Orientation and Calibration of Long Focal Length Cameras in Digital Close-Range Photogrammetry

Christos Stamatopoulos
Department of Infrastructure Engineering
The University of Melbourne

Submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

June 2011
Abstract

Digital close-range photogrammetry has traditionally been associated with measurements over short-to-medium distances. There are various applications that could benefit from the use of photogrammetry over very short or long distances. To acquire images over such distances, lenses of very long focal length are required. However, their application has been limited for a number of technical reasons. The aim of the research undertaken for this PhD has been to develop the mathematical models, algorithms, computational procedures and operational methodologies necessary to facilitate photogrammetric measurements using long focal length cameras. Notable developments achieved within the research include the identification and treatment of the numerical issues presented in the collinearity equations when long focal length lenses are employed, the formulation of an innovative orthogonal projection model that supports self-calibration of long focal length cameras; and an analysis of and improvements in the image formation pipeline, the results of which have a potentially significant impact on all applications of vision metrology.

Following a brief overview of the major mathematical procedures of close-range photogrammetry to build a basic framework for the subsequent chapters, the thesis presents an in-depth analysis and solution techniques for least squares estimation problems with the focus being on ill-conditioned systems of equations. A detailed investigation on the image formation process follows. RAW images are then exploited in the design of a new image formation methodology, which facilitates improved integrity of image point positions. The development of a new photogrammetric simulation system is then
presented. This allows the creation of simulated datasets that can aid in the testing of the developed mathematical models for the photogrammetric measurement using long focal length lenses. In seeking to overcome numerical issues and ill-conditioning of the photogrammetric normal equations system presented with the use of long focal length lenses, it is demonstrated that the instability in the solution via traditional self-calibrating bundle adjustment of narrow-angle photography can be attributed to an incomplete formulation of the partial derivatives of the extended collinearity equations with respect to the principal point parameters. Thus, a self-calibration model with expanded coefficients is presented and evaluated with both simulated and real photogrammetric networks. Finally, an new orthogonal projection model for self-calibration of narrow field of view cameras is developed to accommodate high accuracy close-range photogrammetry with very long focal length lenses. A fully automatic bundle adjustment process incorporating the orthogonal projection model has been developed and its performance has been experimentally evaluated. This new formulation provides an enhanced precision of the recovered object point coordinates, compared to the collinearity equations approach.
Declaration

This is to certify that

i. This thesis comprises only my original work towards the PhD;

ii. Due acknowledgment has been made in the text to all materials used;

iii. This thesis is less than 100,000 words in length exclusive of tables, figures, appendices and references.

Christos Stamatopoulos
Acknowledgements

Above all, I would like to formally thank the finest PhD supervisors one could hope for, Professor Clive Fraser and Dr. Simon Cronk. They have been amazing mentors and have inspired me to complete a PhD without having to give up on other commitments and passions. Their guidance, sound technical advice and endless support are greatly appreciated.

To my parents, I would like to say a most sincere thank you for their continual encouragement and support, and for genuine interest shown in my work throughout the course of this PhD. I also have the best brother in the world, so special thanks should be extended to him.

I would also like to thank all of my friends at university, Eldar Rubinov, Grant Hausler, Ida Jazayeri, Sheelan Vaez and Fabrizio Girardi for their friendship and all the great memories in the penthouse. I specially enjoyed the non-technical talks which created an amazing environment in which to work on a daily basis.

I must also extend my special thanks to my mates back in Greece, Grigoris Skouras, Iordanis Kasapoglou, Angelos Tsakiroglou, Dimitrios Mallis, Akis Karfopoulos, Filippos Zisopoulos and Yannis Patras for always being there for me. On a similar note, I would also like to thank my friends that I train with here in Melbourne, especially Leonardo Pimentel and Eamonn McCallum, who helped me continue my passion of capoeira and for our awesome times together. Training on a daily basis helped me clear my mind of research related issues, and without it my sanity would not have been retained.
## Contents

List of Figures xi

List of Tables xiv

List of Algorithms xv

Glossary xvi

1 Introduction 1
   1.1 Statement of the Problem 3
   1.2 Research Approach 3
   1.3 Thesis Organisation 5

2 Close-Range Photogrammetry 7
   2.1 Introduction 7
   2.2 Analytical Restitution 8
      2.2.1 Collinearity Equations 9
   2.3 Departures from Collinearity 10
      2.3.1 Non-physical Perturbations 10
      2.3.2 Radial Distortion 11
      2.3.3 Decentring Distortion 13
      2.3.4 Linear Distortion 13
   2.4 Relative & Exterior Orientation 14
   2.5 Bundle Adjustment 17
   2.6 Automation 20
## CONTENTS

2.6.1 Coded and Single Target Identification .................................. 22

### 3 Least Squares Estimation

3.1 Introduction  ................................................................. 24
3.2 Gauss-Markov Model ......................................................... 26

#### 3.2.1 General Least Squares .............................................. 28

3.3 Computing the Least Squares Solution ..................................... 31

#### 3.3.1 Triangular Systems .................................................... 32

- 3.3.1.1 Gaussian Elimination and LU Factorisation .............. 32
- 3.3.1.2 Cholesky Decomposition ............................... 34
- 3.3.1.3 Error Analysis for the Triangular Factorisation Methods ............................................. 36

#### 3.3.2 Orthogonal Decomposition ........................................... 36

- 3.3.2.1 Householder Reflection and Givens Rotations .......... 37
- 3.3.2.2 Gram-Schmidt ........................................... 39
- 3.3.2.3 The QR Factorisation ................................. 40
- 3.3.2.4 Eigenvalue and Schur Decompositions .............. 42
- 3.3.2.5 The Singular Value Decomposition .................. 43

#### 3.3.3 Summary of Least Squares Computation Methods ............... 47

### 4 Digital Image Formation and Formatting ................................. 49

4.1 Introduction ............................................................... 49
4.2 Camera Sensors ............................................................ 50

#### 4.2.1 Pixel Colour Formation Process ................................ 54

4.2.2 True Colour Formation ................................................. 56

4.3 Image Creation Process .................................................. 57

#### 4.3.1 Black Point Subtraction .......................................... 58

#### 4.3.2 Bad Pixel Removal .................................................. 58

#### 4.3.3 Dark Frame, Bias Subtraction & Flat-Field Correction .... 59

#### 4.3.4 Green Channel Equilibrium Correction ...................... 60

#### 4.3.5 De-noising ............................................................ 62

#### 4.3.6 Colour Scaling ....................................................... 64

#### 4.3.7 Bayer Interpolation .................................................. 65

vii
5.4 Implementation of Image Coordinates Files

6 Long Focal Length Camera Calibration using the Central Perspective Model
   6.1 Introduction
   6.2 Problem Identification
   6.3 Calibration Model
   6.4 Numerical Stability
      6.4.1 Implementation of the SVD
      6.4.2 Solution by Direct Decomposition of the Design Matrix A
      6.4.3 Solution via Normal Equations
         6.4.3.1 Modified Standard Fold-in
         6.4.3.2 Modified Reverse Fold-in
   6.5 Testing Regime
   6.6 Partial Derivatives
   6.7 Experimental Evaluation
      6.7.1 Preliminary Testing
      6.7.2 Self-calibration Networks
         6.7.2.1 Case 1 - Lens of 300mm Focal Length
         6.7.2.2 Case 2 - Lens of 400mm Focal Length
         6.7.2.3 Case 3 - Macro Lens of 135mm Focal Length
   6.8 Concluding Remarks

7 Long Focal Length Camera Calibration using an Orthogonal Projection Model
   7.1 Introduction
   7.2 Orthogonal Projection Model
      7.2.1 Derivation of the Orthogonal Projection Model
      7.2.2 Transformation from Central Perspective to Affine Projection Coordinates
      7.2.3 Relationship between Orientation Parameters of the Central Perspective and Orthogonal Projection Models
   7.3 Photogrammetric Orientation Procedure
7.3.1 Bundle Adjustment without Self-Calibration . . . . . . . . 131
7.3.2 Bundle Adjustment with Self-Calibration . . . . . . . . 134
7.3.3 Experimental Results and Discussion . . . . . . . . . . . . 137
7.3.3.1 Case 1 - Lens of 300mm Focal Length . . . . . . 137
7.3.3.2 Case 2 - Lens of 400mm Focal Length . . . . . . 138
7.4 Concluding Remarks . . . . . . . . . . . . . . . . . . . . . . . . 139

8 Conclusions 141

Author’s Published Work 144

References 145

A Appendix 150
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Relationship between image and object point coordinates</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Interior orientation</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Exterior orientation parameters</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Coplanarity</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>Multi-station bundle adjustment</td>
<td>18</td>
</tr>
<tr>
<td>2.6</td>
<td>Flowchart of a fully automatic exterior orientation process using coded targets</td>
<td>21</td>
</tr>
<tr>
<td>2.7</td>
<td>Retro-reflective coded targets</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>RGB imaging sensors</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>(a) Beam-splitter based</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>(b) CFA based</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Schematic representation of a Bayer RGB sensor</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>Similarity of film and Foveon X3 sensor</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>(a) Film</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>(b) Foveon</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Schematic representation of a Foveon sensor</td>
<td>53</td>
</tr>
<tr>
<td>4.5</td>
<td>Schematic representation of a Bayer RGB sensor with antialiasing filter</td>
<td>54</td>
</tr>
<tr>
<td>4.6</td>
<td>IR Cut Filter</td>
<td>54</td>
</tr>
<tr>
<td>4.7</td>
<td>Comparison of Bayer RGB with Foveon X3 in: (a,c) sensor level, (b,d) pixel level</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>(a) Close-up on the Bayer matrix. Here, 12 photosites, coloured for the needs of the illustration</td>
<td>55</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

(b) Bayer RGB pixel information ........................................ 55
(c) Foveon X3 sensor ...................................................... 55
(d) Foveon X3 pixel information .......................................... 55
4.8 Test pattern for Green channel equilibration ....................... 61
4.9 Plots of RAW data along three gradients ........................... 61
(a) Black - Red gradient .................................................. 61
(b) Black - Green gradient .............................................. 61
(c) Black - Blue gradient ................................................ 61
4.10 Gamma correction curve ............................................ 68
4.11 De-gamma correction curve ......................................... 69
4.12 The four possible variations of the Bayer RGB sensor ............ 72
4.13 Object target array and camera configuration for the dataset taken with the Nikon D200 ............................................. 76
4.14 Object target array and camera configuration for the dataset taken with the GSI INCA3 ............................................. 78
5.1 The user interface of the developed photogrammetric simulator . 92
5.2 Camera list in the simulator software ................................. 93
5.3 Example of tile rendering ............................................ 94
5.4 Camera calibration model used in the simulator software ........... 96
5.5 A coded target in the simulator environment ....................... 97
5.6 Bi-linear pixel value interpolation in 2 x 2 neighbourhood ........ 98
6.1 Normal equations matrix pattern ..................................... 107
6.2 7-station, 21-image network configuration for Case 1, 300mm focal length ............................................................... 115
6.3 7-station, 21-image network configuration for Case 2, 400mm focal length ............................................................... 117
6.4 13-station, 39-image network configuration for combined Case 2, 400mm focal length ............................................... 118
6.5 Example of a coded target affected by chromatic aberration ....... 118
6.6 27-image network configuration for Case 3, macro lens network, 135mm focal length ....................................................... 119
LIST OF FIGURES

7.1 Constant scale parameter $s$ ........................................ 125
7.2 Flowchart for automated orientation procedure based on the orthogonal projection model ........................................ 129
List of Tables

3.1 Computational cost of least squares algorithms ................. 47

4.1 Values of the Green channel (12-bit) for the central part of a retro-
reflective target ......................................................... 62
4.2 Estimates of mean positional standard error for the INCA3 network 77
4.3 Estimates of the mean positional standard errors for Green channel 79
4.4 Estimates of mean positional standard error for JPEG dataset
using all channels for centroiding ................................. 80
4.5 Estimates of the mean positional standard errors for Red channel 82
4.6 Estimates of the mean positional standard errors for Blue channel 83
4.7 Estimates of the mean positional standard errors using the com-
bined bundle approach ........................................... 85

6.1 Network characteristics for a subset of the simulated networks .. 102
6.2 Standard errors of object point coordinates for Case 1 .......... 116
6.3 Standard errors of object point coordinates for the 39-image
combined network of Case 2 ......................................... 117
6.4 Standard errors of object point coordinates for the 27-image macro
lens network ......................................................... 120

7.1 Camera calibration results for Case 1, 300mm lens ............... 137
7.2 Standard errors of object point coordinates for Case 1 .......... 138
7.3 Standard errors of object point coordinates for Case 2 .......... 138
List of Algorithms

5.1 Distortion addition algorithm pseudo code for OpenGL . . . . . . 95
Glossary

2D Two Dimensional
3D Three Dimensional
AA Anti-Aliasing
API Application Programming Interface
CCD Charged Couple Device
CFA Colour Filter Array
CGI Computer Generated Imagery
CMOS Complementary Metal Oxide Semi-conductor
CPU Central Processing Unit
DLT Direct Linear Transformation
DSLR Digital Single-Lens Reflex
EO Exterior Orientation
EXIF Exchangeable Image File Format
FLOP Floating point Operations
FoV Field of View

GPGPU General-Purpose computation on Graphics Processing Units
GPU Graphics Processing Unit
GSL GNU Scientific Library
GUI Graphical User Interface
HVS Human Vision System
ICC International Colour Consortium
ICF Infrared Cut Filter
IO Interior Orientation
JPEG Joint Photographic Experts Group
LS Least Squares
LZW Lempel-Ziv-Welch
MGS Modified Gram-Schmidt
OpenCL Open Computing Language
OpenGL Open Graphics Library
RO Relative Orientation
SLR Single-Lens Reflex
SVD Singular Value Decomposition; a ma-trix decomposition of the form $A = UV^T$
TIFF Tagged Image File Format
UI User Interface
Close-range photogrammetry, as the name suggests, has been traditionally limited to short to medium camera-to-object distances. With the growing use of off-the-shelf digital cameras for photogrammetric measurement, however, requirements are emerging to perform measurements over: a) long distances, for applications in construction engineering, deformation monitoring and traffic accident reconstruction; as well as b) very short distances for applications such as digital documentation and 3D visualisation of cultural heritage objects via image based approaches. Such measurements often require the use of long focal length lenses both to keep the spatial resolution high and to optimise the angular measurement precision. It has long been recognised that there can be practical photogrammetric impediments to the employment of very long focal length lenses with small format cameras, these centering principally upon potential difficulties in analytical orientation and subsequently self-calibration. As focal length increases, so the field of view becomes narrower. This can impact adversely on the performance of the conventional central perspective, collinearity equation.
model, since the bundle of rays can approach, in effect, a parallel projection.

Fully automatic camera self-calibration generally employs object point arrays in which some or all of the targeted points are coded. A push-one-button calibration operation is nowadays anticipated and so long as well-known principles such as convergent imaging, the use of orthogonal camera roll angles and, desirably, use of a 3D target distribution are adopted, a successful outcome can be anticipated for medium and wide angle lenses. In applying network orientation with self-calibration to images with very narrow fields of view, however, problems can arise through overparameterization, ill-conditioning and subsequent numerical instability in the normal equations of the bundle adjustment. Recovery of satisfactory camera calibration parameters is often precluded in such ‘weak geometry’ cases, due to linear dependencies that arise between the interior and exterior orientation parameters.

The non-linearity of the collinearity equations is considered an inherent obstacle when it comes to the self-calibration of long focal length lenses, since deterioration of the linear independence of the interior and exterior orientation parameters can be anticipated. This partially accounts for why recent research on this topic has focused more on the development of alternative linear models to accommodate such network geometries. For example, Ono & Hattori (2002) developed an orthogonal projection model to address the issue of long distance measurements in close-range photogrammetry. Even though their model was successful, it had limitations in that calibration of interior orientation elements was ignored and object space control points were required for the calculation of the initial exterior orientation parameters. For the measurement of small objects, Rova et al. (2008) implemented a parallel projection model that requires parallel projection images taken with telecentric lenses, with the bundle adjustment incorporating a simplified interior orientation model. For either of these two cases, a fully automatic self-calibration procedure was not reported.

The development of an automated calibration process for consumer grade digital cameras with long focal length lenses forms the main topic of this research. The aim of this development is two-fold: to improve the robustness and precision of recovery of camera interior and exterior orientation parameters, and to extend the applicability of self-calibration to cameras with fields of view as narrow as a
1.1 Statement of the Problem

Robust and reliable recovery of camera calibration parameters, and especially interior orientation elements, via self-calibration has been a long standing problem in close-range photogrammetric measurements involving very long focal length lenses. Linear dependencies that arise between the interior orientation parameters typically preclude a stable solution of the photogrammetric bundle adjustment. While a stable solution might be obtained when camera self-calibration is omitted, a significant degradation in the accuracy of object point coordinates will be presented. Despite the possibility of perturbations in interior orientation parameters being absorbed by the exterior orientation parameters due to projective coupling, proper calibration of the camera is warranted for high-accuracy measurement applications such as structural deformation monitoring and industrial metrology.

The use of narrow field of view imagery leads to the deterioration and numerical instability of the commonly used collinearity equations, mainly due to the near-parallel projection of the rays of light. It is anticipated that as the field of view decreases significantly, the collinearity equation model reaches its applicable limits, since the bundle of rays approach, in effect, a parallel projection. Even if the numerical instability issues or the ill-conditioning presented in such cases is handled properly, the estimation of the unknown bundle adjustment parameters might be inaccurate, or of poor accuracy. Thus, alternative projective geometry formulations should be investigated because of their potential to handle such weak geometric cases.

1.2 Research Approach

In order to address the problems accompanying the use of long focal length lenses, a research program has been designed and carried out. Initially, an in-depth investigation into different numerical methods for the solution of least-squares estimation problems is performed. Algorithms that can possibly help in
1.2 Research Approach

eliminating the numerical stability issues that arise due to excessive projective
coupling in narrow field of view images are identified.

Given the impact of even the smallest inaccuracies in the ill-conditioned
and consequently unstable self-calibrating bundle adjustment of cameras with
very long focal lengths, an analysis of the typical image creation is presented.
The aim is to determine both possible error sources and the subsequent metric
accuracy loss that is associated with the commonly used JPEG file format.
Improvements in the image creation pipeline can potentially enhance the accuracy
of the photogrammetric measurement process with narrow field of view imagery.

To facilitate further investigation into alternative mathematical and computa-
tional models, the use of a photogrammetric simulator needs to be employed. The
simulator that was specifically developed for this research played an important
role in providing a controlled environment, where new algorithms could be tested
and evaluated.

In seeking to overcome problems encountered in the self-calibration of cameras
with lenses of very long focal length, three prospective approaches have been
adopted, two centred on the traditional collinearity equations and one on an
orthogonal projection model. In the first approach, numerical issues arising
in the collinearity equation are initially evaluated. In this process, various
algorithms that offer increased numerical stability in such ill-conditioned cases
are applied within the photogrammetric bundle adjustment. For the second
approach, a re-examination of the traditional linearization of the collinearity
equations with additional parameters is carried out to reveal shortcomings that
manifest themselves when long focal length lenses are employed.

The third approach centres on development of an alternative, orthogonal
projection model that is better suited to the ‘difficult’ and invariably weaker
geometries encountered in photogrammetric networks employing cameras with
very narrow fields of view. Under the orthogonal projection model, a method for
fully automatic camera calibration via bundle adjustment with self-calibration is
proposed.
1.3 Thesis Organisation

A brief introduction to close-range photogrammetry is provided in Chapter 2. Basic photogrammetric concepts of interior, relative and exterior orientation are outlined. The analytical formulation of the bundle adjustment is then reviewed. In the author’s opinion, it is necessary to have a full understanding of the basic operations of photogrammetric orientation before embarking on any aspect involving measurements with lenses of long focal length.

Most estimation problems require some kind of adjustment of observations. The most common adjustment approach is least squares estimation which is described in Chapter 3. The background to least squares is provided along with coverage of a variety of techniques that can be used to solve the resulting linear systems. The focus is on techniques that can help in minimising the ill-conditioning effect which is produced by the narrow fields of view of long focal length lenses.

Chapter 4 reports on an investigation into the photogrammetric use of RAW imagery. Initially, basic information regarding camera sensors and the procedures involved in colour formation and image storage are described. The errors introduced in these processes are identified and analysed in order to provide an alternative approach to the traditional use of JPEG images.

Chapter 5 provides information on the photogrammetric simulator developed for this project. Simulated data sets were used throughout this research in order to test the developed algorithms. The software features, along with various issues encountered during the development process, are discussed.

Chapter 6 focuses on the self-calibration of long focal length lenses via the central perspective model. Practical limits and numerical issues of the collinearity model applied to imagery with narrow field of view are identified. The less than fully successful attempts to address numerical issues via algorithms that offer enhanced stability, showed that attention needed be turned to the formulation of the functional model. A modification of the partial derivative formulation of the camera interior orientation is shown to be a viable option. Experimental testing performed confirms its applicability. Accuracy aspects of the proposed approach are also detailed.
1.3 Thesis Organisation

The orthogonal projection model that accommodates long distance measurements in close-range photogrammetry is discussed in Chapter 7. The mathematical derivation and photogrammetric procedures involved are presented. Two bundle adjustment cases, with one including an interior orientation model, are detailed. A comparison between the previous approach using the collinearity equations and the newly developed orthogonal projection model is provided.
2

Close-Range Photogrammetry

2.1 Introduction

As the field of close-range photogrammetry advances, more and more applications that require high accuracy measurements are emerging. One such large group of applications requires the use of long focal length lenses to perform high precision measurements over very short or long distances. A central aspect in any metric application is the camera system calibration. However, one of the practical impediments to the adoption of long focal length cameras in close-range photogrammetry, is the difficulty in network exterior orientation (EO) and self-calibration. This chapter reviews modern analytical camera calibration techniques in order to provide a framework upon which to support and understand the analysis carried out in subsequent chapters. While discussing current state-of-the-art photogrammetric procedures, it is important to mention the advances made towards automation. These involve mainly the use of coded and single targets. Accuracy and identification aspects of the targets form the first part of
this research, so a basic understanding of their operation and usage is essential.

The chapter opens with a description of the characteristics of close-range photogrammetry. The influence of the calibration of interior orientation (IO) parameters on high precision close-range photogrammetry is discussed in Section 2.3. Section 2.4 and 2.5 summarise the procedures of photo-triangulation and the reconstruction of shape and position of an object from two or more images. Finally, information on coded targets, which play a significant role in developing fully automatic photogrammetric measurements, is provided in Section 2.6.

2.2 Analytical Restitution

There is an extensive body of literature available on the historical background of photogrammetry and its development. A brief overview is presented by Mikhail et al. (2001) while more details can be found in Luhmann et al. (2006b). An analytical review of the history of close-range photogrammetry along with the developments of theory, algorithms and sensors is provided in Atkinson (1996). As Mikhail et al. (2001) state, the practical development of photogrammetric triangulation was for a long time limited by computational capability. With the computational evolution provided by powerful personal computers, the mathematical models used in triangulation became more rigorous and capable, allowing for the simultaneous solutions of networks of thousand of photos and points.

Nonetheless, despite rapid advances in computer science and the move to digital sensors, the fundamental operations of photogrammetry, namely interior, relative and exterior orientation, as well as resection and intersection, have remained the same. These basic mathematical procedures are essential for the calculation of image orientation parameters and object point coordinates. Comprehensive understanding of the underlying geometrical relationship between the image space and the object space is required in this process. In close-range photogrammetry, the fundamental model describing the above relationship is the central perspective model. A key aspect in improving the photogrammetric restitution process is the ability to account for additional errors introduced in the system by the imaging sensors. By doing so, it is possible to achieve accuracies
that exceed 1 : 100000 in object space with off-the-shelf digital single-lens reflex (DSLR) cameras (Fraser, 1997a; Fraser et al., 2005; Luhmann, 2010). System calibration is a necessary procedure in photogrammetric triangulation for such high accuracies to be achieved.

2.2.1 Collinearity Equations

The mathematical formulation of the relationship between image and object space can be described by the collinearity equations, which derive from central projection. The fundamental characteristic of such a relationship is that the perspective centre, the image point and the corresponding object point all lie on a straight line in space. Perturbations to this relationship, however, give rise to departures from collinearity. Figure 2.1 shows the relationship between the

![Figure 2.1: Relationship between image and object point coordinates.](image)

coordinates $x$, $y$ of an image point $p$ and the coordinates $X$, $Y$, $Z$ of an object point $P$ within a basic interior orientation model. This can be mathematically
2.3 Departures from Collinearity

formulated as

\[
\begin{align*}
x &= x_p - cr_{11} (X - X_0) + r_{12} (Y - Y_0) + r_{13} (Z - Z_0) + \Delta x \\
y &= y_p - cr_{21} (X - X_0) + r_{22} (Y - Y_0) + r_{23} (Z - Z_0) + \Delta y
\end{align*}
\] (2.1)

The parameters \(x_p\) and \(y_p\) are the coordinates of the principal point and \(c\) is the principal distance or focal length. The parameters \(r_{ik}\) appearing in Equation 2.1 are the elements of the rotation matrix \(R\) which describes the three-dimensional orientation of the image with respect to the \(XYZ\) object coordinate system. The inherent departures from collinearity, due mainly to lens distortion, must be considered for metric measurement, with these perturbations to the collinearity equations being described by the terms \(\Delta x\) and \(\Delta y\).

2.3 Departures from Collinearity

Departures from collinearity can be described accurately by mathematical formulations. Brown (1971) introduced the well-known 10-parameter camera calibration model to model perturbations in image coordinates that occur during the physical imaging process. In seeking appropriate parameters for the functions \(\Delta x\) and \(\Delta y\), Brown (1971) considered the three principal sources of departures from collinearity which are physical in nature. These are symmetrical radial distortion, decentering distortion and linear distortion. Thus,

\[
\begin{align*}
\Delta x &= \Delta x_{radial} + \Delta x_{decentring} + \Delta x_{linear} \\
\Delta y &= \Delta y_{radial} + \Delta y_{decentring} + \Delta y_{linear}
\end{align*}
\] (2.2)

2.3.1 Non-physical Perturbations

Within the collinearity equation model, errors in the adopted values of interior orientation elements \(c, x_p\) and \(y_p\) will also give rise to image coordinate perturbations, and it is important to consider the calibration of these parameters as they are required for accurate definition of the interior orientation of the camera system. Additionally, as will be explained later, the modelling of
parameters describing image coordinate perturbations is a function of the interior orientation elements. Consequently, the determination of accurate values for these parameters is of significant importance. Figure 2.2 illustrates the geometric impact of departures from collinearity in the image space. The principal distance \( c \) is defined as the distance of the perspective centre from the image plane in the negative \( z \)-axis direction. When focused at infinity, \( c \) is approximately equal to the focal length of the lens. The origin \( O \) of the coordinate system is located in the centre of the image plane. For analytical calculation, however, the image coordinate system is translated to correspond with the perspective centre \( pp(x_p, y_p) \) which is the point where the optical axis intersects the focal plane.

![Figure 2.2: Interior orientation.](image)

### 2.3.2 Radial Distortion

Symmetrical radial distortion in photogrammetry is represented as an odd-ordered polynomial series, as a consequence of the nature of the Seidel aberrations:

\[
\Delta r' = K_1 r'^3 + K_2 r'^5 + K_3 r'^7 + \ldots
\]  

(2.3)
where the $K_i$ terms are the coefficients of radial distortion and $r'$ is the radial distance from the principal point.

$$r' = \sqrt{x'^2 + y'^2} = \sqrt{(x - x_p)^2 + (y - y_p)^2}$$  (2.4)

Figure 2.2 illustrates the deviations that occur due to radial distortion. Radial distortion can have both negative and positive values. Positive radial distortion is often referred to as pincushion distortion, while negative distortion is known as barrel distortion, due to the resulting geometry deformation in the image plane.

For wide-angle lenses, the third-, fifth- and seventh-order terms are often required. For the majority of medium angle lenses nowadays employed in cameras, the third-order $K$ term is sufficient to account for the induced aberrations, at least for applications that do not demand highest accuracy. Although radial distortion is present in long focal length lenses, its magnitude is generally small, and for metric applications, decreases with increasing focal length (Fraser & Al-Ajlouni, 2006). As the cubic component of radial distortion is the most significant and the combined contribution of the $K_2$ and $K_3$ terms are generally quite small, the distortion can be adequately profiled with just the $K_1$ term. Additionally, the projective coupling between the principal distance and lens distortion can help in eliminating components of the error as well. It is not uncommon for digital cameras to utilise only a central part of the available field of view, which in combination with long focal length lenses, can lead to a radial profile that may not depart from a linear function of the form $\Delta r = K_0 r$. This linear profile can be absorbed by a change in the principal distance, which will in turn indicate that there is no significant radial distortion.

