UCGE Reports
Number 20295

Department of Geomatics Engineering

New Strategies for Combining GNSS and Photogrammetric Data

(URL: http://www.geomatics.ucalgary.ca/research/publications)

by

Cameron Ellum

September 2009

UNIVERSITY OF CALGARY

New Strategies for Combining GNSS and Photogrammetric Data

by

Cameron MacKenzie Ellum

A DISSERTATION
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA
SEPTEMBER, 2009

Abstract

The basic concept of integrating GNSS and photogrammetry dates back more than 30 years and for at least the past decade it has been ubiquitous in both aerial and terrestrial mapping. Throughout all its history the basic technique for integrating the two technologies has been the same: GNSS data is post-processed using a Kalman filter yielding positions and these positions are then used as constraining information in a photogrammetric least-squares bundle adjustment. As evidenced by its long and essentially unaltered use the existing strategy works well, nevertheless it has some drawbacks. From a theoretical perspective, the integration is sub-optimal while the information flow is only in the one direction. From an operational perspective, the current approach is unwieldy: (at least) two processing packages are required.

The objective of the research contained within this work is to examine new strategies for integrating GNSS and photogrammetric data that alleviate the aforementioned limitations of the current integration strategy. Specifically, two new integration strategies are introduced, implemented, and tested:

- Inter-processor communication between a kinematic GNSS Kalman filter and a photogrammetric bundle adjustment.
- A combined least-squares adjustment of both GNSS and photogrammetric observations.

The first strategy introduces two-way communication between the GNSS and pho-

togrammetric processors, while the second strategy introduces measurement-level integration within a single processor. The combined adjustment also allows some more flexible GNSS processing options; for instance, a non-fixed base station or the use of observations when there are less than the 4 normally required.

Testing of the inter-processor communication strategy showed that it could help GNSS positioning following signal outages, yet this improvement does not necessarily translate into improved photogrammetric mapping accuracy. Testing of the combined adjustment demonstrated how photogrammetric control could replace a fixed GNSS base station, and how use of use of GNSS observations during partial signal blockages (when they would otherwise be discarded) could help bound the mapping and exposure position error growth. The combined adjustment testing also showed that the exposure positions derived from a typical aerial block of imagery have too much noise to substantially improve the GNSS positioning; consequently, mapping accuracy also does not improve.

Acknowledgments

If my acknowledgments are long, it is because I have many to acknowledge.

The Professional

Many companies and individual supplied data that, even when not ultimately used, was appreciated. These include Camal Dharamdial at The Orthoshop, Joe Hutton and Mohammed Mostafa at Applanix, Jan Skaloud at EPFL, and Kris Morin at Leica Geosystems.

The generosity of several funding organisations made my graduate studies comfortable. Most notably, I was the fortunate and proud recipient of an Izaak Walton Killam Memorial Scholarship. It was also with pride that I received the Robert E. Altenhofen Memorial Scholarship from the ASPRS. Lastly, NSERC and the Department of Geomatics provided funding through my supervisor's research grants and graduate research supplements, respectively.

NovAtel provided data, access to a signal simulator, employment, and a leave-of-absence from that employment during which I was able to largely complete my thesis; these contributions are gratefully acknowledged.

The members of my examining committee all provided feedback, corrections, and suggestions that improved the quality of this thesis.

The Personal

My deepest thanks goes to my supervisor and friend, Dr. Naser El-Sheimy. I was his first Graduate student, and I'm sure there were times when he (and I) thought I would be his last. However, through all our long time together he was consistently supportive, (more than) fair, and wise. In many roles – as a researcher, manager, promoter – he is without peer. A regret from my graduate studies is that I only ever learned a fraction of what he had to teach.

My parents, too, probably thought my Ph.D. would never end. While it has been some years since they could help me with my homework, they have helped me throughout my studies in virtually every other way. They always gave more than I asked for, and I asked for much.

Only with the encouragement and support from my wife Lynn was the completion of this thesis possible. When I look back upon my graduate studies it will not be this thesis I remember, it will be meeting her.

I cannot thank my daughter Cassie for helping with the progress of my thesis, since her smiles often drew me away from it. However, I can thank her for providing me with perspective: "Daddy, hug?" is worth more than any degree, any publication, or any academic accolade.

Contents

List of Tables

List of Figures

List of Acronymns

ADR Accumulated Delta (or Doppler) Range

AEOS Automated Empirical Optimisation of Software

ANOVA ANalysis-Of-VAriance

API Application Program Interface

ATLAS Automatically Tuned Linear Algebra Software

BLAS Basic Linear Algebra Subprograms

COCOMO COnstructive COst MOdel

CODE Center for Orbit Determination in Europe

CPU Central Processing Unit

CUPT Co-ordinate Update

DLL Dynamic-Linked Library

DSO Direct Sensor Orientation

ECEF Earth-Centred Earth-Fixed

GCP Ground Control Point

GIS Geographic Information System

GLONASS GLObal NAvigation Satellite System

GNSS Global Navigation Satellite System

GPS Global Positioning System

HTML Hypertext Markup Language

IGS International GNSS Service

ISO Integrated Sensor Orientation

IMU Inertial Measurement Unit

INS Inertial Navigation System

LAMBDA Least-squares Ambiguity Decorrelation Adjustment

LAPACK Linear Algebra Package

LIDAR LIght Detection and Ranging

LOS Line-of-Sight

NGS National Geodetic Survey

NIMA National Imagery and Mapping Agency

PRN Pseudo-Random Noise

PPP Precise Point Positioning

PV Position/Velocity

PVA Position/Velocity/Acceleration

RINEX Receiver INdependent EXchange Format

RMS Root Mean Square

RTTI Run-Time Type Information

RTK Real-Time Kinematic

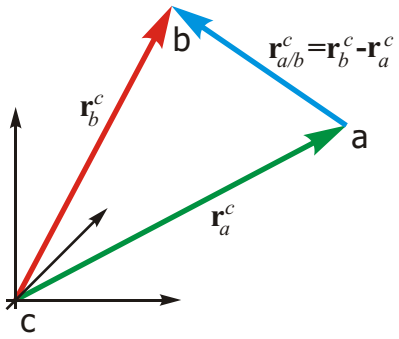
TEC Total Electron Content

VISAT Video-INS-SATellite

Notation

Convention

Vectors: Vectors are shown using bold lowercase letters and symbols. Position vectors, indicated by '**r**', have both a superscript and a subscript. The former indicates which frame the vector is expressed in, and the latter indicates the start and end points of the vector, separated by '/'. If the start point of a vector is the same as the origin of the frame in which the vector is expressed, then it is not shown. For example, $\mathbf{r}^c_{a/b}$ is the position of 'b' with respect to 'a', expressed in the 'c' frame. The same position, relative to the origin of the 'c' frame, would be $\mathbf{r}^c_a$. Both vectors are shown below.



Position Vectors

Matrices: Matrices are shown using bold uppercase letters and symbols. Rotation matrices between co-ordinate systems, indicated by '**R**', have a superscript and a subscript denoting the two co-ordinate frames. For example, $\mathbf{R}^b_a$ is the matrix that rotates vectors in co-

ordinate system 'a' to vectors in co-ordinate system 'b'. The elementary rotation matrices, corresponding to rotations about the x, y and z axes, respectively, are indicated by $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$.

Time: Differences between times use the same sub/superscript notational convention as is used for differences in position. For example, t_a^b is the time of event a measured in the b time-frame. Time-frame denoting superscripts are neglected in time differences, since all time frames are one-dimensional and with the same scale. Thus, $t_{a/b}^c = t_{a/b}^d = t_{a/b}$.

Chapter 1

Introduction

Satellite positioning and photogrammetry are complementary geomatic technologies. Photogrammetry enables the efficient and remote measurement of large numbers of widely separated positions. However, with only image measurements it operates in a frame with undefined scale, rotation, and position. Furthermore, the relative nature of image measurements cause networks constructed from them to deform over space, limiting photogrammetry's relative accuracy. Positioning using GNSSs, in contrast, is inefficient for the bulk measurement of widely separated points, but it provides positions that are both accurate and whose accuracy is indifferent to location. Used together then, GNSS positioning and photogrammetry enable the efficient and remote bulk determination of positions with both high absolute and relative accuracy.

The basic concept of integrating GNSS and photogrammetry dates back more than 30 years (Brown, 1976); its realisation began over 20 years ago as one of the first applications of kinematic GNSS positioning (Ackermann, 1984; Lucas, 1987); and for at least the past decade it has been ubiquitous in both aerial and terrestrial mapping. Throughout all its history the basic technique for integrating the two technologies has been the same: GNSS data is post-processed using a Kalman filter yielding positions, and these positions are then used as constraining information in a photogrammetric least-squares bundle adjustment.

The constraints are applied in the bundle adjustment using (interpolated) parameter observations of the exposure positions; hence the technique can be termed integration by position observations.

1.1 Objectives

It would be inaccurate and unfair to state that the current position-observation strategy has any serious limitations; its long and essentially unaltered use is a testament to its merit. Nevertheless, there is room for improved GNSS/photogrammetric integration. In the current scheme, information is transferred from the GNSS processor to the photogrammetric processor, but not vice-versa. From a theoretical perspective, the integration is sub-optimal while the information flow is only in the one direction. From an operational perspective, the current approach is unwieldy: (at least) two processing packages are required. This requires operators to be familiar with multiple software packages, and necessitates tedious and potentially error-inducing transfer of information between processors.

The objective of the research contained within this work is to examine new strategies for integrating GNSS and photogrammetric data that alleviate the aforementioned limitations of the current integration strategy. Specifically, two new integration strategies will be introduced, implemented, and tested:

- Inter-processor communication between a kinematic GNSS Kalman filter and a photogrammetric bundle adjustment.
- A combined least-squares adjustment of both GNSS and photogrammetric observations.

The first strategy introduces two-way communication between the GNSS and photogrammetric processors, while the second strategy introduces measurement-level integration within a single processor. The two-way communication and measurement integration should improve both the reliability and accuracy of the GNSS/photogrammetric integration; in particular,

the additional photogrammetric information should result in more accurate GNSS positioning and in more reliable ambiguity resolution. The single combined-adjustment processor should streamline the integration process.

1.2 Outline and Contributions

The work in this thesis combines estimation, GNSS, and photogrammetry; consequently, in the first three chapters of this thesis background is provided on all. Specifically,

- Chapter two provides a background on estimation using least squares and Kalman filtering. The formulae and material in this chapter will frequently be referred to in later chapters. The contents of this chapter is available in introductory estimation textbooks, and readers familiar with both least squares and Kalman filtering can safely skip ahead. The review of least squares solution constraints and solution strategies other than via the normal equations are, however, less commonly encountered in estimation reviews.

- Chapter three introduces and reviews satellite positioning. Because the primary intended audience for this thesis is photogrammetrists, this review is moderately detailed, with particular emphasis given to GNSS error mitigation and ambiguity resolution. Except for the presentation of ambiguity validation as an ANalysis-Of-VAriance (ANOVA) test (which the author has not encountered elsewhere), there is no new material in this chapter. The review of the observation equations, however, considers observations made on different frequencies – a topic covered rarely or as an afterthought in most other reviews.

- Chapter four reviews photogrammetry. This includes a comprehensive review of existing strategies for integrating GNSS and photogrammetric data. Some new theory and analysis are also introduced in this chapter: the effects of incorrect ambiguity

resolution on exposure station position are analysed, and the georeferencing image measurement equations are introduced and derived.

The balance of the thesis describes the implementation of the integration strategies and provides results from their use. Drawing upon the material of the previous three chapters, Chapter five contains details the implementation of the combined adjustment and the inter-processor communication strategies. Practical software-development considerations rarely included in geomatic engineering publications are covered in this Chapter, with a particular focus on software optimisation. Chapter six contains observations, results, and analysis from tests of the new integration strategies. Lastly, in Chapter seven the key findings from this work are reviewed and suggestions for related further research and development are given.

The relationships between this thesis's material are shown in Figure 1.1. The material presented in the estimation chapter is used in all subsequent chapters. The satellite positioning and photogrammetry chapters are independent of one another, but both draw upon the estimation material. The final three chapters naturally draw together all of the preceding background and other material.

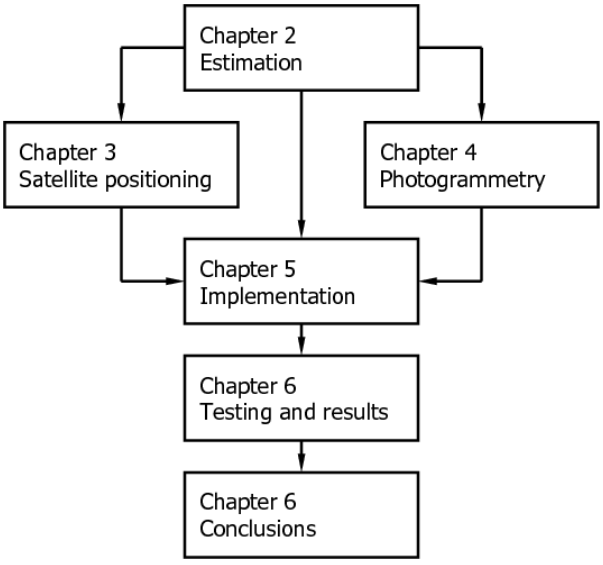


Figure 1.1: Thesis flowchart

Chapter 2

Estimation

The estimation principles introduced in this chapter will be used throughout the balance of this thesis. Satellite positioning, introduced and reviewed in the following chapter, uses both least squares and Kalman filtering to estimate positions, while a core part of ambiguity resolution is constrained least squares. In photogrammetry, large least-squares adjustments are used to obtain three-dimensional user-space co-ordinates from two-dimensional image co-ordinates.

2.1 Least Squares

Least squares is an estimator that determines the parameters of a model from redundant measurements. If the model is linear with respect to the measurements, then the relationship between the model parameters and measurements can be described by the linear equation

$$\mathbf{A}\mathbf{x} = \mathbf{l} \tag{2.1}$$

where $\mathbf{x}$ are the parameters and $\mathbf{l}$ the measurements. Unfortunately, with redundant, noisy, measurements the above equation cannot be perfectly satisfied. To make the system con-

sistent, the measurements have to be 'adjusted' by small amounts,

$$\mathbf{Ax} = \mathbf{l} + \mathbf{v},\ E\{\mathbf{v}\} = \mathbf{0},\ E\{\mathbf{vv}^T\} = \mathbf{C_l} \tag{2.2}$$

In this equation, $\mathbf{v}$ are the adjustments to the measurements, known as residuals. $\mathbf{C_l}$ is the measurement covariance matrix.

A descriptive definition of a least squares solution is one where the sum of the squared weighted residuals is minimised. The equivalent mathematical definition is

$$f(\mathbf{v}) = \mathbf{v}^T\mathbf{C_l}^{-1}\mathbf{v} = minimum. \tag{2.3}$$

where $\mathbf{C_l}$ is the variance-covariance matrix of the observations. Solving for the model parameters while satisfying this condition is known as least squares minimisation.

2.1.1 Derivation

The least squares normal equations can be derived from a number of approaches. Three of these are reviewed below. Several simplifications that ease the derivations are made. In particular, a distinction between cofactor and covariance matrices is not made, nor is a notational distinction made to indicate that the parameters solved for are estimated and not the actual parameters. Apart from this latter change, the notation of Krakiwsky (1990) is (generally) followed throughout.

Differentiation

The simplest manner with which to derive the least squares equations is to observe that $f(\mathbf{v})$ will be at an extrema when its derivative is zero (Rogers, 2003; Walpole et al., 2007).

Accordingly, taking the derivative,

$$\begin{aligned}\frac{\partial f(\mathbf{v})}{\partial \mathbf{x}} &= \frac{\partial f(\mathbf{v})}{\partial \mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} \\ &= \left(2\mathbf{v}^T\mathbf{C}_l^{-1}\right)(\mathbf{A}) \\ &= 2\left(\mathbf{A}\mathbf{x}-\mathbf{l}\right)^T\mathbf{C}_l^{-1}\mathbf{A} \\ &= 2\mathbf{x}^T\mathbf{A}^T\mathbf{C}_l^{-1}\mathbf{A} \ - 2\mathbf{l}^T\mathbf{C}_l^{-1}\mathbf{A}, \end{aligned} \tag{2.4}$$

and setting to zero,

$$\begin{aligned} 0 &= 2\mathbf{x}^T\mathbf{A}^T\mathbf{C}_l^{-1}\mathbf{A} \ - 2\mathbf{l}^T\mathbf{C}_l^{-1}\mathbf{A} \\ \mathbf{x}^T\mathbf{A}^T\mathbf{C}_l^{-1}\mathbf{A} &= \mathbf{l}^T\mathbf{C}_l^{-1}\mathbf{A} \\ \mathbf{A}^T\mathbf{C}_l^{-1}\mathbf{A}\mathbf{x} &= \mathbf{A}^T\mathbf{C}_l^{-1}\mathbf{l}. \end{aligned} \tag{2.5}$$

Or,

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{l}. \tag{2.6}$$

This extrema will be the desired minima if the second derivative is positive. Differentiating Equation (2.4) again,

$$\frac{d^2 f(\mathbf{v})}{d\mathbf{x}^\mathbf{2}} = 2\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}. \tag{2.7}$$

As a variance-covariance matrix, $\mathbf{C}_\mathbf{l}^{-1}$ is, by definition, positive-definite. Consequently, the second derivative will be greater than zero, and so the extrema given by Equation (2.6) will be a minimum.

Equation(2.5) is the least squares system of equations, and Equation (2.6) is the solution to this system. The co-efficient matrix and vector of constants in the system of equations

are known as the normal matrix and normal vector. Respectively, they are

$$\mathbf{N} = \mathbf{A}^T \mathbf{C}_\mathbf{l}^{-1} \mathbf{A} \tag{2.8}$$

$$\mathbf{u} = \mathbf{A}^T \mathbf{C}_\mathbf{l}^{-1} \mathbf{l}. \tag{2.9}$$

With these abbreviations, the least squares solution can compactly be expressed as

$$\mathbf{x} = \mathbf{N}^{-1} \mathbf{u}. \tag{2.10}$$

The covariance of the estimated parameters can be found by performing error propagation on the adjusted parameters using the covariance of the observations,

$$\mathbf{C}_\mathbf{x} = \left(\frac{\partial \mathbf{x}}{\partial \mathbf{l}} \right) \mathbf{C}_\mathbf{l} \left(\frac{\partial \mathbf{x}}{\partial \mathbf{l}} \right)^T. \tag{2.11}$$

Differentiating Equation (2.5),

$$\left(\frac{\partial \mathbf{x}}{\partial \mathbf{l}} \right) = \left(\mathbf{A}^T \mathbf{C}_y^{-1} \mathbf{A} \right)^{-1} \mathbf{A}^T \mathbf{C}_\mathbf{l}^{-1}, \tag{2.12}$$

and evaluating Equation (2.11) yields,

$$\mathbf{C}_\mathbf{x} = \mathbf{N}^{-1}. \tag{2.13}$$

Constrained Optimisation

A more flexible derivation of the least squares equations can be approached by viewing least squares as a constrained minima problem. In this case, $f(\mathbf{x})$ in Equation (2.3) is minimized under the constraint

$$\mathbf{A}\mathbf{x} = \mathbf{l} + \mathbf{v} \;\Rightarrow\; \mathbf{g}(\mathbf{v}, \mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{l} - \mathbf{v} = \mathbf{0} \tag{2.14}$$

The standard technique of solving a constrained minima problem is to use Lagrange multipliers (also known as Lagrange undetermined multipliers). The Lagrange multipliers are nuisance parameters used in the Lagrangian or variation function. In this case, the function is

$$\phi = \mathbf{v}^T \mathbf{C}_\mathbf{l}^{-1} \mathbf{v} - 2\mathbf{k}^T (\mathbf{A}\mathbf{x} - \mathbf{l} - \mathbf{v}) \tag{2.15}$$

where $\mathbf{k}$ are the Lagrange multipliers (the scalar -2 is used just for later convenience). The minima of the variation function occurs when the partial derivatives with respect to $\mathbf{v}$ and $\mathbf{x}$ are zero,

$$\frac{\partial \phi}{\partial \mathbf{v}} = -2\mathbf{v}^T \mathbf{C}_\mathbf{l}^{-1} + 2\mathbf{k}^T \mathbf{I} = \mathbf{C}_\mathbf{l}^{-1} \mathbf{v} + \mathbf{I}\mathbf{k} = \mathbf{0} \tag{2.16}$$

$$\frac{\partial \phi}{\partial \mathbf{x}} = -2\mathbf{k}^T \mathbf{A} = \mathbf{A}^T \mathbf{k} = \mathbf{0}. \tag{2.17}$$

These equations, together with Equation (2.2), form a system of equations. Rearranging Equation (2.16),

$$\mathbf{k} = -\mathbf{C}_\mathbf{l}^{-1} \hat{\mathbf{v}} \tag{2.18}$$

and observing that, from Equation (2.2),

$$\mathbf{v} = \mathbf{l} - \mathbf{A}\mathbf{x} \tag{2.19}$$

leads to

$$\mathbf{k} = -\mathbf{C}_\mathbf{l}^{-1} (\mathbf{l} - \mathbf{A}\mathbf{x}) \tag{2.20}$$

that, when substituted into Equation (2.17), produces the same expression as in Section 2.1.1,

$$\begin{aligned}
\mathbf{A}^T\mathbf{k} &= \mathbf{0} \\
-\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\left(\mathbf{l}-\mathbf{A}\mathbf{x}\right) &= \mathbf{0} \\
-\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}\mathbf{l}+\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{0} \\
\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}\mathbf{l}
\end{aligned}$$

The constrained minima derivation of the least squares solution is the technique most often used in Geodesy (cf. Mikhail, 1976; Krakiwsky, 1990; Leick, 1995; Cooper and Robson, 1996; Kuang, 1996). It is easily extended to account for more complex models: for instance, additional constraints on the parameters.

Maximum Likelihood

Neither of the two previous approaches say anything about the probability distribution of either the observations or the estimated parameters. If, however, the residuals are known (or, more likely, assumed) to be normally distributed, then least squares can be derived as a maximum likelihood estimation problem.

When the probability distribution of the residuals is considered, the relationship between the observations, parameters, and residuals is given by

$$\mathbf{A}\mathbf{x} = \mathbf{l} + \mathbf{v},\ \ \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\mathbf{l}). \tag{2.21}$$

The only difference between this equation and Equation (2.2) is that this equation specifically describes the probability distribution of the residuals: they are normally distributed with zero-mean and covariance $\mathbf{C}_\mathbf{l}$. The corresponding joint probability density function of the residuals is a multivariate normal distribution (or multinormal distribution, cf. Mikhail,

1976), given by

$$f_{\mathbf{x}}(\mathbf{v}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{C_l}|}} \exp\left[-\frac{1}{2}(\mathbf{v} - E\{\mathbf{v}\})^T \mathbf{C_l}^{-1}(\mathbf{v} - E\{\mathbf{v}\})\right], \tag{2.22}$$

where $|\mathbf{C_l}|$ is the determinant of $\mathbf{C_l}$. Since the residuals are zero-mean, $E\{\mathbf{v}\} = \mathbf{0}$, and

$$f_{\mathbf{x}}(\mathbf{v}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{C_l}|}} \exp\left(-\frac{1}{2}\mathbf{v}^T \mathbf{C_l}^{-1}\mathbf{v}\right) \tag{2.23}$$

Reversing the roles of the parameters and residuals, so that the parameters are the variables and the residuals are the given information, produces the likelihood function

$$\mathcal{L}_{\mathbf{v}}(\mathbf{x}) = f_{\mathbf{x}}(\mathbf{v}) \tag{2.24}$$

At this point, it can already be seen that minimising $\mathbf{v}^T \mathbf{C_l}^{-1}\mathbf{v}$ will maximise $f_{\mathbf{x}}(\mathbf{v})$ and hence $\mathcal{L}_{\mathbf{v}}(\mathbf{x})$. However, the maximum likelihood estimator is typically derived by maximising the log-likelihood function. Accordingly,

$$\ln \mathcal{L}_{\mathbf{v}}(\mathbf{x}) = \mathcal{L}^*_{\mathbf{v}}(\mathbf{x}) = -\frac{N}{2}\ln 2\pi - \frac{1}{2}\sqrt{|\mathbf{C_l}|} - \frac{1}{2}\mathbf{v}^T \mathbf{C_l}^{-1}\mathbf{v} \tag{2.25}$$

The maximum of this function is obtained when $\mathbf{v}^T \mathbf{C_l}^{-1}\mathbf{v}$ is minimised. This, in turn, will occur when its derivative is zero,

$$\frac{d}{d\mathbf{x}}\mathcal{L}^*_{\mathbf{v}}(\mathbf{x}) = \frac{d}{d\mathbf{x}}\mathbf{v}^T \mathbf{C_l}^{-1}\mathbf{v} = 0. \tag{2.26}$$

This derivative is the same as was used in the derivation of the least squares equations by differentiation (Equations (2.4) through (2.5), above); hence, the solution will be the same, and the maximum likelihood estimator of the parameters is given by

$$\mathbf{x} = \left(\mathbf{A}^T \mathbf{C_l}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{C_l}^{-1}\mathbf{l}. \tag{2.27}$$

This is the same expression as for the least squares estimate, proving the equivalency of the two estimates.

To reiterate: the equivalency between the maximum likelihood and least squares estimators exists only when the residuals are normally distributed. If the residuals have some other distribution then the parameters solved for are the best only in the sense that they minimise the residuals. This is likely not the estimate desired, and so the assumption of normality is typically tested for after a least squares adjustment.

2.1.2 Non-linear observation equations

Until now, the observational system has been linear. Frequently, however, the relationship between the parameters is non-linear. The least squares technique easily accommodates this through first-order Taylor series linearisation. A non-linear system,

$$\mathbf{f}(\mathbf{x}) = \mathbf{l} - \mathbf{v} \tag{2.28}$$

can be approximated by a first-order Taylor series expansion as

$$\mathbf{f}(\mathbf{x^0}) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\left(\mathbf{x^0}\right)\left(\mathbf{x} - \mathbf{x^0}\right) = \mathbf{l} - \mathbf{v} \tag{2.29}$$

$$\mathbf{A}\boldsymbol{\delta} = -\mathbf{w} - \mathbf{v} \tag{2.30}$$

where $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\left(\mathbf{x^0}\right)$, $\boldsymbol{\delta} = \left(\mathbf{x} - \mathbf{x^0}\right)$ and $\mathbf{w} = \mathbf{f}(\mathbf{x^0}) - \mathbf{l}$. This equation is linear, and can be adjusted using any of the previously shown techniques. The only difference is that the solution vector $\boldsymbol{\delta}$ is now a set of corrections to the current point of expansion $\mathbf{x^0}$. The adjustment is run iteratively, until the corrections become insignificant.

2.1.3 Solution strategies

In Section 2.1.1, the existence of the least squares estimate was shown by deriving the normal equations by a variety of techniques. Using the normal equations the least squares

solution is easily found by Cholesky decomposition. However, the normal equations are not the only technique of solving least squares problems. Two other techniques commonly used are the QR and singular value orthogonal decompositions.

Cholesky Decomposition of the Normal Equations

In Geomatics (outside of physical Geodesy, at least) the least squares estimate is almost always found using the Cholesky decomposition of the normal matrix (e.g., Mikhail, 1976; Krakiwsky, 1990; Leick, 1995; Cooper and Robson, 1996; Kuang, 1996). Recalling Equation (2.10), the normal system of equations is

$$\mathbf{N}\mathbf{x} = \mathbf{u}. \tag{2.31}$$

Since the normal matrix $\mathbf{N}$ is symmetric and positive-definite it can efficiently be decomposed via Cholesky decomposition into

$$\mathbf{N} = \mathbf{L}\mathbf{L}^T. \tag{2.32}$$

Or, equivalently,

$$\mathbf{N} = \mathbf{R}^T\mathbf{R}. \tag{2.33}$$

Using the Cholesky decomposition, the normal equations solution can can be rewritten as

$$\begin{aligned} \mathbf{L}\mathbf{L}^T\mathbf{x} &= \mathbf{u} \\ \mathbf{L}\mathbf{z} &= \mathbf{u} \end{aligned} \tag{2.34}$$

with

$$\mathbf{L}^T\mathbf{x} = \mathbf{z}. \tag{2.35}$$

To solve for $\mathbf{x}$ the Equation (2.34) is solved for $\mathbf{z}$, and then Equation (2.35) is solved for $\mathbf{x}$. Because both systems are triangular they can be solved very easily.

The problem with solving least squares problems using the normal equations and Cholesky decomposition is that, for some problems, the normal matrix can become ill-conditioned. This ill-conditioning results from the implicit squaring of elements in the $\mathbf{A}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{A}$ product. When an ill-conditioned normal system is solved, the estimated parameters may not be reliable. Fortunately, in most adjustments done in geomatics the normal equations are well-conditioned. However, there are some instances – for example, when fitting higher order polynomials – where the conditioning of the normal matrix becomes an issue.

QR Decomposition

The potential ill-conditioning of the normal equations is why most mathematicians advocate using orthogonal decompositions when solving least squares problems. A common choice for orthogonal decomposition is a QR decomposition. This decomposition factorizes a matrix into an orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$,

$$\mathbf{A} = \mathbf{QR}. \tag{2.36}$$

The QR decomposition solution to the least squares problem is most easily shown using the normal system of equations and a QR factorisation of the design matrix $\mathbf{A}$, assuming an identity measurement covariance matrix $\mathbf{C}_\mathbf{l}$,

$$\begin{aligned}
\mathbf{A}^T\mathbf{Ax} &= \mathbf{A}^T\mathbf{l} \\
(\mathbf{QR})^T\mathbf{QRx} &= (\mathbf{QR})^T\mathbf{l} \\
\mathbf{R}^T\mathbf{Q}^T\mathbf{QRx} &= \mathbf{R}^T\mathbf{Q}^T\mathbf{l} \\
\mathbf{QRx} &= \mathbf{l}
\end{aligned} \tag{2.37}$$

Thus, factorising the coefficient matrix $\mathbf{A}$ by QR decomposition and solving the resulting system of equations is equivalent to solving the normal system of equations. A non-identity covariance matrix is easily considered by noting that any normal system of equations can be converted to the above form by decomposing $\mathbf{C_l}$ into $\mathbf{L}\mathbf{L}^T$,

$$\begin{aligned} \mathbf{A}^T\mathbf{C_l}\mathbf{A} &= \mathbf{A}^T\mathbf{C_l}\mathbf{l} \\ \mathbf{A}^T\mathbf{L}\mathbf{L}^T\mathbf{A} &= \mathbf{A}^T\mathbf{L}\mathbf{L}^T\mathbf{l} \\ \widetilde{\mathbf{A}}^T\widetilde{\mathbf{A}}\mathbf{x} &= \widetilde{\mathbf{A}}^T\widetilde{\mathbf{l}} \end{aligned} \tag{2.38}$$

It is, of course, also possible to derive the QR decomposition solution to the least squares problem without starting from the normal system of equations; see, for instance, Lawson and Hanson (1974).

Singular value decomposition

A technique for solving least squares problems that is even more stable than using a QR decomposition is to use a singular value decomposition. The singular value decomposition of a matrix is given as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{2.39}$$

where $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of $\mathbf{A}$, and both $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. The singular values (but not necessarily their order) are unique, while the orthogonal matrices are not. Substituting the singular value decomposition of $\mathbf{A}$ into the normal system of equations and, for convenience, assuming an identity measurement

covariance matrix $\mathbf{C_l}$ yields

$$\begin{aligned}
\mathbf{A}^T\mathbf{A}\mathbf{x} &= \mathbf{A}^T\mathbf{l} \\
\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} &= \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T\mathbf{l} \\
\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} &= \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{l} \\
\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} &= \mathbf{l}
\end{aligned} \tag{2.40}$$

Thus, performing a singular value decomposition of $\mathbf{A}$ and solving the resulting system is a least squares solution. Since $\mathbf{\Sigma}$ is diagonal and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal the solution is trivial, although the singular value decomposition of $\mathbf{A}$ is not.

Sequential least squares

Sometimes in adjustments the observations naturally fall into two groups that are statistically independent. For example, there may be different types of observations or the observations may be collected at different epochs. When this is the case, it can be convenient to process the observations sequentially. Considering two groups of independent observations, the normal system can be written as

$$\begin{aligned}
\begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}^T\begin{pmatrix}\mathbf{C}_{\mathbf{l}_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{\mathbf{l}_2}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}\mathbf{x} &= \begin{pmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{pmatrix}^T\begin{pmatrix}\mathbf{C}_{\mathbf{l}_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{\mathbf{l}_2}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{l}_1\\ \mathbf{l}_2\end{pmatrix} \\
\left(\mathbf{A}_1^T\mathbf{C}_{\mathbf{l}_1}\mathbf{A}_1 + \mathbf{A}_2^T\mathbf{C}_{\mathbf{l}_2}\mathbf{A}_2\right)\mathbf{x} &= \left(\mathbf{A}_1^T\mathbf{C}_{\mathbf{l}_1}\mathbf{l}_1 + \mathbf{A}_2^T\mathbf{C}_{\mathbf{l}_2}\mathbf{l}_2\right)
\end{aligned} \tag{2.41}$$

The solution from all observations is

$$\mathbf{x} = \left(\mathbf{N}_1 + \mathbf{A}_2^T\mathbf{C}_{\mathbf{l}_2}\mathbf{A}_2\right)^{-1}\left(\mathbf{u}_1 + \mathbf{A}_2^T\mathbf{C}_{\mathbf{l}_2}\mathbf{l}_2\right) \tag{2.42}$$

where $\mathbf{N}_1 = \mathbf{A}_1^T \mathbf{C}_{\mathbf{l}_1} \mathbf{A}_1$ and $\mathbf{u}_1 = \mathbf{A}_1^T \mathbf{C}_{\mathbf{l}_1} \mathbf{l}_1$. Using the Sherman-Morrison-Woodbury Formula (see Appendix A), this can be expressed by

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{N}_1^{-1} \mathbf{A}_2^T \left(\mathbf{C}_{\mathbf{l}_2} + \mathbf{A}_2 \mathbf{N}_1^{-1} \mathbf{A}_2^T \right)^{-1} \left(\mathbf{l}_2 - \mathbf{A}_2 \mathbf{x}_1 \right) \tag{2.43}$$

In other words, if a solution has already been found using one set of observations, then this solution can be updated by the second set of observations. The covariance can similarly be updated Krakiwsky (1990),

$$\mathbf{C}_{\mathbf{x}} = \mathbf{C}_{\mathbf{x}_1} - \mathbf{N}_1^{-1} \mathbf{A}_2^T \left(\mathbf{C}_{\mathbf{l}_2} + \mathbf{A}_2 \mathbf{N}_1^{-1} \mathbf{A}_2^T \right)^{-1} \mathbf{A}_2 \mathbf{N}_1^{-1} \tag{2.44}$$

Obviously, the normal matrix inversion required in the first step of this two-stage approach implies that the original problem is well-posed. In other words, it is not possible to combine two rank-deficient problems and get a single unique solution using this approach.

If the inverted normal matrix is not required – for instance, while iterating a non-linear adjustment – then it is apparent that the normal matrix in Equation (2.42) can be built-up by summing the normal matrices and vectors,

$$\left(\sum_{i=0}^{n} \mathbf{N}_i \right) \mathbf{x} = \sum_{i=0}^{n} \mathbf{u}_i \tag{2.45}$$

Unlike the two-step sequential solution of Equations (2.43) and (2.44), each contribution in the summation-of-normals can be rank-deficient; a unique solution is possible as long as the sum is not.

2.1.4 Constraints

Frequently in least squares adjustments the behaviour of the parameters is somehow constrained. For example, in an aerial photogrammetric adjustment if a number of unknown points are on a lake then this information can be used within the adjustment: either the

points can be constrained to all have the same elevation, or, if the elevation of the lake is known, the points can be constrained to the known elevation. The least squares solution equations of the above sections can be modified to account for such parameter constraints.

Constrained Minima Problem

The derivation of the least squares problem with parameter constraints is easily done by once again treating the problem as a constrained minima problem. The only difference is an additional constraint equation,

$$\mathbf{Gx} = \mathbf{d}, \tag{2.46}$$

leading to an expanded variation function with a second set of Lagrangian multipliers,

$$\phi = \mathbf{v}^T \mathbf{C}_\mathbf{l}^{-1} \mathbf{v} - 2\mathbf{k}_1^T \left(\mathbf{Ax} - \mathbf{l} - \mathbf{v}\right) - 2\mathbf{k}_2^T \left(\mathbf{Gx} - \mathbf{d}\right). \tag{2.47}$$

Once again, the minima of the variation function occurs when the partial derivatives with respect to $\mathbf{v}$ and $\mathbf{x}$ are zero,

$$\frac{\partial \phi}{\partial \mathbf{v}} = \mathbf{C}_\mathbf{l}^{-1}\mathbf{v} + \mathbf{I}\mathbf{k}_1 = \mathbf{0} \tag{2.48}$$

$$\frac{\partial \phi}{\partial \mathbf{x}} = \mathbf{A}^T\mathbf{k}_1 + \mathbf{G}^T\mathbf{k}_2 = \mathbf{0}. \tag{2.49}$$

Eliminating the residuals and Lagrange multipliers from the system of equations formed by Equations (2.48), (2.49), and (2.2) leads to an expression for the parameters in the presence of constraints,

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{N}^{-1}\mathbf{G}^T\left(\mathbf{G}\mathbf{N}^{-1}\mathbf{G}^T\right)^{-1}\left(\mathbf{d} - \mathbf{G}\mathbf{x}^0\right) \tag{2.50}$$

This solution in the presence of constraints has the intuitive and attractive property that it is the unconstained solution, $\tilde{\mathbf{x}}$, modified for the constraints. The covariance matrix of the

parameters has a similar form,

$$\mathbf{C}_{\mathbf{x}} = \mathbf{C}_{\tilde{\mathbf{x}}} - \mathbf{N}^{-1}\mathbf{G}^T\left(\mathbf{G}\mathbf{N}^{-1}\mathbf{G}^T\right)^{-1}\mathbf{G}\mathbf{N}^{-1} \tag{2.51}$$

An additional observation can be made from this equation: namely, the covariance of the constrained solution will always be less than that of the unconstrained solution.

The explicit derivation of Equations (2.50) and (2.51) – i.e., the steps from Equations (2.48) and (2.49) – not provided above can be found in Mikhail (1976). The solution with constraints can also be derived from sequential least squares, Equation (2.43), by setting the covariance of the second set of observations (i.e., the constraints) to zero ($\mathbf{C}_{l_2} = \mathbf{0}$). This is only possible because the inverse of this covariance matrix is not required.

A common use of constraints is to update the results of an adjustment when some subset of the parameters are set to known values. In other words, the parameter vector is divided into two subsets,

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}. \tag{2.52}$$

and one of these subsets is constrained to known values,

$$\mathbf{x}_1 = \tilde{\mathbf{x}}_1 \tag{2.53}$$

In this case,

$$\mathbf{G} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{2.54}$$

Evaluating Equations (2.50) and (2.51) with this constraint matrix yields expressions for the unconstrained parameters,

$$\check{\mathbf{x}}_2 = \mathbf{x}_2 - \mathbf{C}_{\mathbf{x}21}\mathbf{C}_{\mathbf{x}11}^{-1}\left(\mathbf{x}_1 - \check{\mathbf{x}}_1\right), \tag{2.55}$$

and the unconstrained parameter covariance,

$$\mathbf{C}_{\mathbf{x}22} = \mathbf{C}_{\mathbf{x}^0 22} - \mathbf{C}_{\mathbf{x}^0 21}\mathbf{C}_{\mathbf{x}^0 11}\mathbf{C}_{\mathbf{x}^0 21}^T. \tag{2.56}$$

Of course, an alternate way to deal with constraints of the form in Equation (2.53) is to simply repeat the adjustment and estimate only the unconstrained parameters. For large adjustments, however, this is less efficient than applying the constraints to an existing solution.

Parameter Elimination

In a least squares adjustment a computationally-expensive full inversion of the normal matrix is only necessary if the covariance of the adjusted parameters is required: see Equation (2.13). However, the previous expressions for the least squares solution with hard constraints all required the inverted normal matrix. Consequently, if the adjusted parameter covariance is not required then they are not ideal.

Shum and Szeliski (1999) present a technique of applying hard constraints without requiring the inverted normal matrix. A single constraint on the parameters can be written as

$$\mathbf{g}^T\mathbf{x} = d. \tag{2.57}$$

This constraint can be arbitrarily scaled without altering the relationships between the parameters. Accordingly, scaling it so that element k is 1.0,

$$g_k^{-1}\mathbf{g}^T\mathbf{x} = g_k^{-1}d \tag{2.58}$$

$$\tilde{\mathbf{g}}^T\mathbf{x} = \tilde{d} \tag{2.59}$$

The k^{th} column of $\mathbf{A}$, $\mathbf{a}_k$, can be zeroed by subtracting this linear constraint,

$$\left(\mathbf{A} - \mathbf{a}_k\tilde{\mathbf{g}}^T\right)\mathbf{x} = \mathbf{l} - \mathbf{a}_k\tilde{d} \tag{2.60}$$

$$\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{l}} \tag{2.61}$$

$\tilde{\mathbf{A}}$ is now rank deficient and thus not solvable using least squares techniques developed in Section 2.1.3. The rank deficiency can, however, be remedied by combining both Equations (2.61) and (2.57) in a new normal system,

$$\begin{pmatrix}\tilde{\mathbf{A}}\\ \tilde{\mathbf{g}}^T\end{pmatrix}^T\begin{pmatrix}\tilde{\mathbf{A}}\\ \tilde{\mathbf{g}}^T\end{pmatrix}\mathbf{x} = \begin{pmatrix}\tilde{\mathbf{A}}\\ \tilde{\mathbf{g}}\end{pmatrix}^T\begin{pmatrix}\tilde{\mathbf{l}}\\ \tilde{d}\end{pmatrix}$$
$$\left(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\right)\mathbf{x} = \left(\tilde{\mathbf{A}}^T\tilde{\mathbf{l}} + \tilde{\mathbf{g}}\tilde{d}\right). \tag{2.62}$$

Or,

$$\tilde{\mathbf{N}}\mathbf{x} = \tilde{\mathbf{u}} \tag{2.63}$$

Expanding the normal matrix,

$$\begin{aligned}\tilde{\mathbf{N}} &= \tilde{\mathbf{A}}^T\tilde{\mathbf{A}} + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\\ &= \left(\mathbf{A} - \mathbf{a}_k\tilde{\mathbf{g}}^T\right)^T\left(\mathbf{A} - \mathbf{a}_k\tilde{\mathbf{g}}^T\right) + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\\ &= \mathbf{A}^T\mathbf{A} - \mathbf{A}^T\mathbf{a}_k\tilde{\mathbf{g}}^T - \tilde{\mathbf{g}}\mathbf{a}_k{}^T\mathbf{A} + \mathbf{g}\mathbf{a}_k{}^T\mathbf{a}_k\tilde{\mathbf{g}}^T + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\\ &= \mathbf{N} - \mathbf{n}_k\tilde{\mathbf{g}}^T - \tilde{\mathbf{g}}\mathbf{n}_k{}^T + \mathbf{g}n_{kk}\tilde{\mathbf{g}}^T + \tilde{\mathbf{g}}\tilde{\mathbf{g}}^T\end{aligned} \tag{2.64}$$

and normal vector,

$$\begin{aligned}\tilde{\mathbf{u}} &= \bar{\mathbf{A}}^T\tilde{\mathbf{l}} + \mathbf{g}\tilde{d} \\ &= \left(\mathbf{A} - \mathbf{a}_k\tilde{\mathbf{g}}^T\right)^T\left(\mathbf{l} - \mathbf{a}_k\tilde{d}\right) + \mathbf{g}\tilde{d} \\ &= \mathbf{A}^T\mathbf{l} - \mathbf{A}^T\mathbf{a}_k\tilde{d} - \tilde{\mathbf{g}}{\mathbf{a}_k}^T\mathbf{l} + \tilde{\mathbf{g}}{\mathbf{a}_k}^T\mathbf{a}_k\tilde{d} + \mathbf{g}\tilde{d} \\ &= \mathbf{u} - \mathbf{n}_k\tilde{d} - \tilde{\mathbf{g}}u_k + \tilde{\mathbf{g}}n_{kk}\tilde{d} + \tilde{\mathbf{g}}\tilde{d}. \end{aligned} \tag{2.65}$$

The constrained solution is then the solution of

$$\tilde{\mathbf{N}}\mathbf{x} = \tilde{\mathbf{u}}. \tag{2.66}$$

From Equations (2.64) and (2.65), it can be seen that the k^{th} row and columns of the normal system are being zeroed and then replaced with the constraints. This is what is intuitively expected: a single constraint decreases the degrees of freedom in the adjustment by one, and the k^{th} parameter is completely determined by the constraint. Applying constraints using the above equations is attractive from an implementation standpoint, since the zeroing of the rows in the normal matrix can be done without forming the ancillary products in Equation (2.64). Multiple constraints can be accommodated by updating the normal matrix and vector once for each constraint.

Stochastic Constraints

Both of the previous techniques for applying constraints in a least squares solution worked with deterministic, "hard" constraints. Frequently, however, the constraining information isn't known with absolute certainty. Instead, the relationship between parameters has a random, stochastic, component. Such "soft" constraints can be accomodated in a least squares adjustment using parameter observations.

A set of parameter observations can be expressed by

$$\mathbf{Gx} = \mathbf{d} - \mathbf{v},\ E\{\mathbf{vv}^T\} = \mathbf{C_d} \tag{2.67}$$

where $\mathbf{C_d}$ is the covariance matrix for the set of parameter constraint. This equation has exactly the same form and nature as the non-constraint observation equation; consequently, parameter observations are included in adjustment just like any observation. Since the constraints are independent of the other observations, the complete normal system is

$$\mathbf{A}^T\mathbf{C_l}^{-1}\mathbf{A} + \mathbf{G}^T\mathbf{C_d}^{-1}\mathbf{G} = \mathbf{A}^T\mathbf{C_l}^{-1}\mathbf{l} + \mathbf{G}^T\mathbf{C_d}^{-1}\mathbf{d} \tag{2.68}$$

If the constraints are pure parameter equalities, as is frequently the case, then this reduces to

$$\mathbf{A}^T\mathbf{C_l}^{-1}\mathbf{A} + \mathbf{C_d}^{-1} = \mathbf{A}^T\mathbf{C_l}^{-1}\mathbf{l} + \mathbf{C_d}^{-1}\mathbf{d} \tag{2.69}$$

Often, the exact weight of the stochastic constraints is unimportant. Very loose constraints can be used to prevent divergence in an iterative adjustment or to minimally define the datum without unduly affecting the solution. Alternatively, very tight constraints can effectively serve as deterministic constraints. However, care must be taken in adding parameter observations with too-high weight, lest the normal matrix become badly scaled and ill-conditioned.

