

OPTIMAL IMAGE-AIDED INERTIAL NAVIGATION

NILESH S. GOPAUL

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

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE  
YORK UNIVERSITY  
TORONTO, ONTARIO

AUGUST 2018

© NILESH S. GOPAUL, 2018

## **Abstract**

The high demand for low-cost multi-sensor integrated kinematic positioning and navigation systems, for example as the core of direct-georeferencing technique in mobile mapping, is continuously driving more and more research and development activities. The effective and sufficient utilization of cameras as navigation sensors is among the most recent scientific research and high-tech industry development subjects. Cameras are relatively inexpensive, easy to interface with, and can provide very precise angular resolution. The research is motivated by the requirement of (a) calibrating off-the-shelf camera(s) prior to navigation and (b) the fusion of imaging and inertial sensors in poor global navigation satellite system (GNSS) or GNSS denied environments. The three major contributions of this dissertation are:

- The development and analysis of a camera auto-calibration and system calibration algorithm for a GNSS, IMU and stereo camera system that is based on the scale-restraint equation. The camera auto-calibration is first performed to obtain the lens distortion parameters, up-to-scale baseline length and the relative orientation between the stereo cameras. Then, the system calibration is introduced to recover the camera lever-arms, and the bore-sight angles with respect to the IMU, and the absolute scale of the camera using the GNSS-aided inertial navigation solution. The auto-calibration bundle adjustment utilizes the scale restraint equation, which is free of object coordinates. Such a method is often called structureless bundle adjustment. The number of parameters to be estimated is significantly reduced in comparison with the ones in a self-calibrating bundle adjustment based on the

collinearity equations. Therefore, the proposed method is computationally more efficient. Test results showed that the scale-restraint equation required approximately 4 times more measurements than the collinearity equations to achieve comparable calibration accuracy while using only 0.1% of the computational resources.

- The development of a loosely-coupled visual odometry aided inertial navigation algorithm. The pose changes are pairwise time-correlated, i.e. the measurement noise vector at the current epoch is only correlated with the one from the previous epoch. The fusion of the two sensors is usually performed using a Kalman filter. The standard Kalman filter runs under the assumption that the process noise vector and measurement noise vector are white, i.e. independent and normally distributed with zero means. However, this assumption does not hold when fusing visual odometry and IMU measurements. It is well-known that the solution of the standard Kalman filter becomes suboptimal if the measurements are colored or time-correlated. Time-correlated errors are usually modelled by a shaping filter. The shaping filter developed in this dissertation uses Cholesky factors as coefficients derived from the variance and covariance matrices of the measurement noise vectors. The test results with real data showed that the proposed algorithm reduced the position drifts by 20% and 8% when compared to the standard Kalman filter and the Kalman filter with the conventional shaping filter respectively. Furthermore, the method can seamlessly be blended into an existing state-of-the-art GNSS aided-IMU system.

- The development of a tightly-coupled stereo multi-frame aided inertial navigation algorithm for reducing position and orientation drifts. Usually, the image aiding based on the visual odometry uses the tracked features only from a pair of the consecutive image frames. The proposed method integrates the features tracked from multiple overlapped image frames for reducing the position and orientation drifts. Hence, the proposed method is referred as multi-frame visual odometry (MFVO). Previous multi-frame methods, which are sometimes referred as *sliding window* methods, use batch estimators that jointly estimate the vehicle's pose and feature positions. However, the size of the parameter vector can become impractically large when the number of features in view is high. Furthermore, it is difficult to integrate these methods optimally into an existing GNSS/INS integration architecture. In the proposed MFVO method, the measurement equation system is derived from Simultaneous Localization and Mapping (SLAM) measurement equation system where the landmark positions in SLAM are algebraically eliminated by time-differencing the measurements at two consecutive epochs. However, the resulting time-differenced measurements are time-correlated. Through a sequential de-correlation, the Kalman filter measurement update can be performed sequentially and optimally. The main advantages of the proposed algorithm are (a) the reduction of computational requirements when compared to SLAM and (b) a seamless integration into an existing GNSS aided-IMU system.

## **Acknowledgement**

I would like to express thanks and appreciation to my supervisor, Dr.-Ing. Jianguo Wang, for his support and guidance during my doctoral studies. Furthermore, I would like to thank my committee members, Dr. Bruno Scherzinger and Dr. Baoxin Hu, for their valuable advices and comments.

I also would like to thank my colleague Dr. Jason Kun Qian for his work on developing the YUMIS navigation system and for helping me collect the data which has been invaluable towards the research. I would like thank the EOL laboratory at York University directed by Dr. Baoxin Hu and Dr.-Ing. Jianguo Wang.

Special thanks to the Elia family for awarding me the Elia Scholarship. Their financial support has been immense. I sincerely acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), and also give my acknowledgment to Applanix Corporation for granting me permission to use the POSGNSS software.

Finally, I would like to thank my family, friends and colleagues, for their support through all the years.

# Table of contents

Abstract .....	ii
Acknowledgement.....	v
Table of contents .....	vi
List of tables.....	x
List of figures .....	xii
List of symbols.....	xv
List of operators .....	xviii
List of abbreviations and acronyms .....	xx
1.Introduction.....	1
1.1. Background .....	1
1.2. Research objectives .....	7
1.3. Dissertation outline.....	7
2.Literature review .....	9
2.1. Camera auto-calibration and system calibration .....	9
2.1.1. Auto-calibration.....	11
2.1.2. System calibration .....	15
2.2. Image aided-INS integration .....	17
3.Estimation theory and navigation sensor overview .....	26
3.1. Mathematical preliminaries.....	26
3.1.1. Direction cosine matrix .....	26
3.2. Estimation theory .....	29
3.2.1. Least squares estimation.....	29
3.2.2. Kalman filter in discrete time.....	33
3.2.3. Outlier detection via statistic tests.....	37
3.3. Coordinate frames and transformations .....	39
3.3.1. Inertial frame .....	40
3.3.2. Earth frame .....	40
3.3.3. Navigation frame.....	41
3.3.4. Computer frame.....	42
3.3.5. Body frame .....	42
3.3.6. Camera frame .....	43
3.3.7. n'-frame.....	43

3.3.8. Summary .....	44
3.4. Global Navigation Satellite System .....	44
3.4.1. GNSS position.....	45
3.4.2. Velocity and track angle.....	46
3.4.3. GNSS compass .....	48
3.4.4. Summary .....	48
3.5. Inertial Navigation.....	49
3.5.1. Navigation equations.....	49
3.5.2. INS error models .....	51
3.5.3. Sensor errors and sensor error model .....	53
3.5.4. Mechanization Equations .....	54
3.5.5. Alignment.....	55
3.5.6. GNSS aided-inertial navigation.....	56
3.6. Photogrammetry and image-based navigation .....	59
3.6.1. Mathematical photogrammetry .....	59
3.6.2. Camera error sources.....	65
3.6.3. Camera auto-calibration considerations .....	68
3.6.4. Image-based navigation.....	70
4. Structureless stereo camera auto-calibration and system calibration .....	78
4.1. Introduction .....	78
4.2. Camera calibration .....	79
4.2.1. Structureless camera auto-calibration .....	79
4.2.2. Camera system calibration .....	85
4.2.3. Parameter Initialization .....	86
4.3. Computational complexity .....	89
4.4. Pre-analysis using the simulated data.....	94
4.4.1. SRE combination test .....	97
4.4.2. Comparison of COL and SRE auto-calibration algorithms.....	98
4.5. Laboratory tests and results.....	102
4.5.1. System description .....	102
4.5.2. Determination of lens distortion parameters .....	104
4.5.3. Comparison between COL and SRE.....	107
4.6. Summary .....	110
5. Loosely-coupled visual odometry aided inertial navigation.....	113

5.1. Introduction .....	113
5.2. Design of the Kalman filter .....	114
5.2.1. Relative measurements .....	115
5.2.2. Pairwise time correlated measurements .....	118
5.2.3. Pairwise time-correlation in stereo visual odometry .....	127
5.3. Test results with the simulated data .....	131
5.4. Loosely-coupled stereo VO-aided inertial navigation.....	136
5.5. Summary .....	139
6. Tightly-coupled stereo multi-frame visual odometry aided-inertial navigation .....	141
6.1. Introduction .....	141
6.2. Stereo Multi-frame visual odometry .....	142
6.2.1. Discussions .....	150
6.3. Test results with simulated data .....	152
6.3.1. VO KF-PTC versus MFVO.....	155
6.4. Tightly-coupled stereo MFVO aided inertial navigation .....	156
6.5. Summary .....	159
7. Experiments: road tests and results.....	163
7.1. Introduction .....	163
7.2. YUMIS system and dataset information .....	163
7.2.1. Loosely-coupled GNSS aided inertial navigation .....	166
7.3. Structureless stereo camera calibration .....	170
7.3.1. Calibration interval and measurement information.....	171
7.3.2. Camera and system calibration results .....	174
7.3.3. Evaluation.....	178
7.3.4. Discussion .....	185
7.4. Loosely-coupled stereo visual odometry aided-INS .....	186
7.4.1. Stereo visual odometry solution .....	186
7.4.2. VO-aided inertial navigation solution and evaluation.....	190
7.4.3. Discussion .....	197
7.5. Tightly-coupled stereo MFVO aided-INS.....	198
7.5.1. SLAM and MFVO-aided inertial navigation solution and evaluation ..	199
7.5.2. Discussion .....	204
8. Conclusion and future work.....	207
8.1. Conclusion and contributions .....	207

8.2. Future work and recommendations .....	209
References .....	212
Appendices .....	235
Appendix A: KF-PTC state covariance update in Joseph stabilized form .....	235
Appendix B: Sequential block inversion of a lower triangle matrix .....	236