From Equation 2.3, the necessary corrections to the $x, y$ coordinates follow as

$$\Delta x'_{\text{radial}} = x \frac{\Delta r}{r'}$$
$$\Delta y'_{\text{radial}} = y \frac{\Delta r}{r'}$$  (2.5)

The coefficients $K_i$ are usually highly correlated with each other. However, this does not typically affect their calculation.
2.3 Departures from Collinearity

2.3.3 Decentring Distortion

The misalignment of the individual elements of the lens along the optical axis introduces another error known as decentring distortion. This can be compensated by the functions below that were derived by Brown (1966):

\[
\Delta x'_{\text{decentring}} = P_1 (r'^2 + 2x'^2) + 2P_2 x'y' \\
\Delta y'_{\text{decentring}} = P_2 (r'^2 + 2y'^2) + 2P_1 x'y' 
\]

(2.6)

Although it is not obvious from the above formulation, there is a strong coupling of the decentring distortion parameters with the principal point offsets \(x_p\) and \(y_p\) (Gruen & Huang, 2001). This correlation increases with increasing focal length and it can be problematic for calibration of long focal length lenses. Images with higher convergence angles can help in diminishing the coupling effect. The image coordinate correction values for decentring distortion rarely exceed a few pixels maximum, and as a consequence of this and the projective coupling to IO, decentring distortion is often ignored (Fraser, 1997a; Gruen & Huang, 2001).

2.3.4 Linear Distortion

Digital imaging systems that have rectangular light sensitive elements instead of square can present deviations of the image coordinate system with respect to orthogonality and scale uniformity. The metric consequences of such deviations are described by two parameters, the affinity and shear, and can be modelled as

\[
\Delta x'_{\text{linear}} = B_1 x' + B_2 y' \\
\Delta y'_{\text{linear}} = 0 
\]

(2.7)

Linear distortion is usually ignored in close-range photogrammetry because of the high geometric integrity of modern-day digital cameras with nominally square pixels (Gruen & Huang, 2001; Shortis & Beyer, 1996).
2.4 Relative & Exterior Orientation

Prior to the calculation of the three dimensional (3D) digital model of the photographed object, a preparatory phase where the images are oriented is performed. The procedure is known as exterior orientation and it involves six parameters per image, these describing the spatial position and orientation of the camera station coordinate system with respect to the object space coordinate system. Figure 2.3 illustrates the six parameters, namely three translations $X_o, Y_o, Z_o$ and three rotations $\omega, \varphi, \kappa$ that describe this relationship.

In close-range photogrammetry, the calculation of the exterior orientation (EO) parameters usually starts with just a pair of images, a process known as relative orientation (RO). Relative orientation describes the procedure in which the translation and rotation parameters of one image with respect to another are determined in a common coordinate system. Even though there are various ways to calculate the relative orientation, the most common approach is using the coplanarity condition. This process is the critical first step in the determination of a photogrammetric network when control points are not available and thus spatial resection cannot be applied. Alternative options are available when there is knowledge of control points in the object space. Information on such cases can be found in Mikhail et al. (2001), Kraus (2007) and Luhmann et al. (2006b).

The coplanarity approach relies on the fact that an object $P$ and the two perspective centres $O_1$ and $O_2$ must lie in a plane. This plane is the epipolar...
plane defined by the vectors \( \mathbf{b}, \mathbf{r}_1 \) and \( \mathbf{r}_2 \), as shown in Figure 2.4. This can be expressed in mathematical terms as the scalar triple product of the three vectors, which must be equal to zero. Hence,

\[
r_1(\mathbf{b} \times \mathbf{r}_2) = 0
\]

The coplanarity condition can also be expressed in analytical form as the solution for the determinant of a 3 \( \times \) 3 matrix.

\[
\begin{vmatrix}
  b_x & b_y & b_z \\
  x_1 & y_1 & z_1 \\
  x_2 & y_2 & z_2
\end{vmatrix} = 0 \quad \text{(2.8)}
\]

or

\[
\begin{pmatrix}
  x_1 & y_1 & -c
\end{pmatrix}
\begin{pmatrix}
  0 & b_z & -b_y \\
  -b_z & 0 & b_x \\
  b_y & -b_x & 0
\end{pmatrix}
\begin{pmatrix}
  x_2 \\
  y_2 \\
  -c
\end{pmatrix} = 0 \quad \text{(2.9)}
\]

Figure 2.4: Coplanarity.

For a given image pair, a minimum of five point correspondences is required to solve the unknown parameters via the coplanarity model, in what is termed
2.4 Relative & Exterior Orientation

dependent RO (Mikhail et al., 2001). All of the twelve EO elements of the two images appear in the coplanarity equations. RO involves the determination of five degrees of freedom. Hence, seven parameters must be fixed, which is generally achieved by establishing the origin of the object space coordinate system at the first camera station, setting its orientation to zero and assigning an arbitrary value to the baseline component \( b_x \).

The main difficulties in the implementation of the algorithm for dependant RO for image pairs having arbitrary geometry is the determination of reasonable initial values. This can cause problems in cases where long focal length lenses are used, since the narrow field of view can give rise to excessive projective coupling of the interior and exterior orientation parameters. The adopted methodology, developed by Cronk (2007), uses a Monte Carlo strategy for determining the initial RO parameter values, where a large number of possible RO solutions are assessed. A detailed description of this process can be found in Cronk et al. (2006) and Cronk (2007).

Following the determination of the EO of an image pair with a network of multiple images, it is possible to calculate the orientation of the rest of the images. A preparatory step in which the approximate object space coordinates are calculated is first needed. This procedure is called spatial intersection and it uses the collinearity equations to determine the coordinates in object space by intersecting the image rays from two or more images. When only an image pair is available, a total of four equations, containing three unknowns, is obtained with the solution generally being obtained via least squares. If additional points are available, a multi-ray least squares solution can be further employed to increase accuracy and reliability, and to allow for observational outlier detection.

With the acquisition of at least three non-collinear object points, the exterior orientation parameters of every image can be calculated. The process of determining the exterior orientation parameters of an image from known points is known as resection. Again, the resection method is based on the collinearity equations and can be thought of as a special case of a general solution (termed a bundle adjustment) with only one image and known object points. In the case where the minimal three points are known, the system has no degrees of freedom, as six equations are formulated for the six unknown EO parameters. A least
squares approach can be performed to obtain better results, as well as outlier
detection when additional points are available.

2.5 Bundle Adjustment

Although the procedures explained in Section 2.4 are fundamental to photogram-
metric triangulation, they are seldom used in practice except as a first step for
generating approximations for a bundle adjustment, which is a general solution
of the collinearity equations, originally formulated by Duane Brown in 1958 (eg
Brown, 1971). The bundle adjustment refers to the bundle of rays of light leaving
3D object space and converging to each camera perspective centre. The ‘bundle
adjustment’ is essentially the simultaneous relative orientation of all bundles
with respect to one another that leads to the optimal calculation of the EO
parameters and the object point coordinates. Optionally, it can also include
additional calibration parameters such as the camera IO parameters, in order to
account for the departures from collinearity. This process is commonly called a
self-calibrating bundle adjustment, due to the inclusion of the camera calibration
parameters. Figure 2.5 illustrates a typical case of a bundle adjustment where
multiple stations, object points and possibly cameras form the photogrammetric
network.

The bundle adjustment constitutes a general solution of the collinearity
equations. For the least squares adjustment, the collinearity equations have to
be linearised using the Taylor expansion series, as they do not have a linear
form. The calculation of the differential quotients needed for the linearisation
process requires approximate values of some of the unknown parameters, which
can be acquired with the methods presented in the previous sections. After the
linearisation process the bundle adjustment will have the form of

$$ A_1 x_1 + A_2 x_2 + A_3 x_3 = b $$

(2.10)

The unknowns of Equation 2.10, and consequently the unknowns of a bundle
adjustment, are the selected camera calibration parameters for each cam-
era represented by the vector $x_1$, the exterior orientation parameters $x_2^T =$
2.5 Bundle Adjustment

Figure 2.5: Multi-station bundle adjustment.

\[
\begin{bmatrix}
X_o & Y_o & Z_o & \omega & \varphi & \kappa
\end{bmatrix}
\]
for each image station and the object point coordinates \(x_3^T = [X \ Y \ Z]\) for each point. Usually the vector \(x_1\) consists of 8 parameters with \(x_1^T = [c \ x_p \ y_p \ K_1 \ K_2 \ K_3 \ P_1 \ P_2]\). Thus, for a network of \(l\) cameras, \(m\) images and \(n\) points, the total unknowns are \(8l + 6m + 3n\), should a basic 8-parameters camera calibration model be considered. The observations most of the time are solely image coordinates measurements, so for a network of similar size as the one already mentioned, the number of observations is \(2mn\), if all the points are visible in every image. Multi-image photogrammetric networks with many objects points are usually highly redundant in observations. The matrix \(b\) is the image coordinate discrepancy vector between the approximate and the observed values, with

\[
b = \begin{bmatrix}
x^o - x \\
y^o - y
\end{bmatrix}
\]

where \(x^o\) and \(y^o\) are the approximate calculated image coordinates.

The \(A_1\) matrices are the configuration (design) matrices that correspond to each of the unknowns vectors \(x_1\). More specifically, the partials derivatives with
respect to the self-calibration parameters for an image \(i\) and point \(j\) are

\[
A_{1ij} = \begin{bmatrix}
-\frac{\bar{x}}{c^o} & -1 & \bar{x}r^2 & \bar{x}r^4 & \bar{x}r^6 & 3\bar{x}^2 + \bar{y}^2 & 2\bar{x}\bar{y} \\
-\frac{\bar{y}}{c^o} & 0 & -1 & \bar{y}r^2 & \bar{y}r^4 & \bar{y}r^6 & 2\bar{x}\bar{y} & \bar{x}^2 + 3\bar{y}^2
\end{bmatrix}
\]

with \(c^o\) being the preliminary value of principal distance, \(\bar{x} = x - x_p\) and \(\bar{y} = y - y_p\).

The partials derivatives for the exterior orientation parameters are

\[
A_{2ij} = \begin{bmatrix}
\frac{df_x}{dX} & \frac{df_x}{dY} & \frac{df_x}{dZ} & \frac{df_x}{d\omega} & \frac{df_x}{d\phi} & \frac{df_x}{d\kappa} \\
\frac{df_x}{dX} & \frac{df_y}{dY} & \frac{df_y}{dZ} & \frac{df_y}{d\omega} & \frac{df_y}{d\phi} & \frac{df_y}{d\kappa}
\end{bmatrix}
\]

and those for the object point coordinates have the form

\[
A_{3ij} = \begin{bmatrix}
\frac{df_x}{dX} & \frac{df_y}{dY} & \frac{df_y}{dZ}
\end{bmatrix}
\]

The ability to introduce different weights for the observations is possible by providing a weight matrix \(P_{ij}\) for each observation with

\[
P_{ij} = C_{ij}^{-1} = \begin{bmatrix}
\frac{1}{\sigma_x^2} & 0 \\
0 & \frac{1}{\sigma_y^2}
\end{bmatrix}
\]

Then, the final normal equations structure of a self-calibrating bundle adjustment with observation weighting will have the form of

\[
A^T PAx + A^T Pb = 0
\]

where

\[
\begin{bmatrix}
A_1^T P A_1 & A_1^T P A_2 & A_1^T P A_3 \\
A_2^T P A_1 & A_2^T P A_2 & A_2^T P A_3 \\
A_3^T P A_1 & A_3^T P A_2 & A_3^T P A_3
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
+ \begin{bmatrix}
A_1^T P b \\
A_2^T P b \\
A_3^T P b
\end{bmatrix} = 0
\]

The bundle adjustment constitutes an over-determined, large, sparse geometric estimation problem. Chapter 3 provides information on the calculation of the unknowns for such a linear system of equations. However, as the photogrammetric bundle adjustment has a specific sparse matrix pattern, it is possible to take its structure into account in order to minimise the total operations required for
calculation of the solution. Two techniques have mainly been employed. The first
one relies on subdividing the matrix into blocks in order to reduce the size of the
operating matrices. The second method, which is based on the previous approach,
employs sequential estimation techniques to further reduce the calculations by
operating on each observation sequentially. These approaches can even support
real-time measurements, as discussed in Brown (1976b) and Edmundson (1997).

One of the problems that can arise in the self-calibrating bundle adjustment
is that in many cases, due to the projective coupling between the unknown
parameters, the determination of the interior orientation parameters is of
doubtful accuracy. In narrow field of view photogrammetric measurements,
where the the bundle of rays can approach, in effect, a parallel projection
the projective coupling, is aggravated. Therefore, to apply self-calibration, a
careful selection of the parameters is needed. Network design with multiple
camera stations, convergent imagery and the use of orthogonal roll angles (κ-
rotations) is mandatory when endeavouring to reduce the correlation between
system parameters, but this may not always be feasible in applications involving
long focal length lenses.

2.6 Automation

With the introduction of digital cameras and digital image processing, it became
possible to simplify and automate the image measurement process that precedes
all the analytical calculation procedures. The use of coded targets is one such
automation technique (Van Den Heuvel et al., 1993) that enables fast and effective
camera calibration and network orientation. Coded targets have been widely
used in vision metrology applications to automate exterior orientation calculation
(Cronk, 2007; Fraser, 1997b; Ganci & Hanley, 1998; Luhmann et al., 2006b).
They are designed in such a way that they may be automatically detected and
identified by image processing methods. Only a sufficient number of targets to
allow the determination of exterior orientation parameters need to be coded to
support automatic measurement. The remainder may be standard, circular retro-
reflective targets which can be detected and linked to other points via intersection
or back projection. Figure 2.6 presents the flowchart of a photogrammetric
network orientation and object point calculation procedure where the solution is calculated automatically through the use of coded targets.

![Flowchart of a fully automatic exterior orientation process using coded targets](image_url)

**Figure 2.6:** Flowchart of a fully automatic exterior orientation process using coded targets.

Within the research conducted, coded and single silver retro-reflective targets, as shown in Figure 2.7, were used to mark object points in the scene. Especially for applications that involve measurements over long distances where there is less
control of the lighting conditions, the use of retro-reflective targets is considered necessary. The use of a camera flash to illuminate the object scene can help overcome lighting issues, as the retro-reflective targets will provide accurate image measurements even under poor lighting conditions. It is important in the network design to ensure that the relative size ratio between the object and the coded targets will facilitate successful image scanning. Cronk (2007) states that the size of a single circular blob should not be smaller than $4 \times 4$ pixels.

### 2.6.1 Coded and Single Target Identification

There are two tasks to be performed in the process of computing the subpixel locations of targets in the images: identification and location. The identification stage aims to ascertain locations which may contain targets. Various approaches can be found in the literature, for example template matching (Van Den Heuvel et al., 1993) and intensity-thresholding (Clarke et al., 1994). Due to the advantages it offers, the intensity-thresholding approach has been widely used in industrial metrology (Shortis et al., 1994). In a sequential processing of each image row, the pixels are scanned in order, comparing colour brightness values with the previous pixel. If the difference is greater than a certain threshold, the region is classified as a potential target. Once the full image is scanned, the potential targets are merged based on whether they lie on adjacent lines. The areas of interest are then further refined to ensure maximum recovery and correctness of the targets. Finally, a filtering process for each target is performed with the purpose of discarding incorrect targets based on geometric and
2.6 Automation

radiometric tests. The second task is the calculation of the centroid of the target image which precisely and accurately determines the target coordinates. The most common technique, due to its efficiency, is the calculation of the centroid via a binary approach, however alternative approaches that perform ellipse or Gaussian distribution fitting have been used. Shortis et al. (1995) and Shortis et al. (1994) provide an investigation and in-depth comparison of the accuracy and precision of various centroiding techniques. An additional identification step exists for coded targets. In this procedure, targets have to be initially assessed via a clustering algorithm which groups single targets based on properties such as closeness, size and overall brightness to ascertain if they form part of a coded target or not. Once targets have been clustered and prospective codes identified, the angular and spatial relationships of each combination of targets within a potential cluster are examined and decoded (Cronk et al., 2006; Shortis et al., 2003).
3.1 Introduction

This section provides a background on least squares (LS) estimation models employed for bundle adjustment of photogrammetric networks involving long focal length lenses. The least squares adjustment method is commonly used in over-determined linear and non-linear systems of equations to provide an optimal computational solution. Since photogrammetry is an estimation problem, least squares techniques are widely used. The least squares solution is known by different names in various scientific disciplines and it has been an indispensable tool since its invention by Gauss and Legendre around 1800. It has been estimated that the solution of least squares equations applies at some stage, to about 75 percent of all scientific problems. However, when any of these problems reach the stage of numerical computation they contain the same central problem, namely that the required linearisation of a non-linear functional model, the collinearity equations in this case, into a linear system of equations.
3.1 Introduction

The basic linear least squares problem can be stated as the minimisation of the Euclidean length (2nd - norm) of a real $m$-vector $x_0 = Ax - b$ given a real $m \times n$ matrix $A$ of rank $k \leq \min(m, n)$ and a real $m$-vector $b$. Of course this problem is not limited to real numbers, though the complex case arises much less frequently. On many occasions the purpose of least squares computation is not merely to find some set of numbers that ‘solve’ the problem but rather to find a means to obtain additional quantitative information describing the relationship of the solution parameters to the data.

In order to calculate any of these quantities, a functional or mathematical model that expresses their relationship with the observations is needed. The functional model attempts to describe the geometric and physical relationship between the known observations and the unknown derived quantities. In particular, observations or measurements are often made in order to determine other quantities that cannot be measured directly. Such is the case in photogrammetry, where the $XYZ$ coordinates of the object points along with the position and orientation of the image stations have to be recovered from the $x, y$ observations of the image coordinates. However, functional models are not limited to such relationships, as one can add further parameters to model systematic errors which occur according to some known pattern. Typical examples are the metric calibration parameters of a camera, such as the principal distance and principal point coordinates, along with the coefficients of radial, decentering and linear distortion. These can be added to the functional model to describe it more completely and to correct for the systematic errors.

In this chapter, relevant basics of linear algebra will be discussed and methods used to solve linear equation systems will be reviewed. The focus will be on the special case of ill-conditioned systems, where the conventional solution via least squares is prone to instability. Additionally, various methods that have been used by numerical analysts to handle ill-conditioned scenarios are presented. To better understand how the algorithms work, the ideas behind the factorisations used are analysed so as to find the most appropriate approach to use in ill-conditioned cases within photogrammetry.
### 3.2 Gauss-Markov Model

The functional model can compensate for systematic observational errors. However, measurements also contain errors that exhibit a random behaviour and the functional model is incapable of describing their behaviour. The stochastic model performs this role. The usual stochastic model of the photogrammetric bundle adjustment is the Gauss-Markov model, and this is commonly termed as the least squares method of observation equations.

Given an \(m\)-vector \(b\) of observations and an \(n\)-vector \(x\) of unknowns, the linear or linearised functional model that relates the two is \(Ax = b + \epsilon\), where \(\epsilon\) is an \(m\)-vector of the observational errors and \(A\) is the \(m \times n\) design matrix \((m \geq n)\).

The stochastic model that describes the error vector \(\epsilon\) minimises the Euclidean norm \(\|\epsilon\|_2\). Thus the expectancy of the error vector is zero \(E\{\epsilon\} = 0\). Then, the expectancy of the observations is

\[
E\{b\} = E\{Ax - \epsilon\} = E\{Ax\} - E\{\epsilon\} = Ax \Rightarrow b = Ax
\]

\[
D(b) = C = \sigma_0^2Q
\]

with

\[
Q = P^{-1}
\]

where \(D\) is the dispersion operator, \(C\) the positive-definite covariance matrix of observations, \(\sigma_0\) the \(a\ priori\) variance of the unit weight and \(P\) the observational weight matrix. The vector of residuals \(v\) is the difference between the adjusted values of observations and their corresponding observed values:

\[
\hat{v} = \hat{b} - b = A\hat{x} - b
\]

Equation 3.4 forms the so-called observation equations. The least squares solution minimises the Euclidean length of the residual vector \(\|Ax - b\|\). Or, if the weight is included \((Ax - b)^TP(Ax - b)\) is minimised, which is more familiar as \(v^TPv\). For the special case that the weight matrix is diagonal the least squares can be
written as:

\[ f = p_1 v_1^2 + \cdots + p_n v_n^2 = \sum_{i=0}^{n} p_i v_i^2 \]  

(3.5)

When a unit matrix is adopted for the weight matrix, just the sum of the squared residuals is minimised and from this simpler case comes the name ‘least squares’.

If the minimised quantity is named \( f \), then \( f \) is a function of all the errors \( v \) that in turn are a function of the unknown vector \( x \) because \( v = Ax - b \). Due to this, \( f \) is also a function of the unknowns \( f(x) \) and thus the values of the unknown parameters that minimise \( f \) can be defined by zeroing out the derivatives:

\[ \frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial x_2} = 0, \quad \cdots, \quad \frac{\partial f}{\partial x_m} = 0 \]

The necessary calculation of the above leads directly to the normal equations:

\[ N\hat{x} - u = 0 \Rightarrow N\hat{x} = u \]  

(3.6)

with

\[ N = A^T P A \]  

(3.7)

and

\[ u = A^T P b \]  

(3.8)

Equation 3.6 arises directly from the condition that the residual vector \( b - Ax \) for the least squares solution must be orthogonal to the column space of \( A \). This condition is expressed by the equation

\[ A^T (b - A\hat{x}) = 0 \]  

(3.9)

The solution of the normal equations leads to estimates for \( \hat{x} \) and \( \sigma^2_0 \).

\[ \hat{x} = N^{-1} u \]  

(3.10)

\[ \sigma^2_0 = \frac{v^T P v}{m - n} \]  

(3.11)

where \( m \) is the number of observations, and \( n \) is the number of the unknown
3.2 Gauss-Markov Model

parameters. The difference \( m - n \) is also known as the redundancy, or the degrees of freedom. The variance-covariance matrix of the estimated parameters is

\[
\mathbf{C}_\mathbf{\hat{x}} = \sigma_0^2 \mathbf{Q}_\mathbf{\hat{x}}
\]  

(3.12)

where \( \mathbf{Q}_\mathbf{\hat{x}} \) is the co-factor matrix of the parameters with

\[
\mathbf{Q}_\mathbf{\hat{x}} = \mathbf{N}^{-1}
\]  

(3.13)

The development of least squares depends upon a linear mathematical model. Non-linear functions are usually linearised using the Taylor series expansion ignoring the terms of order two and higher. In such a case, the least squares solution provides a vector of corrections to initial approximate parameter values.

### 3.2.1 General Least Squares

When the functional model combines both observations and parameters in a form that cannot be described by the observations equations, as in Equation 3.4, then a more general least squares technique has to be used. In such cases, which occurs when the observations cannot be expressed as an explicit function of the parameters above, the equations of the functional model contain \( m \) observations \( y_i^a \) as well as \( n \) unknown parameters \( x_i^a \). Given a functional model, the minimum number of observations necessary to uniquely determine a solution is \( m_0 \) while the redundancy is \( r = m - m_0 \) with \( n \leq r \). The equivalent system of \( s \) equations will have an initial form of

\[
g_1(y_1^a, y_2^a, \ldots, y_m^a, x_1^a, x_2^a, \ldots, x_n^a) = 0 \\
g_2(y_1^a, y_2^a, \ldots, y_m^a, x_1^a, x_2^a, \ldots, x_n^a) = 0 \\
\vdots \quad \vdots \quad \vdots \\
g_r(y_1^a, y_2^a, \ldots, y_m^a, x_1^a, x_2^a, \ldots, x_n^a) = 0
\]

(3.14)
3.2 Gauss-Markov Model

where \( s = (m - m_o) + n = r + n \). The measured observations \( y^b_i \) will differ from the actual values \( y^a_i \) due to random errors by

\[
y^b_i = y^a_i + v_i
\]  

(3.15)

It is more common for the general least squares case to arise with nonlinear equations. Thus, if the functions of Equation 3.14 are linearised using the approximate values \( x^o \) of the unknown parameters \( x^a \), the observations \( y^a \) and the observations \( y^b \), then the linearised equations are derived

\[
g_i(y_1^a, y_2^a, \ldots, y_m^a, x_1^a, x_2^a, \ldots, x_n^a) = 0
\]

\[
g_i(y_1^b, y_2^b, \ldots, y_m^b, x_1^o, x_2^o, \ldots, x_n^o)
\]

\[
+ \left. \frac{\partial g_i}{\partial y^a_1} \right|_{o,b} (y_1^a - y_1^b) + \cdots + \left. \frac{\partial g_i}{\partial y^a_m} \right|_{o,b} (y_m^a - y_m^b)
\]

\[
+ \left. \frac{\partial g_i}{\partial x^a_1} \right|_{o,b} (x_1^a - x_1^o) + \cdots + \left. \frac{\partial g_i}{\partial x^a_n} \right|_{o,b} (x_n^a - x_n^o)
\]

\[
= w_i - b_{i1}v_1 - \cdots - b_{im}v_m + a_{i1}\delta x_1 + \cdots + a_{in}\delta x_n = 0
\]

for \( i = 1, 2, \ldots, s \)

where \( w_i \) is an \( s \)-size vector of residuals of each equation \( g_i \) with

\[
w_i = g_i(y_1^b, y_2^b, \ldots, y_m^b, x_1^o, x_2^o, \ldots, x_n^o)
\]

\( b_{ij} \) are the elements of an \( s \times m \) coefficient matrix in respect to the measured parameters with

\[
b_{ij} = \left. \frac{\partial g_i}{\partial y^a_j} \right|_{o,b} = \left. \frac{\partial g_i}{\partial y^a_j} \right|_{o,b}(y_1^b, y_2^b, \ldots, y_m^b, x_1^o, x_2^o, \ldots, x_n^o)
\]

\( a_{ij} \) are the elements of an \( s \times n \) coefficient matrix in respect to the unknown parameters with

\[
a_{ij} = \left. \frac{\partial g_i}{\partial x^a_j} \right|_{o,a} = \left. \frac{\partial g_i}{\partial x^a_j} \right|_{o,a}(y_1^b, y_2^b, \ldots, y_m^b, x_1^o, x_2^o, \ldots, x_n^o)
\]
and $\delta x_i$ is the difference of the unknown from the approximate parameters

$$\delta x_i = x_i^a - x_i^o$$

With the above naming conventions, the system takes the form

$$Ax - Bv + w = 0 \quad (3.17)$$

If $P$ is the weight matrix of the measured observations $y^b$, then the calculation of the least squares solution of Equation 3.17 can be found with a simple procedure. The first step is to calculate the matrices $M, N$ and vector $u$ of the corresponding normal equations system:

$$M = BP^{-1}B^T \quad (3.18)$$
$$N = A^TM^{-1}A \quad (3.19)$$
$$u = A^TM^{-1}w \quad (3.20)$$

Then the solution of the general least squares normal equations leads to estimates for $\hat{x}$ and $\sigma_0^2$.

$$\hat{x} = -N^{-1}u \quad (3.21)$$
$$\sigma_0^2 = \frac{v^TPv}{s-n} \quad (3.22)$$

The vector of residuals $v$ can be calculated as

$$v = P^{-1}B^TM^{-1}(w + A\hat{x}) \quad (3.23)$$

while the variance-covariance matrix of the estimated parameters is

$$C_{\hat{x}} = \sigma_0^2Q_{\hat{x}} \quad (3.24)$$

where $Q_{\hat{x}}$ is the co-factor matrix of the parameters with

$$Q_{\hat{x}} = N^{-1} \quad (3.25)$$
3.3 Computing the Least Squares Solution

In order to obtain the solution vector \( \hat{x} \) the inversion of Equation 3.7 or 3.19 is required. The determinant of the normal equations can sometimes be small, hence in these cases, the solution for the unknown coefficients is likely to be unstable. It is natural to consider how well determinant size measures ill-conditioning. If the determinant \( \text{det}(A) \) of a matrix \( A \) is zero, then the matrix is definitely singular. So, is \( \text{det}(A) \approx 0 \) equivalent to near singularity? Unfortunately, there is little correlation between the numerical value of the determinant of \( A \) and the condition of a linear system \( Ax = b \). A very well conditioned matrix can have a very small determinant and an ill-conditioned matrix may have a large determinant. Thus, proper terminology is warranted. A well-conditioned linear system is one with the property that all small perturbations of the measurements will lead to small changes to the estimated parameters. On the contrary, an ill-conditioned system is one with the property that small perturbations of the measurements can lead to large changes in the estimated parameters (Burden & Faires, 2005; Golub & Loan, 1996; Stoer & Bulirsch, 2002; Trefethen & Bau, 1997). It is also necessary to distinguish between two things, the accuracy of the estimated coefficients and the value of the sum of squares of the errors. When the system is numerically unstable, the coefficients may be poorly determined and of doubtful reliability, but still the sum of the squares of the errors can be close to a minimum.

There are various other ways that can be used in order to solve the normal equations without inversion, which is something not advisable nor necessary. One can use direct methods or iterative methods to solve normal equations. A direct method for solving a system of linear equations is a method that, after a certain finite number of steps, gives the exact solution, disregarding rounding errors. For systems similar to Equation 3.1 where the matrix \( A \) is fully populated, direct elimination methods are almost always more efficient (Dahlquist & Bjorck, 2003). However, when \( A \) is sparse, iterative methods offer certain advantages, and for some very large sparse systems, they are indispensable. Iterative methods give a sequence of approximate solutions, even when carried to the number of iterations that is of practical interest. This property might suggest that the solution is not reliable; however, even direct methods are inexact when carried out on a
3.3 Computing the Least Squares Solution

computer. A significant advantage of the iterative methods is that they can give useful results with a plausible number of iterations while providing notable speed improvements.