2.2 Kalman Filtering

The Kalman filter is the technique used almost universally in geomatics for estimating kinematic or dynamic systems. Its widespread use is a consequence of its recursive nature that lends itself well to software implementation, and its well-proven applicability to many estimation problems in geomatics, even when the assumptions behind its operation are not exactly satisfied.

Kalman filtering is introduced and derived below. Many of the quantities used in the Kalman filtering equations have direct analogues in least squares; however, an alternate notation is commonly used. Below, the notation of Brown and Hwang (1997) is used. In this notation, the subscript k is used, for example, to denote a quantity at time t_k.

2.2.1 Derivation

The derivation of the Kalman filtering equations begins with a number of assumptions or conditions. First, it is assumed that a discrete random process exists whose behaviour can be modelled by

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k,\ E\{\mathbf{w}_k\} = \mathbf{0}\,\forall\, k,\ E\{\mathbf{w}_k \mathbf{w}_j^T\} = \begin{cases} \mathbf{Q}_k & j = k \\ 0 & j \neq k \end{cases} \tag{2.70}$$

In this model $\mathbf{x}$ are the parameters of the process, collectively known as the *state vector*. The parameters at two epochs are deterministically related through the *transition matrix* $\mathbf{\Phi}$ and stochastically related through the *process noise* $\mathbf{w}$. At discrete intervals, the process parameters can be related to observations by

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k,\ E\{\mathbf{v}_k\} = \mathbf{0}\,\forall\, k,\ E\{\mathbf{v}_k \mathbf{v}_j^T\} = \begin{cases} \mathbf{R}_k & j = k \\ 0 & j \neq k \end{cases} \tag{2.71}$$

Furthermore, it is assumed that when these measurements occur they can be used to improve the current estimate of the parameters by combining both according to

$$\hat{\mathbf{x}}_k^+ = \tilde{\mathbf{K}}_k \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{z}_k. \tag{2.72}$$

The as-yet undetermined coefficient matrices $\tilde{\mathbf{K}}_k$ and $\mathbf{K}_k$ govern how the the measurements and parameters are combined into updated parameter estimates. The state vectors before

and after the updated, $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k^+$, respectively, are "capped" to denote that they are estimates. These estimates differ from the true values by

$$\mathbf{e}_k^- = \hat{\mathbf{x}}_k^- - \mathbf{x}_k \tag{2.73}$$

$$\mathbf{e}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k. \tag{2.74}$$

Expanding Equation (2.74) using Equations (2.71) and (2.73) produces another expression for the error in the updated estimate,

$$\begin{aligned}\mathbf{e}_k^+ &= \tilde{\mathbf{K}_k}\hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{z}_k - \mathbf{x}_k \\ &= \tilde{\mathbf{K}_k}\left(\mathbf{e}_k^- + \mathbf{x}_k\right) + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\right) - \mathbf{x}_k \\ &= \left(\tilde{\mathbf{K}_k} + \mathbf{K}_k\mathbf{H}_k - \mathbf{I}\right)\mathbf{x}_k + \tilde{\mathbf{K}_k}\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k. \end{aligned} \tag{2.75}$$

Since $E\{\mathbf{v}_k\} = \mathbf{0}$ and $E\{\mathbf{e}_k^-\} = \mathbf{0}$, an unbiased estimate, $E\{\mathbf{e}_k^+\} = \mathbf{0}$, can only exist if

$$\tilde{\mathbf{K}_k} = \mathbf{I} - \mathbf{K}_k\mathbf{H}_k \tag{2.76}$$

Thus, the updated parameters are given by

$$\begin{aligned}\hat{\mathbf{x}}_k^+ &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{z}_k \\ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-\right) \end{aligned} \tag{2.77}$$

By comparison with Equation (2.71), it can be seen that difference term in this equation is a predicted residual, often termed the *innovation*, that is the difference between the actual and predicted observations,

$$\begin{aligned}\hat{\mathbf{v}}_k^- &= \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^- \\ &= \mathbf{z}_k - \hat{\mathbf{z}}_k^- \end{aligned} \tag{2.78}$$

making Equation (2.77)

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \hat{\mathbf{v}}_k^- \tag{2.79}$$

This relationship makes it clear that the Kalman filter works by using corrections to the predicted observations to correct the predicted state vector. The influence of the observational corrections is determined by the Kalman gain, $\mathbf{K}_k$. Rearranging this equation leads to another interpretation of the Kalman gain as a mapping of errors in observational space to errors in parameter space,

$$\hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^- = \mathbf{K}_k \left(\mathbf{z}_k - \hat{\mathbf{z}}_k^- \right). \tag{2.80}$$

The covariance of the updated parameters (or, more precisely, of their errors), is given by

$$\mathbf{P}_k^+ = E\left\{ \hat{\mathbf{e}}_k^+ \hat{\mathbf{e}}_k^{+T} \right\} = E\left\{ \left(\hat{\mathbf{x}}_k^+ - \mathbf{x}_k \right) \left(\hat{\mathbf{x}}_k^+ - \mathbf{x}_k \right)^T \right\}. \tag{2.81}$$

The error term can be expanded

$$\begin{aligned}
\hat{\mathbf{x}}_k^+ - \mathbf{x}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right) - \mathbf{x}_k \\
&= \left(\hat{\mathbf{x}}_k^- - \mathbf{x}_k \right) + \mathbf{K}_k \left(\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right) \\
&= \left(\hat{\mathbf{x}}_k^- - \mathbf{x}_k \right) - \mathbf{K}_k \mathbf{H}_k \left(\hat{\mathbf{x}}_k^- - \mathbf{x}_k \right) + \mathbf{K}_k \mathbf{v}_k \\
&= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{e}_k^- + \mathbf{K}_k \mathbf{v}_k
\end{aligned} \tag{2.82}$$

Thus,

$$\begin{aligned}
\mathbf{P}_k^+ &= E\left\{ \left(\hat{\mathbf{x}}_k^+ - \mathbf{x}_k \right) \left(\hat{\mathbf{x}}_k^+ - \mathbf{x}_k \right)^T \right\} \\
&= E\left\{ \left[\left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{e}_k^- + \mathbf{K}_k \mathbf{v}_k \right] \left[\left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{e}_k^- + \mathbf{K}_k \mathbf{v}_k \right]^T \right\} \\
&= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) E\left\{ \mathbf{e}_k^- \mathbf{e}_k^{-T} \right\} \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right)^T + \mathbf{K}_k E\left\{ \mathbf{v}_k \mathbf{v}_k^T \right\} \mathbf{K}_k^T \\
&+ \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) E\left\{ \mathbf{e}_k^- \mathbf{v}_k^T \right\} \mathbf{K}_k^T + \mathbf{K}_k E\left\{ \mathbf{v}_k \mathbf{e}_k^{-T} \right\} \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right)^T
\end{aligned}$$

Assuming that the measurement errors are uncorrelated with the estimation error, $E\left\{\mathbf{e}_k^- \mathbf{v}_k^T\right\} = E\left\{\mathbf{v}_k \mathbf{e}_k^{-T}\right\} = \mathbf{0}$, and observing that $E\left\{\mathbf{e}_k^- \mathbf{e}_k^{-T}\right\} = \mathbf{P}_k^-$ and $E\left\{\mathbf{v}_k \mathbf{e}_k^{-T}\right\} = \mathbf{R}_k$, allows this equation to be simplified to

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\, \mathbf{P}_k^- \,(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{2.83}$$

Equations (2.74) and (2.83) are general expressions for the updated parameters and their covariance. They apply for any gain that satisfies the assumptions made in their derivation. For a specific gain, however, a specific criteria of optimality is required. The criteria used in Kalman filtering is that the mean squared error in the estimated parameters is minimised. Equivalently, the error's mean squared sum is minimised,

$$f(\mathbf{e}) = \mathbf{e}^T \mathbf{e} = minimum. \tag{2.84}$$

This sum is equal to the trace of $\mathbf{P}_k^+$, and so this criteria is equivalent to minimising the variance of the updated parameters. The trace, in turn, is minimised when its derivative is set to zero,

$$\frac{\partial \,\mathrm{trace}(\mathbf{P}_k^+)}{\partial \mathbf{K}_k} = -2\,(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\, \mathbf{P}_k^- \mathbf{H}_k^T + 2\mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T = \mathbf{0} \tag{2.85}$$

Solving this equation for $\mathbf{K}_k$ yields the Kalman gain that produces a minimum mean square error estimate,

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{2.86}$$

Using this gain in Equations (2.74) and (2.83) produces the estimates for the parameters and their covariance updated by a set of measurements. The update equations require that the parameters and their covariance have been projected to the time of the measurement. The parameter projection is governed by Equation (2.70), ignoring the noise,

$$\hat{\mathbf{x}}_{k+1} = \mathbf{\Phi}_k \hat{\mathbf{x}}_k \tag{2.87}$$

The projected parameter covariance is

$$\begin{aligned}
\mathbf{P}_{k+1}^{-} &= E\left\{\hat{\mathbf{e}}_{k+1}^{-}{\hat{\mathbf{e}}_{k+1}^{-}}^{T}\right\} \\
&= E\left\{\left(\hat{\mathbf{x}}_{k+1}^{-} - \mathbf{x}_{k+1}\right)\left(\hat{\mathbf{x}}_{k+1}^{-} - \mathbf{x}_{k+1}\right)^{T}\right\} \\
&= E\left\{\left(\mathbf{\Phi}_k\hat{\mathbf{x}}_k^{-} - \mathbf{\Phi}_k\mathbf{x}_k + \mathbf{w}_k\right)\left(\mathbf{\Phi}_k\hat{\mathbf{x}}_k^{-} - \mathbf{\Phi}_k\mathbf{x}_k + \mathbf{w}_k\right)^{T}\right\}. \\
&= E\left\{\left(\mathbf{\Phi}_k\mathbf{e}_k - \mathbf{w}_k\right)\left(\mathbf{\Phi}_k\mathbf{e}_k - \mathbf{w}_k\right)^{T}\right\}. \\
&= \mathbf{\Phi}_k E\left\{\mathbf{e}_k\mathbf{e}_k^{T}\right\}\mathbf{\Phi}_k^{T} + \mathbf{\Phi}_k E\left\{\mathbf{e}_k\mathbf{w}_k^{T}\right\} + E\left\{\mathbf{w}_k\mathbf{e}_k^{T}\right\}\mathbf{\Phi}_k^{T} + E\left\{\mathbf{w}_k\mathbf{w}_k^{T}\right\}. \qquad (2.88)
\end{aligned}$$

If $E\{\mathbf{e}_k\mathbf{w}_k^T\} = \mathbf{0}$, then this reduces to

$$\mathbf{P}_{k+1}^{-} = \mathbf{\Phi}_k\mathbf{P}_k^{-}\mathbf{\Phi}_k^{T} + \mathbf{Q}_k. \qquad (2.89)$$

This derivation of the Kalman filtering equations closely follows that of Brown and Hwang (1997), except for the development for the updated parameters that mirrors that of Gelb (1974). Both texts are standard references in geomatics for Kalman filtering. A somewhat common modification to the derivation is to derive the gain by completing the square (Gao and Sideris, 2001).

It should be noted that the only criteria used above for deriving the Kalman filtering equations was that the summed mean square error was minimised. However, as with the least squares estimator, the Kalman filter can be derived by other techniques. In particular, if Gaussian distributions are assumed throughout then the derivation of the Kalman filter equations can also be approached as a maximum likelihood problem.

2.2.2 Summary

Kalman filtering requires two models:

1. A dynamic model, Equation (2.70)

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k\mathbf{x}_k + \mathbf{w}_k$$

2. An observation model, Equation (2.71)

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

Using these two models, the minimum mean error estimate can be found by recursively applying two sets of equations:

1. Measurement update
   (a) Compute the gain using Equation (2.86)

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}$$

   (b) Update the parameters using Equation (2.77)

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right)$$

   (c) Compute the parameter covariance using Equation (2.83)

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^- \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$$

   Alternately, this expression can be shortened to

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^- \tag{2.90}$$

   with the caveat that rounding and truncation errors may make result not symmetric.

2. Prediction
   (a) Predict parameters using Equation (2.87)

$$\hat{\mathbf{x}}_{k+1} = \mathbf{\Phi}_k \hat{\mathbf{x}}_k$$

   (b) Predict parameter covariance using Equation (2.89)

$$\mathbf{P}_k^- = \Phi_{k+1} \mathbf{P}_k \Phi_k^T + \mathbf{Q}_k$$

In practice, a Kalman filter is rarely implemented directly using the above equations. To improve the computational efficiency and reduce memory requirements, advantage is taken of matrix sparsity, especially during the prediction step where the transition matrix often has only one or two non-zero elements per row. Intermediate matrix products are also typically cached and re-used. For instance, if the $\mathbf{H}_k \mathbf{P}_k^-$ product is stored, then it can be

re-used in both the gain calculation,

$$\mathbf{K}_k = \left(\mathbf{H}_k \mathbf{P}_k^-\right)^T \left[\left(\mathbf{H}_k \mathbf{P}_k^-\right) \mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \tag{2.91}$$

and the covariance update,

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \left(\mathbf{H}_k \mathbf{P}_k^-\right). \tag{2.92}$$

The subtraction from $\mathbf{P}_k^-$ in Equation (2.92) can be done in-place, meaning storage need not be allocated for separate copies of the pre- and post-updated covariance matrices.

2.2.3 Non-linear observation equations

Like least-squares, Kalman filtering can also be adapted for non-linear observation equations,

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2.93}$$

Again, the observation model is linearised by a truncated Taylor series expansion. The point of expansion can either be some nominal state vector or the current estimate of the state vector. In the latter case, the filter is known as an *extended Kalman filter* and the linearised observation equation is

$$\mathbf{z}_k = \mathbf{h}\left(\hat{\mathbf{x}}_k^-\right) + \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\left(\hat{\mathbf{x}}_k^-\right)\left(\mathbf{x} - \hat{\mathbf{x}}_k^-\right) + \mathbf{v} \tag{2.94}$$

$$= \mathbf{H}_k \boldsymbol{\delta}_k^- + \left(\mathbf{h}\left(\hat{\mathbf{x}}_k^-\right) - \mathbf{v}_k^-\right) \tag{2.95}$$

Of course, the true state vector $\mathbf{x}$ is not known, and so it is assumed that the parameter corrections $\boldsymbol{\delta}_k^-$ are zero. Consequently, the predicted residual (or innovation) used to update the parameters by Equation (2.79) becomes

$$\hat{\mathbf{v}}_k^- = \mathbf{z}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^-\right). \tag{2.96}$$

and, as before, the updated parameters are

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \hat{\mathbf{v}}_k^-$$

Alternatively, the entire filter can work with the parameter corrections, in which case,

$$\hat{\boldsymbol{\delta}}_k^+ = \mathbf{K}_k \hat{\mathbf{v}}_k^-, \tag{2.97}$$

since, again, the predicted parameter corrections are zero. A filter whose state vector consists of parameter corrections is known as an error-state filter. It requires an external process that predicts the total states using the state transition model.

2.2.4 Random Process Models

Key terms in the Kalman filtering equations are the transition matrix and process noise covariance matrix. Together, they govern the behaviour of the discrete random process over time. These matrices are normally derived from a continuous random process model,

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{u}(t) \tag{2.98}$$

where, as before, $\mathbf{x}(t)$ is the state vector, while $\mathbf{u}(t)$ is a vector of random inputs known as *forcing functions*. For time-invariant systems – i.e., where $\mathbf{F}$ does not depend on the time – the transition matrix can be calculated from the continuous coefficient matrix by (Gelb, 1974)

$$\boldsymbol{\Phi}(t_k, t_{k+1}) = \exp\left(\mathbf{F}(t_{k+1} - t_k)\right) = \exp\left(\mathbf{F}(\Delta t)\right). \tag{2.99}$$

This function can be numerically evaluated by, for instance, approximating the exponent by a truncated power series expansion,

$$\exp\left(\mathbf{F}(\Delta t)\right) = \mathbf{I} + \mathbf{F}\Delta t + \frac{\mathbf{F}^2\Delta t^2}{2!} + \cdots + \frac{\mathbf{F}^i\Delta t^i}{i!} + \cdots \tag{2.100}$$

More often, however, the power series expansion is used to deduce an analytical expression for the transition matrix. Analytical expressions can also be found using Laplace transformations, recognising that

$$\exp\left(\mathbf{F}(\Delta t)\right) = \mathcal{L}^{-1}\left(s\mathbf{I} - \mathbf{F}\right)^{-1}. \tag{2.101}$$

For higher-dimension systems the analytical inversion of $(s\mathbf{I} - \mathbf{F})$ is not trivial; however, computer algebra systems such as Maple (Maplesoft, 2008) or Maxima (Maxima, 2008) can perform both the symbolic inversion and the inverse Laplace transform, greatly simplifying the calculation of the transition matrix.

Just as the transition matrix is derived from the former term in Equation (2.98), the process noise covariance matrix is derived from latter. To do so, the integral

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_k, \tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^T(\tau)\mathbf{\Phi}^T(t_k, \tau)d\tau \tag{2.102}$$

is evaluated, where $\mathbf{Q}(\tau)$ contains the spectral densities of the random inputs (Gelb, 1974). These spectral densities may be supplied by the manufacturer of a piece of equipment. Alternatively, they may have to be determined by analysing a random process over an extended period. In practice, however, the values of the process noise covariance matrix are often determined directly by tuning a Kalman filter.

The random process model for an entire system consists of some combination of single-state random process models. The three random process models most often encountered are the random constant, random walk, and first-order Gauss-Markov models.

Random Constant

A random constant is a variable whose rate-of-change is zero. Accordingly,

$$\dot{x}(t) = 0 \tag{2.103}$$

Intuitively, the equivalent discrete process is

$$x_{k+1} = x_k \tag{2.104}$$

Because the state transition relationship is purely deterministic, the process noise variance is zero,

$$q_k = 0 \tag{2.105}$$

Any parameter whose mean value is not expected to change can be modelled as random constant. For example, the co-ordinates of a fixed station.

Random Walk

Random walk occurs when a white noise signal is integrated,

$$x(t) = \int_0^t w(\tau) d\tau \tag{2.106}$$

Accordingly,

$$\dot{x}(t) = w(t) \tag{2.107}$$

If w has a Gaussian probability density function, then the process is termed as a Wiener or Brownian-motion process (Brown and Hwang, 1997). The discrete equivalent of Equation (2.107) is

$$x_{k+1} = x_k + \omega_k \tag{2.108}$$

The variance of the noise can be calculated using Equation (2.102) with unity $\mathbf{\Phi}$ and $\mathbf{G}$, assuming time-invariant noise

$$q_k = q\Delta t. \tag{2.109}$$

Figure 2.1 shows the behaviour of a random walk process for various process noise variances.

In many cases, a random walk model will be used where the dynamics actually call for

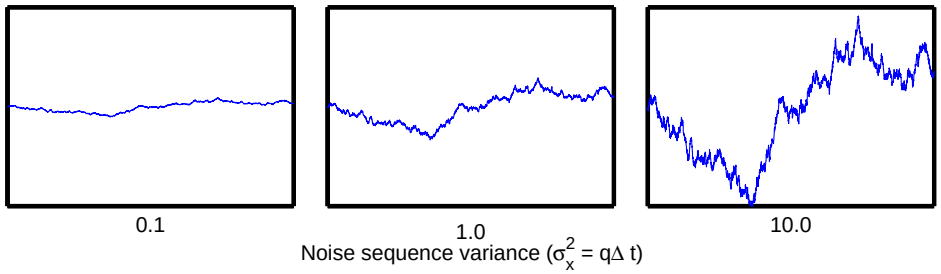


Figure 2.1: Random walk

a random constant. This is to avoid numerical problems for a long-running filter: with a random constant the covariance of the parameters will eventually approach zero, causing numerical problems (Brown and Hwang, 1997).

First-Order Gauss-Markov

A first-order markov process is described by the continuous differential equation

$$\dot{x}(t) = -\beta x(t) + w. \tag{2.110}$$

where β is the inverse of the correlation time. If w has a Gaussian probability density function, then the process is first-order Gauss-Markov. The discrete equivalent of Equation (2.110) follows from Equation (2.99),

$$x_{k+1} = e^{-\beta(t_{k+1}-t_k)}x(t_k) + w_k = e^{-\beta\Delta t}x(t_k) + w_k. \tag{2.111}$$

From Equation (2.102), the process noise variance is

$$q_k = \frac{q}{2\beta}\left[1 - \exp\left(-2\beta\Delta t\right)\right] \tag{2.112}$$

Figure 2.2 shows the behaviour of a first-order Gauss-Markov processes for various noise variances and correlation times. The last panel in this figure closely resembles the last panel in Figure 2.1, showing that a random walk process is basically first-order Gauss-Markov

process with an infinite correlation time: a conclusion that is also evident by examination of the equations describing the two processes.

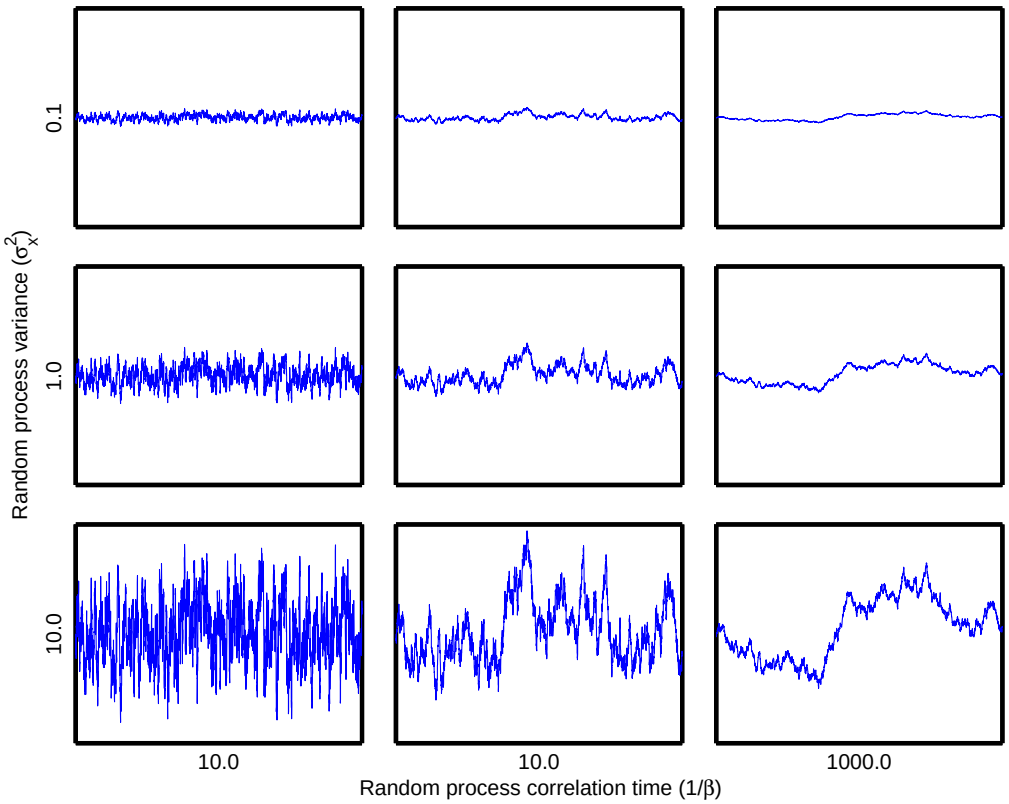


Figure 2.2: First-Order Gauss-Markov

Chapter 3

Satellite Positioning

As mentioned in the introduction, satellite positioning has become ubiquitous in aerial and terrestrial mobile mapping. Indeed, for newer remote-sensing technologies, like linear array or LIght Detection and Ranging (LIDAR) scanners, it is an enabling technology.

The aspects of GNSS introduced and reviewed here will be used when implementing the new GNSS/photogrammetric integration strategies described in the next chapter. Obviously, not all aspects of positioning using GNSS will be covered below; rather, a selection of topics critical or particularly important from an implementation standpoint will be covered. This includes:

- The observation equations: Measurement-level integration of GNSS and photogrammetric measurements will be done in the combined adjustment integration technique described in Chapter 5. Accordingly, the GNSS measurement equations are introduced here.

- Error mitigation: For high-accuracy GNSS positioning using a single receiver or for relative positioning over medium-to-long baselines, mitigation of various systematic observation errors is critical. The error mitigation techniques covered in this chapter will be used when implementing the GNSS parts of the new integration strategies.

- Ambiguity resolution: Resolving the full-cycle carrier-phase ambiguities is the key to obtaining the highest-accuracy GNSS positions, and one of the goals of the new integration strategies is to improve the ability to do ambiguity resolution. Hence, the theory and implementation of ambiguity resolution are covered below.

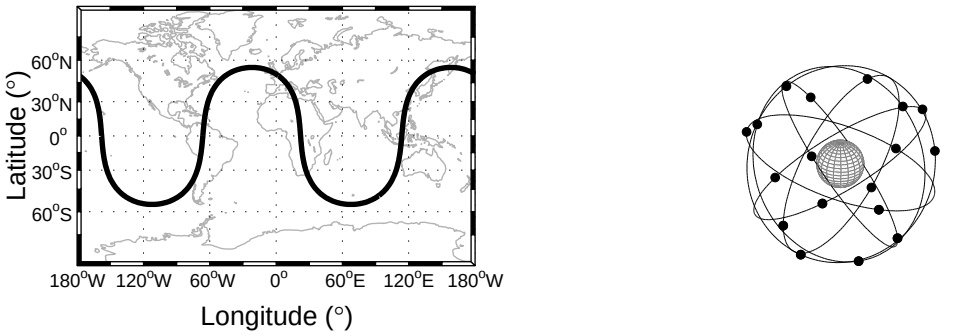
3.1 Systems Overview

GNSS infrastructure is commonly divided into three categories: the receivers, the satellites, and the monitoring stations, analysis centres, and upload facilities of the controlling organisations. These are commonly termed the user, space, and control segments, respectively. From the user's perspective, it is only the space segment that is obviously different, since modern receivers generally track and provide measurements from all systems and the operations of the control segment are generally hidden.

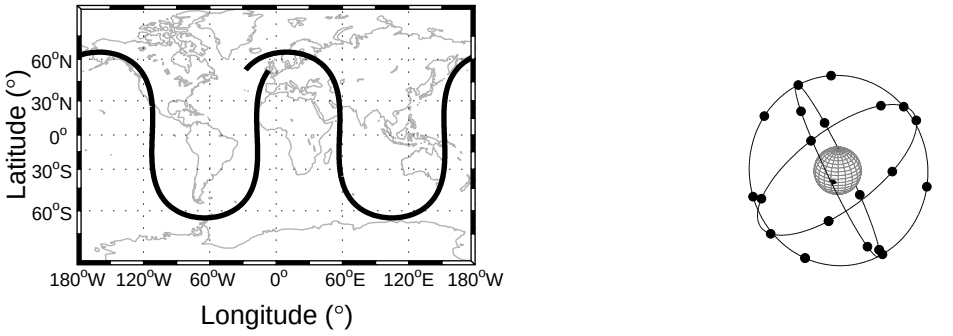
The space segments of the current three best-defined GNSS are briefly outlined in Table 3.1. The most relevant difference between the three GNSS is that the GLObal NAvigation Satellite System (GLONASS) orbits have a greater inclination than either GPS or Galileo. The effects of this higher inclination can be seen in Figure 3.1: GLONASS satellites have ground tracks that reach higher latitudes. Thus, GLONASS can provide better geometry for users closer to the poles.

Table 3.1: GNSS space segments

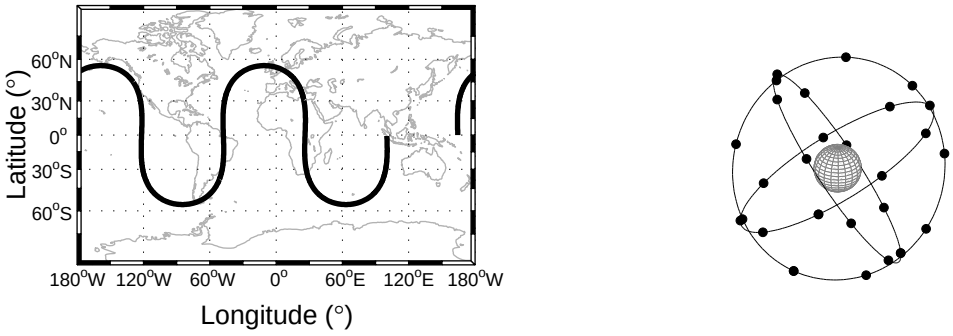
	GPS	GLONASS	Galileo
Satellites for full operation	24	21	27
Orbital radius	26 560 km	25 510 km	29 600 km
Orbital inclination	55°	64.8°	56°
Orbital planes	6	3	3
Orbital distribution	4 satellites/plane	7 satellites/plane	9 satellites/plane
Orbital period	11h 58m	11h 15m	14h 5m



(a) GPS



(b) GLONASS



(c) Galileo

Figure 3.1: GNSS space segments ground tracks and constellations

3.2 Observation Equations

Many types of information are available from GNSS receivers: from signal properties like correlation values, signal-to-noise or carrier-to-noise density ratios, and Doppler shifts, to data like satellite orbital parameters, ionospheric models, velocity, and positions. When viewed as a ranging measurement device, however, a receiver delivers two critical types of information: pseudoranges and carrier-phases. One or both of these information is always used in positioning, although there is considerable variation in how they are formulated into measurements suitable for use in Least Squares or Kalman filtering estimation.

3.2.1 Undifferenced observations

From the perspective of navigation and positioning algorithms, the fundamental observations output by a GNSS receiver are the undifferenced pseudorange and undifferenced carrier phase.

Pseudoranges

Essentially, a pseudorange observation is a measurement of a time interval. Specifically, the time it takes for a signal emitted by a GNSS satellite to be received by a GNSS receiver. Ideally, this time interval, together with the signal velocity, could be used to calculate the distance ρ between the satellite and the receiver using

$$\rho = c\left(t_{receive}^{GNSS} - t_{emit}^{GNSS}\right) \tag{3.1}$$

where c is the speed of light (i.e., the signal velocity), and t_{emit}^{GNSS} and $t_{receive}^{GNSS}$ are the emission and reception times, respectively. This equation, however, presupposes that both the emission and reception times are available in a common GNSS time frame. In reality the receiver measures the reception time using its own clock, and infers the emission time from information in the GNSS signal, constructed using the satellite's clock. These two clocks

are almost certainly not aligned. Instead, as depicted in Figure 3.2, the reception time $t^{rx}_{receive}$ measured by the receiver differs from the GNSS system time by the receiver clock offset $t_{GNSS/rx}$, and the transmission time broadcast by the satellite $t^{sv}_{transmit}$ differs by the satellite clock offset $t_{GNSS/sv}$,

$$\begin{aligned}\rho &= c\left[\left(t^{rx}_{receive} - t_{GNSS/rx}\right) - \left(t^{sv}_{emit} - t_{GNSS/sv}\right)\right] \\ &= c\left(t^{rx}_{receive} - t^{sv}_{emit}\right) - c\left(t_{GNSS/rx} - t_{GNSS/sv}\right). \end{aligned} \tag{3.2}$$

The interval $(t^{rx}_{receive} - t^{sv}_{emit})$ is what is actually measured by a GNSS receiver. This interval, scaled by the speed of light, is used to derive the pseudorange observation p,

$$p(t^{rx}_{receive}) = c\left(t^{rx}_{receive} - t^{sv}_{transmit}\right) = \rho + c\,t_{GNSS/rx} - c\,t_{GNSS/sv}. \tag{3.3}$$

It is important to note that the pseudorange observation is made within the receiver at intervals governed by the receiver clock, hence the addition of the $t^{rx}_{receive}$ in the equation above. The true reception time can, of course, be determined using the receiver clock offset, with the caveat that the accuracy of true time is limited by the accuracy of the clock offset.

The "pseudo" in the pseudorange observation name is in acknowledgment to the fact the range is not the true range from the satellite to receiver, and is instead a range biased by the receiver and satellite clock offsets. The pseudorange observation is also commonly called the code range measurement, in reference to the code message modulated on the transmitted signal from which the satellite's emission time is inferred.

The pseudorange observation equation is a parametric equation relating the pseudorange measurement p with the receiver and satellite positions, and the receiver and satellite clock offsets. Estimates of the satellite-related quantities, however, are transmitted in the GNSS signal, and typically only the receiver co-ordinates and clock offset are treated as unknowns in GNSS-related estimation. The connection between the receiver's position and the pseudorange measurement can be made more explicit by replacing the distance ρ with

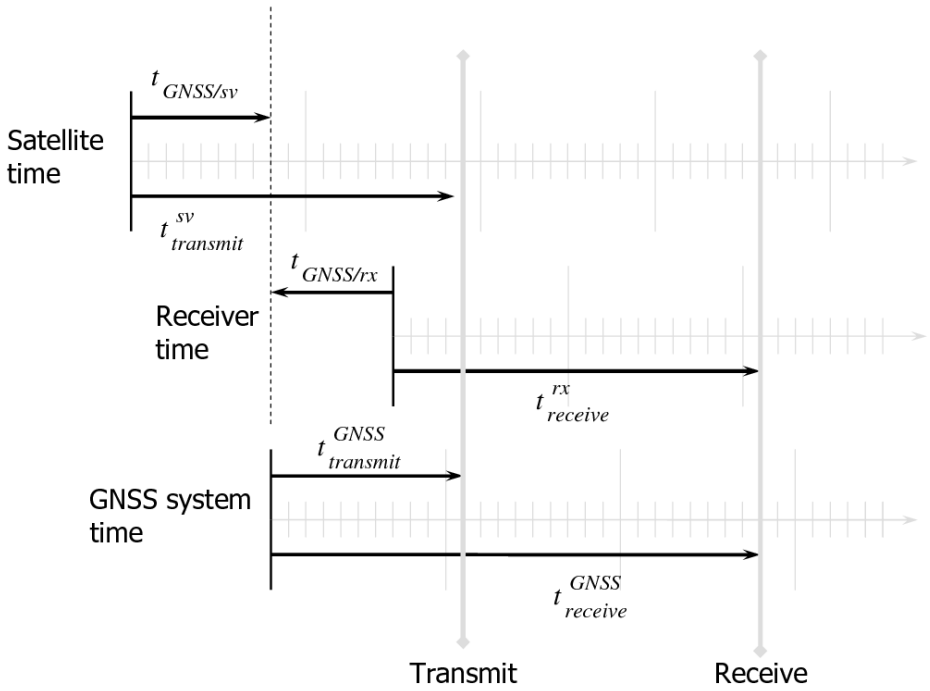


Figure 3.2: Relationships between satellite, receiver, and GNSS system time frames

the length of the vector connecting the satellite to the receiver,

$$p(t^{rx}_{receive}) = \left|\mathbf{r}_{rx/sv}\right| + c\,t_{GNSS/rx} - c\,t_{GNSS/sv} \tag{3.4}$$

where

$$\begin{aligned}\mathbf{r}_{rx/sv} &= \mathbf{r}_{sv} - \mathbf{r}_{rx} \\ &= \begin{pmatrix} x_{sv} \\ y_{sv} \\ z_{sv} \end{pmatrix} - \begin{pmatrix} x_{rx} \\ y_{rx} \\ z_{rx} \end{pmatrix}\end{aligned} \tag{3.5}$$

Carrier Phase

Besides the pseudorange, the carrier phase is the other observable made by GNSS receivers that is used for positioning. A carrier phase signal is a sinusoidal electromagnetic wave. A

cycle of such a signal occurs when the signal undergoes one complete oscillation, and the rate at which this occurs is the signal's frequency,

$$f = \frac{d\phi}{dt}. \tag{3.6}$$

In Equation (3.6), the carrier phase ϕ and frequency f have units of *cycles* and *cycles*$/s$, respectively. For a constant frequency, the integration of this relationship yields

$$\phi(t) = ft + \phi_0. \tag{3.7}$$

This is essentially a velocity relationship, where the linear unit is cycles instead of metres or other distance measure.

The carrier phase measured by a GNSS receiver is actually a beat phase. A beat phase is the difference between two phases. In this case, the phases are those of an incoming signal from a satellite and a replica generated by the receiver. Taken at the receiver's receive time $t^{rx}_{receive}$, the difference between these phases is

$$\phi(t^{rx}_{receive}) = \phi^{rx}(t^{rx}_{receive}) - \phi^{sv}(t^{rx}_{receive}). \tag{3.8}$$

The important property that decides the final carrier phase equation is that the phase of the received signal at $t^{rx}_{receive}$ is equivalent to the phase emitted by the satellite at its clock time (Leick, 1995),

$$\phi^{sv}(t^{rx}_{receive}) = \phi^{sv}(t^{sv}_{emit}). \tag{3.9}$$

So,

$$\phi(t^{rx}_{receive}) = \phi^{rx}(t^{rx}_{receive}) - \phi^{sv}(t^{sv}_{emit}). \tag{3.10}$$

Expanding the latter two terms of this equation using Equation (3.7) yields

$$\begin{aligned}\phi(t^{rx}_{receive}) &= (\phi^{rx}_o + f t^{rx}_{receive}) - (\phi^{sv}_o + f t^{sv}_{emit}) \\ &= f\,(t^{rx}_{receive} - t^{sv}_{emit}) + \phi^{rx}_o - \phi^{sv}_o \end{aligned} \tag{3.11}$$

From Equation (3.3), the time-difference is

$$t^{rx}_{receive} - t^{sv}_{emit} = \frac{\rho}{c} + t_{GNSS/rx} - t_{GNSS/sv} \tag{3.12}$$

that, when substituted in Equation (3.11), results in

$$\phi(t^{rx}_{receive}) = f\frac{\rho}{c} + f\,t_{GNSS/rx} - f\,t_{GNSS/sv} + \phi^{rx}_o - \phi^{sv}_o \tag{3.13}$$

One last, but important, refinement is necessary to arrive at a final, usable, carrier phase observation equation. The receiver can measure the instantaneous, fractional, part of the phase. It can also count the number of full-cycle phases as they change over time. However, it cannot determine the ϕ^{sv}_o integration constant in Equation (3.13). It must also assign some arbitrary value to its own integration constant, ϕ^{rx}_o. The net effect is that the phase measurement output by the receiver will include an ambiguous number of full-cycle beat phases. The integration constants can be accounted for by replacing them with a single ambiguity term, N,

$$\phi(t^{rx}_{receive}) = f\frac{\rho}{c} + f\,t_{GNSS/rx} - f\,t_{GNSS/sv} + N. \tag{3.14}$$

N, like ϕ, is in units of *cycles*. The ambiguity is technically the beat phase ambiguity, but for conciseness it will in the future just be termed the carrier phase ambiguity. The ambiguity is frequently conceptualised as being the number of full-cycle phases between the receiver and the satellite; however, this is only true under an idealised set of conditions

(Blewitt, 1997).

As with the pseudorange equation, the relationship between the unknown receiver coordinates and the carrier phase observation can be made more explicit by replacing ρ with $\left|\mathbf{r}_{rx/sv}\right|$,

$$\phi = \frac{f}{c}\left|\mathbf{r}_{rx/sv}\right| + f\,t_{GNSS/rx} - f\,t_{GNSS/sv} + N \tag{3.15}$$

It is, perhaps, more intuitive to use wavelength instead of frequency, in which case

$$\phi = \lambda^{-1}\left|\mathbf{r}_{rx/sv}\right| + \lambda^{-1}c\,t_{GNSS/rx} - \lambda^{-1}c\,t_{GNSS/sv} + N \tag{3.16}$$

By scaling using the wavelength, the undifferenced carrier phase observation can also be expressed in metres,

$$\Phi = \left|\mathbf{r}_{rx/sv}\right| + c\,t_{GNSS/rx} - c\,t_{GNSS/sv} + \lambda N. \tag{3.17}$$

Although it may seem more natural to express the observation in metres, such a form is disadvantaged by the ambiguity term, λN, losing its integer nature. As will be detailed later, recovering the integer ambiguity is the key to extremely accurate GNSS positioning.

It was noted above that GNSS receivers can count the number of full-cycle phases as they change over time: indeed, the carrier phase observable is frequently termed the Accumulated Delta (or Doppler) Range (ADR) (Spilker, 1996a). The relative nature of the phase measurements is, in fact, what makes them useful. Were a new ambiguity required for each observation, the number of unknowns in an adjustment involving carrier phases would never be less than the number of observations. The change in carrier phase is due to the Doppler effect that, in turn, results from satellite and receiver motion. Apparent Doppler is also caused by variations in the satellite and receiver oscillators or changes in signal path refractivity.

The preceding derivation of the carrier phase undifferenced observation equation may, at first glance, appear to be overly complex, and is invariably omitted from reviews of the

GNSS observation equations. However, such a development is necessary if the source of the receiver and satellite clock offsets in the equation are to be shown. The presence of the clock terms is important because only with them can the carrier phase equation be treated equivalently to the code range observation equation.

3.2.2 Single-difference Observations

The undifferenced code and carrier phase ranges are the fundamental positioning measurements available from GNSS receivers. By differencing like-types of these measurements, however, a number of derived measurements can be created. The attractiveness of these derived measurements is in the cancellation of the satellite and receiver clock biases, or the mollification of common atmospheric and other error sources.

Differencing measurements made to two different satellites by a common receiver results in the *between-satellite single difference*. Denoting the reference satellite with b (for base), the between-satellite single difference code range observation to the i^{th} satellite is,

$$\begin{aligned} p^{b/i} &= \left(\left|\mathbf{r}_{rx/i}\right| + c\,t_{GNSS/rx} - c\,t_{GNSS/i}\right) - \left(\left|\mathbf{r}_{rx/b}\right| + c\,t_{GNSS/rx} - c\,t_{GNSS/b}\right) \\ &= \left(\left|\mathbf{r}_{rx/i}\right| - \left|\mathbf{r}_{rx/b}\right|\right) - c\left(t_{GNSS/i} - t_{GNSS/b}\right) \\ &= \left(\left|\mathbf{r}_{rx/i}\right| - \left|\mathbf{r}_{rx/b}\right|\right) - c\,t_{b/i}. \end{aligned} \tag{3.18}$$

In this difference, the common receiver clock bias cancels out. The same also occurs when the carrier-phases expressed in metres are differenced,

$$\begin{aligned} \Phi^{b/i} &= \left(\left|\mathbf{r}_{rx/i}\right| + c\,t_{GNSS/rx} - c\,t_{GNSS/i} + \lambda^i N^i\right) \\ &\quad - \left(\left|\mathbf{r}_{rx/b}\right| + c\,t_{GNSS/rx} - c\,t_{GNSS/b} + \lambda^b N^b\right) \\ &= \left|\mathbf{r}_{rx/i}\right| - \left|\mathbf{r}_{rx/b}\right| - c\left(t_{GNSS/i} + t_{GNSS/b}\right) + \lambda^i N^i - \lambda^i N^b \\ &= \left|\mathbf{r}_{rx/i}\right| - \left|\mathbf{r}_{rx/b}\right| - c\,t_{b/i} + \lambda^b N^i - \lambda^i N^b \end{aligned} \tag{3.19}$$

Unfortunately, when the carrier phases are expressed in cycles, the receiver clock will only cancel out when both signals are at the same frequency. In general, then, it remains in the equation,

$$\begin{aligned}\phi^{b/i} &= \left(\frac{f^i}{c}\left|\mathbf{r}_{rx/i}\right| + f^i\, t_{GNSS/rx} - f^i\, t_{GNSS/i} + \phi_o^{rx} - \phi_o^i + N^i\right) \\ &- \left(\frac{f^b}{c}\left|\mathbf{r}_{rx/b}\right| + f^b\, t_{GNSS/rx} - f^b\, t_{GNSS/b} + \phi_o^{rx} - \phi_o^b + N^b\right) \\ &= \left(\frac{f^i}{c}\left|\mathbf{r}_{rx/i}\right| - \frac{f^b}{c}\left|\mathbf{r}_{rx/b}\right|\right) + (f^i - f^b)t_{GNSS/rx} \\ &- f^i\, t_{GNSS/i} + f^b\, t_{GNSS/b} + N^{b/i}. \end{aligned} \tag{3.20}$$

In practice, the between-satellite single difference is rarely used except as an intermediate stage in forming double-differences, described in Section 3.2.3, below. The receiver coordinates are the only remaining unknown parameters in Equation (3.18), and so between-satellite single differenced pseudoranges can also be used for single-point positioning.

A more useful single difference is the *between-receiver single difference*. As suggested by its name, this difference is formed by differencing measurements made at two receivers to a common satellite. For the pseudoranges, the resulting single difference is

$$\begin{aligned}\Delta p_{m/r} &= \left(\left|\mathbf{r}_{r/sv}\right| + c\, t_{GNSS/r} - c\, t_{GNSS/sv}\right) - \left(\left|\mathbf{r}_{m/sv}\right| + c\, t_{GNSS/m} - c\, t_{GNSS/sv}\right) \\ &= \left(\left|\mathbf{r}_{r/sv}\right| - \left|\mathbf{r}_{m/sv}\right|\right) - c\left(t_{GNSS/r} - t_{GNSS/m}\right) \\ &= \left(\left|\mathbf{r}_{r/sv}\right| - \left|\mathbf{r}_{m/sv}\right|\right) - c\, t_{m/r}. \end{aligned} \tag{3.21}$$

The clock bias of the satellite common to both observations is removed by this difference.

The same bias is also eliminated when carrier-phases expressed in metres are differenced,

$$\begin{aligned}\Delta\Phi_{m/r} &= \left(\left|\mathbf{r}_{r/sv}\right| + c\,t_{GNSS/r} - c\,t_{GNSS/sv} + \lambda N_m\right) \\ &\quad - \left(\left|\mathbf{r}_{m/sv}\right| + c\,t_{GNSS/m} - c\,t_{GNSS/sv} + \lambda N_r\right) \\ &= \left|\mathbf{r}_{r/sv}\right| - \left|\mathbf{r}_{m/sv}\right| + c\left(t_{GNSS/r} - t_{GNSS/m}\right) + \lambda\left(N_r - N_m\right) \\ &= \left|\mathbf{r}_{r/sv}\right| - \left|\mathbf{r}_{m/sv}\right| + c\,t_{m/r} + \lambda N_{m/r}, \end{aligned} \tag{3.22}$$

or when carrier-phases expressed in cycles are differenced,

$$\begin{aligned}\Delta\phi_{m/r} &= \left(\frac{f}{c}\left|\mathbf{r}_{r/sv}\right| + f\,t_{GNSS/r} - f\,t_{GNSS/sv} + \phi_o^{rx} - \phi_o^{i} + N_r\right) \\ &\quad - \left(\frac{f}{c}\left|\mathbf{r}_{m/sv}\right| + f\,t_{GNSS/m} - f\,t_{GNSS/sv} + \phi_o^{rx} - \phi_o^{b} + N_m\right) \\ &= \frac{f}{c}\left(\left|\mathbf{r}_{rx/sv}\right| - \left|\mathbf{r}_{rx/sv}\right|\right) + f\left(t_{GNSS/r} - t_{GNSS/m}\right) + N_r - N_m \\ &= \frac{f}{c}\left(\left|\mathbf{r}_{rx/sv}\right| - \left|\mathbf{r}_{rx/sv}\right|\right) + f t_{m/r} + N_{m/r}. \end{aligned} \tag{3.23}$$

If the clocks of the two receivers in the single-difference can be absolutely synchronised – using, for example, a shared internal or external clock – then the $c\,t_{m/r}$ or $f\,t_{m/r}$ clock term can be removed. This approach is sometimes taken in GNSS attitude determination systems, where multiple antennas are connected to a single receiver. The between-receiver single difference pseudorange observations can be also used for relative positioning (see Cannon's paper on desktop at home).

