose Tells the program to provide a running commentary on what it is doing. Setting the verbose flag is useful when the user is uncertain of the parameters being used with the model, or when feedback is required on the progress of the model [for an error log for example]. The information produced by setting the verbose flag is sent to standard error.

-tground Transforms data from a stereo SPOT-1 image multiplet to geocentric object space. The SPOT-1 permits up to 32 images in the stereo multiplet. The data format expected at standard input and produced on standard output are given below:
Input data format: Image stereo

ft_code X1 Y1 X2 Y2 ... Xi Yi ... rest of line ...

Where ft_code is an integer feature code, 
[X1,Y1], [X2,Y2], [Xi,Yi] are the coordinates of corresponding pixels in the images which comprise 
the stereo multiplet.

Output data format:

ft_code X Y Z ... rest of line ...

[X, Y, Z] is the position of the corresponding point in object space, expressed in geocentric coordinates.

-tooimage look tells the program to transfer data from ground space to the image space of the selected look.

-nscenes tells the camera model how many scenes are expected per image-strip. This value is assumed 
to be the same for each image-strip in the stereo multiplet.

-hfiles header_file_list reads in the list of header files associated with the stereo multiplet. The first 
header file read in is associated with the leftmost image of the stereo multiplet. The number of header 
files specified must agree exactly with the number of scenes specified via the n_scenes parameter. The 
header files are in the format produced by the Laserscan readcct program when operating in hdr mode.

-relax_tol tolerance Sets the tolerance for the Powell direction set minimiser. A value of 0.00001 is 
recommended.

-max iter max_iers Sets the upper iteration limit. At present this defaults to 200. For practical pur­ 
poses an upper limit of 40-50 is recommended.

-wr_attitude attitude_file tells the program to write attitude and orbital segment data to attitude_file in 
standard sensor modelling system format. This attitude data may be used if the model is needed at a 
later data. It may also be imported by other sensor models supported by the system, for example glsm(1) for simulation purposes

-rd_attitude attitude_file tells the program to read in attitude and orbital segment data from the file 
attitude_data. This attitude and orbital segment data is then used to form the camera model. Note that 
if rd_attitude is specified no header data is required, therefore under these conditions, a -hfiles flag is 
not parsed and will cause an error.

-test places the camera model in test mode. In this mode the camera model produces diagnostic infor­ 
mation in the files <test_result>.rchk, and <test_result>.pchk. The output of the file <test_result>.rchk is 
of the form:

minimiser_iteration checkpoint_ARMS_residual checkpoint_RRMS_residual

The first column gives the iteration number of the relaxer. The second column gives the absolute RMS 
error and the third column the relative [ray-ray] skewness error of the camera model relative to a set of independent check points, which given in the file <check_pt_file>. The format of this file is given 
below:

ft_code X Y Z

where ft_code is a feature code associated with a given check point. X, Y and Z is the geocentric
position of the check point in object space. The performance of the relaxer may be assessed by using a suitable plotting program, for example the UNIRAS based filter megagraph to display the absolute and relative RMS errors as a function of the relaxer iteration number:

For a good PAN model, the checkpoint ARMS residual should be less than about 15 metres.
For a good XS model, the checkpoint ARMS residual should be less than about 50 metres.

The second file <test_result>.gchk, contains the vector error at each of the independent check points. If N checkpoints are used, <test_results>.gchk will contain N lines, each of the form:

```
ft_code X Y Z X' Y' Z'
```

ft_code is the feature code associated with a given check-point. X,Y and Z is the measured geocentric position of the feature in object space. X', Y', Z' is the corresponding geocentric position of the feature code generated by the camera model. This may be turned into an explicit vector error in a given local co-ordinate system by using one of the co-ordinate conversion filters such as gcutm(l), gcgeo(1), or gelz3(l) in test mode.

This vector error may be assessed using a suitable vector display program, for example the UNIRAS based vector_plot program:

The Third file <test_result>.ichk, contains the vector error at each of the independent check points in image space. If N checkpoints are used, <test_results>.ichk will contain N lines, each of the form:

```
ft_code X Y X' Y'
```

where X and Y are the line and sample positions of a given pixel in the image which has been measured by a photogrammetric operative.

X' and Y' are the corresponding points in image space predicted by the camera model back-transform.

A full set of error plots may be generated for a particular camera model run by running the camera model in test mode and then using the shell scripts SPOTefg and SPOTefp to generate a set of vector plot files. These scripts are intended to be interactive, and their usage should be self explanatory.

In the statistics file generated with the plot files, the following values would indicate a good PAN camera model:

IRMSE: 0.0-1.5 pixels, ARMSE: 0.0-15.0 metres [in both plan and height], RRMSE: 0.0-10.0 metres [in both plan and height].

-gcpf gcp_point_file specifies the GCP point file. The control points given within this file are used to orient the camera to ground. The format of the gcp_point_file is identical to the check_point_file described above.

-secpf corner_point_file tells the program to read in a file of scene corner points which have been extracted from the SPOT-1 header and converted to geocentric co-ordinates. There is good reason to believe that inclusion of scene corner will produce zero GCP model with an absolute RMS plan error of less than 100 metres. The format of the corner point file is:

```
ft_code l s X Y Z l' s' X' Y' Z'
```

where:

[l,s], [l',s'] are the pixel positions of a given corner point in the logical left and logical right images respectively and [X,Y,Z], [X',Y',Z'] are the corresponding positions of the scene corner points in the image space.
geocentric co-ordinate system. The inclusion of scene corner points in the absolute orientation is only possible with the `-gcp_cost_function` rather than `-rrs_cost_function` flagged.

`-prune sigma_limit` shift prunes the control and checkpoint datasets using the given `sigma_limit` as a cut-off.

The use of shift pruning is not recommended unless there is a reasonable statistical sample \([5+]\) griound control points and/or checkpoints.

If the shift pruning option is selected with the camera model in `verbose` mode, the individual deviation of each of the shift vectors from the mean is recorded on `stderr`. This may then be used as an additional tool for dealing with erroneous check/ GCP datasets supplied to the camera model.

`-rrs_cost_function` uses a relaxation cost function which minimises the RMS distance between the ray-ray space intersections and the supplied ground control points when performing an absolute orientation.

`-gcpd_cost_function` uses a relaxation cost function which minimises the perpendicular distance between a given ray and the nearest supplied GCP. This effectively orients each of the sensor looks in a given SPOT stereo model separately. The tests which were conducted under the aegis of the RAE/LSL subcontract appear to indicate that this option yields absolute RMS errors which are between 2 and 3 metres better than the RRSKEW option described above. Note `DO_RRSKEW` and `DO_RGCPD` are mutually exclusive options.

`-orbit_correction` use an uncorrelated linear orbit shift as input to the cost function used to determine the absolute orientation.

`-la_correction` use linear rotations about the sensor X [along track] and Y [cross track] reference axes as input to the cost function used to determine the relative orientation.

`-itsize` set the blocking factor in the line dimension of the sensor. The default value for this should not, generally be changed.

`-stsize` set the blocking factor in the sample dimension of the sensor. The default value for this should not, generally be changed.

`-startscene` tells the sensor model which scene is to be considered as the starting scene [the central line of which is line zero] in the case of a strip of stereo scenes.

`-strpchk` enables strip checking if flagged. The strip checking facility uses the SPOT-1 header files to ensure that headers supplied for a strip are contiguous. If an out of sequence header is found, an error is generated, and processing is abandoned.

Note that in the case of strips, the relative overlap between any images which preport to be in a strip are calculated from the SPOT-1 headers irrespective of whether the images are contiguous.

`-restricted_ip` tells the camera model in back transform mode to ignore any image space points which fall outside the default image plane \([6000 \times 6000 \text{ PAN}, 3000 \times 3000 \text{ XS}]\) of the SPOT-1 camera model.
EXAMPLE OF USE.

The following example shows typical use of the spotl filter to transform a set of Gruen stereo matched points from image space to ground space:

```
cat -n gruen_in | awk '{print $1,$3,$2,$5,$4,$11,$12}' | spotl -argf
  spot1.agf | gclz3 -nskip |
  compress >ground_dat.Z&
```

The file spot1.agf contains a canned command tail for the SPOT-1 filter. A typical example would be:

```
# example control file for SPOT1M
# 12th February 1990

# verbose switch on - give a log of all that happens
-verbose

# iteration limit for Powell
-max_iter 100

# use header data to set up model
-hfiles hdr.left hdr.right

# tolerance for fast back-transform
-bt_tol 2.0e-16

# prune input ground control and checkpoint datasets
-prune 4.0

# cost function parameters
-rrs_cost_function
-orbit_correction
-la_correction

# test file parameters
-test check.itir itir.chk

# ground control point file
-gcpf itir.om
```

NOTE

This program requires the following support libraries in order to function:


BUGS

The program does not have any sensible defaults for the -asw and -psw switches. These switches will be provided with sensible defaults in the final production version of the camera model.
DIAGNOSTICS
The program uses the *pups(3)* error handling system: Diagnostics should be self explanatory

REFERENCES

COPYRIGHT
Program: (C) 1989 University College London, Gower Street, London WC1E 6BT UK

PUPS support libraries: (C) 1985-1991 M.A. O'Neill

PROGRAMMER
M.A. O'Neill.

SEE ALSO
*lz3gc(1),gclz3(1),geogc(1),gegeo(1),utmgec(1),gcutm(1),tospot(1), fromspot(1),gproc(1),vec(1),pups(3)*
NAME

glsm – Advanced along track stereo sensor simulator.

SYNOPSIS

glsm [ -help ] [ -version ] [ -usage ] [ -slots ] [ -argf argument_file ] [ nice scheduling_level ] [ -nsences
scnes ] [ -hfiles header_file_1, header_file_2, .. ] [ -toground ] [ -toimage look ] [ -relax_tol tolerance ] [ -max_iter max iters ] [ -w attitude attitude_file ] [ -rd attitude attitude_file ] [ -rrs_cost_function ] [ -gcpd_cost_function ] [ -orbit_correction ] [ -la_correction ] [ -prune pruning_limit ] [ -bt_tol ] [ -test check_pl file test_result ] [ -gcpf gcp_point file ] [ -scpI corner_point file ] [ -skew_limit max skewness ] [ -startscene scene ] [ -ltsize count ] [ -stsize count ] [ -strpchk ] [ -restricted_ip ] [ -max_image_obs_err pixels ] [ -max_gnd_obs_err metres ] [ -inputlook look ] [ -orbit_correction ] [ -fov view_field_list ] [ -atp_angles list_of_aip_angles ] [ -ctp_angles present, desired, bias ] [ -rearth local_earth_radius ] [ -height simulated_hi_above_earth ] [ -newpix npix, image_size, ground_size ]

< Ascii_list_in

> Ascii_list_out

DESCRIPTION

glsm is a model of the SPOT-1 sensor. It provides a mechanism for transforming points between a geo­
centric object space and SPOT-1 image space. SPOT-1 is one of a family of rigid linear sensor models
presently supported by the O’Neill-Dowman sensor supported by the O’Neill-Dowman sensor modelling
system.

OPTIONS

-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is
displayed. The program then exits.

-version displays the program version number.

-usage displays the commands tail options which are accepted by this program.

-slots displays the pups(3) library dependancies for this program.

-nice scheduling_level determines the niceness at which the program is scheduled. In keeping with all
UNIX user processes. The scheduling_level must be a cardinal number between 0 and 20, the default
niceness for glsm is 4.

-argf argument_file tells the program to take its command line arguments from the specified
argument_file. The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

-verbose Tells the program to provide a running commentry on what it is doing. Setting the verbose
flag is useful when the user is uncertain of the parameters being used with the model, or when feedback
is required on the progress of the model [for an error log for example]. The information produced by
setting the verbose flag is sent to standard error.
-toground Transforms data from a stereo SPOT-1 image multiplet to geocentric object space. The SPOT-1 permits up to 32 images in the stereo multiplet. The data format expected at standard input and produced on standard output are given below:

**input data format**

```
ft_code X1 Y1 X2 Y2 ... Xi Yi ... rest of line ...
```

Where `ft_code` is an integer feature code, `[X1,Y1],[X2,Y2],...,[Xi,Yi]` are the co-ordinates of corresponding pixels in the images which comprise the stereo multiplet.