In the next section, the focus is on direct solution methods. One way to solve a linear system involves the factorisation of the system to an equivalent triangular form where the solution vector can then be computed via forward or backward substitution. There are various ways to create such a decomposition. One of the most important methods, because it is used by other decompositions, is Gaussian elimination where elements are eliminated in a systematic way. Additionally, the LU and Cholesky factorisations are among other commonly employed methods. While triangular systems are quite trivial in their solution, there can be stability problems that arise with their use.

Another useful and practical way to solve normal equation systems relies on orthogonalisation methods. Such techniques have been widely used and among them are the Householder, Gram-Schmidt, Givens transformation, QR factorisation, eigenvalue decomposition and the Singular Value Decomposition (SVD). In these cases, for a system $Ax = b$ where the matrix $A$ is of full rank, one does not need to create the normal equation matrix $N$. This has practical advantages. The use of orthogonal factorisations leads to matrices that are made up from unitary and diagonal matrices, allowing for simple solution of the linear system.

3.3.1 Triangular Systems

3.3.1.1 Gaussian Elimination and LU Factorisation

Gaussian elimination is the simplest, most familiar and most commonly used method to solve a linear system of equations. In the method of Gaussian elimination, to solve a linear system of equations such as Equation 3.6, the given system is transformed in steps by approximate rearrangements and linear
3.3 Computing the Least Squares Solution

combination of equations into a system of the form

$$Rx = c,$$

$$R = \begin{bmatrix}
    r_{11} & \cdots & r_{1n} \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & r_{nn}
\end{bmatrix}$$

which has the same solution as the initial system. The right hand side of Equation 3.6 is transformed in the exact same way as the rows of the matrix $N$. The matrix $R$ is an upper triangular matrix, so that $Rx = c$ can easily be solved by back substitution, as long as $r_{ii} \neq 0$, $i = 1, \ldots, n$:

$$x_i = \frac{u_i - \sum_{k=i+1}^{n} r_{ik}x_k}{r_{ii}} \quad \text{for} \quad i = n, n-1, \ldots, 1$$

In Gaussian elimination it is common to permute the order of the rows or columns of the matrix being operated on, which is known as pivoting. The Gaussian elimination process without pivoting can sometimes be unusable for solving general linear systems, even if they are of full rank. This instability is related to the failure of continuing the elimination process when the pivot element is equal to zero. With pivots, one can always produce a triangular system, if the matrix is not singular. However, even if no failure occurs, pivot selection is advisable in the interests of numerical stability. This pivot strategy is called complete pivoting, where the element selected to be eliminated is always the largest entry of the current submatrix. The computational cost of the Gaussian elimination operation to create the triangular matrix is $2n^3/3$ floating point operations (FLOPs).

In Gauss elimination the right hand side is treated simultaneously; though in many situations, the right hand side is not known from the beginning. For this kind of situation, decomposition by the LU decomposition into a lower and an upper triangular matrix has proven quite useful. This is achieved by introducing zeros below the diagonal, on each column, one at a time. Additionally it reduces the operations needed to $n^2$ to solve the system of Equation 3.6. The LU decomposition is nothing more than a Gaussian elimination where the multipliers
of rows and columns are saved to the matrix $L$, while the matrix $U$ is similar to the triangular matrix obtained by Gaussian elimination. The matrix $L$ is a unit lower triangular matrix. The process of creating matrices of such form is widely known as Gauss transformation.

### 3.3.1.2 Cholesky Decomposition

When solving a linear system by Gaussian elimination, approximately $2n^3/3$ intermediate results need to be stored. However, it is possible to rearrange the calculations so that the elements in $L$ and $U$ are determined directly. There are various techniques to calculate such compact methods, such as Doolittle’s and Crout’s (Dahlquist & Bjorck, 2003). For symmetric positive-definite matrices, these compact schemes become more attractive, since no pivoting is needed. Thus, for any $n \times n$ positive definite matrix, there is a unique $n \times n$ upper triangular matrix $U$ such that

$$U^TU = N \quad (3.26)$$

This very popular method of decomposing $N$ in $U^TU$ is known as Cholesky decomposition, or the square root method. Assuming that the matrix $N$ of Equation 3.7 is positive definite, then the solution $\hat{x}$ of Equation 3.6 can be obtained by solving the two triangular systems

$$U^Ty = d \quad (3.27)$$

and

$$Ux = y \quad (3.28)$$

These systems can now be solved via a forward and back substitution, respectively.

Alternatively, the process of solving Equation 3.6 for $y$ can be accomplished as part of the Cholesky decomposition of an appropriate augmented matrix $\tilde{A} = [A : b]$ and

$$\tilde{N} = \tilde{A}^T P A \equiv \begin{bmatrix} N & d \\ d^T & \omega^2 \end{bmatrix} \quad (3.29)$$

with $\omega^2 = b^T Pb$. The upper triangular Cholesky factorisation of $\tilde{N}$ is then of
3.3 Computing the Least Squares Solution

the form

$$\tilde{N} = \tilde{U}^T\tilde{U}$$

(3.30)

with

$$\tilde{U} = \begin{bmatrix} U & y \\ 0 & p \end{bmatrix}$$

(3.31)

where $U$ and $y$ of Equation 3.31 satisfy Equations 3.26 and 3.27. Furthermore, it is easily verified that $p$ of Equation 3.31 satisfies

$$|p| = ||b - Ax||$$

where $\hat{x}$ is the vector minimising $||b - Ax||$.

For convenience, the details of computing the decomposition will be described using the notation of Equation 3.26. Obviously the algorithm also applies directly to Equation 3.30. From Equation 3.26 the following equations are derived:

$$\sum_{k=1}^{i} u_{ki} u_{kj} = p_{ij} \quad \text{for} \quad i = 1, \cdots, n; j = 1, \cdots, n$$

(3.32)

Solving for $u_{ij}$ in the equation involving $p_{ij}$ leads to the following equations, which constitute the Cholesky factorisation algorithm:

$$v_i = p_{ii} - \sum_{k=1}^{i-1} u_{ki}^2$$

(3.33)

$$u_{ii} = v_i^{1/2}$$

(3.34)

$$u_{ij} = \frac{p_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}}{u_{ii}} \quad j = i + 1, \cdots, n$$

(3.35)

In actual computation there arises the possibility that the value of $v_i$, computed using Equation 3.33 for the Cholesky decomposition of the positive-definite matrix $N$, may be negative due to round-off errors (Lawson & Hanson, 1995). Such errors can arise cumulatively in the computations associated with Equations 3.7, 3.33, 3.34 and 3.35. However this can be avoided with various methods that will not to be discussed here. The required FLOPs for the Cholesky
3.3 Computing the Least Squares Solution

decomposition are approximately \(n^3/3\).

3.3.1.3 Error Analysis for the Triangular Factorisation Methods

In order to acquire an equivalent triangular system, the matrix \(N\) of the normal equations has to be used. Methods that use the symmetric matrix \(N\) require only about a half as many operations as the algorithms that use the matrix \(A\), as in the case of orthogonal decompositions. However, the operations must be performed using precision \(\eta^2\) to adequately encompass the same class of problems as can be treated using algorithms that work directly on \(A\) with precision \(\eta\). Additionally, as will be explained in the Section 3.3.2.5, the condition number of the matrix \(N\) is squared such that the sensitivity of the result \(x\) changes.

As a result of applying floating point arithmetic to the matrix \(N\) of Equation 3.6 a lower, upper or both triangular matrices can be produced to solve the linear system. In the ideal situation in which no round-off errors occur during the entire solution process, except when \(A\) and \(b\) are stored, then \(fl(u) = u + \epsilon\) and \(fl(N) = N + \epsilon\) is non-singular. The solution \(\hat{x}\) will satisfy:

\[
(N + E)\hat{x} = (b + e) \tag{3.36}
\]

It should therefore not come as a surprise that the algorithm can return an inaccurate \(\hat{x}\) if \(A\) is ill conditioned relative to the machine precision, \(u_k\infty(A) \approx 1\).

3.3.2 Orthogonal Decomposition

The difficulties in computing the solution from the least squares normal equations have already been referred to. In this section, an analysis of other methods that work without necessarily forming the normal equations, is undertaken. These are based on orthogonal decomposition, which can be applied to any rectangular matrix. Such a decomposition yields as a result a product of orthogonal and diagonal or triangular matrices. An \(n \times n\) square matrix \(Q\) can be called orthogonal if

\[
Q^TQ = I \tag{3.37}
\]
3.3 Computing the Least Squares Solution

Golub & Plemmons (1980) advocate orthogonal decompositions for their stability and efficiency. Orthogonality has a very prominent role to play in matrix computations and provides a most reliable solution procedure. A significant property of orthogonal matrices for least squares is that the Euclidean norm is preserved under multiplication. Householder reflections, Givens rotations and Gram-Schmidt are central to this process and the next section begins with a discussion of these important transformations.

3.3.2.1 Householder Reflection and Givens Rotations

Householder reflections (Householder, 1958) and Givens rotations (Givens, 1958) are the key to generation of many of the orthogonal decompositions and thus they have an important role in matrix computations. It is helpful to examine the geometry associated with the rotations and reflection at a $2 \times 2$ level. An orthogonal rotation $Q$ has the form

$$
\begin{bmatrix}
\cos(\theta) & \sin(\theta) \\
-\sin(\theta) & \cos(\theta)
\end{bmatrix}
$$

If $y = Q^T x$, then $y$ is obtained by rotating $x$ counterclockwise through an angle $\theta$. A $2 \times 2$ orthogonal matrix $Q$ is a reflection if it has the form

$$
\begin{bmatrix}
\cos(\theta) & \sin(\theta) \\
\sin(\theta) & -\cos(\theta)
\end{bmatrix}
$$

If $y = Q^T x = Qx$, then $y$ is obtained by reflecting the vector $x$ across the line defined by

$$
S = \text{span} \left\{ \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \right\}
$$

Reflections and rotations are computationally attractive because they are easily constructed and can be used to introduce zeros into a vector by properly choosing a rotation angle or the reflection plane.

Householder reflections are similar in two ways to Gauss transformations. They are rank-1 modifications of the identity and they can be used to zero selected
3.3 Computing the Least Squares Solution

components of a vector. Let \( v \in \mathbb{R}^n \) be non zero. An \( n \times n \) matrix \( P \) of the form

\[
P = I - \frac{2}{v^Tv}vv^T
\]

is called a Householder reflection. In particular, suppose \( x \in \mathbb{R}^n \neq 0 \) is given and \( Px \) is needed to be a multiple of \( e_1 = I_n(:,1) \). Note that

\[
Px = \left( I - \frac{2}{v^Tv}vv^T \right) x = x - \frac{2v^Tx}{v^Tv}v
\]

and \( Px \in \text{span}\{e_1\} \) implies that \( v \in \text{span}\{x, e_1\} \). Setting \( v = x + ae_1 \) gives

\[
v^Tx = x^Tx + ax_1
\]

and

\[
v^Tv = x^Tx + 2ax_1 + a^2
\]

and therefore

\[
Px = \left( 1 - \frac{2}{x^Tx+2ax_1+a^2} \right) x - \frac{2v^Tx}{v^Tv}e_1
\]

In order for the coefficient of \( x \) to be zero \( a \) has to be set to \( a = \pm \|x\|_2 \)

\[
v = x \pm \|x\|_2 e_1 \Rightarrow Px = \left( I - \frac{2vv^T}{v^Tv} \right) x = \mp \|x\|_2 e_1
\]

Further information on the practical details of solving least squares with Householder transformation can be found in Businger & Golub (1965) and Golub (1965).

Householder reflections are useful for introducing zeros on a grand scale, but in the case where it is necessary to be more selective, Givens rotations are the
3.3 Computing the Least Squares Solution

transformation of choice. These are rank-2 corrections to the identity of the form

\[ G(i,k,\theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \vdots \\ 0 & \cdots & -c & \cdots & s & \cdots & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}_{i \, k} \]  

(3.44)

where \( c = \cos(\theta) \) and \( s = \sin(\theta) \). Premultiplication by \( G(i,k,\theta)^T \) amounts to a counterclockwise rotation of \( \theta \) radians in the \((i,k)\) coordinate plane. Indeed if \( \mathbf{x} \in \mathbb{R}^n \) and \( \mathbf{y} = G(i,k,\theta)^T \mathbf{x}, \) then

\[
\mathbf{y}_j = \begin{cases} 
 cx_i - sx_k & j = i \\
 sx_i + cx_k & j = k \\
 x_j & j \neq i, k 
\end{cases}
\]  

(3.45)

From these formulae it is clear that it is possible to force \( \mathbf{y}_k \) to be zero by setting

\[
c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \quad s = \frac{-x_k}{\sqrt{x_i^2 + x_k^2}}
\]  

(3.46)

Thus, it is a simple matter to zero a specified entry in a vector by using a Givens rotation. In practice, there are better ways to compute \( c \) and \( s \) than Equation 3.46.

3.3.2.2 Gram-Schmidt

The Gram-Schmidt process of triangular orthogonalisation centres upon making the columns of a matrix orthonormal via a sequence of matrix operations that can be interpreted as multiplication on the right by upper-triangular matrices. It is apparent that Gram-Schmidt orthogonalisation can be regarded as another method for decomposing a matrix \( \mathbf{A} \) into the product of a matrix with orthogonal columns and a triangular matrix \( \mathbf{A} = \mathbf{QR} \). Unfortunately, the Gram-Schmidt
method has very poor numerical properties in that there is typically a severe
loss of orthogonality among the computed $q_i$ elements of the matrix $Q$. More
details on the stability problems of the classical Gram-Schmidt are available
in Rice (1966). Björck (1967, 1994) established the modified Gram-Schmidt
(MGS) method, which has superior numerical stability and requires the same
computational operations. Given an $m \times n$ matrix $A$ with $\text{rank}(A) = n$
the following algorithm computes the factorisation $A = QR$ where $Q$ has
orthonormal columns and $R$ is upper triangular.

$$q_i = a_i \quad \text{for} \quad i = 1, \cdots, n \quad (3.47)$$

$$q_j = a_j - \sum_{i=1}^{n} r_{ij} q_i \quad \text{for} \quad j = 2, \cdots \quad (3.48)$$

where

$$r_{ij} = \frac{a_{j}^{(i)T} q_i}{q_i^T q_i} \quad (3.49)$$

$$a_{j}^{(i)T} = a_j - \sum_{k=1}^{i-1} r_{kj} q_k \quad (3.50)$$

Application of published tests of computer programs (Wampler, 1969) have
indicated that MGS and Householder programs are of essentially equivalent
accuracy. However, the number of arithmetic operations required by MGS is
larger than for Householder, because MGS is always dealing with column vectors
of length $m$, whereas Householder triangularization deals with successively shorter
columns.

3.3.2.3 The QR Factorisation

The QR factorisation of Francis (1961, 1962) and Kublanovskaya (1961) is an
improvement on the older LR method of Rutishauser (1958), while both are
iterative methods for the computation of the eigenvalues of an $m \times n$ matrix
$A$. Such a decomposition yields an orthogonal matrix $Q \in \mathbb{R}^{m \times m}$ and an upper

3.3 Computing the Least Squares Solution

triangular matrix $R \in \mathbb{R}^{m \times n}$

$$A = QR$$

(3.51)

To calculate this decomposition, the matrix has to be initially transformed to tri-diagonal form. There are a variety of methods to obtain such a form, by the use of either Householder, Givens rotations or Gram-Schmidt orthogonal similarity transformations. The most notable algorithms are extensively detailed in Golub & Loan (1996). Additionally Wilkinson & Reinsch (1971) provide a detailed description of how a QR-iteration is performed.

To solve an equation system by least squares, one must solve the equivalent to $Ax = b$, namely

$$QRx = b \Rightarrow Rx = Q^Tb$$

(3.52)

which is straightforward to solve with back substitution, since $R$ is upper triangular.

If the normal equation matrix $N$ is used instead of matrix $A$, then the resulting matrix $R$ is mathematically the same triangular matrix acquired by Cholesky decomposition because of the uniqueness of $R$

$$A^T A = R^T Q^T Q R = R^T R$$

(3.53)

However, this has no advantage if the only purpose is computation of the least squares solution, while additionally it may not provide correct results in ill-conditioned systems. To address stability problems in the classical implementation, a normalisation process is required (Dahlquist & Bjorek, 2003; Golub & Loan, 1996). The normalisation involved in the process of calculating the QR decomposition requires $n$ square roots, resulting in rather slow computation time compared to Cholesky. This is the reason that the photogrammetric bundle adjustment has mostly been implemented using Cholesky decomposition. Golub & Loan (1996) offer a detailed comparison of normal equation decomposition versus the QR while Golub & Loan (1996) and Trefethen & Bau (1997) discuss the accuracy and stability of the alternative QR decomposition approaches. Various high performance issues pertaining to the QR factorisation are addressed by Mattingly et al. (1989) and Knight (1995).
3.3 Computing the Least Squares Solution

The advantages of approaching least squares solutions with the QR decomposition are well known. It is considered by many mathematicians as the most stable algorithm with respect to the propagation of error. Trefethen & Bau (1997) characterise it as one of the most important factorisations. It is significant though, to note that the computed matrix $Q$ will not be orthogonal to working accuracy when $A$ is very ill-conditioned. A modified QR that can manage these ill-conditioned cases has been introduced by Businger & Golub (1965) but there are problems associated with its implementation. Golub & Pereyra (1973) have shown how a QR with column pivoting can be used to determine a minimiser $x$ for a linear system $Ax = b$ where the $m \times n$ matrix $A$ has a rank of $r = rank(A) < min(m,n)$.

3.3.2.4 Eigenvalue and Schur Decompositions

An eigenvalue decomposition of a non rank-defective square matrix $N$ is a factorisation of the form

$$N = X \Lambda X^{-1}$$

(3.54)

where $\Lambda$ is an $n \times n$ diagonal matrix of the eigenvalues and $X$ is an $n \times n$ matrix of the eigenvectors. To apply such a decomposition, it is necessary to calculate the normal equations in order to acquire the square matrix $N$. In addition, for symmetric matrices such as the calculated matrix $N$ of the normal equations, the eigenvectors can be chosen such that they are orthogonal to each other using Schur’s decomposition (Golub & Loan, 1996; Trefethen & Bau, 1997). If the columns of a matrix $Q \in \mathbb{R}^{m \times m}$ contain linearly independent eigenvectors of $N \in \mathbb{R}^{m \times m}$, the Schur decomposition of $N$ is

$$N = QTQ^T$$

(3.55)

where $T$ is an upper triangular matrix. The eigenvalue decomposition expresses a change of basis to ‘eigenvector coordinates’. For a normal equation system $Nx = u$, and by utilising Equation 3.55, the change of basis can be visualised better in the form

$$T(Q^T x) = (Q^T b)$$
3.3 Computing the Least Squares Solution

Thus to compute \( Nx \), \( x \) can be expanded in the basis of columns of \( Q \) and then apply \( T \). The result can be interpreted as a vector of coefficients of a linear combination of the columns of \( Q \). Eigenvalues and eigenvectors are useful because an eigenvalue analysis can simplify solutions of certain problems by reducing a coupled system to a collection of scalar problems. The solution of a least squares problem formed similarly to Equation 3.6 can be given by initially solving the system

\[
Ty = Q^T u
\]

via backward substitution, and then calculating the solution \( x \), with

\[
x = Qy
\]

There are various methods to calculate an eigenvalue decomposition. Since in the calculation process unitary transformations are involved, the resulting algorithms tend to be numerically stable. However, some algorithms have advantages over others, as detailed in Golub & Loan (1996) and Trefethen & Bau (1997). Schur decomposition is among them and it is considered one of the most stable algorithms since it can operate even on rank defective matrices.

3.3.2.5 The Singular Value Decomposition

The singular value decomposition is important for conceptual reasons. Many linear problems can be better understood by using the SVD. This decomposition is closely related to the eigenvalue-eigenvector decomposition of a symmetric non-negative definite matrix \( A^T A \) due to the theme of diagonalising a matrix by expressing it in terms of a new basis. Let \( A \) be an \( m \times n \) matrix of rank \( k \). Then there is an \( m \times n \) orthogonal matrix \( U \), an \( n \times n \) orthogonal matrix \( V \) and an \( n \times n \) diagonal matrix \( S \) such that

\[
A = USV^T
\]

Here the diagonal entries of \( S \) are arranged in a decreasing way with exactly \( k \) being greater than 0. These diagonal entries of \( S \) are called the singular values of \( A \).
3.3 Computing the Least Squares Solution

There are fundamental differences between the SVD and the eigenvalue decomposition. One is that the SVD uses two different bases whereas the eigenvalue has just one (the eigenvectors). The second is that the SVD uses orthonormal bases, whereas the eigenvalue decomposition uses a basis that generally is not orthogonal. A third is that not all matrices have an eigenvalue decomposition, but all matrices have a singular value decomposition. The relationship between the eigenvalues is that by calculating $A^T A$ using the SVD decomposed matrix $A$, the following expression is obtained:

$$A^T A = (USV^T)^T (USV^T) = VDV^T$$

(3.59)

where $D = S^T S = S^2$. It is seen that the singular values of $A$ are actually the positive square roots of the respective diagonal entries of $D$. However, Equation 3.59 is the eigenvalue decomposition of $A^T A$ and thus the elements of $D$ are its eigenvalues.

To compute the SVD one must first bi-diagonalise the matrix. This can be achieved by any of the Householder, Givens or Jacobi transformations. The singular values of a matrix can then be computed rapidly, and in a numerically stable manner, by a method developed by Golub & Reinsch (1970) which is closely related to the QR method. An alternative algorithm, named R-SVD, detailed in Chan (1982) can calculate the SVD faster. In comparing the two processes, it has been concluded that the second method known as R-SVD is more efficient unless $m \approx n$. The FLOPs required are $14mn^2 + 8n^3$ and $6mn^2 + 20n^3$, respectively.

The SVD is a particularly revealing, complete orthogonal decomposition. It provides a neat expression for solving least squares and the norm of the residual $p = ||Ax - b||$. Additionally, linear column dependencies can be identified. The columns of the matrix $V$ associated with small singular values may be interpreted as indicating near linear dependencies among the columns of $A$.

To solve the linear system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$ one has to replace it with the equivalent problem

$$USV^T x = b \Rightarrow SV^T x = U^T b$$

(3.60)
3.3 Computing the Least Squares Solution

Computation of

\[ g = U^T b \]  \hspace{1cm} (3.61)

leads to

\[ Sp \cong g \]  \hspace{1cm} (3.62)

where \( p \) is related to \( x \) by the orthogonal linear transformations

\[ x = Vp \]  \hspace{1cm} (3.63)

Since \( S \) is diagonal, the effect of each component of \( p \) upon the residual norm is immediately obvious. Introducing a component \( p_j \) with the value

\[ p_j = \frac{g_j}{\sigma_j} \]  \hspace{1cm} (3.64)

yields the candidate solution vector \( x \) from

\[ x = Vp = \sum_{j=1}^{r} p_j v^j \]  \hspace{1cm} (3.65)

where \( r = rank(A) \) and \( v^j \) denotes the jth column vector of \( V \). If the intermediate calculations are ignored, the solution vector can be obtained directly from

\[ x = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i \]  \hspace{1cm} (3.66)

The residual norm \( p \) associated with the candidate solution is defined by

\[ p^2 = \|Ax - b\|_2^2 = \sum_{i=r+1}^{m} (u_i^T b)^2 \]  \hspace{1cm} (3.67)

One of the most valuable aspects of the SVD is that it can sensibly deal with the concept of matrix rank. Rounding errors and close to ill-conditioned matrices make rank determination a non-trivial exercise. Indeed, for some small \( \epsilon \), it is important to be able to calculate the \( \epsilon \)-rank of the matrix which can be defined
3.3 Computing the Least Squares Solution

by

$$\text{rank}(\mathbf{A}, \epsilon) = \min_{\|\mathbf{A} - \mathbf{B}\|_2 \leq \epsilon} \text{rank}(\mathbf{B})$$

Thus, if \( \mathbf{A} \) is obtained from measurements with each \( a_{ij} \) correct to within, say, \( \pm 0.0001 \), then it might make sense to look at \( \text{rank}(\mathbf{A}, 0.0001) \). It should be noted that the singular values of a matrix \( \mathbf{A} \) are uniquely determined, even though the orthogonal matrices \( \mathbf{U} \) and \( \mathbf{V} \) are not. The singular values of a matrix are very stable with respect to changes in elements of the matrix. Perturbations of the elements of a matrix produce perturbations of the same, or smaller, magnitude in the singular values. In cases where the matrix \( \mathbf{A} \) with rank \( k < r = \text{rank}(\mathbf{A}) \) is ill-conditioned, some of the later singular values can be expected to be significantly smaller than the earlier. In such a case, some elements of the later values of Equation 3.64 may be undesirably large. Typically, one hopes to find an index \( k \) for Equation 3.66 or 3.65 such that all coefficients \( p_j \) are acceptably small for \( j \leq r \), all singular values \( s_j \) for \( j \leq r \) are acceptably large and the residual norm \( p \) is acceptably small. This technique can be used in order to successfully solve ill-conditioned matrices via an approximation with a nearby well-conditioned system. By ignoring the small singular values, or treating them as zeros, a more satisfactory solution can be achieved. Additionally, one can treat in a similar manner the components of the vector \( \mathbf{g} \) of Equation 3.61 if they are smaller than the assumed uncertainty in \( \mathbf{b} \).

Another important characteristic of the singular value decomposition is the calculation of the condition number of a matrix. The definition of the condition number \( \kappa \) is \( \kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \). However, one has to calculate the inverse of \( \mathbf{A} \), which is generally not available or sometimes not possible to calculate, so the condition number \( \kappa \) of an \( m \times n \) matrix \( \mathbf{A} \) can be found easily from the singular values of matrix \( \mathbf{A} \) by

$$\kappa = \frac{S_1}{S_n} \quad (3.68)$$

If \( \kappa(\mathbf{A}) \) is large, then small relative perturbations in \( \mathbf{A} \) and \( \mathbf{b} \) will produce large relative perturbations in \( \mathbf{x} \), and the problem of solving \( \mathbf{Ax} = \mathbf{b} \) is ill-conditioned.
3.3 Computing the Least Squares Solution

3.3.3 Summary of Least Squares Computation Methods

As has been discussed, every method has advantages and disadvantages. From a mathematical point of view, it is preferable to avoid using the normal equations due to the reasons mentioned in Section 3.3.1.3. However, when it comes to computer programming, the aim is to employ efficient algorithms that will run in almost real time. The use of sequential algorithms has been mentioned by Mikhail & Ackermann (1976) and some algorithms are more easily adapted to real-time matrix decomposition. A summary of the calculation cost of the presented algorithms is given in Table 3.1.

<table>
<thead>
<tr>
<th>LS Algorithm</th>
<th>FLOP Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Equations</td>
<td>$mn^2 + n^3/3$</td>
</tr>
<tr>
<td>LU Decomposition</td>
<td>$2n^3/3$</td>
</tr>
<tr>
<td>Cholesky Decomposition</td>
<td>$n^3/3$</td>
</tr>
<tr>
<td>Householder Orthogonalisation</td>
<td>$2mn^2 - 2n^3/3$</td>
</tr>
<tr>
<td>Modified Gram-Schmidt</td>
<td>$2mn^2$</td>
</tr>
<tr>
<td>Givens Orthogonalisation</td>
<td>$3mn^2 - n^3$</td>
</tr>
<tr>
<td>Householder Bi-diagonalization</td>
<td>$4mn^2 - 4n^3/2$</td>
</tr>
<tr>
<td>R-Bi-diagonalization</td>
<td>$2mn^2 + 2n^3$</td>
</tr>
<tr>
<td>Golub-Reinsch SVD</td>
<td>$4mn^2 + 8n^3$</td>
</tr>
<tr>
<td>R-SVD</td>
<td>$2mn^2 + 11n^3$</td>
</tr>
</tbody>
</table>

A detailed evaluation and analysis of the numerical stability of most of the algorithms presented here can be found in Trefethen & Bau (1997). Through the analysis carried out, it is possible to understand which algorithms can be used in photogrammetry. The Cholesky factorisation is the most suitable when using normal equations, due to its speed and stability. The QR algorithm is the most appropriate choice among the orthogonal decompositions when the design matrix $A$ is used without forming the normal equations. The SVD is the optimal choice for ill-conditioned systems, or when a more in-depth analysis is needed. At the very minimum, this discussion should indicate how difficult it can be to choose the ‘right’ algorithm. Thus, the choice of the appropriate method should take
3.3 Computing the Least Squares Solution

into consideration variables such as stability, number of FLOPs, perturbations and possibly the ability to handle ill-conditioning.
4.1 Introduction

The basic perspective projection model of photogrammetry describes a transformation between two Cartesian coordinate systems, the image space system \((x, y, z)\) and that for the object space \((X, Y, Z)\). Implicit in the referencing of image points is that the \(x, y\) coordinates accurately represent the internal geometry of the focal plane at the time the image is recorded. That means that the integrity of image point positions has to be retained throughout the image preprocessing stage. However, the process of image formation and formatting can have an impact upon image point position as expressed by the \(x, y\) coordinates, and it is imperative for subsequent photogrammetric measurement that image perturbations due to sensor non-linearities and to digital image file creation are fully described, modelled and mitigated to the maximum extent possible.
4.2 Camera Sensors

Given that the integrity of image geometry is such an important factor in the optimisation of measurement accuracy in close-range photogrammetry, this thesis research has included an investigation into digital image formation and formatting, so as to provide a more comprehensive insight into the metric characteristics of digital imagery. Findings from this research contribute to later research into camera self-calibration in close-range photogrammetry.