## List of tables

Table 2.1	The components of the parameter vector and the .....	13
Table 4.1	Camera auto-calibration parameters (L and R denote the left and right camera).....	83
Table 4.2	Four combinations of the scale restraint equation in four views .....	83
Table 4.3	Dimension of the parameter vector (COL vs. SRE) .....	89
Table 4.4	Matrix operations and the corresponding number of flops.....	90
Table 4.5	Flops count required for computing COL.....	92
Table 4.6	Flops count required for computing SRE .....	93
Table 4.7	SRE stereo auto-calibration parameter .....	97
Table 4.8	Square-root of the trace of the parameter variance-covariance .....	98
Table 4.9	Left camera lens distortion parameters .....	101
Table 4.10	Right camera lens distortion parameters.....	101
Table 4.11	Relative orientation of right camera w.r.t left camera (†Free parameter) .....	102
Table 4.12	Number of points and parameters .....	102
Table 4.13	Image distortion coefficients combinations .....	104
Table 4.14	The correlation coefficients of the focal length error .....	106
Table 4.15	The correlation coefficients of the principal point error .....	106
Table 4.16	Left camera lens distortion parameters .....	108
Table 4.17	Right camera lens distortion parameters.....	108
Table 4.18	Baseline and relative orientation estimates. †Free parameter.....	108
Table 4.19	Number of the points and parameters .....	109
Table 4.20	Means, standard deviations and RMS of the deviations between the corresponding 3D reference points and estimated points. ....	110
Table 4.21	Means, standard deviations and RMS of the image residuals. ....	110
Table 5.1	The dimensions of the state vectors in the KF, KF-ST and KF-PTC .....	135
Table 7.1	Crossbow IMU440CA technical specification (partial) .....	165
Table 7.2	Dataset properties.....	166
Table 7.3	IMU440CA sensor error model parameters.....	168
Table 7.4	The lens distortion parameters with the left camera .....	174
Table 7.5	The lens distortion parameters with the right camera.....	175
Table 7.6	The relative orientation of the right camera w.r.t the left camera.....	175
Table 7.7	The estimated lever-arms, scale and bore-sight angles.....	176

Table 7.8	Difference between the estimated and measured lever arm components	176
Table 7.9	Number of points, objects, iterations, flops and $\hat{\sigma}_0^2$	177
Table 7.10	COL, SRE1 and SRE2 parameter list and size	177
Table 7.11	Average number of points per frame used in each section.	182
Table 7.12	The VO translation and rotation drift rates	184
Table 7.13	The mean, standard deviation and rms of the differences between the VO and GNSS-aided inertial integrated navigation solution	190
Table 7.14	The translation and rotation drift rates from the VO-aided inertial integrated navigation	194
Table 7.15	The translation and rotation drift rates from the TC SLAM and TC MFVO algorithms	202

## List of figures

Figure 2.1	Overview of a typical IA-INS system.....	18
Figure 2.2	Matching of point features from a pair of stereo frames .....	19
Figure 3.1	The i-, e-, n-, n <sup>c</sup> -, b-, c- and n'- frames. ....	44
Figure 3.2	GNSS Pseudorange positioning.....	45
Figure 3.3	Inertial navigation mechanization [Grewal et al, 2013].....	55
Figure 3.4	Loosely coupled GNSS/IMU integration architecture.....	57
Figure 3.5	Tightly coupled GNSS/IMU integration architecture.....	57
Figure 3.6	The collinearity condition.....	61
Figure 3.7	Two view geometry .....	62
Figure 3.8	Three view geometry .....	63
Figure 3.9	The geometrical relationship between an IMU and.....	67
Figure 3.10	The corner (left) and blob (right) masks .....	71
Figure 4.1	Stereo Vision.....	80
Figure 4.2	Four view feature matching (left) and .....	82
Figure 4.3	The six possible combinations to constrain four views with two SREs ...	84
Figure 4.4	Two and three overlapping regions.....	85
Figure 4.5	Flops vs. number of stereo points ( $m$ ).....	94
Figure 4.6	The 2D trajectory, landmarks (left) .....	95
Figure 4.7	3D trajectory, landmarks in view, left image and right image at epoch 46 .....	96
Figure 4.8	Number of stereo points per frame vs. epoch .....	97
Figure 4.9	The number of the stereo points per frame (top) .....	99
Figure 4.10	The standard deviations of $\Delta f_L$ , $\Delta x_{L,0}$ , $\Delta y_{L,0}$ , $k_{L,1}$ , $k_{L,2}$ and $k_{L,3}$ .....	99
Figure 4.11	The standard deviations of $\Delta f_L$ , $\Delta x_{L,0}$ , $\Delta y_{L,0}$ , $k_{L,1}$ , $k_{L,2}$ and $k_{L,3}$ .....	100
Figure 4.12	Stereo Camera System with IMU (Left). 90x60cm Checkerboard (right) .....	103
Figure 4.13	Image measurements for the calibration.....	103
Figure 4.14	X, Y, Z and 3D ranging errors for combinations 1 to 8.....	105
Figure 5.1	Timeline between epochs $k-m$ and $k$ . ....	115
Figure 5.2	Overlap of frames $k-2$ , $k-1$ and $k$ .....	129
Figure 5.3	The top view of the vehicle trajectory with visible landmarks (left) .....	132
Figure 5.4	The 2D VO standard deviations and the numbers of the features .....	133

Figure 5.5	True position and heading RMSE (left).....	135
Figure 5.6	Loosely-coupled VO aided-inertial navigation with.....	139
Figure 6.1	$j$ as a function of overlap percentage .....	151
Figure 6.2	The top view of the vehicle trajectory with the visible landmarks (left)	152
Figure 6.3	The true position and heading RMS errors (left) .....	154
Figure 6.4	The dimensions of the state vectors with SLAM and MFVO, .....	155
Figure 6.5	The comparison between the VO/KF-PTC and MFVO approaches .....	156
Figure 6.6	Tightly coupled MFVO aided-inertial navigation .....	159
Figure 7.1	GPS and IMU, and Controller in YUMIS system [Qian, 2017].....	164
Figure 7.2	The 2D overview of the trajectory with starting point, end point and base position (left) and the velocity and attitude profiles (right).....	166
Figure 7.3	The GPS position standard deviation ( $1-\sigma$ ) (top-left). .....	167
Figure 7.4	The estimated position accuracy ( $1-\sigma$ ) and orientation accuracy ( $1-\sigma$ ) ..	169
Figure 7.5	The GPS system innovations (left) .....	169
Figure 7.6	The estimated accelerometer and gyroscope biases.....	170
Figure 7.7	The calibration interval (left) and a stereo pair with the matched points (right) .....	171
Figure 7.8	The number of the features (top), the minimum, mean and maximum ranges of the 3D features (bottom) for COL, SRE1 and SRE2. ....	173
Figure 7.9	The standard deviations of the EO parameter for the image frames .....	173
Figure 7.10	The thirty-five 200m sections with Section ID.....	179
Figure 7.11	The numbers, and mean ranges of the 3D features for the divided sections. The colors of the plot corresponds the sections in Figure 7.10. ....	180
Figure 7.12	3D position and orientation drifts of the individual sections.....	183
Figure 7.13	The visual odometry solution; the position change (top).....	188
Figure 7.14	The standard deviation ( $1-\sigma$ ) of the VO solution.....	188
Figure 7.15	The number of the features at each epoch (top), the minimum, average and maximum object feature ranges (middle), and the percentage of the shared features between the current and previous VO solution epochs (bottom). .....	189
Figure 7.16	The differences between the VO and GNSS-aided.....	190
Figure 7.17	The 2D overview of Section 27 (best) and Section 25 (worst).....	192
Figure 7.18	The drifts in 3D position and orientation from KF, KF-ST and KF-PT for the individual color coded sections.....	193
Figure 7.19	Section 27 position change system innovation and $1-\sigma$ error bounds.....	195
Figure 7.20	Section 27 orientation change system innovation and $1-\sigma$ error bounds	196

Figure 7.21	The system innovations of the position changes and 1- $\sigma$ error bounds ..	196
Figure 7.22	The system innovations of the orientation changes and 1- $\sigma$ error bounds .....	197
Figure 7.23	The drifts in 3D position and orientation from .....	200
Figure 7.24	The size of the state vector (top). The number of features per frame and the average number of features per frame during the respective sections (middle).....	203
Figure 7.25	The histograms of the standardized system innovation vectors.....	204

## List of symbols

By convention vectors are represented by boldface lowercase characters and matrices by boldface uppercase characters.

$\theta$	pitch angle
$\boldsymbol{\theta}_x^y$	vector of Euler angles from $x$ -frame to $y$ -frame
$\lambda$	geodetic longitude
$\phi$	roll angle
$\boldsymbol{\varphi}$	rotation vector or rotation vector error of a DCM
$\varphi$	geodetic latitude
$\psi$	heading or yaw angle
$\boldsymbol{\Psi}$	rotation vector error of a DCM
$\omega_e$	magnitude of the rotation rate of the Earth ( $7.2921158 \times 10^{-5} \text{ rad s}^{-1}$ )
$\boldsymbol{\omega}_{xy}^z$	angular rate vector of the $y$ -frame with respect to the $x$ -frame projected to the $z$ -frame
$\boldsymbol{\Omega}_{xy}^z$	skew symmetric matrix of vector $\boldsymbol{\omega}_{xy}^z$
$\boldsymbol{\Phi}$	transition matrix
$\sigma$	standard deviation
$\sigma_0^2$	variance factor
$\boldsymbol{\Psi}$	coefficient matrix of the process noise
$\mathbf{0}$	zero vector
$\mathbf{O}$	zero matrix

$\mathbf{a}^y$	acceleration vector in the $y$ -frame
$\mathbf{b}$	baseline vector
$\mathbf{b}_a$	accelerometer bias vector
$\mathbf{b}_g$	gyroscope bias vector
$\mathbf{C}_i$	DCM rotation matrix about the $i$ -axis
$\mathbf{C}_x^y$	3D transformation matrix from $x$ -frame to $y$ -frame
$\mathbf{d}$	system innovation vector
$e$	first eccentricity of a reference ellipsoid
$f$	focal length
$f(\cdot), \mathbf{f}(\cdot)$	system function
$h$	ellipsoidal height
$h(\cdot), \mathbf{h}(\cdot)$	measurement function
$\mathbf{H}, \mathbf{H}_x$	Jacobian matrix a model at the parameter/states vector
$\mathbf{H}_z$	Jacobian matrix a model at the measurement vector
$\mathbf{I}$	identity matrix
$\mathbf{K}$	Kalman gain matrix
$\mathbf{la}^y$	lever-arm vector in the $y$ -frame
$\mathbf{l}^y$	position vector of 3D features in the $y$ -frame
$\mathbf{N}$	matrix of normal equations
$\mathbf{m}^y$	SLAM landmark position vector in the $y$ -frame
$\mathbf{P}$	covariance matrix of the state vector or parameter vector

<b>Q</b>	covariance matrix of the process noise vector
<b>r</b>	geographic position vector (latitude, longitude, height)
<b>R</b>	covariance matrix of the measurement vector
$R_E$	curvature radius of meridian
$R_N$	curvature radius of prime vertical
$s, \mathbf{s}$	scale factor, scale factor vector
<b>S</b>	covariance matrix of the system innovation vector
$t_k$	time $t$ at epoch $k$
	<b>w</b> process noise vector
<b>W</b>	weight matrix
<b>x</b>	state vector or parameter vector
$x, y, z$	Cartesian coordinates in the image space in the $y$ -frame
$\mathbf{x}^y$	position vector in the image space in the $y$ -frame
$X, Y, Z$	Cartesian coordinates in the object space in the $y$ -frame
$\mathbf{X}^y$	position vector in the object space in the $y$ -frame
<b>v</b>	measurement noise vector
$\mathbf{v}^y$	velocity vector in the $y$ -frame
<b>z</b>	measurement vector