3.2.3 Double-difference Observations

With the between-satellite single-difference removing the receiver clock bias and the between-receiver single difference removing the satellite clock bias, an obvious step is to difference the two single-differences into an observation that has both clock biases removed. The resulting observation is called the *double-difference*, and is the observation used almost exclusively

for high-accuracy positioning.

The double-difference can be formed in two ways: either two between-receiver single differences are themselves differenced, or two between-satellite single differences are differenced. For instance, with pseudoranges the former approach yields

$$\begin{aligned} p_{m/r}^{b/i} &= p_{m/r}^{i} - p_{m/r}^{b} \\ &= \left[\left(\left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{m/i}\right|\right) - c\,t_{m/r}\right] - \left[\left(\left|\mathbf{r}_{r/b}\right| - \left|\mathbf{r}_{m/b}\right|\right) - c\,t_{m/r}\right] \\ &= \left(\left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{m/i}\right|\right) - \left(\left|\mathbf{r}_{r/b}\right| - \left|\mathbf{r}_{m/b}\right|\right), \end{aligned} \tag{3.24}$$

and the latter approach,

$$\begin{aligned} p_{m/r}^{b/i} &= p_{r}^{b/i} - p_{m}^{b/i} \\ &= \left[\left(\left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{r/b}\right|\right) - c\,t_{b/i}\right] - \left[\left(\left|\mathbf{r}_{m/i}\right| - \left|\mathbf{r}_{m/b}\right|\right) - c\,t_{b/i}\right] \\ &= \left(\left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{r/b}\right|\right) - \left(\left|\mathbf{r}_{m/i}\right| - \left|\mathbf{r}_{m/b}\right|\right). \end{aligned} \tag{3.25}$$

Of course, the result from both approaches is the same, since

$$\begin{aligned} \left(\left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{r/b}\right|\right) - \left(\left|\mathbf{r}_{m/i}\right| - \left|\mathbf{r}_{m/b}\right|\right) &= \left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{m/i}\right| - \left|\mathbf{r}_{r/b}\right| + \left|\mathbf{r}_{m/b}\right| \\ &= \left(\left|\mathbf{r}_{r/i}\right| - \left|\mathbf{r}_{m/i}\right|\right) - \left(\left|\mathbf{r}_{r/b}\right| - \left|\mathbf{r}_{m/b}\right|\right). \end{aligned}$$

In practice there is a slight implementation advantage to using the between-receiver differences, as the $p_{m/r}^{b}$ term common to all double-differences for a set of satellites need only be calculated once.

The double-difference observable for the carrier phase is, like the single differences, complicated by unequal frequencies. When the carrier-phases are expressed in cycles, different

frequencies again mean that the single-difference receiver clock is not removed,

$$\begin{aligned}
\phi^{b/i}_{m/r} &= \phi^{i}_{m/r} - \phi^{b}_{m/r} \\
&= \left[\frac{f_i}{c}\Big(\big|\mathbf{r}_{r/i}\big| - \big|\mathbf{r}_{m/i}\big| \Big) + f_i t_{r/m} + N\right] \\
&- \left[\frac{f_b}{c}\Big(\big|\mathbf{r}_{r/b}\big| - \big|\mathbf{r}_{m/b}\big| \Big) + f_b t_{r/m} + N\right] \\
&= \frac{f_i}{c}\Big(\big|\mathbf{r}_{r/i}\big| - \big|\mathbf{r}_{m/i}\big| \Big) - \frac{f_b}{c}\Big(\big|\mathbf{r}_{r/b}\big| - \big|\mathbf{r}_{m/b}\big| \Big) + (f_i - f_b)\; t_{r/m} + N
\end{aligned} \tag{3.26}$$

The $(f_i - f_b)\; t_{r/m}$ term creates problems when using the double-differenced carrier phases for positioning. It can be mitigated using the relative receiver clock offset $t_{r/m}$ estimated from between-receiver single differences, or it can be estimated within an adjustment or Kalman filter. Any error in this estimate, however, will impact the double-difference positioning solution (Dai et al., 2001). Even more critically, errors in the relative clock offset will impact the resolution of the double-difference integer ambiguities.

When the carrier-phases are expressed in metres, a related problem arises: different frequencies result in a $\left(\lambda^i N^i_{m/r} - \lambda^b N^b_{m/r}\right)$ term that, in combination, is no longer an integer,

$$\begin{aligned}
\Phi^{b/i}_{m/r} &= \Phi^{i}_{m/r} - \Phi^{b}_{m/r} \\
&= \Big(\big|\mathbf{r}_{r/i}\big| - \big|\mathbf{r}_{m/i}\big| + c\, t_{m/r} + \lambda^i N^i_{m/r}\Big) \\
&- \Big(\big|\mathbf{r}_{r/b}\big| - \big|\mathbf{r}_{m/b}\big| + c\, t_{m/r} + \lambda^b N^b_{m/r}\Big) \\
&= \Big(\big|\mathbf{r}_{r/i}\big| - \big|\mathbf{r}_{m/i}\big| \Big) - \Big(\big|\mathbf{r}_{r/b}\big| - \big|\mathbf{r}_{m/b}\big| \Big) + \left(\lambda^i N^i_{m/r} - \lambda^b N^b_{m/r}\right).
\end{aligned} \tag{3.27}$$

Alternatively, the $\left(\lambda^i N^i_{m/r} - \lambda^b N^b_{m/r}\right)$ term can be factored into (Wang, 2000)

$$\begin{aligned}
\lambda^i N^i_{m/r} - \lambda^b N^b_{m/r} &= \lambda^i N^i_{m/r} - \lambda^i N^b_{m/r} - \lambda^b N^b_{m/r} + \lambda^i N^b_{m/r} \\
&= \lambda^i N^{b/i}_{m/r} + \left(\lambda^i - \lambda^b\right) N^b_{m/r}.
\end{aligned} \tag{3.28}$$

In Equation‘(3.28), the ambiguity $N_{m/r}^{b/i}$ is integer, but there is now a term involving $N_{m/r}^{b}$: the single-difference ambiguity for the base satellite. As with the clock term in Equation (3.26), the quality of both positioning and ambiguity resolution depends on how well this term is known.

In addition to removing the satellite and receiver clock biases, double-differencing also substantially removes several other errors on GNSS signals not considered above. These errors and double-differencing's mitigating effect on them will be covered in Section 3.3. The error-mitigating properties of the double-difference are the reason for its near-exclusive use in high-accuracy GNSS positioning.

3.2.4 Frequency Combinations

An entirely different category of data combinations to the inter-satellite and inter-receiver combinations just described are those that result when multiples of observations made between the same satellite and receiver are added or subtracted. Such combinations are most frequently formed using carrier phase measurements; these combinations are, consequently, termed linear phase combinations. The primary purpose of these combinations is to alter the wavelength of the carrier phase observable to either make ambiguity resolution easier or increase the accuracy of the observable.

A linear phase combination ϕ_c is formed when two phase measurements ϕ_{f1} and ϕ_{f2} are combined by

$$\phi_c = a\,\phi_{f_1} + b\,\phi_{f_2}, \tag{3.29}$$

where a and b are the coefficients of the combination. The frequency of the combination, is, similarly

$$f_c = a\,f_1 + b\,f_2, \tag{3.30}$$

which leads to a wavelength of

$$\lambda_c = \frac{c}{f_c} = \frac{\lambda_1 \lambda_2}{a\,\lambda_2 + b\,\lambda_1}. \tag{3.31}$$

From Equation (3.29) it is apparent that any linear combination will always have a higher noise in cycles than an individual measurement; however, the same may not be true for the noise in distance units. Assuming uncorrelated phase measurements and the same noise on each frequency, the variance, in cycles, of the combination will be

$$\sigma^2_{\phi_c} = (a^2 + b^2)\sigma^2_{\phi_{f_1}}. \tag{3.32}$$

while the the variance in distance units is

$$\begin{aligned} \sigma^2_{\Phi_c} &= \frac{\lambda_c^2\left(a^2 + b^2\right)}{\lambda_1^2}\sigma^2_{\Phi_{f_1}} \\ &= \frac{\lambda_2^2\left(a^2 + b^2\right)}{\left(a\,\lambda_2 + b\,\lambda_1\right)^2}\sigma^2_{\Phi_{f_1}}. \end{aligned} \tag{3.33}$$

The noise of the combination will be greater or less than f1's depending on the coefficient ratio in the above equation. Consider, for example, the two simplest combinations: those where $a = 1$ and $b = 1$ or $a = 1$ and $b = -1$. With GPS L1-L2, these combinations have wavelengths of 10.7 cm and 86.2 cm, respectively. The corresponding distance variances are $0.6\,\sigma^2_{\Phi_{f1}}$ and $41.0\,\sigma^2_{\Phi_{f1}}$. These variances seem intuitively correct: the combination with a wavelength shorter than L1 has a distance variance less than L1's, and vice versa. However, consider also the combination $a = 77$ and $b = -60$. Here, intuition fails, as the combination has a wavelength of only about 0.6 cm (or $\approx 0.3\lambda_1$), but has a distance variance of $\sigma^2_{\Phi_c} = 10.4\sigma^2_{\Phi_{f1}}$; the wavelength of this combination is even smaller than with $a = 1$ and $b = 1$, yet its noise is an order of magnitude greater than L1's

The three combinations just described are the most common and important used in GPS. The first two combinations, where $a = 1$ and $b = 1$ or $a = 1$ and $b = -1$, are termed the

narrow-lane and wide-lane, respectively. The names arise because in the former case the wavelength of the resulting combination is smaller (i.e., narrower) than either of the two input signals, and in the latter case the wavelength is longer (i.e., wider). The narrow-lane's shorter wavelength makes ambiguity resolution more challenging, but once the ambiguities are resolved the measurement's lower noise makes the measurement more accurate than a single-frequency measurement. Conversely, the wide-lane's longer wavelength facilitates ambiguity resolution, but the measurement is noisier. The third combination, with $a = 77$ and $b = -60$, is the ionospheric-free linear combination: so called because it is virtually free of ionospheric delay. Other GNSS have similar ionosphere-eliminating combinations.

There has been a moderate amount of study into linear combinations other than the narrow-lane, wide-lane, and ionospheric-free combination, cf. Cocard and Geiger (1994); Han and Rizos (1996a); Collins (1999); Radovanovic et al. (2001). In practice, however, none of these other combinations have found wide usage.

3.2.5 Other differences and data combinations

In addition to the single-differences, double-differences, and frequency combinations derived in the previous sections, there are many other data combinations that can be formed using the basic undifferenced pseudorange and carrier phase observables. For instance, differences can be formed across time. The most common example of this is the triple-difference that results when double-differences from difference epochs are differenced. Also, combinations can be formed between pseudoranges and carrier phases. A common use of such inter-observation-type combinations is in carrier-phase smoothing where the much more accurate carrier-phase is used to modulate the noise on the pseudorange observation (Hatch, 1982).

3.2.6 Combinations of data combinations

The data combinations described above are themselves often combined. For example, narrow-lane carrier phase observations are invariably double-differenced. Typically, these

combinations are formed sequentially: the double-difference narrowlane observation, for instance, can be formed by first calculating the double-difference on each frequency using Equation (3.26), and then using those double differences in Equation (3.29).

As linear operations, the data combinations can also be expressed in matrix-vector form. This form is convenient for expressing combinations of data combinations in a single equation, and for determining the covariance of the final data combination. For example, if $\mathbf{l}$ contains the undifferenced carrier phases observed at both stations, then the double-difference narrowlane carrier phase observations can be calculated from

$$\phi_c = \mathbf{A}_c \boldsymbol{\nabla} \boldsymbol{\Delta} \mathbf{l} \tag{3.34}$$

where $\boldsymbol{\Delta}$, $\boldsymbol{\nabla}$, and $\mathbf{A}_c$ are, respectively, coefficient matrices that perform a single-difference, double-difference, and narrowlane linear combination. Continuing the example, if the observation vector $\mathbf{l}$ consists of 2 commonly observed dual-frequency carrier phases,

$$\mathbf{l} = \begin{pmatrix} \phi^i_{mf1} & \phi^i_{mf2} & \phi^b_{mf1} & \phi^b_{mf2} & \phi^i_{rf1} & \phi^i_{rf2} & \phi^b_{rf1} & \phi^b_{rf2} \end{pmatrix}^T$$

then the coefficient matrices would be

$$\boldsymbol{\Delta} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix},$$

$$\boldsymbol{\nabla} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix},$$

and

$$\mathbf{A}_c = \begin{pmatrix} 1 & -1 \end{pmatrix}.$$

Together, then, the total coefficient matrix is

$$\begin{aligned}\mathbf{A} &= \mathbf{A}_c \boldsymbol{\nabla} \boldsymbol{\Delta} \\ &= \begin{pmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}\end{aligned}$$

This example shows how for one usable double-differenced narrow-lane carrier phase observable, *eight* fundamental undifferenced carrier phase measurements are required.

3.3 Errors and their mitigation

The undifferenced observation equations given in Section 3.2.1 are for errorless GNSS measurements. In reality, of course, the measurements are influenced by a host of random and systematic errors. Considering the systematic errors that have the largest impact, the code range observation equation would more completely be expressed by

$$p = \left| \mathbf{r}_{rx/sv^b} + \mathbf{r}_{sv^b/sv} \right| + c\, t_{GNSS/rx} - c \left(t_{GNSS/sv^b} + t_{sv^b/sv} \right) \tag{3.35}$$

$$+ T(\mathbf{r}_{rx}, \mathbf{r}_{sv}) + I(\mathbf{r}_{rx}, \mathbf{r}_{sv}, f) + b(f, k). \tag{3.36}$$

The additional terms in this equation are for the following errors:

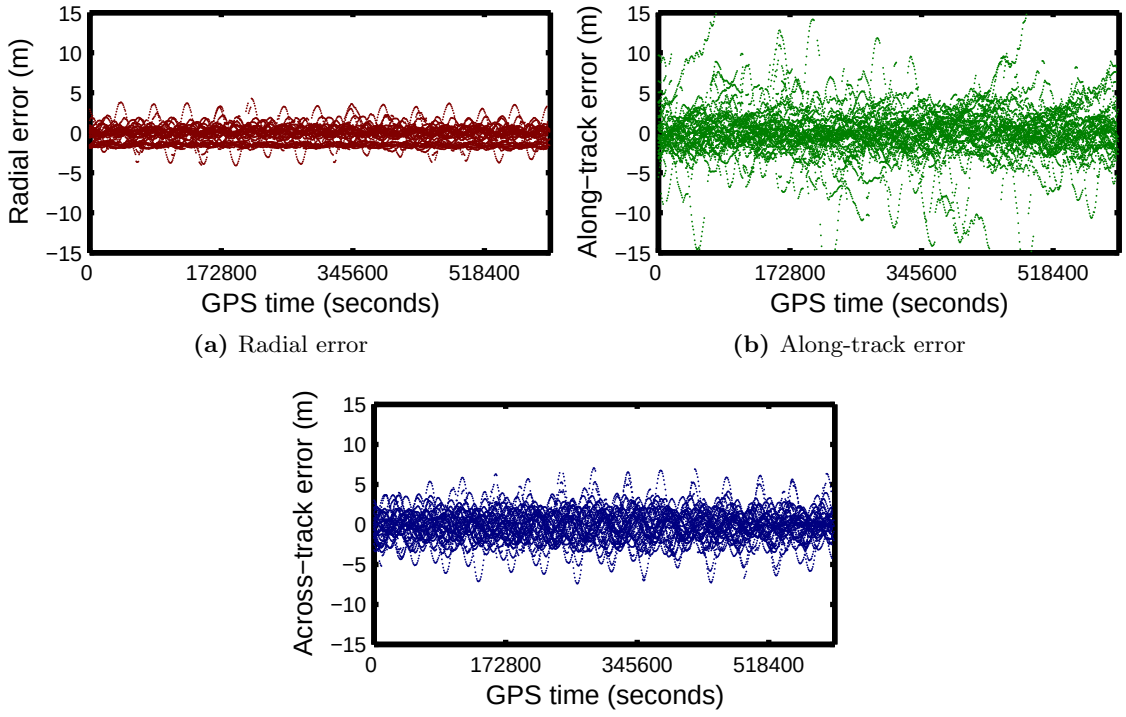
- Errors in the broadcast satellite positions and clock biases: $\mathbf{r}_{sv^b/sv}$ and $t_{sv^b/sv}$, respectively.
- Tropospheric delays that depend upon the path the satellite signal takes when travelling through the troposphere, and hence on the satellite and receiver positions: $T(\mathbf{r}_{rx}, \mathbf{r}_{sv})$.
- A frequency dependent ionospheric delay, also dependent upon the signal path: $I(\mathbf{r}_{rx}, \mathbf{r}_{sv}, f)$.
- A frequency and channel dependent hardware bias: $b(f, k)$.

These errors have the largest impact on the measurements, are those most often considered in GNSS texts, and, correspondingly, are those considered here.

3.3.1 Satellite position errors

The orbital parameters broadcast by GNSS satellites are extrapolated based upon data collected from a limited number of ground stations belonging to the controlling institutions. Because of the extrapolation and limited data used, these orbits contain errors, and so, consequently, do the satellite positions calculated from them. Figures 3.3a through 3.3c, for example, show the radial, along-track, and across-track satellite position errors for the entire GPS constellation during a one-week period in May of 2003. The radial error, is the most significant for positioning, was the smallest component of this total error, with a Root Mean Square (RMS) error of just over 1.1 metres. Peaks of nearly 5 metres, however, can be seen to regularly occur.

In relative positioning the impact of satellite position errors is quite small, as the projection of the error onto the measurements to each receiver is – for a reasonable receiver separation, at least – essentially equal. For undifferenced observations, however, the satellite position error will be a significant error source, both in absolute terms and relative to other errors. With undifferenced GNSS observations there are three options for dealing with the satellite position errors. First, the satellite positions could simply be weighted in the adjustment instead of being fixed. Second, measurement variance could be increased. However, perhaps the most obvious way to deal with the orbital errors is to practically eliminate them using precise ephemerides. Precise ephemerides are either observed or predicted orbits that are made available by a number of organisations including the United States' National Imagery and Mapping Agency (NIMA) and the International GNSS Service (IGS). In the case of the latter, several products are available, with accuracies ranging from 1.1 metres cm for predicted orbits to better than 5 cm for observed orbits with a two-week latency (IGS, 2008). Either accuracy is well below the expected noise level of the undifferenced



(a) Radial error

(b) Along-track error

(c) Across-track error

Figure 3.3: Satellite orbit and clock errors (Entire constellation, GPS week 1217)

pseudorange measurements.

To determine satellite positions between samples, polynomial interpolation is normally used (Hofmann-Wellenhof et al., 2001). Precise ephemerides typically have a sample interval of 15 minutes, and Table 3.2 shows that for satellite positions at this rate there is no benefit to be had from going above an 8th order polynomial interpolator. Data for the table was generated using precise ephemerides from NASA's Jet Propulsion Laboratory (JPL) that have a sample interval of 30 seconds (JPL, 2003).

3.3.2 Satellite clock errors

Unlike a satellite position error, a satellite clock bias error will manifest itself directly as a range error. Most of the satellite clock biases can be removed using correction coefficients broadcast as part of the satellite ephemeris. The residual error that remains, however, can

Table 3.2: Satellite position and clock error using polynomial interpolation and precise ephemerides and clocks with a sample interval of 15 minutes (entire constellation, GPS week 1217)

Order of	Position Error (m)		Clock Error (m)	
Interpolant	RMS	Maximum	RMS	Maximum
1	32357.75	52228.95	0.16	6.65
2	2557.98	4190.80	0.08	0.74
3	161.90	269.94	0.08	0.70
4	14.96	24.90	0.08	0.73
5	1.18	2.11	0.08	0.71
6	0.12	0.25	0.08	0.73
7	0.01	0.04	0.08	0.71
8	0.01	0.03	0.08	0.73
9	0.01	0.03	0.08	0.71
10	0.01	0.03	0.08	0.73

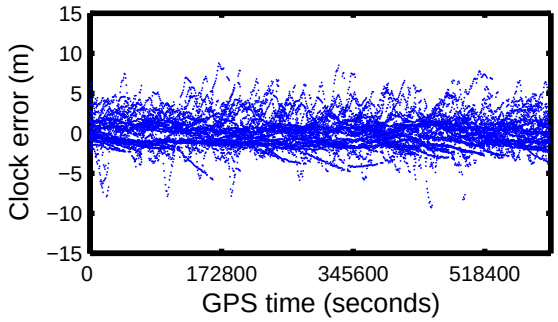


Figure 3.4: GPS satellite clock bias errors (Entire constellation, GPS week 1217)

still be significant. Figure 3.4 shows the difference between the broadcast and precise GPS satellite clock corrections. When compared with Figures 3.3a through 3.3c, it can be seen that the residual satellite clock error is at about the same level as the satellite orbit errors. For the one week period shown in the figure, the RMS clock error for the entire constellation was just under 2 m, and the maximum error was close to 10 m.

The residual satellite clock errors are only a consideration with undifferenced GNSS observations. As shown in Equations (3.18) and (3.24), the satellite clock bias cancels out entirely in between-satellite differenced and double-differenced observations; hence, so does the clock bias error. With undifferenced observations, precise clock corrections can be used

to virtually eliminate residual satellite clock bias errors. Such corrections are normally included with precise ephemerides, and, just as with precise ephemerides, polynomial interpolation can be used to determine the correction between sample epochs. Because the satellite clocks are very stable, a lower-order interpolator can be used than that for the positions. From Table 3.2, it can be seen that a second-order interpolator yields reasonable results when 15-minute clock corrections are available. Maximum errors of close to a metre, however, may still be cause for concern. Fortunately, clock corrections at a 5-minute sample interval are also available from the IGS. Table 3.3 shows that when these higher-rate corrections are used with a third-order interpolator, the maximum error can be reduced to under half a metre.

Table 3.3: Satellite clock error using polynomial interpolation and precise clocks with a sample interval of 5 minutes (entire constellation, GPS week 1217)

Order of Interpolant	RMS	Maximum
1	0.04	1.98
2	0.04	0.52
3	0.03	0.44
4	0.04	0.49
5	0.03	0.44

3.3.3 Troposphere

The error caused by the lower atmosphere of the earth – up to about 40 km altitude – on GNSS signals is commonly referred to as tropospheric delay. This error would, perhaps, be better referred to as the 'neutral atmosphere' error as it is caused by both the stratosphere and the troposphere, but the effect of the latter is much larger and hence the more common terminology. Regardless of its name the effect of the lower atmosphere is a delay or retardation in transmission of the GNSS signal from the satellite to the receiver. It effects both pseudoranges and ranges determined using the carrier phase equally, and, because the neutral atmosphere is non-dispersive at GNSS frequencies, it also has equal impact across

all signal frequencies.

The tropospheric delay results from the atmospheric gases having a greater refractive index than that of free space. This, in turn, causes the speed of light in the lower atmosphere to be less than its value in free space and consequently the GNSS signal takes longer to travel to the receiver than it otherwise would (Spilker, 1996b). Changes in the refractive index with varying height also result in an additional delay by causing the GNSS signals to bend as they pass through the atmosphere. This bending means that the signal travels farther than it otherwise would, and again the effect is a signal delay. The change in refractivity itself is commonly divided into two categories: the change resulting from the dry atmosphere (the 'dry part') and the change resulting from the water vapour in the atmosphere (the 'wet part'). The former causes about 90% of the change in refractivity while the latter causes about 10% (Langley, 1998a; Hofmann-Wellenhof et al., 2001).

The technique most often used to mitigate the tropospheric delay is to model it for a signal at zenith, and then apply a mapping function to relate the zenith delay to the delay at lower elevation angles. Many models and mapping functions have been developed, and a review of both can be found in Zhang (1999). They differ according to the following criteria:

- Whether the wet and dry effects are modelled separately.
- Whether surface measurements of temperature or pressure are used, or if empirically derived values bases upon parameters like latitude, height, or time-of-year are used.
- The numerical constants used in the model.

Figure 3.5 shows the estimates from two tropospheric models contrasted with the measured tropospheric delay for two GPS satellite and Figure 3.6 shows the corresponding model errors. The models used are the UNB2 and UNB3 which were used with the Niell mapping function (Collins et al., 2001; Niell, 1996). The model error was calculated by subtracting the measured range (corrected for the ionospheric delay using the filtered ionospheric-free estimate and for the receiver and satellite clock errors using precise corrections) from the

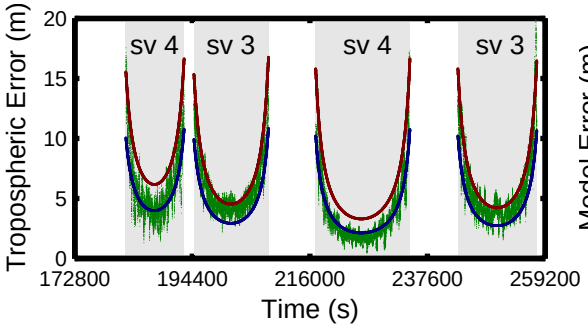


Figure 3.5: Derived tropospheric delays and delays calculated using the UNB2 and UNB3 tropospheric models (SVs 3 and 4, day 2 of GPS Week 1217)

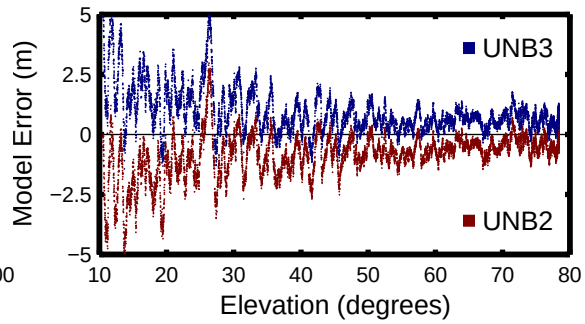


Figure 3.6: Error in tropospheric delays calculated using the UNB2 and UNB3 tropospheric models (SV 1, day 2 of GPS Week 1217)

true range (calculated using precise receiver and satellite co-ordinates). These figures show how the delay itself can cause a $2 - 25$ m error on a single range measurement (Brunner and Welsch, 1993). However, even a simple model should reduce the range error to about 1 m or better. The *residual tropospheric error* not removed by modelling can be handled by including a zenith delay as an additional parameter in a least-squares adjustments or Kalman Filter (Brunner and Welsch, 1993; Langley, 1998a). Note, however, that a constant bias shared by all satellites is equivalent to a additional receiver clock offset; in other words, it will be removed in single point positioning (described in Section 3.4.1).

The dry component of lower atmosphere itself is largely homogeneous, and the error it causes for the same satellite at two locations will frequently be highly correlated (the signal from the satellite will travel through sections of the troposphere with similar characteristics). Therefore, differencing between measurements made at different locations can eliminate much of the common error. This technique also removes the wet atmospheric effects, although these effects have greater spatial and temporal variability and their effect at one receiver are less likely to be the same as their effect at another receiver. Obviously the greater the separation between stations the less the correlation in the errors. This includes separation due to height differences, and thus the tropospheric error can be significant in airborne GNSS receivers (for example, those used in aerial photogrammetry). The improve-

ment from using differential or relative positioning is difficult to quantify, as it depends on many factors including the vertical and horizontal separation between receivers and the degree to which the troposphere decorrelates. However, the troposphere typically does not tend to decorrelate until about 15 km (Langley, 1998b), so if the receiver separation is less than this the residual tropospheric range errors (after correction by a model) should be at the centimetre level.

The lower atmosphere also has additional effects on GNSS signal, including attenuation and random amplitude and phase scintillations (changes). However, these effects are minimal and are only significant for short periods of time and for satellites at low elevation angles (Spilker, 1996b).

3.3.4 Ionosphere

The ionosphere is the region of the earth's atmosphere extending from approximately 50 km to 1000 km above the earth's surface. Like the lower atmosphere its primary effect is to change the time it takes the GNSS signal to reach the receiver. However, unlike the lower atmosphere the ionosphere effects the code and carrier phase measurements differently. The effect on the code is a delay: its velocity is reduced from its free space value. Put more precisely, the ionosphere reduces the group velocity of the GNSS signal, which is the velocity at which the energy or information in the wave is transmitted (Pain, 1993; Griffiths, 1989). Thus, the effect on the code is termed the *ionospheric group delay*. For the carrier phase the equal-in-magnitude but opposite effect is observed: the phase appears to advance, and the phase velocity exceeds its free space value. This effect, which is termed the *ionospheric phase advance*, means that the velocity of the phase in the ionosphere is greater than the velocity of light in free space. This would appear to violate the physical law which states that nothing can travel faster than the speed of light. However, this limitation applies only to the transmission of information or energy. The carrier phase itself carries neither information nor energy, and despite it travelling at a velocity greater than that of light actual

communication does not occur faster than the speed of light (Langley, 1993; Klobuchar, 1996).

Both the delay of the code and the advance of the carrier phase are – like the tropospheric delay – dependant upon the change in the index of refraction experienced by a GNSS signal as it passes through the ionosphere. The index of refraction is, in turn, dependant upon the integrated electron density (the number of free electrons) along the GNSS signal's path. This integrated density is commonly referred to as the Total Electron Content (TEC) and the free electrons themselves are the result of ionizing radiation on the upper atmosphere (Langley, 1998a).

Single frequency GNSS users have two options to reduce the effect of the ionosphere on their measurements. The first method is to model its effect. For GPS, a simple model is included in the navigation message (Klobuchar, 1986; Feess and Stephens, 1986). Unfortunately, as shown in Figure 3.7, this model performs only moderately well at estimating the delays; in general, it is effective at removing approximately 50% of the ionospheric effect. More sophisticated models are available, but their performance is not significantly better than the broadcast model (Langley, 1998a). The second technique to reduce the error is to use differential or relative positioning. Satellite signals passing through approximately the same region of the ionosphere will have similar errors, and thus when the measurement from one receiver is subtracted from the measurement from another receiver much of the error will cancel. As with the tropospheric error, as the distances between the receivers increases, the amount of common error between them lessens and differencing between is correspondingly less successful at removing the ionospheric error.

In addition to the above techniques, dual frequency receivers have a much more effective method to correct for the ionospheric effect. For radio waves at GNSS frequencies the ionosphere is a dispersive (frequency-dependent) medium. This means that radio waves of different frequencies will suffer from different errors. This allows the first-order effect of the ionosphere to be eliminated by forming the appropriate combination of the pseudoranges

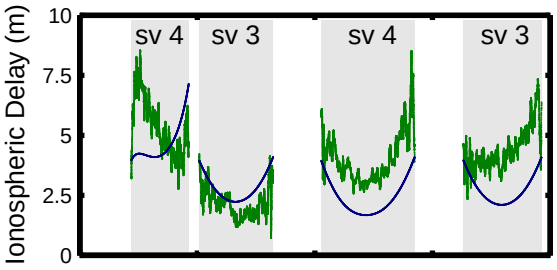


Figure 3.7: Comparison of broadcast ionospheric model estimated delays with measured delays (SV 1, day 2 of GPS Week 1217)

or carrier phases observed on the two different frequencies; i.e., the ionospheric-free combination introduced in Section 3.2.3. This combination removes over 99% of the ionospheric error but with the undesirable side-affect of substantially scaling the measurement noise. For this reason, the iono-free combination is not normally used with short-baseline relative GNSS, as most of the ionospheric error can be eliminated by differencing between receivers as detailed above.

A final technique for handling ionospheric delays is to include them in an adjustment or Kalman filter as unknown parameters. This approach is normally combined with one or more of the above techniques: for instance, the broadcast model and differencing can first be used to first reduce the errors, and then the residual remaining errors estimated. Because each satellite (or double-differenced pair) has a different ionospheric delay, the number of additional parameters in the adjustment or filter will be increased. However, for carrier-phase positioning the increased computational cost and reduced redundancy are well worth it, as estimating ionospheric delays greatly benefits ambiguity resolution (Liu, 2003; Richert, 2005).

Before correction the ionosphere can cause errors between $1 - 100\,\mathrm{m}$ on GNSS range measurements, although errors of between $10 - 30\,\mathrm{m}$ are more typical (Jorgensen, 1989; Klobuchar, 1986). As detailed above, single frequency users can reduce this error by more than half using the broadcast GPS ionosphere model while dual frequency users can eliminate it nearly entirely using the iono-free combination. The amount of error corrected for using

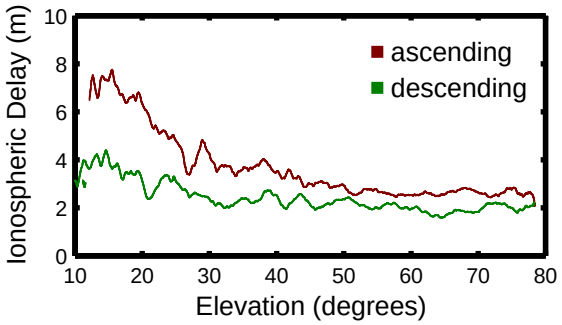


Figure 3.8: Ionospheric delays
(SV 1, day 2 of GPS Week 1217)

differential techniques is – owing to the ionosphere's spatial and temporal variability – impossible to predict. Generally, kinematic GNSS integer ambiguity resolution – the key to precise GNSS positioning and an indication of the ionospheric activity – should be possible on baselines between $20 - 30$ km. However, the errors induced by the ionosphere depend on the degree of sunspot activity. During periods of high sunspot activity the ionosphere becomes particularly active, and the greater the activity the more rapidly the ionosphere decorrelates. Consequently, relative GNSS becomes less effective and it can be difficult to perform ambiguity resolution on baselines as short as 10 km (Langley, 1998b). Sunspot activity generally follows an 11 year cycle (the most recent peak was during 2000), although "storms" can occur at any time.

Figure 3.8 shows the C/A code ionospheric delay for a single GPS satellite observed by an IGS station in Yellowknife, Canada (the same station is used in all the examples that follow). The data in this figure was generated using the ionospheric-free linear combination of code measurements in conjunction with precise satellite and receiver differential code delays provided by Center for Orbit Determination in Europe (CODE) (CODE, 2003). The delays were, furthermore, smoothed using a 5-minute moving average filter. In addition to showing the absolute magnitude of the ionospheric delays, this figure also illustrates the relationship between satellite elevation and ionospheric delay.

In addition to the code delay and carrier phase advance, the ionosphere can also have

additional effects on the GNSS observables. The most troubling additional effects, and the most difficult to handle, are short term variations in amplitude and phase of the received signals. These variations, which are commonly referred to as ionospheric scintillation, are caused by rapid changes in the TEC along the signal path. These changes cause short term signal attenuation which make it difficult for a receiver to reliably and continuously track the signal (Klobuchar, 1996). They also cause rapid changes in phase, again making it difficult for the receiver to maintain lock on the signal. The problem is especially severe for receivers tracking the L2 frequency from pre-modernised GPS satellites because of the already decreased SNR on that frequency that is a result of the codeless and semi-codeless techniques required to track it.

For both the ionospheric and tropospheric errors a note should be made regarding recent efforts to use multiple reference station networks to reduce the error and increase differential GNSS reliability. Multiple reference networks involve using several GNSS base stations to model the atmospheric effects over a wide area. The advantages of this approach include (Lachapelle et al., 2000):

- Increased accuracy: the multiple reference stations allow better modelling of spatially correlated errors such as those caused by the ionosphere or troposphere
- Larger coverage: the distance to the nearest base station can be much larger than with a single reference station
- Increased robustness: if one station fails, other stations can continue to provide differential corrections

The goal of the multi-reference station approach is normally to improve the accuracy of real-time GNSS; however, it can also be used in post-mission GNSS as well. A review of the different multi-reference station techniques can be found in Fotopoulos and Cannon (2001).

3.4 Adjustment of GNSS Observations

GNSS observations are adjusted using the strategies and equations in Section 2.1. The vast majority of adjustments fall into two categories: either undifferenced pseudorange observations are used, or a combination of double-differenced code and carrier phase observations are used. The former category is termed single point positioning, so-called because it only requires observations from a single receiver, while the latter category is termed relative positioning, so-called because the output from the adjustment is the relative vector between two receivers.

3.4.1 Single-Point Positioning

In single point positioning, undifferenced pseudorange observations given by Equation (3.4) are adjusted. Thus, the observations vector is nominally given by

$$\mathbf{l} = \begin{pmatrix} p_1 & p_2 & p_3 & \cdots & p_n \end{pmatrix}^T. \tag{3.37}$$

The observations can, alternatively, be a linear combination of undifferenced pseudorange measurements, although only the ionospheric-free combination is commonly used.

The unknown parameters are the co-ordinates of the antenna, contained within the receiver-satellite length $|\mathbf{r}_{rx/sv}|$, and the receiver clock offset, $t_{GNSS/rx}$,

$$\mathbf{x} = \begin{pmatrix} x_{rx} & y_{rx} & z_{rx} & t_{GNSS/rx} \end{pmatrix}^T \tag{3.38}$$

If observations from more than one GNSS are being adjusted simultaneously, then the parameter set in Equation (3.38) can be expanded to include one clock offset for each GNSS observed. For instance, if both GPS and GLONASS observations are present, then

$$\mathbf{x} = \begin{pmatrix} x_{rx} & y_{rx} & z_{rx} & t_{GPS/rx} & t_{GLONASS/rx} \end{pmatrix}^T. \tag{3.39}$$

The design matrix for a single observation in an adjustment involving only observations from a single GNSS, is

$$\mathbf{a} = \begin{pmatrix} \frac{x_{rx/sv}}{|\mathbf{r}_{rx/sv}|} & \frac{x_{rx/sv}}{|\mathbf{r}_{rx/sv}|} & \frac{x_{rx/sv}}{|\mathbf{r}_{rx/sv}|} & c \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{r}}_{rx/sv}^T & c \end{pmatrix}. \tag{3.40}$$

The first three columnar elements in each row of the design matrix are the co-ordinates of the unit vector from the antenna to each satellite. This vector is alternatively termed the Line-of-Sight (LOS) vector or vector of direction cosines. As hinted to by this last name, this vector can be expressed as a function of sines and cosines,

$$\hat{\mathbf{r}}_{rx/sv} = \mathbf{R}_l^e \begin{pmatrix} \sin(az)\cos(el) \\ \cos(az)\cos(el) \\ \sin(el) \end{pmatrix}. \tag{3.41}$$

where $\mathbf{R}_l^e$ is the rotation matrix between the local-level and Earth-Centred Earth-Fixed (ECEF) frames, and az and el are, respectively, the azimuth and elevation of the satellite. This formulation more explicitly shows how the design matrix is a function of geometry.

The unit vector elements of the design matrix will be very small values compared to the final speed-of-light element. To avoid a highly scaled normal matrix, the system is typically re-parametrised so that $c\,t_{GNSS/rx}$ is adjusted for, rather than just $t_{GNSS/rx}$. When so parametrised, the final element in each row of the design matrix becomes 1.0.

Because the undifferenced pseudorange equation is non-linear with respect to the parameters, non-linear least squares must be used. From Equation (2.28), the mathematical model is

$$\mathbf{A}\boldsymbol{\delta} = -\mathbf{w} \tag{3.42}$$

The geometric connection between the co-ordinate errors in $\boldsymbol{\delta}$ and the misclosures $\mathbf{w}$ can be seen by examining each row in this system of equations, disregarding the clock offset term. Each row is a dot product between the satellite-to-receiver line-of-sight vector $\mathbf{a}$ and the

co-ordinate errors $\boldsymbol{\delta}$,

$$w_i = \hat{\mathbf{r}}_{rx/sv} \, |\boldsymbol{\delta}_i| \cos\theta. \tag{3.43}$$

In other words, the misclosures are the projection of the co-ordinate errors onto the satellite line-of-sight vector. The inverse solution, then, is the projection of the misclosures onto the co-ordinate errors.

3.4.2 Relative Positioning

When double-differenced observations are used, the choice of parameters and observations becomes a little more complex than with undifferenced observations. For example, widelane or ionospheric-free observables could be used together with the L1 observable, and widelane and ionospheric-free ambiguities estimated in addition to L1 ambiguities. The most natural choice, however, is to use the double-differenced pseudoranges and carrier phases on all available frequencies and estimate the ambiguities on each frequency. In this case, assuming pseudoranges and carrier phases are available on two frequencies for all n satellites, and that the carrier phases are expressed in cycles, the observations are

$$\mathbf{l} = \begin{pmatrix} p_{m/r}^{b/1}(f_1) & p_{m/r}^{b/2}(f_1) & \cdots & p_{m/r}^{b/n-1}(f_1) \\ \phi_{m/r}^{b/1}(f_1) & \phi_{m/r}^{b/2}(f_1) & \cdots & \phi_{m/r}^{b/n-1}(f_1) \\ p_{m/r}^{b/1}(f_2) & p_{m/r}^{b/2}(f_2) & \cdots & \phi_{m/r}^{b/n-1}(f_2) \\ \phi_{m/r}^{b/1}(f_2) & p_{m/r}^{b/2}(f_2) & \cdots & p_{m/r}^{b/n-1}(f_2) \end{pmatrix}^T. \tag{3.44}$$

The minimal parameters are

$$\mathbf{x} = \begin{pmatrix} x_m & y_m & z_m & x_r & y_r & z_r & \\ & & & N_{m/r}^{b/1}(f_1) & N_{m/r}^{b/2}(f_1) & \cdots & N_{m/r}^{b/n}(f_1) \\ & & & N_{m/r}^{b/1}(f_2) & N_{m/r}^{b/2}(f_2) & \cdots & N_{m/r}^{b/n}(f_2) \end{pmatrix}^T \tag{3.45}$$

This is the *minimal* parameter set because the ambiguities are only constant if the satellites are constantly tracked. If tracking of a satellite is interrupted, the integration constant in Equation (3.13) may change; consequently, so will the ambiguity, and a new one must be estimated.

A row of the design matrix corresponding to a pseudorange measurement is given by

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}_m & \mathbf{a}_r \end{pmatrix} \tag{3.46}$$

where

$$\mathbf{a}_m = \begin{pmatrix} \frac{x_{m/i}}{|\mathbf{r}_{m/i}|} - \frac{x_{m/b}}{|\mathbf{r}_{m/b}|} & \frac{x_{m/i}}{|\mathbf{r}_{m/i}|} - \frac{x_{m/b}}{|\mathbf{r}_{m/b}|} & \frac{x_{m/i}}{|\mathbf{r}_{m/i}|} - \frac{x_{m/b}}{|\mathbf{r}_{m/b}|} \end{pmatrix} \tag{3.47}$$

and

$$\mathbf{a}_m = \begin{pmatrix} \frac{x_{r/i}}{|\mathbf{r}_{r/i}|} - \frac{x_{r/b}}{|\mathbf{r}_{r/b}|} & \frac{x_{r/i}}{|\mathbf{r}_{r/i}|} - \frac{x_{r/b}}{|\mathbf{r}_{r/b}|} & \frac{x_{r/i}}{|\mathbf{r}_{r/i}|} - \frac{x_{r/b}}{|\mathbf{r}_{r/b}|} \end{pmatrix} \tag{3.48}$$

For a carrier phase observation, a row in the design matrix is

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}_m & \mathbf{a}_r & \mathbf{a}_N \end{pmatrix} \tag{3.49}$$

with

$$\mathbf{a}_m = \begin{pmatrix} \frac{f_i}{c}\frac{x_{m/i}}{|\mathbf{r}_{m/i}|} - \frac{f_b}{c}\frac{x_{m/b}}{|\mathbf{r}_{m/b}|} & \frac{f_i}{c}\frac{x_{m/i}}{|\mathbf{r}_{m/i}|} - \frac{f_b}{c}\frac{x_{m/b}}{|\mathbf{r}_{m/b}|} & \frac{f_i}{c}\frac{x_{m/i}}{|\mathbf{r}_{m/i}|} - \frac{f_b}{c}\frac{x_{m/b}}{|\mathbf{r}_{m/b}|} \end{pmatrix}, \tag{3.50}$$

$$\mathbf{a}_r = \begin{pmatrix} \frac{f_i}{c}\frac{x_{r/i}}{|\mathbf{r}_{r/i}|} - \frac{f_b}{c}\frac{x_{r/b}}{|\mathbf{r}_{r/b}|} & \frac{f_i}{c}\frac{x_{r/i}}{|\mathbf{r}_{r/i}|} - \frac{f_b}{c}\frac{x_{r/b}}{|\mathbf{r}_{r/b}|} & \frac{f_i}{c}\frac{x_{r/i}}{|\mathbf{r}_{r/i}|} - \frac{f_b}{c}\frac{x_{r/b}}{|\mathbf{r}_{r/b}|} \end{pmatrix} \tag{3.51}$$

and $\mathbf{a}_N$ simply being an array of ones.

Given how the double-differenced observations are formed, it is perhaps not surprising that the double-differenced design matrix can also be formed by differencing. The elements in Equation (3.47) are, for instance, the differences between the elements of the undifferenced pseudorange design matrices for the base and i^{th} satellites. Reusing these matrices is attractive from an implementation standpoint.

3.5 Kinematic Positioning

Although kinematic positioning can be done in a Least-Squares adjustment, it is far easier (and more sensible) to do it using a Kalman filter. Since the measurements are non-linear, an extended Kalman filter is used. The state vector can consist of either the total states (e.g., Equation (2.79)) or the error states (e.g., Equation (2.97)); typically, the former is chosen so that an external prediction engine for the total states is not required. Since the same measurement equations are used as those used in least squares adjustments, the linearised observation equations are the same.

The complete state transition model for a GNSS Kalman filter is a collection of smaller, independent transition models for like-parameter subsets. For instance, each co-ordinate/velocity interaction is independent of other co-ordinates/velocities.

3.5.1 Position state transition models

For the position states and their derivatives, three models are used, depending upon the expected dynamics of the platform the antenna is mounted upon:

- The position model
- The Position/Velocity (PV) model
- The Position/Velocity/Acceleration (PVA) model

Typically, all co-ordinate states will use the same model; however, this is not required. For instance, an automobile may use PVA models for its horizontal motion, but only a PV model for its vertical motion.