When a digital image is recorded, the camera needs to perform a significant amount of processing to provide a viewable image to the user. This processing includes corrections for sensor non-linearities and non-uniformities, auto-focus, white balance adjustment, colour interpolation, colour correction, compression and other optional optimisations. These procedures are certainly significant for photogrammetry as they affect the metrics of the image. Since most cameras and digital image file formats were not created for photogrammetric purposes, these steps cannot be omitted. However, for digital SLR cameras that can save the unprocessed pixel values, it is possible to enforce a more ‘photogrammetrically preferable’ workflow.

Within this chapter, a basic explanation of how typical frame imaging sensors work is presented in Section 4.2, while an in-depth analysis of the typical image creation process is given in Section 4.3. Section 4.4 provides insight into the use of unprocessed image files as well as into the associated development work. Finally, an evaluation of proposed image processing methods is presented in section 4.6.

4.2 Camera Sensors

A colour image requires at least three colour samples at each location. Computer images often use red (R), green (G), and blue (B) for representation of true-colour. For a camera to be able to record these colours it would need three separate sensors. One approach is to use beam-splitters (Parulski & Spaulding, 2003) along the optical path to project the image onto three separate sensors as illustrated in Figure 4.1a. Using a colour filter in front of each sensor, three full-channel colour images are obtained. This is a costly approach as it requires three sensors and moreover the relative positions and orientation of these sensors would have to be accurately known or determined for photogrammetric applications.
4.2 Camera Sensors

(a) Beam-splitter based.  (b) CFA based.

Figure 4.1: RGB imaging sensors.

To minimize cost and size, most commercial digital cameras acquire imagery using a single electronic sensor overlaid with a colour filter array (CFA), as shown in Figure 4.1b, such that each sensor pixel only samples one of the three primary colours. To restore a full-colour image from CFA samples, the two missing colour values at each pixel need to be estimated from the neighbouring samples. This process is commonly known as CFA demosaicing or interpolation and it is one of the most important components of the colour image generation pipeline.

The most commonly used CFA pattern is the Bayer pattern (Bayer, 1976), a schematic diagram of which is shown in Figure 4.2. The Bayer array is composed of green pixels on a quincunx grid, and the red and blue pixels on rectangular grids. The result is an image where 50% of the pixels are green, 25% red and

Figure 4.2: Schematic representation of a Bayer RGB sensor.
4.2 Camera Sensors

25% blue. The green channel is measured at a higher sampling rate than the other two because the peak sensitivity of the human visual system (HVS) lies in the medium wavelengths, corresponding to the green portion of the spectrum. Alternative patterns have also been used, for example a cyan, magenta, yellow and green pattern (CMYG).

There are various types of sensors but they can all be classified into two basic categories, as explained above, depending on the way that they produce the true-colour image:

i. Those that record only one colour at each pixel location, or

ii. those that record complete RGB colour information at every pixel

Regardless, all sensors are based on the same physics concept, namely the photoelectric effect. Most sensors fall under the first category, while two different technologies, CCD and CMOS sensors are used for capturing images digitally. A brief history of CCD camera evolution is provided by Cronk (2007). Nowadays CMOS sensors, are gaining popularity; some years ago CCD sensors were used in virtually all high end cameras, and CMOS sensors in the rest. However, today almost all released cameras use CMOS class sensors.

Despite the increased popularity of the sensors that record only one colour at each pixel location, during the last decade, sensors that are able to record complete RGB colour information in every pixel have become more accessible. The Foveon X3 is the only sensor that supports such an approach to capture full colour information. Before the development of the Foveon X3, Foveon cameras were initially known as high-end digital portrait camera systems built around a colour-separation beam-splitter prism assembly similar to that shown in Figure 4.1a.

The operation of the Foveon X3 sensor is quite different from the Bayer filter image sensor. Its development has been inspired by the way that the layers of chemical emulsion produce the colour in film, which is illustrated in Figure 4.3. The Foveon X3 is a direct image sensor that uses three layers of pixels embedded in silicon, as shown in Figure 4.4. The layering of pixels takes advantage of the fact that red, green and blue light penetrates silicon to different depths. Foveon X3 sensor technology benefits from using all the light and sensing all colours at
4.2 Camera Sensors

![Image](image1.png)

(a) Film.  
(b) Foveon.

Figure 4.3: Similarity of film and Foveon X3 sensor.

every pixel location. The order in which each colour is recorded is related to its wavelength. Starting with the lowest wavelength, the top layer records blue, the middle layer records green and the bottom layer records red. As a result, each stack of pixels directly records all of the light at each point in the image.

For CFA-based cameras, an anti-aliasing (AA) filter is also placed in front of the sensor as shown in Figure 4.5. It basically removes the frequencies in the image that are higher than the limits for the the Bayer filter cell size, in other words, it blurs the details that are thinner than the Bayer filter itself, with the purpose of reducing the Moire-like image artefacts.

![Image](image2.png)

Figure 4.4: Schematic representation of a Foveon sensor.

Generally, for every type of sensor, an infrared filter, usually the IR Cut Filter (ICF), is also included to reduce the sensor sensitivity to infrared light. The ICF of a Nikon D70 is shown in Figure 4.6
4.2 Camera Sensors

4.2.1 Pixel Colour Formation Process

The imaging process is usually modelled as a linear process between the light radiance arriving at the camera and the pixel intensities produced by the sensor. In a CCD/CMOS camera, there is a rectangular grid of electron-collection sites laid over a silicon wafer to record the amount of light energy reaching each of the pixels, as shown in Figures 4.7a and 4.7b. A Foveon X3 direct image sensor consists of three layers of pixels, as illustrated in Figure 4.7c, while the information generated for each pixel is shown in Figure 4.7d.

Independent of the sensor type, when photons strike these sensor sites, electron-hole pairs are generated, and the electrons generated at each site are collected over a certain period of time. In the zones where there is no incoming light, the read value is not zero, but a so-called ‘black level’. Once a certain light level is reached, the sensor becomes saturated, which means that it is not capable of producing a higher electrical signal. The level at which this happens is called the saturation level. Saturated cells can flow over to adjacent cells, producing blooming. CCD sensors are more prone to blooming than CMOS sensors. Each sensor has a characteristic dynamic range that can be defined as the relationship between the highest values that the sensor can register without saturation and the deepest shadows where the camera registers some signal distinguishable from...
4.2 Camera Sensors

(a) Close-up on the Bayer matrix. Here, 12 photosites, coloured for the needs of the illustration.

(b) Bayer RGB pixel information.

(c) Foveon X3 sensor.

(d) Foveon X3 pixel information.

Figure 4.7: Comparison of Bayer RGB with Foveon X3 in: (a,c) sensor level, (b,d) pixel level.

the background noise. The electron count recorded in each cell of the sensor is eventually converted to a pixel value.

The resultant image, consisting of either individual, or all of the red, green and blue colour pixels, is a digital camera’s equivalent of a negative in film photography. This image, commonly referred to as a RAW image, contains all the unprocessed information \(^1\) captured during the exposure, without any of the camera ‘software’ enhancements applied. In the above format, the pixels are

\(^1\)Not always true, as will be explained in Section 4.3.1
4.2 Camera Sensors

normally represented by 10-bits (0 - 1023 brightness values), 12-bits (0 - 4095 brightness values) or 14-bits (0 - 16383 brightness values) per colour channel, depending upon the imaging sensor. Also, some newer high-end digital SLR cameras offer a bit depth of 16-bits per pixel of colour information (0 - 65535 brightness values).

4.2.2 True Colour Formation

For CFA based sensors, the demosaicing step is required to produce full RGB colours per pixel. If the measured RAW image is divided by measured colour into three separate images, the demosaicing process resembles a typical image interpolation problem. Straightforward approaches such as nearest neighbour or bilinear interpolation are feasible, but the results are generally not pleasing, since artefacts created by the interpolation can often degrade image quality. Due to this, more specialised algorithms, e.g. interpolation using a threshold-based variable number of gradients (VNG) (Chang et al., 1999), Patterned Pixel Grouping (PPG) and others, that will be mentioned further on, were created. The methods adopted for demosaicing algorithms that reconstruct the full colour image can be divided into three generic groups:

i. Heuristic approaches

ii. Reconstruction approaches

iii. Image formation modelling

Additionally, based on whether inter-channel correlation is employed or not, the demosaicing algorithms are separated into two further categories, those using single channel interpolation, and those exploiting inter-channel correlation.

There is a plethora of algorithms that have been developed over the years to address interpolation in CFA based sensors. A list containing the existing demosaicing algorithms, from the creation of the Bayer RGB filter in 1976 through to the present day, can be found at http://www.danielemenon.netsons.org/top/demosaicking-list.php (Last accessed: 22 March 2011). Further information and evaluation is offered by Gunturk et al. (2005) and Li et al. (2008),
with the latter containing some of the current state-of-the-art algorithms of this field. Shortis et al. (2005) reported on an investigation that aimed in developing a demosaicing algorithm with the purpose of improving the quality of photogrammetric measurements. Such efforts however, have not been compared to other demosaicing algorithms but only aimed at the modification of a specific algorithm in order to improve the quality of stereo measurements.

It is noteworthy that commercial software does not yet appear to support any of the current state-of-the-art algorithms, and the interpolation choices are thus limited. Most of the camera manufacturers provide proprietary software in order to facilitate the transformation from RAW images to other formats such as JPEG or TIFF. Interestingly, but not surprisingly, there is no software that performs just the interpolation task, as the purpose of every software is to provide optimal images for viewing by the user. For that reason, most software performs various additional corrections as will be explained later on. The drawback of using any currently available commercial software is that there is no actual knowledge of either the internal algorithms or what effect they have on the geometric properties of the resulting image.

However, an open source utility named dcraw (Coffin, 2010), which inherently allows for auditability and also provides a variety of demosaicing algorithms, bridges that gap. The dcraw software is currently being used as the core application in various software systems that support RAW image conversion.

4.3 Image Creation Process

In this section, the steps required to form an image from the RAW data are outlined. The differences between performing this transformation in the camera and as a post processing operation are highlighted. It should be noted that the image creation procedure detailed in this section is focused on Bayer RGB-based cameras since they are mainly used nowadays. Nonetheless, the procedure is quite similar when different sensors types are employed.
4.3 Image Creation Process

4.3.1 Black Point Subtraction

As mentioned in Section 4.2.1, in the total absence of light, the sensor returns greater than zero intensity values. If these values are not subtracted from the RAW data the resulting images will not look vivid. Unfortunately, some cameras, for example those from Nikon, perform this subtraction step prior to saving the RAW image. Generally, by subtracting the black point from the RAW data it makes it rather difficult to reduce any noise later on. More sophisticated methods can be applied when this step is performed as a post process such as channel-wise subtraction of the minimum recorded values.

4.3.2 Bad Pixel Removal

Almost every digital camera has a few dead pixels present in the sensor. A bad pixel may be ‘dead’ (dark) or ‘stuck on’ (hot). When a bad pixel is present in the CFA, the camera will generate an RGB pixel that is both ‘the wrong brightness’ and the ‘wrong’ colour. For example, for a 2×2 Bayer interpolation, a ‘hot Red’ (or ‘hot Blue’) sub-pixel will add 100% R (or B) to the output, whilst a ‘hot G’ pixel will add 50% G to the output RGB pixel. A ‘dark R’ (or ‘dark B’) will remove all Red (or all Blue) from the RGB, whilst a ‘dark G’ will remove half the Green. The effect of individual bad pixels may or may not be visible, but Hot pixels will really stand out against a black background, as commonly seen in photogrammetric images of retro-reflective targets. The centroiding of retro-reflective targets can be adversely affected from bad pixels.

Bad pixels can be removed in cases where the processing is not performed in the camera but instead at a later stage with the help of software such as dcrw. This can be achieved by specifying the (x, y) location of the pixel, so that its value can either be interpolated from neighbouring pixels or completely ignored in the subsequent steps. In the unfortunate case of ‘bad’ neighbouring pixels the user has to select which ones to eliminate.
4.3.3 Dark Frame, Bias Subtraction & Flat-Field Correction

This step is an optional part of the image creation procedure since most of the corrections are usually performed as a post-process and not in the camera. The corrections to be explained have been used on various occasions in scientific imaging in order to improve the images, and they are supported in many readily accessible software systems. They can help with dead or hot pixels and can compensate for any uneven illumination caused by the sensor. These techniques take advantage of the fact that a component of the image noise, known as fixed-pattern noise, is the same from one image to another and is mostly related to noise produced from the sensor, dead or hot pixels.

Dark frame subtraction in digital imaging is a way to minimise image noise, especially for images recorded with long exposure times. In order to generate a dark frame, an image should be recorded with no light, thus essentially producing an image of sensor noise. Most cameras can nowadays take a dark frame after an exposure, with matching shutter speed, and subtract it from the RAW data before processing. It is an optional step and is only performed when exposure is longer than about one second. This is a crucial operation to calibrate any imaging sensor but it is sometimes impractical to spend twice the amount of time in shooting images, so post-processing of the image is preferable. In the case that the dark frame is generated as a post-process, the corrections are applied in a similar way by subtracting them from the original image so that they correct any of the pattern noise created. It should be noted that dark frame subtraction is a technique that has been used in digital photogrammetry to improve image quality (Seiz & Baltsavias, 2000; Shortis & Beyer, 1996).

Sensor noise increases with exposure time and temperature; however amplifier noise is independent of exposure time and only marginally dependant on temperature. Thus, it is possible to isolate sensor noise from amplifier noise and obtain image noise due to the electronics. This noise is commonly referred as bias-level and is attributable to the DC voltage maintained in the camera electronics to bias the semiconductor and keep the signal detected by the A-to-D converter, which converts sensor output to discrete bits, from going negative. To
map the magnitude of this error, a dark frame can be taken at an extremely fast shutter speed, e.g. the shortest exposure time allowed by the camera, so that it ideally contains only the amplifier noise. Consequently, correcting for the amplifier noise can be achieved by subtracting the acquired bias frame from the image in a similar manner to the dark-frame subtraction.

The goal of flat-field correction is to remove artefacts caused by variations in the pixel-to-pixel sensitivity of the CCD sensor. Pixels differ by a certain amount in how sensitive they are to light, so one pixel will look darker than its neighbour and another pixel will look lighter even though the same amount of light hits all three. Because a flat field is supposed to be evenly illuminated across the image format, it allows for the acquisition of the sensitivity errors which can later on be used to correct an image. Flats are applied by numerical division after being normalised so that the brightness is relative to the image being applied.

4.3.4 Green Channel Equilibrium Correction

There is not a lot of information regarding this specific step, however various programs, such as Adobe Photoshop DNG and recently dcraw, implement Green channel equilibrium correction. In the Adobe DNG Software Development Kit, there is a clear indication of the purpose of this correction. Its aim is to match the differences of the values of the green pixels in the blue/green rows with the green pixels in the red/green rows. Evidently, it is only applicable to CFA images that use a Bayer pattern filter array. In various sensors, the values of the green cells of red/green rows diverge from each other due to differences in sensitivity.

Llorens (2010) has proposed a simple experiment to check if the green cells in the red/green row have a different sensitivity compared to the green cells in the blue/green row. Initially an image is created with gradients from black to pure red, green, blue as well as magenta, yellow and cyan as shown in Figure 4.8. Then, the direct RAW data, without subtracting the black point nor the application of white balance, is taken for the four channels in the central zone of each gradient. The Green channel is considered as two different colours, with G1 being the green that is located at the rows that contain the red pixels and G2 being that located in the rows that contain the blue pixels. By plotting the
4.3 Image Creation Process

![Image of test pattern for Green channel equilibration.](image)

**Figure 4.8:** Test pattern for Green channel equilibration.

measured values for each channel for the red, green and blue gradients, the charts shown in Figure 4.9 are created.

![Plot of RAW data along three gradients.](image)

(a) Black - Red gradient.  
(b) Black - Green gradient.  
(c) Black - Blue gradient.

**Figure 4.9:** Plots of RAW data along three gradients.

It is clear from the plot of the red gradient, shown in Figure 4.9a, that the G1 channel is raised more than the G2 channel, which is somehow ‘dragged’ by the red channel. However, the graph of the green and the blue gradients, as can be seen in Figure 4.9b and 4.9c, respectively, shows no difference among the Green channels.
4.3 Image Creation Process

It can be concluded that the green cells are equally sensitive to light, but one of them is more sensitive to the red light than the other or the red cells are bleeding red information into it. This phenomenon occurs in photogrammetric images as well and is distinguishable in places where retro-reflective targets are present. The Green channel RAW values for a Nikon D200 camera, showing the central part of a retro-reflective target, are presented in Table 4.1. In this table the first row contains the green/red pixels while the second holds the blue/green pixels. In a similar manner to the test performed, the blue/green row demonstrates higher values/sensitivity. As can be seen, the Nikon D200 camera is indeed affected by this problem.

Table 4.1: Values of the Green channel (12-bit) for the central part of a retro-reflective target.

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1841</td>
<td>0</td>
<td>3794</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4022</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4022</td>
</tr>
<tr>
<td>3796</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4022</td>
</tr>
<tr>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4022</td>
</tr>
<tr>
<td>3768</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>4022</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4021</td>
<td>0</td>
<td>4022</td>
</tr>
<tr>
<td>1668</td>
<td>0</td>
<td>3551</td>
<td>0</td>
<td>3967</td>
<td>0</td>
<td>3967</td>
<td>0</td>
</tr>
</tbody>
</table>

4.3.5 De-noising

De-noising is commonly used in image processing in order to improve the brightness or colour information produced by the sensor. Due to the various error sources, no sensor can provide perfect data and image sensors are not an exception. The origin of noise can be related to various aspects, as explained in the previous section, including sensor inaccuracy, quality and the quantisation process. It is quite difficult to accurately model such random and unpredictable sources of noise. Recently released cameras use de-noising in their imaging pipeline in order to improve the resulting image.

With digital SLR cameras, image noise becomes more apparent for high ISO
sensitivity or under difficult light conditions. Noise may be visible as colour (chroma) and spot brightness (luminance), and it is generally more apparent in darker areas of the image where smooth tones are expected. Fortunately, software processing can help restore smoothness without excessively blurring details. A common practice is to perform noise removal in the final full colour image. However, dealing with noise in the earliest possible stage of the image creation process is a much more promising approach. Noise filtering before the demosaicing of RAW data will result in more accurate results. Performing demosaicing with noisy data will add artificial, synthetic noise to the data, resulting in a noise model which is even more complex than that encountered at the beginning stage. A good de-noising algorithm should preserve all edges and high-contrast details, while getting rid of most of the noise. De-noising is not a simple process and sometimes additional de-noising is applied in the final image in order to improve the results.

There are a number of open-source and commercial systems that incorporate the de-noising procedure before CFA demosaicing. However, most commercial software does not disclose information regarding the specific algorithms employed. In the open source community, the most commonly used software packages employ Wavelet De-noising or Fake Before Demosaicing De-noising (FBDD) (Gózd & Rodríguez, 2010). Wavelet de-noising can provide good results, but sometimes it can be quite aggressive, so that some detail is not preserved. It operates on each of the four channels (R, G1, B, G2) of the RAW image separately. FBDD is smoother as it eliminates most visible noise leaving 99% of the detail intact, while significantly reducing chroma noise. This produces good results using different de-noising techniques applied after demosaicing. However, while FBDD de-noising does not affect the contrast or saturation, it leaves a lot of small noise (chroma and luminance), so it is preferable in really noisy images. Additionally, it may decrease sharpness and contrast in the red channel (Gózd & Rodríguez, 2010).

Wavelet de-noising can also be performed at the final image formation stage as well. However, there are more sophisticated interpolation algorithms available for this task, such as the GREYCstoration (GREYC, 2011).
4.3.6 Colour Scaling

In order to provide good colour reproduction, all digital cameras include a white balance operation. White balance requires adjusting the RGB signal levels provided by the image sensor to correct for the colour temperature of the light source used to illuminate the scene. This signal adjustment must be addressed when the RAW conversion is performed. Even though there are various ways to achieve white balancing, the most common is to transform the data acquired by the sensor to new values that are appropriate for colour reproduction or display. This can be achieved by balancing the colour data from the colour channels of the sensor, depending on the ‘colour temperature’ of the light source, essentially by scaling each channel. Most cameras have several custom white balance settings in addition to an automatic setting that estimates the white balance based on the spectral content of the image. Automatic white balance algorithms work well most of the time, but they can fail for unusual subjects, for example, when a strong colour dominates the scene. When RAW images are used, white balance information is just embedded in the metadata. Thus, there is the possibility of using more refined algorithms when post-processing is performed.

An additional step is required when the RAW image is post-processed or when the camera has to save images into another format. The pixel colour intensities are generally quantised at 10-, 12- or 14-bits per channel. Thus, in order to store the RGB colour information for a pixel, 30-bits (3 and \( \frac{3}{4} \) bytes), 36-bits (4 and a half bytes) or 42-bits (5 and \( \frac{1}{4} \) bytes) are required depending on the sensor. Unfortunately there is no file format that supports such bit values; thus, each colour has to be represented in 16-bits by adding blank values to the file in a process called padding. This would result in each pixel requiring 6 bytes, but the 16-bit file could contain all of the original information in the RAW file as well as interpolated data. Although possible, this does result in large files which could pose a problem in the in-camera processing. In the case of post processing, where there is no problem of storage and processing power, 16-bits are preferable so that there is no loss of the dynamic range of the sensor. Unfortunately, for in-camera processing, a selection has to be made by the manufacturers depending on the hardware. The first option is to keep the existing bits for the subsequent
4.3 Image Creation Process

correction steps until the final stage of storage is reached, where the data is down- 
scaled to 8-bits. The second possibility is to down-scale the colour information to 8-bit at this stage, in order to facilitate the upcoming calculations. Obviously, cameras that support 16-bit do not require the above step, unless storing the images in an 8-bit file format.

Optionally, post-processing software can be applied and sometimes cameras can perform some type of optical correction to compensate for the lens errors. One of the most common is a chromatic aberration correction where the red and the blue channels are scaled by certain factors. This type of correction has been investigated in the field of photogrammetry by Luhmann et al. (2006a).

4.3.7 Bayer Interpolation

The step of demosaicing is the most important part in the true colour image creation process. When a camera records in any other format than RAW, the RAW image is converted to the desired format in the camera. The in-camera interpolation algorithm is a compromise between reasonable quality of output and the time it takes to convert the RAW image into the desired format. This whole process has to be performed in a limited amount of memory and has to be quick enough for the camera lag to be minimal, while providing good results for all the images. Indisputably, this is not beneficial for photogrammetric purposes, as the process may result in accuracy loss in the intensity information due to the interpolation. When the camera saves the RAW image, it is saved as it was captured by the sensor and all the conversion can take place in the computer after the image has been downloaded. The difference is that at this stage there is no problem to run a less efficient post-processing step employing more sophisticated algorithms. Additionally, a variety of different algorithms can be tested and evaluated with the purpose of finding the most suitable for a particular set of images.

Various algorithms include an additional step to refine the image. This post-demosaicing refinement utilises the recently fully populated image colours to optimise their estimates by using the same interpolation scheme in order to generate more consistent colour values and thus reduce the demosaicing artefacts.
4.3.8 Image Sharpening

The processing used in digital cameras often includes edge sharpening. The algorithms that are used in the Bayer interpolation differ in their ability to both produce clean edges and handle the level of noise. Additionally, image sharpening is used to correct image blur caused by the lens, optical anti-aliasing filter, and the sensor’s aperture as well as to provide a subjectively sharper image. There are various ways to enhance the edges of objects and take care of noise. However, in-camera processing does not use sophisticated algorithms and thus most of the time a median filter, i.e. a Gaussian filter, is applied to reduce the noise in the signal. With this technique it is possible to sharpen the image as well, by subtracting the blurred image which is created with the help of Gaussian blur, from the original image, making sure there is compensation for the loss of brightness.

4.3.9 Colourspace Conversion

A colourspace is an abstract mathematical model describing the way colours can be represented as tuples of numbers, typically as three colour components. Every device has its own description of colour, based on what the manufacturer of the device thinks it should be, or on the technical limitations of the device, and rarely, if ever, do they match each other in terms of numerical values. It is due to this situation that colourspace space conversion is needed, so that one can maintain consistent colour information from one device to another. This is achieved with a colour-managed workflow incorporating and implementing International Colour Consortium (ICC) profiles.

ICC profiles are nothing more than files that numerically describe a characteristic of a certain device or colourspace. They are used when converting from one space to another. When doing so, two profiles are needed, a source profile and a destination profile. The two widely used profiles among the DSLR cameras are sRGB and AdobeRGB. The translation between different colourspaces is not absolute, which results in perturbations of the original colour information.
4.3 Image Creation Process

4.3.10 Gamma Correction

Gamma correction controls the overall brightness of an image. Images that are not properly corrected will look washed out. Gamma correction is important in displaying an image accurately. To explain gamma correction, a basic knowledge of how monitors and cameras work is needed.

Digital sensors, both CCD and CMOS, perceive light in a linear fashion. That means that the voltage generated in each pixel and hence the pixel level emerging from the A-to-D converter is proportional to exposure. So, for example, in the real world, two lights of the same intensity directed on the same spot will illuminate the area with twice the intensity of a single light. This property is known as linear gamma. However, neither all electronic devices nor human vision behave in this way. The human eye perceives light in a logarithmic fashion. Twice as much light is perceived as brighter, but not twice as bright. Conversely, eyes are more sensitive to shadow detail than a camera sensor. For electronic devices, if the voltage running over an LCD display, for example, is doubled, the intensity of light emitting from that crystal does not double, as it does in nature, and therefore is not linear either. In order to obtain proper viewing of the colours, a correction known as gamma correction is applied. Gamma correction is necessary to redistribute tonal information to more closely correspond to the way the human eye perceives brightness. An example of such a gamma correction curve is shown in Figure 4.10.

Hence, the colourspaces used to map the pixel levels to visible colours for standard image files are intentionally non-linear. The luminance represented by a pixel is not proportional to the pixel level but is related to the pixel level by a transfer function that is approximated fairly well by a single type of mathematical function, a power function. This power function has the general equation

\[ \text{Lum} = \text{pixel level}^\gamma \]

To obtain the pixel intensity values, the RAW output of the image sensor, which is proportional to exposure, must be converted to a standard colour space using
4.3 Image Creation Process

Gamma is the exponent of the equation that relates the luminance to pixel level. Every colourspace has a characteristic gamma. In Figure 4.10, the first process of converting light energy (RAW data) to file pixels is shown. The second process of converting file pixel to display or print is shown in Figure 4.11. The two processes are complementary in the sense that when they are combined they give a straight line.

It should be evident from these graphs that RAW conversion compresses pixel levels representing high luminance and expands pixel levels representing low luminance. This means that the converted file, with $\gamma = 2.2$ in this example, has relatively fewer pixel levels in the highlights and more in the shadows. As mentioned above, this turns out to be an advantage when human vision is considered. However, perturbations in the image pixel values can significantly alter the metrics of the image and affect the target scanning and centroiding.

**Figure 4.10:** Gamma correction curve.

\[
pixellevel = \text{RAWpixellevel}^{1/\gamma} \approx \text{exposure}^{1/\gamma}
\]
process, which will subsequently have an impact to the following photogrammetric procedures.

### 4.3.11 Digital Image Formats

The final step in the process of colour image formation is the storage of the reconstructed true-colour image. The two file formats that are mainly used in digital imaging, are JPEG and TIFF, with the first being nowadays more commonly employed. The JPEG file format is a lossy compression image file format which contains 8-bits \(^1\) per channel. It supports a variable compression which allows for a selectable trade off between storage size and quality. The TIFF file format is a much more flexible format than JPEG, as it operates as a container that can hold various additional objects such as other JPEGs or vector-based objects. The ability to store image data in a lossless format is what makes TIFF a really useful archiving format. It is also the main reason that TIFF is

---

\(^{1}\)Non-official patches are available for lossless compression, as well as 16-bit encoding data per channel (256 distinct values)
utilised widely. Additionally, TIFF offers the ability of using Lempel-Ziv-Welch (LZW) compression, a lossless data-compression technique, in order to reduce the file size. Due to the special way that TIFF holds the data just as a container, it can support either integer or floating point values of 8-, 16- and 32-bit data depth per channel.