## List of operators

$\cdot_i$	the $i^{\text{th}}$ element of a vector
$\cdot_{ij}, \cdot_{i,j}$	the $i^{\text{th}}$ row and the $j^{\text{th}}$ column element of a matrix
$\cdot_k$	epoch $k$ (as the subscript)
$\cdot_x$	$x$ - component of a 3D vector (as the subscript)
$\cdot_y$	$y$ - component of a 3D vector (as the subscript)
$\cdot_z$	$z$ - component of a 3D vector (as the subscript)
$\cdot(t)$	a function of time
$\delta \cdot$	perturbation of
$\hat{\cdot}$	an estimated or adjusted value
$\bar{\cdot}$	a computed value
$\cdot^{(0)}$	an approximate value (as the superscript)
$\cdot^-$	a time update (prediction) in a Kalman filter (as the superscript)
$\cdot^+$	a measurement update in Kalman filter (as the superscript)
$\bullet$	the dot product operator
$\times$	the cross-product operator or multiplication operator
$\cdot \times$	the skew-symmetric form of a vector
$\cdot^T$	transpose of a vector or a matrix
$\cdot^{-T}$	transpose of a matrix inverse or vice versa
$\Delta \cdot$	difference, change
$\Sigma \cdot$	summation operator

$ \cdot $	the absolute value
$\ \cdot\ $	the Euclidean norm
$==$	Boolean equal operator
$\approx$	is approximately equal to
$[a, b]$	the interval between $a$ and $b$
$c(\cdot)$	the cosine operator
$ceil(\cdot)$	the ceiling function
$chol(\cdot)$	the lower triangle Cholesky operator
$diag(\cdot)$	the diagonal matrix in the form of a vector or block matrices
$E[\cdot]$	the expectation of a variable or a vector
$exp(\cdot)$	an exponential function
$s(\cdot)$	the sine operator
$sqrt(\cdot)$	the square root operator
$tr(\cdot)$	the trace of a matrix operator

## List of abbreviations and acronyms

2-D	two dimensional
3-D	three dimensional
a.k.a.	also known as
BA	bundle adjustment
CCD	charge-coupled device
CMOS	complementary metal–oxide–semiconductor
COL	collinearity equations
DCM	direction cosine matrix
ECEF	Earth-Centered Earth Fixed
EKF	extended Kalman filter
EO	exterior orientation
d.o.f.	degrees of freedom
GM	Gauss-Markov
GMST	Greenwich Mean Sidereal Time
GNSS	Global Navigation Satellite System
IBN	image-based navigation
IMU	inertial measurement unit
INS	inertial navigation system
IO	interior orientation
KF	Kalman filter
LiDAR	light detection and ranging
LS	least-squares

n/a	not available
max	maximum
MEMS	Micro-Electro-Mechanical Systems
MFVO	multi-frame visual odometry
min	minimum
px	pixels
RANSAC	random sample consensus
rms	root mean square
RTK	real-time kinematic
SLAM	simultaneous localization and mapping
SRE	scale-restraint equation
VCV	variance covariance
WGS84	World Geodetic System 1984
w.r.t.	with respect to
YUMIS	York University Multisensor Integrated System

# **1. Introduction**

This chapter provides an introduction to this dissertation. Section 1.1 contains a background which is followed by the research objectives in Section 1.2. And finally Section 1.3 outlines the dissertation.

## **1.1. Background**

The high demand for direct-georeferencing technology with low-cost multisensor integrated kinematic positioning and navigation systems in mobile mapping and direct georeferencing is continuously driving more research and development activities. Mobile mapping involves the collection of data to produce maps while in continuous motion. GNSS aided-inertial navigation systems are widely used for making maps efficiently on mobile platforms through direct georeferencing. The direct georeferencing method uses the position and orientation information to geo-code each pixel or point collected by a camera or LiDAR system, respectively, without the use of extensive ground control points. The position accuracy of the georeferenced pixels or points depends on the accuracy of navigation solution.

GNSS provides long term high accuracy of absolute position and velocity solution but does not work in indoor or urban-canyon environments. INS, on the other hand, works in all environments but its solution accuracy deteriorates with time. An integrated GNSS-INS system can take their advantages to determine the trajectory of a moving platform in position, velocity and attitude and so on. During any GNSS outage the accuracy of the navigation solution depends solely on the quality of inertial navigation sensors. Navigation and tactical grade of inertial measurement units (IMUs) exhibit low solution

drift rate but are very expensive and not easily accessible to the public in civilian applications. Hence, more and more low-cost IMUs have been made available during the past decade. Low grade of MEMS IMUs are considerably cheaper and easily available but accumulate large errors over a relatively short period of time. To reduce the INS errors in poor GNSS or GNSS denied environments, other sensors can be added to the navigation system, for example Wi-Fi positioning, barometer, odometer and magnetometer to name a few [Aggarwal et al, 2010]. In the past decade, image-aiding has become a hot topic in multisensor integrated navigation. Cameras are inherently high-bandwidth and therefore have the high potential for very precise angular resolution and are readily available and easy to interface with [Miller et al, 2011]. Furthermore, it is inexpensive in comparison with other self-contained electro-optical sensors such as laser ranging (LIDAR) [Shen and Liu, 2005].

In order to use cameras as navigation sensors, they have to be calibrated first, which refers to the determination of the focal length, the principal point offset and the image distortion parameters. Furthermore, the determination of the translational offsets and the orientation (boresight) angles between the individual sensors in a multi-sensor system are also part of the calibration. Usually, the traditional camera calibration consists of capturing images containing an array of the reference targets with their coordinates accurately known in a laboratory [Wolf and Dewitt, 2000]. However, such methods require the setup of enough reference points and the calibrated parameters can become invalid during field operations, e.g., due to camera assembly/disassembly, replacement and/or bumps [Teller et al, 2010].

Alternative to the traditional calibration techniques are the auto-calibration (or self-calibration) methods. An auto-calibration refers to the determination of the camera parameters from a sequence of the overlapped images without setting up ground control points (GCPs) or specific calibration targets. The main advantages are: (a) the procedure can be fully automated, (b) the calibration can be performed in-field, and (c) the accuracy of the estimated calibration parameters can be improved by the applied information from other sensors in the system. Typically, an auto-calibration is performed in a bundle adjustment based on the extended collinearity equations where the interior orientation parameters (IOPs), the exterior orientation parameters (EOPs) and the object coordinates (a.k.a. landmark coordinates) are estimated. However, they are computationally expensive due to the very large number of object position parameters.

Another way of performing the auto-calibration is through structureless bundle adjustment methods [Faig, 1975; Cefalu et al, 2016]. These methods are based on the epipolar and scale consistency constraints and are free from object coordinates. The number of unknown parameters in the bundle adjustment is drastically reduced. Hence, this PhD research developed a camera calibration method that could precisely calibrate camera parameters in a GNSS/IMU/Stereo cameras integrated system exclusive of the object coordinate parameters. It applies the three-view scale-restraint equation [Bethel, 2003; Ghosh, 2005], with which the measurements are processed exclusively in the image space. Therefore, it does not allocate large memory and computing resources.

Once the cameras are calibrated, they are ready for navigation. There are two main analytical approaches to extract navigation information from image measurements, namely visual Simultaneous Localization and Mapping (SLAM) and visual odometry

(VO) [Murcott et al, 2011]. The former simultaneously determines the motion and the map while the latter only focuses on the motion of consecutive frames. The advantage of SLAM is that the navigation accuracy can be increased by detecting and applying loop-closures in scenarios when the same locations are visited more than once [Liu and Zhang, 2012]. However, this benefit does not hold when the same locations are not revisited and absolute positioning information, such as GNSS or position fixes, are available. The main drawback with the SLAM is the increase of the computational requirement as more and more landmarks are added to the map. The VO, on the other hand, can maintain a constant dimensionality in system parameterization since the output is only the pose changes of a moving object and does not take up a large memory.

The fusion of imaging and inertial sensors can be performed using batch processing methods [Indelman et al, 2013b; Leutenegger et al, 2015; Forster et al, 2017] or using a Kalman filter [Veth et al, 2006; Mourikis et al, 2009; Sazdovski et al, 2011; Bloesch et al, 2015; Liu et al, 2016]. The Kalman filter is generally preferred since the data processing is conducted sequentially epoch after epoch. Furthermore, GNSS-aided inertial navigation via a Kalman filter is well-known so that it is natural to employ Kalman filtering for image-aiding in the GNSS/IMU/stereo cameras integrated kinematic positioning and navigation. The SLAM-based image-aiding method [Sazdovski et al, 2011] uses the standard form of the extended Kalman filter and does not require any modification to the current GNSS-aided INS architectures. The VO-based image-aiding methods, however, require special attention to the following two specifics with the pose changes: (a) they are relative in nature and (b) *pairwise* correlated in terms of time. The relative measurements can be processed using the *stochastic cloning* Kalman filter

[Roumeliotis et al, 2002], which relates the positions and attitudes of a moving vehicle between two consecutive image frames. A shaping filter, which is a differential or difference equation with white noise input and output of a certain correlation function [Grewal, 2001], is usually used to model time-correlated measurements. The state vector is then augmented with the state vector components of the shaping filter and the resulting system model is in the form of a linear dynamic system driven by white noise [Grewal, 2001]. However, the conventional shaping filter for time-correlated measurements in [Bryson and Henrikson, 1968; Gelb, 1974] does not adequately model the *pairwise* time-correlated case and is therefore suboptimal with its application in the VO aided inertial navigation.

Similar to any dead-reckoning navigation technique, the incremental VO estimates accumulate errors and drifts over time. The drifts in the VO estimates can be reduced by utilizing image measurements from more than two consecutive frames; specifically, the last  $m$  frames ( $m > 2$ ). This approach has been employed in [Mourikis and Roumeliotis 2007; Fraundorfer et al, 2010; Clement et al, 2015; Wen et al, 2016], where the pose and feature positions are jointly estimated at the local level. However, the number of parameters in these methods increases as more features are observed and could become impractical when the dimension of the parameter vector is too high. Furthermore, the optimal integration of multi-frame image measurements in the current state-of-the-art GNSS aided inertial navigation is not so obvious.