**output data format**

```
ft_code X Y Z ... rest of line ...
```

`[X, Y, Z]` is the position of the corresponding point in object space, expressed in geocentric coordinates.

- **toimage** *look* tells the program to transfer data from ground space to the image space of the selected *look*.

- **-nsscenes** tells the camera model how many scenes are expected in the stereo multiplet.

- **-hfiles** *header_file_list* reads in the list of header files associated with the stereo multiplet. The first header file read in is associated with the leftmost image of the stereo multiplet. The number of header files specified must agree exactly with the number of scenes specified via the *n_scenes* parameter. The header files are in the format produced by the Laserscan readcct program when operating in *hdr* mode.

- **-nscenes** *number_of_scenes* Tells the program the number of scenes in the current strip of images being processed. This is an untidy feature which will be removed in a production version of the camera model.

- **-relax_tol** *tolerance* Sets the tolerance for the Powell direction set minimiser. A value of 0.00001 is recommended.

- **-max_iter** *max_iters* Sets the upper iteration limit. At present this defaults to 200. For practical purposes an upper limit of 40-50 is recommended.

- **-wr_attitude** *attitude_file* tells the program to write attitude and orbital segment data to *attitude_file* in standard sensor modelling system format. This attitude data may be used if the model is needed at a later date. It may also be imported by other sensor models supported by the system, for example glsm(1) for simulation purposes.

- **-rd_attitude** *attitude_file* tells the program to read in attitude and orbital segment data from the file *attitude data*. This attitude and orbital segment data is then used to form the camera model. Note that if *rd_attitude* is specified no header data is required, therefore under these conditions, a *-hfiles* flag is not parsed and will cause an error.

- **-test** places the camera model in test mode. In this mode the camera model produces diagnostic information in the files <test_result>.rchk, and <test_result>.pchk. The output of the file <test_result>.rchk is of the form:

```
minimiser_iteration checkpoint_ARMS_residual checkpoint_RRMS_residual
```

The first column gives the iteration number of the relaxer. The second column gives the absolute RMS
error and the third column the relative [ray-ray] skewness error of the camera model relative to a set of
independent check points, which given in the file <check_pt_file>. The format of this file is given
below:

\[ \text{ft_code} \quad X \quad Y \quad Z \]

where \text{ft_code} is a feature code associated with a given check point. \( X, Y \) and \( Z \) is the \textit{geocentric position} of the check point in object space. The performance of the relaxer may be assessed by using a suitable plotting program, for example the UNIRAS based filter megagraph to display the absolute and relative RMS errors as a function of the relaxer iteration number:

For a good PAN model, the checkpoint_ARMS residual should be less than about 15 metres.

For a good XS model, the checkpoint_ARMS residual should be less than about 50 metres.

The second file <test_result>.gchk, contains the \textit{vector error} at each of the independent check points. If \( N \) checkpoints are used, <test_result>.gchk will contain \( N \) lines, each of the form:

\[ \text{ft_code} \quad X \quad Y \quad Z \quad X' \quad Y' \quad Z' \]

\text{ft_code} is the feature code associated with a given check-point. \( X,Y \) and \( Z \) is the \textit{measured geocentric position} of the feature in object space. \( X', Y', Z' \) is the corresponding \textit{geocentric position} of the feature code generated by the camera model. This may be turned into an explicit vector error in a given \textit{local co-ordinate system} by using one of the co-ordinate conversion filters such as \texttt{gcutm(1)}, \texttt{gceo(1)}, or \texttt{gclz3(1)} in \textit{test mode}.

This vector error may be assessed using a suitable vector display program, for example the UNIRAS based \texttt{vector_plot} program:

The third file <test_result>.ichk, contains the \textit{vector error} at each of the independent check points in image space. If \( N \) checkpoints are used, <test_result>.ichk will contain \( N \) lines, each of the form:

\[ \text{ft_code} \quad X \quad Y \quad X' \quad Y' \]

where \( X \) and \( Y \) are the line and sample positions of a given pixel in the image which has been \textit{measured} by a photogrammetric operative. \( X' \) and \( Y' \) are the corresponding points in image space \textit{predicted} by the camera model back-transform.

A full set of error plots may be generated for a particular camera model run by running the camera model in test mode and then using the shell scripts \texttt{SPOTefg} and \texttt{SPOTefp} to generate a set of vector plot files. These scripts are intended to be interactive, and their usage should be self explanatory.

In the statistics file generated with the plot files, the following values would indicate a \textit{good} PAN camera model:

\textit{IRMSE}: 0.0-1.5 pixels, \textit{ARMSE}: 0.0-15.0 metres [in both plan and height], \textit{RRMSE}: 0.0-10.0 metres [in both plan and height].

\texttt{-gcpf gcp_point_file} specifies the GCP point file. The control points given within this file are used to orient the camera to ground. The format of the \texttt{gcp_point_file} is identical to the \texttt{check_point_file} described above.

\texttt{-scpf corner_point_file} tells the program to read in a file of scene corner points which have been extracted from the SPOT-1 header and converted to geocentric co-ordinates. There is good reason to believe that inclusion of scene corner will produce zero GCP model with an absolute RMS plan error of
The format of the corner point file is:

\[
\text{ft_code} \quad l \quad s \quad X \quad Y \quad Z \quad l' \quad s' \quad X' \quad Y' \quad Z'
\]

where:

\([l, s]\) are the pixel positions of a given corner point in the logical left and logical right images respectively and \([X, Y, Z] \text{ and } [X', Y', Z']\) are the corresponding positions of the scene corner points in the geocentric co-ordinate system.

The inclusion of scene corner points in the absolute orientation is only possible with the \(\text{-gcp\_cost\_function}\) rather than \(\text{-rrs\_cost\_function}\) flagged.

\text{-prune} \(\text{sigma\_limit}\ \text{shift}\) \text{prunes} the control and checkpoint datasets using the given \(\text{sigma\_limit}\) as a cut-off.

The use of shift pruning is not recommended unless there is a reasonable statistical sample \([5+]\) ground control points and/or checkpoints.

If the shift pruning option is selected with the camera model in \text{verbose} mode, the individual deviation of each of the shift vectors from the mean is recorded on \(\text{stderr}\). This may then be used as an additional tool for dealing with erroneous check/GCP datasets supplied to the camera model.

\text{-rrs\_cost\_function} uses a relaxation cost function which minimises the RMS distance between the ray-ray space intersections and the supplied ground control points when performing an absolute orientation.

\text{-gcpd\_cost\_function} uses a relaxation cost function which minimises the perpendicular distance between a given ray and the nearest supplied GCP. This effectively orients each of the sensor looks in a given SPOT stereo model separately. The tests which were conducted under the aegis of the RAE/LSL subcontract appear to indicate that this option yields absolute RMS errors which are between 2 and 3 metres better than the RRSKEW option described above. Note \(\text{DO\_RRSKEW}\) and \(\text{DO\_RGCPD}\) are mutually exclusive options.

\text{-orbit\_correction} use an uncorrelated linear orbit shift as input to the cost function used to determine the absolute orientation.

\text{-la\_correction} use linear rotations about the sensor X [along track] and Y [cross track] reference axes as input to the cost function used to determine the relative orientation.

\text{-ltsize} set the blocking factor in the line dimension of the sensor. The default value for this should not, generally be changed.

\text{-stsize} set the blocking factor in the sample dimension of the sensor. The default value for this should not, generally be changed.

\text{-startscene} tells the sensor model which scene is to be considered as the \text{starting scene} [the central line of which is line zero] in the case of a \text{strip} of stereo scenes.

\text{-stripchk} enables strip checking if flagged. The strip checking facility uses the SPOT-1 header files to ensure that headers supplied for a strip are contiguous. If an out of sequence header is found, an error is generated, and processing is abandoned.

Note that in the case of strips, the relative overlap between \text{any} images which preport to be in a strip are calculated from the SPOT-1 headers irrespective of whether the images are contiguous.
-restricted_ip tells the camera model in back transform mode to ignore any image space points which fall outside the default image plane.

-max_image_obs_err image_obs_err specifies the image observation error limit when simulating image observation error. This limit is specified as a muffin-tin pixel radius.

-max_gnd_obs_err gnd_obs_err specifies the ground observation error limit when simulating operator observation error. This limit is specified as a muffin-tin radius in metres.

-inputlook look specifies the look [read from the camera attitude file] from which the simulated orbit is to be synthesised.

-fov view_field_list specifies the field of view in radians for each look of the sensor.

-atp_angles list_of_atp_angles specifies the along track pointing angles for each look in the simulated sensor.

-ctp_angles present, desired, bias permits the user to change the cross track pointing angle from the implicit one read in with the orbit segment data. This is especially useful to synthesise nadir looks from non-nadir input data. The input [ present ] and desired look angles are specified in radians. The bias angle is used to ensure that all the rays in a test DEM are in fact mapped into the simulated ray plane. This is not strictly needed if the restricted_ip flag is set.

-rearth local_earth_radius is used to specify the local earth radius [used by the look angle recomputation routine.

-height flying_hieght is used to simulate the flying hieght of the sensor above the local earth surface.

-newpix npix, image_size, grnd_size is used to specify the number of pixels in the simulated sensor, and the sizes of each pixel in image and ground space.

EXAMPLE OF USE.
The following example shows typical use of the glsm filter to transform a set of Gruen stereo matched points from image space to ground space:

cat -n gruen_in | awk '{print $1,$3,$2,$5,$4,$11,$12}' | glsm -argf glsm.agf | gclz3 -nskip | compress >ground_dat.Z&

The file glsm.agf contains a canned command tail for the SPOT-1 filter. A typical example would be:

# example control file for GLSM
# 12th February 1990

# verbose switch on - give a log of all that happens
-verbose

# iteration limit for Powell
-max_iter 100

# Read in a simulated orbit segment
-rdatt itir.att

# tolerance for fast back-transform
-bttol 2.0e-16

# prune input ground control and checkpoint datasets
-prune 4.0

# cost function parameters
-rrs_cost_function
-orbit_correction
-la_correction

# test file parameters
-test check.itir itir.chk

# ground control point file
-gcpf itir.orn

# sensor simulation parameters # for ITIR sensor
#
# input required bias
-ctp_angles 21.78 0.1 0.0038
-atp_angles 0.0 29.4
-inputlook 0
-fov 6.09 5.19
-height 705000.0
# npix image size gnd size
-newpix 4000 19.5 15.0

NOTE
This program requires the following support libraries in order to function:

DIAGNOSTICS
The program uses the pups(3) error handling system: Diagnostics should be self explanatory

REFERENCES

COPYRIGHT
program: (C) 1989 University College London, Gower Street, London WC1E 6BT UK
PUPS support libraries: (C) 1985-1991 M.A. O'Neill

PROGRAMMER
M.A. O'Neill.
SEE ALSO
lz3gc(1), gclz3(1), gcgeol(1), utmgc(1), gcutm(1), tospot(1), fromspot(1), gproc(1), vec(1), pups(3)
NAME

gcutm - Convert between geocentric and Universal Transverse Mercator co-ordinate systems.

SYNOPSIS

```
gcutm -projf projection_file [ -osgb ][ -nice cardinal ][ -argf argument_file ][ -help ][ -test ]
```

```
< ASCII_list_in

> ASCII_list_out
```

DESCRIPTION

gcutm Converts between the geocentric and Universal Transverse Mercator [UTM] co-ordinate systems using a user supplied ellipsoid which is specified via the `projf` flag. The geocentric co-ordinates to be transformed are read from stdin, and the resultant geographical co-ordinates are written out on stdout. The filter `gcutm` is based on a Pascal program written by Gugan (1987), for co-ordinate conversion on a PDP 11/73 minicomputer. When the `test` flag is not specified, `gcutm` expects the following ASCII input

```
ft_code X Y Z ... rest of line ...
```

and produces the following ASCII output data format:

```
ft_code eutm nutm height ... rest of line ...
```

OPTIONS

- **-help** invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The process then exits.

- **-osgb** if the osgb flag is set the filter produce output in the British Ordnance Survey National Grid co-ordinate system. Since the Airey ellipsoid covering the British Isles is hardwired into the program, there is no need to specify a projection file, if the osgb flag is specified.