4.3.12 Summary
Outputting pixel values to any file format, apart from RAW, involves processing the data in order to produce a file that reflects all the discussed pre-processing steps. When the processing is performed in the camera, many choices are made for the user. But these might not be optimal from a photogrammetric standpoint. The outcome of this process are files that are meant to be ‘ready for use’ with the correct appearance, without any further processing being needed. This stands true especially for JPEG images, since JPEG is a format that was created with the main purpose of being a display format. To save storage space, it removes all details not perceptible to the human eye. However, should any mistake be made (too dark, wrong colour balance, etc.) in any of the image creation steps, it is impossible to correct. Raw photos, on the other hand, preserve the full colour gamut of the camera without applying any type of correction to the pixel colour values.

4.4 Image Storage in Photogrammetry
Not surprisingly, digital image creation and storage has not been developed with photogrammetry as its main focus. Traditionally, JPEG or TIFF images are used in close-range photogrammetry. However, the heavy processing performed to acquire the final image introduces various alterations to the original pixel values and potentially their geometric relationships. This can have a measurable impact in photogrammetric exploitation of the imagery. Cronk (2007) has shown that while there is no significant difference in target recognition, there is definitely a difference in feature point centroiding when JPEG instead of RAW images are used, and further investigation on this issue is still warranted.
4.4 Image Storage in Photogrammetry

The explanation of the JPEG/TIFF image creation given in the foregoing section illustrates the extent of processing that needs to be performed. Indisputably, the use of RAW images is a better choice for photogrammetry as all the described pre-processing stages are either embedded in the metadata or they are not performed. In this Section, an investigation into the use of RAW images in photogrammetry, and into the problems that accompany them, is presented. An enhanced method developed to handle RAW images for photogrammetric use, which eliminates most of the issues, is also proposed.

4.4.1 RAW

Under normal circumstances, image measurement in photogrammetry requires sharp and vivid images for accurate selection of feature points. However, this is something that one could argue about, especially for industrial or high precision photogrammetry. In such cases, the scene tends to be more structured as the measurement process most often relies on the use of retro-reflective targets and thus the need for ‘optimal’ true-colour images diminishes. As a result, the most significant operation is the precise recognition and identification of the targets. Ideally, a photogrammetric retro-reflective target would consist of a black background colour (pixel brightness values close to the minimum possible value) and bright circular or elliptical white blobs (pixel brightness values close to highest possible value). The identification of such targets reduces to calculating the intensity difference of neighbouring pixels as explained in Section 2.6. Therefore, the use of RAW imagery is feasible but nonetheless limited to specific applications.

Working with RAW imagery, though, does not come without problems. It is rather difficult to read the actual sensor data from such files as most camera manufacturers do not provide the specifications for their RAW file implementation, and the data in the image file is often encrypted. Additionally, the metadata that contains all the camera information needed to decode the image is much more complicated to decrypt. Nonetheless, an open-source project, named ‘dcraw’, has provided reverse-engineered decryption algorithms along with access to the majority of the RAW proprietary file formats. Various commercial software can also read RAW image files, but this operation is sometimes limited in the
information that can be extracted from the metadata due to licensing issues.

It should be noted again that RAW files do not provide a true-colour image; instead, every pixel holds information only for a specific channel. Taking care of the gaps is the most significant task in scanning the images for retro-reflective targets. Since the scanning algorithm adopted for this research employs weighted centroiding, its performance is optimal only when the green channel is used, since the Red and Blue channels each occupy only \(\frac{1}{4}\) of the image. An important aspect of this step is that prior knowledge of the CFA array is needed so that the colour of each pixel is known. For example, there are four different variations for a Bayer RGB array, the top-left corner of which is shown in Figure 4.12. This is essentially one of the main drawbacks to usage of the RAW format in photogrammetry as it would imply that knowledge of the CFA array of virtually every sensor on the market is known and handled properly.

\[
\begin{array}{cccc}
R & G & G & R \\
G & B & B & G \\
B & G & G & R \\
G & B & R & G \\
\end{array}
\]

Figure 4.12: The four possible variations of the Bayer RGB sensor.

### 4.4.2 Interpolated Raw Images

In order to gain the advantages of using the RAW format without the arduous path of trying to support all camera brands, a different solution is proposed. By using dcraw, it is possible to decrypt a RAW file and create a true-colour image. However, due to dcraw’s open-source nature, it can be modified to accommodate photogrammetric needs. This can be accomplished by changing the actual process of the image creation by removing every step that can possibly modify the acquired RAW values. Evidently, the only step that is necessary is the CFA interpolation for the creation of a full RGB image. Optionally, specific correction steps, such as post-demosaicing refinement, that would improve the outcome, can be used. Another advantage of this procedure is that due to the creation of full colour RGB images, they can be used for all photogrammetric applications. Additionally, since no gaps are present in the image, the scanning
algorithm can be used in either of the three channels, which is useful for colour coded targets (Cronk, 2007) or for other applications (Luhmann et al., 2006a).

4.5 Software Development & Implementation

In order to test the use of RAW image files in photogrammetry, access to the RAW data values of the sensor is needed. As dcraw is a standalone application, an equivalent C++ library version of this software, named LibRaw (LibRaw, 2011), was selected for the reported research. The use of a library leads to easier implementation within Australis (Photometrix, 2011), the testing software platform used. LibRaw provides an application programming interface (API) to its internal functions which in turn makes it possible to gain RAW image support.

Since the colour information is not altered in either of the proposed methods, the image values remain in the bit depth that was used by the camera. Generally, the intensity range is either 10-, 12-, 14- or 16-bits. In order to handle each bit depth properly, various internal components of the current functions that take care of displaying and scanning the images had to be changed. Furthermore, for the interpolated RAW files, the storing of the data posed another problem. Re-inventing an image file format that would properly support such bit-depths would require considerable effort. For that purpose, the TIFF file format was selected to be the container of the pixel information. Apart from the case where the bit-depth is 16, the remaining three bit-depths are not supported by the TIFF file specification. Thus, in a similarly way to the the RAW file format, special attention is needed when displaying and scanning the images. In order to support every camera, these four different bit-depth values have to be properly handled. However, storing the interpolated data is not a necessity, as the RAW files can be used to gain access to the colour information at any time, which is a more efficient approach.

4.5.1 Demosaicing Algorithms

For the interpolated RAW files, a demosaicing algorithm is needed in order to interpolate the missing colour information. Careful selection of the demosaicking
4.5 Software Development & Implementation

algorithm is required as it is an important aspect of full colour image creation. Dcraw and consequently LibRaw support four types of interpolation:

- Bilinear

- Interpolation using a Threshold-based variable number of gradients (VNG) (Chang et al., 1999)

- Patterned Pixel Grouping (PPG)

- Adaptive homogeneity directed demosaicing (AHD) (Hirakawa & Parks, 2003)

Following a careful initial review of the literature on demosaicing algorithms, various newer algorithms that could possibly perform better than these four types were found. Seven additional demosaicing methods were therefore implemented in LibRaw for testing:

- AHD interpolation with built-in anti-aliasing, developed by Lee (2009a)

- Colour demosaicing via directional linear minimum mean square-error estimation (LMMSE) (Zhang & Wu, 2005), developed by Martinec (2010)

- Colour demosaicing using variance of colour differences (VCD) (Chung & Chan, 2006), developed by Lee (2009b)

- Adaptive filtering for colour filter array demosaicing (AFD) (Lian et al., 2007), developed by Lee (2010)

- Combination of AHD and VCD, developed by Lee (2009b)

- DCB, created and developed by Góźdź (2010)

- Aliasing Minimization and Zipper Elimination (AMaZE), created and developed by Martinec & Lee (2010)

Furthermore, for VCD and DCB, a post-demosaicing refinement step was also available and was implemented. For the VCD algorithm the refinement was based on the Enhanced Effective Color Interpolation (EECI) algorithm (Chang & Tan,
4.6 Experimental Application

2005), which was developed by Lee (2009b). The adapted algorithms that were implemented and incorporated by the author into LibRaw have been accepted and are publicly available as a part of the LibRaw library (LibRaw, 2011).

Availability of source code for the demosaicing algorithms, regardless of the programming language, played an important role in the selection process. Those that were implemented are considered among the state-of-the-art algorithms for image demosaicing. Every algorithm has its own advantages and disadvantages. Some perform well only on noise-free images, while others show advantages on noisy images. The most recent algorithms are the DCB and AMaZE, and these are considered to perform well for different type of images.

4.6 Experimental Application

4.6.1 Description

The principal aim of the experimental program conducted was to examine the extent of variation in the mean positional standard error of object target point coordinates which accompany perturbations in the positioning of the centroid of the retro-reflective targets in a typical close-range photogrammetric network adjustment. By utilising the ability of modern DSLR cameras to store both RAW and JPEG images, it is possible to use the RAW images to create various additional datasets, as proposed in Section 4.4.

Each of the datasets was created via a unique process, such as utilising a different demosaicing algorithm and/or in the adoption of a specific step of the image creation process. By processing and comparing the impact in the accuracy of object point determination for each dataset, it was possible to quantify the merits of different creation processes. The main intention of this comparison was to reveal the differences resulting from:

- use of the typical JPEG image creation process datasets
- the adoption of different demosaicing algorithms, as well as different channels, when compared to each other as well as compared to the JPEG and RAW datasets
To achieve the experimental aims, a total of 43 photogrammetric adjustments of the same network were computed. The experimental test network adopted was not atypical of an imaging geometry that might be employed in an engineering or industrial photogrammetric survey. Figure 4.13 illustrates the geometry of the network, which comprised 42 images. The recorded images were taken with different kappa rotations at each station, with each rotation being either 0°, 90° or 270°. A total of 899 3D targeted points were present in the scene, these comprised 34 coded and 627 single targets. A Nikon D200 camera with a Tokina 18mm fixed focal length lens was used for the photography. It should be noted that the resulting RAW imagery from a Nikon D200 has a bit depth of 12-bits. The processing of the datasets was performed in Australis, where a 10-parameter correction model was used.

For all datasets, each of the RGB channels was used separately for the scanning and centroiding of the retro-reflective targets. Firstly, the Green channel results are presented, as it is the channel with most information and thus is expected to
produce the most accurate results. The evaluation of the use of the Red and Blue channel follows in the next sections. It should be noted that the use of the Red and Blue channels is mainly applicable in special cases, such as where colour coded targets are employed, or for applications such as modelling chromatic aberration (Luhmann et al., 2006a). Experimental testing was also performed for the modelling of the chromatic aberration, as proposed by Luhmann et al. (2006a), with the use of RAW imagery as it was considered that the initial approach of using the RGB channels derived from JPEG imagery was not appropriate due to the way that the colour information is handled in JPEG compression.

A second dataset was also acquired with a GSI INCA3 (Geodetic Systems Inc., 2011) camera so that an object point coordinate comparison could be carried out. The network recorded with the INCA3 was used as a ‘standard’ in view of its significantly higher accuracy. Even though the datasets were acquired at the same time to avoid any perturbation in the object point array, an absolute comparison of the object point accuracies is meaningless in this case as there are differences in the camera sensor (Bayer RGB sensor compared to a grayscale), the actual datasets (quantity and location of the stations), as well as the processing of the data, which for the INCA3 dataset was performed in V-STARS (Geodetic Systems Inc., 2011) using a different camera calibration model. The network acquired with the GSI INCA3, shown in Figure 4.14, was comprised of 59 convergent stations where each image was recorded with a kappa rotation of either 0°, 90°, 180° or 270°. The results of the INCA3 network adjustment are presented in Table 4.2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(\sigma_X)</th>
<th>(\sigma_Y)</th>
<th>(\sigma_Z)</th>
<th>Overall Bundle Pixel RMS ((\mu m))</th>
<th>Pixel Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCA3</td>
<td>0.005</td>
<td>0.009</td>
<td>0.004</td>
<td>0.006</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 4.2: Estimates of mean positional standard error for the INCA3 network.
4.6 Experimental Application

4.6.2 Results per Channel

4.6.2.1 Green Channel

Table 4.3 lists the magnitude of the positional standard error $\sigma_X$, $\sigma_Y$, $\sigma_Z$ along each axis, as well as the mean positional standard error $\sigma_{XYZ}$ for the Green channel. The first column presents the different algorithms as well as the JPEG and RAW datasets. The sixth column presents the RMS of the $xy$ residuals and the last column shows the estimated image referencing pixel accuracy.

Generally, all the examined cases except for the JPEG present no variations in the RMSE of the $xy$ residuals, with the Bilinear, DCB-Enhanced, and the RAW having the lowest values. As regards the mean positional standard error
among the demosaicing algorithms, the differences were not of a significant magnitude. The Green channel occupies 50% of the sensor size and thus even the simplest Bilinear algorithm provided exceptional results. However, even though a straightforward algorithm such as Bilinear interpolation can provide good results, in the experiment performed it was seen that it had a large number of rejected 2D image points compared to the other datasets, which could be related to the demosaicing process. The highest accuracy of 0.0127 mm is provided by the RAW and DCB Enhanced demosaicing algorithm that presented the lowest RMSE of the xy residuals. The AFD, AMaZE and LMSSE algorithms followed, with an
accuracy of approximately 0.013 mm, proving that they are among the current state-of-the-art demosaicing algorithms that can be used for photogrammetric purposes. The use of the proposed method provides an approximate 30% increase over the green JPEG channel. It should be noted that the JPEG dataset presented two false positive target recognitions that were not presented in any of the other datasets. As expected, the higher dynamic range provided by the use of RAW imagery allows for better recognition results. This can be really helpful in cases where photogrammetric measurements are not as controlled as in this experiment.

The aim of this experimental evaluation was to highlight the differences of the examined cases per channel. A more accurate comparison would be to compare the Green channel to a JPEG dataset that is not limited to only one channel. Due to the way that JPEG encodes colour the colour information is not as accurate when referring to a specific channel. The representation of the colours in a JPEG image consist of three components:

1. $Y'$, known as the luma component that represents the brightness of a pixel,
2. $C_B$, being the first chroma component that represents the blue colour, and
3. $C_R$, being the second chroma component that represents the red colour

Even after the decoding of the JPEG image, the colours are still 'mixed', due to the initial creation of the JPEG colour components. Thus, the use of only a specific channel for the intensity centroiding algorithm is not optimal, and in the case of retro-reflective coded targets, even though the green channel is preferred for the scanning of the targets, the centroiding process should operate on all three channels. Table 4.4 show the accuracy of the same JPEG dataset with the different centroiding procedure. Overall accuracy was increased by 14%
4.6 Experimental Application

compared to that obtained using only the green channel. The DCB Enhanced dataset presents an approximately 24% increase in accuracy over the JPEG case. Nonetheless, an accurate figure cannot be given as it may vary depending on the dataset as well as the accuracy (bit-depth) of the colour information which can vary from either of 10-, 12-, 14- or 16-bits. It is apparent from the discussion so far, that the use of the proposed methods to create the image datasets for photogrammetric purposes has a significant advantage over the use of typical JPEG imagery.

4.6.2.2 Red Channel

Table 4.5 presents the results of the datasets for the Red channel, in the same format as for Table 4.3. As initially foreseen, a substantial variation is present in the mean positional errors of each demosaicing case. The RMSE of image coordinates from the bundle adjustment fluctuates from 0.24 to 0.54µm with the AFD, LMMSE and VNG having the lowest values. As expected, the cases with the lowest RMSE values also present the most accurate referencing of the image points which consequently leads to the highest mean positional accuracies in the object space coordinates. The AFD demosaicing demonstrates the highest accuracy of 0.0136mm, followed by LMSSE with 0.014mm and VNG with 0.015mm. The remaining demosaicing algorithms present accuracies varying from 0.018 – 0.023mm. The JPEG dataset has the lowest accuracy of 0.032mm and similarly to the green JPEG dataset it has two false positive target recognitions.

Despite the Red channel occupying only the 25% of the image sensor, some algorithms succeeded in producing accuracy close to that acquired by the Green channel. Compared to these datasets, the JPEG case presented an exceptionally large number of rejected 2D points. The use of interpolated RAW data showed at minimum an approximately 28% increase achieved by the modified AHD algorithm. The highest accuracy was produced by the AFD demosaicing, which showed a 57% increase compared to the JPEG.
4.6 Experimental Application

Table 4.5: Estimates of the mean positional standard errors for Red channel.

<table>
<thead>
<tr>
<th>Demosaicing Case</th>
<th>σ_X (mm)</th>
<th>σ_Y (mm)</th>
<th>σ_Z (mm)</th>
<th>Overall RMS (µm)</th>
<th>Bundle Pixel Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>0.027</td>
<td>0.045</td>
<td>0.023</td>
<td>0.032</td>
<td>0.54</td>
</tr>
<tr>
<td>Bilinear</td>
<td>0.15</td>
<td>0.030</td>
<td>0.015</td>
<td>0.020</td>
<td>0.34</td>
</tr>
<tr>
<td>VNG</td>
<td>0.014</td>
<td>0.021</td>
<td>0.011</td>
<td>0.015</td>
<td>0.27</td>
</tr>
<tr>
<td>PPG</td>
<td>0.021</td>
<td>0.032</td>
<td>0.016</td>
<td>0.023</td>
<td>0.40</td>
</tr>
<tr>
<td>AHD</td>
<td>0.021</td>
<td>0.031</td>
<td>0.016</td>
<td>0.022</td>
<td>0.39</td>
</tr>
<tr>
<td>AHD-mod</td>
<td>0.021</td>
<td>0.032</td>
<td>0.016</td>
<td>0.023</td>
<td>0.39</td>
</tr>
<tr>
<td>LMSSE</td>
<td>0.012</td>
<td>0.020</td>
<td>0.010</td>
<td>0.014</td>
<td>0.25</td>
</tr>
<tr>
<td>AFD</td>
<td>0.011</td>
<td>0.019</td>
<td>0.011</td>
<td>0.014</td>
<td>0.24</td>
</tr>
<tr>
<td>VCD</td>
<td>0.016</td>
<td>0.027</td>
<td>0.016</td>
<td>0.020</td>
<td>0.34</td>
</tr>
<tr>
<td>AHD, VCD</td>
<td>0.019</td>
<td>0.029</td>
<td>0.015</td>
<td>0.021</td>
<td>0.37</td>
</tr>
<tr>
<td>AHD, VCD Refined</td>
<td>0.017</td>
<td>0.025</td>
<td>0.013</td>
<td>0.018</td>
<td>0.31</td>
</tr>
<tr>
<td>DCB</td>
<td>0.017</td>
<td>0.030</td>
<td>0.017</td>
<td>0.021</td>
<td>0.37</td>
</tr>
<tr>
<td>DCB Enhanced</td>
<td>0.017</td>
<td>0.025</td>
<td>0.013</td>
<td>0.019</td>
<td>0.32</td>
</tr>
<tr>
<td>AMaZE</td>
<td>0.021</td>
<td>0.026</td>
<td>0.016</td>
<td>0.021</td>
<td>0.36</td>
</tr>
</tbody>
</table>

4.6.2.3 Blue Channel

The Blue channel produced similar results to the Red channel, as can be seen from Table 4.6. In general, the datasets show a big variation in the RMSE of the xy residuals with the values fluctuating from 0.24 to 0.37µm. Similarly to the Red channel, the AFD, LMSSE and VNG algorithms present the lowest RMS values and consequently indicate the highest mean positional accuracy. More specifically, the AFD presents an overall accuracy of 0.0134mm and the LMSSE along with the VNG follow with 0.0138mm and 0.0147mm, respectively. Most
of the remaining algorithms have an accuracy that ranges from $0.017 - 0.019$mm with the exception of the Bilinear and AHD-mod algorithms.

As with the Red channel, the same demosaicing algorithms were able to achieve accuracies as high as the Green channel. When compared to the JPEG, the AFD had a 35% improvement in the mean positional accuracy.

**Table 4.6:** Estimates of the mean positional standard errors for Blue channel.

<table>
<thead>
<tr>
<th>Demosaicing Case</th>
<th>$\sigma_X$ (mm)</th>
<th>$\sigma_Y$ (mm)</th>
<th>$\sigma_Z$ (mm)</th>
<th>Overall RMS ($\mu$m)</th>
<th>Bundle Pixel Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG</td>
<td>0.019</td>
<td>0.027</td>
<td>0.015</td>
<td>0.020</td>
<td>0.36</td>
</tr>
<tr>
<td>Bilinear</td>
<td>0.018</td>
<td>0.032</td>
<td>0.017</td>
<td>0.022</td>
<td>0.37</td>
</tr>
<tr>
<td>VNG</td>
<td>0.014</td>
<td>0.020</td>
<td>0.011</td>
<td>0.015</td>
<td>0.26</td>
</tr>
<tr>
<td>PPG</td>
<td>0.015</td>
<td>0.026</td>
<td>0.015</td>
<td>0.019</td>
<td>0.33</td>
</tr>
<tr>
<td>AHD</td>
<td>0.015</td>
<td>0.026</td>
<td>0.015</td>
<td>0.019</td>
<td>0.34</td>
</tr>
<tr>
<td>AHD-mod</td>
<td>0.020</td>
<td>0.033</td>
<td>0.017</td>
<td>0.023</td>
<td>0.40</td>
</tr>
<tr>
<td>LMSSE</td>
<td>0.011</td>
<td>0.019</td>
<td>0.011</td>
<td>0.014</td>
<td>0.24</td>
</tr>
<tr>
<td>AFD</td>
<td>0.011</td>
<td>0.019</td>
<td>0.011</td>
<td>0.013</td>
<td>0.24</td>
</tr>
<tr>
<td>VCD</td>
<td>0.015</td>
<td>0.025</td>
<td>0.013</td>
<td>0.018</td>
<td>0.32</td>
</tr>
<tr>
<td>AHD, VCD</td>
<td>0.016</td>
<td>0.028</td>
<td>0.017</td>
<td>0.020</td>
<td>0.36</td>
</tr>
<tr>
<td>AHD, VCD Refined</td>
<td>0.015</td>
<td>0.023</td>
<td>0.012</td>
<td>0.017</td>
<td>0.30</td>
</tr>
<tr>
<td>DCB</td>
<td>0.015</td>
<td>0.025</td>
<td>0.015</td>
<td>0.018</td>
<td>0.33</td>
</tr>
<tr>
<td>DCB Enhanced</td>
<td>0.014</td>
<td>0.025</td>
<td>0.013</td>
<td>0.017</td>
<td>0.31</td>
</tr>
<tr>
<td>AMaZE</td>
<td>0.015</td>
<td>0.024</td>
<td>0.013</td>
<td>0.017</td>
<td>0.31</td>
</tr>
<tr>
<td>RAW</td>
<td>Not Applicable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.6.2.4 Summary

It is apparent that the imagery created with a higher dynamic range, in this case 12-bits, offers increased centroiding accuracy which leads to a significant impact in the internal precision of the photogrammetric network, and this was verified by the experimental program performed. The proposed approach is able to provide a much higher mean positional accuracy when compared to the use of JPEG imagery.

The results of the investigation confirmed the theoretical relationship between each channel and their accuracy. Indisputably, the Green channel is the most accurate, as it demonstrates less variation in the obtained accuracies among the examined cases. The Blue channel results show that it is more accurate than the Red channel by a small margin. It is important to note that improved results are anticipated with newer cameras that are able to record colour information with a higher dynamic range of 14- or 16-bits.

The current state-of-the-art algorithms were expected to perform better in the Red and Blue channel. The image demosaicing process however, does not aim for optimal results in photogrammetry, but instead tries to provide the most pleasing visual effect for display. Further knowledge of the exact semantics of colour interpolation and reproduction is required in order to understand why every algorithm performs in a specific manner.

4.6.3 Results for the Combined Bundle Adjustment

In this test, a combined bundle adjustment of all the observations of each channel was performed, resulting in a 126 station network. Since every image is comprised of the three RGB channels, the exterior orientation is identical for each image-channel triplet. In order to ‘fix’ the exterior orientation for every RGB triplet, the bundle adjustment was modified so that it only calculated the partials for one image per triplet. However, due to the different wavelength, which is the main effect of chromatic aberration, every channel has different calibration parameters. For this reason, three cameras were introduced to the project in order to properly accommodate the differences of the channels. A 10-parameter interior orientation model was selected. Among the demosaicing datasets, the AFD was chosen, due
to its high precision in every channel.

With this process, an enhanced accuracy of about 38% compared to the most accurate demosaicing algorithm of the Green channel was achieved, as shown in Table 4.7. Luhmann et al. (2006a) has also mentioned that an enhancement of about a factor of 1.3 can be achieved when modelling for the chromatic aberration. The results presented here also agree with this statement when compared to the proposed method of using the RAW data for the Green channel. However, when compared to the JPEG dataset, a significant accuracy increase of approximately 53% is achieved. Moreover, what is appealing about the approach discussed is its simplicity, since the required changes to software systems to implement the modelling of chromatic aberration are trivial. Even though this technique leads to a larger dataset due to the inclusion of the two extra channels, the processing time is only increased by a few of seconds.

**Table 4.7:** Estimates of the mean positional standard errors using the combined bundle approach.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\sigma_X$ (mm)</th>
<th>$\sigma_Y$ (mm)</th>
<th>$\sigma_Z$ (mm)</th>
<th>Overall RMS ($\mu m$)</th>
<th>Bundle Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFD R, G, B</td>
<td>0.007</td>
<td>0.010</td>
<td>0.006</td>
<td>0.008</td>
<td>0.24</td>
</tr>
</tbody>
</table>
5.1 Introduction

In the previous chapters, the basic analytics of close-range photogrammetry and digital image formation, as well the general least squares formulations required for providing solutions to the bundle adjustment, have been reviewed. This chapter narrows its focus to the specific challenge of developing a photogrammetric simulator. Simulation of the projective camera model has been widely used since the late 70’s, and the use of network simulators has been reported by many researchers, e.g. Mason (1995), Shih (1996), Otepka (2004) and Piatti & Lerma (2006), with every implementation differing in the level of sophistication involved.

The acquisition of images with long focal length lenses can pose practical difficulties for a number of reasons. While it may be easy to have a controlled environment for wide and medium angle lenses, provision of the required image arrangement can become quite perplexing for long focal lenses. Difficulties arise in testing a variety of scenarios, as the creation of a test field can be limited by object
space constraints when long distance measurements are performed. Additionally, acquisition of different lenses of long focal length for this research was not possible. For macro lenses, the main identified issue was the creation of coded targets of a proper size-to-object scene ratio. Thus, to facilitate the research on the use of long focal length lenses without initially being limited by other problems, a software system for network simulation was considered necessary.

The photogrammetric simulator discussed in this chapter is conceptually different from earlier implementations. Its purpose is not only to simulate the network but also to create simulated images that can actually be used for automatic exterior orientation of the network in other software. The explicit knowledge of the object point coordinates and the possibility to use virtually any camera and station settings, as well as distortion parameters, has proven to be a very useful research tool. This provides the opportunity of testing new methods and algorithms in an even more controlled environment.

To realise the simulator, various options for creating a 2D perspective image from a 3D data scene were evaluated, as discussed in Section 5.2. Section 5.3 describes the implementation of the chosen platform and the problems encountered in the process, while Section 5.4 presents an alternative option for higher accuracy simulations.

### 5.2 3D to 2D Development Options

Computers provide various tools to create a 2D perspective image from a 3D object scene. The available options can be categorised into two groups, depending on how the computer produces the results. The first group uses the graphics processor unit (GPU) directly, which is specifically manufactured to perform such calculations, while the second group relies on computations performed by the central processing unit (CPU). Both of these options are well known and used by game developers, special effects creators and general graphic designers; their aim being to render 3D scenes as 2D images.

While these methods are widely implemented, they lack the features that photogrammetrists consider important. The absence of additional distortion errors in the imaging process undermines their use in a photogrammetric network.
5.2 3D to 2D Development Options

5.2.1 Hardware Rendering

The first category relies on the computer’s graphics processor to perform the required calculations, something that is commonly known as hardware rendering. In order to gain access to such low-level commands, an API that provides access to the GPU is used. The most common implementations are:

- OpenGL,
- Microsoft’s Direct3D,
- Sun’s Java 3D,
- MESA 3D

Initially, their use was mainly intended for gaming and general software that required a 3D environment, for example 3D content creation. Nowadays, a much broader use from fields such as manufacturing, CAD, medicine and virtual reality is seen, due to advantages that they offer. GPUs provide a significant boost in performance due to their ability to perform billions of calculations per second. This has led to the creation of additional APIs that promote the use of general purpose graphics processing units (GPGPUs). Initially, manufacturers provided their own specific libraries, however over the past several years, open computing language (OpenCL) has succeeded in offering a unified approach. It should be noted that OpenCL is not exactly part of hardware rendering, as the offered API does not aim at graphics creation.

There are many accounts of these APIs in the literature and thus only a brief report will be presented here. Open graphics library (OpenGL) is one of the oldest existing APIs of its kind and nowadays it has become an open standard. Its development is undertaken by the Khronos Group (2011) in which most of the graphics card companies participate, making it widely adopted. Direct3D is
a proprietary API that has been developed by Microsoft and can only be used in Microsoft’s operating systems. The Java 3D API runs on top of either OpenGL or Direct3D, depending on the operating system used, as an extra layer providing a real object oriented concept and various additions. MESA 3D is an unofficial open-source (due to licensing) implementation of the OpenGL specifications. OpenCL is one of the newest frameworks, developed by the Khronos Group (2011) with the purpose of providing open standards for general purpose computing using GPUs.