This dissertation is focused on developments of theoretical and practical techniques for image and IMU integration. Two image-IMU integration algorithms were developed: (a) loosely-coupled visual odometry aided inertial navigation (LC VO aided-INS) and (b)

tightly-coupled multi-frame visual odometry aided inertial navigation (TC MFVO aided-INS).

The LC VO aided-INS employs a Kalman filter algorithm that models the *pairwise* time correlated VO measurements. The shaping filter for this type correlation uses Cholesky factors as the coefficients derived from the variance and covariance matrices of the measurement noise vectors. The state vector is then augmented with the de-correlated measurement noise vector which results in the form of the standard Kalman filter. The test results showed that the proposed algorithm performs better than the existing Kalman filter algorithms and provides more realistic covariance estimates.

The TC MFVO aided-INS algorithm integrates image data from multiple stereo frames without involving feature positions in the state vector. The measurement equation system in this MFVO is derived from the SLAM measurement equation system where the landmark position parameters are algebraically eliminated by time-differencing the measurements at two consecutive epochs. However, the resulting time-differenced measurements are time-correlated. Through a sequential de-correlation algorithm, the Kalman filter measurement update can be performed sequentially and optimally. The proposed MFVO algorithm uses far less computation resources while producing identical navigation solution in comparison with the SLAM method.

The derived system and measurement equations for both the LC VO aided-INS and the TC MFVO aided-INS algorithms are in the form of the standard Kalman filter. Therefore, they can be easily integrated into the current state-of-the-art GNSS aided-INS architectures.

## **1.2. Research objectives**

The objectives of the dissertation are to:

- Design, implement and evaluate a structureless camera auto-calibration and system calibration for a GNSS/IMU/Stereo camera integrated system based on the scale-restraint equation. Compare (a) the accuracy of the estimated calibration parameters and (b) the computational complexity of the proposed method with the auto-calibration algorithm based on the collinearity equations.
- Develop Kalman filter algorithm for processing pairwise time-correlated measurements. Then, implement the algorithm in a loosely coupled stereo VO aided-INS. Finally, evaluate and compare the accuracy of the proposed algorithm with the standard Kalman filter and the Kalman filter with the conventional time-correlated measurements.
- Develop and implement an optimal technique for fusing the multi-frame visual odometry and IMU measurements. Then, evaluate and compare the performance of proposed method to visual SLAM aided-INS.

## **1.3. Dissertation outline**

The remainder of the dissertation is structured as follows: Chapter 2 gives a literature review of multisensor fusion and integration navigation, camera calibration and image-aided inertial integrated navigation while Chapter 3 summarizes the fundamentals of estimation theory, GNSS, inertial navigation and image-based navigation. Right after, the structureless camera auto-calibration and system calibration algorithms for a GNSS/IMU/Stereo camera integrated are developed in Chapter 4. Chapter 5 presents the

loosely-coupled visual odometry aided inertial navigation. Then, the tightly-coupled Kalman filter is presented for fusing multi-frame visual odometry and INS tightly coupled stereo multi-frame aided inertial navigation algorithm in Chapter 6. Chapter 7 further gives the test results and conducts performance analysis of the proposed algorithms developed in Chapters 4, 5 and 6 using data collected from YUMIS system by the EOL lab of York University. At the end, Chapter 8 summarizes the dissertation with the conclusions of the research and recommendations for future work.

## **2. Literature review**

This chapter contains a literature review. Section 2.1 focuses on camera calibration while Section 2.2 focuses on reviewing image-aided inertial integrated navigation.

### **2.1. Camera auto-calibration and system calibration**

Image-based navigation (IBN) algorithms assume that the camera system is geometrically calibrated prior to its use and the calibration parameters do not change over time. There are many definitions of camera calibration in the literature. In general, regardless of the various definitions, a camera is considered as calibrated if its focal length, principal point offset and image distortion parameters are known [Remondino and Fraser, 2006]. The determination process of these parameters is referred to as camera calibration. In photogrammetry, the mathematical model for camera calibration involves the extension of the collinearity equations through additional parameters that model the distortions. The distortion model generally requires five or more point correspondences from multiple overlapping images and is fit through a least-squares bundle adjustment [Remondino and Fraser, 2006].

In multisensor integrated system consisting of a stereo camera system, the translational offsets and the orientation angles between the individual sensors are unknown after assembly. In a GNSS, IMU and stereo cameras integrated system, these geometric parameters are the 3D baseline vector and the relative orientations between two cameras [Prokos et al, 2012], the lever-arms and bore-sight angles [Bender et al, 2013] of the reference camera with respect to a specific reference point of the system. The determination of these unknowns is referred as system calibration.

There are different camera and system calibration techniques that solve some or all the parameters and can be categorized as follows:

**Laboratory calibration:** determines the focal length and principal point offset using goniometers, compactors, collimators or other optical alignment instruments in a laboratory setting [Clarke and Fryer, 1998, Wolf and Dewitt, 2000, Ghosh, 2005]. This type of the methods is usually employed in high accuracy metric cameras and almost never in low- cost off-the-shelf cameras.

**Traditional calibration:** consists of capturing images containing an array of the 3D reference targets, whose coordinates are accurately known i.e., pre-surveyed [Wolf and Dewitt, 2000]. The reference targets can be in two or three planes orthogonal to each other [Zhang, 2004] or in a calibration cage [Moe et al, 2010]. These methods provide a very accurate calibration results but are expensive to setup and maintain. An easier setup is to employ planar grid, such as checkerboard [Zhang, 2000]. It is reasonably accurate, simple to produce and more practical to use. The parameters are usually estimated through a bundle adjustment or the Levenberg-Marquardt algorithm [Remondino and Fraser, 2006]. However, these parameters are calibrated in such an environment that may not necessarily be the same as the real working environment.

**Auto (or self) -calibration:** performs the calibration by using a sequence of the overlapping images without the use of any reference target and does not require an elaborate setup. The calibration can be performed in any environment with texture and close-range objects. The methods in this category are therefore more flexible and practical than the *traditional* methods. However, they may not be able to achieve the same accuracy level as *traditional* methods. Besides, the absolute scale of the camera

system cannot be known without additional information. Similar to the Traditional calibration methods, the calibration parameters are estimated through a bundle adjustment (BA) or the Levenberg-Marquardt algorithm.

**System calibration:** involves the determination of lever-arms [Bender et al, 2013], boresight angles [Mostafa, 2001] and the absolute scale of the camera system [Kelly et al, 2011] in a multisensor integrated system. These parameters can only be obtained via external information. For example, lever-arms can be measured using survey equipment, boresight angles can be recovered with GCPs and the absolute scale can be estimated using GNSS measurements as the reference.

**In-field (or in-flight) calibration:** can be considered as a combination of the *traditional*, auto and system calibration methods. This is performed when system calibration parameters are not available or become invalidated during in-field operations, e.g., due to camera assembly/disassembly, replacement, or bumps [Teller et al, 2010].

### 2.1.1. Auto-calibration

The most widely used mathematical model for camera auto-calibration is the well-known extended collinearity, which consists of the collinearity equations and the image distortion model [Fraser, 2012]. The auto-calibration procedure can be categorized into block-invariant [Kenefick et al, 1972; Ghosh, 1988] and photo-variant [Moriwa, 1980] approaches. The former assumes that the distortion is constant in a set of images, while the latter assumes that the distortion changes between images. Most auto-calibration approaches involving digital cameras are block-invariant.

The calibration parameters can be determined in a bundle adjustment (BA) [Ghosh, 1988; Triggs et al, 2000] or in the SLAM framework [Civera et al, 2009; Kelly and

Sukhatme, 2009; Kelly et al, 2011; Keivan and Sibley, 2014]. This process involves the simultaneous estimation of the calibration parameters, the exterior orientation and the positions of the stationary objects. In photogrammetry, BA is the preferred method for this purpose. The parameters are usually estimated by using least-squares (LS) [Triggs et al, 2000] or the Levenberg-Marquardt (LM) algorithm [Levenberg, 1944; Hartley and Zisserman, 2003]. The size of the computed Jacobian matrix and normal equation system can be large. Solving this linearized system can be inefficient in terms of the memory and computation loading and can be impractical especially when the number of the involved exterior orientation parameters and the involved objects is large [Jeong et al, 2012].

There are several methods proposed to reduce the computation and memory load in the BA by exploiting the sparsity of the Jacobian matrix and normal equation. Lourakis and Argyros [2009] presented the Sparse Bundle Adjustment (SBA) by constructing a dense normal matrix from the non-zero Jacobian blocks, in which the Cholesky decomposition and back-substitution method were used to solve the parameters. Konolige [2010] improved the efficiency of the SBA with the Sparse Sparse Bundle Adjustment (sSBA) by employing a highly-optimized Cholesky decomposition solver. The efficient Incremental Smoothing and Mapping (iSAM) algorithm was developed by Kaess et al [2008], where the parameters were updated by a QR factorization of the naturally sparse normal matrix and by only re-computing matrix entries that actually changed. Kaess et al [2011] further improved variable reordering and re-linearization in iSAM2 by implementing a Bayes tree data structure. Although these methods have improved the computational efficiency of solving the BA problem, the number of estimated parameters can still be large. As an example, consider a set 100 stereo images viewing

10000 objects. The total number of parameters the being estimated in the BA is 30625 (see Table 2.1) and 98% of them are the position vector of the objects. The object coordinates are not particularly in need since the goal is to obtain the calibration parameters. If they can be removed or omitted from the system of equations, then the memory and computational usage for solving the BA problem can be drastically reduced.

Table 2.1 The components of the parameter vector and the corresponding size in a stereo camera auto-calibration bundle adjustment

Number of stereo images	100
Number of observed objects	10000
<b>Parameter</b>	<b>Size</b>
Focal length error, principal point error	$2 \times 3$
Image distortion (10 parameter model)	$2 \times 10$
Stereo baseline and relative orientation (one baseline component is fixed)	$2+3$
Exterior Orientation (one EO parameter is fixed)	$6 \times (100-1)$
Object position parameters	$3 \times 10000$
<i>Total parameter vector size</i>	<i>30625</i>

Several approaches have been proposed to reduce the order of the BA. Dang et al [2009] introduced a BA with the reduced order for their stereo self-calibration algorithm. The  $x$  and  $y$  components of the object positions were algebraically eliminated from the equation system, only the depth ( $z$ ) component of the objects remains in the parameter vector, which reduced the parameter dimension by almost  $2/3$ . The Schur complement trick was used in [Triggs et al., 2000, Jeong et al, 2012] where the dimension of the linear equation system was reduced such that only exterior orientation (EO) parameters were estimated. These methods are useful for cases where only the EO parameters are required by the user. However, the Jacobian matrix requires a good approximation of the object positions which can be difficult to obtain if the camera system is not calibrated.