- **-argf argument_file** tells the filter to take its command line arguments from the specified `argument_file`. The data format of the argument file is:

```
# ... optional comment line ...
-argflag [1] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

# ... optional comment line ...

argument_parameter [1,n]
```

- **-nice cardinal** determines the niceness at which the filter is scheduled. In keeping with all UNIX user processes, nice must be a cardinal between 0 and 20. The default niceness for gcutm is 4.

- **-projf proj_file** specifies the ellipsoid to be used by the filter. This flag must be supplied as it is not sensible to assume a default ellipsoid. The data format expected in the `proj_file` is:

```
a: ellipsoid semi-major axis
b: ellipsoid semi-minor axis
x_origin: grid origin X
y_origin: grid origin Y
lat_origin: Latitude origin
```
long_origin: Longitude origin
x_offset: Offset between geocentric origin and the earth's centre [x]
y_offset: Offset between geocentric origin and the earth's centre [y]
z_offset: Offset between geocentric origin and the earth's centre [z]
ro: scale
ogrid: Not used [set to 0.0]

-nskip enables the first line of a dataset to be retained. The default is to assume that the first line of the data piped to this filter is blank, in which case, the first line of output from the filter is similarly blank.

-test enables the filter to process test format files from the SPOT camera modeller program, spotl, so that they may be viewed using a suitable vector graphing program, for example, vec. If the test option is requested, the filter expects an additional 3 columns of data. In the context of the SPOT camera modeller, gcgeo expects the following data:

ft_code X_ch Y_ch Z_ch X_c Y_c Z_c ... rest of line ...

Where X_ch, Y_ch and Z_ch are the components of a camera model check vector, and X_c, Y_c and Z_c, are the components of the corresponding ground vector produced by the camera modeller. Both sets of co-ordinates are expressed in the geocentric co-ordinate system.
The gcgeo filter produces the following output data when in test mode:

ft_code eutm_ch nutm_ch height_ch deutm dnutm dheight ... rest of line ...

Where eutm_ch, nutm_ch and height_ch are the components of a camera model check vector, and deutm, dnutm, and dheight are the components of the difference vector between the check vector and the corresponding ground vector produced by the camera modeller. Both sets of co-ordinates are expressed in the Universal Transverse Mercator co-ordinate system.

EXAMPLE OF USE
gcutm -nskip -projf ellipsoid.dat -nice 10 < geocentric_ASCII_list > utm_ASCII_list

BUGS
The conversion of geocentric co-ordinates to OSGB co-ordinates is currently supported by this filter. It should really be performed by a separate filter.

SEE ALSO

COPYRIGHT
PROGRAM
(C) 1990 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES
(C) 1985-1991 M.A. O'Neill

PROGRAMMER
M.A. O'Neill.
SEE ALSO

gcgeo(1), gc1z3(1), lz3gc(1), utmgc(1), geogc(1), spot1m(1), pups(3)
NAME
utmgc – Convert between Universal Transverse Mercator and geocentric co-ordinate systems.

SYNOPSIS
utmgc -projf ellipsoid_file [ -osgb ][ -help ][ -argf argument_file ][ -nice cardinal ][ -gcp ]
< utm_ASCII_list_in
> geocentric_ASCII_list_out

DESCRIPTION
utmgc Converts between the Universal Transverse Mercator and geocentric co-ordinate systems using a
user supplied ellipsoid. The UTM co-ordinates to be transformed are read from stdin, and the resultant
geocentric co-ordinates are written out on stdout. The filter utmgc is based on a Pascal program written
by Gugan (1987), for co-ordinate conversion on a PDP 11/73 minicomputer. utmgc expects the follow­
ing input data

ft_code eutm nutm height ... rest of line ...

and produces the following output data format:

ft_code X Y Z ... rest of line ...

OPTIONS
-argf argument_file tells the filter to take its command line arguments from the specified argument_file.
The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2]
argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2]
argument_parameter [1,n]

argout cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user
processes, nice must be a cardinal between 0 and 20. The default niceness for utmgc is 4.

-gcp read in data in GCP mode. Normally the filter expects data to be in the following mode:

ft_code e_lcc n_lcc height ... rest of line ...

GCP mode is to enable the ground co-ordinates in GCP files to be transformed from Universal
Transverse Mercator to Geocentric. The data format expected is that of the GCP input file for the cam­
era modeller, spotl:
ft_code line_1 sample_1 line_2 sample_2 e_utm n_utm height ... rest of line ...

-projf ellipsoid_file specifies the ellipsoid to be used by the filter. This flag must be supplied as it is not sensible to assume a default ellipsoid. The data format expected in the projf file is:

- **a**: ellipsoid semi-major axis
- **b**: ellipsoid semi-minor axis
- **x_origin**: grid origin X
- **y_origin**: grid origin Y
- **lat_origin**: Latitude origin
- **long_origin**: Longitude origin
- **x_offset**: Offset between geocentric origin and the earths centre [x]
- **y_offset**: Offset between geocentric origin and the earths centre [y]
- **z_offset**: Offset between geocentric origin and the earths centre [z]
- **ro**: scale ogrid: Not used [set to 0.0]

EXAMPLE OF USE

```
utmgc -nskip -projf ellipsoid.dat < geographical_ASCII_list > geocentric_ASCII_list
```

BUGS

Currently transformation between OSGB and geocentric is supported by this filter. It should really be performed by a seperate filter.

SEE ALSO


COPYRIGHT

PROGRAM

(C) 1991 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES

(C) 1985-1991 M.A. O'Neill

PROGRAMMER

M.A. O'Neill.

SEE ALSO

gclz3(1),gcutm(1),gcgeo(1),geogc(1),lz3gc(1),spot1m(1),pups(3)
NAME
gclz3 - Convert between geocentric and French Lambert [Lambert Conformal Conic Zone 3] co-ordinate systems.

SYNOPSIS
gclz3 [ -help ][ -argf argument_file ][ -nice cardinal ][ -test ]
< ASCII_list_in > ASCII_list_out

DESCRIPTION
gclz3 Converts between the geocentric and Lambert zone 3 co-ordinate systems using the Clark (1880) ellipsoid. The geocentric co-ordinates to be transformed are read from stdin, and the resultant Lambert zone 3 co-ordinates are written out on stdout. The filter lz3gc is based on a Pascal program written by Gugan (1987), for co-ordinate conversion on a PDP 11/73 minicomputer. When the test flag is not specified, gclz3 expects the following input data

ft_code X Y Z ... rest of line ...

and produces the following output data format:

ft_code e_lcc n_lcc height ... rest of line ...

OPTIONS

- help invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The process then exits.

- argf argument_file tells the filter to take its command line arguments from the specified argument_file. The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2] argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2] argument_parameter [1,n]

- nice cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user processes, nice must be a cardinal between 0 and 20. The default niceness for gclz3 is 4.

- test enables the filter to process test format files from the SPOT camera modeller program, spot1, so that they may be viewed using a suitable vector graphing program, for example, vec. If the test option is requested, the filter expects an additional 3 columns of data. In the context of the SPOT camera modeller, gclz3 expects the following data:

ft_code X_ch Y_ch Z_ch X_c Y_c Z_c ... rest of line ...

Where X_ch, Y_ch and Z_ch the components of a camera model check vector, and X_c, Y_c and Z_c, are the components of the corresponding ground vector produced by the camera modeller. Both sets of co-ordinates are expressed in the geocentric co-ordinate system.

The gclz3 filter produces the following output data when in test mode:

ft_code e_lcc_ch n_lcc_ch height_ch de_lcc dn_lcc dheight ... rest of line ...
Where $X_{ch}$, $Y_{ch}$ and $Z_{ch}$ are the components of a camera model check vector, and $dX$, $dY$, and $dZ$ are the components of the difference vector between the check vector and the corresponding ground vector produced by the camera modeller. Both sets of co-ordinates are expressed in the Lambert Zone 3 [French Lambert] co-ordinate system.

**EXAMPLE OF USE**

```
gclz3 -nskip -nice 10 < French_Lambert_ASCII_list > geocentric_ASCII_list
```

**BUGS**

Currently only supports Lambert zone 3, could be made more general.

**SEE ALSO**


**COPYRIGHT**

PROGRAM

(C) 1989 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES

(C) 1985-1991 M.A. O’Neill

PROGRAMMER

M.A. O’Neill.

**SEE ALSO**

gcutm(1),gcgeo(1),lz3gc(1),utmgc(1),geogc(1),spot1m(1),pups(3)
NAME
lz3gc - Convert between French Lambert [Lambert Conformal Conic Zone 3] and geocentric co­ordinate systems.

SYNOPSIS
lz3gc [ -help ] [ -argf argument_file ] [ -nice cardinal ] [ -gcp ]

< ASCII_list_in > ASCII_list_out

DESCRIPTION
lz3gc Converts between the French Lambert and geocentric co-ordinate systems using the Clark (1880) geoid. The French Lambert co-ordinates to be transformed are read from stdin, and the resultant geocentric co-ordinates are written out on stdout. The filter lz3gc is based on a Pascal program written by Gugan (1987), for co-ordinate conversion on a PDP 11/73 minicomputer.

lz3gc expects the following input data format:

ft_code e_lcc n_lcc height ... rest of line ...

and produces the following output data format:

ft_code X Y Z ... rest of line ...

OPTIONS
-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The process then exits.

-argf argument_file tells the filter to take its command line arguments from the specified argument_file.

The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1.1] argument_parameter [1.2]

argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1.1] argument_parameter [1.2]

argument_parameter [1,n]

-nice cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user processes, nice must be a cardinal between 0 and 20. The default niceness for lz3gc is 19.

-gcp read in data in GCP mode. Normally the filter expects data to be in the following mode:

ft_code e_lcc n_lcc height ... rest of line ...

GCP mode is to enable the ground co-ordinates in GCP files to be transformed from Lambert Conformal Conic to Geocentric. The data format expected is that of the GCP point file for the camera modeller, spot1:

ft_code line_1 sample_1 line_2 sample_2 e_lcc n_lcc height ... rest of line ...
EXAMPLE OF USE
The following example shows how the \textit{lz3gc} filter may be used to calculate the disparity limits associated with a given ASCII input file:

\begin{verbatim}
nice lz3gc < French_Lambert_ASCII_list > geocentric_ASCII_list
\end{verbatim}

BUGS
Currently only support Lambert zone 3, could be made more general.

SEE ALSO

COPYRIGHT
PROGRAM
(C) 1989 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES
(C) 1985-1991 M.A. O'Neill

PROGRAMMER
M.A. O'Neill.

SEE ALSO
gclz3(1),gcutm(1),gcgeo(1),utmgc(1),geogc(1),spotlm(1),pups(3)
NAME
gcgeo - Convert between geocentric and geographical co-ordinate systems.

SYNOPSIS
gcgeo -projf projection_file [ -nice cardinal ] [ -argf argument_file ] [ -help ] [ -test ]

< ASCII_list_in > ASCII_list_out

DESCRIPTION
gcgeo Converts between the geocentric and geographical co-ordinate systems using a user supplied ellipsoid which is specified via the projf flag. The geocentric co-ordinates to be transformed are read from stdin, and the resultant geographical co-ordinates are written out on stdout. The filter gcgeo is based on a Pascal program written by Gugan (1987), for co-ordinate conversion on a PDP 11/73 mini-computer. When the test flag is not specified, gcgeo expects the following input data

ft_code X Y Z ... rest of line ...

and produces the following output data format:

ft_code lat long height ... rest of line ...

OPTIONS

-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The process then exits.

-argf argument_file tells the filter to take its command line arguments from the specified argument_file. The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

-nice cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user processes, nice must be a cardinal between 0 and 20. The default niceness for gcgeo is 19.

-projf ellipsoid_file specifies the ellipsoid to be used by the filter. This flag must be supplied as it is not sensible to assume a default ellipsoid. The data format expected in the projf file is:

a: ellipsoid semi-major axis
b: ellipsoid semi-minor axis
x_origin: grid origin X
y_origin: grid origin Y
lat_origin: Latitude origin
long_origin: Longitude origin
x_offset: Offset between geocentric origin and the earth's centre [x]
y_offset: Offset between geocentric origin and the earth's centre [y]
z_offset: Offset between geocentric origin and the earth's centre [z]
ro: scale
ogrid: Not used [set to 0.0]

-test enables the filter to process test format files from the SPOT camera modeller program, spotl, so
that they may be viewed using a suitable vector graphing program, for example, vec. If the test option
is requested, the filter expects an additional 3 columns of data. In the context of the SPOT camera
modeller, gcgeo expects the following data:

ft_code X_ch Y_ch Z_ch X_c Y_c Z_c ... rest of line ...