5.2.2 Software Rendering

Software rendering uses just the CPU to perform the calculations needed to produce a simulated image and can be split into two main categories: real-time rendering, and offline rendering. Real-time rendering is not really used nowadays due to the growth in use of graphics cards. Thus, only offline rendering algorithms are considered here. Contrary to real-time rendering, performance is only of secondary priority with offline rendering. Offline rendering has been mainly used in the film industry to create high-quality renderings of lifelike scenes. Even though commercial real-time graphics hardware is achieving ever-higher quality and more programmability, the most photorealistic computer generated imagery (CGI) still requires software rendering. The algorithms that have mainly been used are:

- ray tracing,
- ray casting and
- scan-line rendering

Ray tracing algorithms simulate the way that rays of light travel in the real world. This job is performed backwards in most of these programs, for reasons that are beyond the scope of this report. Typically, each ray must be tested for intersection with some subset of the objects in the scene. Once the nearest object has been identified, the algorithm will estimate the incoming light at the point of intersection, examine the material properties of the object, and combine
this information to calculate the final intensity and colour of the pixel. Certain illumination algorithms and reflective or translucent materials may require more rays to be re-cast into the scene (Whitted, 1980). Obviously, the user has to provide the positions of the camera, light sources and every object, along with its surface material, for the scene to be created. While this technique can achieve a very high degree of realism, its main disadvantage is slow computation speed. There has been considerable effort and research towards implementing ray tracing algorithms that work effectively in real time. OpenRT is the most well known implementation that has been proven to perform well (Dietrich et al., 2004; Wald et al., 2004) and it also provides a sophisticated OpenGL-like API that can be used as a framework in other software. With the advent of OpenCL however, various optimisations have been reported.

Ray casting (Appel, 1968) works in a similar manner to ray tracing but it does not compute the new tangents that a ray of light might take after intersecting a surface on its way from the eye to the source of light. Scan-line rendering (Wylie et al., 1967) is an algorithm for visible surface determination in 3D computer graphics, that works on a row-by-row basis rather than a polygon-by-polygon or pixel-by-pixel basis. All of the polygons to be rendered are first sorted by the top \( y \) coordinate at which they first appear, then each row or scan-line of the image is computed using the intersection of a scan-line with the polygons on the front of the sorted list, while the sorted list is updated to discard no-longer-visible polygons as the active scan-line is advanced down the picture. The way that these two algorithms work eliminates the possibility of accurately rendering reflections, shadows and various other effects, but they are significantly faster. Their use is not so popular and it is hard to find implementations of the algorithms that can be easily developed as a part of a software system.

5.2.3 Selection Analysis

As part of the reported research, an evaluation of the described implementations was performed in order to decide on the most suitable options. The two main selection criteria set were the ability of at least close to real-time rendering and the possibility of modifying the algorithms so that they support features required for
photogrammetric simulation. Additionally, it was of interest to be able to create different object scenes in an efficient manner, so that various photogrammetric networks could be simulated. From the software rendering algorithms, ray tracing was tested as it is the only implementation that is still active in the computer vision community. OpenRT is a commercial ray tracing software and thus alternative options had to be found. Pov-ray, a well-known ray tracing software within the open-source community, was selected. The images produced had more than satisfactory quality, however, ray tracing was not considered sufficiently flexible, especially for the simple tasks that had to be performed. The diversity of photorealistic effects brings forth an unnecessary complexity that does not allow for automation of the process or for easy development of photogrammetric network simulator software. Furthermore, the whole process requires a significant amount of time from the project creation to the actual rendering of the image, while the images cannot be viewed until the rendering process is completed. This would require experimentation to record suitable images. It should be noted that even though specific software was evaluated, the results are considered valid for the whole group of ray tracing software. Thus, the software rendering approach was disregarded as a viable option.

All the hardware rendering options provide an API that makes it easier to implement them in software and provide a customised graphical user interface (GUI). The selection among those was performed by keeping in mind the portability in various architectures, but also the GUI framework and the selected programming language, that will be discussed in 5.3. OpenGL was thus selected since it is both cross-platform and an open standard.

### 5.3 Implementation

The photogrammetric simulator was designed with the goal of producing an efficient and user-friendly program. The Qt cross-platform application framework was chosen to deploy this application, due to the portability and advantages it offers in comparison to other user interface (UI) frameworks. The Qt framework provides convenient access to OpenGL functions and is written in the C++ programming language, which was the selected development language.
5.3 Implementation

The developed simulator software enables real-time rendering for navigation in the object space, while explicit exterior orientation can also be input. A screenshot of the simulator environment is illustrated in Figure 5.1. Automatic as well as semi-automatic test field creation, where the user provides desirable feature points, is supported. The simulator software provides pre-set settings for more than a thousand cameras for simulation purposes, while manual addition of virtually any camera is supported. In the case of manual camera entry, additional basic camera geometry is required, e.g. the sensor’s width and height, pixel size and lens focal length, so that the FoV can be computed for proper visualization and image creation. Figure 5.2 presents the camera selection interface provided to the user. For the purpose of introducing artificial error into the images due to perturbations from the collinearity equations, a 10-parameter camera model was used, as explained in Section 2.3. The main output product is a simulated JPEG image that also embeds exchangeable image file format (EXIF) metadata tags. It thus forms a complete measuring procedure that involves all the steps performed when normal image datasets are acquired, with all the relevant errors that are inherent in the JPEG image creation and the scanning algorithm. The most significant issues presented in the development were the creation of high resolution images, a process which is constrained by the graphics processor, and

![Figure 5.1: The user interface of the developed photogrammetric simulator.](image)
5.3 Implementation

Figure 5.2: Camera list in the simulator software.

the addition of the camera error model, along with the creation of virtual coded targets.

5.3.1 High Resolution Image Creation

Graphics cards have specific hardware limitations regarding the image resolution that they are able to support. Current graphics cards offer a maximum digital resolution of $2560 \times 1600$. In order to create a simulated image of a size similar to those created by digital cameras, much higher resolution has to be achieved. Newer GPUs support a feature called frame-buffering, which provides the fastest and easiest way of manipulating graphics in a virtual device with the help of
5.3 Implementation

OpenGL. Many old graphics cards do not support a frame-buffer device at all, or if they do it is limited to a low resolution, while current state-of-the-art GPUs support a maximum frame buffer resolution of up to $16384 \times 16384$ (AMD/ATI, 2011; Intel, 2011; nVidia, 2011). In order to support both old and new cards, an algorithm called ‘tiled rendering’ was also implemented. This process subdivides

![Figure 5.3: Example of tile rendering.](image)

the image into a regular grid comprised of smaller resolution images, as illustrated in Figure 5.3. These smaller images can then be rendered separately without the use of the frame-buffer device. Every rendered part is saved and when the whole rendering process is completed, the pieces are merged to create the final high resolution image.

5.3.2 Camera Error Model

None of the hardware or software rendering options detailed above offer a camera distortion model. The way that OpenGL handles the projection transformation allows for the addition of the principal point offset directly to the projection matrix for rendering. However, as previously stated, in digital close-range photogrammetry, the image distortion model commonly used is a 10-parameter model. To achieve its implementation, an additional rendering step after the image creation has to be performed as a post-processing procedure. In this step, distortion is added to the current error free image that is rendered by OpenGL.

The image rows and columns of the distorted image are iterated and the
corresponding location in the error-free image is found in millimetres for each pixel. This is followed by an interpolation in order to calculate the proper intensity and colour of the pixel. This process is discussed in detail in the next section. The algorithm pseudo-code is provided in Algorithm 5.1.

Algorithm 5.1: Distortion addition algorithm pseudo code for OpenGL.

```c
for (int y = 0; y < nImageRows; y++)
{
    for (int x = 0; x < nImageCols; x++)
    {
        CorrectX_mm = getCorrectX_mm(x);
        CorrectY_mm = getCorrectY_mm(y);

        RGB(x, y) = getInterpolatedValue(CorrectX_mm, CorrectY_mm);
    }
}
```

Figure 5.4 shows the camera calibration parameter dialog implemented in the developed simulation software.

5.3.3 Virtual Targets

The concept of coded targets can be recreated in the virtual 3D environment, provided that the internal geometry comprising the targets are known. This project utilised the layout of coded targets that was developed by Cronk (2007). As long as the relative coordinates of the centers of each of the code bit is known, the rendering is simplified to the drawing of single targets on a black background. The rendering of a target in plain white colour was expected to provide better recognition, due to the high contrast with a black background. To the contrary, experiments performed showed that the scanning algorithm was unable to recognise the majority of targets properly.

A more detailed investigation had to be performed into the various reasons responsible for the quality of the targets. The rasterization process, the interpolation method, the compression of the JPEG images and the way that the scanning algorithm works were the main elements responsible for the outcome. The analysis of real imagery showed that retro-reflective targets produced a radial
fading intensity effect, starting from the centre of every code bit. This effect was reproduced in the simulator as well, for every code bit, with a central white circle along with five outer consecutive circle rings, each with a progressive 5% less intensity value, as shown in Figure 5.5. This approach yielded far better results for recognition and it was then possible to perform automated measurements.

Since the interpolation methods used to add the distortion in the image impact upon the accuracy of network orientation, various interpolation algorithms were developed and tested:

- zero order interpolation: nearest neighbour
- first order interpolation: bi-linear interpolation
- second order interpolation: bi-cubic convolution or Lagrange polynomials

In order to evaluate the various interpolation methods, the exact same networks
were created and then processed with Australis (Photometrix, 2011). The theoretical accuracy of a centroid operator is approximately $0.03 - 0.05$ pixel if circular or elliptical white targets on dark background are used (Luhmann et al., 2006b).

### 5.3.3.1 Nearest Neighbour

Nearest neighbour is a simple method of interpolation that is fast. Experimental testing showed that a systematic error of approximately $1.2\mu m$ ($0.2$ of a pixel) in the bundle adjustment, or $0.2mm$ in the XYZ coordinates of the object points (1 : 30000), was introduced in perfect data. The nearest neighbour interpolation method is known to lead to approximate and visually poorer results, which is the main reason for the systematic error.

### 5.3.3.2 Bi-linear Interpolation

Bi-linear interpolation was evaluated next, with the results being significantly improved. This interpolation method takes into account a $2 \times 2$ matrix of adjacent pixel values. The weight value given to each pixel is calculated based on the relative area that is contained by the new pixel, as illustrated in Figure 5.6.

Test cases showed that the bundle adjustment error decreased to approximately $0.3\mu m$ ($0.05$ of a pixel), while the equivalent error in the XYZ coordinates...
dropped to 0.01\,mm or 1 : 130000. This interpolation method led to a more accurate recognition of the target centroid and thus a more precise recovery of the camera parameters and object point coordinates.

5.3.3.3 Bi-cubic Interpolation

The Bi-cubic interpolation method was also implemented and tested in order to compare it with the bi-linear approach, since it has proven in most cases to produce better results. It uses a 4x4 environment for the interpolation, which results in computations up to 10 times slower than the nearest neighbour algorithm. The algorithm used to calculate the bi-cubic convolution is

$$df(x) = \begin{cases} 
(\alpha + 2) |x|^3 - (\alpha + 3) |x|^2 + 1, & \text{for } |x| < 1 \\
\alpha |x|^3 - 5\alpha |x|^2 + 8\alpha |x| - 4\alpha, & \text{for } 1 \leq |x| < 2 \\
0, & \text{for all other cases}
\end{cases} \quad (5.1)$$

Different $\alpha$ values presented no significant differences in the accuracy of recovery of the camera calibration, exterior orientation and object space coordinates, with the final chosen value being $\alpha = -0.75$. The results of this interpolation method were similar to the bi-linear approach, and due to the extensive calculations and computation time it does not offer any significant advantages over the bi-linear approach.
5.3.4 Error Analysis and Additional Methods

An investigation into the error introduced in the simulator was performed in order to discover possible optimisations in the applied procedure. The error that was presented can be considered for further analysis as having three components:

- Rasterization,
- Interpolation, and
- JPEG compression

The rasterization process is performed by OpenGL when rendering the elements and is rather difficult to investigate. Thus, in order to improve the rasterization process, alternative choices of rendering have to be considered. The interpolation error can be analysed by removing the post-processing phase of the introduced distortion. In the experimental testing performed on the same simulated networks, the bundle adjustment error was reduced to 0.2\( \mu m \) (0.03 of a pixel) and the accuracy of object space coordinates determination was 1 : 150000 when no interpolation is performed. The relevant error in the interpolation process is approximately 0.02\( mm \) in the object point coordinates. In order to find what components of the remaining error belong to rasterization and to JPEG compression, respectively, the same process can be performed using the lossless TIFF image format. This procedure could not be evaluated as the correct handling of TIFF images was not possible within the Australis platform, although steps are being taken to alleviate this issue for future research.

5.4 Implementation of Image Coordinates Files

A different technique that does not have to deal with the rasterization process of the JPEG image creation was considered essential in order to increase the precision. The Australis platform has the ability to read and use image coordinate files (*.icf) that contain just \( xy \) image point measurements. This feature is common in smart cameras, e.g. the INCA (Geodetic Systems Inc., 2011) camera,
where a microprocessor is also included to handle all the image processing tasks such that the measured coordinates can be exported directly. The creation of such files is trivial in an environment such as the simulator, since the transformation calculations can be acquired directly from OpenGL. Similar approaches have been used by many industrial metrology companies, such as Geodetic Systems Inc. (2011) and Geometric Software (2011).

The creation of an .icf file can be performed with high accuracy since this procedure is just a 3D to 2D transformation via the use of perspective projection. Again, the problems are manifested in the addition of the camera distortion model. In this approach, since the image coordinates are known, a different procedure is applied in which the image coordinates are inversely corrected. This requires an iterative operation, where an approximation of the distorted point is improved on every iteration. In each iteration, the approximated distorted point is corrected and evaluated against the initial coordinates. The iterative process finishes when the corrected distorted point actually coincides with the given error-free point, or when it is considered to be close enough depending upon a threshold. The default threshold value was set to $\frac{1}{1000}$ of the set pixel size. This procedure is really efficient, as thousands of points can be processed in milliseconds.

With this type of process, it is possible to generate image coordinates of high precision. Experimentation with this simulation approach indicated that there was no error in the XYZ coordinates, while the interior orientation parameters were recovered with almost absolute decimal precision when the created networks where processed in Australis. The ability to add random perturbations in the image coordinates was then subsequently implemented, and proved a very useful addition to the simulation software.
Long Focal Length Camera Calibration using the Central Perspective Model

6.1 Introduction

In applying network orientation with self-calibration to images with very narrow fields of view, problems can arise through overparameterization, ill-conditioning and subsequent numerical instability in the normal equations of the bundle adjustment. The application limits of the current collinearity model are discussed in Section 6.2.

In seeking to overcome problems encountered in the self-calibration of cameras with narrow fields of view, two prospective approaches are considered in this chapter. The first is to investigate means to better accommodate numerical ill-conditioning with the help of the matrix decomposition algorithms discussed in Chapter 3. This investigation and the experimental testing performed on the numerical stability of the collinearity equations is reported in Section 6.4. The
second option, presented in Section 6.6, is to look to the formulation of the functional model. Finally, Section 6.7 presents an evaluation of the proposed functional model modifications, with various test cases.

## 6.2 Problem Identification

In order to evaluate the limits of the applicability of the collinearity equation to cases of very narrow fields of view, initial tests were carried out using mostly simulated data. The use of simulated data was preferable at this stage as it was possible to simulate cameras with different focal lengths, while retaining more control over the complete photogrammetric orientation process. The adopted simulated networks had a typical imaging geometry that is employed in an engineering or industrial photogrammetric survey. Coded and normal targets were used as object feature points, to facilitate automatic network orientation and self-calibration. A number of networks were examined, each representing use of a camera with different focal length and consequently field of view. The different networks were recorded with a principal distance interval of 10\(mm\), starting with a focal length of 100mm (FoV 13.5°) and increasing to 400mm (FoV 3.4°). The network characteristics for a subset of the examined networks are listed in 6.1.

<table>
<thead>
<tr>
<th>Network</th>
<th>Focal Length (mm)</th>
<th>FoV</th>
<th>Number of stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.0</td>
<td>13.5</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>150.0</td>
<td>9.0</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>200.0</td>
<td>6.7</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>250.0</td>
<td>5.4</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>300.0</td>
<td>4.5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>350.0</td>
<td>3.9</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>400.0</td>
<td>3.4</td>
<td>15</td>
</tr>
</tbody>
</table>

The simulated 15-image networks were then solved using the Australis photogrammetric software via a self-calibration bundle adjustment. As explained in Chapter 5, the developed simulation software provided fine-scale control over
the camera interior and exterior orientation, and the object points coordinates. Thus, the results were able to be compared directly to the values that were set in the simulation environment. The principal aim was to identify the properties and the application limits of the collinearity equations for long focal length lenses. Even though a firm figure for the minimum allowable FoV value cannot be given, special attention is typically required when it falls below $10^\circ$. Such numerical issues were not present when the the bundle adjustment was employed without self-calibration, while in some datasets, focal length calibration was feasible.

### 6.3 Calibration Model

The experimental testing of networks with imagery having narrow FoV verified that the problems of long focal length lenses arise mainly due to the inclusion of the interior orientation parameters as unknowns for lenses with a field of view of less than approximately $10^\circ$. In Section 2.3, a detailed analysis of the introduced errors was presented and the magnitude of each error source was discussed. For long focal length lenses, a careful selection of calibration parameters has to be performed, especially as the correlation between interior and exterior orientation parameters increases with increasing focal length. The well-known 8-parameter ‘physical’ camera calibration model developed by Brown (1971) has been found to be near universally applicable in close-range photogrammetry. The photogrammetric properties of long focal length lenses are well recognised and have previously been noted by Fraser & Al-Ajlouni (2006), Fryer & Fraser (1986), Läbe & Förstner (2004), Noma et al. (2002) and Wiley & Wong (1995).

The calculation of principal distance and principal point coordinates is of equal importance for long as for short focal length lenses, in spite of the opportunities for projective compensation in the photogrammetric orientation of narrow field of view imagery. Also, the radial distortion is metrically very significant and needs to be taken into account for any photogrammetric application. As mentioned previously, radial distortion is universally modelled via the well-known odd-ordered polynomial expression comprising terms to seventh order. However, for zoom lenses, the third-order coefficient $K_1$ is usually sufficient to describe the radial distortion profile. The maximum radial distortion occurs at the minimum
zoom focal length and for lenses exhibiting only barrel distortion it decreases as the zoom focal length increases. For many zoom lenses there will be a zero crossing between the barrel distortion at short focal lengths and the pincushion distortion at long focal lengths (Fraser & Al-Ajlouni, 2006). As discussed in Section 2.3.3, the decentring parameters $P_1, P_2$ can be ignored, due to their minimal impact in metric accuracy, as well as their high correlation with the principal point offset, which can give rise to singularities when long focal length lenses are employed.

The image coordinate correction model adopted for self-calibration of long focal length lens can then comprise the quite familiar 4-parameter subset of Brown’s model:

$$
x_{\text{corr}} = x - x_p + (x - x_p)K_1 r^2 - \frac{x}{c} dc
$$

$$
y_{\text{corr}} = y - y_p + (y - y_p)K_1 r^2 - \frac{y}{c} dc
$$

(6.1)

where $c$ is the principal distance and $r$ the radial distance, with

$$
r = \sqrt{(x - x_p)^2 + (y - y_p)^2}
$$

(6.2)

6.4 Numerical Stability

As mentioned in Section 3.3, there are algorithms that can help in solving weakly conditioned or close-to-singular linear systems. As a component of the reported investigation, an analysis of selected numerical techniques applicable to poorly conditioned or near-singular linear models has been performed. A range of triangular and orthogonal factorisations were tested and their numerical properties and stability evaluated. It was found that the characteristics and simplicity of the SVD make it a logical choice for application to the self-calibrating bundle adjustment of networks of images with narrow fields of view.

In the next section, the implementation of the SVD in the photogrammetric bundle adjustment is discussed. The purpose is to evaluate the results of attempting to overcome the weak geometric aspects presented when the field of view becomes narrow. In this process, two alternative approaches are considered. The self-calibrating bundle adjustment is initially set up as a
direct solution of the design matrix $A$ of Equation 2.10, since this is the most straightforward implementation and it is more numerically stable than using the normal equations. In the second approach, the SVD is implemented for bundle adjustments utilising normal equations and incorporating both the standard and reverse fold-in techniques. This affords a comparison of solution stability against the direct solution.

### 6.4.1 Implementation of the SVD

Australis is an off-line digital close-range photogrammetric image measurement, orientation, triangulation and sensor calibration system. Since it provides all standard photogrammetric procedures, it is an ideal tool for the implementation and evaluation of a bundle adjustment solution via the SVD. The Australis platform is developed in the C++ programming language, which was also the selected language for this research project. For the implementation and computation of the SVD, two different algorithms were tested, with the first being proposed by Golub & Reinsch (1970) and Wilkinson & Reinsch (1971), and the second being the Jacobi SVD. The decision to implement the second algorithm came as a consequence of the investigation carried out by Čepek & Pytel (2005) into the use of the SVD in surveying networks. They concluded that results obtained using the first algorithm were not satisfactory since convergence was sometimes lost in the diagonalization phase for ill-conditioned matrices. In such cases, the SVD failed to give better numerically results compared to other approaches, for example the Cholesky decomposition. Čepek & Pytel (2005) also suggested that the Jacobi method is superior and could possibly provide more accurate singular value results than any of the other methods. The difference between the two algorithms is that the Jacobi SVD uses Givens rotation to calculate the first phase of bi-diagonalization whereas the algorithm proposed by Golub & Reinsch (1970) uses Householder reflections. In order to evaluate the use of the Jacobi algorithm, an implementation by the GNU Scientific Library, known as GSL, was used.
6.4 Numerical Stability

6.4.2 Solution by Direct Decomposition of the Design Matrix \( A \)

Upon linearization of the collinearity equations to form the design or configuration matrix \( A \) of the observation equations for the bundle adjustment, partial derivatives with respect to the interior, exterior and object point parameters forming Equation 2.10 are calculated. The solution of the observation equations, which take the form \( Ax = b \), is the most straightforward case versus a normal equation solution with or without fold-in applied, since no post-processing is required after the calculation of the partial derivatives. Consequently, the rows of the created linear system are equal to the rows of the design matrix \( A \). This results in high computation load, especially in large photogrammetric networks, due to the amount of calculations required to acquire the SVD of such a large sparse matrix.

However, the acquisition of the solution vector of this linear system is straightforward. After the calculation of the SVD of matrix \( A \) in the system \( Ax = b \), the smallest 7 values are zeroed in order to acquire the pseudo-inverse, to overcome the rank defect in the network. The correction vector for all the unknowns can be calculated with the algorithm provided in Equation 3.66. The cofactor matrix \( Q \) can be computed without the inversion of \( A \), by using the matrices from the SVD decomposition:

\[
Q = (A^T A)^{-1} = V S^{-2} V^T
\]  

(6.3)

6.4.3 Solution via Normal Equations

Even though the use of the normal equations might not be considered good practise from a numerical point of view, it offers certain advantages. The size of the normal equation matrices does not depend on the number of observations, but only on the number of cameras, images and object points in the network. For \( l \) cameras, \( m \) images and \( n \) object points the row and column size of the normal equations is \( 4l + 6m + 3n \), whereas \( A \) will have \( 2mn \) rows if all points are measured in all images. In the absence of parameter weighting, the normal
6.4 Numerical Stability

Equations are written as

\[
\begin{bmatrix}
N_{11} & N_{12} & N_{13} \\
N_{12}^T & N_{22} & N_{23} \\
N_{13}^T & N_{23}^T & N_{33}
\end{bmatrix}
\begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix} =
\begin{bmatrix}
c_1 \\
c_2 \\
c_3
\end{bmatrix}
\]  

(6.4)

where \( \hat{x}_1 \) is a \( 4l \) vector of corrections to the interior orientation parameters, \( \hat{x}_2 \) is an \( 6m \) vector of corrections to elements of exterior orientation and \( \hat{x}_3 \) is a \( 3n \) vector of corrections to the coordinates of object points. The matrices \( N_{ij} \) and vectors \( c_{ij} \) represent the normal equation contributions from the image coordinate observations and initial parameter values.

The normal equation matrix \( N \) created in the photogrammetric bundle adjustment has a specific pattern, illustrated in Figure 6.1. The solution of such a system can be simplified in cases where the sparseness pattern is taken into account. There are two solution approaches for photogrammetric block adjustment. These take advantage of the special structure of the normal equations. The first is forward, or standard fold-in, proposed and described in Brown (1976a), and the second is reverse fold-in (Fraser & Legac, 1994). In the next sections, two different ways of restructuring the standard and reverse fold-in
are presented. These two modified fold-in approaches are required for the proper
determination of the parameters via an SVD solution. As will be presented later,
the main difference between the modified and the original algorithms is that
the interior orientation matrix \( N_{11} \) has to be eliminated along with the chosen
eliminated parameters. Otherwise, the numerical stability of the linear system
can be adversely affected.

### 6.4.3.1 Modified Standard Fold-in

The modified standard fold-in is similar to the standard fold-in with the only
difference being that the camera parameters are also eliminated, along with the
object point parameters. By restructuring Equation 6.4 the following is obtained:

\[
\begin{bmatrix}
N_{11} & N_{13} & N_{12} \\
N_{13}^T & N_{33} & N_{23}^T \\
N_{12}^T & N_{23} & N_{22}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_3 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
c_1 \\
c_3 \\
c_2
\end{bmatrix}
\] (6.5)

Then the above system can be further simplified to

\[
\begin{bmatrix}
\hat{N} & \hat{N} \\
\hat{N}^T & \hat{N}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_3
\end{bmatrix} =
\begin{bmatrix}
c_1 \\
c_3
\end{bmatrix}
\] (6.6)

The elimination of \( \hat{x} \) from Equation 6.6 yields the reduced normal equations
system

\[
(\hat{N} - \hat{N}^T\hat{N}^{-1}\hat{N})\hat{x} = (\hat{c} - \hat{N}^T\hat{N}^{-1}\hat{c})
\] (6.7)

If

\[
P = \hat{N} - \hat{N}^T\hat{N}^{-1}\hat{N}
\] (6.8)

and

\[
t = \hat{c} - \hat{N}^T\hat{N}^{-1}\hat{c}
\] (6.9)

then the linear system \( P\hat{x} = t \) can be solved by using the SVD in a similar way
to that described in Section 6.4.2. Finally, utilising \( \hat{x} \), the correction vector of
the object point coordinates is calculated by

\[
\hat{x} = \hat{N}^{-1}(\hat{c} - \hat{N}\hat{x})
\] (6.10)
6.4 Numerical Stability

In order to calculate the covariance matrix of the camera station parameters, the decomposed matrices from the SVD can be used. The inverse, or in this case the generalised inverse, of matrix \( S \) can be calculated as

\[
\hat{Q}_x = P^{-1} = VS^{-1}U^T
\]  
(6.11)

However, the matrix \( P \) of Equation 6.11 and consequently the inverse \( S^{-1} \) of the singular values are not of full rank, due to the lack of datum definition. Thus, in order to properly calculate the covariance matrix of the camera station parameters, the last 7 elements of the diagonal matrix \( S^{-1} \) have to be set explicitly to zero. The covariance matrix of the camera parameters and object point coordinates is calculated by

\[
\hat{Q}_x = \hat{N}^{-1}(I + \hat{N}\hat{Q}_x\hat{N}^T\hat{N}^{-1})
\]  
(6.12)

6.4.3.2 Modified Reverse Fold-in

In the modified reverse fold-in solution the camera and image parameters are eliminated. By partitioning Equation 6.4 as

\[
\begin{bmatrix}
N_{11} & N_{12} & N_{13} \\
N_{12}^T & N_{22} & N_{23} \\
N_{13}^T & N_{23}^T & N_{33}
\end{bmatrix}
\begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix}
= 
\begin{bmatrix}
c_1 \\
c_2 \\
c_3
\end{bmatrix}
\]  
(6.13)

Equation 6.13 can be simplified to

\[
\begin{bmatrix}
\hat{N} & \hat{N} \\
\hat{N}^T & \hat{N}
\end{bmatrix}
\begin{bmatrix}
\hat{x} \\
\hat{x}
\end{bmatrix}
= 
\begin{bmatrix}
\hat{c} \\
\hat{c}
\end{bmatrix}
\]  
(6.14)

With the elimination of \( \hat{x} \) from Equation 6.14, the reduced normal equation system is obtained:

\[
(\hat{N} - \hat{N}^T\hat{N}^{-1}\hat{N})\hat{x} = (\hat{c} - \hat{N}^T\hat{N}^{-1}\hat{c})
\]  
(6.15)

By setting

\[
P = \hat{N} - \hat{N}^T\hat{N}^{-1}\hat{N}
\]  
(6.16)
and
\[ t = \ddot{c} - \bar{N}^T \dot{\bar{N}}^{-1} \dot{c} \]  
(6.17)

the correction vector of the object points \( \ddot{x} \) can be obtained again by using the SVD as explained in 6.4.2. Then, the correction vector \( \dot{x} \) of the camera and image parameters is calculated by
\[ \dot{x} = \dot{\bar{N}}^{-1}(\dot{c} - \bar{N} \ddot{x}) \]  
(6.18)

Similar to the modified standard fold-in process, the covariance matrices are
\[ \dot{Q}_x = P^{-1} = VS^{-1}U^T \]  
(6.19)
\[ \dot{Q}_x = \dot{N}^{-1}(I + N\dot{Q}_x N^T \dot{N}^{-1}) \]  
(6.20)

### 6.5 Testing Regime

In order to ensure the robustness of the SVD algorithm, various tests with existing multi-image close-range photogrammetric data sets were performed. In the case of wide angle lenses, the calibration results should be the same as those obtained via a ‘normal’ bundle adjustment, and this was found to be so for all the cases examined. Differences in the estimated precision of the object points arose in some cases; however Fraser (1982) has shown that use of the pseudo-inverse does not lead to a minimal a-posteriori variance of object point coordinates and thus the different standard error values were to be expected.