Calibration methods that are free of object coordinates are referred as structureless BA or light BA. They typically employ two-view constraints, e.g. the coplanarity equations, or three view constraints, e.g. the trifocal tensor [Hartley, 1997], or both. Faig [1975] developed an auto-calibration method which employed the coplanarity equation. Furthermore, a control restraint condition was included to recover the absolute orientation of the images. More recently, Rodriguez et al. [2011a, 2011b] developed the Global Epipolar Adjustment (GEA) using the two-view coplanarity constraint in their bundle adjustment. Cefalu et al [2016] implemented a similar approach as in [Rodriguez et al., 2011b], which included the image distortion model into the measurement equation by Brown [1971]. It has shown that the GEA required less number of iterations than the SBA using the LM algorithm. Furthermore, the number of parameters is fixed per image pair. However, the estimated translation vectors between the images are ambiguous and do not have physical meaning since coplanarity constraint does not ensure that all images are with the same scale. Scale consistency is important in a multisensor integrated navigation system and for general applications. Three view constraints can be used to ensure the scale consistency between the views. Steffen et al. [2012] proposed a structureless relative BA which combined the epipolar and trifocal constraints between images. The relative representation of the camera positions improved the numerical condition of the equation system and is also statistically equivalent to the classical bundle adjustment. Indelman [2012] implemented the incremental light bundle adjustment (iLBA) and derived a three-view constraint system involving three equations, i.e., two epipolar equations and a third three-view constraint equation, for the scale consistency. Indelman et al. [2013a] integrated the iLBA with IMU measurements for robotic

navigation. Both Steffen et al. [2012] and Indelman [2012] used monocular vision and chained the entire image set by constraining images {1, 2, 3}, then images {2, 3, 4}, and so forth. This chaining ensures that all images operate on the same scale. However, they assumed that the cameras have been calibrated.

Auto-calibration algorithms have a rank deficiency of order seven (i.e. 3D position, 3D orientation and scale). Hence, they require the minimal constraints to define the network datum, which can be done by applying the minimum constraint free-network adjustment, or through explicit minimal control point [Remondino and Fraser, 2006]. In free-network adjustment one can fix one camera position and one orientation. Then one coordinate component of a second position or the distance between the two cameras can be fixed [Cefalu et al, 2016]. If stereo cameras are used, then the length of the stereo baseline is treated as a free parameter [Hartley and Zisserman, 2003]. In order to recover the absolute scale, certain external information is needed.

### **2.1.2. System calibration**

The boresight angles between an IMU and a camera system can be determined with or without ground control points (GCPs). In the first case, the camera orientation is first computed using GCPs. Then the IMU body-to-mapping frame direction cosine matrix (DCM) is determined at the time of exposure. Finally, the boresight angles are recovered by comparing the two sets of orientations [Škaloud et al, 1996, Mostafa, 2001]. In the second case, the boresight angles are treated as constant parameters in a bundle adjustment [Pinto, 2002, Heipke et al, 2002, Mostafa, 2002, Bender et al, 2013]. These two methods are typically applied in calibrating aerial photogrammetric survey systems with the known IO parameters.

Camera-IMU lever-arms can be measured using survey equipment or estimated in the BA. Bender et al [2013] presented an in-flight graph based the BA approach for system calibration between a rigidly mounted camera and an IMU. Image point features together with the solution of the GNSS aided-inertial navigation position and orientation were used as measurements. This method simultaneously computed the IOPs as well as the lever arms and boresight angles between the two systems. However, it also required one GCP at least in-order to recover the z-component of the lever-arm vector. Kelly and Sukhatme [2009] proposed a camera-IMU self-calibration method within the SLAM framework implemented by an unscented Kalman filter. The lever-arms and mounting angles, the IMU gyroscope and accelerometer biases, the local gravity vector and landmarks could all be recovered from camera and IMU measurements alone. However, they assumed that the internal camera parameters were known beforehand. Mirzaei and Roumeliotis [2008] presented a similar tightly-coupled approach using an iterative extended Kalman filter, but, in need of known landmark position.

Auto-calibration algorithms in free-network adjustment mode require external information to compute absolute scale of the camera system. Kelly et al [2011] focused on determining the absolute scale of both the scene and the baseline in a stereo rig using GNSS measurements. Their approach was similar to the photogrammetric BA and the structure from motion algorithms. They could recover the baseline and the relative orientation between the two cameras and the lever-arms between the GNSS antenna and the reference camera.

Accordingly, this dissertation develops a camera auto-calibration algorithm using a structureless bundle adjustment for a stereo camera system. Furthermore, system

calibration is performed using GNSS/IMU data to recover the boresight angles, lever-arms and absolute scale of the camera system. Both camera auto-calibration and system calibration parameters are estimated simultaneously using the least square method.

## **2.2. Image aided-INS integration**

In an image aided-INS (IA-INS), the performance of the inertial navigation system can be improved by fusing the measurements derived from images taken by an on-board camera system. Typically, point features are extracted and matched from consecutive overlapping image frames using image processing techniques. Then the IA-INS algorithm uses these point features as measurements to estimate the navigation states via batch processing [Indelman et al, 2013b; Leutenegger et al, 2015; Forster et al, 2017] or a Kalman filter [Veth et al, 2006; Mourikis et al, 2009; Sazdovski et al, 2011; Bloesch et al, 2015; Liu et al, 2016].

The camera system can consist of a single or stereo camera. Monocular vision can only estimate the trajectory only up to an unknown scale. Stereo vision, on the other hand, avoids the scale ambiguity inherent in monocular vision when the stereo baseline is known. Furthermore, monocular vision requires three consecutive frames in order to transfer the relative scale and this tends to reduce the stability of the system [Scaramuzza and Fraundorfer, 2011].

Since the early 1980s, IA-INS research has been conducted. Moravec [1980] introduced one of the first image-only motion estimation using stereo cameras. Merhav and Bresler [1986] developed an online image-based velocity-to-height ratio estimation algorithm and its integration with on board navigation sensors. With the availability of digital cameras in the 1990s, the typical data process of modern IA-INS consists of three

main steps (a) point feature detection (b) point feature matching and (c) pose (or pose change) estimation and navigation state update. Figure 2.1 overviews a typical modern IA-INS.

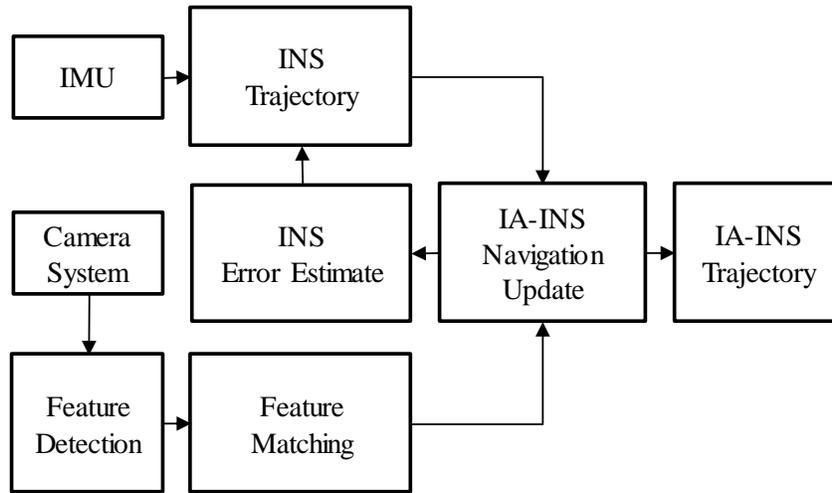


Figure 2.1 Overview of a typical IA-INS system

In the feature-detection step, stable points, such as corners and blobs, are located on the images. For navigation applications, the detector must be repeatable, i.e. it should ideally be able to find the same point features in multiple frames. Many feature detectors have been developed, for example, Harris [Harris et al., 1988], SIFT [Lowe, 1999] and SURF [Bay et al., 2008]. Once the features have been identified in each frame, they are matched across multiple frames. This is achieved by first constructing a feature descriptor using pixels around the point. The descriptor vectors on an image are then matched against descriptors from other images in order to obtain correspondences. Constrained matching techniques can be employed to reduce the number of potential matching candidates and there can reduce the search time. In the case of stereo vision, the search can be performed along the epipolar lines between the stereo pairs [Bin Rais, et al, 2003]. Between consecutive frames, the locations of the features on the next frame can be

predicted using a motion model [Scaramuzza and Fraundorfer, 2011] or using the motion estimate from the inertial navigation solution [Veth et al, 2006]. This can effectively reduce the search radius, increase the efficiency and help to prevent aliasing. Figure 2.2 shows point features extracted and matched from a stereo pair.

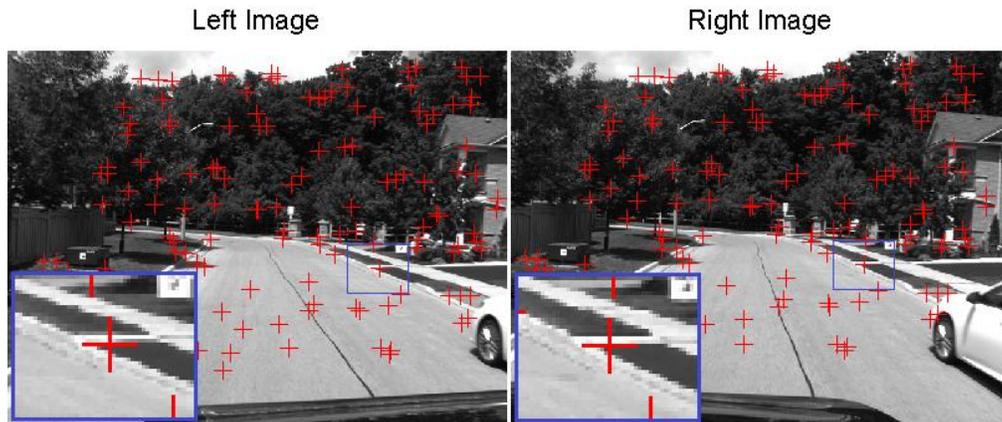


Figure 2.2 Matching of point features from a pair of stereo frames

The pose estimation and navigation state update are typically based on visual SLAM [Davison, 2003; Konolige, K and Agrawal, 2008; Alcantarilla et al, 2012] or visual odometry (VO) [Nister et al, 2004; Konolige, et al 2007; Gopaul et al, 2017]. The former applies the well-established SLAM algorithms, while the latter tracks common features from the consecutive image frames [Murcott et al, 2011].