Position model

The position model assumes that the antenna is stationary. Instinctively, a random constant state transition model is called for. However, as noted in Section 2.2.4, a random walk model is often used to avoid the numerical difficulties that a random constant model can lead to. Accordingly, the transition matrix is, simply,

$$\mathbf{\Phi} = \begin{pmatrix} 1 \end{pmatrix}. \tag{3.52}$$

Some token process noise variance is added to prevent the numerical problems.

Position-Velocity model

The PV model is for a moving platform whose velocities are assumed to follow a Gauss-Markov process,

$$\begin{pmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{pmatrix} = \begin{pmatrix} 0 & k \\ 0 & -\beta \end{pmatrix} \begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} + \begin{pmatrix} 0 \\ s \end{pmatrix} u(t) \tag{3.53}$$

The term k scales the velocities to a change in position. If both the position and velocities are in the ECEF frame, then $k = 1$; if the positions are geodetic co-ordinates and the velocities local-level, then $k = M + h$ and $k = (N + h)\cos(\phi)$ for latitude and longitude, respectively. The s term is a scale applied to the unit white noise $u(t)$.

The transition matrix for this model can be determined by a power series expansion, Equation (2.100),

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{F}\Delta t + \frac{\mathbf{F}^2\Delta t^2}{2!} + \cdots + \frac{\mathbf{F}^i\Delta t^i}{i!} + \cdots$$

This requires the powers of the dynamics matrix,

$$\mathbf{F}^2 = \begin{pmatrix} 0 & -k\boldsymbol{\beta} \\ 0 & \boldsymbol{\beta}^2 \end{pmatrix}, \; \mathbf{F}^3 = \begin{pmatrix} 0 & k\boldsymbol{\beta}^2 \\ 0 & -\boldsymbol{\beta}^3 \end{pmatrix}, \; \cdots \tag{3.54}$$

From this, the individual elements of the transition matrix are,

$$\begin{aligned} \boldsymbol{\phi}_{12} &= k\Delta t - k\frac{1}{2!}\Delta t^2\beta + k\frac{1}{3!}\Delta t^3\beta^2 + \cdots \\ &= \beta^{-1}\left(1 - 1 + k\beta\Delta t - k\beta^2\frac{1}{2!}\Delta t^2 + k\beta^3\frac{1}{3!}\Delta t^3 + \cdots\right) \\ &= k\beta^{-1}\left[1 - \exp\left(-\beta\Delta t\right)\right] \end{aligned} \tag{3.55}$$

And,

$$\boldsymbol{\phi}_{22} = 1 - \Delta t\beta + \frac{1}{2!}\Delta t^2\beta^2 - \frac{1}{3!}\Delta t^3\beta^3 + \cdots = \exp\left(-\beta\Delta t\right)$$

By inspection, the remaining elements, $\boldsymbol{\phi}_{11}$ and $\boldsymbol{\phi}_{21}$, are 1 and 0, respectively. So,

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & k\beta^{-1}\left[1 - \exp\left(-\beta\Delta t\right)\right] \\ 0 & \exp\left(-\beta\Delta t\right) \end{pmatrix} \tag{3.56}$$

Position-Velocity-Acceleration model

The PVA model assumes that the accelerations follow a Gauss-Markov process,

$$\begin{pmatrix} \dot{x}(t) \\ \ddot{x}(t) \\ \dddot{x}(t) \end{pmatrix} = \begin{pmatrix} 0 & k & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\beta \end{pmatrix} \begin{pmatrix} x(t) \\ \dot{x}(t) \\ \ddot{x}(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ s \end{pmatrix} u(t) \tag{3.57}$$

Using the transfer function, $\mathcal{L}^{-1}\left(s\mathbf{I}-\mathbf{F}\right)^{-1}$, and stating the result directly,

$$\mathbf{\Phi}=\begin{pmatrix}1 & k\Delta t & k\beta^{-2}\left[\beta\Delta t-1+\exp\left(-\beta\Delta t\right)\right]\\ 0 & 1 & \beta^{-1}\left[1+\exp\left(-\beta\Delta t\right)\right]\\ 0 & 0 & \exp\left(-\beta\Delta t\right)\end{pmatrix} \tag{3.58}$$

3.5.2 Other state transition models

Depending on the type of observations used in the filter, additional states are required in the state transition model:

- When undifferenced pseudoranges observations are used, the state vector must include terms for the receiver clock bias and rate. A typical dynamic model for these states is a 2-state random walk (Brown and Hwang, 1997).
- When double-differenced carrier phases are used, the state vector must include terms for the ambiguities. The ambiguities are modelled as a random walk; a small amount of process noise is added to prevent numerical problems.
- For longer baselines, where the ionospheric error is inadequately mitigated by double-differencing, ionospheric states are added to the filter. The ionospheric states are modelled as either random walk (cf. Odijk, 2002; Julien et al., 2004), or Gauss-Markov model (cf. Liu and Lachapelle, 2002; Han and Dai, 2007).

3.6 Other GNSS Positioning Techniques

For completeness, it should be noted that there are additional techniques of processing GNSS beyond those detailed above. Recently, for instance, there has been much interest in Precise Point Positioning (PPP), where sophisticated error mitigation is used that enables carrier-phase ambiguities to be estimated for undifferenced carrier-phase observations, cf. Gao and Shen (2002). Highly-accurate relative undifferenced carrier-phase measurements can also

be used explicitly in a Kalman filter, as described by Ford and Hamilton (2003). Between-receiver single differences are occasionally used in a manner akin to double-differences for relative positioning. Unlike with double-differences, however, a receiver clock offset must be estimated in the adjustment (Vollath and Doucet, 2007).

It seems natural to wonder why between-satellite single differences are not used for positioning. This has the presumed benefit that the receiver clock offset would not have to be estimated with these observations. However, the clock term that must be included in the adjustment with undifferenced observations absorbs some residual errors, making single point positioning more accurate (Beran et al., 2004).

3.7 Ambiguity Resolution

Least squares adjustments and Kalman filters work with real numbers. Consequently, all of the parameters output by both are real-valued (or, in computer science terminology, float numbers). However, as indicated in the previous sections, determining the integer number of the carrier phase ambiguities is necessary for highest accuracy GNSS positioning. The process of determining the integer ambiguities is termed ambiguity resolution. It can be divided into two stages: estimation and validation (Teunissen, 1995). Ambiguity resolution is also known as 'fixing' the ambiguities; a parameter set than includes integer ambiguities is known as a fixed solution.

3.7.1 Estimation

The mapping of the ambiguities from real number to integers is done using an *integer estimator*. Following Teunissen (1999b), three integer estimators are reviewed below. None of the estimators are unique to ambiguity resolution: all could be used for any estimation problem that has an integer solution space: cf. Agrell et al. (2002) or Kampes and Hanssen (2004).

Rounding

The simplest and perhaps most intuitive technique to convert the real valued ambiguities to integers is to round them. Denoting rounding by [], the integer ambiguities $\check{\mathbf{n}}$ are obtained from the real valued ambiguities $\mathbf{n}$ by

$$\check{\mathbf{n}} = \begin{pmatrix} [n_1] \\ [n_2] \\ \vdots \\ [n_n] \end{pmatrix} \tag{3.59}$$

Rounding will only provide satisfactory results if an adjustment or filter has converged to a high quality result. However, for this to occur, very long observation time spans are required. This requirement generally makes integer rounding an impractical choice for estimator.

Bootstrapping

The bootstrapping estimator combines rounding with sequential, constrained, least squares adjustments. It starts with the real-valued parameter with the smallest variance, rounding it to an integer,

$$\check{n}_1 = [n_1] \tag{3.60}$$

Next, this parameter is constrained to its rounded value and the remaining parameters are updated by performing a constrained least squares adjustment. Using a scalar version of the constrained least squares solution, Equation (2.55) developed in Section 2.1.4, each parameter is updated by

$$n_i = n_i^0 - \frac{\sigma_{n_i n_1}}{\sigma_{n_1}^2} \left(n_1 - \bar{n}_1 \right), i = 2...n. \tag{3.61}$$

The parameter with the next smallest variance is next rounded to an integer and the remaining real-valued parameters are again updated using constrained least squares. This

process repeats until all the parameters are fixed to integers. From the repeat application of Equation (3.61), a general expression for the i^{th} parameter can be derived,

$$\check{n}_i = \left[n_i - \sum_{j=1}^{i-1} \frac{\sigma_{n_i n_{j|J}}}{\sigma^2_{n_{j|J}}} \left(n_{j|J} - \check{n}_{j|J} \right) \right]. \tag{3.62}$$

where the j/J subscript denotes the j^{th} parameter obtained through constraints on the previous $j-1$ parameters.

Integer Least-Squares

Integer least squares is an estimator that attempts to 'optimally' estimate integer-natured parameters. As suggested by its name, the optimality criteria is, like normal least squares, that the sum of the squared weighted residuals is minimised. Repeating Equation (2.3), this criteria is,

$$f(\mathbf{v}) = \mathbf{v}^T \mathbf{C}_\mathbf{l}^{-1} \mathbf{v} = minimum.$$

The constraint on the solution, in normal least squares given by Equation (2.14), is modified to

$$\mathbf{A}\mathbf{x} = \mathbf{l} - \mathbf{v}, \mathbf{x} \in \ \mathbb{Z}^p \tag{3.63}$$

In other words, the parameters must be integers.

There is no closed-form expression for solving Equation (2.3) subject to Equation (3.63); instead, a search procedure must be used. Conceptually, this procedure is:

1. A range of candidate integers is selected for the first parameter. This range is centered around the current real-valued estimate, and has a width determined from the current estimated variance. This interval is commonly called the search width.

2. The parameter is set to the first integer within this range. The remaining parameters and their covariance information are updated by constrained least squares as is done in integer bootstrapping.

3. For the second, updated, parameter another range of candidate integers is selected. The variance, and hence search width for this parameter will be reduced by the bootstrapping in the previous step.

4. Steps 2 and 3 are repeated for all remaining parameters. The result is a set of integer estimates for the parameters. The sequence is then unwound and repeated until all possible integer combinations have been generated.

During the procedure, the integer constraining of previous parameters may shrink the search width for the current parameter to such a degree that no integers lie within it. In such a case, the search unwinds and the dead-end combination of integers is rejected. The combination with the minimum $\mathbf{v}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{v}$ will be the integer least squares estimate. The range of all combinations generated is termed the integer search space. It is contained in a hyper-ellipsoid centered around the real-valued estimate.

Decorrelation

Both bootstrapping and integer least-squares are complicated by parameter correlations that result from the real-valued estimation process. In the case of bootstrapping, these correlations may make one ambiguity appear to be more precise than it actually is. Consider, for instance, three real-valued ambiguities whose covariance matrix is given as

$$\mathbf{C_n} = \begin{pmatrix} 0.25 & 0 & 4.5 \\ 0 & 0.16 & 0.4 \\ 4.5 & 0.40 & 100 \end{pmatrix} \tag{3.64}$$

In this case, the second ambiguity has the apparent smallest variance: $\sigma_{n_2} = 0.16$. It would, accordingly, be the first ambiguity fixed in the bootstrapping process. However, the second ambiguity is only slightly correlated with third ambiguity, while the first is highly correlated, and some of the variability of the first ambiguity is due to the variability of

the third. The impact of these correlations can be viewed when the ambiguity covariance matrix is decorrelated,

$$\mathbf{C_n} = \mathbf{L_n D_n L_n^T}, \quad \mathbf{D_n} = \begin{pmatrix} 0.046 & 0 & 0 \\ 0 & 0.158 & 0 \\ 0 & 0 & 100 \end{pmatrix} \tag{3.65}$$

Using these results, the first ambiguity would be the starting point of the bootstrapping process, and, because the bootstrapping process is not unique, this might yield a different combination of integers. In the case of integer least squares, the correlations do not change the final result, but can cause very large search widths. Consequently, many integer combinations would have be searched for, making the process inefficient.

The above decorrelation analysis hints at the technique to mitigate the affects of the correlations: an integer-preserving decorrelating transformation. This transformation uses a matrix $\mathbf{Z}$ that decorrelates the covariance matrix of the ambiguities. That is,

$$\mathbf{C_z} = \mathbf{Z}^T \mathbf{C_n} \mathbf{Z}. \tag{3.66}$$

To make the ambiguity estimates consistent with this decorrelation, they must also be adapted,

$$\mathbf{z} = \mathbf{Z}^T \mathbf{n} \tag{3.67}$$

Two conditions are imposed on the decorrelating transformation (Teunissen, 1995):

1. Both the forward and reverse transformations must map integers to integers. This is necessary so that the ambiguity problem does not lose its inherent integer nature, and so that the integer estimators developed above can still be used on the decorrelated ambiguities.

2. The transformation must be volume-preserving. This is necessary so that the search

space of the decorrelated ambiguities is no worse than with the original ambiguities.

The above conditions, in turn, put two constraints on the decorrelating matrix:

1. All elements must be integers
2. The determinant of $\mathbf{Z}$ must be ± 1

A matrix that has these properties is call a unimodular matrix (Xu et al., 1995). Both conditions can be reduced to an equivalent single condition: $\mathbf{Z}$ and $\mathbf{Z}^{-1}$ are both integral (Hassibi and Boyd, 1998). This condition that the elements are integral implies that a perfect decorrelation is not possible; instead, they are decorrelated "as much as possible". In addition to being nearly decorrelated, the transformed ambiguities are also more precise (i.e., have smaller diagonal $\mathbf{C_z}$ values) than than the original ambiguities.

There are several techniques for generating the unimodular transformation. In the originating work of Teunissen (1993), integer Gauss transformations were applied. Later, Han and Rizos (1995), Liu et al. (1999), and Xu (2001) used unit triangular factorisations, while Hassibi and Boyd (1998) used an integer Gram-Schmidt orthogonalization. Common to all approaches is that an iterative process is used. This is necessary because the integer rounding in each step of their respective decorrelations modifies the existing problem.

Together, the ambiguity decorrelation and integer least squares make up the often-referenced Least-squares Ambiguity Decorrelation Adjustment (LAMBDA) method of ambiguity resolution (Teunissen, 1993, 1995). The two procedures are independent of each other. However, as noted above, the efficiency of the integer least squares process is much higher when the ambiguities are decorrelated, since far fewer ambiguities must be searched for. Pseudocode for both the decorrelation and integer least square processes in LAMBDA can be found in de Jonge and Tiberius (1996).

3.7.2 Validation

The integer ambiguities output by one of the above estimators are just that: estimates. They are, hopefully, the correct ambiguities; but, they may not be. As with any estimation procedure before a solution is accepted a validation procedure should be invoked. Only once the integer ambiguity combination passes this procedure will the ambiguities be considered fixed. It should be noted that most of the validation tests depend on the estimated real-valued solution. If the stochastic or deterministic measurement models are deficient, then this solution may itself be invalid. Consequently, before attempting ambiguity validation – indeed, before attempting integer estimation – validation should be performed on the real-valued estimated solution.

Acceptance Test

The first step often taken in a validation procedure is to perform a statistical acceptance test on the estimated integer ambiguity set. Intuitively, the null hypothesis tested is that the estimated integer ambiguities are representative of their true values,

$$H_0 : E\{\mathbf{n}\} = \check{\mathbf{n}}. \tag{3.68}$$

The corresponding alternative hypothesis is

$$H_a : E\{\mathbf{n}\} \neq \check{\mathbf{n}}. \tag{3.69}$$

Testing H_0 against H_a is a general linear hypothesis test, and the standard technique for performing such a test is using ANOVA. Table 3.4 contains the ANOVA table for the ambiguity acceptance test. In this table, n are the number of observations, m are the total number of parameters including the ambiguities, and p are the number of ambiguities. The test statistic is the ratio between the mean square errors due to all the parameters, and the mean square errors due to just the ambiguities: Ω_0 and Ω_0 in Table 3.4, respectively. This

Table 3.4: Ambiguity Acceptance ANOVA test table

Source of variation	Degrees of Freedom	Sum of Squares	Mean Square
All parameters	$n-m$	$\mathbf{v}^T\mathbf{C}_\mathbf{l}^{-1}\mathbf{v}=\tilde{\Omega}$	$\frac{\tilde{\Omega}}{n-m}=\tilde{\sigma}_0^2$
Non-ambiguity parameters	$n-(m-p)$	Ω	$\frac{\Omega}{n-(m-p)}=\sigma_0^2$
Ambiguities	p	$\tilde{\Omega}-\Omega=R$	$\frac{R}{p}=\sigma_0^2$

ratio follows an F-distribution; consequently, the ambiguity acceptance test is

$$\frac{\frac{\Delta\Omega_0}{p}}{\frac{\tilde{\Omega}_0}{n-m}} > F(p, n-m). \tag{3.70}$$

where F is the value from the F-distribution with $(p, n-m)$ degrees-of-freedom. If this test passes, then the null hypothesis is accepted. Otherwise, it is rejected, and either the adjustment or filter should continue to use the real-valued ambiguities, or another set of ambiguities should be estimated and validated.

A required quantity for an ANOVA test is the residual sum-of-squares for the integer-ambiguity constrained solution. From Leick (1995), this can be calculated using the residuals without constraints,

$$R = \tilde{\Omega}_0 \tag{3.71}$$

Or, even better, the sum of squares for just the ambiguities can also be calculated directly,

$$R = (\mathbf{n}-\check{\mathbf{n}})^T\,\mathbf{C}_n^{-1}\,(\mathbf{n}-\check{\mathbf{n}})\,. \tag{3.72}$$

The ambiguity acceptance test can be viewed from several, ultimately equivalent perspectives. One view of the test is that it checks if the fixed-ambiguities, replaced by constants, are superfluous to the solution. If so, then they can be removed from the real-valued estimation. An alternative view is that the test seeks confirmation that the estimated integer ambiguity set lies within the confidence region of the float ambiguity set (Han and Rizos, 1996b). Equivalently, the test is checking that the distance between the float and

fixed solutions is within limits set by the float solution (Verhagen, 2004).

Discriminant Test

After an acceptance test, the next test typically performed in a validation procedure is a discriminant test. The discriminant test is necessary because there may be multiple ambiguity combinations that both fit the model and pass the acceptance test. However, there is only one true ambiguity combination. This true combination must be discriminated from the pretenders. Obviously, this test is only useful if the integer estimator provides multiple candidate combinations. So, from the above estimators, only the integer least squares can be used in conjunction with this test. Typically, only two ambiguity combinations are tested: the combinations with the smallest and second smallest weighted sums of squared residuals.

The discriminant test most commonly applied is a ratio test. This test seeks to confirm what is intuitively desired: that best ambiguity combination is significantly better than any other combination. If the combinations with the smallest and second smallest weighted sums of squared residuals are denoted by subscripts 1 and 2, then the ratio test can be expressed as the using these quadratic forms,

$$\frac{\mathbf{v}_2^T \mathbf{C}_2^{-1} \mathbf{v}_2}{\mathbf{v}_1^T \mathbf{C}_l^{-1} \mathbf{v}_1} = \frac{\Omega_2}{\Omega_1} > k. \tag{3.73}$$

Alternatively, the ratio test can also be expressed as the ratio of the variabilities in the solution due only to the ambiguities (Euler and Schaffrin, 1990),

$$\frac{(\mathbf{n}_2 - \check{\mathbf{n}}_2)^T \mathbf{C}_{n_2}^{-1} (\mathbf{n}_2 - \check{\mathbf{n}}_2)}{(\mathbf{n}_1 - \check{\mathbf{n}}_1)^T \mathbf{C}_{n_1}^{-1} (\mathbf{n}_1 - \check{\mathbf{n}}_1)} = \frac{R_2}{R_1} > k. \tag{3.74}$$

Verhagen (2004) and Richert (2005) both indicate that the latter test outperforms the former. Regardless of the choice of test, if it passes then it is concluded that the best ambiguity combination is indeed significantly better than all other combinations, and can

be accepted as being the true ambiguities.

The choice of criteria in the above tests is open to some debate. Empirically derived constant values of 1.5 (Han and Rizos, 1996b), 2.0 (Wei and Schwarz, 1995), and 3.0 (Leick, 1995) have been used, with smaller-valued constants being used where there is more confidence in the error and covariance modelling. These constants can, unfortunately, give a very conservative acceptance rate (Verhagen, 2007). Alternatively, some authors have suggested that the test criteria should be drawn from an $F-$distribution, since the ratio is equivalent to a ratio of aposteriori variance factors. However, Verhagen (2004) argues that this approach is not sound since the two solutions are not independent.

Success Rate

An alternative or complementary approach to post-integer-estimation validation is to not attempt integer estimation unless the estimated success of the procedure meets some criteria. The success rates for rounding and bootstrapping integer estimators are developed in Teunissen (1998). For the latter estimator, the success rate lower bound is

$$P(\check{\mathbf{n}} = \mathbf{n}) \geq \prod_{j=1}^{n} \left[2\Phi \left(\frac{1}{2\sigma_{n_{j|J}}} \right) - 1 \right], \tag{3.75}$$

with Φ, the standard normal cumulative probability distribution, given by

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{u^2}{2}} du. \tag{3.76}$$

Unfortunately, no such straightforward expression for the success rate of integer least-squares exists and so some approximation must be used (Verhagen, 2003). The bootstrapped lower bound is a good candidate for this, since the integer least-squares success rate is always greater than or equal to to the bootstrapped rate (Teunissen, 1999a).

The success rate of the integer estimation can be calculated prior to the actual estimation; indeed, with an assumed stochastic model, even at the planning stage (Wang et al.,

2000). In fact, it can even guide the selection of a subset of ambiguities that will meet the required success rate, even if the complete set of ambiguities would not (Radovanovic, 2002). A high success rate can also be used to confirm that the fixed ambiguities can be considered as deterministic, a supposition that the discriminant tests above depended upon (Verhagen, 2003).

Other Methods

Another category of ambiguity validation includes tests over multiple epochs, rather than the single epoch tests from above. For instance, the carrier phase residuals over time can be tracked. If they start to diverge, then at some point in history the ambiguities might have been incorrectly resolved, and the ambiguities should be re-initialised (Gao et al., 1996). Unfortunately, all of the positions up to that point might have been biased. A more reliable technique is to not consider the ambiguities fixed until the same set of ambiguities have been fixed for certain number of epochs (Gao et al., 1996; Hu et al., 2004; Han and Dai, 2007). A similar approach, described by Han (1997), is to look at the change in ionospheric delay calculated from the fixed-ambiguity carrier phases between two epochs. This technique works across cycle slips: if there is a cycle slip, then the instantaneous ambiguity estimated on that satellite should change by as much as observed phase changes. Obviously, if there are no cycle slips then this is, effectively, equivalent to the previous repeat-ambiguities test.

Another technique for validating ambiguities in sequential processing is to perform repeated resolutions over several epochs while checking that the same ambiguities are consistently found. The acceptance and discriminant test both assume that the float solution is unbiased, and consequently they can break down when the reality does not fit with this assumption. Checking for repeated ambiguities guards against short-period errors like multipath or ionosphere scintillation that induce short-term biases in the float solution.

Ambiguity validation is especially tricky in Real-Time Kinematic (RTK) applications. By their very nature, such applications need ambiguities to be quickly resolved so that

functional or operational requirements can be met. Unfortunately, the negative impacts of wrongly fixed ambiguities can be higher than with post-processing applications. Consequently, the validation procedures found in real-time software are often a balancing act. For instance, to reduce the time-to-fix, the ratio test criteria might be reduced, while some ad-hoc validation procedures might be implemented to guard against (grossly) wrong fixes.

3.7.3 Solution Update

Once an integer ambiguity combination has been estimated and validated, then the adjustment or filter needs to account for the ambiguity information. In a least squares adjustment, the solution and its covariance is updated using the already-well-used Equations (2.55) and (2.56). As ambiguity resolution is typically the last stage in an GNSS adjustment, all that remains is to output this final, most-accurate, solution. A Kalman filter can also have constraint equations applied within it; however, an equivalent but more computational-friendly approach is to remove the fixed ambiguities from the state vector altogether. Provided all ambiguities were estimated and validated, the filter would then operate without ambiguities until either observations were made to a previously unused satellite, or a cycle-slip occurred on an existing satellite.

3.8 Filtering Using Geodetic Co-ordinates

In both Sections 3.4.1 and 3.4.2, the antenna position was parametrised in the ECEF frame. This is the most common approach in GNSS filters, and provides for the simplest equations. It is, however, possible to parametrise the system using geodetic co-ordinates. That is,

$$\mathbf{x} = \begin{pmatrix} \phi & \lambda & h & t_{GNSS/rx} \end{pmatrix}^T . \tag{3.77}$$

Obviously, the elements of the design matrix corresponding to the co-ordinates must change to accommodate this. This is most easily done using the chain rule,

$$\begin{pmatrix} \frac{\partial p}{\partial \phi} & \frac{\partial p}{\partial \lambda} & \frac{\partial p}{\partial h} \end{pmatrix} = \begin{pmatrix} \frac{\partial p}{\partial x^l_{rx/sv}} \frac{\partial x^l_{rx/sv}}{\partial \phi} & \frac{\partial p}{\partial y^l_{rx/sv}} \frac{\partial y^l_{rx/sv}}{\partial \lambda} & \frac{\partial p}{\partial z^l_{rx/sv}} \frac{\partial z^l_{rx/sv}}{\partial h} \end{pmatrix} \tag{3.78}$$

$$= \frac{\partial p}{\partial \mathbf{r}^l_{rx/sv}} \begin{pmatrix} \frac{\partial x^l_{rx/sv}}{\partial \phi} & \frac{\partial y^l_{rx/sv}}{\partial \lambda} & 1 \end{pmatrix}^T \tag{3.79}$$

The first set of partials are just the elements of $\hat{\mathbf{r}}_{rx/sv}$, rotated into a local-level frame,

$$\frac{\partial p}{\partial \mathbf{r}^l_{rx/sv}} = \mathbf{R}^l_e \hat{\mathbf{r}}_{rx/sv}. \tag{3.80}$$

The second set of differential quantities scale the former set for geodetic co-ordinates. With the latitude, for example, this is done by recognising that a differentially small change in latitude is related to a differentially small change in easting ($\hat{x}^l_{rx/sv}$) via the meridian radius of curvature at h,

$$\frac{\partial x^l_{rx/sv}}{\partial \phi} = M + h. \tag{3.81}$$

Similarly, for longitude,

$$\frac{\partial y^l_{rx/sv}}{\partial \lambda} = (N + h)\cos(\phi) \tag{3.82}$$

A benefit of parametrising the adjustment in terms of geodetic co-ordinates is that it is easier to apply geodetic-co-ordinate related constraints like, for instance, a height constraint. It is also a prerequisite if GNSS measurements are to be simultaneously filtered with another navigation system whose position is in a local-level frame: for instance, with a local-level inertial mechanisation. Lastly, it directly outputs latitude and longitude that are, for most users, the form of co-ordinates ultimately desired. Drawbacks of the parametrisation are its increased complexity and computational load.

Chapter 4

Photogrammetry

In analytical photogrammetry, the image measurements of object space features are adjusted using least squares and mathematical models describing the imaging process. The process by which this is done – the bundle adjustment – is amazingly flexible and powerful. In contemporary aerial and terrestrial mapping, one of the ways that this flexibility and power is leveraged is by the incorporation of GNSS data.

This chapter provides the following:

- A review of the image measurement equations. This review deviates slightly from a typical review in two aspects: first, a consistent vector notation is used throughout; and second, attention is drawn to co-ordinate frame definition and its importance.

- A comprehensive overview of current techniques for integrating together GNSS and photogrammetric data. To the author's knowledge, such a review that brings together all of the existing integration strategies cannot be found elsewhere. The review briefly sidetracks with an analysis of the impact of incorrect ambiguity resolution on exposure position estimates.

- The derivation of new image measurement equations that are more suitable when integrating together satellite positioning and photogrammetry. These equations will

be used when implementing the new integration strategies in Chapter 5.

4.1 Image measurement observation equations

A simplified projective model that serves as the basis for analytical photogrammetry is the *central perspective projection*. In this model, light rays reflected from points in object space are all assumed to meet at a single point known as the perspective center. A positive image of the object space is then produced from the intersections of these light rays with a plane orthogonal to the optical axis of the camera. This imaging geometry is depicted in Figure 4.1.

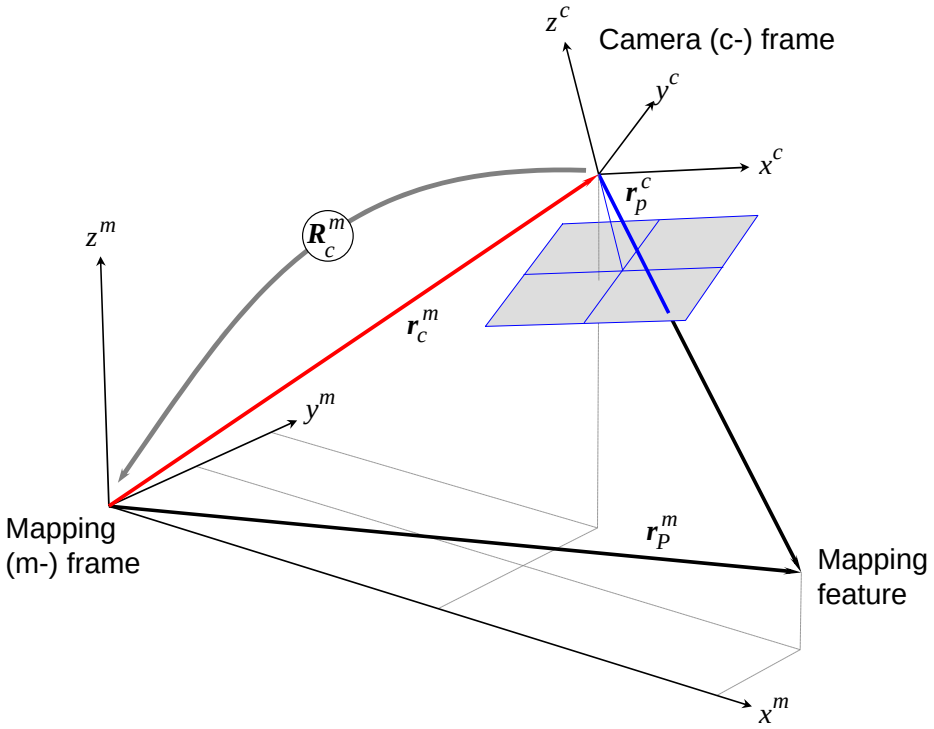


Figure 4.1: The central perspective projection

From Figure 4.1, it can be seen that the object (or *m*apping) space co-ordinates $\mathbf{r}_P^M$ of a point are related to its *c*amera co-ordinates $\mathbf{r}_p^c$ by a three-dimensional conformal transfor-

mation,

$$\mathbf{r}_P^m = \mathbf{r}_c^m + \mu \mathbf{R}_c^m \mathbf{r}_p^c, \tag{4.1}$$

where $\mathbf{r}_c^m$ is the position of the camera perspective center in the object space (or *m*apping) co-ordinate frame, $\mathbf{R}_c^m$ is the rotation matrix between the camera co-ordinate frame and the mapping co-ordinate frame, and μ is the scale between the camera frame and the mapping frame for point P. Rearranging Equation (4.1) yields the reverse transformation

$$\mathbf{r}_p^c = \mu^{-1} \mathbf{R}_m^c \left(\mathbf{r}_P^m - \mathbf{r}_c^m \right), \tag{4.2}$$

more compactly expressed by

$$\begin{aligned} \mathbf{r}_p^c &= \mu^{-1} \mathbf{R}_m^c \mathbf{r}_{c/P}^m \\ &= \mu^{-1} \mathbf{r}_{c/P}^c. \end{aligned} \tag{4.3}$$

This expression actually consists of three equations,

$$x_p^c = \mu^{-1} x_{c/P}^c \tag{4.4a}$$

$$y_p^c = \mu^{-1} y_{c/P}^c \tag{4.4b}$$

$$z_p^c = \mu^{-1} z_{c/P}^c \tag{4.4c}$$

The last of these equations can be made explicit in μ^{-1} and substituted into the former two equations, yielding

$$x_p^c = z_p^c \frac{x_{c/P}^c}{z_{c/P}^c} \tag{4.5a}$$

$$y_p^c = z_p^c \frac{y_{c/P}^c}{z_{c/P}^c}. \tag{4.5b}$$

These two equations are called the collinearity equations, and are so termed because the

perspective centre, image point, and object point that they involve are assumed to lie on a single line.

The step from collinearity equations (4.5) to the image measurement equations depends on how the camera and image axes are defined. The most common definition, depicted in Figure 4.2, defines the positive camera x, y, and z axes as right, up, and back, respectively, and the image axes as right and up, respectively. Replacing the elements of $\mathbf{r}_p^c$ with the image measurements (x_p^i, x_p^i) and principal distance c for this definition results in image measurements equations

$$x_p^i = -c\frac{x_{c/P}^c}{z_{c/P}^c} \tag{4.6a}$$

$$y_p^i = -c\frac{y_{c/P}^c}{z_{c/P}^c}. \tag{4.6b}$$

or, expanding all terms,

$$x_p^c = -c\frac{r_{m11}^c(x_P - x_c) + r_{m12}^c(y_P - y_c) + r_{m13}^c(z_P - z_c)}{r_{m31}^c(x_P - x_c) + r_{m32}^c(y_P - y_c) + r_{m33}^c(z_P - z_c)} \tag{4.7a}$$

$$y_p^c = -c\frac{r_{m21}^c(x_P - x_c) + r_{m22}^c(y_P - y_c) + r_{m23}^c(z_P - z_c)}{r_{m31}^c(x_P - x_c) + r_{m32}^c(y_P - y_c) + r_{m33}^c(z_P - z_c)}. \tag{4.7b}$$

where r_{mij}^c are the elements of the $\mathbf{R}_m^c$ rotation matrix. This form of the collinearity equations is that most commonly encountered in photogrammetric literature. The same form also results when the left-up-forward camera axis and right-down image axis definition of Figure 4.2 is used. This latter axes definition is useful because it enables digital co-ordinates (i.e., co-ordinates commonly used in computers) to be used for the image measurements together with the standard form of the collinearity equations. Of course, other definitions are possible. For example, the right-down-forward camera and right-down image axes of Figure 4.4 would also be useful for digital co-ordinates. In this case, however, the collinearity

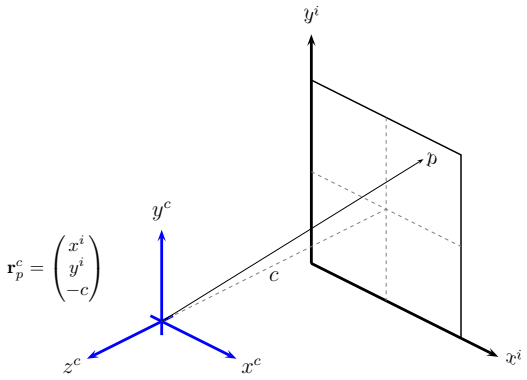


Figure 4.2: Right-up-back camera axes

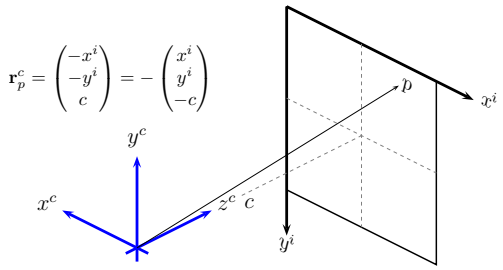


Figure 4.3: Left-up-forward camera axes

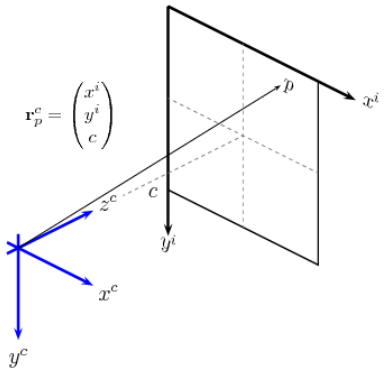


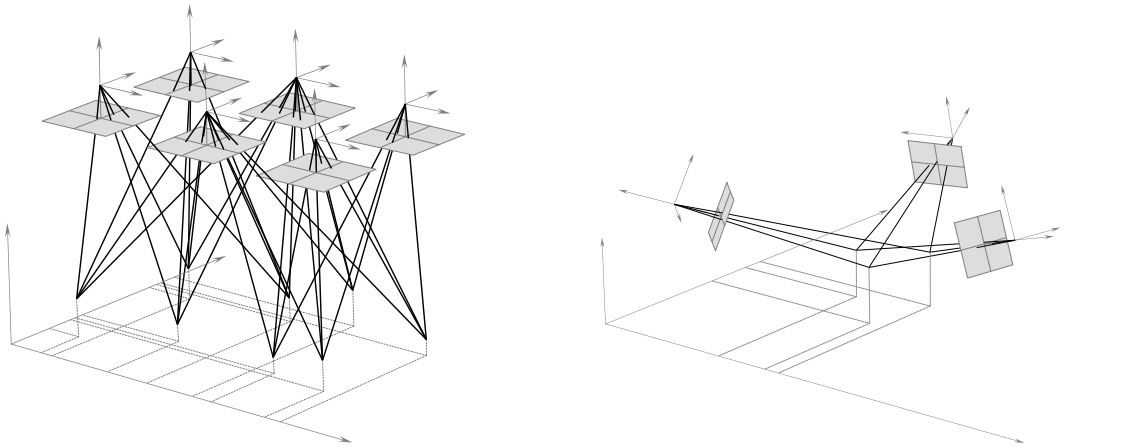
Figure 4.4: Right-down-forward camera axes

equations would become

$$x_p^i = c\frac{x_{c/P}^c}{z_{c/P}^c} \tag{4.8a}$$

$$y_p^i = c\frac{y_{c/P}^c}{z_{c/P}^c}. \tag{4.8b}$$

Although the definition of camera and image axes may appear trivial, it is, in fact, critically important to determining the $\mathbf{R}_m^c$ matrix.



(a) Aerial photogrammetric network lattice

(b) Convergent close-range photogrammetric network lattice

Figure 4.5: Common photogrammetric network lattices

4.2 Adjustment of Photogrammetric Networks

When measurements of common object features are made from multiple images, a photogrammetric network is formed. Such networks consist of three-dimensional lattices in which cameras are joined to object-space features by image measurements. Figure 4.5, for instance, shows two common network lattices encountered in photogrammetry: one formed by aerial images and one formed by convergent close range images. This figure shows how a "bundles" of light rays from the object space features can be seen meeting at each perspective centre. In photogrammetric network adjustments these bundles are – subject to the image measurement weight – adjusted as a unit. Hence, the simultaneous least-squares adjustment of photogrammetric networks is termed a *bundle adjustment*.

The image measurement equations developed above can be expressed as a set of parametric equations

$$\mathbf{l} = \mathbf{f}(\mathbf{x}) \tag{4.9}$$

where $\mathbf{l}$ consists of the image measurements (x_p^i, x_p^i) for all images, and $\mathbf{x}$ consists of the position and attitude parameters, again for all images. This set of equations, once linearised, can be adjusted using any of the techniques listed in Section 2.1.3.

4.3 Incorporating GNSS Data in Bundle Adjustments

Currently, the integration of GNSS data in photogrammetric bundle adjustments is universally done using GNSS-derived exposure station position observations. This is a two-step technique. In the first step, raw GNSS measurements are processed in a kinematic GNSS Kalman filter, yielding estimates of antenna position and position covariance at the GNSS measurement epochs. Using a linear or other low-order polynomial, positions and covariance information corresponding to the exposure times are then interpolated from these positions. In the second step, the estimates of exposure-station antenna position are used in a photogrammetric adjustment as position parameter observations. The nominal form of these equations is

$$\mathbf{r}_a^M = \mathbf{r}_c^M + \mathbf{R}_c^M \mathbf{r}_a^c \tag{4.10}$$

where $\mathbf{r}_a^M$ is the GNSS antenna position observation that is related to the camera perspective centre $\mathbf{r}_c^M$ through the camera/GNSS antenna lever-arm $\mathbf{r}_a^c$. $\mathbf{R}_c^M$ is the rotation matrix that aligns the reference frame of the camera with that of the mapping space.

Equation (4.10) can be augmented with similarity transformation terms that account for differences between the GNSS and mapping frame datums. The expanded equation is then

$$\mathbf{r}_a^e = \mathbf{r}_M^e + \mu \mathbf{R}_M^e \left(\mathbf{r}_c^M + \mathbf{R}_c^M \mathbf{r}_a^c \right) \tag{4.11}$$

where $\mathbf{r}_M^e$, μ, and $\mathbf{R}_M^e$ are, respectively, the translation, scale, and rotation between the mapping and GNSS datums. These terms can be included in the adjustment as unknown parameters, although for them to be determinable both ground control and GNSS position observations must be included in the adjustment.

4.3.1 Modelling errors by linear polynomials

When the positions-observations integration strategy was first devised, both receiver technology and ambiguity resolution techniques were less advanced than they are today. As a

result, the ambiguities resolved in the GNSS processor were less reliable, leading to incorrect position estimates. To combat this, Equation (4.11) had bias and time-dependent linear drift terms added to it, leading to the form of the equation that is used near-universally today,

$$\mathbf{r}_a^e = \mathbf{r}_M^e + \mu \mathbf{R}_M^e \left(\mathbf{r}_c^M + \mathbf{R}_c^M \mathbf{r}_a^c \right) + \mathbf{a}_0^e + \mathbf{a}_1^e \left(t - t_0 \right) \tag{4.12}$$

The position bias and drift terms, denoted by $\mathbf{a}_0^e$ and $\mathbf{a}_1^e$, respectively, are estimated in the adjustment. Normally, each strip of imagery (or, more generally, each approximately linear segment of the aircraft's trajectory) gets its own set of these parameters. If ground control is also used in the adjustment, then these two parameters can also account for inconsistencies between the ground control and GNSS datums. Using the shift and drift parameters for this purpose, however, negates the possibility of also estimating the similarity transformation parameters in the adjustment due to the strong coupling between the two sets of parameters. The shift and drift parameters are also coupled with elements of the camera interior orientation and lens distortion, and can thus can also account for errors in the camera calibration.

An obvious question regarding the shift and drift parameters is: How well do they model position errors due to incorrectly resolved ambiguities? To answer this, GPSoft's SatNav GPS simulator (GPSoft, 2003) was used to simulate 1000 s of double-difference carrier-phase GPS observations (at a typical aerial mapping aircraft speed of 60 m/s, this corresponds to a flight line of 60 km). For each epoch of observations, ambiguity biases of 1 and -2 cycles were added to the first two double-differences, and the resulting observations used for positioning. The resulting second-order and higher co-ordinate errors, determined from the residuals of a linear polynomial fit to each co-ordinate component, are shown in Figure 4.6. The unmodelled non-linear errors for this period are less than 0.5 cm, suggesting that the shift and drift parameters are, indeed, acceptable for modelling and removing position errors due to incorrectly resolved ambiguities.

Although it might be expected that the linearity of position errors cause by incorrect

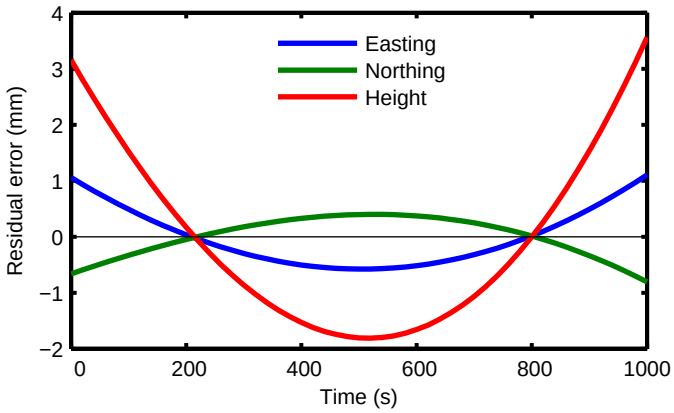


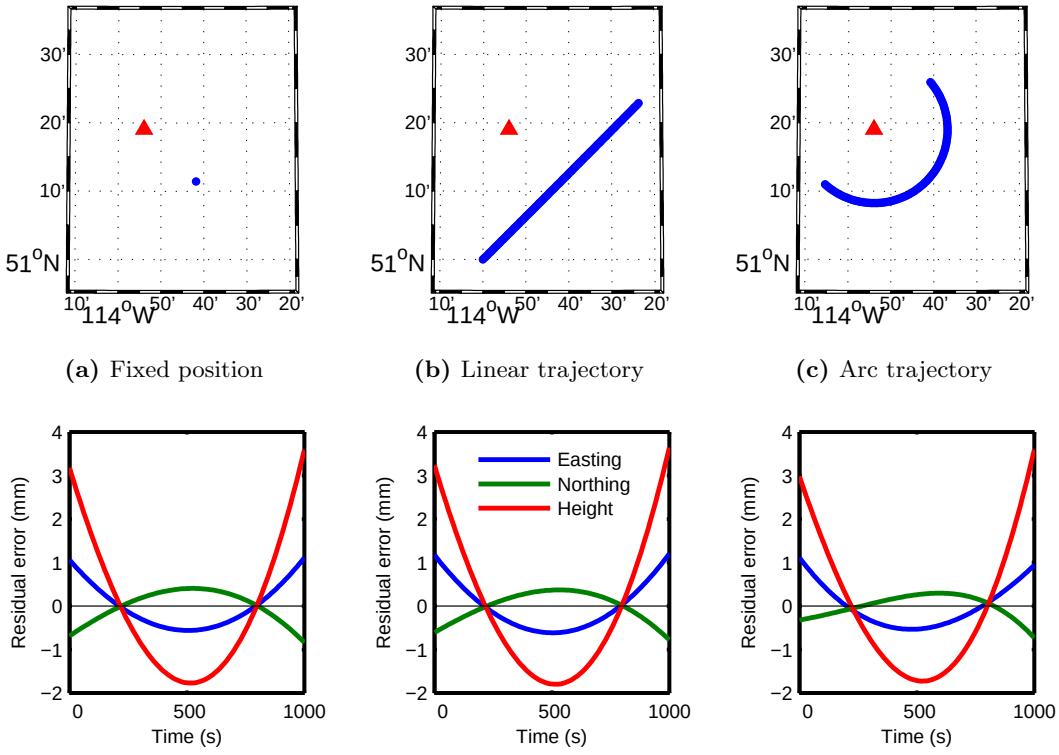
Figure 4.6: Non-linear residual co-ordinate errors caused by two incorrectly resolved double-difference ambiguities of 1 and 2 cycles

ambiguities depends on factors such as the master/remote separation or linearity of the trajectory, in actuality it depends only on the satellite constellation geometry and the magnitude of the incorrect ambiguities. Figure 4.7, for instance, shows the residual non-linear co-ordinate errors for three different remote trajectories: linear, circular, and fixed. The difference in both the actual and residual non-linear errors between the three trajectories is less than 1 mm. Similarly, for any trajectory the master/remote separation can be doubled without *any* effect on the co-ordinate errors.