It is noteworthy that, in the case of the normal equations, incorrect results were obtained whenever the camera interior orientation parameters were not eliminated along with the object point parameters or the camera station parameters, depending on the fold-in used. This was probably due to numerical issues related to the calculation of the normal equations. This led to minor modifications to the standard and reverse fold-in algorithms that were discussed in Sections 6.4.3.1 and 6.4.3.2. This problem did not arise in the solution via direct factorization of the design matrix \( A \) presented in Section 6.4.2.

In the next stage of testing, unsuccessful and unsatisfactory self-calibrations...
6.5 Testing Regime

that were performed in Section 6.2, were re-evaluated via application of the SVD in the expectation of improved numerical behaviour. A numerical analysis of the linear system was performed for these cases, which showed neither abnormalities nor any linear or near-linear dependencies between the parameters compared to the wide angle datasets. While improved results were anticipated through use of the pseudo-inverse by means of the SVD, the results obtained were instead the same for all practical purposes, to those achieved using a regular bundle adjustment employing Cholesky decomposition of the normal equations.

In order to quantify the numerical stability of the regular bundle adjustment solution for weak-geometry networks, the direct solutions with the SVD via decomposition of the design matrix \( A \) and via the normal equations were computed. By providing poor initial approximates for interior orientation, these networks were made close to ill-conditioned due to high correlation between the interior and exterior orientation parameters. The performed tests showed that the SVD approach is not as sensitive to ill-conditioning when compared to the normal solution via a Cholesky decomposition, for example. Considering the direct factorisation of \( A \) and normal equations in the SVD, however, no differences were observed. The normal bundle adjustment solution would either fail to provide a correct result, or provide the correct result in more iterations than the SVD.

As already mentioned, the above tests were also performed using the Jacobi implementation of the SVD. In order to check the numerical properties of different algorithms, numerical analysts often use a condition that is known as backward stability (Trefethen & Bau, 1997). In this condition, an algorithm, or in this case a factorisation, is merged back to the initial argument and the result is evaluated against the initial value. For example, for the case of the SVD where \( A = U \Sigma V^T \), the matrix \( A \) is re-calculated from the decomposed matrices and then compared to its original form. As long as the computation reveals parameter values close to the initial values within a set threshold of the computational accuracy \( O(\epsilon_{\text{machine}}) \), the algorithm is considered backwards stable. This stability test was used to examine the numerical properties of the two different SVD implementations, the Jacobi and that proposed by Golub & Reinsch (1970). The Jacobi algorithm presented more accurate results compared to the Golub & Reinsch (1970) method, though
the differences were not significant. Despite this, the occurrence of convergence 
loss that is presented in the diagonalization phase mentioned by Čepek & Pytel 
(2005) has not been encountered in any of the performed experiments. On the 
contrary, in many cases, the orthogonal matrix $U$ calculated by the Jacobi SVD 
showed a loss of orthogonality. It is noteworthy that the singular values produced 
by the two SVD decompositions were quite different.

The testing carried out indicated conclusively that the SVD is a more 
stable approach compared to the traditional solution approach, via Cholesky 
for example. However, this conclusion cannot necessarily be extended to 
networks of real images as it was only drawn on the basis of an experiment 
with specifically simulated data that highlighted the numerical properties of 
the different algorithms. Significant insight was indeed gained into the self-
calibration of narrow-field-of-view cameras, especially in regard to the fact that 
the application of numerical tools such as orthogonal decomposition does not 
necessarily enhance the recovery of calibration parameters, and so practical limits 
to focal length still apply. Attention needed instead to be turned to the second 
option mentioned in Section 6.1, namely the formulation of the functional model.

### 6.6 Partial Derivatives

Upon linearization of the collinearity equations to form the configuration matrix 
$A$ of the observation equations for bundle adjustment, partial derivatives with 
respect to the unknown calibration parameters forming Equation 6.1 were 
determined. Traditionally, this has yielded coefficients of $-1$ for the parameters 
$x_p$ and $y_p$, resulting in a model that has served the photogrammetric community 
well for 40-odd years. However, given the impact of even the smallest inaccuracies 
in the ill-conditioned and consequently unstable equation system for the self-
calibrating bundle adjustment that can arise when cameras with very long focal 
lengths are involved, it behoves us to look again at the determination of partial 
derivatives of the image correction model, especially terms for the principal point 
coordinates.

As is apparent from Equation 6.1, the correct terms related to parameters $x_p$
and \( y_p \) in the matrix \( A \) of partial derivatives are not the commonly employed

\[
\begin{bmatrix}
  c & x_p & y_p & K_1 \\
  \partial x (\frac{U}{W}) & 1 & 0 & (x - x_p)r^2 \\
  \partial y (\frac{V}{W}) & 0 & -1 & (y - y_p)r^2
\end{bmatrix}
\]

but instead

\[
\begin{bmatrix}
  c & x_p & y_p & K_1 \\
  \partial x (\frac{U}{W}) & 1 & 0 & -K_1r^2 - 2K_1 \bar{x}^2 \\
  \partial y (\frac{V}{W}) & -2K_1 \bar{x} \bar{y} & -1 & -K_1r^2 - 2K_1 \bar{y}^2
\end{bmatrix}
\]

where

\[
U = r_{11}(X - X_o) + r_{12}(Y - Y_o) + r_{13}(Z - Z_o) \\
V = r_{21}(X - X_o) + r_{22}(Y - Y_o) + r_{23}(Z - Z_o) \\
W = r_{31}(X - X_o) + r_{32}(Y - Y_o) + r_{33}(Z - Z_o)
\]

and

\[
\bar{x} = x - x_p \\
\bar{y} = y - y_p
\]

with \( r_{ij} \) being the elements of the rotation matrix \( R \).

It will be shown that this small expansion or correction to the configuration matrix can greatly enhance the recovery of interior orientation parameters in the self-calibration of cameras with long focal length lenses, even though the magnitude of \( K_1 \) might only be of the order of \( 10^{-5} \). In situations where the correlation between camera parameters is not high, the presence of small errors in the coefficients of the \( A \) matrix might not have a significant impact upon the estimation of interior orientation parameters, and consequently upon exterior orientation and object space coordinates. However, the opposite can be true where there is strong projective coupling, as with narrow field of view imagery.

Even though a 4-parameter correction model has been selected as appro-
appropriate for long focal length lenses, it is always possible to add further lens
distortion parameters. For most lenses, the additional parameters will not
lead to increased accuracy. However, in cases where the additional parameters
\((K_2, K_3, P_1, P_2, B_1, B_2)\) are warranted, the partial derivatives of the principal
point offset will involve additional terms, extending the calculations of the
principal point offset as shown in Equation 6.22.

6.7 Experimental Evaluation

6.7.1 Preliminary Testing

In order to validate the proposed model for cameras with narrow fields of view,
and also to assess its practicability, initial tests were carried out using mostly
simulated data. This stage of the experimental program examined the impact
of the expanded partial derivative terms within a ‘standard’ self-calibration.
Again, the datasets created initially for the testing in Section 6.2 were used and
very encouraging results were obtained. Self-calibrations with the new partial
derivative terms led to successful and correct outcomes across the full range of
focal lengths tested, even down to a field of view of 3.4°.

6.7.2 Self-calibration Networks

The investigations with simulated data revealed promising results and conse-
quently further testing with real imagery was performed. Three test cases will
be presented here, the first two involving the self-calibration of a Nikon D200
digital SLR camera, and the third the calibration of a Nikon D80 camera. Case
1 involved a zoom lens set at 300mm (field of view of 4.5°), Case 2 employed a
lens of 400mm focal length (field of view of 3.4°), and Case 3 used a macro lens
at 135mm focal length (field of view of 10°).

6.7.2.1 Case 1 - Lens of 300mm Focal Length

The 7-station, 21-image convergent geometry of Case 1 is shown in Figure 6.2,
the adopted lens being a Nikon ED AF NIKKOR 70-300mm zoom lens, fixed
at 300mm. Three images per station were recorded, at zero, 90° and −90° roll angles, over a camera to object distance of 70m. The approximate distance between adjacent camera stations was 12m. In both this case and Case 2, coded retro-reflective targets were employed to facilitate fully automatic network exterior orientation and self-calibration. A total of 88 coded and 25 single-dot retro-targets were used in Case 1, with strobe illumination being provided by an external Nikon Speedlight SB 800 flash unit. Three such 21-image networks were recorded and processed.

With the expanded coefficients for principal point coordinates in the configuration matrix, the self-calibration of the three networks was successful and no numerical stability issues were encountered. The calibration results showed high repeatability between the different networks, with the precision attained for the interior orientation parameters being standard errors of around 0.15mm for $c$ and better than 0.01mm for $x_p$ and $y_p$. The overall accuracy of object point coordinate determination from one of the three networks is provided in Table 6.2.

The three networks were also solved with the conventional self-calibration model (coefficients of $-1$ for $x_p$ and $y_p$ in the $A$ matrix), in order to compare the results. The self-calibration solution, however, was unstable, and yielded implausible results for the interior orientation, with estimates of the standard
6.7 Experimental Evaluation

Table 6.2: Standard errors of object point coordinates for Case 1.

<table>
<thead>
<tr>
<th>Precision</th>
<th>(mm)</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sigma_X)</td>
<td>0.09</td>
<td>1:70,000</td>
</tr>
<tr>
<td>(\sigma_Y)</td>
<td>0.06</td>
<td>1:98,000</td>
</tr>
<tr>
<td>(\sigma_Z)</td>
<td>0.21</td>
<td>1:29,000</td>
</tr>
<tr>
<td>Mean Std. Error</td>
<td>0.12</td>
<td>1:51,000</td>
</tr>
<tr>
<td>RMS of (xy) residuals</td>
<td>0.83(\mu)m (0.14 pixel)</td>
<td></td>
</tr>
</tbody>
</table>

error for \(c\), \(x_p\) and \(y_p\) being several mm. The poor determinability and repeatability of the camera interior orientation also adversely impacted upon the accuracy of the computed exterior orientation and object point coordinates.

6.7.2.2 Case 2 - Lens of 400mm Focal Length

The network geometry for Case 2, illustrated in 6.3, was similar to that of Case 1, with distinctions being that the camera-to-object distance was 100m, the overall convergence angle was somewhat less (7m between adjacent stations), and the employed focal length was 400mm, the lens being a Nikon 80-400mm VR zoom lens. For this network, 94 coded and 25 single retro-reflective targets were used. In addition to the primary 21-image network, a second set of 18 images was acquired from six additional stations to independently assess the integrity of the initial self-calibration, and especially to assess the stability of recovery of camera parameters. The camera stations for this second network were positioned between those of the 7-station configuration.

The two self-calibrating bundle adjustments utilising the expanded \(A\)-matrix were again free of numerical stability concerns. The estimates of the interior orientation for the two networks displayed no significant differences, even though the geometry was weaker than in Case 1 and the field of view narrower, at 3.4°. The estimated standard errors were 0.8mm for \(c\) and better than 0.01mm for \(x_p\) and \(y_p\). For this case, the recovery of a stable calibration via the traditional additional parameter model (coefficients of \(-1\) for \(x_p\) and \(y_p\) in the \(A\) matrix) was not possible, which highlights the utility of the new approach.
As a final processing step for Case 2, the 21- and 18-image networks were combined in a 13-station, 39-image bundle adjustment, in order to improve the precision of recovery of both calibration parameters and object point coordinates. The results for this self-calibration adjustment, the camera stations of which are shown in Figure 6.4, are summarized in Table 6.3.

Table 6.3: Standard errors of object point coordinates for the 39-image combined network of Case 2.

<table>
<thead>
<tr>
<th>Precision</th>
<th>(mm)</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_X$</td>
<td>0.12</td>
<td>1:72,000</td>
</tr>
<tr>
<td>$\sigma_Y$</td>
<td>0.30</td>
<td>1:30,000</td>
</tr>
<tr>
<td>$\sigma_Z$</td>
<td>0.54</td>
<td>1:17,000</td>
</tr>
<tr>
<td>Mean Std. Error $\sigma_{XYZ}$</td>
<td>0.32</td>
<td>1:28,000</td>
</tr>
<tr>
<td>RMS of $xy$ residuals</td>
<td>1.3$\mu$m (0.21 pixel)</td>
<td></td>
</tr>
</tbody>
</table>

It is noteworthy that the 80-400mm zoom lens of Case 2 had very pronounced chromatic aberration (Figure 6.5) which degraded image quality and thus also the accuracy of image point centroiding. The adverse impact of chromatic aberration can be seen in the higher RMS value of image coordinate residuals, and
6.7 Experimental Evaluation

Figure 6.4: 13-station, 39-image network configuration for combined Case 2, 400mm focal length.

consequently also in poorer than anticipated precision of object point coordinate determination.

Figure 6.5: Example of a coded target affected by chromatic aberration.

6.7.2.3 Case 3 - Macro Lens of 135mm Focal Length

A network of 27 images of the surface of a 50-cent coin formed Case 3. The images were recorded, three each at nine basic camera station locations, using a Nikon D80 DSLR camera with a Sigma AF 105mm macro lens, from a distance of 40cm. Figure 6.6 illustrates the convergent imaging geometry employed. Since there was no focal length ring on the lens, the principal distance value could only be guessed prior to calibration from the magnification index, which was set to
1:3. The principal distance obtained in the subsequent self-calibration turned out to be 134.8mm (field of view of approximately 10°). Seventeen coded targets comprising half-millimetre diameter black dots on white paper were placed around the object to facilitate automatic exterior orientation and self-calibration. Even though the field of view of the macro lens was more than double that of the zoom lenses employed in Cases 1 and 2, and the network displayed strong geometry, the conventional self-calibration model (coefficients of \(-1\) for \(x_p\) and \(y_p\) in the \(A\) matrix) was unstable and produced erroneous results for interior orientation parameters.

The perspective model approach with modified partial derivative terms, on the other hand, was able to reliably provide a repeatable and correct solution for the camera self-calibration. The resulting precision for the object space coordinates is summarised in Table 6.4.

### 6.8 Concluding Remarks

In its adoption of two possible approaches to enhance the performance of self-calibration of close-range, narrow field of view cameras, this part of the
6.8 Concluding Remarks

<table>
<thead>
<tr>
<th>Precision</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_X$</td>
<td>1.2 1:49,000</td>
</tr>
<tr>
<td>$\sigma_Y$</td>
<td>1.5 1:40,000</td>
</tr>
<tr>
<td>$\sigma_Z$</td>
<td>0.8 1:68,000</td>
</tr>
<tr>
<td>Mean Std. Error $\sigma_{XYZ}$</td>
<td>1.2 1:50,000</td>
</tr>
<tr>
<td>RMS of $xy$ residuals</td>
<td>1.1$\mu$m (0.18 pixel)</td>
</tr>
</tbody>
</table>

Table 6.4: Standard errors of object point coordinates for the 27-image macro lens network.

Investigation has yielded two noteworthy findings. The first is that the SVD can provide enhanced insight into both the numerical conditioning of a bundle adjustment solution and the quantifiable extent of projective coupling between parameters, but it appears not to offer an avenue for improved photogrammetric orientation of long focal length imagery. This finding is useful as it indicates that in order to handle the photogrammetric orientation of very narrow field of view imagery, for example from photography with zoom or telephoto lenses, the focus of attention needs to be upon the formulation of the functional model and not necessarily upon prospects for enhancing the solution of an ill-conditioned set of linear equations. The second is that it has been demonstrated that much of the problem of instability in the solution of the traditional self-calibrating bundle adjustment of narrow-angle photography can be attributed to an incomplete formulation extended collinearity equations and partial derivatives with respect to the coordinates of the principal point.
Long Focal Length Camera Calibration using an Orthogonal Projection Model

7.1 Introduction

In this chapter, alternative functional models for photogrammetric orientation, which are better suited to the ‘difficult’ and invariably weaker geometries encountered in long distances measurements using close-range photogrammetry, are considered. In computer vision, various models that are based on projective geometry have been widely used. However, some of them provide low stability and accuracy due to the algebraic indefiniteness of the linear equation systems involved, even though they can be treated as linear forms Ono et al. (2004). Both the affine (Okamoto, 1993) and orthogonal projection models (Ono & Hattori, 2002) have at times been proposed specifically for cases of terrestrial photogrammetric measurement over long distances. The orthogonal projection model is a rigorous model that can be derived from the central perspective model.
In section 7.2, further insight into the analytical formulation of the orthogonal projection model, and the relationship between the orthogonal projection, central perspective and affine models, is provided. Section 7.3 presents the complete photogrammetric approach that is required to obtain the final object space coordinates. A self-calibrating bundle adjustment is formulated for the orthogonal projection model in order to evaluate and compare the results of various test cases against those from the collinearity equations of Section 6.6. The results of this comparison are presented in Section 7.3.3.

## 7.2 Orthogonal Projection Model

The mathematical formulation of the orthogonal projection model can be presented in two steps. The first is derivation of the more generic affine model and the second is formulation of the orthogonal projection model. This orthogonal projection formulation, which can be cast as a bundle adjustment, is quite rigorous in the sense that it is derived from the central perspective model and it is thus more than a simple empirical formulation. Such a relationship does not exist between the affine model and the perspective model in the absence of additional constraints.

### 7.2.1 Derivation of the Orthogonal Projection Model

Under the central perspective model, the collinearity equation expression commonly used in close-range photogrammetry to describe the projection of an object point into its corresponding image point is given by

\[
\begin{bmatrix}
    x \\
    y \\
    -c
\end{bmatrix} = \lambda \begin{bmatrix}
    r_{11} & r_{12} & r_{13} \\
    r_{21} & r_{22} & r_{23} \\
    r_{31} & r_{32} & r_{33}
\end{bmatrix} \begin{bmatrix}
    X - X_o \\
    Y - Y_o \\
    Z - Z_o
\end{bmatrix}
\]  

(7.1)

where \( \lambda \) is the scale factor, \( c \) the principal distance, \( r_{ij} \) the elements of the rotation matrix and \( X_o, Y_o, Z_o \) the coordinates of the perspective centre. The scale factor \( \lambda \) is an unknown value which varies for each object point. If \( \lambda \) is substituted by
a constant scale parameter $s$, Equation 7.1 can be rewritten as

$$
\frac{s}{\lambda} \begin{bmatrix} x \\ y \\ -c \end{bmatrix} = s \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X - X_o \\ Y - Y_o \\ Z - Z_o \end{bmatrix} \tag{7.2}
$$

and, by transposing $X_o, Y_o, Z_o$ to the left side, Equation 7.2 can be recast as

$$
\begin{bmatrix} x - X'_o \\ y - Y'_o \\ -s \frac{c}{\lambda} - Z'_o \end{bmatrix} = s \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{7.3}
$$

where

$$
\begin{bmatrix} X'_o \\ Y'_o \\ Z'_o \end{bmatrix} = -s \begin{bmatrix} r_{11} & r_{12} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \tag{7.4}
$$

The first and second rows of Equation 7.3 express the affine projection model:

$$
\begin{bmatrix} x \\ y \end{bmatrix} = s \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} X'_o \\ Y'_o \end{bmatrix} \tag{7.5}
$$

This model requires the transformation from central perspective image coordinates to affine coordinates, an operation that will later be discussed. The number of independent parameters is eight and, geometrically, the eight orientation parameters for an affine image are considered to be three image rotations, two translation elements $X'_o, Y'_o$, the image scale $s$ and two rotation parameters describing the relationship between projected rays and the normal to the image plane. By generalising Equation 7.5, the collinearity equation equivalents for the affine projection model are derived as

$$
\begin{align*}
x_a &= A_1 X + A_2 Y + A_3 Z + A_4 \\
y_a &= A_5 X + A_6 Y + A_7 Z + A_8
\end{align*} \tag{7.6}
$$
7.2 Orthogonal Projection Model

The affine projection model allows oblique projection to an image plane. Addition of constraints for orthogonal projection to Equation 7.6 then leads to the orthogonal projection model. Because the generalised coefficients $A_i$ are derived from the components $r_{ij}$ of the rotation matrix and the scale parameter $s$, they should have the following properties of an orthogonal rotation matrix:

- The vectors $\mathbf{a}_x = (A_1, A_2, A_3)$ and $\mathbf{a}_y = (A_5, A_6, A_7)$ must be perpendicular to each other and thus the dot product $a_x \cdot a_y$ has to be zero. This will enforce the rays from the object to be orthogonal to the image plane, such that
  \[ A_1A_5 + A_2A_6 + A_3A_7 = 0 \]  
  (7.7)

- The norms of $a_x$ and $a_y$ must be equal, meaning that the scale in the $x_a$ direction is equivalent to that in the $y_a$ direction, so that
  \[ A_1^2 + A_2^2 + A_3^2 = A_5^2 + A_6^2 + A_7^2 \]  
  (7.8)

The difference between the affine and orthogonal projection models is that the latter allows only perpendicular projection to an image plane. The orthogonal projection model has six independent parameters. The two constraints of Equations 7.7 and 7.8 reduce the degrees of freedom of Equation 7.6 from eight to six. More precisely, the principal distance, and a distance from the projection centre to the object point along the optical axis, are replaced by a uniform scale factor. The distinction between the position of the principal point and the horizontal position of the projection centre with respect to the image plane then becomes meaningless. This situation can be visualised by letting the projection centre approach an infinite position along the optical axis.

7.2.2 Transformation from Central Perspective to Affine Projection Coordinates

It has been recognised (Fraser & Yamakawa, 2004; Hattori et al., 2000; Ono & Hattori, 2002) that for both the affine and orthogonal projection models to be applicable, an initial conversion from a central perspective to an affine image
is warranted. It has been stated that in very narrow fields of view, this can be avoided since the subsequent accuracy loss will be minimal. The image coordinate transformation is the same for both the affine and the orthogonal projection models.

Although mathematically any value can be assigned to the scale factor $s$, the application of an orthogonal projection to a frame camera becomes more realistic when the scale factor is calculated in a specific way. From a practical point of view, $s$ is adjusted so as to scale down the average photographic distance to the same length as the principal distance, as per Figure 7.1. If $H$ is the average photographic distance in the $Z$ direction, ie $H = Z - Z_o$, then $s$ can be calculated as follows:

$$s = -\frac{r_{33}c}{Z - Z_o} = -\frac{r_{33}c}{H}$$

(7.9)

Equation 7.1 can be recast as

$$\begin{bmatrix} X - X_o \\ Y - Y_o \\ Z - Z_o \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{21} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ -c \end{bmatrix}$$

(7.10)
7.2 Orthogonal Projection Model

and the third row of Equation of 7.10 can then be used to calculate \( \lambda \) as

\[
\lambda = \frac{r_{13}x + r_{23}y - r_{33}c}{Z - Z_o} \quad (7.11)
\]

By substituting Equations 7.9 and 7.11 into 7.2, the expressions for transforming from central perspective image coordinates to affine coordinates are obtained:

\[
x_a = \frac{Z - Z_o}{H} \left( \frac{r_{33}c}{r_{33}c - r_{13}x - r_{23}y} \right) x
\]

\[
y_a = \frac{Z - Z_o}{H} \left( \frac{r_{33}c}{r_{33}c - r_{13}x - r_{23}y} \right) y
\]

Additionally, a calibration model can be incorporated into the transformation. Again, a basic 4-parameter calibration model, shown in Equation 6.1, is selected. Thus Equation 7.12 is recast to

\[
x_a = \frac{Z - Z_o}{H} \left( \frac{r_{33}c}{r_{33}c - r_{13}x_{corr} - r_{23}y_{corr}} \right) x_{corr}
\]

\[
y_a = \frac{Z - Z_o}{H} \left( \frac{r_{33}c}{r_{33}c - r_{13}x_{corr} - r_{23}y_{corr}} \right) y_{corr}
\]

7.2.3 Relationship between Orientation Parameters of the Central Perspective and Orthogonal Projection Models

The orthogonal projection model is a rigorous model that derives from the central perspective model. Therefore, it is possible to determine the relationship of the exterior orientation parameters between the two models. From the definition of the orthogonal projection model it is known that

\[
\begin{bmatrix}
a_1 & a_2 & a_3 \\
a_5 & a_6 & a_7 \\
\end{bmatrix} = s \begin{bmatrix}
r_{11} & r_{12} & r_{13} \\
r_{21} & r_{22} & r_{23} \\
\end{bmatrix}
\]

Since the norm of each of the vectors \( \vec{r}_1 = (r_{11}, r_{12}, r_{13}) \) and \( \vec{r}_2 = (r_{21}, r_{22}, r_{23}) \) is equal to 1, then

\[
s^2 = A_1^2 + A_2^2 + A_3^2
\]

126
7.2 Orthogonal Projection Model

From Equations 7.8 and 7.14 the elements of \( \vec{r}_1 \) and \( \vec{r}_2 \) are easily determined. The vector \( \vec{r}_3 = (r_{31}, r_{32}, r_{33}) \) can be estimated by considering the geometric features of the rotation matrix:

\[
    r_{11}^2 + r_{21}^2 + r_{31}^2 = 1 \quad (7.16)
\]

so that

\[
    r_{31} = \pm \sqrt{1 - r_{11}^2 - r_{21}^2} \quad (7.17)
\]

and

\[
    r_{32} = \pm \sqrt{1 - r_{12}^2 - r_{22}^2} \quad (7.18)
\]

\[
    r_{33} = \pm \sqrt{1 - r_{13}^2 - r_{23}^2} \quad (7.19)
\]

Additionally, from the properties of the rotation matrix:

\[
    r_{11}r_{31} + r_{12}r_{32} + r_{13}r_{33} = 0 \quad (7.20)
\]

\[
    r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33} = 0 \quad (7.21)
\]

If \( c \) is given, \( Z_o \) can be calculated from Equation 7.9 as

\[
    Z_o = \frac{r_{33}c}{s + Z} \quad (7.22)
\]

The following expression is then obtained from the general formula of the affine model and via Equation 7.4:

\[
    \begin{bmatrix}
        A_4 \\
        A_8
    \end{bmatrix} = - \begin{bmatrix}
        A_1 & A_2 & A_3 \\
        A_5 & A_6 & A_7
    \end{bmatrix} \begin{bmatrix}
        X_o \\
        Y_o \\
        Z_o
    \end{bmatrix} \quad (7.23)
\]

If the above equations are solved for \( X_o, Y_o \), then the location of the perspective centre is obtained as

\[
    Y_o = \frac{A_1A_8 - A_4A_5 - A_3A_5Z_o + A_1A_7Z_o}{A_2A_5 - A_1A_6} \quad (7.24)
\]
7.3 Photogrammetric Orientation Procedure

\[
X_o = -\frac{A_1 + A_2 Y_o + A_3 Z_o}{A_1}
\]  

(7.25)

In cases where the parameters of the central perspective model are known, calculation of the \(A_i\) parameters of the orthogonal projection model using Equations 7.9 and 7.14 is quite straightforward. Equation 7.9, however, requires some prior knowledge of the object points coordinates. In the case that this calculation needs to be performed at the initial stage of the bundle adjustment, approximate values are required. The determination of these approximate object point coordinates will be explained in the following section.

7.3 Photogrammetric Orientation Procedure

Based on the algorithms presented, a new and completely automated photogrammetric orientation procedure has been developed. Various steps are involved in the process of calculating initial approximate values, as well as at every iteration of the bundle adjustment. The procedure is summarised in the flowchart of Figure 7.2.

This process for bundle adjustment based on the orthogonal projection model is distinct from previously reported methods, which have tended to concentrate on experimental verification of the appropriateness of the model. For example, in the approach of Ono & Hattori (2002), initial approximate values for the object point coordinates needed to have been measured by non-photogrammetric means, namely ground survey via a total-station. Perturbed values of these coordinates were then used to calculate approximate values of the orthogonal projection orientation parameters via a direct linear transformation (DLT), which itself is prone to instability and even numerical singularity when employed for computing exterior orientation of long focal length lenses.