The SLAM technique incrementally builds a consistent map of landmarks in an unknown environment whilst the simultaneous determination of the location of the mobile system is being conducted [Dissanayake et al, 2006; Durrant-Whyte and T. Bailey, 2006a; Thrun et al. 2008]. The state vector consists of the navigation states (position, velocity and/or orientation) and the landmark positions. The SLAM algorithm requires map maintenance where newly visible landmarks observed from the

environment are added to the map and landmarks that are no longer visible or to be revisited are removed from the map.

Visual SLAM tends to be more accurate than VO, since the map retains the memory of the measurements over multiple frames while VO only employs measurements from two latest consecutive image frames [Scaramuzza and Fraundorfer, 2011]. Furthermore, the accuracy of visual SLAM can be increased by detecting and applying any loop-closure in scenarios when any location is visited more than once [Liu and Zhang, 2012]. However, the application of loop-closures may be irrelevant in cases where past locations are not revisited and when absolute position measurements, such as GNSS position or position fixes, become available. Furthermore, the main drawback with the SLAM is the computational load increases in the order of  $O(n^2)$ , wherein  $n$  is the number of landmarks in the map.

The computation complexity can be reduced though approximate and suboptimal methods. Guivant and Nebot [2001] introduced a suboptimal EKF-SLAM method where only a subset of the landmarks' variance-covariance (VCV) matrix is considered during the measurement update. The VCV estimate then becomes more conservative. Julier [2001] used the Schmidt-Kalman filter, in which only the camera pose and a limited subset of the landmarks are updated. The computational costs become linear in relation to the number of the landmarks in the state vector. Even though suboptimal methods trade optimality for computation and memory usage, they can also degrade or even cause the KF estimates to diverge. Since VO concerns only in determining the trajectory and does not have to deal with landmark positions, it is computationally more efficient than visual SLAM and can work in constant state vector size [Williams and Reid, 2010].

The image-aided inertial integrated navigation can be achieved using a batch processing (e.g. bundle adjustment [Indelman et al, 2013b] and non-linear least-squares [Leutenegger et al, 2015] and graph-based optimization [Forster et al, 2017]) or a Kalman filter [Veth et al, 2006; Mourikis et al, 2009; Sazdovski et al, 2011; Bloesch et al, 2015; Liu et al, 2016]. Batch processing methods process all measurements simultaneously to compute all the parameters. However, these methods can be impractical when the dimension of the parameter vector becomes large. The Kalman filter, on the other hand, is a recursive process, more computationally efficient and more practical. Furthermore, it is preferred since the state-of-the-art GNSS-aided inertial integrated navigation typically employs a Kalman filter.

Similar to multisensor integrated navigation systems in general, the integration schemes of vision-aided inertial navigation can be divided into *loosely-* and *tightly-coupled* approaches [Corke et al, 2007]. A loosely-coupled system consists of two parallel estimation processes that run at different rates and exchange information. The first filter processes the image measurements to obtain the pose or pose change. Then a second filter performs the visual-inertial integration using the output of the first one as measurements. The tightly-coupled approach directly combines image measurements (2D or 3D) and inertial measurements in a single and optimal filter. A loosely-coupled system integrates two well-known subsystems and tends to be computationally more efficient [Leutenegger et al, 2015]. However, the estimation of the camera biases is almost impossible [Li and Mourikis, 2013]. The state vector in tightly coupled systems can include camera biases. However, this requires extensive filter model tuning and increases the computational loading [Corke et al, 2007].

In a VO aided INS [Roumeliotis et al, 2002; Tardif et al, 2010, Sirtkaya et al, 2013], the VO measurement are *relative* in nature, that is, they are the differences of the positions and attitudes of a moving vehicle between two image frames, from which the VO estimates were derived [Roumeliotis et al, 2002]. The measurement equation is a function of the state vector at the current epoch  $k$  and the previous epoch  $k-1$  (or some epoch in the past). Equation (2.1) illustrates this model:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{x}_{k-1}) + \mathbf{v}_k \quad (2.1)$$

where  $\mathbf{z}_k$  is the measurement vector,  $\mathbf{x}_k$  and  $\mathbf{x}_{k-1}$  are the state vectors,  $\mathbf{v}_k$  is the measurement noise vector, and  $\mathbf{h}(\cdot)$  is the nonlinear measurement model. The current systems augment the state equations to accommodate the relative measurements [Roumeliotis et al, 2002, Konolige et al, 2007, Tardif et al, 2010]. The augmented state vector contains two copies of the original one. The first copy  $\mathbf{x}_k$  evolves with time, while the second copy  $\mathbf{x}_{k-1}$  remains stationary. They are then related to each other through the measurement model in (2.1). This approach increases the accuracy of the estimated states and improves the robustness of the system [Roumeliotis et al, 2002].

Another issue with VO is that two consecutive VO estimates are time correlated. The position and attitude change at epoch  $k$  is derived from the tracked features at epochs  $k$  and  $k-1$ . Some of the common features at epoch  $k-1$  are also used to derive the relative position and attitude change between epochs  $k-1$  and  $k-2$ . Since there are no common feature points are shared at epochs  $k$  and  $k-2$ , only two consecutive VO measurements (i.e. at epochs  $k$  and  $k-1$ ) are correlated. Hence, they are *pairwise* time-correlated, which was first coined by Bierman [2006]. The Kalman filter in the standard form

assumes that the process noise vector and the measurement noise vector are white and conform to normal distributions with their expectations of zero. However, this assumption is not satisfied with the VO measurements. If the aiding is performed with the standard Kalman filter and the measurement noise vector is colored or time-correlated, the solution of the states will become suboptimal. Neglecting significant time-correlated errors can degrade the performance of the filter. In this case, the Kalman filter is usually augmented with a shaping filter that handles the time-correlated measurements [Bryson and Henrikson, 1968; Gelb, 1974]. However, the commonly used shaping filter as in [Bryson and Henrikson, 1968; Gelb, 1974] not only assumes that the measurement noise are correlated with the ones from the previous epoch, but also with the ones before them, i.e., from epochs  $k-2$ ,  $k-3$  and so on. Hence, it cannot appropriately model the *pairwise* time-correlated measurements. If it is employed in VO-aided inertial integrated navigation, the Kalman filter solution will not produce optimal results. Bierman [2006] introduced a sequential method for whitening pairwise time-correlated measurements for a time series, which uses the Cholesky factors derived from the measurement variances and covariances. The algorithm is efficient, since it is recursive and does not require all the measurements simultaneously available for computation. Mourikis et al [2007] developed the Stochastic Cloning-Kalman filtering equations to deal with *pairwise* correlated measurements, which involved augmenting the state vector with the feature observations and then estimated camera pose in the following epoch. Although the position and orientation estimates were optimal, the size of the state vector and variance-covariance matrix increases as more observations are made available. This obviously requires more computation and memory resources.

This dissertation proposes a novel method for processing pairwise time-correlated measurements in a Kalman filter. The corresponding shaping filter uses Cholesky factors as coefficients that are derived from the measurement noise variance and covariance matrices. The state vector is augmented with the de-correlated measurement noise vector which results in the form of the standard Kalman filter. The advantages of the proposed algorithm can be summarized as follows: (a) the shaping filter models the VO measurement noise characteristics correctly (b) the Kalman filter can provide a more realistic covariance estimates (c) the size of the state vector is constant and (d) can be easily integrated into an existing GNSS aided-INS architecture.

The drift of VO pose estimates can be reduced by utilizing image measurements from more than two consecutive frames, specifically, the last  $m$  frames ( $m > 2$ ). These approaches, which are often referred as *sliding window filter* or *windowed bundle adjustment*, employ batch processing estimators and jointly estimates the vehicle's pose and feature positions at the local level [Fraundorfer et al, 2010; Clement et al, 2015; Wen et al, 2016]. They have also been implemented in integrated visual-INS systems [Leutenegger et al, 2015; Qin et al, 2017]. Leutenegger et al [2015] employed a non-linear least-squares estimator where the cost function combined the weighted reprojection errors for visual landmarks and inertial error terms for a stereo system. Qin et al [2017] proposed a non-linear optimization-based estimator for a monocular-IMU system using pre-integrated IMU factors. Moreover, they included a procedure for relocalization and loop closure. The number of parameters in these *sliding window* methods increases as more features are observed and can become impractical when the dimension of the parameter vector becomes too high. Furthermore, it is difficult to integrate these methods

optimally with the current GNSS/INS integration architecture since they do not employ a Kalman filter. Mourikis and Roumeliotis [2007] developed the Multi-state Constraint Kalman Filter (MSCKF), in which the state vector contained vehicle poses in a variable window and did not contain feature positions as the feature positions were handled separately in a batch processor. This method computes 3D feature position in the global frame, which can be problematic if the position errors are too large and they require all measurements related to a feature from the last  $m$  poses to have been observed before the update is performed.

This dissertation proposes MFVO aided inertial navigation algorithm that integrates image data from multiple frames without involving feature or landmark positions in the state vector. This is accomplished by algebraically eliminating the landmark position vector in the SLAM measurement equation at two consecutive epochs. However, the derived measurements are time-correlated. Through a sequential de-correlation algorithm, the system and measurement equations take form of the standard Kalman filter. Thus the Kalman filter measurement update can be performed sequentially and optimally. The advantages of the MFVO algorithm aided inertial navigation can be summarized as follows: (a) the integration of image measurements from multiple frames without involving landmark positions (b) the reduction of computational requirements when compared to SLAM and (c) a seamless integration into an existing state-of-the-art GNSS aided-INS architecture.

### **3. Estimation theory and navigation sensor overview**

This chapter provides the fundamentals and background that will be required for the development in Chapters 4, 5 and 6. The relevant mathematical techniques are reviewed in Section 3.1. Section 3.2 overviews least-squares and Kalman filter techniques. Section 3.3 defines the coordinate systems and the transformation between coordinate systems used in the research. Sections 3.4, 3.5 and 3.6 present the fundamental concepts of GNSS, INS and image-based navigation (IBN) respectively.

#### **3.1. Mathematical preliminaries**

The section reviews the mathematical techniques and operations that are used in this dissertation. The following topics are presented: direction cosines matrices and mathematics of the rotation vector.

##### **3.1.1. Direction cosine matrix**

Direction cosine matrix (DCM) is an essential element in multisensor integrated navigation, which transforms a 3D vector from one coordinate frame to another and is also called the transformation matrix.