Where X_ch, Y_ch and Z_ch the components of a camera model check vector, and X_c, Y_c and Z_c, are
the components of the corresponding ground vector produced by the camera modeller. Both sets of
co-ordinates are expressed in the geocentric co-ordinate system.

The gcgeo filter produces the following output data when in test mode:

ft_code lat_ch long_ch height_ch dlat dlong dheight ... rest of line ...

Where lat_ch, long_ch and height_ch are the components of a camera model check vector, and dlat,
dlong, and dheight are the components of the difference vector between the check vector and the
corresponding ground vector produced by the camera modeller. Both sets of co-ordinates are expressed
in the geographical [lat, long, height] co-ordinate system.

EXAMPLE OF USE

gcgeo -nskip -projf ellipsoid.dat -nice 10 < geocentric_ASCII_list > geographical_ASCII_list

SEE ALSO

D.J. Gugan, Practical aspects of topographic mapping from SPOT imagery, Photogrammetric record,

COPYRIGHT

PROGRAM

(C) 1989 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES

(C) 1985-1991 M.A. O’Neill

PROGRAMMER

M.A. O’Neill.

SEE ALSO

gcutm(1),gclz3(1),lz3gc(1),utmgc(1),geogc(1),spot1m(1),pups(3)
NAME
geogc - Convert between geographical and geocentric co-ordinate systems.

SYNOPSIS
geogc -projf ellipsoid_file [ -help ] [ -argf argument_file ] [ -nice cardinal ] [ -gcp ]
< ASCII_list_in > ASCII_list_out

DESCRIPTION
geogc Converts between the geographical and geocentric co-ordinate systems using a user supplied ellipsoid. The geographical co-ordinates to be transformed are read from stdin, and the resultant geocentric co-ordinates are written out on stdout. The filter geogc is based on a Pascal program written by Gugan (1987), for co-ordinate conversion on a PDP 11/73 minicomputer.
.geogc expects the following input data format:

ft_code lat long height ... rest of line ...

and produces the following output data format:

ft_code X Y Z ... rest of line ...

OPTIONS
-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The process then exits.
.B -argf argument_file tells the filter to take its command line arguments from the specified argument_file. The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

-nice cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user processes, nice must be a cardinal between 0 and 20. The default niceness for geogc is 19.

-gcp read in data in GCP mode. Normally the filter expects data to be in the following mode:

ft_code e_lcc n_lc height ... rest of line ...

GCP mode is to enable the ground co-ordinates in GCP files to be transformed from geographical to geocentric. The data format expected is that of the GCP point file for the camera modeller, spotl:

ft_code line_1 sample_1 line_2 sample_2 e_lcc n_lc height ... rest of line ...

-projf ellipsoid_file specifies the ellipsoid to be used by the filter. This flag must be supplied as it is not sensible to assume a default ellipsoid. The data format expected in the projf file is:

a: ellipsoid semi-major axis
b: ellipsoid semi-minor axis
x_origin: grid origin X
GEOGC(1)  USER COMMANDS  GEOGC(1)

y_origin:       grid origin Y
lat_origin:     Latitude origin
long_origin:    Longitude origin
x_offset:       Offset between geocentric origin and the earths centre [x]
y_offset:       Offset between geocentric origin and the earths centre [y]
z_offset:       Offset between geocentric origin and the earths centre [z]
ro:             scale
ogr:            Not used [set to 0.0]

EXAMPLE OF USE
  geogc -nskip -proj ellipsoid.dat < geographical_ASCII_list > geocentric_ASCII_list

SEE ALSO

COPYRIGHT
PROGRAM
  (C) 1991 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES
  (C) 1985-1991 M.A. O'Neill

PROGRAMMER
  M.A. O'Neill.

SEE ALSO
  gclz3(1),gcutm(1),gcgeo(1),utmgc(1),lz3gc(1),spotlm(1),pups(3)
NAME
platepix – Convert between pixel and plate image co-ordinate systems.

SYNOPSIS
pixplate [ -nice cardinal ] [ -argf argument_file ] [ -help ] [ -imsize cardinal ] [ -ccdsize cardinal ]
< ASCII_list_in
> ASCII_list_out

DESCRIPTION
pixplate Converts between pixel and plate image co-ordinate systems. The plate co-ordinates to be
transformed are read from stdin, and the resultant pixel co-ordinates are written out on stdout. platepix
expects the following input data format [expressed in pixel co-ordinates] :

ft_code line_1 sample_1 line_2 sample_2 ... rest of line ...

and produces the following output data format [expressed in plate co-ordinates] :

ft_code line_1 sample_1 line_2 sample_2 ... rest of line ...

OPTIONS
-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is
displayed. The process then exits.

-argf argument_file tells the filter to take its command line arguments from the specified argument_file.
The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2]

argument_parameter [1,n]

-nice cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user
processes, nice must be a cardinal between 0 and 20. The default niceness for pixplate is 4.

-imsize cardinal is used to specify the size of the image in pixels. For SPOT, this is nominally 6000
when operating in PAN mode and 3000 when operating in XS mode. pixplate defaults to the value
consistent with SPOT PAM mode [6000].

-ccdsize cardinal is used to specify the size of the ccd detectors in microns. For SPOT, this is nomi-
nally 13 when operating in PAN mode and 26 when operating in XS mode. pixplate defaults to the
value consistent with SPOT PAN mode [13].

EXAMPLE OF USE
platepix -nice 10 < pixel_ASCII_list > plate_ASCII_list
COPYRIGHT
PROGRAM
   (C) 1991 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES
   (C) 1985-1991 M.A. O'Neill

PROGRAMMER
   M.A. O'Neill.

SEE ALSO
   platepix(1), imdisp(1), spotlm(1), pups(3)
NAME
platepix – Convert between plate and pixel image co-ordinate systems.

SYNOPSIS
platepix [ -nice cardinal ] [ -argf argument_file ] [ -help ] [ -imsize cardinal ] [ -ccdsize cardinal ]

< ASCII_list_in

> ASCII_list_out

DESCRIPTION
platepix Converts between plate and pixel image co-ordinate systems. The plate co-ordinates to be transformed are read from stdin, and the resultant pixel co-ordinates are written out on stdout. platepix expects the following input data format [expressed in plate co-ordinates]:

ft_code line_1 sample_1 line_2 sample_2 ... rest of line ...

and produces the following output data format [expressed in pixel co-ordinates]:

ft_code line_1 sample_1 line_2 sample_2 ... rest of line ...

OPTIONS
-help invokes the on-line context help system for the geometrical camera modeller. This "man" page is displayed. The process then exits.

-argf argument_file tells the filter to take its command line arguments from the specified argument_file. The data format of the argument file is:

# ... optional comment line ... -argflag [1] argument_parameter [1,1] argument_parameter [1,2]
argument_parameter [1,n]

# ... optional comment line ... -argflag [2] argument_parameter [1,1] argument_parameter [1,2]
argument_parameter [1,n]

-nice cardinal determines the niceness at which the filter is scheduled. In keeping with all UNIX user processes, nice must be a cardinal between 0 and 20. The default niceness for platepix is 4.

-imsize cardinal is used to specify the size of the image in pixels. For SPOT, this is nominally 6000 when operating in PAN mode and 3000 when operating in XS mode. platepix defaults to the value consistent with SPOT PAM mode [6000].

-ccdsize cardinal is used to specify the size of the ccd detectors in microns. For SPOT, this is nominally 13 when operating in PAN mode and 26 when operating in XS mode. platepix defaults to the value consistent with SPOT PAN mode [13].

EXAMPLE OF USE
platepix -nice 10 < plate_ASCII_list > pixel_ASCII_list

Sun Release 4.1 Last change: 12th July 1991
COPYRIGHT

PROGRAM
(C) 1991 University College London, Gower Street, London WC1E 6BT UK

LIBRARIES
(C) 1985-1991 M.A. O'Neill

PROGRAMMER
M.A. O'Neill.

SEE ALSO
pixplate(1),imdisp(1),spot1m(1),pups(3)
NAME
gproc – Sun dynamic data display and graphing filter.

SYNOPSIS
gproc [ -argf command_tail_file ][ -help ][ -master ][ -slave ][ -file file_name ][ -xmin minimum_x_value ][ -xstep x_step_value ][ -ymin minimum_y_value ][ -ymax maximum_y_value ][ -ystretch y_stretch_factor ][ -ydisp display_mode ][ stdin_file ] stdout_file ]

DESCRIPTION
gproc is a display tool for looking at and dynamically interacting with XY graphs. It is a SUN implementation, with slight restrictions of the original gproc system which was developed for the BBC Microcomputer System [O'Neill, 1988a]. gproc permits the user to look at XY graphical data, magnifying if this is necessary. In addition, a rubber banding mode enables the user to changed the data by dragging the displayed section of graph, until the necessary changes have been accomplished. Although gproc is principally an interactive tool, most of the parameters which may be altered dynamically, either from the keyboard, or via the walking menu or dialog subsystems of the sunview windowing environment, may be supplied as parameters within the command line tail when the program is called. gproc is capable of reading data either from pipes (of which it forms a display component), or from regular UNIX files. The current version of gproc does not support dynamically alterable file pathnames, although it is anticipated that this feature will be added later versions of the program. In addition to supporting standard UNIX pipes, gproc also supports the notion of a master slave control pipe. In this mode, a single gproc master, is able to control what is displayed by a number of gproc slaves. The creation of master and slave processes is controlled via a set of switches in the command tail. gproc is intended to be used as a data viewing and manipulation tool in conjuction with other processes. Like the original BBC based gproc package, it can be used to define data windows, upon which an operation, for example, quadrature, differentiation, or Fourier transforming will be applied. The interactive data alteration mode of Sun gproc was developed to interactively craft transfer functions for the Fast Hartley based general purpose filtering program, see fhamp(1), [O'Neil, 1988b]

Interactive mode
In interactive mode the following devices are connected to the sunview notifier.

M1 the leftmost mouse button. This is used to activate the dragging cursor, by depressing the mouse button, which enables the form of the displayed graph to be altered using rubber banding.

M2 The middle mouse button. This is used to scroll the displayed graph. If the cursor is above the dotted line when M2 is depressed, the graph is scrolled upwards by an amount proportional to the Y distance between the cursor and the dotted line. If the cursor is below the dotted line when M2 is depressed, the graph is scrolled downwards by an amount proportional to the distance between the cursor and the dotted line.

M3 the rightmost mouse button is used to activate the sunview walking menu system. The use of the menu system should be obvious, but the basics of its use are documented in the appropriate Sun User Guide, in which the philosophy of the windowing environment sunview, is explained.

R1 Function key R1 is used to set marker 1 at the current Y cursor position.

R2 Function key R2 is used to set marker 2 at the current Y cursor position.

R4 Function key R5 is used to reset marker 1.

R5 Function key R2 is used to reset marker 2.
For further details of the interactive part of the gproc interface see [O'Neill, 1988a]

In addition the alpha keyboard is used to enter data into CLI dialogs which may be produced in response to some menu selections.

Data file formats.

gproc expects data in the so called sun gproc format. This is essentially a numerical function in which the step associated with the X data is constant.
The data format consists of a header line, which defines the X limits of the numerical function and the number of data points in it:

\[<\text{data_pts}> \quad <\text{X_start}> \quad <\text{X_step}>\]

Where:
\(<\text{data_pts}>\) is the number of data points within the numerical function.
\(<\text{X_start}>\) defines the start of the numerical functions domain.
\(<\text{X_step}>\) defines the step length separating the \(i\)th and the \((i+1)\)th value within the domain of the numerical function.
Each line of data if of the form:

\[<\text{id_code}> \quad <\text{Y_value}>\]
Where:
\(<\text{id_code}>\) is an integer identification code.
\(<\text{Y_value}>\) is the value of the numerical function for the \(i\)th point.

.SH OPTIONS

-argf argument_file, invokes the command tail file facility. Command tails which are used repetitively may be stored in files which conventionally have the extension .agf. The argf parameter causes the contents of such a file to be loaded into the standard argument vector, argv, and the argument count, argc to be updated accordingly. The command argument file must of course contain valid arguments for vec.

-help invokes the help utility, which displays this man page.