The procedure developed here is more closely related to the typical close-range photogrammetric methodology, where exterior orientation is carried out using imagery alone, without the provision of any externally measured object point coordinate information. First, a relative orientation is performed between selected images using the conventional, perspective model based coplanarity equations (Cronk et al., 2006; Luhmann et al., 2006b). Approximate exterior orientation
7.3 Photogrammetric Orientation Procedure

Figure 7.2: Flowchart for automated orientation procedure based on the orthogonal projection model.

parameters \((X_0, Y_0, Z_0, \omega, \varphi, \kappa)\) and object points coordinates \(X, Y, Z\) are then obtained for remaining images in the network via resection and spatial intersection. A more detailed explanation of the orientation procedures used is offered in Section 2.4. Additionally, a least squares estimation involving all the
object points takes place in order to acquire a more precise solution and to allow for outlier detection within the image measurement data.

The next stage involves the calculation of orientation parameters of the orthogonal projection model. In this process, the central perspective exterior orientation values are transformed to their equivalent orthogonal projection values. This is a two-step procedure, where the first step involves the calculation of the affine orientation parameters since the orthogonal projection constraints cannot be ensured initially. There are two alternative ways to acquire the exterior orientation parameters of the affine models. The first involves the calculation of the scale parameter $s$ from Equation 7.9 by using a random 3D point and the elements of the rotation matrix $R$. Then, it is possible to calculate the affine exterior orientation parameters by using Equations 7.14 and 7.23. In the second option, the affine image coordinates are first computed from Equation 7.13, after which there is a determination of the parameters of the affine projection model with the use of Equation 7.6 by treating the orientation parameters as unknowns.

The optimal approach for the recovery of the affine orientation parameters is with the second approach, as it involves a least squares adjustment of all the points of an image for the calculation of the unknown parameters $A_i$. A subsequent least squares adjustment with the orthogonal projection constraints, Equations 7.7 and 7.8, is then performed in order to acquire the orientation parameters of the orthogonal projection model.

Once the orthogonal orientation parameters are known, the equivalent central perspective parameters can be updated in order to reflect the current values of the orthogonal projection model. This step is significant mostly for the calculation of both $Z_o$ and the rotation elements that are involved in the transformation of the central perspective coordinates to their corresponding affine values. Despite the fact that the rest of the exterior orientation parameters are not used in the orthogonal projection model, they are nevertheless needed in order to visualise the results, since it is not possible to conceptualise an orthogonal space as Euclidean subspace. Additionally, a final refinement in the transformation of the central perspective to affine coordinates is performed, once all the parameters are known.

Even though the orthogonal projection model is quite similar to the central perspective model, as has been illustrated by Ono & Hattori (2002), some
differences can be expected due to the adjustment procedures involved. Thus, a spatial intersection process using the orthogonal projection model is carried out in order to refine the object point coordinates. The authors’ experience is that by doing so, the subsequent bundle adjustment converges in fewer iterations.

Two bundle adjustment cases are now considered, the first being the simpler case without self-calibration, and the second being where the calibration parameters forming Equation 6.1 are included.

### 7.3.1 Bundle Adjustment without Self-Calibration

In the simpler case of the orthogonal projection model that is formulated as shown in Equation 7.6, the form of the normal equations does not differ from that of the perspective model. The linearised form of the model can be expressed as

\[
A_2x_2 + A_3x_3 = b \tag{7.26}
\]

where \(x_2\) represents the sensor exterior orientation parameters and \(x_3\) the object point coordinates. The \(A_1\) matrices are the corresponding configuration matrices and \(b\) is the image coordinate discrepancy vector. The corresponding normal equations for a network of \(m\) photos containing \(n\) measured points in object space then follow as

\[
\begin{bmatrix}
N_{22} & N_{23} \\
N_{23}^T & N_{33}
\end{bmatrix}
\begin{bmatrix}
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix}
= 
\begin{bmatrix}
c_2 \\
c_3
\end{bmatrix} \tag{7.27}
\]

where \(N_{ij}\) and \(c_i\) represent contributions arising solely from image coordinate observations, \(\hat{x}_2\) is an \(8m\) vector of corrections to elements of exterior orientation \((A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8)\), and \(\hat{x}_3\) is a \(3n\) vector of corrections to the coordinates of object points \((X, Y, Z)\). For the calculation of the elements of matrices \(N_{ij}\), the partial derivatives of Equation 7.6 with respect to the unknown parameters are computed with the resulting configuration matrix being

\[
A = \begin{bmatrix} 
A_2 & A_3 
\end{bmatrix} = 
\begin{bmatrix}
X & Y & Z & 1 & 0 & 0 & 0 & 0 & A_1 & A_2 & A_3 \\
0 & 0 & 0 & 0 & X & Y & Z & 1 & A_5 & A_6 & A_7
\end{bmatrix} \tag{7.28}
\]
7.3 Photogrammetric Orientation Procedure

The two constraints of Equations 7.7 and 7.8 have to be accounted for with the orthogonal projection model. This can be achieved by either bordering the normal equation matrix of Equation 7.27 and solving the least squares adjustment in a two-step algorithm or, preferably, by adding the constraints to the current normal equations:

\[
\begin{bmatrix}
N_{22} + H^T w^T H & N_{23} \\
N_{23}^T & N_{33}
\end{bmatrix}
\begin{bmatrix}
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix}
= \begin{bmatrix}
c_2 - H^T w d \\
c_3
\end{bmatrix}
\] (7.29)

Here, \(H\) is the \(2m \times 8m\) matrix of additional constraints, \(d\) the corresponding \(2m\) discrepancy vector and \(w\) the \(2m \times 2m\) matrix of weights assigned to the constraints. The matrix \(H\) is formed as

\[
\begin{bmatrix}
2A_1 & 2A_2 & 2A_3 & 0 & 2A_5 & 2A_6 & 2A_7 & 0 \\
A_5 & A_6 & A_7 & 0 & A_1 & A_2 & A_3 & 0
\end{bmatrix}
\] (7.30)

In the general form presented by Equation 7.29, the normal equations will be rank deficient, since an explicit definition of the object space coordinate datum has not been made, ie no control points have been employed. In the case of affine and orthogonal projection models, the rank deficiency is 12 (Okamoto, 1993). In order to specify the datum, additional constraints have to be introduced. Although there are a number of ways to impose the required minimal constraints, the most advantageous approach is generally considered to be the adoption of inner constraints (eg Fraser 1982, 1984) where the 12 linearly independent vectors of the inner constraint matrix \(G\) satisfy the relationship \(A G = 0\). The inner constraint matrix can be conveniently partitioned into two components, \(G_2\) relating to the exterior orientation parameters and \(G_3\) relating to the object space coordinates,
7.3 Photogrammetric Orientation Procedure

with the elements for each being

$$G_2^T = \begin{bmatrix}
    A_1 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & A_3 & 0 & 0 & 0 \\
    0 & A_1 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & A_3 & 0 \\
    0 & 0 & A_1 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & A_3 \\
    0 & 0 & 0 & A_1 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 \\
    A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & A_7 & 0 & 0 \\
    0 & A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & A_7 & 0 \\
    0 & 0 & A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & A_7 \\
    0 & 0 & 0 & A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 \\
    A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & A_7 & 0 & 0 \\
    A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & A_7 & 0 & 0 \\
    A_5 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & A_7 & 0 & 0 \\
\end{bmatrix}$$

and

$$G_3^T = \begin{bmatrix}
    -X & -Y & -Z & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & -X & -Y & -Z & -1 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 0 & 0 & -X & -Y & -Z & -1 \\
\end{bmatrix}$$

(7.31)

Although minimal constraints for both exterior orientation and object point coordinates have been presented for the sake of completeness, only the constraint matrix $G_3$ is applied in the presented approach, in order to achieve minimum mean variance for the object space coordinates (Fraser, 1982). The bordered normal equations then take the following form:

$$\begin{bmatrix}
    N_{22} + H^T w H & N_{23} & 0 \\
    N_{23}^T & N_{33} & G_3 \\
    0 & G_3^T & 0 \\
\end{bmatrix} \begin{bmatrix}
    \hat{x}_2 \\
    \hat{x}_3 \\
    \kappa \\
\end{bmatrix} = \begin{bmatrix}
    c_1 - H^T w d \\
    c_2 \\
    0 \\
\end{bmatrix}$$

(7.32)

If the linear system of Equation 7.32 is recast to

$$\begin{bmatrix}
    N & G \\
    G^T & 0 \\
\end{bmatrix} \begin{bmatrix}
    \hat{x} \\
    \kappa \\
\end{bmatrix} = \begin{bmatrix}
    c \\
    0 \\
\end{bmatrix}$$

(7.33)
7.3 Photogrammetric Orientation Procedure

the solution vector \( \hat{x} \) can easily be calculated as:

\[
R = N + G^T G \tag{7.34}
\]

\[
\hat{x} = R^{-1}c \tag{7.35}
\]

The covariance matrix \( Q_\hat{x} \) of the adjusted elements of exterior orientation and the object point coordinates can be evaluated from

\[
Q_\hat{x} = R^{-1} \tag{7.36}
\]

It should be noted that the standard or the reverse fold-in, along with sequential processing, similarly to the conventional bundle adjustment, is possible. Further information on the conventional bundle adjustment solution can be found in Brown (1976b) and Edmundson (1997).

7.3.2 Bundle Adjustment with Self-Calibration

The complete orthogonal projection model can be found by combining Equations 7.6, 7.13 and 6.1:

\[
\begin{aligned}
\left( Z - Z_o \right) & \left( \begin{array}{c}
\frac{r_{33}c(x - x_p + (x - x_p)K_1r^2)}{H} \\
\frac{r_{33}c - r_{13}(x - x_p + (x - x_p)K_1r^2) - r_{23}(y - y_p + (y - y_p)K_1r^2)}{H}
\end{array} \right) \\
- A_1X - A_2Y - A_3Z - A_4 & = 0
\end{aligned}
\]

\[
\begin{aligned}
\left( Z - Z_o \right) & \left( \begin{array}{c}
\frac{r_{33}c(y - y_p + (y - y_p)K_1r^2)}{H} \\
\frac{r_{33}c - r_{13}(x - x_p + (x - x_p)K_1r^2) - r_{23}(y - y_p + (y - y_p)K_1r^2)}{H}
\end{array} \right) \\
- A_5X - A_6Y - A_7Z - A_8 & = 0
\end{aligned}
\] \tag{7.37}

This non-linear model involves both observations and unknown parameters, which necessitates linearization to the following form for subsequent bundle adjustment:

\[
A_1x_1 + A_2x_2 + A_3x_3 + Bv + w = 0 \tag{7.38}
\]
Here, $A_1$, $A_2$, $A_3$ are configuration matrices for interior and exterior orientation parameters and object point coordinates, respectively; $B$ is the matrix of partial derivatives with respect to the observations; and $w$ is the functional value corresponding to the approximate parameter values and observed image coordinates. The resulting normal equations for the network of $l$ cameras and $m$ photos containing $n$ measured points in object space then follow as

$$
\begin{bmatrix}
N_{11} & N_{12} & N_{13} \\
N_{12}^T & N_{22} & N_{23} \\
N_{13}^T & N_{23}^T & N_{33}
\end{bmatrix}
\begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix} =
\begin{bmatrix}
c_1 \\
c_2 \\
c_3
\end{bmatrix}
$$

(7.39)

where the subscripts 1, 2 and 3 now relate to camera calibration parameters, exterior orientation and object point coordinates. Thus, $\hat{x}_1$ is a $4l$ vector of corrections to elements of interior orientation $(c, x_p, y_p, K_1)$, $\hat{x}_2$ is an $8m$ vector of corrections to elements of exterior orientation $(A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8)$, and $\hat{x}_3$ is the $3n$ vector of corrections to the point coordinates $(X, Y, Z)$.

In order to facilitate a more straightforward calculation of partial derivatives in the linearization of Equation 7.37, the rotation angles $r_{ij}$ can be expressed in terms of the orthogonal orientation parameters and considered as constants along with the term $\frac{Z - Z_0}{H}$. Treating the rotation angles as constants is not considered an issue as their values are improved iteratively. The case where the rotation angles are included in the model as unknowns was formulated and tested with no success, possibly due to over-parameterization. The linearization of the principal point offset should use the complete form of the correction model (Equation 6.1) as explained earlier, in order to obtain proper correction values. By naming the two expressions of Equation 7.37 $f_x$ and $f_y$, the configuration matrix $A$ can be presented in terms of its component matrices $A_i$ as follows:

$$
A = \begin{bmatrix}
A_1 & A_2 & A_3
\end{bmatrix}
$$

$$
= \begin{bmatrix}
\frac{\partial f_x}{\partial c} & \frac{\partial f_x}{\partial x_p} & \frac{\partial f_x}{\partial y_p} & \frac{\partial f_x}{\partial K_1} \\
\frac{\partial f_y}{\partial c} & \frac{\partial f_y}{\partial x_p} & \frac{\partial f_y}{\partial y_p} & \frac{\partial f_y}{\partial K_1} \\
-X & -Y & -Z & -1 & 0 & 0 & 0 & -A_1 & -A_2 & -A_3 \\
0 & 0 & 0 & 0 & -X & -Y & -Z & -1 & -A_5 & -A_6 & -A_7
\end{bmatrix}
$$

(7.40)
7.3 Photogrammetric Orientation Procedure

The matrix of partial derivatives with respect to the observations will have the form

\[
B = \begin{bmatrix}
\frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\
\frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y}
\end{bmatrix}
\]  

(7.41)

The expressions for the partial derivative terms of configuration matrices \( A_1 \) and \( B \) are provided in the Appendix A.

In implementing the final bundle adjustment, the addition of the free-net and the orthogonal projection constraints to the normal equations needs to be performed in a two-step algorithm, since the constraints of Equations 7.7 and 7.8 cannot be incorporated in the same way as for the bundle adjustment without additional camera calibration parameters. Instead, double bordering of the normal equation matrix is required to apply the inner and orthogonal projection constraints, the rank defect again being 12:

\[
\begin{bmatrix}
N_{11} & N_{12} & N_{13} & 0 & 0 \\
N_{12}^T & N_{22} & N_{23} & 0 & H \\
N_{13}^T & N_{23}^T & N_{33} & G_3 & 0 \\
0 & 0 & G_3^T & 0 & 0 \\
0 & H^T & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3 \\
\kappa \\
\lambda
\end{bmatrix}
= \begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
0 \\
z
\end{bmatrix}
\]  

(7.42)

This system of normal equations is initially solved without the orthogonal constraints being applied:

\[
\hat{x}_o = -R^{-1}c
\]  

(7.43)

where

\[
R = (N + GG^T)
\]  

(7.44)

Then, the solution with the orthogonal projection constraints applied follows from

\[
S = HR^{-1}H^T
\]  

(7.45)

\[
T = R^{-1}H^TS^{-1}
\]  

(7.46)

\[
\hat{x} = \hat{x}_o + T(z - H\hat{x}_o)
\]  

(7.47)

Finally, the covariance matrix \( Q_x \) of the adjusted parameters can be obtained
7.3 Photogrammetric Orientation Procedure

from

\[ Q_\delta = R^{-1} - THR^{-1} \]

(7.48)

7.3.3 Experimental Results and Discussion

In order to validate and assess the proposed approach using the orthogonal projection model, the same two test cases of Section 6.7.2 involving networks comprising imagery with very narrow fields of view were used. This allowed both a comparison of the results between collinearity and orthogonal projection based models and an assessment of the degree of repeatability in the recovery of camera calibration parameters.

7.3.3.1 Case 1 - Lens of 300mm Focal Length

All adjustment solutions were obtained without any numerical stability issues and the calibrations showed a high repeatability, consistent with the precision of recovery of the camera parameters. Table 7.1 lists the calibration parameters obtained in one of the networks for the collinearity model with expanded coefficients for principal point coordinates in the configuration matrix, and for the orthogonal projection model. It is evident that both models provide essentially identical results.

<table>
<thead>
<tr>
<th>Table 7.1: Camera calibration results for Case 1, 300mm lens.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>( c )</td>
</tr>
<tr>
<td>( x_p )</td>
</tr>
<tr>
<td>( y_p )</td>
</tr>
<tr>
<td>( K_1 )</td>
</tr>
</tbody>
</table>

The overall accuracy of object point coordinate determination for the same network is summarised in Table 7.2, where it can be seen that the two different models yielded comparable results, with the orthogonal projection model producing marginally higher precision in the XY direction.
7.3 Photogrammetric Orientation Procedure

### Table 7.2: Standard errors of object point coordinates for Case 1.

<table>
<thead>
<tr>
<th>Perspective Model</th>
<th>Orthogonal Projection Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mm)</td>
<td>Scale</td>
</tr>
<tr>
<td>$\sigma_x$</td>
<td>0.09</td>
</tr>
<tr>
<td>$\sigma_y$</td>
<td>0.06</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.21</td>
</tr>
</tbody>
</table>

Mean Std. Error $\sigma_{XYZ}$ | 0.12 | 1:51,000 | 0.11 | 1:55,000 |

RMS of $xy$ residuals | 0.83$\mu$m | 0.82$\mu$m |

### 7.3.3.2 Case 2 - Lens of 400mm Focal Length

In regard to accuracy in object space in Case 2, the perspective and orthogonal models yielded significantly different estimates of XYZ coordinate precision. It is apparent from Table 7.3 that the performance of the perspective model has deteriorated with the increase of the camera-to-object distance and the longer focal length. The orthogonal model produced markedly higher precision in XY coordinates (ie non-depth directions), and perhaps as should be anticipated, as its relative performance is enhanced as the field of view narrows.

### Table 7.3: Standard errors of object point coordinates for Case 2.

<table>
<thead>
<tr>
<th>Perspective Model</th>
<th>Orthogonal Projection Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>(mm)</td>
<td>Scale</td>
</tr>
<tr>
<td>$\sigma_x$</td>
<td>0.12</td>
</tr>
<tr>
<td>$\sigma_y$</td>
<td>0.30</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Mean Std. Error $\sigma_{XYZ}$ | 0.32 | 1:28,000 | 0.22 | 1:41,600 |

RMS of $xy$ residuals | 1.3$\mu$m | 1.3$\mu$m |

Whereas numerical issues did not arise in the bundle adjustments via the proposed orthogonal projection models in Cases 1 and 2, instability issues might be anticipated in the cases of very weak geometry (eg the centre three stations of Figure 6.3 only), compounded by poor initial estimates for camera principal distance. In such circumstances, which should be avoided in normal practise, the SVD can be employed instead of inner constraint bordering in order to remove
the datum defect in the normal equations, as explained in Section 6.4.2. Even though the computational cost of the SVD is very high in comparison to inner constraints, the author’s experience is that it is a very robust approach. In investigating alternative solutions it was found that the introduced singularities were often tied to the use of the minimal constraints. The application of inner constraints for both exterior orientation ($G_2$) and object points coordinates ($G_3$) yielded similar results to the SVD. The bordered normals then take the form of

$$
\begin{bmatrix}
N_{11} & N_{12} & N_{13} & 0 & 0 \\
N^T_{12} & N_{22} & N_{23} & G_2 & H \\
N^T_{13} & N^T_{23} & N_{33} & G_3 & 0 \\
0 & G^T_2 & G^T_3 & 0 & 0 \\
0 & H^T & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3 \\
\kappa \\
\lambda
\end{bmatrix}
= 
\begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
0 \\
z
\end{bmatrix}
(7.49)
$$

However, the application of the SVD and the constraints shown in Equation 7.49 do not provide minimum variances for the object point coordinates, $\text{trace}\{N_{33}\} = \text{min}$, which is desirable (Fraser, 1982). In the case of the SVD, a full trace minimisation $\text{trace}\{N\} = \text{min}$ is accomplished, whereas in the case of the inner constraints of Equation 7.49 variances for both the exterior orientation and object point coordinates are minimised, i.e. $\text{trace}\{N_{22}, N_{33}\} = \text{min}$. Nonetheless, after better interior orientation approximates are computed with either of the two approaches, minimal constraints minimisation is possible.

As an alternative, it is possible to use the expanded coefficients of the design matrix $A$, as explained in Section 6.6, to calculate the initial IO parameters. Since there are no significant differences in the calculated IO values between the two different models, even the simpler case of the orthogonal projection model without interior orientation, that does not present such stability issues, can be used.

7.4 Concluding Remarks

The newly developed close-range photogrammetric orientation model using orthogonal projection has been shown to be a viable alternative to the collinearity approach. It is ideally suited to cases where the FoV becomes narrower than, say,
3°–4°. Through real-world scenarios it was proven that the orthogonal projection model can facilitate accurate and reliable recovery of camera parameters via a self-calibrating bundle adjustment, within a computational process that lends itself to full automation. Numerical instabilities arising in weak geometric cases have also been investigated. It has been shown that the SVD can successfully address such cases.
This thesis has presented an account of research undertaken into enhanced high accuracy digital close-range photogrammetry involving cameras with long focal length lenses. The main research goal was to develop automated methods for the self-calibration of such cameras, via the bundle adjustment of multi-image photogrammetric networks. The aims, which have largely been realised, have culminated in the development and implementation of practical algorithms, software routines and methodologies. An investigation into two complementary subjects, the digital image creation pipeline and the photogrammetric simulator, came as a natural consequence of the core research aim.

The integrity of image point position is of great importance for every photogrammetric application. Following an analytical review of digital image creation procedures, it was concluded that it is possible to alter the original image formation process in favour of photogrammetry, as well as take advantage of the higher dynamic range that is offered by the RAW image files. Within the investigation, various alternative demosaicing algorithms have also been
developed and evaluated. The results show that increased precision can be anticipated with the proposed methodology. Future research could adopt different strategies of image formation as well as investigate or develop new demosaicing algorithms.

Practical difficulties arising in the acquisition of images and the need for a more controlled measurement environment for cameras employing long focal length lenses led to the creation of a photogrammetric simulator. The newly developed simulator allows the creation of simulated data in a simple and efficient way, supporting single and coded retro-reflective targets. The ability to create simulated images along with the addition of distortion, due to departures from collinearity, has proven an indispensable utility for this research.

With the use of simulated data, the application limits of focal length and effectively field of view, were identified and it was highlighted that the instability introduced in the least squares adjustment is caused due to the inclusion of the IO calibration model. Three different approaches were then investigated to enhance the stability of long focal length photogrammetric networks, with the first two being applied to the collinearity equations. Initially, least squares algorithms that can significantly improve the numerical instability were evaluated. The application of SVD was tested in photogrammetric networks employing long focal length lenses, but this did not alleviate the initially encountered stability problems. Nonetheless, significant insight was gained through this process, and this led to two noteworthy findings. The first was that the experimental testing performed with simulated images in Chapter 6 as well as with real images in Chapter 7, demonstrated the superior numerical properties of the SVD compared to other approaches, for example the Cholesky decomposition. The second finding, was that the focus of attention needed to be upon the formulation of the mathematical model rather than different numerical analysis tools for solving the bundle adjustment. A modification of the calculation of IO partial derivatives was then proposed. While the proposed expansion of the terms for $x_p$ and $y_p$ in the configuration matrix $A$ were able to mitigate the solution instability to a considerable extent, the perspective model could still be expected to display shortcomings as the FoV narrowed. This was indicated by inflated variance estimates in the covariance matrix of parameters, as the
field of view became narrower than, say, $3^\circ - 4^\circ$. In such circumstances, which correspond in practical terms to the use of very long telephoto lenses on digital SLR cameras, the orthogonal projection model provided a viable alternative to the collinearity approach. A fully automated photogrammetric orientation procedure was developed for this purpose as a part of the Australis photogrammetric platform. Potential future work may include optimisation and possibly sequential processing of the image point observations in order to both provide more efficient algorithms and reduce the computational time.

Both the proposed methods for self-calibration of cameras employing long focal length lenses represent extensions of the existing theory. The author feels that these new development contribute to the current state-of-the-art in analytical close-range photogrammetry as well as increase the practical scope and the applications of digital close-range photogrammetry.
Author’s Published Work


Stamatopoulos, C. and Fraser, CS (2011). An orthogonal projection model for photogrammetric orientation of long focal length imagery. *AfricaGEO 2011*
References


REFERENCES

Francis, J. (1961). The qr transformation. a unitary analogue to the lr transformation. i. The Computer Journal, 4, 265–271. 40


Feyer, J. & Fraser, C. (1986). On the calibration of underwater cameras. The Photogrammetric Record, 12, 73–85. 103


REFERENCES


148
REFERENCES


Expressions for partial derivatives forming the configuration matrix $A$ related to camera calibration parameters

$$\frac{\partial f_x}{\partial c} = \frac{Z - Z_o}{H} \frac{r_{33} x_{corr} (r_{13} x_{corr} + r_{23} y_{corr})}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})}$$
\[
\frac{\partial f}{\partial x_p} = \frac{Z - Z_o}{H} \left( \frac{r_{33} c (-1 - K_1 r^2 - 2 K_1 (x - x_p)^2) (r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})^2} + \frac{x_{corr} (r_{13} (-1 - K_1 r^2 - 2 K_1 (x - x_p)^2) - 2 r_{23} K_1 (x - x_p) (y - y_p))}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})^2} \right)
\]

\[
\frac{\partial f}{\partial y_p} = \frac{Z - Z_o}{H} \left( -2 r_{33} c K_1 (x - x_p) (y - y_p) \frac{r_{33} c - r_{13} x_{corr} - r_{23} y_{corr}}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})^2} \right)
\]

\[
\frac{\partial f}{\partial K_1} = \frac{Z - Z_o}{H} \left( \frac{r_{33} c (x - x_p) r^2 (r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})^2} \right)
\]

\[
\frac{\partial f}{\partial c} = \frac{Z - Z_o}{H} \left( -r_{33} y_{corr} \frac{r_{13} x_{corr} + r_{23} y_{corr}}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})} \right)
\]

\[
\frac{\partial f}{\partial x} = \frac{Z - Z_o}{H} \left( -2 r_{33} c K_1 (x - x_p) (y - y_p) \frac{r_{33} c - r_{13} x_{corr} - r_{23} y_{corr}}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})^2} \right)
\]

\[
+ \frac{y_{corr} (r_{13} (-1 - K_1 r^2 - 2 K_1 (x - x_p)^2) - 2 r_{23} K_1 (x - x_p) (y - y_p))}{(r_{33} c - r_{13} x_{corr} - r_{23} y_{corr})^2} \right)
\]
\[
\frac{\partial f_y}{\partial y_p} = \frac{Z - Z_o}{H} \left( \frac{r_{33}c \left( 1 - K_1 r^2 - 2K_1 (y - y_p)^2 \right) (r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2} \right)
\]
\[
+ y_{corr} \left( -1 - K_1 r^2 - 2K_1 (x - x_p)^2 \right) \left( r_{33}c - r_{13}x_{corr} - r_{23}y_{corr} \right)^2
\]

\[
\frac{\partial f_y}{\partial K_1} = \frac{Z - Z_o}{H} \left( \frac{r_{33}c (x - x_p) r^2 (r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2} \right)
\]
\[
+ x_{corr} \left( -1 - K_1 r^2 - 2K_1 (x - x_p)^2 \right) \left( r_{33}c - r_{13}x_{corr} - r_{23}y_{corr} \right)^2
\]

Expressions for partial derivatives forming the configuration matrix B related to image coordinate observations

\[
\frac{\partial f_x}{\partial x} = \frac{Z - Z_o}{H} \left( \frac{r_{33}c \left( 1 + K_1 r^2 + 2K_1 (x - x_p)^2 \right) (r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2} \right)
\]
\[
+ x_{corr} \left( 1 + K_1 r^2 + 2K_1 (x - x_p)^2 \right) \left( r_{33}c - r_{13}x_{corr} - r_{23}y_{corr} \right)^2
\]

\[
\frac{\partial f_x}{\partial y} = \frac{Z - Z_o}{H} \left( \frac{2r_{33}cK_1 (x - x_p) (y - y_p) (r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2} \right)
\]
\[
+ x_{corr} \left( 2r_{33}cK_1 (x - x_p) (y - y_p) + 2r_{13}K_1 (x - x_p) (y - y_p)^2 \right) \left( r_{33}c - r_{13}x_{corr} - r_{23}y_{corr} \right)^2
\]
\[
\frac{\partial f_y}{\partial x} = \frac{Z - Z_o}{H} \left( \frac{2r_{33}cK_1 (x - x_p) (y - y_p) (r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2} \right.
\]
\[
+ \frac{y_{corr} (r_{13} (1 + K_1 r^2 + 2K_1 (x - x_p)^2) + 2r_{23}K_1 (x - x_p) (y - y_p))}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2}
\]
\[
\frac{\partial f_y}{\partial y} = \frac{Z - Z_o}{H} \left( \frac{r_{33}c (1 + K_1 r^2 + 2K_1 (y - y_p)^2) (r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2} \right.
\]
\[
+ \frac{y_{corr} (2r_{13}K_1 (x - x_p) (y - y_p) + r_{23} (1 + K_1 r^2 + 2K_1 (y - y_p)^2))}{(r_{33}c - r_{13}x_{corr} - r_{23}y_{corr})^2}
\]
Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
STAMATOPOULOS, CHRISTOS

Title:
Orientation and calibration of long focal length cameras in digital close-range photogrammetry

Date:
2011

Citation:

Persistent Link:
http://hdl.handle.net/11343/36256

File Description:
Orientation and calibration of long focal length cameras in digital close-range photogrammetry

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.