###### *3.1.1.1. DCM and Euler angles*

The DCM can be expressed by three successive rotations about any three axes and the angles of rotation are called Euler angles. There is no unique sequence to apply the successive rotations. The most commonly used sequence rotations, called the Tait-Bryan angles, performs the rotations about three distinct axes. For example, the Tait-Bryan angles between the navigation frame (see Section 3.3.3) and the body frame (see Section 3.3.5) are the roll ( $\phi$ ), pitch ( $\theta$ ) and heading ( $\psi$ ) angles. The first rotation is performed

about the  $z$ -axis by the heading angle, the second rotation about the  $y$ -axis by the pitch angle and the final rotation is performed about the  $x$ -axis by the roll angle [Bekir, 2007; Rogers, 2007]. The sequential transformation from the navigation frame to the body can be expressed as

$$\begin{aligned}\mathbf{x}^b &= \mathbf{C}_x(\phi)\mathbf{C}_y(\theta)\mathbf{C}_z(\psi)\mathbf{x}^n \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{pmatrix} \begin{pmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{pmatrix} \begin{pmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}^n\end{aligned}\quad (3.1)$$

with  $ca = \cos a$  and  $sa = \sin a$  for  $a = \phi, \theta$  or  $\psi$ . Combining the sequential rotations in Equation (3.1), the DCM (transposed) becomes as

$$\mathbf{C}_b^n(\boldsymbol{\theta}_b^n) = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix}\quad (3.2)$$

where  $\boldsymbol{\theta}_b^n = (\phi, \theta, \psi)^T$ . Given the DCM, the Euler angles can be recovered by

$$\begin{aligned}\phi &= \tan^{-1}\left(\frac{C_{32}}{C_{33}}\right) \\ \theta &= \sin^{-1}(-C_{31}) \\ \psi &= \tan^{-1}\left(\frac{C_{21}}{C_{11}}\right)\end{aligned}\quad (3.3)$$

### 3.1.1.2. DCM and the rotation vector

Instead of performing three sequential rotations to transform between two coordinate systems, one can use the rotation vector  $\boldsymbol{\phi}$  to perform one rotation around a single fixed axis. The DCM can be expressed in terms of the rotation vector by the Rodrigues formula [Roger, 2007, p 30]

$$\mathbf{C}_x^y = c|\boldsymbol{\phi}|\mathbf{I} + s|\boldsymbol{\phi}|(\mathbf{u}\times) + (1 - c|\boldsymbol{\phi}|)\mathbf{u}\mathbf{u}^T\quad (3.4)$$

where  $|\boldsymbol{\varphi}|$  is the rotation angle and  $\mathbf{u}$  is the rotation unit vector. If the rotation angle is small, then Equation (3.4) can be approximately written as

$$\mathbf{C}_x^y \approx \mathbf{I} + |\boldsymbol{\varphi}|(\mathbf{u} \times) = \mathbf{I} + \boldsymbol{\varphi} \times \quad (3.5)$$

Given the DCM  $\mathbf{C}_x^y$ , the magnitude of the rotation vector can be computed with [Bekir, 2007, pp45]

$$|\boldsymbol{\varphi}| = 2 \cos^{-1} \left( \frac{1}{2} \sqrt{\text{tr}(\mathbf{C}_x^y) + 1} \right)$$

(3.6)

### 3.1.1.3. DCM differential equation

The DCM differential equation is given as [Rogers, 2007, p29]

$$\dot{\mathbf{C}}_x^y = \mathbf{C}_x^y \boldsymbol{\Omega}_{yx}^x = -\boldsymbol{\Omega}_{xy}^y \mathbf{C}_x^y \quad (3.7)$$

where  $\boldsymbol{\Omega}_{yx}^x$  is the skew-symmetric form of the angular rate vector  $\boldsymbol{\omega}_{yx}^x$ . In the linearized form, (3.7) can be written as [Rogers, 2007]

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\varphi} \times \boldsymbol{\omega}_{yx}^x + \delta \boldsymbol{\omega}_{yx}^x = -\boldsymbol{\Omega}_{xy}^y \boldsymbol{\varphi} + \delta \boldsymbol{\omega}_{yx}^x \quad (3.8)$$

### 3.1.1.4. Rotation vector differential equation

Rotation vector differential equation can be written as [Salyshev, 2004, p30]

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\omega} + \frac{1}{2} \boldsymbol{\varphi} \times \boldsymbol{\omega} + \left( 1 - \frac{|\boldsymbol{\varphi}| \sin|\boldsymbol{\varphi}|}{2(1 - \cos|\boldsymbol{\varphi}|)} \right) \mathbf{u} \times (\mathbf{u} \times \boldsymbol{\omega}) \quad (3.9)$$

where  $\boldsymbol{\omega}$  is the angular rate vector between frames. For small angle  $\|\boldsymbol{\varphi}\|$ , Equation (3.9) can be rewritten as

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\omega} + \frac{1}{2} \boldsymbol{\varphi} \times \boldsymbol{\omega} + \frac{1}{12} \boldsymbol{\varphi} \times (\boldsymbol{\varphi} \times \boldsymbol{\omega}) \quad (3.10)$$

### 3.1.1.5. Quaternion

The Quaternion is a four-dimensional vector that can be used to represent rotations and attitude. Quaternions are useful in avoiding singularity problems in attitude representation when the pitch and roll angles are close to 90 degrees. However, this is not expected in the research presented in the dissertation. Therefore, it will not be used. Readers interested in this topic can refer to [Salychev, 2004; Rogers, 2007, etc.].

## 3.2. Estimation theory

This section summarizes least-squares estimation, Kalman filtering and outlier detection with the help of statistic tests. Least squares method and Kalman filter were originally developed based on linear systems and then extended to non-linear ones. Most systems are inevitably non-linear in nature. Therefore, the concepts presented here starts with non-linear and linearized counterparts.

### 3.2.1. Least squares estimation

The least squares (LS) algorithm is a method for estimating unknown parameters in a measurement model by minimizing the weighted sum of measurement residuals squared. Consider the following non-linear implicit measurement equation system:

$$\mathbf{h}(\mathbf{x}, \mathbf{z}, \mathbf{v}) = \mathbf{0}, \mathbf{v} \sim N(\mathbf{0}, \mathbf{R}) \quad (3.11)$$

wherein  $\mathbf{x}$  is the  $n \times 1$  unknown parameter vector;  $\mathbf{z}$  is the  $p \times 1$  measurement vector;  $\mathbf{v}$  is the  $p \times 1$  measurement noise vector; and  $\mathbf{h}(\cdot)$  is the  $m \times 1$  vector of the measurement functional model.  $\mathbf{v}$  is generally assumed to be normally distributed with its zero-mean and positive definite variance-covariance (VCV) matrix  $\mathbf{R}$

$$\mathbf{R} = \sigma_0^2 \mathbf{W}^{-1} \quad (3.12)$$

where  $\sigma_0^2$  is the variance of unit weight and  $\mathbf{W}$  is the measurement weight matrix. To estimate the unknown parameters using least-squares method, the measurement model needs to be usually linearized about the measurement vector  $\mathbf{z}$  and an given approximate  $\mathbf{x}^{(0)}$  of  $\mathbf{x}$ . The first order Taylor series expansion gives:

$$\begin{aligned}\mathbf{0} &= \mathbf{h}(\hat{\mathbf{x}}, \hat{\mathbf{z}}) \\ &= \mathbf{h}(\mathbf{x}^{(0)} + \delta\mathbf{x}, \mathbf{z} - \mathbf{v}) \\ &\approx \mathbf{h}(\mathbf{x}^{(0)}, \mathbf{z}) + \mathbf{H}_x \delta\mathbf{x} - \mathbf{H}_z \mathbf{v}\end{aligned}\tag{3.13}$$

where  $\delta\mathbf{x}$  is the correction vector of  $\mathbf{x}^{(0)}$ ,  $\mathbf{h}(\mathbf{x}^{(0)}, \mathbf{z})$  is the misclosure vector,  $\mathbf{H}_x$  is the  $m \times n$  Jacobian matrix of  $\mathbf{h}(\cdot)$  at  $\mathbf{x}^{(0)}$  and vector  $\mathbf{H}_z$  is the  $m \times p$  Jacobian matrix of  $\mathbf{h}(\cdot)$  at  $\mathbf{z}$ .

The most probable value of the parameter correction vector  $\delta\mathbf{x}$  is the vector  $\delta\hat{\mathbf{x}}$  that minimizes the weighted sum of squares of the measurement residuals  $\hat{\mathbf{v}}$  (i.e. the difference between the measurement vector and the model). This is achieved by computing  $\delta\hat{\mathbf{x}}$  such that the following the cost function  $J$  is minimized

$$J = \hat{\mathbf{v}}^T \mathbf{R}^{-1} \hat{\mathbf{v}}.\tag{3.14}$$

The LS solution for the parameter correction vector and its variance-covariance matrix is given by (derivations omitted here):

$$\begin{aligned}\delta\hat{\mathbf{x}} &= -\mathbf{N}^{-1} \mathbf{H}_x^T \mathbf{M}^{-1} \mathbf{h}(\mathbf{x}^{(0)}, \mathbf{z}) \\ \hat{\mathbf{P}} &= \hat{\sigma}_0^2 \mathbf{P} = \hat{\sigma}_0^2 \mathbf{N}^{-1} \\ \mathbf{M} &= \mathbf{H}_z \mathbf{W}^{-1} \mathbf{H}_z^T \\ \mathbf{N} &= \mathbf{H}_x^T \mathbf{M}^{-1} \mathbf{H}_x\end{aligned}\tag{3.15}$$

wherein

$\mathbf{N}$  is the coefficient matrix of the normal equation system,

$\hat{\mathbf{P}}$  is the variance-covariance matrix of the estimated parameters, and

$$\hat{\sigma}_0^2 = \frac{\hat{\mathbf{v}}^T \mathbf{W} \hat{\mathbf{v}}}{m-n}, (m > n) \text{ is the a-posteriori variance of unit weight.}$$

The trace of the  $\hat{\mathbf{P}}$  matrix can be interpreted as a measure of the overall accuracy of the estimated parameters [Caspary, 2000]. The measurement residual vector and the corresponding VCV

$$\begin{aligned} \hat{\mathbf{v}} &= \mathbf{W}^{-1} \mathbf{H}_z^T \mathbf{M}^{-1} [\mathbf{h}(\mathbf{x}^{(0)}, \mathbf{z}) + \mathbf{H}_x \delta \tilde{\mathbf{x}}] \\ \Sigma_{\hat{\mathbf{v}}\hat{\mathbf{v}}} &= \sigma_0^2 \mathbf{W}^{-1} \mathbf{H}_z^T \mathbf{M}^{-1} [\mathbf{I} - \mathbf{H}_x (\mathbf{H}_x^T \mathbf{M}^{-1} \mathbf{H}_x)^{-1} \mathbf{H}_x^T \mathbf{M}^{-1}] \mathbf{H}_z \mathbf{W}^{-1} \end{aligned} \quad (3.16)$$

In the case where a-priori information of some or all of the parameters are known with VCV  $\mathbf{P}_{xx}$ , the solution then can be expressed as

$$\begin{aligned} \mathbf{N} &= \mathbf{P}_{xx}^{-1} + \mathbf{H}_x^T \mathbf{M}^{-1} \mathbf{H}_x \\ \delta \tilde{\mathbf{x}} &= \mathbf{N}^{-1} \mathbf{H}_x^T \mathbf{M}^{-1} \delta \mathbf{z} \\ \hat{\mathbf{P}} &= \hat{\sigma}_0^2 \mathbf{P} = \hat{\sigma}_0^2 \mathbf{N}^{-1} \end{aligned} \quad (3.17)$$

The formulation in (3.17) can be useful for processing the measurements sequentially. For more discussion of least-squares techniques, refer to [Gelb, 1974, Bierman, 2006, Simon, 2006, etc.].