-master invokes master mode for this particular gproc process. In master mode, the command which the user gives to the master process, are echoed to its slaves. Thus, master/ slave mode may used, for example to compare the form of two functions in the same region of space. The present version of gproc uses a standard UNIX pipe to communicate between the master and slave processes. In the present version of the system, only one slave process is permitted. It is expected that in future versions of the system, the number of slaves will be increased, either by daisy chaining, using standard UNIX pipes, or by adapting the basic gproc system to communicate using the Remus distributed processing system, see remus(3). If the user specifies both master mode, and slave mode for the same gproc process a command line argument error abort will result.

-slave is used to set up a gproc process as a.l slave process. In this mode, the process is slaved to a master process, form which it receives commands via a UNIX pipe. If the user specifies both master mode, and slave mode for the same gproc process a command line argument error abort will result.

-xmin \(x_{-}\text{min\_value}\) is used to specify the start of the domain, via the real parameter \(x_{-}\text{min\_value}\).

-xstep \(x_{-}\text{step\_value}\) is used to specify the domain step length via the real parameter \(x_{-}\text{step\_value}\).

-ymin \(y_{-}\text{min\_value}\) is used to specify the minimum value in the range, via the real parameter
y_min_value.

-ymax y_max_value is used to specify the maximum value in the range, via the real parameter y_max_value.

-ystretch y_stretch_value is used to specify the stretch factor for the range, via the real parameter y_stretch_value.

-ydisp display_type is used to set the initial display type, logarithmic or linear via the string parameter, display_type, which may be set either to the string 'log' or the string 'lin'. Any other string value will cause an error abort.

-pipe is used to inform the program that data will be read from a UNIX pipe. If the data read from sun gproc format, an error abort will occur. The pipe and file options are mutually exclusive, and must not both appear on the same command line. Failure to comply with this condition will cause an argument error abort.

-file file_name is used to specify a file, in the current directory, from which the data to be viewed is to be read. If this file cannot be found, or if the data in the file does not adhere to sun gproc format, and error abort will result. The pipe and file options are mutually exclusive, and must not both appear on the same command line. Failure to comply with this condition will cause an argument error abort.

BUGS

The logarithmic display option is not working correctly and may cause core dumps. Also the file segment magnification markers may disappear if one is inadvertently placed over another.

REFERENCES

[O'Neill, 1988a] O'Neill, M.A., Gproc - An integrated system for the processing of numerical data, Software - Practice and Experience,


COPYRIGHT

PROGRAM
(C) 1991 M.A. O'Neill

LIBRARIES
(C) 1985-1991 M.A. O'Neill

PROGRAMMER
M.A. O'Neill.

SEE ALSO
vec(1), pups(3)
NAME
vec - SPOT camera model vector error display program.

SYNOPSIS

DESCRIPTION
vec is a display tool for looking at camera errors generated by SPOT camera models. It is used to produce pin diagrams, which represent the vector error in either image space or ground space. The program is capable of operating both as a component within a UNIX pipeline, or standalone. All of the command line arguments which may be supplied to the program are optional. The system default values are set up to be sensible within the context of displaying SPOT camera error data. The arguments which are supplied statically, via the command tail may also be changed dynamically while the program is running. This dynamic changing of arguments is accomplished via walking menus and dialog boxes which are part of the sunview windowing environment.

With the exception of the -pipe argument, any command which may be entered as a command line argument may also be entered interactively, via the sunview menu system, which operates in conjunction with pop-up dialog boxes. vec also supports a user defined mode which enables it to be extended to the plotting of vector fields of a more general nature.

Interactive mode
M3 the rightmost mouse button is used to activate the sunview walking menu system. The use of the menu system should be obvious, but the basics of its use are documented in the appropriate Sun User Guide, in which the philosophy of the windowing environment sunview, is explained.

In addition the alpha keyboard is used to enter data into CLI dialogs which may be produced in response to some menu selections.

Data file formats.
vec expects files in the following formats.

Object mode format:

```
ft_code x_base y_base z_base x_vec y_vec z_vec
```

Where:

- `ft_code` is a feature code associated with a given error vector.
- `x_base`, `y_base`, and `z_base` are the position vector of the error vector, relative to the origin in 3-space.
- `x_vec`, `y_vec`, and `z_vec` are the components of the error vector in 3-space.

Image mode format:

```
ft_code x_base y_base x_vec y_vec
```

Where:

- `ft_code` is a feature code associated with a given error vector.
x_base, and y_base, are the position vector of the error vector, relative to the origin in 2-space.

x_vec, and y_vec, are the components of the error vector in 2-space.

If a datafile read by vec is inconsistent with the current operating mode, an error dialog, is produced.

Operating modes.

vec has two principal modes of operation. image mode, in which two dimensional image-space pin diagrams are produced, and object mode in which three dimensional object-space pin diagrams are produced. In image mode, the error vector is visualised as a solid line. A square box at one end of the vector, replaces the conventional arrow, in showing the sense of the vector.

In object mode, the error vector is projected in the following manner. The XY projection of the error vector is displayed in standard vector format, that is as a line with an arrowhead, the sense of which indicates the direction of the XY projection. The Z component of the vector is displayed as a line, perpendicular to the XY projection of the vector and centred about the attachment point of the vector to the projection space. The sense of the Z component of the vector may be deduced from the way in which the vector is plotted. If the vector is plotted as a solid line, then the Z component of the vector is positive, given a left handed 3-spatial axis system. If the vector is projected as a dotted line, the sense of the Z component is correspondingly negative.

REMOTE COMMAND INTERFACE

vec supports the concept of a non resident operations, which may be initiated from a shell dialog selected via the vec menu system. The extended operations are UNIX commands whose I/O adheres to a very simple transfer protocol, LSDT (Large Scale Data Transfer Protocol), which permits multiple processes which may be running on different machines, to share data via a simple file paradigm. Essentially, the master process, in this case vec must supply the process spawned by the shell with a file whose name is of the form <command>.sbt, where <command> is the name of the command which is to be run by the process spawned by the shell. In the present version of vec, the process is spawned on a host, selected by the user via an interactive menu system using the Sun rpc mechanism indirectly via the system and rsh calls. It would be equally easy to spawn the processes using the rmexec primitive of the Remus interprocess communication system. The remote processess passes data back via a file of the form <command>.pd. This file, contains processed data in a format suitable for vec, to load and display. If the remote process fails to return a result within an appropriate timeout interval, vec will produce an appropriate error dialog, and return to ready status. The extended command feature has been provided to permit the user to run appropriate programs from within the vec environment.

OPTIONS

-argf argument_file, invokes the command tail file facility. Command tails which are used repetitively may be stored in files which conventionally have the extension .agf. The argf parameter causes the contents of such a file to be loaded into the standard argument vector, argv, and the argument count, argc to be updated accordingly. The command argument file must of course contain valid arguments for vec.

-mloxp xy_plan_mag_lo. is used to set the lower magnification limit for the two dimensional space in which the error vectors are projected. Its default value is 1. The variable xy_plan_mag_lo, which must be integer is used to set the lower magnification limit. If the lower magnification limit exceeds the upper limit, an error message is printed on stderr, and the process is aborted.

-mhixyp xy_plan_mag_hi. is used to set the upper magnification limit for the two dimensional space in which the error vectors are projected. Its default value is 500. The parameter xy_plan_mag_hi, which
must be integer is used to set the upper magnification limit. If the upper magnification limit is less than
the lower limit, an error message is printed on stderr, and the process is aborted.

-magxy p xy_plan_initial_mag sets the default magnification factor for the X and Y components of
the error vector. The default value for this parameter is 201. If the supplied parameter resolves outside the
limits defined by xy_plan_hi, and xy_plan_lo, and error message is printed on stderr, and the process is
aborted.

-milogy xy_vector_mag_lo. is used to set the lower magnification limit for the X and Y components of
the error vector. Its default value is 1. The parameter xy_vector_mag_lo, which must be integer, is used
to set the lower magnification limit. If the lower magnification limit exceeds the upper limit, an error
message is printed on stderr, and the process is aborted.

-mhixy xy_vector_mag_hi. is used to set the upper magnification limit for the X and Y components of
the error vector. Its default value is 500. The parameter xy_vector_mag_hi, which must be integer, is
used to set the lower magnification limit. If the upper magnification limit is less than the lower limit, an
error message is printed on stderr, and the process is aborted.

-magzy xy_vector_initial_mag sets the default magnification factor for the X and Y components of the
error vector. The default value for this parameter is 201. If the supplied parameter resolves outside the
limits defined by xy_vector_hi, and xy_vector_lo, and error message is printed on stderr, and the pro-
cess is aborted.

-milogy z_vector_lo is used to set the lower magnification limit for the Z component of the error vector.
Its default value is 1. The parameter z_vector_lo, which must be an integer, is used to set the lower
magnification limit. If the lower magnification limit exceeds the upper limit, an error message is printed
on stderr, and the process is aborted. If the system is in image mode, this argument is ignored.

-mhizy z_vector_hi is used to set the lower magnification limit for the Z component of the error vector.
Its default value is 500. The parameter z_vector_hi, which must be an integer, is used to set the upper
magnification limit. If the upper magnification limit is less than the lower limit, an error message is
issued on stderr, and the process is aborted. If the system is in image mode, this argument is ignored.

-magzy z_vector_initial_mag sets the default magnification factor for the Z component of the error vec-
tor. The default value for this parameter is 201. If the supplied parameter resolves outside the limits
defined by z_vector_hi, and z_vector_lo, and error message is printed on stderr, and the process is
aborted.

-xorg x_origin sets the effective X-origin of the two dimensional space in which the error vectors are
projected via the parameter x_origin, which is a float. If the autoscaling facility is enabled, this param-
eter is effectively ignored.

-xorg x_origin sets the effective X-origin of the two dimensional space in which the error vectors are
projected via the parameter x_origin, which is a float. If the autoscaling facility is enabled, this param-
eter is effectively ignored.
-\texttt{yorg} \textit{y\textsubscript{origin}} sets the effective Y-origin of the two dimensional space in which the error vectors are projected via the parameter \textit{y\textsubscript{origin}}, which is a float. If the \textit{autoscaling} facility is enabled, this parameter is effectively ignored.

\textit{size xy\textsubscript{size}} sets the size of the two dimensional space in which the error vectors are projected via the parameter, \textit{xy\textsubscript{size}}, which is a float. If the \textit{autoscaling} facility is enabled, this parameter is effectively ignored.

-\texttt{cut view\textsubscript{cut}} is used to specify the principal plane of projection for \textit{object} data, via the parameter \textit{view\textsubscript{cut}}. The default value of \textit{view\textsubscript{cut}} is XY. If \textit{image mode} is selected, this option is effectively ignored.

-\texttt{noshowzp} is used to select whether or not the Z value of a 3-vector is to be projected when in \textit{object mode}. If the parameter \texttt{noshowzp} is given in the command tail, only the in-plane components of the field vector are displayed. If \textit{image mode} is selected, this parameter is effectively ignored.

-\texttt{noauto} resets the autoscaling flag. The default state of the system is to have the autoscaling flag enabled. In this mode the parameters \textit{x\textsubscript{origin}}, \textit{y\textsubscript{origin}}, and \textit{size}, are automatically selected so that the all the data within the current error-file can be seen.

-\texttt{file residual\_file\_name} is used to supply a data file from which error vector data is read. The file name is passed via the string parameter \textit{residual\_file\_name}, which may be a maximum of 255 characters in length. If the \texttt{pipe} option is requested, this parameter is effectively ignored, and data is read preferentially from the pipe.

-\texttt{pipe < residual\_file\_name tell vec} that data is to be read from the \textit{pipe file} specified in the command line. The \texttt{pipe} option takes precedence over the \texttt{file} option, if both are specified.

-\texttt{mode operating\_mode} is used to set up the \textit{operating} mode of the program. At present, the program has 5 preset modes of operation, \textit{geocentric, geographical, LZ3, line and sample, and fiducal} which are of use when displaying output from SPOT camera models. In addition, a \textit{user defined} mode of operation is supported. In \textit{user mode}, the user is able to supply the information which will be displayed on the axes and scalebars via additional arguments in the command line tail.