To show analytically why only the magnitude of the incorrect ambiguities and the satellite constellation geometry affect the co-ordinate errors, consider the linearised double-difference GNSS carrier-phase observation equations, where the "perfect" carrier-phase measurements $\mathbf{l}$ are perturbed by a vector of ambiguity errors $\boldsymbol{\Delta}\mathbf{l}$,

$$\mathbf{l} + \boldsymbol{\Delta}\mathbf{l} = \mathbf{A}\mathbf{x} \tag{4.13}$$

In Equation (4.13), $\mathbf{x}$ is the vector of GNSS co-ordinates and $\mathbf{A}$ is the double-difference GNSS design matrix (i.e., the Jacobian of the double-difference GNSS observations with respect to the antenna co-ordinates). Using the normal equations, the least squares solution to this



(a) Fixed position **(b)** Linear trajectory **(c)** Arc trajectory

(d) Fixed position non-linear residual co-ordinate errors **(e)** Linear trajectory non-linear residual co-ordinate errors **(f)** Arc trajectory non-linear residual co-ordinate errors

Figure 4.7: Invariance of non-linear co-ordinate errors caused by incorrect ambiguities for different remote trajectories

equation is

$$\tilde{\mathbf{x}} = \left(\mathbf{A}^T\mathbf{P}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{P}\mathbf{l} + \left(\mathbf{A}^T\mathbf{P}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{P}\boldsymbol{\Delta}\mathbf{l} \tag{4.14}$$

$$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\Delta}\mathbf{x}.$$

So, the least-squares solution with the ambiguity errors is equivalent to the solution in the absence of the errors $\mathbf{x}$, plus a term which is a function only of the double-difference design matrix and the ambiguity errors $\boldsymbol{\Delta}\mathbf{x}$. This error term is itself the least-squares solution of

$$\boldsymbol{\Delta}\mathbf{l} = \mathbf{A}\boldsymbol{\Delta}\mathbf{x} \tag{4.15}$$

For a fixed master station the design matrix $\mathbf{A}$ is dependent only upon the remote and satellite positions; hence, the master/remote separation is not a factor in the non-linearity. Also, the direction cosines that the design matrix consists of change significantly only through satellite and not receiver motion, and therefore the trajectory of the vehicle has a negligible impact on $\mathbf{A}$, and, ultimately, on $\boldsymbol{\Delta}\mathbf{x}$.

Of course, the preceding analysis was a simplistic one, and the small residual non-linear errors shown in Figure 4.6 may not be representative. Using real data, Jacobsen and Schmitz (1996), for example, observed that non-linear errors of some decimeters were still present in incorrect ambiguity-corrupted GPS positions after a linear trend had been removed. To examine this claim, the above 1000 s experiment was repeated every 200 s for an entire week, and the maximum unmodelled 3D position errors are shown in Figure 4.8. The RMS of these maximum errors was less than 1.5 cm, and the maximum never exceeded 11 cm. This test suggests that the ambiguity errors in Jacobsen and Schmitz (1996) were either unusually large, that there was an extremely unfavourable satellite geometry, or that are other causes of non-linearity not considered in the above simulation. For example, in the GNSS processor there may be interactions between the carrier-phase solution and the code-range solution that lead to additional non-linearities.

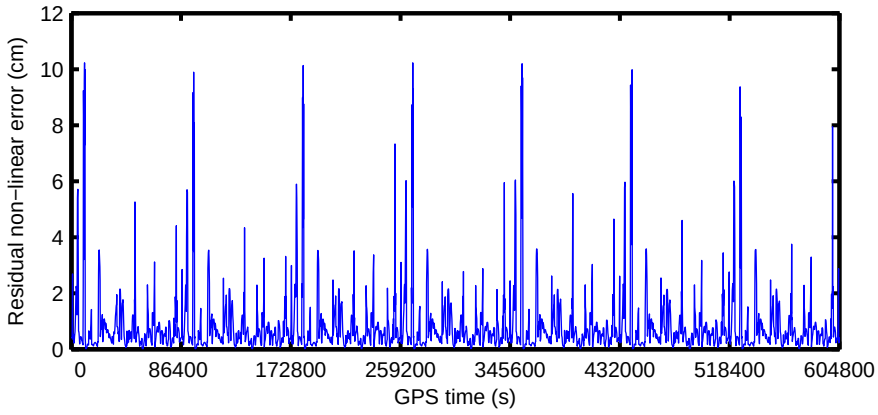


Figure 4.8: Maximum non-linear residual co-ordinate errors caused by two incorrectly resolved double-difference ambiguities of 1 and 2 cycles

The non-linear residual position errors shown in Figure 4.6 and in the figures of Jacobsen and Schmitz (1996) both appear to have a significant quadratic component. This suggests that adding an additional quadratic term to the conventional linear polynomial error model would result in a model much better at accounting for erroneously-resolved ambiguities. The revised position observation equation would become

$$\mathbf{r}_a^e = \mathbf{r}_M^e + \mu \mathbf{R}_M^e \left(\mathbf{r}_c^M + \mathbf{R}_c^M \mathbf{r}_a^c \right) + \mathbf{a}_0^e + \mathbf{a}_1^e \left(t - t_0 \right) + \mathbf{a}_2^e (t - t_0)^2, \tag{4.16}$$

where $\mathbf{a}_2^e$ is the newly-introduced quadratic term. When the test of Figure 4.8 is repeated with this second-order error model the RMS of the maximum errors is reduced to less than 1.5 cm. More significantly, the maximum errors are reduced to 1.5 cm, as shown in Figure 4.9.

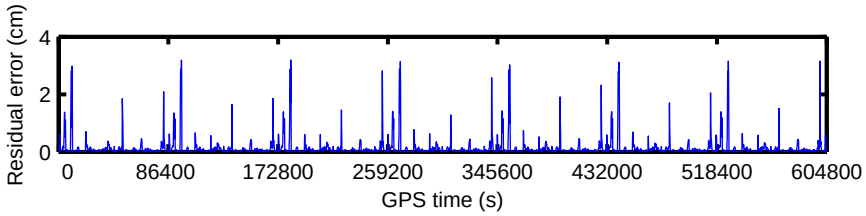


Figure 4.9: Maximum cubic and higher residual co-ordinate errors caused by two incorrectly resolved double-difference ambiguities of 1 and 2 cycles

4.3.2 Modelling errors using range corrections

An alternative technique for modelling GNSS errors in photogrammetric adjustments was investigated at the University of Hanover and Geo++ GmbH in the mid-nineties. In their ingenious approach, outlined in Jacobsen and Schmitz (1996) and Kruck et al. (1996), constant satellite-to-exposure station range corrections are estimated within the bundle adjustment for each GNSS satellite whose ambiguity was not reliably fixed in the kinematic GNSS processor. The development of this technique parallels the analysis of the effect of

ambiguity errors in Equations (4.13) and (4.14). Again, starting from the linearised GNSS range observation equations, the additional range corrections $\mathbf{\Delta l}$ are explicitly separated from the range measurements $\mathbf{l}$,

$$\mathbf{l} + \mathbf{\Delta l} = \mathbf{A x} \tag{4.17}$$

Solving this equation by least squares yields

$$\mathbf{x} = \left(\mathbf{A}^T\mathbf{P}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{P}\mathbf{l} + \left(\mathbf{A}^T\mathbf{P}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{P}\mathbf{\Delta l} \tag{4.18}$$

This equation has two terms: the first is the GNSS co-ordinate vector that would be solved for in the absence of the $\mathbf{\Delta l}$ range corrections, and the second is a vector of co-ordinate corrections that results because of these range corrections. This second term is introduced into the bundle adjustment's GNSS position observation equation, replacing the shift and drift terms from the conventional approach,

$$\mathbf{r}_a^M = \mathbf{r}_M^e + \mu\mathbf{R}_M^e\left(\mathbf{r}_c^M + \mathbf{R}_c^M\mathbf{r}_a^c\right) + \left(\mathbf{A}^T\mathbf{P}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{P}\mathbf{\Delta l} \tag{4.19}$$

The $\mathbf{\Delta l}$ range corrections are then are added to the bundle adjustment as unknown parameters. The design matrix $\mathbf{A}$ and weight matrix $\mathbf{P}$ are provided to the adjustment by the kinematic GNSS processor.

By comparing Equation (4.12) with Equation (4.19), it is apparent that the range corrections are effectively replacing the shift and drift terms from the conventional approach. The difference between the two approaches is that the GNSS errors are now being modelled and compensated for in measurement-space rather than in object-space. The actual integration, however, is still done in position-space.

This integration technique has several advantages over the traditional position observation GNSS/photogrammetry integration strategy, yet it is not quite the "rigorous" integration claimed. Improvements over the traditional approach include:

- the actual GNSS errors are better considered
- the number of unknowns is (in general) reduced
- no cross-strips are required
- GNSS errors can better be separated from datum and interior orientation parameters

In spite of these advantages it is, however, important to note that the actual GNSS ranges themselves are not used in the adjustment, and the integration is still done in object space. In other words, because the actual GNSS and image measurements are not used together, the integration is not yet "complete". Also, the sharing between the GNSS and photogrammetric processors is still only in one direction. In fairness, the creators of the technique do note that "re-substitution of the [range correction] terms [into the GNSS processor] is feasible"; however, they conclude that "it is not of much interest as the [GNSS] processing techniques improve" (Jacobsen and Schmitz, 1996).

It should be noted that a number of additional advantages of the technique are claimed by its creators in Schmitz et al. (2001). However, some of these are not supported by either empirical evidence or logical argument, and were consequently not included in the list above.

There are a number of practical, implementation-related, obstacles that need be overcome with this integration strategy. Firstly, the GNSS design matrices (or the satellite elevations and azimuths, which can be used to construct the design matrices) must be transferred between the GNSS and photogrammetric processors. Most GNSS processors do not output such information, and so a customised processor is required. Secondly, there is the problem of determining which GNSS position observations need to have the additional range corrections applied, and to which positions each range correction applies. This bookkeeping must be performed in the GNSS processor, and again, transferred to the adjustment.

4.3.3 Modelling errors as a random process

In the previous two sections, GNSS errors or inconsistencies were modelled and compensated for deterministically. An entirely different approach is to describe the error behaviour as a time-dependant random process. Stochastic constraints on adjacent exposure GNSS position errors can then be derived from the random process models, and these constraints used in photogrammetric simultaneous adjustments. The advantage of this approach is that it can help deal with inadequacies in the deterministic models; in particular, higher frequency GNSS errors that neither of the previous two error-modelling approaches can accommodate. It also accounts for time-dependent correlations between observed exposure station positions. These correlations, neglected in the deterministic approaches, can lead to an overly-rigid network with overly-optimistic statistics.

The fundamental difference between the two previous error modelling approaches and random-process approach is in how the GNSS errors or inconsistencies are parametrised. The basic equation for incorporating GNSS position observations in a photogrammetric bundle adjustment is

$$\mathbf{r}_a^M = \mathbf{r}_c^M + \mathbf{R}_c^M \mathbf{r}_a^c + \boldsymbol{\Delta}\mathbf{r}_a^M. \tag{4.20}$$

The vector $\boldsymbol{\Delta}\mathbf{r}_a^M$ contains the corrections applied to the observed GNSS co-ordinates $\mathbf{r}_a^M$ to make them consistent with the camera co-ordinates $\mathbf{r}_c^M$. In the two previous error modelling approaches, this term was described using deterministic models that extended over several exposures. For example, in the linear polynomial approach, the model was

$$\boldsymbol{\Delta}\mathbf{r}_a^M = \mathbf{a}_0^e + \mathbf{a}_1^e (t - t_0) \tag{4.21}$$

The model parameters, $\mathbf{a}_0^e$ and $\mathbf{a}_1^e$, are determined in the adjustment, with typically one set of parameters for each flight line. In contrast, with the random-process approach the corrections $\boldsymbol{\Delta}\mathbf{r}_a^M$ for *each* exposure are themselves explicitly adjustment parameters. In other words, each exposure has its own set of correction parameters, and the parameters

for all exposures are solved for in the adjustment. To ensure that the corrections are determinable, stochastic constraints stemming from random process models are applied to multiple corrections.

Gauss-Markov random process models

The random process approach was originated by Lee (1999) for the adjustment of airborne push broom scanner data. The scan lines (exposures) in push broom scanner data are captured at a very high data rate; consequently, their position errors are highly correlated. To accommodate these correlations, first and second-order Gauss-Markov random processes were chosen to describe the behaviour of the GNSS errors. For instance, using the former process the errors in each co-ordinate component are assumed to behave according to

$$\dot{x} = -\beta x + w, \tag{4.22}$$

where, as in Section 2.2.4, β is the inverse of the correlation time and w the Gaussian white noise driving the process. The equivalent discrete expression for the process is

$$\begin{aligned} x(t_i) &= e^{-\beta(t_i - t_{i-1})} x(t_{i-1}) + w_k \\ &= e^{-\beta \Delta t} x(t_{i-1}) + w_k \end{aligned} \tag{4.23}$$

From this equation, stochastic constraint equations can be produced for each co-ordinate component,

$$\begin{aligned} 0 &= \Delta x_a^M(t_i) - e^{-\beta_x \Delta t} \Delta x_a^M(t_{i-1}) \\ 0 &= \Delta y_a^M(t_i) - e^{-\beta_y \Delta t} \Delta y_a^M(t_{i-1}) \\ 0 &= \Delta z_a^M(t_i) - e^{-\beta_z \Delta t} \Delta z_a^M(t_{i-1}). \end{aligned} \tag{4.24}$$

These constraint equations are applied in the adjustment to the GNSS position errors for adjacent exposures (at times t_i and t_{i-1}). The observations' variances are the variances of the respective the white noises (or sequences) w_k. Explicitly, this is

$$\sigma_{w_k}^2 = E\{w_k w_k\} = \sigma^2 \left(1 - e^{-2\beta\Delta t}\right), \tag{4.25}$$

where σ^2 is the variance of the random process. In the originating work of Lee (1999), it is not clear if these rigorously-calculated weights were used for the observations, or if empirically-determined weights ($\sigma_{w_k}^2 = k$) were used instead.

The development for a second-order Gauss-Markov model is, not surprisingly, similar to that of a first-order model. The only differences are that the stochastic constraint equations are considerably more complex, and that they involve the two closest exposures rather than just the immediately adjacent one.

Random-walk random process model

A development of the stochastic constraints model was investigated by Madani and Shkolnikov (2005). In their work, it was recognised that whereas the GNSS errors are highly correlated for push broom scan-lines, they are likely much less correlated for frame images that are captured at larger intervals. They correspondingly concluded that the error behaviour for frame images may adequately be described by a simple random walk process. With this model, the errors in each co-ordinate component are assumed to behave according to

$$\dot{x} = w. \tag{4.26}$$

The equivalent discrete representation is

$$x(t_i) = x(t_{i-1}) + w_k. \tag{4.27}$$

from which the stochastic constraints follow directly,

$$\begin{aligned} 0 &= \Delta x_a^M(t_i) - \Delta x_a^M(t_{i-1}) \\ 0 &= \Delta y_a^M(t_i) - \Delta y_a^M(t_{i-1}) \\ 0 &= \Delta z_a^M(t_i) - \Delta z_a^M(t_{i-1}). \end{aligned} \tag{4.28}$$

Further developing this model, Madani and Shkolnikov (2005) recognised that their original assumption of non-correlated errors was, in fact, likely optimistic, and that the w noise term in Equation (4.27) would not be white noise. To accommodate the time correlations, they divided the noise in Equation (4.27) into a trend term d that accounts for the correlation, and a truly random component $\tilde{w}$,

$$x(t_i) = x(t_{i-1}) + d_k + \tilde{w}_k. \tag{4.29}$$

The stochastic constraint equations that result from this model can be used in least-squares collocation to solve for the unknown parameters. From Krakiwsky (1990), the collocation solution for the unknown parameters is

$$\boldsymbol{\Delta}\hat{\mathbf{x}} = -\Big[\mathbf{A}^T(\mathbf{C}_{dd} + \mathbf{C}_{ww})^{-1}\mathbf{A}\Big]^{-1}\mathbf{A}^T(\mathbf{C}_{dd} + \mathbf{C}_{ww})^{-1}\mathbf{w}, \tag{4.30}$$

where $\mathbf{C}_{dd}$ is the covariance matrix of the trend and $\mathbf{C}_{ww}$ the covariance matrix of the noise. Importantly, to solve for the parameters only the statistical properties of the trend – in the form of a covariance matrix – are required. To populate this covariance matrix, a simple exponential covariance function describing the autocorrelation of the trend d was used,

$$\rho = e^{-kn^2} \tag{4.31}$$

where ρ is the autocorrelation between two n-separated exposures. The parameter k was determined by regression using (it is believed) the co-ordinate differences between adjacent

exposures.

A concluding note on this approach's implementation: because a bundle adjustment is non-linear adjustment, iterations are required. In the first iteration, the trend d is not known. On subsequent iterations, however, an estimate of the trend is available using

$$\mathbf{d} = -\mathbf{C}_{dd}(\mathbf{C}_{dd} + \mathbf{C}_{ww})^{-1}\mathbf{A}\left(\boldsymbol{\Delta}\hat{\mathbf{x}} + \mathbf{w}\right), \tag{4.32}$$

where $\boldsymbol{\Delta}\hat{\mathbf{x}}$ are the parameter corrections from the previous iteration (Krakiwsky, 1990). The trend values so computed can be used to "correct" the stochastic observations.

Notes on approach

Despite their different theoretical underpinnings, the two stochastic constraint approaches given above are, effectively, numerically equivalent. In both cases the normal least-squares solution is modified by the addition of off-diagonal covariance to the observational covariance matrix. For the Gauss-Markov models, the bandwidth of this additional covariance is one or two, depending on whether a first-order or second-order model is used, respectively. For the random walk model with collocation, covariance is, in theory, added to all elements of the observational covariance matrix. In practice, however, the autocorrelation of the trend likely declines rapidly, and so only the near-diagonal terms need be added. In any case, the effect of the additional covariance is to essentially "relax" the network. This gives it an improved ability to accommodate GNSS errors that cannot be modelled conventionally.

Madani and Shkolnikov (2005) claim that their approach provides better results than the shift-and-drift or other polynomial-based approaches . However, this claim is always based upon the adjustment of networks with ground control. The performance with purely GNSS-controlled networks has not been established; indeed, it is not even clear if the approaches would work at all without ground control.

Finally, there is an obvious parallel between the random-process stochastic constraints method and Kalman Filtering of GNSS observations. In both cases a random process is

used to model the error dynamics. The only difference between the two approaches is that here the information about the error dynamics is applied in the adjustment simultaneously rather than sequentially as is done in the Kalman filter.

4.3.4 Perspective

All of the techniques described above for incorporating GNSS information in a photogrammetric adjustment have the same characteristics:

- The integration is always done in two steps: first the GNSS data is processed, and then it is used within a photogrammetric adjustment
- The integration is always done in the position domain
- The flow of information is strictly one-way: from the GNSS processor to the photogrammetric adjustment

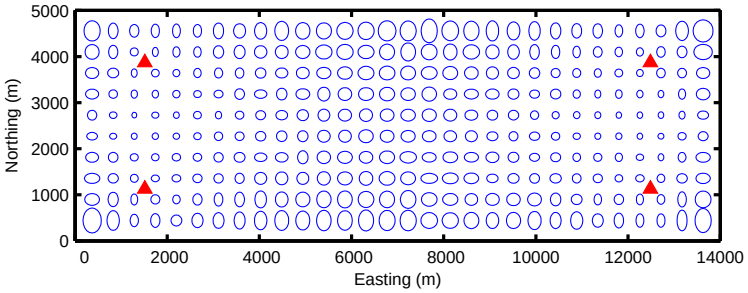
4.3.5 Benefits

The most often stated advantage of GNSS controlled photogrammetry is that it greatly reduces the requirement for ground control points (Kuntu-Mensah and Hintz, 2001; Mikhail et al., 2001; Meade, 2003). Establishing ground-control can be a costly and time-consuming process, particularly in remote areas with poor access. Therefore, reductions in ground-control can have a corresponding reduction in the time and cost of a mapping campaign. Using GNSS controlled photogrammetry, ground control can, in fact, potentially be eliminated altogether, providing that the resulting networks do not need to be tied to an existing ground datum and that the quality control that ground control points provide can be foregone (Mikhail et al., 2001). In practice, however, these conditions are rarely met, and most GNSS controlled aerial mapping campaigns still operate with some (minimal) control.

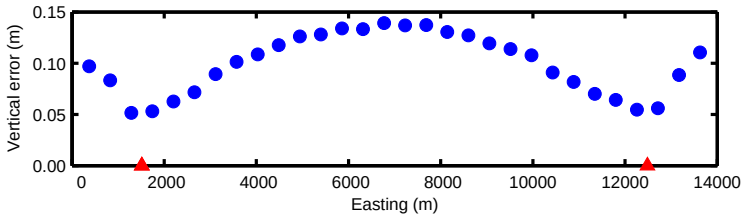
An additional and less often cited advantage of GNSS controlled photogrammetry is that it can provide accuracies that are both higher and more homogeneous than those available

from using ground control. This can be seen in Figures 4.10 and 4.11, where the results from two simulations of an aerial photogrammetric bundle adjustment are shown. For these tests, a block of imagery incorporating two strips of 15 images was simulated and adjusted. In the former figure, the block was controlled using 4 ground control points, whereas in the latter figure it was controlled using exposure station position observations. The upper figures show the horizontal check point errors in the form of error ellipses, and the bottom figures show the mean vertical errors along the east-west lines of check points. By inspection it can be seen that not only are the errors in the GNSS controlled simulation smaller, they are also more uniform across the entire network. In the ground controlled network accuracy is clearly related to the proximity of a object space point to ground control, but in the GNSS controlled block the accuracy only degrades at the edges of the block where there are fewer ray intersections. The reason for the superior accuracy of the GNSS controlled block is because essentially it has 30 evenly distributed control points (i.e., the GNSS exposure station positions), whereas the ground controlled network has only four. This situation is true in general; rarely will a block of imagery have a better distribution of ground control points than is available from GNSS exposure station position observations, and hence it is not surprising that GNSS controlled networks can be more accurate than their ground-controlled equivalents.

In addition to the two major advantages previously described, the use of GNSS controlled photogrammetry also has a number of smaller advantages over the traditional approach. For example, initial approximates for the exposure station position parameters in the bundle adjustment are directly available from the GNSS positions. In addition, by interpolating from the GNSS trajectory, approximate exposure azimuths are also available. The use of GNSS exposure station positions also improves the ability of the adjustment to detect poor or erroneous ground control points.

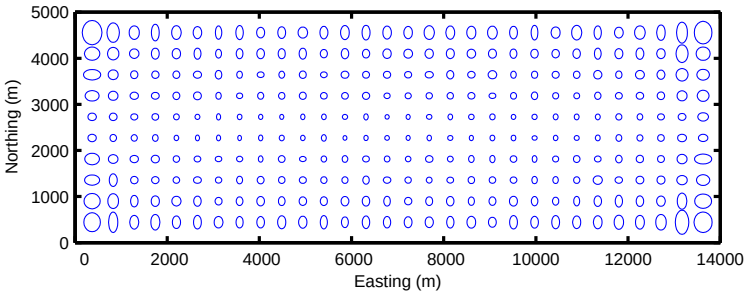


(a) Horizontal accuracy

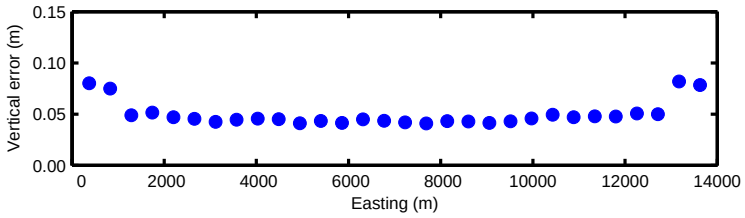


(b) Vertical accuracy profile

Figure 4.10: Ground controlled network



(a) Horizontal accuracy



(b) Vertical accuracy profile

Figure 4.11: GNSS controlled network

4.4 The Georeferenced Image Measurement Equation

When both the position and attitude of a camera can be determined from navigation sensors, the images from the camera are said to be *georeferenced*. The georeferencing information can be used in two ways: as additional parameter observations in photogrammetric bundle adjustments (as is done in GNSS-controlled photogrammetry), or for explicitly specifying the exterior orientation of images, from which object space co-ordinates can be determined by space intersection. The two approaches are termed Integrated Sensor Orientation (ISO) and Direct Sensor Orientation (DSO), respectively.

Unfortunately, and regardless of the approach, the image measurement equations developed in Section 4.1 are not directly compatible with georeferenced images. Those conventional collinearity equations relate an image measurement with a camera's position and attitude. In georeferenced images, however, the position and attitude sensors do not actually measure the camera's orientation as the position sensor cannot physically occupy the camera perspective centre, and the axes of the attitude sensor can never be perfectly aligned with the axes of the camera. This means that the measured position and attitude must be transferred to the camera through the camera-to-position-sensor lever-arm and the camera-to-attitude-sensor rotation matrix.

4.4.1 Derivation

Fortunately, it is straightforward to derive equations that explicitly relate external measurements of position and attitude with image measurements. Figure 4.12 shows the relationship between a position sensor (assumed to be a GNSS antenna), image measurement, and object space position. By examining this figure it can be seen that the co-ordinates of a point in mapping space can be expressed by

$$\mathbf{r}_P^m = \mathbf{r}_a^m + \mathbf{R}_b^m \mathbf{R}_c^b \left(\mu \mathbf{r}_p^c - \mathbf{r}_a^c \right). \tag{4.33}$$

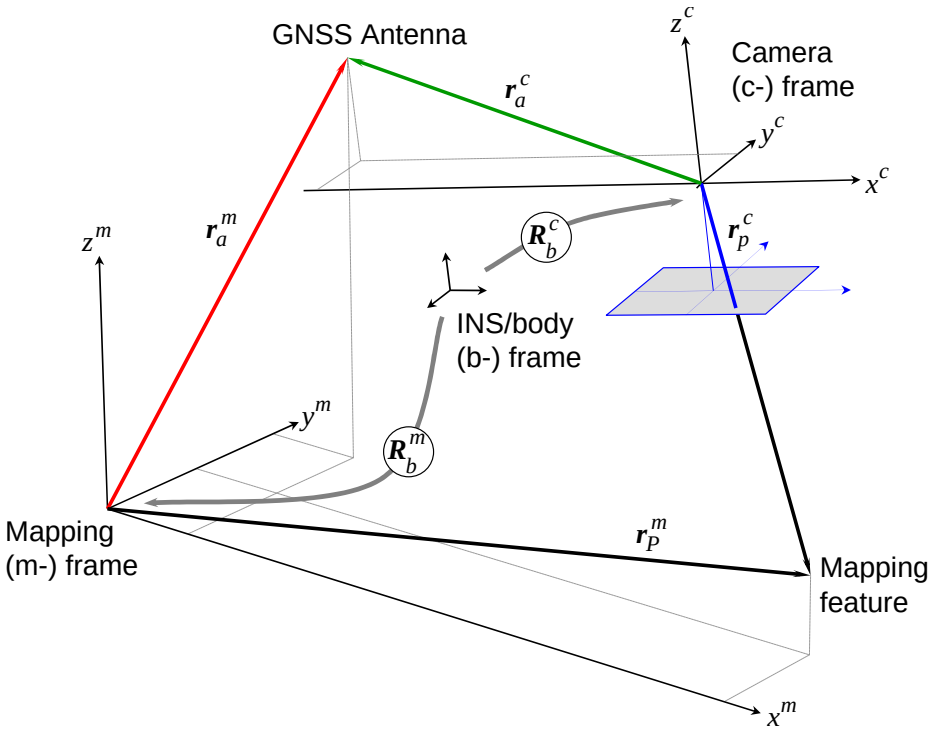


Figure 4.12: Georeferencing

In Equation (4.33), $\mathbf{R}_c^b$ is the rotation matrix that accounts for the misalignment between the axes of the attitude sensor (assumed to be an Inertial Measurement Unit (IMU)) and the axes of the camera. This matrix is called the *boresight matrix*, and the angles used to form it are termed the *boresight angles*. $\mathbf{R}_b^m$ is the rotation matrix that aligns the attitude sensor's axes with those of the mapping frame. This matrix is most commonly formed using roll, pitch, and yaw (or azimuth) angles output by the attitude sensor.

Reversing the relationship expressed by Equation (4.33) so that it becomes explicit in the camera co-ordinates $\mathbf{r}_p^c$ yields,

$$\mathbf{r}_p^c = \mu^{-1}\left[\mathbf{R}_b^c\mathbf{R}_m^b\left(\mathbf{r}_P^m - \mathbf{r}_a^m\right) + \mathbf{r}_a^c\right]. \tag{4.34}$$

Eliminating the third equation in this system of equations results in a pair of equations relating the x and y camera co-ordinates of a point with the position sensor's position and

the elements of the attitude sensor's $\mathbf{R}_b^m$ matrix. Fully expanded, these *georeferenced image measurement equations* are

$$x_p^c = z_p^c \frac{\begin{array}{r}(r_{b11}^c r_{m11}^b + r_{b12}^c r_{m21}^b + r_{b13}^c r_{m31}^b)\,(x_P^m - x_a^m) \\ + (r_{b11}^c r_{m12}^b + r_{b12}^c r_{m22}^b + r_{b13}^c r_{m32}^b)\,(y_P^m - y_a^m) \\ + (r_{b11}^c r_{m13}^b + r_{b12}^c r_{m23}^b + r_{b13}^c r_{m33}^b)\,(z_P^m - z_a^m) + x_{a/P}^c\end{array}}{\begin{array}{r}(r_{b31}^c r_{m11}^b + r_{b32}^c r_{m21}^b + r_{b33}^c r_{m31}^b)\,(x_P^m - x_a^m) \\ + (r_{b31}^c r_{m12}^b + r_{b32}^c r_{m22}^b + r_{b33}^c r_{m32}^b)\,(y_P^m - y_a^m) \\ + (r_{b31}^c r_{m13}^b + r_{b32}^c r_{m23}^b + r_{b33}^c r_{m33}^b)\,(z_P^m - z_a^m) + z_{a/P}^c\end{array}} \tag{4.35a}$$

and

$$y_p^c = z_p^c \frac{\begin{array}{r}(r_{b21}^c r_{m11}^b + r_{b22}^c r_{m21}^b + r_{b23}^c r_{m31}^b)\,(x_P^m - x_a^m) \\ + (r_{b21}^c r_{m12}^b + r_{b22}^c r_{m22}^b + r_{b23}^c r_{m32}^b)\,(y_P^m - y_a^m) \\ + (r_{b21}^c r_{m13}^b + r_{b22}^c r_{m23}^b + r_{b23}^c r_{m33}^b)\,(z_P^m - z_a^m) + x_{a/P}^c\end{array}}{\begin{array}{r}(r_{b31}^c r_{m11}^b + r_{b32}^c r_{m21}^b + r_{b33}^c r_{m31}^b)\,(x_P^m - x_a^m) \\ + (r_{b31}^c r_{m12}^b + r_{b32}^c r_{m22}^b + r_{b33}^c r_{m32}^b)\,(y_P^m - y_a^m) \\ + (r_{b31}^c r_{m13}^b + r_{b32}^c r_{m23}^b + r_{b33}^c r_{m33}^b)\,(z_P^m - z_a^m) + z_{a/P}^c\end{array}} \tag{4.35b}$$

These equations are much more complex than Equations (4.7), the conventional collinearity image measurement equations. However, like the collinearity equations, they can be expressed in the simple form

$$x_p^c = z_p^c \frac{x_{c/P}^c}{z_{c/P}^c} \tag{4.36a}$$

$$y_p^c = z_p^c \frac{y_{c/P}^c}{z_{c/P}^c} \tag{4.36b}$$

where $x^c_{c/P}$, $x^c_{c/P}$, and $x^c_{c/P}$ are the elements of the $\mathbf{r}^c_{c/P}$ vector, which is given by

$$\mathbf{r}^c_{c/P} = \mathbf{R}^c_b \mathbf{R}^b_m \left(\mathbf{r}^m_P - \mathbf{r}^m_a\right) + \mathbf{r}^c_a. \tag{4.37}$$

The georeferenced image measurement equations developed here are essentially a generalisation of the conventional collinearity equations (or, alternatively, the collinearity equations are a special case of the georeferenced image measurement equations). The equations could be even further generalised by adding, for example, terms to account for the transformation between the image and camera co-ordinate frames, or an additional rotation matrix accounting for attitudes being measured relative to a local-level frame rather than relative to an ECEF frame.

Finally, it should be noted that the georeferenced image measurement equations should not be termed collinearity equations, as the position sensor is almost certainly not collinear with the image measurement and object space point.

4.4.2 Benefits and Drawbacks

The chief advantage of the georeferenced image measurement equations – indeed, the reason they were developed – is that they simplify the inclusion of GNSS measurements in a combined simultaneous adjustment with image measurements. This is because in the conventional form of the GNSS observation equations the GNSS measurements are themselves functions of the antenna positions. A combined adjustment is one of the focuses of this research, and more detail on this topic will be given in Chapter 5.

The georeferenced image measurement equations also have a number of benefits when used with multiple-camera photogrammetric systems. In adjustments using the conventional image measurement equations, each camera in such a system must be given its own set of position and attitude parameters for each exposure. At each set of exposures the camera group is then made to act as a rigid system through the addition of relative orientation

constraints to the adjustment, with examples of such constraints being found in He et al. (1992), Chaplin (1999), or Ellum (2002). With the georeferenced image measurement equations, all cameras in such systems can be referenced to a single position and attitude. This both reduces the number of parameters in the adjustment, and negates the requirement for relative orientation constraints.

Finally, there are several smaller advantages to using the georeferenced image measurement equations. First, because the GNSS positions are one of the quantities being adjusted, the GNSS position observation equations are simplified by the removal of the lever-arm term. The removal of this term generalizes the position observation equation, allowing it to be used for any type of position – i.e., the same observation equation can be used for both observed exposure station positions and weighted control points. Adjusting the GNSS positions directly also means that they are one of the quantities output by the adjustment. This allows for easy comparison with the input positions, which in turn simplifies the analysis of the results.

Naturally, the georeferenced image measurement equations have some drawbacks. Calculation of the $\mathbf{r}^c_{c/P}$ terms required for their evaluation requires more operations than is needed with the conventional image measurement equations. Also, derivation of the partial derivatives required for least squares adjustment is more involved. However, differentiating $\mathbf{r}^c_{c/P}$ using the chain rule and applying the quotient rule to the resulting differentials makes the derivation manageable.

4.5 Lever-arm and boresight calibrations

A key issue in direct georeferencing – with or without the georeferenced image measurement equations – is the determination of the camera-to-antenna lever-arm $\mathbf{r}^c_a$ and the camera-to-attitude sensor boresight matrix $\mathbf{R}^c_b$. The processes of determining these quantities are termed a lever-arm and boresight calibration, respectively.

4.5.1 Lever-arm calibration

The most common method for determining the lever-arm is to measure it using conventional survey methods. Unfortunately, the accuracy of this technique is limited by the inability to directly observe the phase and perspective centres of the antenna and camera, respectively. In the former case this limitation can be overcome by making measurements to the base of the antenna and using antenna calibration values, and in film-based cameras the perspective centre's position can be indirectly externally determined by making measurements of the camera's fiducial marks. However, this perspective centre is actually the rear nodal point and not the front nodal point truly required. Also, in digital cameras it is not normally possible to make direct measurements to the image plane, and consequently such a procedure is not possible. For these reasons, the accuracy of a lever-arm externally measured is limited to the centimetre-level

An alternative "pseudo" measurement technique is to use the difference in between positions determined by GNSS observations and those resulting from a photogrammetric space resection or bundle adjustment. However, the accuracy of this technique is dependent upon finding a calibration field that is suitable for both GNSS and photogrammetry – i.e., a field that minimises GNSS errors such as multipath, and has dense and well-distributed targets for the photogrammetric measurements. If such a target field can be found, then the offset vector in the camera co-ordinate frame can be calculated using

$$\mathbf{r}_a^c = \mathbf{R}_m^c \left(\mathbf{r}_a^m - \mathbf{r}_c^m \right). \tag{4.38}$$

When several exposures are available the accuracy of the calibration can be improved by averaging, and, if the covariance of one or both position vectors is known, by weighted averaging. Because of the difficulties in obtaining an accurate exposure position in aerial photogrammetric adjustments, however, this technique is really only practical for land-based systems.

Another method of determining the lever-arm offsets is to include them in a bundle adjustment as unknown parameters, either as part of the parameter set of the conventional GNSS position observation equation, or as part of the parameter set of the georeferenced image measurement equations. However, opinion on this approach is mixed. For instance, Mikhail et al. (2001) indicates that the offsets are usually included, but Ackermann (1992) claims that the offsets cannot be included as they result in singularities in the adjustment. For airborne systems, the truth is somewhere in the middle. The offsets are highly correlated with both the interior and exterior orientation parameters, particularly with the focal length and exposure station positions. Because of this correlation, offsets determined in the adjustment may not be very accurate (Ellum and El-Sheimy, 2002). In close-range photogrammetry, however, convergent imagery decorrelates these parameters and makes the recovery of the offsets more reliable. In any case, to include the offsets in the adjustment as unknowns it is necessary to provide parameter observations of the exposure positions. Otherwise the effects of focal length and z-offset cannot be separated, and the adjustment is rendered singular.

4.5.2 Boresight calibration

A boresight calibration refers to the determination of the rotation matrix $\mathbf{R}_b^c$ that relates the axes of the attitude sensor to the axes of the camera. In the most common method of performing this calibration it is necessary to explicitly determine both the $\mathbf{R}_c^m$ and $\mathbf{R}_c^b$ matrices. The former matrix is determined by photogrammetric resection or in a bundle adjustment, and the latter matrix is determined from measurements made by the attitude sensor. With both rotation matrices available, $\mathbf{R}_b^c$ can be calculated using

$$\mathbf{R}_b^c = \mathbf{R}_b^c \mathbf{R}_m^b. \tag{4.39}$$

Although this calculation can be done with a single exposure, it is, obviously, advantageous to use multiple exposures and to average the results. Of course, it is not possible to simply average the individual elements of the $\mathbf{R}_b^c$ rotation matrices from each exposure station, as the resulting rotation matrix would almost certainly not be orthogonal. Instead, a set of Cardan or Euler angles must be extracted from each exposure station's $\mathbf{R}_b^c$ matrix, those angles averaged, and a final $\mathbf{R}_b^c$ reconstructed. A problem with this procedure, however, can arise when averaging negative and positive angles or angles that straddle quadrant boundaries. For example, averaging 359 degrees and 1 degree will incorrectly yield 180 degrees, and averaging 270 degrees and -90 degrees will incorrectly yield 90 degrees. To overcome this problem the x $(= \sin\alpha)$ and y $(= \cos\alpha)$ components of each angle can be averaged and a final angle reconstructed $(\alpha = \arctan\frac{x}{y})$. It should be noted that simply rectifying the angles between 0 and 2π does not solve this problem (consider the first example).

In practice, images at the edge of the photogrammetric block can be excluded from the above averaging calculation because their adjusted attitude parameters may be less accurate than those images closer to the middle of the block (Škaloud, 1999). An alternative to this procedure is to weight the contribution of each exposure according to its angular standard deviations coming from the adjustment.

Bäumker and Heimes (2001) presented an analogue to the above technique where instead of averaging the angles, an unweighted least-squares adjustment was used to estimate small angular misalignments. This was done after the bundle adjustment, and treated both the orientations from the adjustment and the measured orientations as fixed. Unfortunately, this method, although conceptually more complicated than the one above, will likely give exactly the same results – if not worse, because of the small-angle approximation. Also, it is only suitable for determining small misalignments between sensors, and thus requires a reasonably accurate initial estimate for $\mathbf{R}_b^c$.

Like the lever-arm calibration, it is also possible to determine the $\mathbf{R}_b^c$ matrix in a bun-

dle adjustment (Mostafa, 2001; Ellum and El-Sheimy, 2002; Pinto and Forlani, 2002). In order to do this, the matrix is parameterised and the parameters are adjusted for in the adjustment. For example, if α_1, α_2, and α_3 are three angles, then $\mathbf{R}_b^c$ may be described by

$$\mathbf{R}_b^c = \mathbf{R}_z(\alpha_3)\mathbf{R}_y(\alpha_2)\mathbf{R}_x(\alpha_1). \tag{4.40}$$

Of course, any set of Cardan or Euler angles may be used, or a different set of parameters altogether may be used. Ellum and El-Sheimy (2002) showed that including the boresight angles in the adjustment was the best option for performing a boresight calibration. A disadvantage of this procedure is that the addition of these angles necessitates rather fundamental changes to the implementation of the adjustment, as the collinearity equations become functions of six angles instead of just three. This, in turn, makes the linearisation of the collinearity equations considerably more complex.

Unlike the lever-arm, it is not really feasible to determine the misalignment angles between the camera and IMU by making external measurements to the two instruments. The difficulty here is that it is not generally possible to get accurate orientations of either the IMU's or the camera's axes from external measurements to these sensors.

Chapter 5

Implementation of new integration strategies

The previous chapter reviewed the current approaches for integrating GNSS and photogrammetric data. In this chapter new integration strategies are introduced. Two strategies were implemented and details of their implementation are provided. Particular focus is given to aspects of software-implementation that are rarely covered in Geomatics publications.

5.1 Review of existing integration technique

Before introducing the new GNSS/photogrammetry integration techniques, it is worthwhile to review the limitations of the current position-observations integration techniques. In the current approach, the GNSS data is first processed in an extended Kalman Filter, resulting in exposure station positions. These positions are subsequently used in a bundle adjustment with various error modelling strategies. As shown in Figure 5.1, the transfer of information is only in one direction: the bundle adjustment benefits from the GNSS data, but the GNSS filter does not benefit from the photogrammetric data. Furthermore, with this workflow two software packages are required. Transferring results between two packages is at best tedious;

at worst it can introduce significant delays and/or errors. It also requires technicians to be conversant with multiple software packages.

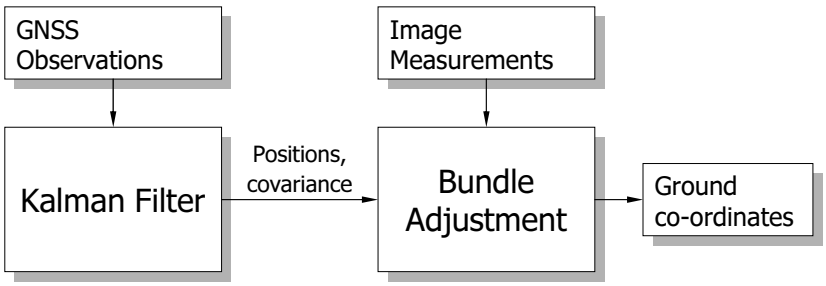


Figure 5.1: Position-observations integration strategy

5.2 New integration techniques

There are three alternative GNSS/photogrammetric strategies that are, conceptually, straightforward developments of existing techniques. In order of increasing complexity, these are:

1. Inter-processor communication between the GNSS Kalman Filter and the bundle adjustment.

2. A combined least-squares adjustment of both photogrammetric and GNSS data.

3. A combined Kalman filter incorporating both photogrammetric and GNSS data.

5.2.1 Inter-processor communication

The most basic development on the current one-way position observations integration strategy is one where the approach is modified so that photogrammetric bundle adjustment feeds Co-ordinate Updates (CUPTs) back into the GNSS Kalman filter. In this technique, shown in schematic form in Figure 5.2, the GNSS processor no longer works in isolation from the photogrammetric processor; instead, it is "aided" by positions output by the bundle adjustment.

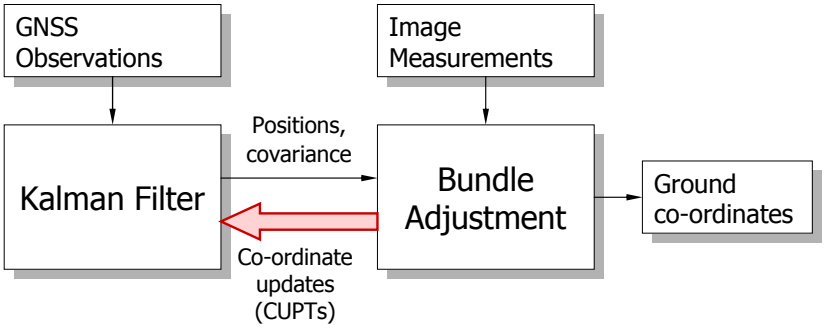


Figure 5.2: Structure of the Combined Adjustment Integration Approach

This integration approach's primary advantage is its ease-of-implementation: the bundle adjustment already outputs the positions and covariance matrices of the exposure stations and a GNSS Kalman filter can easily be adapted to incorporate the CUPTs. The GNSS filter should benefit from the photogrammetric aiding, and if more accurate or more reliable exposure station positions result, then the bundle adjustment results should be improved as well. Unfortunately, the integration is still only at position level, and two processing packages are still required.

The inter-processor communication, being the simplest development of the current GNSS/photo-grammetric integration technique, was implemented and tested. Section 5.4 outlines some implementation aspects, and results from testing of the approach are in Chapter 6.

5.2.2 Combined Adjustment

Integration of the GNSS and photogrammetric data streams can be done at the measurement level with a single processing package in a combined adjustment. In this approach, both the image measurements and the raw GNSS code and carrier phase range observations are used in a single simultaneous least-squares adjustment. A conceptual overview of the combined adjustment is shown in Figure 5.3.

This integration approach has a number of advantages over the existing position-observations integration approach:

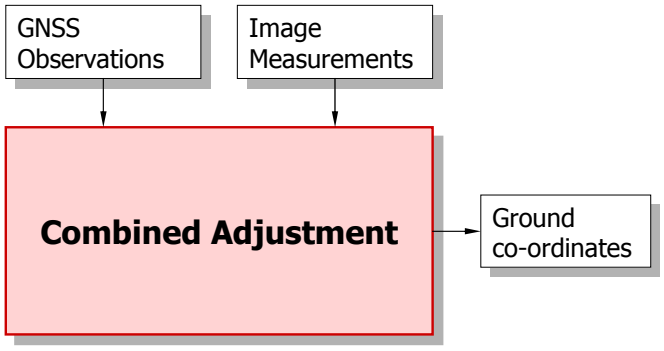


Figure 5.3: Structure of the Combined Adjustment Integration Approach

- The exposure-station positions and ambiguity resolution should be improved by relative information provided by the photogrammetric block.
- The additional redundancy provided by the increased number of observations and the independent nature of the two observation sets should improve the ability to do outlier-detection for both sets of observations.
- GNSS data and processing requirements, such as requiring four observations or having one receiver at a known position, are relaxed or removed. For the examples given, the rank deficiencies that would arise in GNSS-only adjustments in the same configurations can be removed by the addition of the photogrammetric data in the combined adjustment.