-\texttt{proj proj\_mode} is used to supply the projection mode, (2-space or 3-space) which will be used in conjunction with a given user defined display. The string parameter \textit{proj\_mode} may be set to two values, \textit{image} or \textit{object}. Any other value supplied via the \textit{proj\_mode} argument will cause an error abort. This command line argument must only be supplied if \textit{user mode} has been selected. If it is used in conjunction with any other mode, it will cause an argument error abort.

-\texttt{xlabel label} is used to set up the X axis label in \textit{user} mode. The label is passed via the string parameter \textit{label}. This command line argument must only be supplied if \textit{user mode} has been selected. If it is used in conjunction with any other mode, it will cause an argument error abort.

-\texttt{ylabel label} is used to set up the Y axis label in \textit{user} mode. The label is passed via the string parameter \textit{label}. This command line argument must only be supplied if \textit{user mode} has been selected. If it is used in conjunction with any other mode, it will cause an argument error abort.

\textbf{Bugs}

At present the file format detection system does detect all types of file of the wrong format. There is also a labelling bug -- the last point in a dataset remains unlabelled even though a point label may have been requested for that point.
COPYRIGHT
PROGRAM
(C) 1991 M.A. O'Neil

LIBRARIES
(C) 1985-1991 M.A. O'Neil

PROGRAMMER
M.A. O'Neil. [based on an earlier program by A. Anthony]

SEE ALSO
gproc(1),pups(3)
Appendix 6
A simulation study of the ITIR sensor

A6.0 Introduction.

This report discusses a sensor simulation study which has been carried out for the Japanese Geoscience Institute, JAPEX on the ITIR [Intermediate Thermal Infra-red Radiometer] sensor. This sensor is to be included on the NASA polar orbital platform segment of the EOS project.

The basic remit of the current study was to assess the errors in determining a digital elevation model [DEM] for differing locational and altitude accuracies for the ITIR sensor. In addition, further studies were undertaken to assess whether these errors could be adequately compensated for, using a small number of ground control points [GCPs], in conjunction with a Powell Direction-Set relaxation scheme which has already been used successfully in the O'Neill-Dowman SPOT-1 camera model [O'Neill and Dowman, 1989b]. Tests were also conducted to determine the effect of control and check point observation errors on the overall accuracy of the models formed. These statistical tests were similar to those which were used to look at the effect of observation error in the case of the O'Neill-Dowman SPOT-1 Camera Model, [O'Neill and Dowman, 1989c].

A6.1 Methodology.

Since the ITIR mission has not yet been launched, the first problem to be overcome in producing a simulation of the sensor was the construction of an appropriate orbital segment. Because of the experience gained with the SPOT-1 sensor over the three year duration of the Alvey MMI-137 [Real Time 2.5D Vision Systems] project, it was decided that orbit segments to be used in the ITIR simulation would be derived from appropriate SPOT-1 orbit segments, which may be obtained from SPOT-1 header data. The ITIR sen-
A simulation study of the ITIR sensor itself is a derivation of the existing O’Neill-Dowman SPOT-1 camera model, adapted for along track [as opposed to across track] stereoscopy.

A6.1.1 Derivation of base ITIR orbit segment and image co-ordinates.

The derivation of accurate zero-perturbation ITIR orbit segment, and a corresponding set of image co-ordinates was crucial to the success of the present simulation study. This was accomplished in four stages.

a) Formation of a SPOT-1 model covering the region of the ITIR simulation.

b) Creation of a sparse digital elevation model [DEM] for the region of the ITIR simulation.

c) Formation of the ITIR orbit segment from the SPOT-1 orbit segment formed in (a) above.

d) Formation of simulated ITIR image co-ordinates by back transformation of points in the Sparse DEM into the image spaces of the forward and vertical ITIR sensor looks.

A6.1.2 Formation of SPOT-1 model.

The camera model used to form the SPOT-1 model for the purposes of the ITIR sensor simulation study is a modified version of the O’Neill-Dowman SPOT-1 model which was developed under the aegis of the Alvey MMI-137 project and the subsequent LSL/RAE UCL research contract. The modifications made to this basic SPOT-1 camera model under the aegis of the present ITIR sensor simulation study include:

a) Division of the camera model into distinct software modules: Although the LSL/RAE version of the O’Neill-Dowman SPOT-1 camera model does possess a modular structure, it was not deemed to be an adequate basis for a generic camera modelling system, which could form the basis for a number of similar rigid sensor models. The modular structure which was
A simulation study of the ITIR sensor evolved over the duration of the ITIR project was designed in order to provide such generic camera modelling facilities. As a result of these modifications, it was possible to program a rigorous geometrical model of the ITIR sensor which shares about 90% of the software components used in the SPOT-1 model.

b) Addition of software subsystem to permit the interchange of orbit segment and camera data between camera models based upon the generic camera modelling system outlined above. This subsystem was added to the SPOT-1 camera model to enable the orbit segments [for the two SPOT-1 looks] to be passed to the ITIR sensor model.

For the purposes of the present study, the modified SPOT-1 camera modelling software was used to form a SPOT-1 model of 60km x 60km area in Oman. This model was set up using three ground control points. The corresponding orbit segment and camera model data were then stored in a standard format for future use by the ITIR sensor model in the ITIR sensor simulation experiments.

A6.1.3 Generating the sparse digital elevation model.

In order to assess the performance of the ITIR sensor adequately, a fairly large set ground points well distributed over the region of interest is required. This dataset, the sparse digital elevation model [sparse DEM] was generated using the SPOT-1 model formed above.

Given a pair of SPOT-1 images it is possible to find a large number of stereo conjugate points, using an appropriate stereo matching algorithm. These points, or a representative subset thereof, may then be transformed to ground space using the SPOT-1 camera model corresponding to this imagery. In the case of the current simulation, the Otto-Chau Stereo Matcher [Otto and Chau, 1989] was used to find a large number of stereo-correspondences [~ 800,000 corresponding points] between the two looks of the panchromatic SPOT-1 Oman stereo-pair. A binning routine was then used to extract a representative subset of some 430 conjugate points which were evenly distri-
A simulation study of the ITIR sensor

buted over the 60km x 60km area of interest. These points were then transformed from SPOT-1 image space to object space using the SPOT-1 camera model, thus forming the sparse DEM to be used in the ITIR sensor simulation.

A6.1.4 Creation of the ITIR orbit segment.

The ITIR orbit segment was created by transformation of the standard SPOT-1 orbit segment. Firstly, the flying height must be adjusted from ~830 km typical of the SPOT-1 sensor, to the 705 km flying height proposed for the ITIR sensor:

\[ \vec{p}_{i, \text{ITIR}} = h_{i, \text{TR}} \cdot \vec{p}_{i, \text{SPOT-1}} \]  

(A6.1)

Where:

\( p_{i, \text{ITIR}} \) is the \( i^{th} \) position vector in the ITIR orbit segment.

\( p_{i, \text{SPOT-1}} \) is the \( i^{th} \) position vector in the SPOT-1 orbit segment.

\( h_{i, \text{TR}} \) is the height ratio \( \frac{h_{i, \text{TR}} + R_e}{h_{i, \text{SPOT-1}} + R_e} \).

Note that the corresponding orbit segment velocity vector does not change given that the ratio \( \frac{h_{i, \text{TR}} + R_e}{h_{i, \text{SPOT-1}} + R_e} \) approximates to unity [for practical purposes this ratio must be greater than ~0.95]. The constancy of the orbital velocity vector compared to SPOT-1 is in fact a mechanism for simplifying the computation of the ITIR orbit segment from a SPOT-1 orbit segment.

The approximation made is that the ratio \( \frac{h_{i, \text{TR}} + R_e}{h_{i, \text{SPOT-1}} + R_e} \) is close to unity. In the case of the present study, the ratio is fact 0.9827. Given this, the greater curvature of the ITIR orbit [because it is lower than the SPOT-1] orbit, may be ignored. This means that only the sensor position vectors of the SPOT-1 orbit need be amended; the velocity vectors remain unchanged in the simulated ITIR orbit segment relative to the SPOT-1 input orbit segment.
A6.1 Simulation of vertical ITIR looks from oblique SPOT-1 looks.

If the SPOT-1 orbit is pointing obliquely towards the sparse DEM with an off nadir pointing angle of greater than 9.5° degrees, it will have to be rotated in order to adequately simulate the ITIR sensor.

The orbit rotation angle $\theta$, for the $i^{th}$ ITIR sensor position vector, $\mathbf{p}_{iTIR}^i$, is given by the following expression

$$\theta = 180 - ( thi + \sin^{-1} \left( 1 + \frac{ht}{R_e} \cdot \sin thi \right) ) \quad (A6.2)$$

Where:

$\theta$ is the angle, about an axis along the corresponding velocity vector, $\mathbf{v}_{iTIR}$ through which the $i^{th}$ position vector must be rotated to achieve a vertical simulated look.

$ht$ is the flying height of the sensor above the local mean geoid.

$R_e$ is the local earth radius.

$thi$ is the off nadir look angle of the input SPOT-1 orbit segment.

The geometry of the orbit rotation process is shown schematically in Figure A6.1. The sensor reference axis triad for the ITIR sensor, $[\mathbf{v}, \mathbf{e}_l, \mathbf{p}]$, may be formed in the usual manner for push-broom sensors:

$$\mathbf{v} = vunit(\mathbf{v}) \quad (A6.3.1)$$

$$\mathbf{p} = punit(\mathbf{p}) \quad (A6.3.2)$$

$$\mathbf{e}_l = \mathbf{v} \times \mathbf{p} \quad (A6.3.3)$$

Where:

$[\mathbf{v}, \mathbf{e}_l, \mathbf{p}]$ are unit vectors in the direction of the velocity $\mathbf{v}$, position $\mathbf{p}$, and pushbroom $\mathbf{e}_l$ vectors of the sensor respectively.
\( vunit \) denotes the operation of taking the unit vector in the direction of the given argument vector.

In order to comply with the supplied ITIR camera model specifications, the camera parameters were adjusted in the following manner.

a) Pixel size changed from 10m [SPOT-1] to 15m [ITIR]

b) Field of view for forward ITIR look set to 6.09°.

c) Field of view for vertical ITIR look set to 5.19°. Parameters (b) and (c) were inferred from technical documentation on the ITIR camera optics sent to UCL to facilitate the present simulation study.

d) Along track look angle for forward ITIR look set to 37.4°, which corresponds to the projected \( \frac{B}{H} \) ratio for ITIR of 0.6.

e) Along track look angle for vertical ITIR look set to 0°, which corresponds to the projected \( \frac{B}{H} \) ratio for ITIR of 0.6.

Once the zero-perturbation ITIR orbit segment is created, the simulated ITIR image co-ordinates, which correspond to the points in the sparse DEM may be found by back transforming each point in the sparse DEM in turn to yield a pair of conjugate points in ITIR image space. The process used to effect the back transformation is identical to that used in the SPOT-1 camera model, described by O’Neill and Dowman [O’Neill and Dowman, 1989b].

A6.2 Checking the Accuracy of the Base ITIR Orbit Segment.

Once the zero perturbation orbit segment had been created, it was checked by transforming the simulated ITIR image co-ordinates back into ground space co-ordinates and then forming the vector ARMSE statistic by comparing the transformed image points with those of the original sparse DEM. The vector ARMSE is given by the expression:
\[
\text{ARMSE}^2 = \frac{1}{N_{chk}} \sum_{i=1}^{N_{chk}} | \hat{S}_i^* - \hat{G}_i^* |^2
\]

where:

\begin{align*}
\text{ARMSE} & \quad \text{is the Absolute Root Mean Square error for the ITIR model,} \\
\hat{S}_i^* & \quad \text{is the } i^{th} \text{ vector of co-ordinates produced by the intersection of the corresponding rays which forms the } i^{th} \text{ ground co-ordinate predicted by ITIR the camera model,} \\
\hat{G}_i^* & \quad \text{is the corresponding } i^{th} \text{ ground co-ordinate in the sparse DEM.} \\
N_{chk} & \quad \text{is the total number of points in the sparse DEM.}
\end{align*}

The magnitudes of ARMSE's observed for these tests was very low, yielding a mean ARMSE of 0.2 metres, indicating that the process of ITIR orbit segment production was satisfactory.

A6.3 Creation of Perturbed ITIR Orbit Segments.

In order to assess the effects of error, the orbit segment which was computed in the section above must be perturbed with appropriately defined error functions.