In addition, a combined adjustment has the practical benefit to only having to use a single software package to do all the processing.

Due to the benefits outlined above, the combined adjustment was implemented and tested. Section 5.3 details some aspects of its implementation, while the following chapter contains results from its application.

5.2.3 Combined Filter

A final integration option, depicted in Figure 5.4, is a combined Kalman filter. The advantage of this approach is that a kinematic model relating GNSS navigation quantities

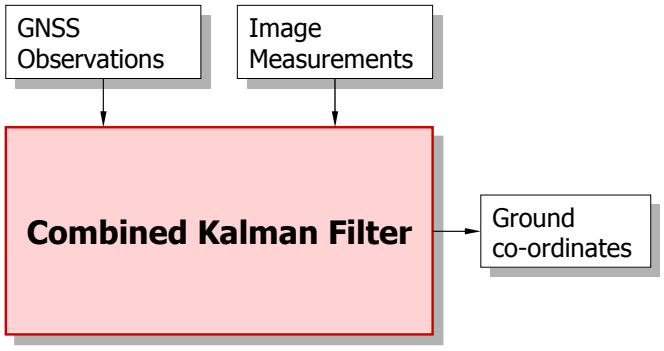


Figure 5.4: Structure of the Combined Filter Integration Approach

(positions, velocities, accelerations) can be used. Unfortunately, because the filter's state vector would have to include all of the photogrammetric and tie/pass point parameters in addition to the GNSS states, the computational requirements would be enormous. It is also significantly more difficult to implement than either of the two previous approaches. For these reasons, this approach was not implemented.

5.3 Combined Adjustment - Implementation Aspects

Despite being the more fundamentally different integration strategy, the combined filter was actually implemented before the inter-processor communication integration strategy. Also, the inter-processor technique reused the bundle adjustment functionality and GNSS code from the combined adjustment. Accordingly, implementation of the combined adjustment is discussed before the inter-processor communication.

There are a number of hurdles that must be overcome to implement a combined GNSS/ photogrammetric adjustment, but none is more significant than the sheer amount of software development required. A metric of the effort involved is the more than 101,000 lines of code of which the software currently consists. The effort required can be illustrated using the basic COnstructive COst MOdel (COCOMO) 81 software estimation model (Boehm, 1981): appropriately considering the software project as "organic", and using the logical lines of code (i.e., 31,400 statements), the basic COCOMO estimated effort is $2.4 * 31.4^{1.05} = 89.5\,\text{man} - \text{months}$

(7.5 years). Fortunately, some of the code was re-used from an existing photogrammetric adjustment (see Ellum, 2002), but the effort required was still considerable.

Internally, the combined adjustment is divided into several logical components. The code itself is also divided into a hierarchy of namespaces that reflect the logical components; this hierarchy is shown in Figure 5.5. The following major namespaces and components can be identified:

- A root adjustment: Provides system-wide functionality. This includes controlling overall adjustment flow and managing shared resources like intermediate output file and debugging facilities.
- A photogrammetric adjustment: Handles the adjustment of photogrammetric data.
- A GNSS adjustment: Handles the adjustment of GNSS data.
- A terrestrial survey adjustment: Handles the adjustment of survey data.
- Text input: Parses and handles input of adjustment data. This includes observations, constraints, constants, configuration, etc.
- Text output: Outputs adjustment results in the same format as the input file. The same format is used to facilitate re-use of adjustment data, and so that syntax-highlighting editors can use the same schemes for both the input and output files.
- MATLAB output: Outputs adjustment results in files compatible with MATLAB. These files can be used to conduct further analysis or visualisation in MATLAB.
- HTML output: Outputs adjustment results in Hypertext Markup Language (HTML) format. The HTML output files are easier to read and enable interactivity (like navigation or sorting) not possible with the text output files.
- OpenGL visualisation: Produces visualisations of the adjustment networks.

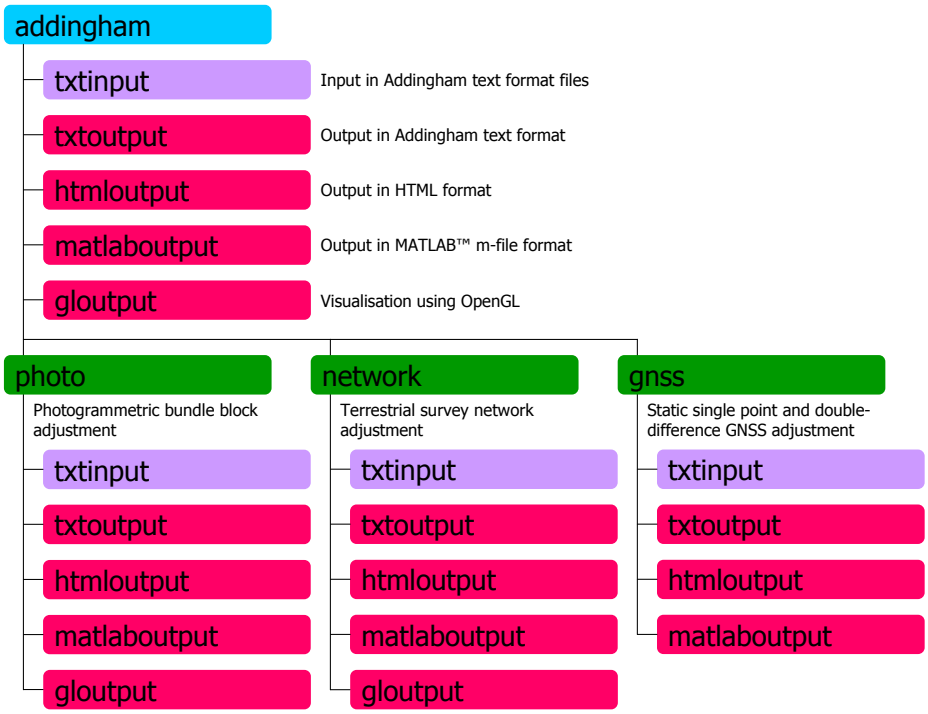


Figure 5.5: Combined Adjustment Namespace Hierarchy

5.3.1 Use of Polymorphism

The combined adjustment was programmed using modern object-oriented and C++ techniques, and heavy use is made of concepts such as template meta-programming, encapsulation, inheritance, and polymorphism. Polymorphism, in particular, is used to genericise much of the adjustment's internal operation. One of these has already been identified: the division of the program into child adjustments. Other interesting examples described below are the implementation of multiple solution engines and of multiple exposure attitude parametrisations.

Child Adjustments

The combined adjustment conceptually contains three mostly-independent child adjustments:

- A photogrammetric bundle adjustment

- A GNSS network adjustment
- A terrestrial surveying adjustment

This conceptual division was realised into a practical division by creating a child adjustment interface that the individual adjustments implement. Each adjustment quite naturally performs a great number of very specialised operations; consequently, the only behaviour that could easily be generalised was the sequence of adjustment steps. For instance, the methods required by each child adjustment for each iteration of the adjustment were:

1. Pre-iteration calculations
2. Process observations
3. Process constraints
   - *Solve least-squares system of equations* (done by root adjustment object)
4. Update parameters
5. Check convergence

These and the other child adjustment interface methods are visible in its collaboration diagram of Figure 5.6.

Each child adjustment is completely unaware of the other child adjustments. The only connection between the three adjustments are the object space position parameters maintained by the root adjustment object. The separation between adjustments is comprehensive: not only are the adjustments their own classes, they are also in their own namespaces, source files, and even static libraries. Indeed, it would easily have been possible to make the individual adjustments loadable at run-time, but this provided no perceived benefit and was not implemented.

Figure 5.6: Collaboration diagram for child adjustment

Solvers

In Section 2.1.3, several different numerical techniques for getting a least-squares solution were derived. Two of these techniques were implemented in the combined adjustment: Cholesky and QR decomposition. Additionally, several variations of each were realised, so that the total set of solvers was:

- Normal equations solver: Uses the normal equations and Cholesky decomposition.
- Symmetric normal equations solver: Uses the normal equations and Cholesky decomposition. Takes advantage of the symmetry of the normal matrix to reduce memory requirements.
- Reduced normal equations solver: Uses the normal equations and Cholesky decomposition. Forms the reduced normal equations (see Granshaw, 1980) to reduce computational requirements.
- Given's solver: Uses Given's rotations to solve the least squares by QR decomposition (Golub and Loan, 1996).
- Gentleman's solver: Uses Gentleman's (i.e., modified or fast Givens) rotations to solve the least squares by QR decomposition (Gentleman, 1973).

All of the solvers process sets of (correlated) observations sequentially by realising a "process observations" method of the generic solver interface. The QR decomposition solvers are inherently sequential, while the normal equations solvers use summation of normals. Other methods in the solver interface handle constraints, calculate the least squares solution, and determine the parameter covariance once the solution has converged.

Exposure Attitudes

There are many ways that rotations in $\mathbb{R}^3$ can be parametrised: Direction cosines, quaternions, various Euler (Cardan) angles, Rodrigues parameters etc. In the combined adjust-

ment, the exposure attitudes were given a generalised interface so that any rotation representation could be used. The interface had two key methods: one method that converted the parameters into a rotation matrix ($\mathbf{R}_M^b$), and another method that calculated the Jacobian with respect to the georeferenced image measurement equation. The specific attitude parametrisations implemented were:

- The ω, ϕ, and κ rotation angles: These parameters are the Euler angle sequence most commonly used in photogrammetry.

- The roll, pitch, and yaw (or azimuth) angles using a right-front-up body frame: These parameters are the Euler angle sequence typically reported by an Inertial Navigation System (INS). Their intuitiveness is also convenient for providing parameter approximates in close-range photogrammetry.

- The roll, pitch, and azimuth angles using a front-right-down body-frame: These parameters share the same benefits as the above set of angles. The front-right-down body frame is that more commonly encountered in the literature (and is an accepted standard, defined in ARINC, 1985), while the right-front-up body frame has historically been taught and used at University of Calgary (cf. Schwarz and Wei, 2000).

- The Rodrigues parameters: Unlike the Euler angles, these parameters do not require cosines to be evaluated. Thus, they require less computations.

Because of the generic interface, other rotation representations could easily be added; for example, quaternions or direction cosines.

The use of polymorphic exposure attitudes created an output problem. The attitudes are stored in a single array (as pointers). When it comes to outputting the adjustment results this array has to be traversed and the data for the individual attitudes output. The problem is that the different attitude parametrisations have different output. For example, the various Euler angle representations have different names for their angles, while

a quaternion attitude has a different number of parameters. Thus, the output routine needs some way of determining the type of attitude so that it can output it appropriately. One way for the attitude type to be determined would be using Run-Time Type Information (RTTI). For performance reasons, however, RTTI is generally frowned upon and was not pursued. Instead, the "visitor" design pattern was used. The visitor pattern essentially enables virtual functions to be added to an existing polymorphic class hierarchy. It is implemented by providing a visitor class that uses function overloads,

```
struct AttitudeVisitor
{
  virtual void visit( const Attitude& ) {};
  virtual void visit( const KappaPhiOmegaAttitude& ) {};
  virtual void visit( const RollPitchAzimuthAttitude& ) {};
  virtual void visit( const RodriguesAttitude& ) {};
  virtual void visit( const QuaternionAttitude& ) {};
};
```

and adding an `accept` method that uses the visitor to the attitude interface,

```
struct Attitude
{
  virtual void accept( AttitudeVisitor& v ) const
    { v.visit(*this); }
};
```

Each derived attitude implements the `accept` method.

```
struct KappaPhiOmegaAttitude : public Attitude
{
  virtual void accept( AttitudeVisitor& v ) const
    { v.visit(*this); }
};

struct RollPitchAzimuthAttitude : public Attitude
{
  virtual void accept( AttitudeVisitor& v ) const
    { v.visit(*this); }
};

// ...
```

Specialised visitors can be derived from the generic visitor. For instance, an output visitor,

```
struct AttitudeOutputVisitor : public AttitudeVisitor
{
```

```
  void visit( const attitudes::KappaPhiOmegaAttitude& attitude )
       { /* Do specialised output */ }
  void visit( const attitudes::RollPitchAzimuthAttitude& attitude )
       { /* Do specialised output */ }
  void visit( const attitudes::RodriguesAttitude& attitude )
       { /* Do specialised output */ }
  void visit( const attitudes::QuaternionAttitude& attitude )
       { /* Do specialised output */ }
};
```

To perform specialised output with this visitor, the `accept` method of the attitude interface is called with a visitor as the parameter,

```
Attitude* attitude = new KappaPhiOmegaAttitude( /* ... */ );
attitude.accept( AttitudeOutputVisitor() );
```

Because the `accept` method is virtual, the derived attitudes' `accept` methods are called (via runtime binding). These, in turn, call the appropriate `visit` routines. Run-time type-dependant behaviour is realised, without the overhead of RTTI.

5.3.2 Network Visualisation

Both the input and output data of the combined adjustment can be visualised in 3D. Visualisation of the combined adjustment's networks serves several practical purposes. For instance, it allows for fast examination of network connectivity. Positions with few connections to the network, or whose connections have poor geometry, can be quickly identified. Visualisation of outlying observations can help identify problem observations or wrongly fixed or incorrectly weighted parameters. The most common use for the visualisation feature, however, was to check the initial exterior orientation approximates that were either specified in the input file or generated by the 4-or-more-point space resection algorithms. The problem of providing initial approximates can be particularly troublesome when using the georeferencing image measurement equations with a non-identity camera-to-body rotation matrix, or when using the unintuitive ω, ϕ, and κ rotation angles in a close-range photogrammetric network.

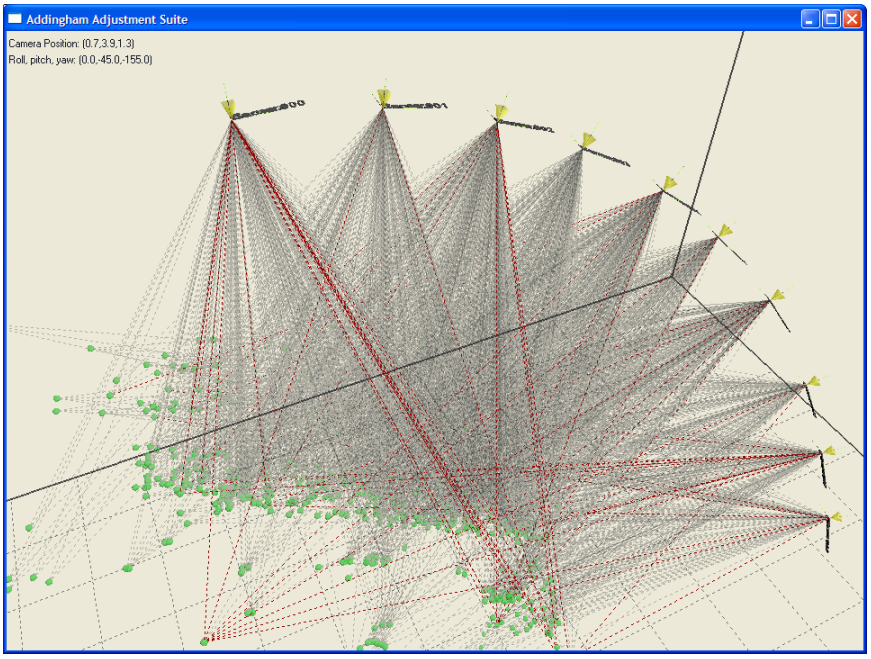
For 3-D visualisation on a Windows platform, there are basically two Application Program Interfaces (APIs) to choose from: OpenGL and Direct3D. Numerous other 3-D engines exist, but they are built on top of these two APIs and their additional functionality is normally intended for gaming or other high performance visualisations. Because platform-independence was one of the aims of the software, and because the learning curve for Direct3D is steeper, the visualisation was implemented using OpenGL. Since OpenGL uses a 3-D cartesian model space, creating the visualisation using OpenGL is reasonably straightforward: the network's real-world co-ordinates can be used directly when creating the lines, polygons, or other graphics primitives.

Two examples of the combined adjustment's network visualisation are shown in Figure 5.7.

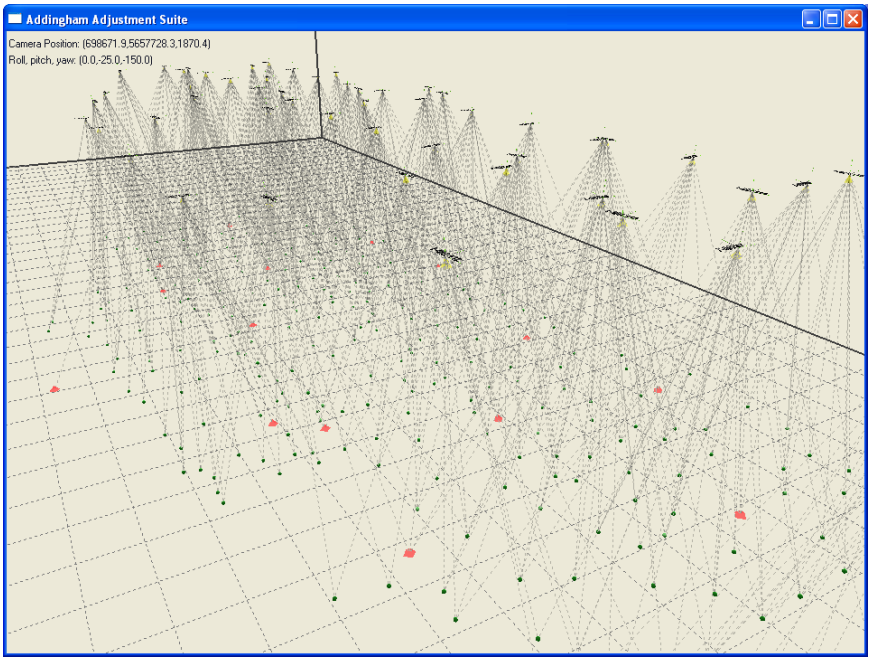
5.3.3 GNSS-specific Implementation Notes

The GNSS processor used in the combined adjustment has a number of idiosyncrasies when compared with other GNSS processors. To begin with, since the exposure events don't coincide with GNSS measurements, the processor can interpolate measurements between GNSS measurement epochs. Polynomial interpolation is used, and tests have shown that linear interpolation causes negligible to non-existent degradation in positioning results. The GNSS adjustment has also been designed from the outset so that multiple (i.e., more than two) GNSS stations can be used simultaneously. While this, in itself, is not too unusual, it is rather unique that none of the stations need to have fixed co-ordinates. Instead of fixed GNSS control, the datum for the entire network can be controlled by photogrammetric ground control, or from fixed positions tied in by survey observations.

It should be emphasised that all the unknown parameters in the combined adjustment, including the GNSS specific parameters, are solved for in a batch (simultaneous) adjustment. This is in contrast to most GNSS processing software, which, even for static periods, uses a Kalman filter operating sequentially in time. The batch adjustment includes both the



(a) Close range network



(b) Aerial network

Figure 5.7: Visualisation of photogrammetric networks

static and kinematic measurement epochs. Even though the adjustment only operates with discrete epochs of GNSS data with no time-dependent connecting dynamic model, it is still necessary to traverse sequentially through each GNSS data file. This is required in order to perform carrier phase smoothing of the code ranges, interpolate observations, detect cycle-slips that cause ambiguities, and track base satellite changes. In fact, each data stream (whether on disk or in memory) must be traversed at least 4 times, once for each of the following tasks:

1. Determine ambiguities
2. Perform iteration
3. Calculate residuals
4. Flag outlying residuals

A two-pronged approach to partial fixing was adopted. As proposed in Section 3.7.2, fixing is only done with the subset of (decorrelated) ambiguities whose bootstrapping lower bound success rate is greater than some threshold. If the fixed ambiguities fail the ratio test, then the ambiguity with the largest conditional variance is removed from the set, and fixing attempted again. The removal of least-precise ambiguity and fixing repeats until either the ratio test passes, or until the number of ambiguities to be fixed falls below a threshold. The ambiguities success rate, ratio test, and fixing minimum thresholds are all user-defined; typical values for each are 99%, 2.0, and 4, respectively. Fixing is not attempted until the ambiguities have converged, thereafter, fixing is attempted at each iteration. Because fixing ambiguities at one iteration will lower the conditional variances for the ambiguities in the following iteration, it is possible that more ambiguities will be able to be fixed in later iterations.

The application of ambiguity constraints follows that of Radovanovic (2002). In this

approach, the ambiguities, once fixed, are constrained with

$$\check{\mathbf{n}} = \mathbf{S}\mathbf{Z}^T\mathbf{n} = \mathbf{G}_{\check{\mathbf{n}}}\mathbf{n} \tag{5.1}$$

where $\mathbf{S}$ is the partial-fixing selector matrix and $\mathbf{Z}$ is the decorrelating transformation matrix. If all of the ambiguities are fixed, $\mathbf{S}$ is the identity matrix. With this strategy, the actual (i.e., not decorrelated) integer ambiguities are never calculated. Indeed, if the ambiguities are only partially fixed then the actual integer ambiguities cannot be calculated, since the product of the integral $\left(\mathbf{Z}^T\right)^{-1}$ and the mixed real-integer vector of partially-fixed and not fixed ambiguities will not, in general, be integral. Were the actual integer ambiguities required, another decorrelating transformation would have to be calculated for only the ambiguities being partially fixed.

Other GNSS-specific implementation aspects of the combined adjustment are:

- The tropospheric delay is estimated using the Black, UNB2, UNB3, or UNB4 tropospheric delay models; see Section 3.3.3.

- When undifferenced pseudoranges are used in the adjustment, satellite position and clock errors are mitigated using precise ephemeris and clock corrections; see Sections 3.3.1 and 3.3.2. With double-differenced observations the broadcast ephemerides and clocks were used, again due to the small base-remote separation. The calculation of satellite positions was hidden behind a 'SpaceSegment' interface, from which precise and broadcast space segments were derived and implemented.

- When double-differenced carrier phase observables are used, the L1 and L2 ambiguities are resolved and the L1 and L2 phase observables are used. In other words, no phase combinations are employed.

- GNSS measurements at the exposure times are interpolated from the GNSS data streams. Polynomial interpolation is used; the polynomial order and number of points used in

its fit is user-definable, but invariably a linear polynomial based upon the two closest epochs was used since no noticeable improvement was observed with higher-order polynomials.

- Ambiguity resolution was done by a decorrelating transformation followed by an integer least-squares search (i.e., the LAMBDA method). The implementation of the ambiguity resolution was essentially taken from de Jonge and Tiberius (1996). Modifications were done to the decorrelating transformation so the $\mathbf{Z}^T$ required by Equation (5.1) was output directly, rather than $\mathbf{Z}$ or $\mathbf{Z}^{-T}$. The search procedure was modified for compactness and efficiency.

- The undifferenced pseudoranges and carrier-phases variances are scaled by an inverse-sin function of the satellite elevation angles, but covariances are not modelled. The mathematical correlations introduced by the differencing are accounted for by transforming the undifferenced covariance matrix using double-differencing coefficient matrix; see Section 3.2.6.

- Measurements and broadcast ephemerides were input using Receiver INdependent EXchange Format (RINEX) format files. Precise ephemeris data was input using IGS/National Geodetic Survey (NGS) SP3 format files.

5.3.4 Normal matrix structure

The complete normal matrix for a combined adjustment can resemble

$$\mathbf{N} = \left(\begin{array}{c|cccc|cc} \mathbf{N}_{\mathbf{r}^m} & * & * & * & * & * & * \\ \hline * & \mathbf{N}_{\mathbf{R}_m^b} & * & * & * & \mathbf{0} & \mathbf{0} \\ * & * & \mathbf{N}_{IOP} & * & * & \mathbf{0} & \mathbf{0} \\ * & * & * & \mathbf{N}_{AP} & * & \mathbf{0} & \mathbf{0} \\ * & * & * & * & \mathbf{N}_{\mathbf{r}_{GNSS}^c/\mathbf{R}_b^c} & \mathbf{0} & \mathbf{0} \\ \hline * & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N}_{t_{GNSS/rx}} & \mathbf{0} \\ * & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N}_n \end{array}\right) \tag{5.2}$$

with:

- $\mathbf{N}_{\mathbf{r}^m}$: Co-ordinate parameters
- $\mathbf{N}_{\mathbf{R}_m^b}$: Exposure attitude parameters
- $\mathbf{N}_{IOP}$: Interior orientation parameters
- $\mathbf{N}_{AP}$: Camera additional parameters (lens, in-plane distortion)
- $\mathbf{N}_{\mathbf{r}_{GNSS}^c/\mathbf{R}_b^c}$: Lever-arm and boresight parameters
- $\mathbf{N}_{t_{GNSS/rx}}$: Receiver clock offsets for stations with undifferenced GNSS observations
- $\mathbf{N}_n$: Double-difference ambiguities

Non-zero off-diagonal blocks are indicated with '$*$'. The matrix is divided into three sections: one each for the photogrammetric and GNSS-specific parameters, and one for the common co-ordinate parameters. The photogrammetric- and GNSS-specific parameters do not appear together in any measurement equations; thus the zero off-diagonal matrices. Of course, fill-in will occur during inversion and so the adjusted parameters will be correlated.

5.3.5 Software Optimisation

There are a number of issues that must be considered before beginning software optimisation. The first of these is that while good-coding techniques should be practiced throughout an application's development, specific focus on optimisation should only come at the end. This is for two reasons: first, a poorly performing application is better than no application at all, and second, it can be self-defeating to try and find bottlenecks while an application is in flux, especially if the optimised code may not even find its way into the final application. The next consideration, again taken before making any code changes, it to determine if optimisation is even necessary. All software is designed for users, and if an optimisation will have an imperceptible or unimportant change on a user's experience with an application, then there is no need to invest the time and cost in implementing it. Finally, software optimisations must be balanced with the requirement to keep code readable and maintainable. If an optimisation results in only a small performance improvement while requiring byzantine code changes that hinder future development, then it is likely not advisable to implement it.

If software performance is an issue, then optimisation should only proceed after an application has been profiled and performance bottlenecks identified. Profiling measures the time taken in a program's functions and the number of times each function is called. Such information enables optimisation efforts to be focused on performance-critical sections of code. Particular focus should be given to time consuming functions that are frequently called. It is important to note that the times measured by a (good) profiler are Central Processing Unit (CPU)-times, not run-times. The former measurement, the actual time spent executing instructions on the CPU, is deterministic; the latter measurement, the real-world time spent in a function, is affected by the sharing of CPU by other processors and consequently less reliable.

Table 5.1 shows the profiler results for the adjustment of an aerial photogrammetric block with 1301 unknown parameters. The top 4 most time consuming operations are listed.

Table 5.1: Top 4 adjustment functions by total CPU-time

Operation	Percentage of total CPU-time	Number of function calls	Average time per call (μs)
Inversion of decomposed normal matrix	62.9	1	9,633,208.18
Solution of normal system of equations	34.2	5	1,047,278.92
Matrix array allocations and deallocations	1.1	660,326	0.26
Matrix multiplications	0.1	153,273	0.13

Two operations, namely the inversion of the normal matrix and the solution of the normal system of equations at each iteration, together consume over 97% of the program's run-time. Clearly, these operations are candidates for optimisation. The next two operations, while both having a huge number of function calls, have a comparably negligible impact on the program's overall run-time. Consequently, at this point, it makes little sense to attempt optimising them.

With performance bottlenecks identified, the first place optimisations should be looked for is in a program's algorithms. For example, in a least-squares adjustment an obvious algorithmic optimisation is to calculate the parameter corrections at each iteration by normal matrix decomposition and back-substitution rather than by inversion and multiplication. With the latter technique requiring an extra $O(n^3)$ operation over the former, even large improvements in the speed of the matrix inversion would not be enough to offset the much larger computational requirements. Only after algorithmic improvements are made should attention be turned to low-level code changes. Both algorithmic and code changes should always be tested to ensure that they do, indeed, improve the performance of the software.

A summary of commonly-accepted practices for optimising software is given in Table 5.2.

Table 5.2: Commonly-accepted practices for optimising software

Optimise at the end of development cycle.
Optimise only if the improvements will benefit users.
Balance optimisations with keeping readable and easily maintainable code.
Determine performance bottlenecks and optimisation improvements through testing.
Optimise algorithms first, code second.

Processor-Tuned Linear Algebra Libraries

The lengthy period spent solving the normal system of equations and inverting the normal matrix in the combined adjustment stems from its generic structure. In other, more specialised, large adjustment implementations, knowledge of the normal matrix's sparse structure is used to greatly speed-up both tasks that involve the normal matrix. In the combined adjustment, however, having the root adjustment object, which is responsible for these tasks, maintain knowledge of the sparsity of the normal matrix would add significant complexity. Therefore, in the implementation of the combined adjustment the full normal matrix was both stored and used in the adjustment calculations. This provoked a search for alternative linear algebra libraries that would offer improved performance for large matrices; a search that culminated in the use of processor-tuned linear algebra libraries.

Processor-tuned linear algebra libraries contain routines, optimised for specific CPUs, that perform common matrix and vector operations, and solve common linear algebraic problems. The libraries have APIs that correspond to two well-known and widely used FORTRAN linear algebra libraries: Basic Linear Algebra Subprograms (BLAS) and the Linear Algebra Package (LAPACK). The former library has routines that perform basic vector and matrix operations such as inner products and matrix-matrix multiplications. The latter library contains routines, built upon those in BLAS, that solve most commonly occurring problems in linear algebra. This includes routines that solve systems of equations that have a symmetric positive-definite coefficient matrix like those occurring in least-squares.

There are two sources for processor-tuned BLAS and LAPACK libraries. First, processor manufacturers generally distribute their own such libraries; for example, Intel has its Math Kernel Library and Advanced Micro Devices (AMD) its Core Math Library (ACML). Of course, these libraries may not be free, and they also may not provide any benefit on processors other than the vendor's own. Fortunately, however, there exists the Automatically Tuned Linear Algebra Software (ATLAS) project which provides freely-available BLAS and LAPACK libraries that can be optimised for virtually any platform or compiler. Distributed as source code, the ATLAS libraries are tuned for the processor on which they are compiled using a process termed the Automated Empirical Optimisation of Software (AEOS). As its name suggests, AEOS works by semi-intelligently testing different software optimisations. Results from these tests are used to automatically generate source code from which optimised BLAS and LAPACK libraries are created.

The dramatic benefit offered by processor-tuned BLAS and LAPACK libraries for large matrix operations is shown in Table 5.3. From this table, it can be seen that the processor-tuned implementation multiplies two 1000×1000 matrices over 95% faster than a conventional C++ implementation. For cholesky decomposition the improvement is not quite as dramatic, but, at nearly 80%, it is still remarkable. For comparison's sake, the results from two other implementations are also shown in Table 5.3: a generic Fortran BLAS/LAPACK, and a conventional 'C' implementation using a double-pointer array. The timings from the latter library confirm that there were no implementation error in the C++ functions, and the timings from the former indicate that even if a processor-tuned implementation is unavailable, then it is still a good idea to delegate large-matrix operations to Fortran.

Of course, using processor-tuned linear algebra libraries is not without some problems. The first of which is that, for small matrices, conventional routines can out-perform those from the tuned libraries. This phenomenon is shown in Figure 5.8, where it can be seen that the processor-tuned matrix multiplication is faster than a conventional C++ implementation only for matrices larger than 10×10. For matrices smaller than this, the tuned

Table 5.3: Comparison of run-times for 1000 × 1000 matrix operations using different linear algebra implementations

	Operation	
Implementation Type	Multiplication	Cholesky decomposition
C++	10.04s	0.47s
Processor-tuned BLAS/LAPACK	0.47s	0.10s
Generic Fortran BLAS/LAPACK	1.24s	0.36s
Conventional 'C'	11.60s	0.47s

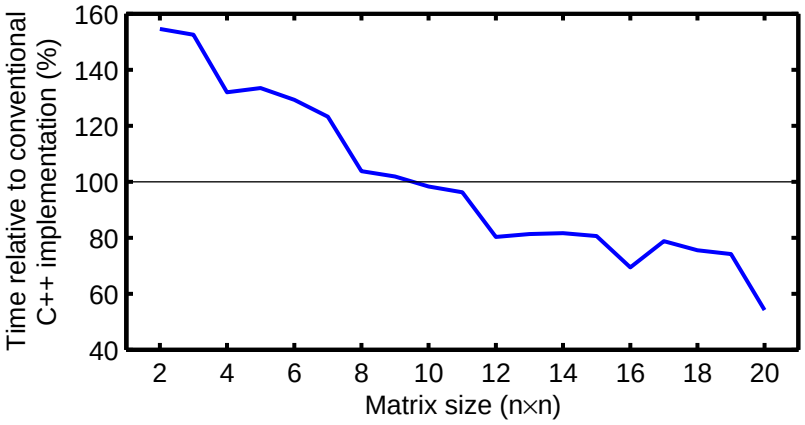


Figure 5.8: Time required for the multiplication of two $n \times n$ matrices by a processor-tuned implementation relative to a conventional C++ implementation

implementation can require over 50% more time. Because the overwhelming majority of matrix operations in the combined adjustment involve smaller matrices, only specific large-matrix operations were replaced with processor-tuned operations, rather than all the matrix operations.

Another problem with the processor-tuned libraries is that of software distribution. Because the ATLAS libraries are processor-specific, it is likely that they will not provide the same performance boost on a computer that has a different processor than that on which they were compiled. Indeed, it is even possible that they will not work at all. This problem is solved in the combined adjustment by putting the tuned-libraries into shared libraries (Dynamic-Linked Librarys (DLLs) on Windows) and distributing sets of shared libraries for

Table 5.4: Top 4 adjustment functions by total CPU-time when using processor-tuned linear algebra libraries

Operation	Percentage of total CPU-time	Number of function calls	Average time per call (μs)
Solution of normal system of equations	57.5	5	274,539.29
Inversion of decomposed normal matrix	25.3	1	604,110.92
Matrix array allocations and deallocations	6.7	660,326	0.24
Matrix multiplications	0.5	153,273	0.14

different processors with the adjustment software. As a fall-back, generic FORTRAN BLAS and LAPACK libraries are also included, as they should provided improved performance on all processors.

After adding the processor-tuned libraries to the adjustment, the profiler output the information in Table 5.4. When compared with the results from Table 5.1, it can be seen that times required for solving the normal system of equations and inverting the normal matrix have been reduced by 74% and 94%, respectively. These tremendous improvements obviously greatly reduce the total run-time of the program. They also have the side-effect of increasing the proportion of total run-time spent in allocating and deallocating matrix memory to an amount large enough to consider optimisation.

Reducing Matrix-Memory Allocations

A typical run of the combined adjustment will make transitory use of many small matrices and vectors. These are predominantly of three types:

- The $\mathbf{A}_i$ Jacobian matrices and $\mathbf{w}_i$ misclosure vectors for individual observations
- Matrices and vectors used in the intermediate calculation of $\mathbf{A}_i$ and $\mathbf{w}_i$ – for example, rotation matrices

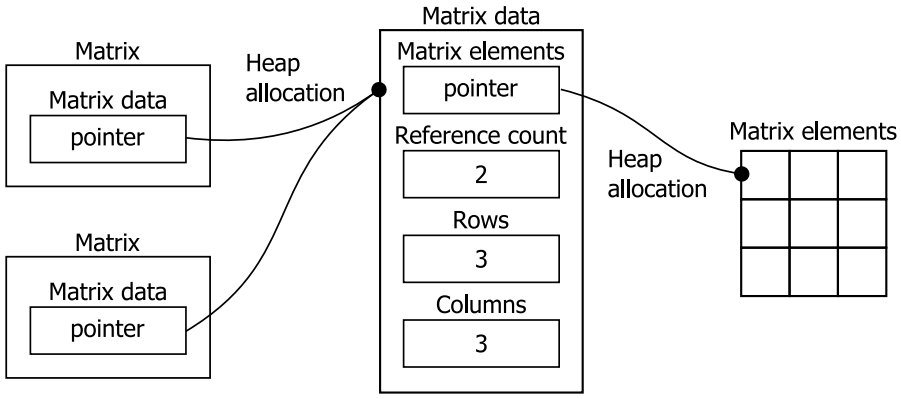


Figure 5.9: Matrix class memory diagram

- Matrices used in the calculation of the summation of normals $\mathbf{N}_i$ terms.

The memory for these matrices is dynamically allocated on the heap. Unfortunately, these heap allocations can be an expensive process, as before they can occur the operating system must first search its heap index for an area big enough for the requested allocation, and then update the index to reflect that a block of memory is now in use. Similarly, when the program is done with the memory the operating system must again update its index to indicate that the block of memory is again free for use. In the combined adjustment, the problem of matrix heap allocations is compounded because its matrix class uses *reference counting*. Reference counting is a memory-management technique where an object's data is stored separately from the object, together with a count of the number of objects using the same data. It has a number of purposes, but in the combined adjustment it is used to reduce the cost of copy operations and to simplify the code. Unfortunately, because of reference counting, the matrix class requires two heap allocations for every instance – one for the reference-shared data object and another for the actual matrix elements. This process, together with the matrix storage scheme, is illustrated in Figure 5.9.

To reduce the number of small matrix-memory heap allocations there are a number of approaches that could be taken. First, already-allocated memory could be re-used. In other words, the required memory could be allocated at the start of the adjustment and

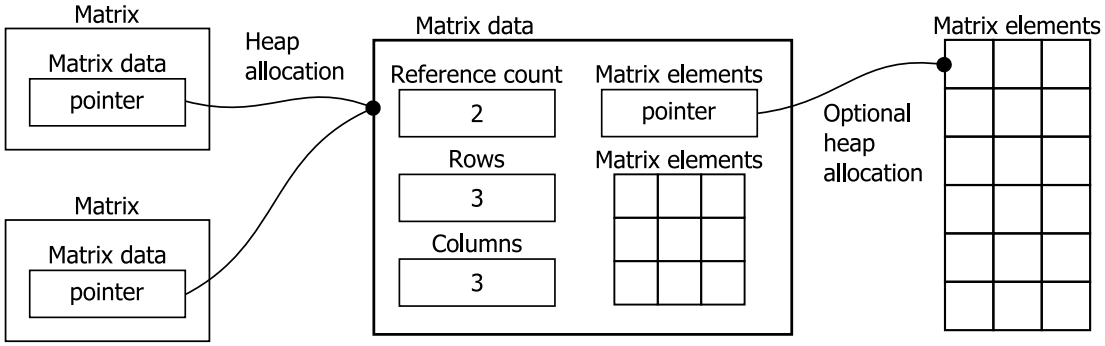


Figure 5.10: Matrix class memory diagram using pre-allocated element memory

not deallocated until the end. Unfortunately, this approach would require the adjustment to be made both more complex and less generic. A second approach is to reduce the allocations required for the matrix elements. This can be done by pre-allocating a fixed number of elements along with the matrix data. Then, as long as the number of matrix elements does not exceed this pre-allocated number, the second allocation is no longer necessary. This approach, depicted for the matrix class in Figure 5.10, is commonly used in string implementations. It is a desirable technique because it requires changes only to the matrix class itself; users of the matrix class need make no changes. For this reason it was implemented in the combined adjustment.

An obvious issue with the pre-allocation of the matrix elements is determining the number of elements to pre-allocate. Too many, and the amount of time and memory wasted by allocating the over-sized element storage may become significant. Too few, and there won't be an appreciable performance benefit. Fortunately, a reasonable pre-allocation size can be determined by examining the sizes of matrices that occur in common adjustments. For instance, in the 1266 image-measurement aerial block used in the profiling results of Tables 5.1 and 5.4, 317,095 matrix-memory allocations take place during the course of the adjustment. Over 75% of these matrices have 9 elements or less, with half of these being 9-element 3×3 rotation matrices. Therefore, pre-allocating 9 elements is a reasonable pre-allocation scheme.

Table 5.5: Top 4 adjustment functions by total run-time when pre-allocating matrix element memory

Operation	Percentage of total CPU-time	Number of function calls	Average time per call (μs)
Solution of normal system of equations	56.7	5	270,847.89
Inversion of decomposed normal matrix	25.3	1	603,206.99
Matrix array allocations and deallocations	2.9	660,326	0.16
Matrix multiplications	0.9	153,273	0.14

After modifying the combined adjustment's matrix class to pre-allocate 9 elements, the profiling results in Table 5.5 were observed. The time required for matrix memory allocation and deallocation has been reduced by 33%. An additional side benefit of the pre-allocated array was that it made for easier debugging, as all matrix elements were, by default, visible in the debugger.

Small Matrix Optimisations

Most of the matrices used in the combined adjustment are small and have a fixed size that is known at compile-time. For these matrices, closed form expressions of common operations can be written which avoid the overhead of loops and non-constant array access that would otherwise be present in more generic algorithms. For example, the product of two 3×3 matrices can be explicitly calculated using

```
template< typename T >
void
matrix_product_3x3( const T A[], const T B[], T C[] )
{
// Macros to simplify array access
#define A(i,j) A[(i-1)*3+(j-1)]
#define B(i,j) B[(i-1)*3+(j-1)]
#define C(i,j) C[(i-1)*3+(j-1)]
    C(1,1) = A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1);
    C(1,2) = A(1,1)*B(1,2)+A(1,2)*B(2,2)+A(1,3)*B(3,2);
    C(1,3) = A(1,1)*B(1,3)+A(1,2)*B(2,3)+A(1,3)*B(3,3);
```

```
    C(2,1) = A(2,1)*B(1,1)+A(2,2)*B(2,1)+A(2,3)*B(3,1);
    C(2,2) = A(2,1)*B(1,2)+A(2,2)*B(2,2)+A(2,3)*B(3,2);
    C(2,3) = A(2,1)*B(1,3)+A(2,2)*B(2,3)+A(2,3)*B(3,3);

    C(3,1) = A(3,1)*B(1,1)+A(3,2)*B(2,1)+A(3,3)*B(3,1);
    C(3,2) = A(3,1)*B(1,2)+A(3,2)*B(2,2)+A(3,3)*B(3,2);
    C(3,3) = A(3,1)*B(1,3)+A(3,2)*B(2,3)+A(3,3)*B(3,3);
#undef A
#undef B
#undef C
}.
```

Similarly, the in-place Cholesky decomposition of a 3×3 matrix can be calculated by

```
template< typename T >
void
matrix_cholesky_decomposition_3x3( T A[] )
{
#define A(i,j) A[(i-1)*3+j]
#define sq(x) ((x)*(x))
    A(1,1) = sqrt( A(1,1) );

    A(2,1) = A(2,1)/A(1,1);
    A(2,2) = sqrt( A(2,2) - sq(A(2,1)) );

    A(3,1) = A(3,1)/A(1,1);
    A(3,2) = (A(3,2) - A(3,1)*A(2,1))/A(2,2);
    A(3,3) = sqrt( A(3,3) - sq(A(3,1)) - sq(A(3,2)) );
#undef A
#undef sq
}.
```

The performance advantage of using fixed-size matrix functions can be significant. Table 5.6 shows the run-times for 1×10^7 multiplications and Cholesky-inversions of a 3×3 matrix using three different types of implementation: ordinary (but templated) C++, functions from the processor-tuned libraries from above, and fixed-size algorithms like those just provided. The fixed-size implementation clearly outperforms either of the other two implementations. It is, for both operations, over 50% faster than the ordinary C++ implementation and over 85% faster than the processor-tuned functions. To be fair, the comparison is not completely even, as the processor-tuned libraries do additional checks on the function arguments that the other implementations do not. However, it is impossible to

Table 5.6: Comparison of run-times for 1×10^7 3×3 matrix operations

	Operation	
Implementation Type	Multiplication	Cholesky inversion
Ordinary 'C++'	1.7s	3.9s
Processor-tuned	4.9s	21.8s
Fixed-size	0.7s	1.7s

remove these checks, and so they must still be considered as part of the overhead of using functions from the processor-tuned libraries.

Of course, creating fixed-size expressions for all sizes and combinations of small matrices would be tedious. Fortunately, C++ templates enable these fixed-size algorithms to be generalised while still maintaining the performance enhancements from them. In order to do this, it must first be recognised that most matrix operations are implemented as a series of nested loops. To implement these operation using C++ templates, it is necessary to start with the inner most loop and work outwards. For example, a matrix-matrix product can be expressed as a series of matrix-vector products, which, in turn, can be expressed as a series of vector-vector inner (dot) products. The last of these products is itself a series of scalar products that can be expressed using a templated class such as

```
template< typename T, size_t n >
struct
inner_product_t
{
    static T calculate(
        const T x[], size_t x_stride,
        const T y[], size_t y_stride )
    {
        return x[0] * y[0] + inner_product_t<T,n-1>::calculate(
            x+x_stride, x_stride, y+y_stride, y_stride );
    }
};.
```

This class contains a single `calculate` static class member function that simply calculates the scalar product of the current head elements in the vectors `x` and `y`, and adds it to the sum of the inner product of the next `n-1` elements. This is similar to run-time recursion

only that it takes place at compile-time. Like recursion, it also requires an end condition. This condition is supplied by a *partial-template specialisation* of the `inner_product_t` class for vectors with a single element,

```
template< typename T >
struct
inner_product_t< T, 1u /* n = 1 */ >
{
    static T calculate(
        const T x[], size_t x_stride,
        const T y[], size_t y_stride )
        { return x[0] * y[0]; }
};.
```

The key to this technique's performance is that the compiler is free to optimise all of the individual `inner_product_t::calculate` function calls *at compile time*. Potentially, if all of the nested `calculate` functions are expanded inline by the compiler, then this would lead to the generated machine code being exactly the same as if an explicit fixed-size inner product had been written instead.

To complete the matrix-matrix multiplication, the same technique is first applied to the matrix-vector multiplications,

```
template< typename T, size_t m, size_t n >
struct
matrix_vector_product_t
{
    static void calculate(
        const T A[], size_t A_stride,
        const T x[], size_t x_stride,
        T y[], size_t y_stride )
    {
        y[0] = inner_product<T,n>::calculate( A, 1, x, x_stride );

        matrix_vector_product_t< T, m-1, n >::calculate(
            A + A_stride, A_stride, x, x_stride, y + y_stride, y_stride );
    }
};,
```

with specialisation

```
template< typename T, size_t n >
struct
matrix_vector_product_t< T, 1u, n >
```

```
{
    static void calculate(
        const T A[], size_t A_stride,
        const T x[], size_t x_stride,
        T y[], size_t y_stride )
        { y[0] = inner_product<T,n>::calculate( A, 1, x, x_stride ); }
};.
```

Finally, the product of two matrices can be calculated using

```
template< typename T, size_t m, size_t n, size_t p >
struct
matrix_product_t
{
    static void calculate(
        const T A[], size_t A_stride,
        const T B[], size_t B_stride,
        T C[], size_t C_stride )
  {
    // Calculate the product of A with the current column in B
    matrix_vector_product<T,m,n>::calculate( A, A_stride,
          B, B_stride, C, C_stride );

    // Calculate remaining columns
    matrix_product_t< T, m, n, p-1 >::calculate(
        A, A_stride, B + 1, B_stride, C + 1, C_stride );
  }
};.
```

Again, with specialisation

```
template< typename T, size_t m, size_t n >
struct
matrix_product_t< MatrixA, MatrixB, Result, m, n, 1 >
{
    static void calculate(
        const T A[], size_t A_stride,
        const T B[], size_t B_stride,
        T C[], size_t C_stride )
    {
        matrix_vector_product<T,m,n>::calculate( A, A_stride,
            B, B_stride, C, C_stride );
    }
};.
```

The `stride` arguments in the above functions are the distances between adjacent columnar elements in the matrices. A `size_t` is a positive integer (i.e., a whole number).

Using these class templates, the product of any two matrices can be calculated at compile time. For example, the product of a 4×2 matrix with a 2×3 matrix can be calculated using

```
matrix_product_t<double,4,2,3>::calculate( A, 4, B, 3, C, 3 );
```

The syntax of this function call can be cleaned up slightly by defining the templated function

```
template< size_t m, size_t n, size_t p, typename T >
void
matrix_product( const T A[], const T B[], T C[] )
{
    return matrix_product_t<T, m, n, p>::calculate(
        A, n, B, p, C, p );
}
```

Leading to function calls such as

```
matrix_product<4,2,3>( A, B, C );
```

The type of `T` would be deduced at compile-time from `A`, `B`, and `C`. The three matrices (arrays) should be of the same type (e.g., **double**, **int**, etc.), although the classes and functions defined above are easily (although perhaps dangerously, if overflows are not considered) modified to handle matrices of different types.

This partial-template specialisation technique can be extended to any operation that uses loops. Of course, more complex operations – like, for example, a Cholesky decomposition – would have increasingly complex code. Also, the success of the technique depends on how well the compiler optimises the templated code. For instance, in tests with Microsoft's C++ compiler (ver. 7.10) the templated matrix multiplication performed some 8% *better* than an explicitly written function. In contrast, with the GNU C++ compiler (MinGW ver. 3.4.2), the templated technique performed over twice as poorly. Part of the problem with the GNU compiler appears to be that it cannot expand the `matrix_product_t::calculate` or `matrix_vector_product_t::calculate` functions inline, as evidenced by a compile-error when forced inline expansion was attempted using the `always_inline` function attribute.

Unfortunately, the adjustment software was not able to take advantage of the speed improvements offered by fixed-size routines. Short of making fundamental code modifica-

Table 5.7: Time required for normal matrix operations using single-precision values

Operation	Average time per call
Solution of normal system of equations	405,708.03
Inversion of decomposed normal matrix	196,229.73

tions, the only way to add such routines to the adjustment's matrix library was to insert size-based tests into existing matrix routines. Depending on the results from these simple **if** statement tests, the matrix operation would then either proceed normally or be delegated to an appropriate fixed-size routine. Surprisingly, however, the overhead of the tests negated the improvement offered by the fixed-size routines. It is possible to modify the code so that these tests are not required, with an obvious and elegant approach being the implementation of a templated fixed-size matrix class and associated templated matrix operation functions. Unfortunately, such an exercise would be prohibitively time-consuming and was therefore not implemented for the combined adjustment. A sample implementation of such a framework is, nevertheless, shown in Appendix B.

Single Precision Normal Matrix

A final note regarding the performance of the combined adjustment involves the use of single-precision matrices in the normal system of equations rather than the more typical double-precision matrices. This is not an optimisation technique. Instead, it is an alternative processing strategy that also has the side-benefit of reducing memory requirements. In theory the use of single-precision values should result in less accurate parameters; however, in practice, for most networks the difference is minor. For instance, in the aerial block used throughout this section, the maximum differences between object space co-ordinates and exposure attitudes were less than 1mm and 1", respectively. In contrast, the difference in adjustment time was significant. As can be seen by comparing values in Table 5.4 and Table 5.7, operations involving the normal matrix took about 30% less time when single-precision values were used instead of double-precision values.

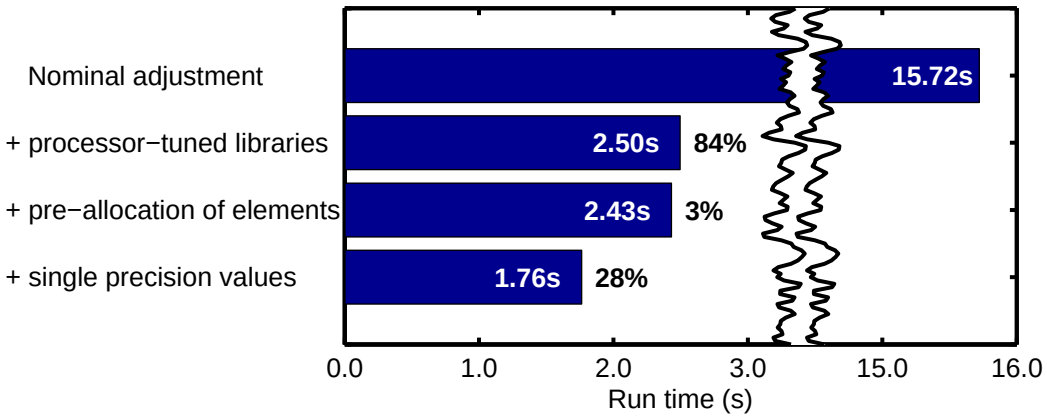


Figure 5.11: Adjustment performance improvements due to software optimisations

Net effect of performance enhancing technique

Figure 5.11 shows the net effect of all the aforementioned optimisations and techniques to improve adjustment performance. The total decrease in adjustment time using all techniques is a very-appreciable 89%.

5.4 Feedback Filter - Implementation Aspects

The feedback filter involved the creation of an all-new GNSS Kalman filter. An existing GNSS Kalman filter was available but it did not use modern ambiguity resolution methods. Furthermore, it was over 10 years old, the source code was poorly and confusingly implemented, and it used an arcane and proprietary GNSS data file format. For all these reasons, it was felt that creating a new GNSS Kalman filter would not be significantly more work than working with the existing filter. Also, the creation of a GNSS Kalman filter was a valuable learning exercise.

5.4.1 Implementation Notes

The GNSS Kalman filter used the same GNSS libraries created for the combined adjustment. Consequently, the GNSS implementation notes of Section 5.3.3 also apply to the Kalman

filter. Kalman-filter specific implementation aspects are:

- A total-state Kalman filter is used; see Section 2.2.3.
- The position states can be parametrised as either geocentric or geodetic co-ordinates; see Section 3.8.
- The navigation states can use either a position, position-velocity, or position-velocity-acceleration dynamic model; see Section 3.5.1.

5.4.2 Application of CUPTs

These positions from the bundle adjustment are incorporated into the Kalman filter using simple state-observation equations. Obviously, the exact form of these equations depends on the co-ordinate frames used by the two processors and the type of Kalman-filter. If the filter positions are parametrised using geocentric co-ordinates, then the CUPT equation is a straight parameter equivalency. The position covariance matrix from the adjustment can also be used directly. On the other hand, if the filter is using geodetic co-ordinates, then the geocentric co-ordinates from the adjustment must first be transformed to geodetic. The geocentric covariance matrix must be rotated into the local-level frame, and the x- and y-components scaled to degrees,

$$\mathbf{C}_{\mathbf{r}^{\phi\lambda h}} = \mathbf{S}_l^{\phi\lambda h}\mathbf{R}_e^l\mathbf{C}_{\mathbf{r}^e}\left(\mathbf{S}_l^{\phi\lambda h}\mathbf{R}_e^l\right)^T. \tag{5.3}$$

where

$$\mathbf{S}_l^{\phi\lambda h} = \begin{pmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{N+h} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.4}$$

A minor complication in this approach is that the exposure times and, consequently, the CUPT filter updates likely do not coincide with GNSS measurements epochs. However, this

is easily handled by having the Kalman Filter predict up to the time of the CUPT before performing the measurement update, as shown in Figure 5.12.

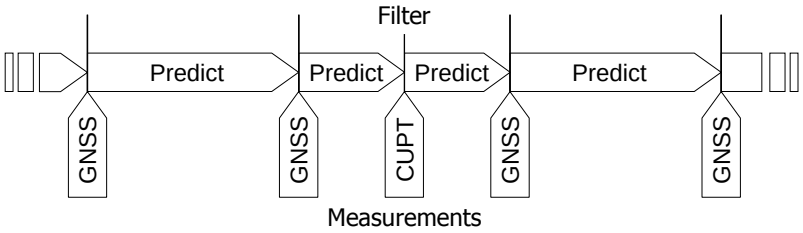


Figure 5.12: Operation of the GNSS Kalman filter

Chapter 6

Testing, Results, and Analysis

The new integration strategies introduced in the previous chapter were tested using two data sets. The first data set was a conventional aerial photogrammetric block. The second was a simulated terrestrial mobile mapping campaign.

6.1 Aerial Photogrammetric Block

The data set used for testing consisted of a block of 84 aerial images captured at a photo scale of approximately 1:5,000. Image acquisition was done using a conventional 9" × 9" metric analogue frame camera with a 6" focal length. Co-ordinates were available for 17 ground points; these points were treated as check points in the tests that follow. Dual-frequency GPS data at 2Hz was collected on the aeroplane and at a master station located approximately 24km from the centre of the block. The arrangement of the data set's exposures and ground points can be seen in Figure 6.1.

6.1.1 Conventional processing

The first tests performed with the data used conventional processing strategies, and were done to establish the noise level inherent in the network. Results from these tests will act as the basis of comparison for the tests of the new integration techniques that follow. The

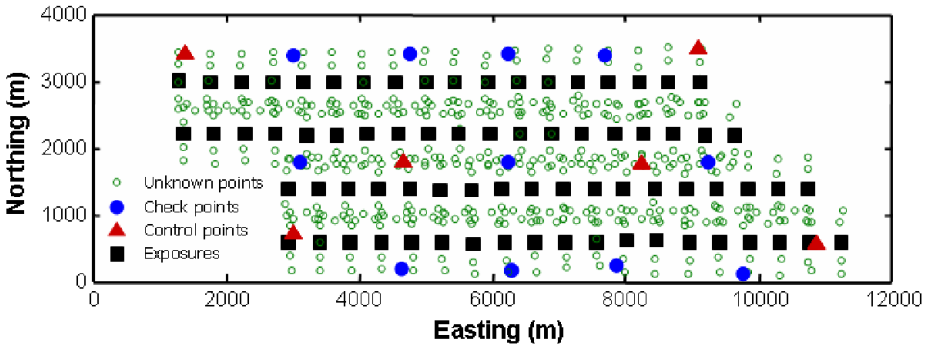


Figure 6.1: Test Field

noise level in the network, due in turn to the image measurement noise, was observed using two configurations: a network controlled using ground points, and a network controlled using the best available GNSS-derived exposure station positions. For the ground controlled network, 6 well-distributed points were selected to act as control and the remaining 11 points were used as check points. Figure 6.1 shows the distribution of these points. For the GNSS-controlled network, exposure station position observations were generated by Waypoint's Grafnav 7.0 commercial kinematic GNSS processor using dual-frequency data. Ambiguities were reported as fixed for all stations. All 17 available check points were used to generate the statistics.

The results for these two network configurations are listed in Tables 6.1 and 6.2. The standard deviations of the check points from both configurations indicates that there is about 10 cm of horizontal and $20 - 25$ cm of vertical "noise" in the network. This noise is, in turn, due to the noise in the image measurements.

Looking at the mean values in Table 6.2, it is apparent that there is a significant trans-

Table 6.1: Check-point statistics for ground-controlled network

	Horizontal	Vertical
Mean (m)	0.19	0.22
Std. Dev. (m)	0.11	0.20
RMS (m)	0.22	0.29

Table 6.2: Check-point statistics for network controlled using best-available GNSS exposure station position observations

	Horizontal	Vertical
Mean (m)	1.30	16.38
Std. Dev. (m)	0.10	0.27
RMS (m)	1.31	16.38

lation between the GPS and check point datums. The bulk of this difference is due to the vertical datum of the check points being based upon orthometric heights; however, there is also a substantial horizontal difference. The translations between the two datums were estimated in an adjustment that included both the exposure station position observations and the fixed ground control points. The estimated translations were then used in all subsequent tests, unless otherwise noted. Since the translations themselves contain estimation error this is hardly an ideal solution, but without access to the original control point survey data there is little else that can be done.

When the adjustment controlled using GNSS-derived exposure station positions is repeated with the datum translations applied, the results are as shown in Table 6.3.

Table 6.3: Check-point statistics for network controlled using best-available GNSS exposure station position observations, datum translations applied

	Horizontal	Vertical
Mean (m)	0.23	0.01
Std. Dev. (m)	0.18	0.27
RMS (m)	0.29	0.26

Another possible contributor of the check-point biases seen when the network is controlled using only GNSS-derived exposure station positions is an inaccurate camera calibration. As described above, the camera used for the data collection was an metric analogue frame camera. The camera calibration supplied with the imagery was used in the previous adjustments, but it is possible that the true interior orientation and lens distortion parameters differed from these calibrated values. To examine this hypothesis an adjustment

was conducted where both the interior orientation parameters and the translations between check point and GPS datums were estimated. The translations differed by 4.5 cm from the translations estimated in the non-self-calibrating adjustment, and both the principal point and focal length moved approximately 30 μm from the supplied calibrated values. When these translation and interior orientation parameters were used in an adjustment controlled using only GNSS-derived exposure station positions an improvement in check-point accuracy was observed: compare Tables 6.3 and 6.4.

Table 6.4: Check-point statistics for network controlled using best-available GNSS exposure station position observations; datum translations and camera interior orientation estimated

	Horizontal	Vertical
Mean (m)	0.17	0.02
Std. Dev. (m)	0.10	0.18
RMS (m)	0.20	0.17

Ultimately, however, the self-calibrated interior orientation parameters were not used in subsequent tests. This was because moderate-to-high parameter correlations between the interior orientation and datum translations drew into question the physical meaningfulness of the estimated parameters. In particular, the focal length had near perfect correlation (> 0.97) with all translations. This is not surprising, as both a change in focal length or a shift in ground co-ordinates away from the focal plane affect the image measurements similarly.

6.1.2 Combined Adjustment

The combined adjustment has primarily been tested by comparing it to the existing technique of position observations. In all tests, the position observations were generated using the Waypoint's Grafnav 7.0 commercial GNSS processor in the same configuration as the adjustment. The comparison of results will primarily be done using the standard deviations of the check point errors. This in acknowledgment of the fact that mean errors are unavoidable due to the translations between the GPS and check point datums.

An important consideration in adjustments incorporating multiple observation types is the relative weighting of the different observation groups. In the tests that follow, the image measurement standard deviations were determined from the aposteriori variance factors from the conventionally controlled adjustments. The value from the variance factors – 4.5 μm – is reasonable for the analytical plotters used for the data collection. For the position observations, the covariances reported by the GNSS processor were used. Lastly, for the GNSS observations, the variances used were conservative estimates based on pseudorange and carrier phase measurement noises reported by GNSS receiver manufacturers in their product literature: for the pseudoranges and carrier phases the variances were 0.25 m and 0.04 m, respectively. Some attempts were made to use the variance factors output by the adjustment to guide the input GNSS observation variances, but the variance factors either lead to impossible measurement weights or worse results; consequently, the use of the variance factors was abandoned.

Undifferenced pseudoranges

The first tests of the combined adjustment were done using undifferenced code ranges. The combined adjustment is compared against the traditional method of position observations in Tables 6.5 and 6.6. The results in these tables appear to indicate that the combined adjustment is significantly more accurate than the conventional approach. These results are, however, misleading. The better accuracy of the combined adjustment is due to the better performance (in this specific case, at least) of the GNSS-engine of the combined adjustment relative to the Grafnav commercial processor used to generate the position observations. When the combined adjustment is used to generate position observations that are subsequently used in the adjustment (by removing the image measurement observations), the results are as shown in Table 6.7. In this case, accuracies are essentially equal to those obtained from a full combined adjustment. This is discouraging, as it implies that the photogrammetric data is not improving the GNSS positioning. This is indeed the case, as

can be seen by comparing the exposure position statistics in Tables 6.8 and 6.9.

Table 6.5: Check-point statistics for combined adjustment done using undifferenced pseudoranges

	Horizontal	Vertical
Mean (m)	2.11	-2.98
Std. Dev. (m)	0.20	0.65
RMS (m)	2.12	3.05

Table 6.6: Check-point statistics for network controlled using GNSS exposure station position observations derived from undifferenced pseudoranges

	Horizontal	Vertical
Mean (m)	1.78	-3.60
Std. Dev. (m)	0.47	1.18
RMS (m)	1.84	3.78

Table 6.7: Check-point statistics for network controlled using GNSS exposure station position observations derived from undifferenced pseudoranges, generated by combined adjustment

Statistic	Horizontal	Vertical
Mean (m)	2.11	-2.98
Std. Dev. (m)	0.20	0.64
RMS (m)	2.11	3.05

The check-point standard deviations from the position observations approach were closely coupled to the weight given to the position observations. For instance, to produce the results in Table 6.6, the position observation variances used were those naturally reported by the commercial processor. However, by decreasing the position observation variances, check-point standard deviations could be brought much closer to the combined adjustment's. The reason why the reduced variances lead to better check-point standard deviations is that over time spans like that required for the aerial survey, the troposphere, ionosphere, satellite orbit and clock errors all have significant bias components. Consequently, the relative accuracy of the position observations is significantly better then their absolute accuracy, and this internal consistency can be used to tighten up the entire network.

It is tempting to believe that adding some photogrammetric ground control might im-

Table 6.8: Exposure station position statistics for combined adjustment done using undifferenced pseudoranges

Statistic	Horizontal	Vertical
Mean (m)	2.33	-3.18
Std. Dev. (m)	0.33	0.47
RMS (m)	2.36	3.22

Table 6.9: Exposure station position statistics for position observations generated by combined adjustment without photogrammetric data

Statistic	Horizontal	Vertical
Mean (m)	2.33	-3.16
Std. Dev. (m)	0.32	0.48
RMS (m)	2.36	3.19

prove the GNSS undifferenced-observation derived exposure positions. However, this is not the case. Table 6.10 shows, for instance, the check point statistics when a single ground point in the centre of the block is fixed in the adjustment. The result is a dramatic decline in position accuracy. The explanation for this is that when ground control is used, it forces the errors in the pseudoranges to be absorbed into the exposure station positions, rather than manifesting themselves as a datum shift.

Table 6.10: Check-point statistics for combined adjustment done using undifferenced pseudoranges with single fixed control point

Statistic	Horizontal	Vertical
Mean (m)	0.98	-1.73
Std. Dev. (m)	0.78	2.86
RMS (m)	1.24	3.27

Single-frequency pseudorange and carrier-phase double-differences

The next set of tests used single-frequency, double-differenced pseudoranges and carrier-phases. With kinematic data and a baseline distance over 20 km, the commercial processor GNSS was, not surprisingly, unable to resolve ambiguities. Accordingly, the combined ad-

justment was first run with ambiguity fixing disabled. The results from the this, compared with the equivalent results using the GNSS processor's exposure station position observations are shown in Tables 6.11 and 6.12. The combined adjustment gives marginally poorer check point positions. This is, in turn, likely a result of poorer GNSS error and covariance modelling in the combined adjustment relative to the commercial processor.

Table 6.11: Check-point statistics for combined adjustment done using double-differenced single-frequency pseudoranges and carrier-phases, float ambiguities

	Horizontal	Vertical
Mean (m)	0.29	-0.10
Std. Dev. (m)	0.17	0.26
RMS (m)	0.33	0.28

Table 6.12: Check-point statistics for network controlled using GNSS exposure station position observations derived from double-differenced single-frequency pseudoranges and carrier-phases

	Horizontal	Vertical
Mean (m)	0.22	0.02
Std. Dev. (m)	0.16	0.24
RMS (m)	0.26	0.23

Unlike the stand-alone kinematic GNSS processor, the combined adjustment can fix and validate the ambiguities, with the resulting check-point statistics shown in Table 6.13. Unfortunately, judging from the mean errors in this table, and from the differences between the adjusted exposure positions and the dual-frequency, fixed ambiguity exposure positions from the GNSS processor in Table 6.3, the ambiguities were not correctly resolved. Again, this points toward deficiencies in the GNSS error and covariance modelling of the combined adjustment. The float solution is deformed, and this directly and negatively impacts ambiguity resolution.

It is interesting to note that the check-point standard deviations for all single-frequency tests are about the same as for the ground- or dual-frequency GNSS exposure positions-controlled networks of Section 6.1.1. This illustrates that a well-constructed block can tolerate less accurate GNSS positions whether explicitly provided or implicitly calculated.

Table 6.13: Check-point statistics for combined adjustment done using double-differenced single-frequency pseudoranges and carrier-phases, fixed ambiguities

	Horizontal	Vertical
Mean (m)	0.37	-0.17
Std. Dev. (m)	0.17	0.26
RMS (m)	0.40	0.31

It also implies that ambiguity resolution is not necessary. Indeed, given the potential deleterious effects of wrongly fixed ambiguities that the conventional shift-and-drift model is intended to compensate for, it is arguably advisable to *not* attempt ambiguity resolution.

Dual-frequency carrier-phases and code ranges double-differences

The final set of tests used dual-frequency double-differenced pseudoranges and carrier-phases. The check-point statistics for the network controlled using the exposure positions generated by the GNSS processor have already been shown in Table 6.3, while results using the combined adjustment are shown in Table 6.14. Disappointingly, the ambiguities appear to have been incorrectly resolved in the combined adjustment. This is not remedied by tightening the validation criteria; rather, ambiguities then fail to resolve. This is, again, almost certainly the result of a deformed float ambiguity solution.

Table 6.14: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases, fixed ambiguities

	Horizontal	Vertical
Mean (m)	0.22	0.09
Std. Dev. (m)	0.17	0.25
RMS (m)	0.27	0.26

Repeating the adjustment and not fixing ambiguities gives the results shown in Table 6.15. In this case, the mean errors are unexpectedly even larger than with the (wrongly) fixed ambiguity results. This is probably a result of the ionosphere not being modelled. The biased float positions make it not surprising that the ambiguities are fixed incorrectly.

Table 6.15: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases, float ambiguities

	Horizontal	Vertical
Mean (m)	0.38	0.22
Std. Dev. (m)	0.18	0.24
RMS (m)	0.42	0.32

The addition of the photogrammetric data does have some impact on the covariance matrix of the ambiguities: it reduces its condition number by a small amount. If the integer least squares search were done using the not-decorrelated ambiguities then there would be a corresponding small decrease in the search volume. However, the search space remains very badly scaled and decorrelation is still practically necessary. After decorrelation the condition numbers of the ambiguity covariance matrices and integer least squares search spaces are essentially equivalent.

Unusual network configurations

A benefit of the combined adjustment is that it enables more flexibility in how the data can be used. Two examples outlined earlier were having a non-fixed GNSS master station and using less than four satellites. Results from both configurations are shown below in Tables 6.16 and 6.17.

Table 6.16: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases, photogrammetric datum control

	Horizontal	Vertical
Mean (m)	0.23	-0.14
Std. Dev. (m)	0.16	0.24
RMS (m)	0.28	0.27

For the results in Table 6.16, the datum was controlled by a single photogrammetric ground control point located near the centre of the block. And, unlike in the previous tests, the estimated datum translations were *not* applied. Accuracies in this case were effectively

Table 6.17: Check-point statistics for double-differenced code range and carrier-phase position observations and 3 satellites

	Horizontal	Vertical
Mean (m)	0.27	-0.13
Std. Dev. (m)	0.18	0.32
RMS (m)	0.33	0.33

as good as the best possible from the network. Use of this configuration neatly avoids any inconsistencies between the GNSS and photogrammetric control datums. Of course, should the differences between the two datums be too large then this would cause errors in relative GNSS baselines implicitly used in the combined adjustment.

For Table 6.17, the number of observations was limited to the 3 observations from the satellites with the lowest Pseudo-Random Noise (PRN) code number. Check-point positions are only slightly worse than when the all satellites are used. Practically speaking, such a situation would never occur with aerial data collection. However, with terrestrial mobile mapping it would inevitably occur due to satellite masking. The combined adjustment would enable carrier-phases observations from satellites that continue to be tracked to be used, even when the total number of satellites drops below 4.

Analysis

The key hope for the combined adjustment was that it would enable improved GNSS positioning. In particular, an improved ability to do ambiguity resolution. Unfortunately, this hope has not been borne out in testing. There are two reasons for this: First, the test data set had $10 - 20\,\text{cm}$ of noise in the photogrammetrically-determined exposure station positions. This is about the same noise level in the float-ambiguity-derived exposure station positions. Thus, the photogrammetric data was adding little information to the exposure positions, and consequently the GNSS positioning was not substantially improved. Second, the GNSS error and covariance modelling was inadequate. This meant that the float solutions were both biased and had deformed covariances, greatly hurting the ambiguity

resolving performance, particularly over longer baselines.

Another observation that can be made from the results presented above is that the combined adjustment offers no real improvement in mapping accuracy to the position observations method. Since the internal structure of the block is so strong, the only place for meaningful improvement would be in better GNSS exposure station positions. Because the exposure stations positions did not improve, neither did the mapping accuracy.

For an aerial data set, the test data set is near-ideal: a metric camera, manual point measurements, dual-frequency data with only a moderate base-remote separation. If the currently-implemented combined adjustment cannot provide a benefit under these circumstances, then it is not likely to provide a benefit for any aerial data set. However, this does not invalidate the method. For instance, improved error and covariance modelling would likely enable successful ambiguity resolution, which would improve the entire process.

An obvious concern is that implementation errors (as opposed to the already-identified inadequacies) are masking the true performance of the combined adjustment. The operation of the photogrammetric component of the combined adjustment has been thoroughly confirmed through the adjustments of simulated data. Furthermore, implementation errors in bundle adjustments tend not to manifest themselves as subtle degradations in performance; instead, catastrophic errors typically occur. With GNSS adjustments, however, the likelihood of subtle, performance degrading errors is higher. To check that this was not the case, several shorter, known, baselines were adjusted. For these shorter baselines the effects the inadequate error and covariance modelling are not significant. In all cases, the combined adjustment results, including ambiguity resolutions, were correct, giving confidence in the implementation.

An observation unrelated to the integration technique was the mapping accuracy indifference to the GNSS data. For instance, using single-frequency data with floating ambiguities gave check-point accuracies that were virtually the same as those available from the most well-controlled network configurations. This indicates that difficult and possibly unreliable

integer ambiguity fixing may not be necessary at all, and that cheaper single-frequency receivers may be sufficient for the most commonly encountered block configurations such as the one used here.

6.1.3 Inter-processor Communication

Testing using the inter-processor communication integration approach will again focus on both the check-point accuracy and on the accuracy of the GNSS positioning. In the latter case, dual-frequency, integer ambiguity positions from the Grafnav GNSS processor were used as the basis of comparison. The processor used for the tests was, as previously outlined, an all new GNSS processor designed specifically for the task. The results from the combined adjustment indicated that noise in the photogrammetric block was at about the same level of noise in a float solution, and that no ambiguity resolution improvement could be expected. Consequently, the testing of the inter-processor communication used only single-frequency data, and only real (float) ambiguities were estimated.

In the first test done using the inter-processor communication approach, no accuracy improvement was seen in either the GNSS positions or the check-point positions. In both cases accuracy after the CUPT feedback was essentially the same as before the CUPT feedback. Given the benign nature of the test network (i.e., its clean GNSS data and good imaging geometry), and the similar results observed in the combined adjustment testing, these results were to be expected. The check-point accuracies both before and after the CUPT feedback, shown in Table 6.18, were at the same level as the nominal tests of Tables 6.1 and 6.3. As with the combined adjustment, using single-frequency data and float ambiguities gave mapping accuracies equivalent to those when dual-frequency data was used with fixed ambiguities. This again implies that difficult and potentially unreliable integer-ambiguity fixing need not always be attempted.

To simulate a more challenging data set, a second test was done in which cycle-slips were induced on all satellites' data streams in between two of the strips, causing the GNSS

Table 6.18: Check-point statistics for inter-processor communication approach

(a) Before CUPTs

	Horizontal	Vertical
Mean (m)	0.19	-0.10
Std. Dev. (m)	0.12	0.20
RMS (m)	0.22	0.22

(b) After CUPTs

	Horizontal	Vertical
Mean (m)	0.21	-0.13
Std. Dev. (m)	0.14	0.20
RMS (m)	0.25	0.23

Table 6.19: GNSS-position statistics for inter-processor communication approach with forced filter reset between strips

(a) Before CUPTs

	Horizontal	Vertical
Mean (m)	0.16	-0.03
Std. Dev. (m)	0.16	0.24
RMS (m)	0.23	0.24

(b) After CUPTs

	Horizontal	Vertical
Mean (m)	0.17	0.08
Std. Dev. (m)	0.11	0.18
RMS (m)	0.20	0.20

filter's ambiguity estimates to be reset. In this case, the feedback of the CUPTs into the GNSS processor did provide a small improvement in the accuracy of the GNSS positions that followed the reset. This improvement is shown in Figures 6.2 and 6.3, and is reflected in the standard deviations and RMSs of Table 6.19. In Figures 6.2 and 6.3 the first CUPT is indicated by the dashed vertical line; position errors following this CUPT are reduced.

Unfortunately, the improved GNSS positioning did not translate into improved photogrammetric mapping accuracy. Here again, however, the reason seems to be that the strength of the photogrammetric block is such that the degraded, cycle-slipped positions do not substantially affect results even before the CUPTs are applied. As can be seen in Table 6.20, accuracies after the CUPT-improved GNSS positions were used in the bundle adjustment were actually slightly worse than before. Since the actual exposure station position observations are themselves (slightly) more accurate, this suggests that the observations are being incorrectly weighted in the adjustment. The weight of the observations comes, in turn, from the output of the GNSS processor, and this indicates that the covariance modelling in

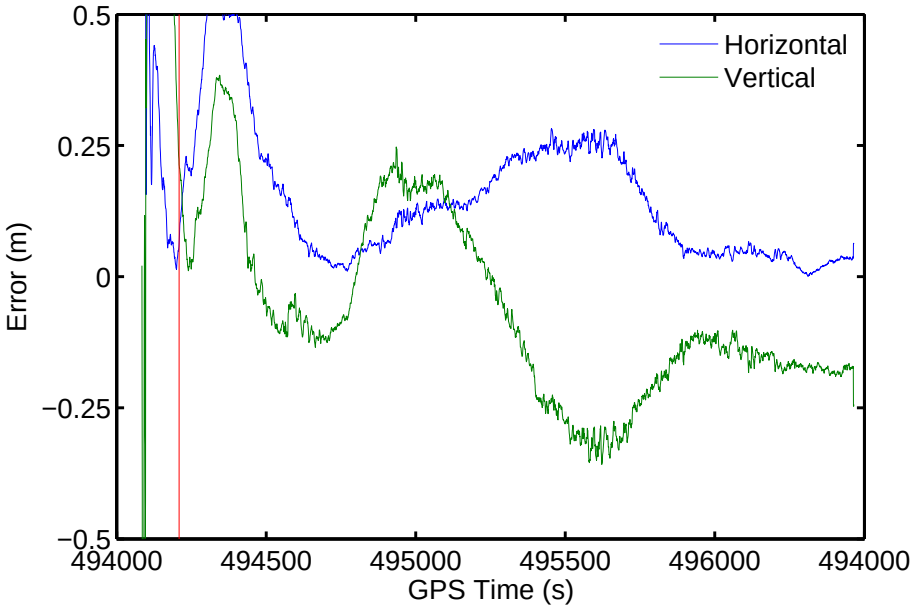


Figure 6.2: GNSS position errors before CUPT feedback

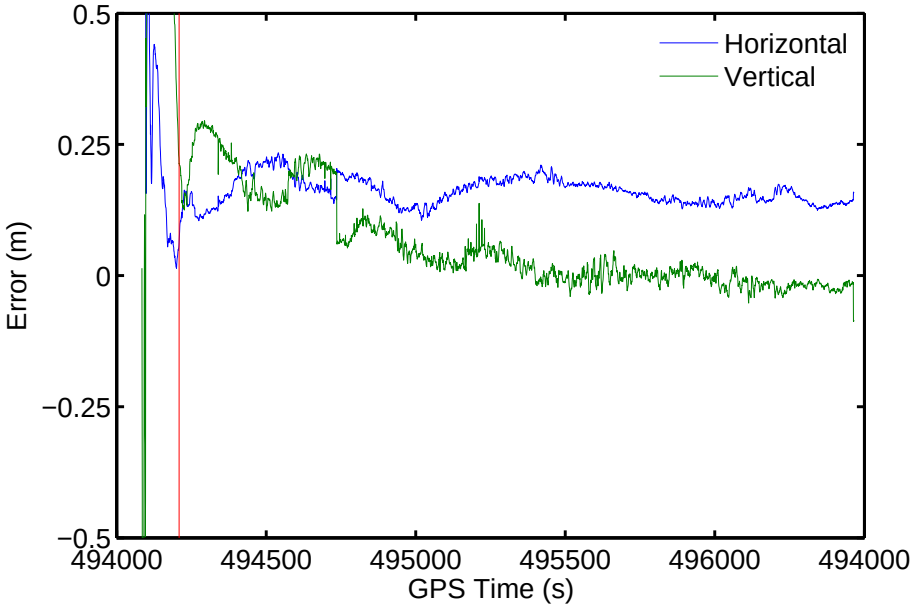


Figure 6.3: GNSS position errors after CUPT feedback

the processor is inadequate.

Table 6.20: Check-point statistics for inter-processor communication approach with forced filter reset between strips

(a) Before CUPTs

	Horizontal	Vertical
Mean (m)	0.23	-0.04
Std. Dev. (m)	0.12	0.16
RMS (m)	0.26	0.16

(b) After CUPTs

	Horizontal	Vertical
Mean (m)	0.26	-0.07
Std. Dev. (m)	0.18	0.21
RMS (m)	0.31	0.22

6.2 Simulated Terrestrial Mobile Mapping Campaign

As reviewed in Section 6.1.2, the results from aerial data set were, depending on the perspective, either inconclusive or disappointing. There were two reasons for the poor results: first, there were datum and calibration uncertainties in the data set that hindered analysis of results; second, inadequacies in the covariance and error modelling, exacerbated by the base station/rover separation, were causing ambiguity resolution to fail.

To enable further analysis while avoiding the two issues just identified, a terrestrial mobile mapping campaign was simulated. For this simulated data set, perfect, known datum translations and camera calibrations were used and more benign GNSS data with a shorter base station/rover separation was produced.

Only the combined adjustment integration strategy was tested with the simulated terrestrial data set. This was because this integration strategy had the greatest theoretical novelty and practical benefits, and also because the time required to do tests with the inter-processor communication strategy was not available.

6.2.1 Simulation Details

GNSS data was simulated by generating GPS signals using a Spirent Communications GSS7700 signal simulator, and collecting observations from those signals using a geodetic-grade NovAtel OEM-V3 receiver. The hardware signal simulator/receiver combination was used instead of simulating the measurements directly using the software measurement simulator of Sec-

tion 4.3.1 because the latter could not output the RINEX format ephemerides required by the adjustment software. Also, the signal simulator, coupled with a real receiver, meant more realistic measurements could be simulated.

The GNSS simulator was used to simulate signals that would be observed by a vehicle travelling on a trajectory representative of that encountered by a land-based mobile mapping system. The trajectory consisted of a rectangular "racetrack" pattern: four 2 km straights connected by corners with radii of 10 m. The maximum velocity on the straights was 60 km/h, while the cornering speed was 15 km/h. Uniform acceleration and deceleration into and out of the corners took place over 250 m. From the signals, dual-frequency observations were taken at 2 Hz. Signals were also generated and 2 Hz dual-frequency observations collected for a fixed base station at one corner of racetrack, with the maximum base/station rover separation being less than 3 km.

Photogrammetric data was simulated using the combined adjustment software operating in simulation mode. In this mode, image measurements are produced by back projection, using given interior orientations, lens distortions, lever-arms, boresight angles, exterior orientations, and object-space feature positions. The camera parameters were based upon real calibration information from the Video-INS-SATellite (VISAT) mobile mapping system. Two cameras, forward facing with a toe-in of about 6 ° and separated by about 2 m, were assumed to be mounted on the roof of a vehicle. Both cameras had focal lengths of 1700 pixels, image extents of 1600×1200 pixels, and no lens distortions. The vehicle was assumed to always be level, and exposures from both cameras were simulated every 15 m. A corridor of object space points along the trajectory was simulated: on both sides of the vehicle points were generated 2 m above and below the antenna and offset from the trajectory by 5 m. Another set of points, level with the antenna were generated with a 10 m offset, also every 10 m, but in-between the first set of points. Image measurements were only simulated for object space points within 40 m of the cameras. This resulted in an average image measurement count of 13 per exposure. Normally distributed random noise was added to the back-projected

image measurements. The standard deviation of this noise was 0.25 pixels in both the x and y image plane directions.

To keep the number of parameters in the combined adjustment manageable, exposures were only simulated for approximately 1.2 km along two sides of the racetrack. A further reduction in parameters was made possible through the use of the georeferenced image measurement equations. Instead of requiring a pair of exterior orientation parameters for each epoch of images – one for each camera – only a single set of exterior orientation parameters are required – a shared (antenna) position and platform attitude. Using the georeferenced image measurements, there are 2415 non-GNSS parameters in the adjustments that follow; without the georeferenced image measurements, there would be 2895 parameters. Furthermore, not using the georeferenced image measurements would necessitate an additional 480 lever-arm constraint equations relating the camera positions to the antenna position, and an additional 240 relative-attitude constraint equations relating the camera-pair attitudes. All of these constraints are implicitly provided "for-free" with the georeferenced image measurements.

The simulated network consisted of 80 exposures, 645 object-space feature positions, and 2120 image measurements. The arrangement of the exposure antenna positions is shown in Figure 6.4. Figure 6.5 shows a detail of how the corridor of object-space points was constructed.

6.2.2 Combined Adjustment Results

As with the aerial data set of Section 6.1, comparisons will be made between object-space coordinate accuracies using the combined adjustment, and those from using the conventional position observations approach. In the latter case, position observations were again generated using Waypoint's Grafnav 7.0 commercial kinematic GNSS processor. The absolute accuracies reported from both techniques must be taken with a healthy dose of skepticism. Efforts were taken to make the simulated data as realistic as possible, but the simulated

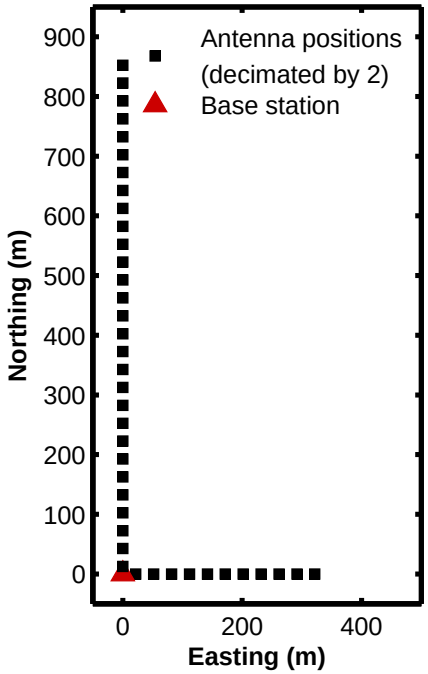


Figure 6.4: Simulated terrestrial network

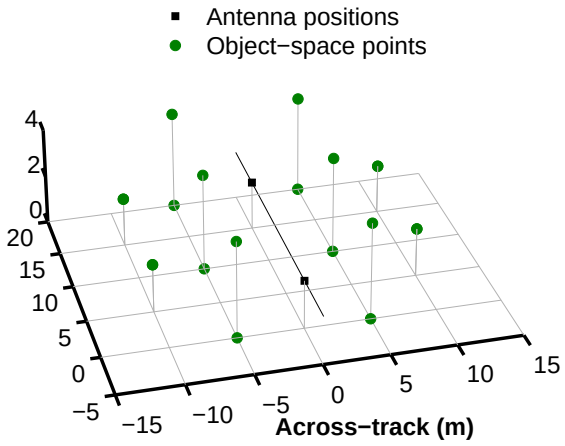


Figure 6.5: Simulated terrestrial network corridor detail

data is still more benign than any real environment. Most notably, the generated signals did not incorporate any multipath.

Undifferenced pseudoranges

As with the aerial data set, the first set of tests with the simulated data set used only undifferenced pseudoranges. Unlike with the real data set, it was not possible to use precise ephemerides or clock corrections. This is because the signal simulator signal simulator generated the ephemerides (contained in the GPS navigation message) by adding error to a real ephemerides. In other words, for the simulated data the real ephemerides is the truth, and the precise ephemerides corresponding to real ephemerides is meaningless. It was thought that the real ephemerides used as input to the simulator could be used as "precise" ephemerides in the combined adjustment. When this was attempted, however, the results were nonsensical, indicating either a possible configuration problem with the signal

simulator.

The check-point statistics from the combined adjustment and conventional position observations strategy are given in Tables 6.21 and 6.22, respectively. Judging from the large mean errors, the combined adjustment performs markedly worse than the position observations approach. However, these mean errors are likely an artifact of the signal simulation. When the combined adjustment's troposphere model is changed (or removed altogether), the mean position errors can shift by several metres. This sensitivity to troposphere model suggests that the smaller mean errors observed in Table 6.22 are primarily a reflection of the close agreement between Grafnav's and the signal simulator's troposphere models, rather than indicating that the positions observations technique outperforming the combined adjustment.

Table 6.21: Check-point statistics for combined adjustment done using undifferenced pseudoranges

	Horizontal	Vertical
Mean (m)	2.34	-4.01
Std. Dev. (m)	0.06	0.06
RMS (m)	2.34	4.01

Table 6.22: Check-point statistics for network controlled using GNSS exposure station position observations derived from undifferenced pseudoranges

	Horizontal	Vertical
Mean (m)	0.25	-0.31
Std. Dev. (m)	0.09	0.08
RMS (m)	0.27	0.32

When focus is given to the standard deviations, the combined adjustment gives results that are slightly more accurate than the position observations approach. As with the aerial adjustment, however, the standard deviations can be brought into closer agreement be varying the weights of the position observations.

Single-frequency pseudorange and carrier-phase double-differences

The next set of tests used single-frequency, double-differenced pseudoranges and carrier-phases. With such a short separation between the base station and rover antenna, both the commercial GNSS processor and the combined adjustment were able to resolve the ambiguities to integers. Tables 6.23 and 6.24 contain the check-point statistics for these two fixed-ambiguity solutions. The combined adjustment results are marginally poorer than the results from using the exposure station position observations, which is still a reflection of the poorer GNSS error and covariance modelling in the combined adjustment relative to the commercial processor.

Table 6.23: Check-point statistics for combined adjustment done using double-differenced single-frequency pseudoranges and carrier-phases, fixed ambiguities

	Horizontal	Vertical
Mean (m)	0.05	0.00
Std. Dev. (m)	0.06	0.02
RMS (m)	0.07	0.02

Table 6.24: Check-point statistics for network controlled using GNSS exposure station position observations derived from double-differenced single-frequency pseudoranges and carrier-phases

	Horizontal	Vertical
Mean (m)	0.04	0.00
Std. Dev. (m)	0.05	0.01
RMS (m)	0.07	0.01

Table 6.25 contains the check-point statistics for the combined adjustment's single-frequency float-ambiguities solution. Results are the same as when the ambiguities are fixed. This indicates that the float solution is of high quality, evidence for which could already be found in the successful ambiguity resolution. Indeed, of the nine L1 ambiguities estimated in the combined adjustment, eight of the real-valued estimates are within a tenth of a cycle of their integer values. After decorrelation, the estimates move even closer to their integer values: no decorrelated ambiguity is more than three-hundredths from its in-

teger value. This illustrates how decorrelation can improve the precision of the ambiguities, possibly allowing the use of a less-optimal integer-estimator than integer least squares. In this case, even rounding would have been sufficient.

Table 6.25: Check-point statistics for combined adjustment done using double-differenced single-frequency pseudoranges and carrier-phases, float ambiguities

	Horizontal	Vertical
Mean (m)	0.05	-0.03
Std. Dev. (m)	0.06	0.02
RMS (m)	0.07	0.03

The quality of the combined adjustment's estimation, both for the ambiguities and the object space positions, is due mostly to the clean GNSS data used. However, it is again a reflection of invariance of GNSS-controlled photogrammetry to the GNSS positions for a well-constructed block of imagery. As in the aerial test data set, the results from this test indicate the ambiguity fixing need not be always attempted.

Dual-frequency pseudorange and carrier-phase double-differences

The chief advantage of dual- or multi-frequency data is the ability to estimate the ionosphere errors, permitting ambiguity resolution over much longer base-rover separations. With the simulated data-set, this separation is very small, and so it is not expected that dual frequency data would provide any appreciable benefit over single-frequency data. The check-point statistics in Tables 6.26, 6.27, and 6.28 confirm this. For both the combined adjustment and conventional position observations integration strategies, results are only marginally better than when single-frequency data is used. The combined adjustment's float solution using dual frequency data is slightly better than the same when single-frequency data is used, but this is likely a consequence of the increased redundancy from using the L2 observations.

Table 6.26: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases, float ambiguities

	Horizontal	Vertical
Mean (m)	0.04	-0.01
Std. Dev. (m)	0.05	0.01
RMS (m)	0.07	0.02

Table 6.27: Check-point statistics for network controlled using GNSS exposure station position observations derived from double-differenced dual-frequency pseudoranges and carrier-phases

	Horizontal	Vertical
Mean (m)	0.04	0.00
Std. Dev. (m)	0.05	0.01
RMS (m)	0.07	0.01

Table 6.28: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases; fixed ambiguities

	Horizontal	Vertical
Mean (m)	0.04	0.00
Std. Dev. (m)	0.05	0.01
RMS (m)	0.07	0.01

Unusual network and data configurations

The simulated data can be used to illustrate the utility of the combined adjustment for network and data configurations different than those available using the traditional position observations integration strategy. As with the aerial data, two obvious candidates are networks controlled using photogrammetric ground control, and networks where position observations would not be available at all or some of the exposures due to insufficient GNSS observations.

Table 6.29 contains the check-point statistics for a network controlled using a single, perfect, control point in about the middle of the block. Results are the same as for the network controlled using the fixed GNSS master station. As explained in Section 6.1.2 above, the benefit of this approach is that it would enable positions to easily be reported in

the photogrammetric control's datum. An additional advantage is that would allow existing control to be reused, rather than new, often mission-specific, GNSS-control being established (although establishing these networks is often made trivial due to the availability of GNSS data and reference stations like those provided by the IGS).