A6.3.1 Simulating the effect of pointing angle error.

One of the major purposes of the current study was to assess the effects of sensor pointing angle variation on the accuracy of acquired imagery. In order to do this, the ITIR orbit segment was perturbed with attitude error functions:

\[
E_{\text{roll}} = \text{roll\_drift}(t)
\]
A simulation study of the ITIR sensor

\[ E_{\text{pitch}} = \text{pitch\_drift}(t) \]  
\[ E_{\text{yaw}} = \text{yaw\_drift}(t) \]

Where:

- \( E_{\text{roll}} \) is the drift about the sensor roll axis at time \( t \),
- \( E_{\text{pitch}} \) is the drift about the sensor pitch axis at time \( t \),
- \( E_{\text{yaw}} \) is the drift about the sensor yaw axis at time \( t \).

A schematic of these axes is shown in Figure A6.2. The three drift angles given in (A6.5.1), (A6.5.2) and (A6.5.3) may be estimated from a set of variogram functions for these quantities. The empirical form of such variogram functions has been investigated in the case of the SPOT-1 sensor by Westin [Westin, 1990]. Because the orbit segment used by the ITIR simulation is derived from a SPOT-1 orbit, it was decided to use Westins’ empirical variogram models to provide a realistic set of pointing model errors for the simulations. The variogram models have the following algebraic forms:

\[ 2\gamma(t \Delta t)_{\text{roll}} = 1774 \Delta t^{1.8} \]  
\[ 2\gamma(t \Delta t)_{\text{pitch}} = 17500 \Delta t^{0.5} + 80000 (1 - \cos(0.93 \Delta t)) \]  
\[ 2\gamma(t \Delta t)_{\text{yaw}} = 3200 \Delta t \]

Where: The variograms are expressed in the unit \([\mu\text{deg}]^2\), and \( \Delta t \) is in seconds. We note that \( \gamma \) is a function of time.

The variogram models, plotted on top of the computed variograms are shown in Figure A6.3. Since the variogram functions, by definition, give the roll, pitch, and yaw error variances at time \( t \), the corresponding pointing angle errors may be determined by taking the square roots of the variogram functions given in (A6.6.1), (A6.6.2) and (A6.6.3) above:

\[ E_{\text{roll}} = \sigma \cdot \sqrt{V_{\text{roll}}} \cdot \text{sgn}(t) \]  

\[ E_{\text{pitch}} = \sigma \cdot \sqrt{V_{\text{pitch}}} \cdot \text{sgn}(t) \]  
\[ E_{\text{yaw}} = \sigma \cdot \sqrt{V_{\text{yaw}}} \cdot \text{sgn}(t) \]
\[ E_{\text{pitch}} = \sigma \cdot \sqrt{V_{\text{pitch}}} \cdot \text{sgn}(t) \] (A6.7.2)

\[ E_{\text{yaw}} = \sigma \cdot \sqrt{V_{\text{yaw}}} \cdot \text{sgn}(t) \] (A6.7.3)

Where:
- \( \sigma \) is a scalar parameter,
- \( \text{sgn} \) is the signum function, meaning take the sign of \( \text{abs}(t) \).
- \( V \) is the appropriate [roll, pitch or yaw] variogram function.

The three attitude errors determined from (A6.7.1), (A6.7.2) and (A6.7.3) may then be used to form a small angle perturbation matrix, \( E (\sigma, t) \) [Thompson, 1969]. This matrix may then be used to perturb the zero-perturbation ITIR sensor attitude matrix \( R_o \), producing the perturbed ITIR sensor attitude matrix, \( R_p(t) \), at time \( t \):

\[ R_p(t) = R_o \cdot E (\sigma, t) \] (A6.8)

Where:
- \( E (\sigma, t) \) is the perturbation matrix computed from the attitude variograms at time \( t \).

A6.3.2 The precise meaning of time \( t \).

It is important that we define precisely what we mean by time \( t \) in context of simulating pointing error using variogram models, as the observed pointing error is a function of a starting time \( \tau \) which may be arbitrarily assigned. For the purposes of the current simulation, \( \tau \) was defined to be the time that line 2000 [the mid-point of the vertical ITIR look] was acquired. Thus, all variogram model times have been measured relative to this starting point.

A6.3.3 Estimation of sensor pointing error from \( \sigma \).

Estimates of the effects of the ITIR sensor pointing errors were then computed by perturbing the ITIR orbit segment using (A6.8) over a time period \( T \) [during which all the sparse DEM points were imaged by both the vertical and forward looks of the ITIR sensor]. This perturbed orbit segment...
was then used in the ITIR sensor model when transforming the simulated ITIR image co-ordinates from image space to ground space. The ARMSE of the points transformed using the perturbed orbit was then computed using (A6.4).

Since the perturbation matrix $E$ is a function of the scalar parameter $\sigma$ as well as time, it is possible to compute the ARMSE as a function of the parameter $\sigma$. Furthermore, (A6.8) may be factorised into component roll, pitch and yaw perturbation matrices. This means that the individual roll, pitch and yaw ARMSE statistics may be readily computed as functions of their respective $\sigma$ parameters within the time period of interest, $T$.

The mean angular pointing error associated with a particular value of $\sigma$, may be estimated by integrating the appropriate variogram function over the acquisition times of the forward and vertical images:

$$< \theta > = \left[ \frac{1}{|v_2 - v_1|} \int_{v_1}^{v_2} v(t) \, dt + \frac{1}{|f_2 - f_1|} \int_{f_1}^{f_2} v(t) \, dt \right]^{0.5} \quad (A6.9)$$

Where:

$< \theta >$ is the expectation value for the angular pointing error,
$v_1, v_2$ are the limits of integration for the vertical.

$f_1, f_2$ are the limits of integration for the forward look. Thus $v_1$ is the time at which the forward sensor starts to scan the DEM. $v_2$ is the time at which the forward sensor stops scanning the DEM,
$v(t)$ is an appropriate variogram function.

A6.4 Assessment of Accuracy.

The accuracy in determining ground co-ordinates using the methods and data given are assessed by using ground control points as check points. These results are given in section A6.6.
A6.5 Simulation of Observation Error.

In addition to looking at pointing angle error, the effect of observation errors in the ground control and checkpoint data was also considered in the present study. Previous experience [O’Neill and Dowman, 1989c] has shown that in many cases, observation errors are often the most significant obstacle to the creation of accurate sensor models for complex space borne sensors, such as SPOT-1 or ITIR. The method of simulating observation errors adopted for the ITIR sensor simulation study was an adaptation of the method used to monitor these errors in the case of the SPOT-1 camera model.

The basis of this method is to use a random number generator to fuzz the parameter which is to be subjected to observation error. For the purposes of both SPOT-1 and ITIR observation error measurement, a simple muffin tin fuzzing function [Muller et al, 1990a; Muller et al, 1990b] using white-noise random deviates was deemed sufficient. However, the use of a more realistic observation error distribution, for example Gaussian or Poisson distribution is possible, with little modification to the basic technique. The form of the muffin tin fuzzing function is given below:

\[ a' = \rho \cdot (\text{rnd} - 0.5) + a \]  

(A6.10)

Where:

- \( a' \) is the fuzzed parameter,
- \( a \) is the input parameter,
- \( \rho \) is the maximum uncertainty radius [muffin-tin radius],
- \( \text{rnd} \) is a generator of white noise random deviates in the range \( 0 < \text{rnd} \leq 1 \). \( \text{rnd} \) may be replaced by another source of deviates, for example, a Gaussian random deviate generator, in order to achieve a more realistic distribution of observation errors.

In order to assess the effects of observation error on an observable parameter \( a \) [\( a \) could be for example the line position of a ground control feature within the ITIR image], the following procedure was adopted: for each uncertainty radius \( \rho \), the mean of the ARMSE for 100 models was computed, in order to achieve a good measure of statistical significance. The above process was then repeated for a number of error-radii ranging between...
a maximum error radius $r_{\text{max}}$, characteristic of observation error for the particular parameter $a$, and zero. Table 1 shows typical values for $r_{\text{max}}$ for the parameters which were subjected to observation error in the current ITIR simulation study.

A6.6 Assessment of the Use of Ground Control.

In addition to looking purely at the sources of probable error in the ITIR sensor system, the present simulation study also looked at how the errors may be reduced using a small number of ground control points. These ground control points were selected in a similar manner to those for SPOT-1 models, being well distributed over the image, and possessing a wide separation in sample space. The ground control points were used to drive a Powell direction-set optimisation scheme [Powell, 1964; Acton, 1970; Brent, 1973], which is identical to that used in the O’Neill-Dowman SPOT-1 camera model.

The effect of 2 ground control points on reducing the error induced by roll pitch and yaw sensor drift was assessed.


It is assumed that the satellite orbit is known without error, and that the absolute acquisition time for each line is also known.

It has been shown with SPOT-1 data that the absolute position of the orbit does not effect the model, that is the relative position of features can be found. This assumption is more certain with an along track stereo system as only one orbit is involved. Any small irregularities in the orbit will be correlated with altitude changes and hence will have no effect. Any gross absolute errors will be corrected for with the use of ground control points.

Errors in timing will cause errors in height and this parameter must be included in the satellite model for along track stereo. A study of height error was not in the remit of the current study, so error free timing has been as-
A simulation study of the ITIR sensor

A6.8 Jitter.

Jitter is assumed to be high frequency attitude change. It is quoted by JAPEX to be 1.08 arcsec/sec or 3.69 m/sec on the ground. It will not be cumulative but will reduce the internal accuracy of a single image. Effectively, it will reduce the precision to which image co-ordinates can be given rather than the accuracy with which the model can be formed. Experience with SPOT-1 data has indicated that jitter will not give rise to significant errors in DEM generation. The effect of jitter may well be more serious in the case of the ITIR sensor than with SPOT-1, as the polar EOS platform is considerably larger than the SPOT-1 platform, thus lowering the frequency of the normal vibrational modes of the platform.

A6.9 Results.

Simulation studies for nadir and off nadir angles of 9.0° and 4.5° of the following sources of error were conducted:

a) The effects of sensor stability about the roll, pitch and yaw axes on observed ARMSE for a sparse DEM of 430 points, well distributed over a single 60km x 60km area in Oman, based on data stereomatched from SPOT-1 model. No ground control is used.

b) The effect of two ground control points with the vector data in (a) on suppressing sensor roll, pitch and yaw error.

c) The effect of observation error in the ground and image co-ordinates of the points comprising the sparse DEM on the observed ARMSE for the ITIR sensor model. No ground control is used.
d) The effect of two ground control points on stabilising the effect of observation error.

A6.10 The Effect of Sensor Pointing Error on Observed ARMSE.

The effects of sensor pointing error without ground control on observed ARMSE are shown in Figures A6.4, A6.5 and A6.6 [vertical look], A6.7, A6.8 and A6.9 [4.5° nadir look] and A6.10, A6.11 and A6.12 [9° off nadir look]. In each case, the observed ARMSE is an almost linear function of σ for large |σ|. For small σ, these functions become increasingly parabolic, with a minimum at σ = 0. As one might expect from the form of the variograms used to derive the pointing errors, the effect of roll error is substantially greater than that of pitch and yaw error. This of course, assumes that the characteristics of platform stability for ITIR are similar to SPOT-1.

The mean pointing angle error as a function of σ is given by the following equations which are derived from the numerical integration of (A6.9).

\[
< \theta_{v\,\text{roll}} > = 0.000001 \cdot \sigma_{\text{roll}} \\
< \theta_{v\,\text{pitch}} > = 0.000005 \cdot \sigma_{\text{pitch}} \\
< \theta_{v\,\text{yaw}} > = 0.000001 \cdot \sigma_{\text{yaw}} \\
< \theta_{f\,\text{roll}} > = 0.000033 \cdot \sigma_{\text{roll}} \\
< \theta_{f\,\text{pitch}} > = 0.000008 \cdot \sigma_{\text{pitch}} \\
< \theta_{f\,\text{yaw}} > = 0.000008 \cdot \sigma_{\text{yaw}}
\]

(A6.11.1)  
(A6.11.2)  
(A6.11.3)  
(A6.11.4)  
(A6.11.5)  
(A6.11.6)

Where:
- \( f \) denotes the forward look,
- \( v \) denotes the vertical look,
A simulation study of the ITIR sensor

roll, pitch, yaw denote the roll, pitch and yaw axes respectively,
$\langle \theta \rangle$ is the appropriate mean pointing angle drift,
$\sigma$ a sigma value corresponding to some ARMSE for which the corresponding mean pointing angle drift is required.

The root mean square error values in Figures A6.4-A6.12 for 2 sigma and 3-sigma have been extracted and converted to attitude errors and the results shown in table A6.1. The attitude errors given are absolute errors which result from attitude changes predicted by the variogram produced for SPOT-1 data. As would be expected from the foregoing description the attitude error is greater as the sensor moves from its assigned starting position. The resulting RMS error comes from a combination of the attitude error from the forward and nadir pointing sensor.

In addition, some further plots have been produced. These give the ARMSE as a function of the biggest sensor pointing error, which is that for the forward looking sensor. Figures A6.24-A6.29 give the individual ARMSE components for the roll, pitch and yaw axes; Figures A6.30-A6.32 give the total ground error as a function of an angular error which is the vector magnitude of the roll, pitch and yaw errors. These figures give the results for the nadir look only, as the differences between the results presented for the nadir look and the two side looking cases were found to be negligible. All of these results are produced without using ground control.

A6.11 The Effect of 2 GCP's on Reducing Pointing Angle Error.

To assess the effect of ground control in reducing the observed ARMSE due to pointing angle errors, the above experiments were repeated for a 4.5° side look, with a two ground control point relaxation. These simulation were only carried out for the 4.5° side pointing case, as there is no difference shown in table A6.1 for side looking cases. Two cost functions \( gcpd \) and \( rrs \) were used in order to determine whether there were any differences.1 These cost functions are described by O'Neill, [Muller et al 1990a; Muller et al, 1990b]. The results of these tests are show in Figures A6.13-A6.18. As it may be seen, an essentially linear absolute orientation scheme using just two
GCP's greatly reduces the observed ARMSE, from, in the worst case, several hundred metres of error down to less than 5m at the 3\(\sigma\) level. The form of the 2-GCP curves is, like the 0-GCP curves, linear in the region |\(\sigma\)| \(\gg\) 0. For the region |\(\sigma\)| = 0, the curve is essentially parabolic, but there are a number of spikes, which are more noticeable in the case of the gcpd cost function. These are due to the multiple local ARMSE minima surrounding the global minimum of the ARMSE, causing erroneous relaxation by the unimodal relaxation scheme used in the ITIR camera model.

A6.12 Effect of Observation Error on Observed ARMSE.

The effect of image and object space observation error on the observed ARMSE is shown in Figures A6.19 [image space observation error] and A6.20 [object space observation error], respectively. These experiments were conducted with the 4.5° off-nadir look only; but very similar results would be expected in the case of the 9° off-nadir and nadir looks. The observed ARMSE rises steadily in an almost linear fashion as the magnitude of the mean observation error rises.

A6.13 Effect of Ground Control on Reducing the Effect of Observation Error.

Significantly, an absolute orientation scheme will not improve a model where the principal source of error is observation error, as illustrated in Figures A6.21 and A6.22 [4.5° off nadir look, rrs cost function]. The reason for this is quite simple. Observation errors are stochastic in nature, and it is exceedingly difficult to persuade an essentially linear optimisation scheme to remove errors of this type. This sort of error also proved to be both prevalent and difficult to remove in the real SPOT-1 models which were tested under the LSL/RAE UCL subcontract, which preceded this study. In the case of the O'Neill-Dowman SPOT-1 camera model, the minimum ARMSE attained, using GPS was about 9-10 metres. The bulk of this error [70%+] may be attributed to image space observation error: with 10m pixels, it is very hard to reliably locate ground control features in image space [Peacegood, 1989].
the case of the ITIR sensor, with a 15m pixel size, this situation is likely to be worse, with a best-case single scene ARMSE of about 12-18m, and an average-case ARMSE of about 30m. In order to reduce the effect of image-space observation error, it would be worthwhile to look at the pull in effect on manually measured GCP's of a Stereo Matcher such as the Gruen adaptive least squares correlator [Gruen, 1985]. Alternatively, the use of non-linear, multimodal relaxation schemes within the camera model, for example, a simulated annealing scheme, with appropriate cost functions, are also worthy of investigation.

A6.14 Conclusions.

A simulator for the ITIR sensor has been built, using SPOT-1 orbital data and a generic sensor modelling system based on the O'Neill-Dowman SPOT-1 camera model. The results of the simulation studies conducted with this model indicate that use of a small number of ground control points is well able to compensate for the effects of pointing angle error, giving worse-case ARMSE's of < 10m, given raw data whose ARMSE is >300m. Observation errors present a more serious problem, unless steps are taken to eliminate them, prior to running the model.

The results can be summarised as follows:

- Based on data derived from SPOT-1, the attitude error introduced over the period of imaging the forward look and the nadir look will amount in roll to 10" at the 2 $\sigma$ level, and 15" at the 3 $\sigma$ level, and to about 3" and 5" in the 2 and 3 $\sigma$ levels in pitch and yaw.

- With no ground control, the sensor positions can be reconstructed to give RMS errors on the ground 90m at 2 $\sigma$ in roll, 10m in pitch and 14m in yaw.

- Side pointing makes no significant difference to the results.

- If two ground control points are used and the model relaxed using the O'Neill-Dowman method, the errors reduce to less than 5m in roll, and insignificant amounts in pitch and yaw.
The addition of observation errors to the simulated observation increases the error so that if observation errors in the image of two pixels is present, the ARMSE is about 20m when ground control is used.

Two schemes are outlined by which observation errors may be eliminated. The first is a pre-filter which uses a Gruen adaptive least squares correlator to reduce the effect of image space observation error before the ITIR camera model is run. This method assumes that the object space positions of the ground control points have been measured accurately, using GPS. The second method uses a multimodal relaxation scheme in conjunction with an appropriate cost function, for example conjugate skewness, to reduce observation error in the camera modelling stage. This is likely to be much more computationally intensive than the pre-filtering method, but is capable in theory of removing object-space, as well as image-space observation error.

The generic camera modelling system which was developed to facilitate modelling of the ITIR sensor is a general purpose system for modelling rigid sensors. The simulator which was developed for ITIR may be used with little modification to simulate and/or model other rigid linear push-broom sensors, for example MISR, MEOSS or ASAS.

A6.15 Recommendations.

The following recommendations are to be made:

a) A study of the effect of orbit positional error is made for the sake of completeness.

b) The study of observation error, and the possible means of removing it is continued.

c) A study is made of the effect of high frequency effects, such as sensor jitter. This is of special relevance to ITIR given the comments made in section A6.8.

d) The effect of varying the base:height and distribution of control points could be studied.
Endnotes to appendix 6

1: These are the RGCPD and RRSKEW cost functions described in chapter 7 of this thesis.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Maximum error radius (R max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line position in image</td>
<td>3 pixels</td>
</tr>
<tr>
<td>Sample position in image</td>
<td>3 pixels</td>
</tr>
<tr>
<td>X ground GCP co-ordinate</td>
<td>75 metres</td>
</tr>
<tr>
<td>Y ground GCP co-ordinate</td>
<td>75 metres</td>
</tr>
<tr>
<td>Z ground GCP co-ordinate</td>
<td>75 metres</td>
</tr>
</tbody>
</table>

Table A6.1. Showing the maximum error radii for parameters subjected to the observation error fuzzing process.
<table>
<thead>
<tr>
<th>Side Look Angle (degrees)</th>
<th>Sigma Level</th>
<th>Roll</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Attitude Error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(seconds)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pitch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(seconds)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yaw</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Attitude Error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(seconds)</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0.4 10</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.6 15</td>
</tr>
<tr>
<td>4.5</td>
<td>2</td>
<td>0.4 10</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.6 15</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>0.4 10</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.6 15</td>
</tr>
</tbody>
</table>

V = nadir telescope  
F = forward telescope

Table A6.2 Errors at ground points for attitude errors at 2 sigma and 3 sigma levels.
Figure A6.1. Showing how the geometry of an ITIR vertical look orbit may be synthesised from an oblique SPOT-1 look.
Figure A6.2. Showing the geometry of the ITIR sensor
Figure 4.3. Showing the empirical form of SPOT-1 attitude variograms [After Westin].
ITIR sensor simulation [9.0 degree off nadir]
Ground error as function of yaw variogram sigma

Figure A.4

Absolute ground error/metres

Variogram sigma
ITIR sensor simulation [9.0 degree off nadir]
Ground error as function of pitch variogram sigma

Figure 4.5

Variogram sigma

Absolute ground error/metres
ITIR sensor simulation [9.0 degree off nadir]
Ground error as function of roll variogram sigma

Figure 4.58
Absolute ground error/metres

Variogram sigma
ITIR sensor simulation [4.5 degree off nadir]
Ground error as function of yaw variogram sigma

Figure A1.7

Variogram sigma

Absolute ground error/metres
ITIR sensor simulation [4.5 degree off nadir]
Ground error as function of pitch variogram sigma

Figure 8

Absolute ground error/metres

Variogram sigma
ITIR sensor simulation [4.5 degree off nadir]

Ground error as function of roll variogram sigma

Figure A.9

Absolute ground error/metres

Variogram sigma
ITIR sensor simulation [nadir]
Ground error as function of yaw variogram sigma
ITIR sensor simulation [nadir]
Ground error as function of pitch variogram sigma

Figure 11

Variogram sigma

Absolute ground error/metres
ITIR sensor simulation [nadir]
Ground error as function of roll variogram sigma
ITIR sensor simulation: 4.5 degree off nadir look, GCPD cf
Showing the effect of relaxation on predicted roll error
ITIR sensor simulation: 4.5 degree off nadir look, GCPD cf
Showing the effect of relaxation on predicted pitch error
ITIR sensor simulation: 4.5 degree off nadir look, GCPD cf
Showing the effect of relaxation on predicted yaw error
ITIR sensor simulation: 4.5 degree off nadir look, RRS cf
Showing the effect of relaxation on predicted roll error
ITIR sensor simulation: 4.5 degree off nadir look, RRS cf
Showing the effect of relaxation on predicted pitch error

Figure A17

Abs. RMS error/metres

Variogram sigma parameter
ITIR sensor simulation: 4.5 degree off nadir look, RRS cf
Showing the effect of relaxation on predicted yaw error

Figure A6.18

Absolute RMS error/metres

Variogram sigma parameter
ITIR sensor simulation 4.5 degree off nadir look, RRS cf
Showing effect of image observation error [without ground control]
ITIR sensor simulation 4.5 degree off nadir look, RRS cf
Showing effect of object observation error [without ground control]
ITIR sensor simulation 4.5 degree off nadir look, RRS cf
Showing effect of image observation error [with ground control]
ITIR sensor simulation 4.5 degree off nadir look, RRS cf

Showing effect of object observation error [with ground control]
ITIR model - sparse DEM pt distribution

Figure Ab.23

Eastings UTM

Northings UTM

metres

key
- cpt

0.0
229.0
458.0
686.0
917.0
1546.0
1775.0

30000.0
32000.0
34000.0
36000.0
38000.0
40000.0
42000.0
44000.0
46000.0
48000.0
50000.0
52000.0
54000.0
56000.0
58000.0
60000.0
62000.0
64000.0
66000.0
68000.0
70000.0
72000.0
74000.0
ITIR sensor simulation [nadir]
Plan error as a function of roll angle

Figure 6.24

Plan error in metres

Roll angle x 1000000 in radians
ITIR sensor simulation [nadir]
Height error as a function of roll angle

Figure A.25

Height error in metres

Roll angle x 1000000 in radians
ITIR sensor simulation [nadir]

Plan error as function of pitch angle

Figure A6.26
ITIR sensor simulation [nadir]
Height error as function of pitch angle

Figure 46.27

table 480

height error in metres

pitch angle x 1000000 in radians
ITIR sensor simulation [nadir]
Plan error as function of yaw angle

Plan error in metres

yaw angle x 1000000 in radians
ITIR sensor simulation [nadir]
Height error as function of yaw angle

Figure A6.2a

Height error in metres

yaw angle $\times 1000000$ in radians
ITIR sensor simulation [nadir]

Plan error as function of yaw, pitch and roll error

Plan error in metres

Angular error x 100000 in radians

Figure A6.30
ITIR sensor simulation [nadir]
Height error as function of yaw, pitch and roll error