Table 6.29: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases; fixed ambiguities; single, high-quality, photogrammetric ground control

	Horizontal	Vertical
Mean (m)	0.04	0.01
Std. Dev. (m)	0.05	0.01
RMS (m)	0.07	0.02

It is likely rare that there would be a single or limited number of high-quality control points available that were also identifiable in imagery. Instead, a more likely scenario would be that there are several "Geographic Information System (GIS)" quality points available. These points, with accuracies at about the decimetre level, may have been collected from an aerial photogrammetric survey or in terrestrial surveys using single-frequency GNSS. To examine the use of such less-perfect control points, five photogrammetric ground control position observations were simulated with co-ordinate errors of 1 decimetre standard deviation. The control points were spaced approximately evenly along the trajectory at about 250 m intervals. Table 6.30, containing the results from the combined adjustment of this network, shows that such less-accurate control points can still give provide good overall network precision. Larger mean and RMS errors indicate that accuracy suffers, but this is a consequence of the network datum being less precisely defined: in effect, the mean errors of the control points determines the mean errors of all the points. Additional control points would reduce the overall mean error, provided that their errors are distributed normally.

The use of multiple photogrammetric ground control points carries some risk: if the control points have errors in them that are not represented by their standard deviations, then this will negatively affect both the adjusted positions and ambiguity resolution. Of

Table 6.30: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases; fixed ambiguities; five photogrammetric ground control points with 1 decimetre standard deviation co-ordinate errors

	Horizontal	Vertical
Mean (m)	0.06	0.04
Std. Dev. (m)	0.05	0.02
RMS (m)	0.07	0.04

course, this is true in general: erroneous control points in any network will distort the adjusted network. With ambiguity resolution, however, control errors can have a compounded impact. The erroneous control points will distort and bias both the network positions and the real-valued ambiguities. This, in turn, can lead to incorrect ambiguity resolution, which can introduce further distortions and biases into the positions.

Aside from the use of photogrammetric datum control, the other novel data configuration that the combined adjustment enables is the use of GNSS-data when the exposure epochs have less than 4 GNSS observations. In terrestrial mobile mapping this is a common occurrence: all or some of the satellite signals can frequently be occluded by trees or buildings along the trajectory. To demonstrate how the combined adjustment can bridge such outages, simulations were done where the middle third of exposures had either no GNSS observations, or had observations from only the three highest-elevation satellites. In the latter case, the simulation assumed that the carrier-phases from the three satellites were tracked continually during the data interruptions on the other satellites. Tables 6.31 and 6.32 contain the check-point statistics from these adjustments. In both cases ambiguities could be resolved both before and after the data interruptions for the satellites not continually tracked.

Of course, the aggregate statistics in Tables 6.31 and 6.32 do not tell the whole story. Figures 6.6 and 6.7 show in more detail the error behaviour during the complete and partial signal blockages, respectively. In both cases, the errors in the object-space positions follow the errors in the exposure positions. Error magnitudes during the blockages are basically a

Table 6.31: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases; fixed ambiguities; complete signal blockage

	Horizontal	Vertical
Mean (m)	0.13	0.08
Std. Dev. (m)	0.15	0.10
RMS (m)	0.20	0.12

Table 6.32: Check-point statistics for combined adjustment done using double-differenced dual-frequency pseudoranges and carrier-phases; fixed ambiguities; partial (all but 3 highest-elevation satellites) signal blockage

	Horizontal	Vertical
Mean (m)	0.05	0.04
Std. Dev. (m)	0.05	0.05
RMS (m)	0.08	0.06

function of the distance from the signal blockage boundary, with errors growing quadratically. When observations from continually-tracked satellites are available, they help bound the error growth: observations from three continually-tracked satellites reduce the maximum position errors from approximately 0.7 m to 0.2 m. The degree of this bounding is dependant on the the number of satellites that are continually tracked; Figure 6.8, for instance, shows that when only two satellites are continually tracked, the maximum position errors are only reduced to 0.35 m.

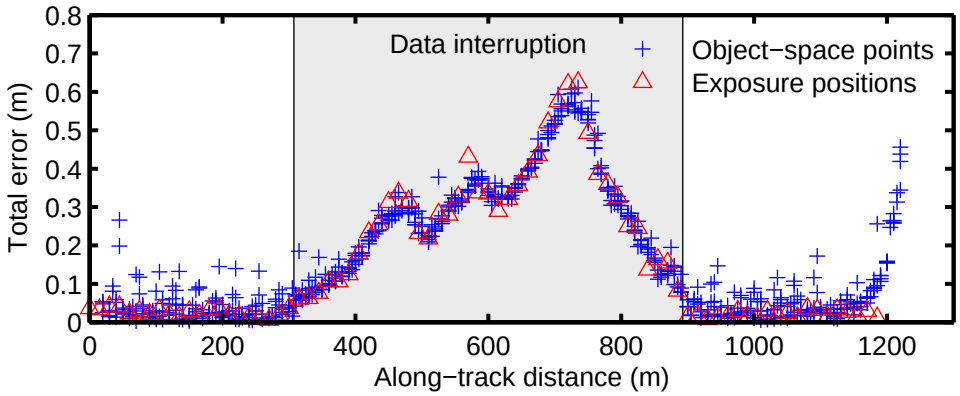


Figure 6.6: Total position errors during complete signal blockage

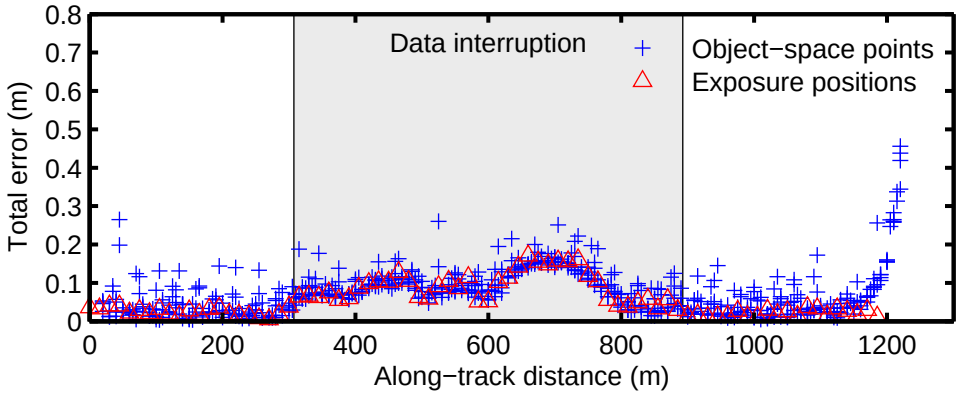


Figure 6.7: Total position errors during partial (all but 3 highest-elevation satellites) signal blockage

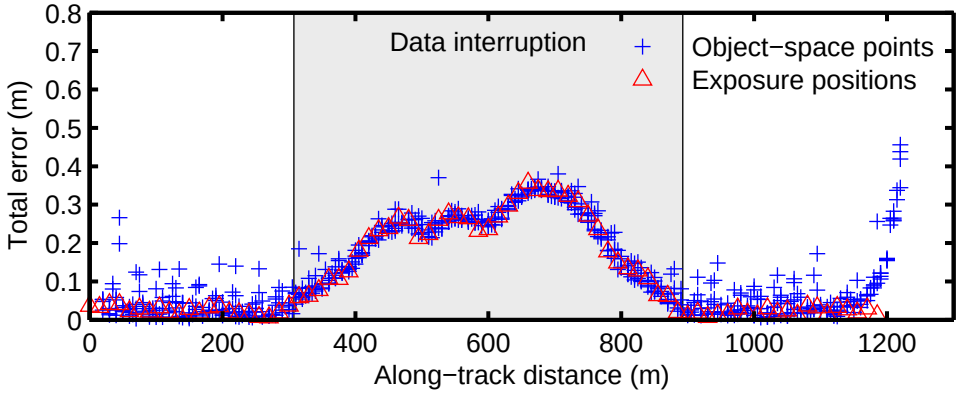


Figure 6.8: Total position errors during partial (all but 2 highest-elevation satellites) signal blockage

Analysis

By removing the calibration and datum uncertainties, and by mitigating the impact of inadequate error and covariance modelling, the testing using the simulated terrestrial mobile mapping data better validates the combined adjustment integration technique than the aerial data set. However, improvements in mapping accuracy are still not observed: in all current data configurations the combined adjustment yielded mapping accuracies that were essentially equivalent to the current position observations integration strategy. The simulated data set was again used to show how photogrammetric ground control can replace a GNSS base station with known co-ordinates, and how less-accurate photogrammetric ground

control can still be used to good effect if present in sufficient numbers. Additionally, the simulated data set demonstrated how the combined adjustment can help bridge GNSS data interruptions. Signals continually-tracked through the interruptions can help bound the position error growth; information from these signals are otherwise discarded in the position observations integration strategy.

In Section 6.1.2, the possibility of their being some implementation error in the combined adjustment was raised. The combined adjustment's successful ambiguity resolution and close agreement with the position observations integration strategy with this simulated data set indicate that this is not the case. Once again, this points towards error and covariance modelling deficiencies.

Chapter 7

Conclusions

In the research described in this thesis, two novel strategies for integrating GNSS and photogrammetric data were introduced, implemented and tested:

- Inter-processor communication between a kinematic GNSS Kalman filter and a photogrammetric bundle adjustment.
- A combined least-squares adjustment of both GNSS and photogrammetric observations.

The motivation behind these new strategies was to improve upon the current position-observations strategy's mapping accuracy, reliability, and operational difficulties.

7.1 Key Findings

Testing of both integration strategies showed some benefits. For the inter-processor communication strategy, feedback of the photogrammetrically derived positions into the GNSS Kalman filter was able to improve platform positioning following signal outages. Unfortunately, this improvement did not translate into improved mapping accuracy. For the combined adjustment, testing demonstrated how it enabled photogrammetric control to replace fixed GNSS base stations with no loss in accuracy, and with some operational benefits

like being able to disregard datum discrepancies. Tests with the combined adjustment also showed how it allowed the use of GNSS observations during partial signal blockages, when, in the position observations approach, they would otherwise be discarded. Use of such observations helped to bound mapping and exposure position error growth. Finally, the combined adjustment greatly simplified and streamlined the data integration process by replacing the two (or more) software packages currently used in current position-observations integration strategy with a single program.

Unfortunately, neither the combined adjustment nor the interprocessor communication integration strategies as implemented were able to improve mapping accuracies or ambiguity resolution for the data sets used in testing. For both strategies the reason for the lack of accuracy improvement is two-fold. First, the resection accuracies from the photogrammetric data were at about the same level as the float ambiguity-derived positions; consequently, neither the GNSS positions nor the ambiguities were seeing an appreciable benefit from the photogrammetric information. Second, the GNSS error and covariance modelling was inadequate. With the real data set, this lead to biased float ambiguity solutions with deformed covariances, effectively rendering impossible ambiguity resolution over medium to long baselines.

A conclusion not related to the integration strategies is that for a well constructed GNSS-controlled block, mapping accuracy is largely indifferent to the GNSS data or processing strategy. With both integration strategies processing single-frequency data and not fixing ambiguities gave mapping accuracies that were equivalent to the accuracies available when dual frequency data was used with fixed ambiguities. The reason for this is that in well-constructed blocks the internal block structure is strong enough to restrict deformation of the network, and the role of the GNSS-derived positions is largely reduced to providing an "averaged" datum-definition. The implication of this is that ambiguity resolution need not be attempted, thus avoiding the reliability issues that arise from its use.

7.2 Perspective

The testing in this thesis primarily focused on the accuracy of the new integration strategies relative to the existing position observations strategy. By this criteria the results from the new strategies were not encouraging. However, this focus on accuracy was, perhaps, a mistaken approach. The accuracy of all integration strategies hinge largely on the GNSS positioning accuracy, whether explicitly, as with the position observations and inter-processor communications approaches, or implicitly, as in the combined adjustment. It was maybe not realistic to expect an entirely new GNSS engine to compete against an industry-leading commercial engine with 20 years of testing, development, and improvements behind it. In retrospect, perhaps more attention should have been given to demonstrating the improved reliability and more flexible data-handling capability of the combined adjustment. These criteria may have been better for showing the benefits of the two new integration strategies.

The progress of this research was hampered by a lack of suitable data sets. Several data sets were generously provided by multiple parties; however, in almost all cases the data sets were missing required information like lever-arms, control points, GNSS base-station data, etc. The reasons for these deficiencies include:

- Most aerial-photogrammetric campaigns are done with little control: sometimes even with as little as a single Ground Control Point (GCP) that is used only to reconcile datum translations. Obviously, a reasonable number of GCPs are necessary in order to calculate meaningful statistics in tests like those done here.

- Lever-arms between the GNSS antenna and the camera are rarely measured directly and they are never measured in a frame aligned with the camera frame. Instead, a combination of lever-arms is measured to some other reference point and in some other frame (or frames). To calculate the antenna-to-camera lever-arm in the camera-aligned-frame a host of intermediate vectors, rotation angles, and rotation sequences are required. If even one of these information is missing, it is not possible to calculate

the required lever-arm.

- Due to the indifference of triangulation results on GNSS positioning accuracy (as demonstrated by the results of Chapter 6), relative techniques are, in some cases, being replaced with PPP. When PPP is used, GNSS base-station data is not available.

For all these reasons, it is surprisingly difficult to obtain complete data sets like those required for research like that done here, even though virtually all aerial-photogrammetry done today is GNSS controlled.

7.3 Additional Contributions

While the majority of this work dealt specifically with the new integration strategies, there were some other contributions. Notably, the introduction of the georeferenced image measurement equations. These equations are much better suited for integrated sensor orientation than the normal collinearity image measurement equations because:

- they can natively use the positions and attitudes output by the mobile mapping system navigation sensors
- they eliminate the requirement for lever-arm and relative-orientation constraints in systems with multiple-imaging sensors, thereby reducing adjustment dimensionality.

An additional contribution was the practical software development considerations covered in Chapter 5. Such considerations and details are typically neglected in the reporting of Geomatics research.

7.4 Further Explorations

There are a great many areas where the existing operation of the combined adjustment implementation could be improved or new functionality added:

- **Improve the covariance modelling**: Weighting the undifferenced observations by elevation angle only and disregarding physical correlations between them was believed to be a cause of the incorrect ambiguity resolution observed, and for the generally disappointing GNSS positioning. Improving the covariance modelling by accounting for correlations between undifferenced observations over space and time would improve results. For instance, at the same epoch the variability of observations on different frequencies due to the troposphere is the same; this correlation should be rigorously accounted for. Similarly, observations made to satellites that are close together will have a similar variability due to the troposphere since the signals will have had similar paths through the troposphere; again, this correlation should be considered.

- **Add atmospheric error parameters to the adjustments using undifferenced pseudoranges**: Parameters accounting for tropospheric and ionospheric delay could be added to the combined adjustment. This would improve adjustment results when undifferenced pseudoranges are used with photogrammetric ground control. For the troposphere, a zenith delay parameter could be added for each observation epoch, and a mapping function used to apply it to the observations. For the ionosphere, an individual delay would have to be added for each satellite. Stochastic constraints could then be added to relate delays at different epochs.

- **Add atmospheric error parameters to the adjustments using double-differenced data**: Like adjustments with undifferenced observations, adjustments with double-difference observations would benefit from the addition of atmospheric error parameters. For single-frequency data, in particular, ionospheric delay parameters might, with photogrammetric ground control, enable successful ambiguity resolution over longer baselines. The ionospheric delay should be observable because of its different sign in the pseudorange and carrier-phase observation equations, and because it would be separable from the ambiguities since the ambiguities are modelled as

constants.

- **Use the undifferenced carrier phases in the adjustment**: The undifferenced carrier phases could be used in the adjustment, and the undifferenced ambiguities estimated. Coupled with precise orbit and clock corrections, this would be akin to doing PPP in the adjustment, except that the photogrammetric network connections should improve the process.

- **Use time-differenced GNSS observations in the adjustment:** It would be straightforward to adapt the combined adjustment to use time-differenced GNSS observations. By differencing carrier-phases over time the ambiguity term can be removed. Consequently, ambiguity resolution would not have to be performed.

- **Use relative GNSS data in the adjustment:** Recently, Blázquez (2008) introduced the concept of using GNSS-derived velocities in bundle adjustments. The velocities were used in observation equations relating exposure positions at adjacent epochs, in a novel approach termed "spatial relative control". While the technique has its limitations (in particular, the difficulties with accelerations over the exposure epoch intervals seem to be understated), it would be interesting to explore its use in the combined adjustment. The concept could potentially be expanded into the observation domain, where instead of positions and velocities being used, (differenced) carrier-phases and Doppler would be used instead.

- **Modify the ambiguity resolution**: Currently, the integer search is conducted for all ambiguities. Even with decorrelation, for multiple baselines the search can still take a long time. A more efficient technique might be to only fix at most n accurate ambiguities at each iteration. At any one iteration this should quicken the search, while sequential fixing of ambiguities over iterations should mean the same final set of fixed ambiguities would ultimately be found.

- **Take advantage of the sparsity of the normal matrix**: Not taking advantage of normal matrix sparsity is the sole feature separating the combined adjustment from commercial photogrammetric bundle adjustments. As explained in Section 5.3.5, the reason it was not implemented was because the generic structure of the adjustment would make a generic implementation difficult. However, an envisioned general scheme that would enable the continued use of a generic solver interface would be as follows: On the first iteration, rather than processing observations, the structure of the normal matrix would instead be analysed and a re-ordering index determined. Then, during subsequent iterations, the observation processing would use this index that maps the original parameter locations to their re-ordered locations.

- **Add support for linear features:** The use of linear features in photogrammetry has recently received much attention. It is natural, therefore, to add support for linear features to the adjustment. The linear features could be used by all components of the combined adjustment. For instance, terrestrial survey observations could be made to the linear features, or GNSS stations could be constrained along linear paths.

- **Add dynamic constraints on the GNSS positions or other GNSS states:** In the combined adjustment there was no connecting kinematic model for the GNSS positions. It would, however, be straightforward to add stochastic constraints based on, for instance, a first-order Gauss-Markov model. Were other GNSS states to be added to the adjustment – for example, ionosphere delay parameters – such stochastic constraints would also be helpful, if not practically required.

Like the combined adjustment, the interprocessor communication approach would also benefit from improved error and covariance modelling in the GNSS processor. Additionally, for the simple testing done in this research the two-way exchange of position observations between the bundle adjustment and the Kalman filter was done manually. Manual exchange is tedious and error-prone; thus, the interaction between the two processors should be au-

tomated. This could most easily be done with a supervisor program that automatically takes the output from each processor and reformats for the other, while controlling the operation of both. Alternatively, the two processors could be combined in a single program, although this runs contrary to the perceived ease-of-implementation benefit of the integration strategy.

One final avenue for further exploration would be to implement the combined Kalman filter integration approach identified in Section 5.2.3.

References

Ackermann, F. 1984. "Utilization of Navigation Data for Aerial Triangulation". In proceedings of *XV ISPRS Congress*, volume 25 of *International Archives of Photogrammetry and Remote Sensing*, pages 1–9, Rio de Janeiro, Brazil. International Society of Photogrammetry and Remote Sensing (ISPRS). Part A3.

Ackermann, F. 1992. "Kinematic GPS Control for Photogrammetry". *The Photogrammetric Record*, 14(80):261–276.

Agrell, E., Eriksson, T., Vardy, A., and Zeger, K. 2002. "Closest Point Search in Lattices". *IEEE Transactions on Information Theory*, 48(8):2201– 2214.

ARINC 1985. *705-5 Attitude and Heading Reference System (AHRS)*. Aeronautical Radio, Incorporated.

Bäumker, M. and Heimes, F.-J. 2001. "New Calibration and Computing Method for Direct Georeferencing of Image and Scanner Data Using the Position and Angular Data of an Hybrid Navigation System". In proceedings of *Proceedings of OEEPE-Workshop Integrated Sensor Orientation*, Hannover, Germany.

Beran, T., Bisnath, S. B., and Langley, R. B. 2004. "Evaluation of High-Precision, Single-Frequency GPS Point Positioning Models". In proceedings of *ION GNSS 2004*, pages 1893–1901, Long Beach. The Institute of Navigation (ION).

Blewitt, G. 1997. "Basics of the GPS Technique: Observation Equations". In Johnson, B., editor, *Geodetic Applications of GPS*, pages 10–54. Nordic Geodetic Commission, Gävle, Sweden.

Blázquez, M. 2008. "A New Approach to Spatio-Temporal Calibration of Multi-Sensor Systems". In proceedings of *XXI ISPRS Congress*, volume 37-B1 of *International Archives of Photogrammetry and Remote Sensing*, Bejing. International Society of Photogrammetry and Remote Sensing (ISPRS).

Boehm, B. W. 1981. *Software Engineering Economics*. Prentice Hall, New York, 3 edition.

Brown, D. C. 1976. "The Bundle Adjustment – Progress and Prospects". In proceedings of *XIII Congress of the ISP*, Helsinki. International Society of Photogrammetry (ISP). Paper 3-03, 33 pages.

Brown, R. G. and Hwang, P. Y. 1997. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc., New York, 3 edition.

Brunner, F. K. and Welsch, W. M. 1993. "Effect of the troposphere on GPS measurements". *GPS World*, 4(1):42–51.

Chaplin, B. A. 1999. "Motion Estimation From Image Sequences". Master's thesis, University of Calgary, Calgary, Canada.

Cocard, M. and Geiger, A. 1994. "Systematic search for all possible widelanes". In proceedings of *Proceedings of The Sixth International Geodetic Symposium on Satellite Positioning*, pages 312–318, Columbus, OH.

Collins, P. 1999. "An Overview of GPS Inter-frequency Carrier Phase Combinations". URL: `http://gauss.gge.unb.ca/papers.pdf/L1L2combinations.collins.pdf`. Unpublished paper. Accessed 18 Sep 2008.

Collins, P., Langley, R., and LaMance, J. 2001. "Limiting Factors in Tropospheric Propagation Delay Error Modelling for GPS Airborne Navigation". In proceedings of *ION 52nd Annual Meeting*, pages 519–528, Cambridge, MA., USA. The Institute of Navigation (ION).

Cooper, M. and Robson, S. 1996. "Theory of Close Range Photogrammetry". In Atkinson, K., editor, *Close Range Photogrammetry and Machine Vision*, pages 9–50. J.W. Arrowsmith, Bristol.

Dai, L., Han, S., and Rizos, C. 2001. "Performance analysis of integrated GPS/GLONASS carrier phase-based positioning". *Journal of Geospatial Information Science*, 4(4):9–18.

de Jonge, P. and Tiberius, C. 1996. "The LAMBDA method for integer ambiguity estimation: implementation aspects". LGR-Series 12, Delft Geodetic Computing Centre.

Ellum, C. 2002. "The Development of a Backpack Mobile Mapping System". Master's thesis, University of Calgary, Calgary, Canada.

Ellum, C. and El-Sheimy, N. 2002. "The Calibration of Image-Based Mobile Mapping Systems". In proceedings of *2nd Symposium on Geodesy for Geotechnical and Structural Engineering*, Berlin, Germany. The International Association of Geodesy (IAG).

Euler, H.-J. and Schaffrin, B. 1990. "On a measure of discernibility between different ambiguity solutions in the static-kinematic GPS-mode". In Schwarz, K.-P. and Lachapelle, G., editors, *International Symposium on Kinematic Systems in Geodesy, Surveying and Remote Sensing (KIS90)*, pages 285–295, Banff, Canada. The International Association of Geodesy (IAG).

Feess, W. and Stephens, S. 1986. "Evaluation of GPS ionospheric time delay algorithm for single-frequency users". In proceedings of *Position, Location and Navigation Symposium (PLANS 1986)*, pages 206–213, Las Vegas, NV. Institute of Electrical and Electronics Engineers (IEEE).

Ford, T. and Hamilton, J. 2003. "A New Positioning Filter: Phase Smoothing in the Position Domain". *Navigation: Journal of the Institute of Navigation (ION)*, 50(2):65–78.

Fotopoulos, G. and Cannon, M. 2001. "An overview of multi-reference station methods for cm-level positioning". *GPS Solutions*, 4(3):1–10.

Gao, Y., McLellan, J., and Schleppe, J. 1996. "An optimized GPS carrier phase ambiguity search method focusing on speed and reliability". *IEEE Aerospace and Electronic System Magazine*, 1(12):22–26.

Gao, Y. and Shen, X. 2002. "A New Method for Carrier Phase Based Precise Point Positioning". *Navigation: Journal of the Institute of Navigation (ION)*, 49(2):109–116.

Gao, Y. and Sideris, M. 2001. "ENGO 563 Lecture Notes – Data Analysis in Engineering". Technical report, Department of Geomatics Engineering, The University of Calgary, Calgary, Canada.

Gelb, A. 1974. *Applied Optimal Estimation*. MIT Press, Cambridge, MA.

Gentleman, W. M. 1973. "Least Squares Computations by Givens Transformations Without Square Roots". *Journal of the Institute of Mathematics and its Applications*, 12(1):51–58.

Golub, G. H. and Loan, C. F. V. 1996. *Matrix computations*. Johns Hopkins University Press, Baltimore, 3 edition.

GPSoft 2003. "Satellite Navigation (SatNav) ToolBox 3.0". URL: `http://www.gpsoftnav.com/satnav.html`. Accessed 10 May 2006.

Granshaw, S. 1980. "Bundle Adjustment Methods in Engineering Photogrammetry". *Photogrammetric Record*, 10(56):181–207.

Griffiths, D. J. 1989. *Introduction to Electrodynamics*. Prentice Hall, Englewood Cliffs, NJ, 2 edition.

Han, S. 1997. "Quality-control issues relating to instantaneous ambiguity resolution for real-time GPS kinematic positioning". *Journal of Geodesy*, 71(6):351–361.

Han, S. and Dai, L. 2007. "Automatic decorrelation and parameter tuning real-time kinematic method and apparatus". United States Patent number 7298319.

Han, S. and Rizos, C. 1995. "A new method for constructing multi-satellite ambiguity combinations for improved ambiguity resolution". In proceedings of *ION GPS 1995*, pages 1145–1153, Palm Springs, CA. The Institute of Navigation (ION).

Han, S. and Rizos, C. 1996a. "Improving the computational efficiency of the ambiguity function algorithm". *Journal of Geodesy*, 70(6):330–341.

Han, S. and Rizos, C. 1996b. "Integrated method for instantaneous ambiguity resolution using new generation GPS receivers". In proceedings of *Position, Location and Navigation Symposium (PLANS 1986)*, pages 254–261, Atlanta, GA. Institute of Electrical and Electronics Engineers (IEEE).

Hassibi, A. and Boyd, S. 1998. "Integer Parameter Estimation in Linear Models with Applications to GPS". *IEEE Transactions on Signal Processing*, 46(11):2938–2952.

Hatch, R. 1982. "The Synergism of GPS Code and Carrier Measurements". In proceedings of *International Geodetic Symposium on Satellite Doppler Positioning*, pages 1213–1231, Las Cruces, NM.

He, G., Novak, K., and Feng, W. 1992. "Stereo Camera System Calibration with Relative Orientation Constraints". In El-Hakim, S. F., editor, *SPIE Vol. 1820 – Videometrics*, pages 2–8, Boston, MA. The International Society for Optical Engineering (SPIE).

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. 2001. *Global Positioning System: Theory and Practice*. Springer-Verlag, Vienna, 5 edition.

Hu, G., Abbey, D. A., Castleden, N., Featherstone, W. E., Earls, C., Ovstedal, O., and Weihing, D. 2004. "An approach for instantaneous ambiguity resolution for medium- to long-range multiple reference station networks". *GPS Solutions*, 9(1):1–11.

IGS 2008. "IGS Products". URL: `http://igscb.jpl.nasa.gov/components/prods.html`. Accessed 16 September, 2008.

Jacobsen, K. and Schmitz, M. 1996. "A New Approach of Combined Block Adjustment Using GPS-Satellite Constellation". In proceedings of *Proceedings of the 18th ISPRS Congress*, International Archives of Photogrammetry and Remote Sensing, Volume 31, PartB3, Vienna, Austria. International Society of Photogrammetry and Remote Sensing (ISPRS).

Jorgensen, P. S. 1989. "An Assessment Of Ionospheric Effects On The GPS User". *Navigation: Journal of the Institute of Navigation (ION)*, 36(2):195–204.

JPL 2003. "Highrate Precise Ephemerides". URL: `ftp://sideshow.jpl.nasa.gov/pub/jpligsac/hirate`. Accessed 16 September, 2008.

Julien, O., Alves, P., Cannon, M. E., and Lachapelle, G. 2004. "Improved triple-frequency GPS/GALILEO carrier phase ambiguity resolution using a stochastic ionosphere modeling". In proceedings of *ION National Technical Meeting 2004*, pages 441–452, San Diego, CA. The Institute of Navigation (ION).

Kampes, B. and Hanssen, R. 2004. "Ambiguity resolution for permanent scatterer interferometry". *IEEE Transactions on Geoscience and Remote Sensing*, 42(11):2446–2453.

Klobuchar, J. A. 1986. "Design and characteristics of the GPS ionospheric time delay algorithm for single frequency users". In proceedings of *Position, Location and Navigation*

Symposium (PLANS 1986), pages 280–286, Las Vegas, NV. Institute of Electrical and Electronics Engineers (IEEE).

Klobuchar, J. A. 1996. "Ionospheric Effects on GPS". In Parkinson, B. W. and Spilker, J. J., editors, *Global Positioning System: Theory and Applications. Vol. I.*, pages 485–515. American Institute of Aeronautics and Astronautics, Inc., Washington, DC.

Krakiwsky, E. J. 1990. *The Method of Least Squares: A Synthesis of Advances.* UCGE Reports Number 10003. Department of Geomatics Engineering, The University of Calgary, Calgary, Canada.

Kruck, E., Wübbena, G., and Bagge, A. 1996. "Advanced Combined Bundle Block Adjustment with Kinematic GPS Data". In proceedings of *Proceedings of the 18th ISPRS Congress*, International Archives of Photogrammetry and Remote Sensing, Volume 31, PartB3, Vienna. International Society of Photogrammetry and Remote Sensing (ISPRS).

Kuang, S. 1996. *Geodetic Network Analysis and Optimal Design: Concepts and Applications.* Ann Arbor Press, Inc, Chelsea, MI.

Kuntu-Mensah, P. and Hintz, R. J. 2001. "Airborne GPS-photogrammetry comes of age". *Surveying and Land Information Systems (SaLIS)*, 61(2):93–102.

Langley, R. B. 1993. "The GPS Observables". *GPS World*, 4(4):52–59.

Langley, R. B. 1998a. "Propagation of the GPS Signal". In Teunissen, P. J. and Kleusberg, A., editors, *GPS for Geodesy*, pages 111–150. Springer Verlag, Berlin, 2 edition.

Langley, R. B. 1998b. "Short Distance GPS Models". In Teunissen, P. J. and Kleusberg, A., editors, *GPS for Geodesy*, pages 457–482. Springer Verlag, Berlin, 2 edition.

Lawson, C. L. and Hanson, R. J. 1974. *Solving least squares problems.* Prentice-Hall, Inc., Englewood Cliffs, NJ.

Lee, C. 1999. *Mathematical Modelling of Airborne Pushbroom Imagery Using Point and Linear Features.* PhD thesis, Purdue University, Purdue, IN.

Leick, A. 1995. *GPS satellite surveying.* John Wiley and Sons, Inc., New York, second edition.

Liu, G. C. and Lachapelle, G. 2002. "Ionosphere weighted GPS cycle ambiguity resolution". In proceedings of *ION National Technical Meeting 2002*, pages 889–899, San Diego, CA. The Institute of Navigation (ION).

Liu, J. 2003. "Implementation and Analysis of GPS Ambiguity Resolution Strategies in Single and Multiple Reference Station Scenarios". Master's thesis, University of Calgary, Calgary, Canada.

Liu, L. T., Hsu, H. T., Zhu, Y. Z., and Ou, J. K. 1999. "A new approach to GPS ambiguity decorrelation". *Journal of Geodesy*, 73(9):478ś490.

Lucas, J. R. 1987. "Aerotriangulation without Ground Control". *Photogrammetric Engineering and Remote Sensing (PE&RS)*, 53(3):311–314.

Madani, M. and Shkolnikov, I. 2005. "Dynamic Drift Model for GPS/INS Post-Processed Trajectory of Frame Camera". In proceedings of *Proceedings of the ISPRS Hannover Workshop 2005: High-Resolution Earth Imaging for Geospatial Information*, Hannover, Germany. International Society of Photogrammetry and Remote Sensing (ISPRS).

Maplesoft 2008. "Maple 12 Professional -Math & Engineering Software - Maplesoft". URL: `http://www.maplesoft.com/products/maple/index.aspx`. Accessed 09 Oct. 2008.

Maxima 2008. "Maxima, a Computer Algebra System". URL: `http://maxima.sourceforge.net/`. Accessed 09 Oct. 2008.

Meade, M. E. 2003. "From the Ground Up: Direct georeferencing in aerial photography". *Point of Beginning*, May.

Mikhail, E. M. 1976. *Observations and Least Squares*. IEP-A Dun-Donnelley, New York.

Mikhail, E. M., Bethel, J. S., and McGlone, J. C. 2001. *Introduction to Modern Photogrammetry*. John Wiley and Sons, Inc., New York.

Mostafa, M. M. R. 2001. "Calibration In Multi-Sensor Environment". In proceedings of *ION GPS 2001*, pages 2693–2699, Salt Lake City. The Institute of Navigation (ION).

Niell, A. 1996. "Global Mapping Functions for the Atmosphere Delay at Radio Wavelengths". *Journal of Geophysical Research*, 11(B2):3227–3246.

Odijk, D. 2002. *Fast precise GPS positioning in the presence of ionospheric delays*. PhD thesis, Delft University of Technology, Delft, The Netherlands.

Pain, H. J. 1993. *The Physics of Vibrations and Waves*. John Wiley & Sons, Chichester, England, 4 edition.

Pinto, L. and Forlani, G. 2002. "A Single Step Calibration Procedure for IMU/GPS in Aerial Photogrammetry". In proceedings of *Photogrammetric Computer Vision*, Graz, Austria. ISPRS, Commission III.

Radovanovic, R. S. 2002. *Adjustment of Satellite-Based Ranging Observations for Precise Positioning and Deformation Monitoring*. PhD thesis, The University of Calgary, Calgary, Canada. URL: `http://www.geomatics.ucalgary.ca/links/GradTheses.html`. UCGE Report Number 20166.

Radovanovic, R. S., Fotopoulos, G., and El-Sheimy, N. 2001. "On optimizing GNSS multifrequency carrier phase combinations for precise positioning". In proceedings of *The international association of Geodesy 2001 scientific assembly*, Budapest. The International Association of Geodesy (IAG).

Richert, T. 2005. "The Impact of Future Global Navigation Satellite Systems on Precise Carrier Phase Positioning". Master's thesis, University of Calgary, Calgary, Canada.

Rogers, R. M. 2003. *Applied Mathematics in Integrated Navigation Systems*. AIAA education series. American Institude of Aeronautics and Astronautics, Inc., Reston, VA, second edition.

Schmitz, M., Wübbena, G., and Bagge, A. 2001. "Benefit of Rigorous Modeling of GPS in Combined AT/GPS/IMU-Bundle Block Adjustment". In proceedings of *OEEPE Workshop on Integrated Sensor Orientation*, Hannover. Organisation Européene d'Etudes Photogrammétriques Expérimentales/ European Organization for Experimental Photogrammetric Research (OEEPE).

Schwarz, K.-P. and Wei, M. 2000. *ENGO 623 Lecture Notes – INS/GPS Integration for Geodetic Applications*. Department of Geomatics Engineering, The University of Calgary, Calgary, Canada. Lecture Notes.

Shum, H.-Y. and Szeliski, R. 1999. "Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment". *International Journal of Computer Vision*, 36(2):101–130.

Škaloud, J. 1999. "Problems in Direct-Georeferencing by INS/DGPS in the Airborne Environment". In proceedings of *ISPRS Workshop on Direct Versus Indirect Methods of Sensor Orientation*, pages 7–15, Barcelona, Spain.

Spilker, J. 1996a. "Fundamental of Signal Tracking Theory". In Parkinson, B., Spilker, J., Axelrad, P., and Enge, P., editors, *Global Positioning System: Theory and Applications. Vol. I.*, pages 245–328. American Institute of Aeronautics and Astronautics, Inc., Washington, DC.

Spilker, J. 1996b. "Tropospheric Effects on GPS". In Parkinson, B., Spilker, J., Axelrad, P., and Enge, P., editors, *Global Positioning System: Theory and Applications. Vol. I.*, pages 469–484. American Institute of Aeronautics and Astronautics, Inc., Washington, DC.

Teunissen, P. J. G. 1993. "Least-squares estimation of the integer GPS ambiguities". In proceedings of *The international association of Geodesy 1993 general meeting assembly*, pages 65–82, Beijing. The International Association of Geodesy (IAG).

Teunissen, P. J. G. 1995. "The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation". *Journal of Geodesy*, 70(1-2):65–82.

Teunissen, P. J. G. 1998. "Success probability of integer GPS ambiguity rounding and bootstrapping". *Journal of Geodesy*, 72(10):606–612.

Teunissen, P. J. G. 1999a. "An optimality property of the integer least-squares estimator". *Journal of Geodesy*, 73(11):587–593.

Teunissen, P. J. G. 1999b. "The probability distribution of the GPS baseline for a class of integer ambiguity estimators". *Journal of Geodesy*, 73(5):275–284.

Verhagen, S. 2003. "On the approximation of the integer least-squares success rate: which lower or upper bound to use?". *Journal of Global Positioning Systems*, 2(2):117–124.

Verhagen, S. 2004. "Integer ambiguity validation: an open problem?". *GPS Solutions*, 8:36–43.

Verhagen, S. 2007. "Reliable positioning with the next generation Global Navigation Satellite Systems". In Kurnaz, S. and Ince, F., editors, *3rd International Conference on Recent Advances in Space Technologies (RAST2007)*, pages 618–623, Istanbul, Turkey. IEEE.

Vollath, U. and Doucet, K. D. 2007. "Multiple-GNSS and FDMA high precision carrier-phase based positioning". United States Patent number 7312747.

Walpole, R. E., Myers, R., Myers, S., and Ye, K. 2007. *Probability and Statistics for Engineers and Scientists*. Pearson Prentice Hall, Upper Saddle River, NJ, eight edition.

Wang, J. 2000. "An approach to GLONASS ambiguity resolution". *Journal of Geodesy*, 74(5):421–430.

Wang, J., Stewart, M., and Tsakiri, M. 2000. "A comparative study of the integer ambiguity validation procedures". *Earth Planets Space*, 52(10):813–817.

Wei, M. and Schwarz, K.-P. 1995. "Fast ambiguity resolution using an integer nonlinear programming method.". In proceedings of *ION GPS 1995*, pages 1101–1110, Palm Springs, CA. The Institute of Navigation (ION).

Xu, P. 2001. "Random simulation and GPS decorrelation". *Journal of Geodesy*, 75(7):408–423.

Xu, P., Cannon, M. E., and Lachapelle, G. 1995. "Mixed integer programming for the resolution of GPS carrier phase ambiguities". In proceedings of *IUGG XXI General Assembly*, pages 1435–1443, Boulder, CO. International Union of Geodesy and Geophysics (IUGG). Also in Technical Report Nr. 2000.2, Department of Geodesy and Geoinformatics, Universitĺat Stuttgart, Germany.

Zhang, J. 1999. *Investigations into the Estimation of Residual Tropospheric Delays in a GPS Network*. PhD thesis, University of Calgary, Calgary, Canada. URL: `http://www.geomatics.ucalgary.ca/links/GradTheses.html`. UCGE Report Number 20132.

Appendix A

Sherman-Morrison-Woodbury Formula

The Sherman-Morrison-Woodbury Formula (often just termed Woodbury Formula) allows an inverse to be updated by new information,

$$\left(\mathbf{A}+\mathbf{UDV}^T\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1} \tag{A.1}$$

A useful alternate form is,

$$\left(\mathbf{A}+\mathbf{UDV}^T\right)^{-1}\mathbf{U}=\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{D}^{-1} \tag{A.2}$$

Proof of Equation (A.1):

$$
\begin{aligned}
&\left(\mathbf{A}+\mathbf{U}\mathbf{D}\mathbf{V}^T\right)^{-1}\left(\mathbf{A}+\mathbf{U}\mathbf{D}\mathbf{V}^T\right)\\
=&\left[\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\right]\left[\mathbf{A}+\mathbf{U}\mathbf{D}\mathbf{V}^T\right]\\
=&\mathbf{A}\mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{U}\mathbf{D}\mathbf{V}^T\\
&-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{A}\\
&-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\mathbf{D}\mathbf{V}^T\\
=&\mathbf{I}+\mathbf{A}^{-1}\mathbf{U}\left[\mathbf{D}-\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}-\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\mathbf{D}\right]\mathbf{V}^T\\
=&\mathbf{I}+\mathbf{A}^{-1}\mathbf{U}\left[\mathbf{D}-\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\left(\mathbf{I}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\mathbf{D}\right)\right]\mathbf{V}^T\\
=&\mathbf{I}+\mathbf{A}^{-1}\mathbf{U}\left[\mathbf{D}-\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)\mathbf{D}\right]\mathbf{V}^T\\
=&\mathbf{I}+\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}-\mathbf{D}\right)\mathbf{V}^T\\
=&\mathbf{I}
\end{aligned}
$$

Proof of Equation (A.2)

$$
\begin{aligned}
&\left(\mathbf{A}+\mathbf{U}\mathbf{D}\mathbf{V}^T\right)^{-1}\mathbf{U}\\
=&\left[\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\right]\mathbf{U}\\
=&\mathbf{A}^{-1}\mathbf{U}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\\
=&\mathbf{A}^{-1}\mathbf{U}\left[\mathbf{I}-\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right]\\
=&\mathbf{A}^{-1}\mathbf{U}\left[\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)-\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right]\\
=&\mathbf{A}^{-1}\mathbf{U}\left[\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}-\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)\right]\\
=&\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{D}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{D}^{-1}
\end{aligned}
$$

Appendix B

Sample Implementation of a Fixed-Size Matrix Class

```
#include <ostream>
#include <iostream>
#include <vector>

// This header contains the matrix multiplication routines for
//  both fixed-sized and variable-sized (i.e., run-time) matrices.
#include "math++/linearalgebra.hpp"

#define TRACE( msg ) \
    std::cerr << __FILE__ << ':' << __FUNCTION__ << ':' << __LINE__ \
              << "\n--␣" << msg << std::endl;

namespace math
{

namespace linalg
{

// Matrix with run-time defined size
template< typename T >
class
Matrix
{
private:
    size_t m_rows, m_cols;
    std::vector<T> m_array;
```

```
public:
    Matrix( size_t m=0, size_t n=0, T x = T() )
        : m_rows(m), m_cols(n), m_array(m*n,x) {}

    // Accessors
    const T* carray() const
        { return &m_array[0]; }

    size_t rows() const
        { return m_rows; }
    size_t cols() const
        { return m_cols; }

    T operator()( size_t i, size_t j) const
        { return m_array[i*cols()+j]; }

    // Modifiers
    T* carray()
        { return &m_array[0]; }

};


namespace fixed_size
{

// Matrix with compile-time defined size
template< typename T, size_t M, size_t N >
class
Matrix
{
private:
    T m_array[M*N];

public:
    Matrix( T x = T() )
        { std::fill( m_array, m_array+M*N, x ); }

    // Accessors
    const T* carray() const
        { return m_array; }

    size_t rows() const
        { return M; }
    size_t cols() const
        { return N; }

    T operator()( size_t i, size_t j) const
        { return m_array[i*M+j]; }
```

```
    // Modifiers
    T* carray()
        { return m_array; }
};


} // namespace fixed_size

// Multiply two matrices of any type
template< typename MatrixA, typename MatrixB, typename MatrixC >
void
product( const MatrixA& A, const MatrixB& B, MatrixC& C )
{
    TRACE( "Run-time-sized matrix multiplication" );

    detail::matrix_product(
        A.carray(), A.rows(), A.cols(), A.cols(), false,
        B.carray(), B.rows(), B.cols(), B.cols(), false,
        C.carray(), C.cols() );
}

// Multiply two fixed-size matrices, storing the result in another
//  fixed-size matrix
template< typename T, size_t M, size_t N, size_t K >
void
product( const fixed_size::Matrix<T,M,K>& A,
         const fixed_size::Matrix<T,K,N>& B,
         fixed_size::Matrix<T,M,N>& C )
{
    TRACE( "Fixed-sized matrix multiplication" );

    // Call the fixed sized matrix product.  M, K, and N
    //  are known at compile time.
    detail::fixed_size::matrix_product<M,K,N>(
        A.carray(), B.carray(), C.carray() );
}


namespace io
{

// Output a matrix
template <class Elem, class Tr, typename Matrix >
inline std::basic_ostream<Elem,Tr>&
operator<<( std::basic_ostream<Elem,Tr>& str, const Matrix& A )
{
    for( size_t i=0; i!=A.rows(); ++i )
    {
```

```
        for( size_t j=0; j!=A.cols(); ++j )
            str << A(i,j) << ' ';
        str << '\n';
    }

    return str;
}


}

} } // namespace math::linalg


int
main( int argc, char* argv[] )
{
    using namespace math::linalg::io;

    // Test multiplication of two fixed-size matrices.
    math::linalg::fixed_size::Matrix<double,3,2> A(1.0);
    {
        math::linalg::fixed_size::Matrix<double,2,4> B(2.0);
        math::linalg::fixed_size::Matrix<double,3,4> C;

        math::linalg::product( A, B, C );

        std::cout << "Result:\n" << C << std::endl;
    }

    // Test multiplication of a fixed-size matrix with a matrix whose
    //  size is defined at run-time.
    {
        math::linalg::Matrix<double> B(2,4,2.0);
        math::linalg::Matrix<double> C(3,4);

        math::linalg::product( A, B, C );

        std::cout << "Result:\n" << C << std::endl;
    }

    return 0;
}
```

The output from the above program is:

```
testfixedsized.cpp:math::linalg::product:107
-- Fixed-sized matrix multiplication
Result:
```

```
4 4 4 4
4 4 4 4
4 4 4 4

testfixedsized.cpp:math::linalg::product:91
-- Run-time-sized matrix multiplication
Result:
4 4 4 4
4 4 4 4
4 4 4 4
```

This shows how the optimised `product` routine is, as desired, used for the fixed-size matrices. An even more sophisticated implementation could use traits, so that the matrix-type-specific function overloads would not be required.