### 3.2.1.1. RANSAC

The **random sample consensus** (RANSAC) [Fischler et al, 1981] algorithm is an iterative parameter estimation of a mathematical model from a set of measurements which contains large number of the outliers. RANSAC first estimates candidate solutions using minimum number of measurements that are randomly selected. Then a consensus set of measurements created with consistent inliers. This process is performed iteratively and the best consensus set is used to estimate the parameters. There are many variations

of this algorithm, for example, the MSAC (M-estimator SAC), the MLESAC (Maximum Likelihood SAC) and the R-RANSAC (speeded RANSAC), etc. Performance analysis and comparisons between different algorithms and applications can be found in [Subbarao et al, 2006, Chum et al, 2008, Choi et al, 2009]. The basic RANSAC algorithm can be summarized as follows:

1. Randomly select the minimum number of measurements to solve the parameters
2. Solve the parameters with the minimum measurement set
3. With the remaining measurements, determine those that fit the model in step 2 within a predefined tolerance and add them to a consensus set
4. If the fraction number of inliers measurements is greater than a predefined threshold, then solve the parameters with the consensus set
5. Repeat steps 1 to 4 until the best consensus set is obtained within  $N$  iterations

The theoretical number of iterations ( $N$ ) can be determined based on the probability ( $p$ ) that the RANSAC algorithm in some iteration in step 1 selects only inliers. If  $u$  is the fraction of inliers in the entire measurement set and  $m_{\min}$  is the minimum number of measurements required for solving the parameters, then  $u^{m_{\min}}$  is the probability that all  $m_{\min}$  points are inliers. This implies that  $1-u^{m_{\min}}$  is the probability that at least one of  $m_{\min}$  points is an outlier. The probability that RANSAC only selects outliers is  $(1-u^{m_{\min}})^N$  and therefore

$$(1-p) = (1-u^{m_{\min}})^N \quad (3.18)$$

Solving for  $N$  one obtains

$$N = \frac{\log(1-p)}{\log(1-u^{m_{\min}})} \quad (3.19)$$

Typically, the probability  $p$  is usually set to 0.99 and  $u$  can be an approximation or obtained empirically. Depending on the scenario, both methods of determining the value of  $u$  were employed in this dissertation.

### 3.2.2. Kalman filter in discrete time

The Kalman filter, developed by Kalman [1960], is a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the covariance. Consider the following discrete system at epoch  $k$  :

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \end{aligned} \quad (3.20)$$

wherein  $\mathbf{x}_k$  is the  $n \times 1$  state vector,  $\mathbf{f}_{k-1}(\cdot)$  is  $n \times 1$  vector of the system functional model,  $\mathbf{w}_{k-1}$  the  $q \times 1$  process noise vector,  $\mathbf{z}_k$  is the  $m \times 1$  measurement vector;  $\mathbf{h}_k(\cdot)$  is the  $m \times 1$  vector of the measurement functional model and  $\mathbf{v}_k$  is the  $m \times 1$  measurement noise vector.  $\mathbf{w}_{k-1}$  and  $\mathbf{v}_k$  are assumed to be  $\mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1})$  and  $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$  with the positive definite variance matrices  $\mathbf{Q}_{k-1}$  and  $\mathbf{R}_k$ , respectively. Furthermore,  $E[\mathbf{v}_i \mathbf{v}_j^T] = \mathbf{0}$  for  $i \neq j$ ,  $E[\mathbf{w}_i \mathbf{w}_j^T] = \mathbf{0}$  for  $i \neq j$  and  $E[\mathbf{v}_i \mathbf{w}_j^T] = \mathbf{0}$  for all  $i$  and  $j$  are assumed. The time update of the state vector is given as follows [Simon, 2006]:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{0}) \\ \mathbf{P}_k^- &= \mathbf{\Phi}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{\Phi}_{k-1}^T + \mathbf{\Psi}_{k-1} \mathbf{Q}_{k-1} \mathbf{\Psi}_{k-1}^T \end{aligned} \quad (3.21)$$

where  $\Phi_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}^+}$  is the  $n \times n$  state transition matrix and  $\Psi_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{w}} \right|_{\mathbf{w}_{k-1}}$  is the  $n \times p$  coefficient matrix of  $\mathbf{w}_{k-1}$ . The measurement update in a minimum variance sense of the state vector gives

$$\begin{aligned}\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}\end{aligned}\tag{3.22}$$

wherein  $\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-}$  and  $\mathbf{K}_k$  is the Kalman gain matrix. The minus (-) and plus (+) superscripts indicate the time-update and measurement update estimates, respectively. The system innovation vector  $\mathbf{d}_k$  and system innovation matrix is given by

$$\begin{aligned}\mathbf{d}_k &= \mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-) \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\end{aligned}\tag{3.23}$$

For more discussion of Kalman filtering technique refer to [Gelb, 1974, Bierman, 2006, Grewal and Andrews, 2001, Simon, 2006, etc]. For the sake of simplicity and illustration of Kalman filter concepts, the remaining part of the dissertation shall use the linear model, i.e.

$$\begin{aligned}\mathbf{x}_k &= \Phi_{k-1} \mathbf{x}_{k-1} + \Psi_{k-1} \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k\end{aligned}\tag{3.24}$$

### 3.2.2.1. Kalman filter with time-correlated noise

The *standard* Kalman filter presented in the previous section assumed that the measurement noise vector  $\mathbf{v}_k$  is not timely correlated, i.e.,  $E[\mathbf{v}_i \mathbf{v}_j^T] = \mathbf{O}$ . Now consider the case where the vector  $\mathbf{v}_k$  is time-correlated (a.k.a. colored measurement noise), i.e.

$E[\mathbf{v}_i \mathbf{v}_j^T] \neq \mathbf{O}$ . The correlation is typically modeled by a shaping filter driven by the white Gaussian noise of zero mean [Gelb, 1974; Grewal and Andrews, 2001; Simon, 2006].

Accordingly, the system and measurement equations become

$$\begin{aligned}
\mathbf{x}_k &= \Phi_{k-1} \mathbf{x}_{k-1} + \Psi_{k-1} \mathbf{w}_{k-1}, \mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\
\mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \\
\mathbf{v}_k &= \Psi_{k-1} \mathbf{v}_{k-1} + \boldsymbol{\eta}_{k-1}, \boldsymbol{\eta}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{\boldsymbol{\eta}_{k-1}})
\end{aligned} \tag{3.25}$$

wherein  $\Psi_{k-1}$  is the  $m \times m$  transition matrix of the time-correlated measurement errors,  $\boldsymbol{\eta}_{k-1}$  is the  $m \times 1$  white driving noise vector with zero mean and the associated positive definite variance matrix  $\mathbf{Q}_{\boldsymbol{\eta}_{k-1}}$ . The covariance matrix of the time-correlated measurements between two consecutive epochs  $k$  and  $k-1$  is given as [Simon, 2006]:

$$\begin{aligned}
\mathbf{R}_{k,k-1} &= E[\mathbf{v}_k \mathbf{v}_{k-1}^T] \\
&= E[(\Psi_{k-1} \mathbf{v}_{k-1} + \boldsymbol{\eta}_{k-1}) \mathbf{v}_{k-1}^T] \\
&= \Psi_{k-1} E[\mathbf{v}_{k-1} \mathbf{v}_{k-1}^T] + E[\boldsymbol{\eta}_{k-1} \mathbf{v}_{k-1}^T] \\
&= \Psi_{k-1} \mathbf{R}_{k-1} + \mathbf{O} \\
&= \Psi_{k-1} \mathbf{R}_{k-1}
\end{aligned} \tag{3.26}$$

Two main approaches used to deal with the time-correlated measurements are the *measurement differencing* and the *state vector augmentation* [Bryson and Henrikson, 1968; Gelb, 1974; Simon, 2006]. The first approach removes the time-correlated part of the measurement noise by time-differencing the measurement equations. However, this method introduces a time latency in the updated state vector since measurements up to epoch  $k$  are used to estimate the state vector at epoch  $k-1$  [Petovello et al, 2009]. More details on the *measurement differencing* method can be found in [Bryson and Henrikson, 1968; Simon, 2006]. The state vector augmentation method extends the system and measurement models to

$$\begin{aligned}
\begin{pmatrix} \mathbf{x}_k \\ \mathbf{v}_k \end{pmatrix} &= \begin{pmatrix} \Phi_{k-1} & \mathbf{O} \\ \mathbf{O} & \Psi_{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{v}_{k-1} \end{pmatrix} + \begin{pmatrix} \Psi_{k-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{w}_{k-1} \\ \boldsymbol{\eta}_{k-1} \end{pmatrix} \\
\mathbf{z}_k &= (\mathbf{H}_k \quad \mathbf{I}) \begin{pmatrix} \mathbf{x}_k \\ \mathbf{v}_k \end{pmatrix} + \mathbf{0}
\end{aligned} \tag{3.27}$$

In short form, the above equations can be written as

$$\begin{aligned}
\tilde{\mathbf{x}}_k &= \tilde{\Phi}_{k-1} \tilde{\mathbf{x}}_{k-1} + \tilde{\Psi}_{k-1} \tilde{\mathbf{w}}_{k-1}, \tilde{\mathbf{w}}_{k-1} \sim N(\mathbf{0}, \tilde{\mathbf{Q}}_{k-1}) \\
\mathbf{z}_k &= \tilde{\mathbf{H}}_k \tilde{\mathbf{x}}_k + \tilde{\mathbf{v}}_k, \tilde{\mathbf{v}}_k \sim N(\mathbf{0}, \tilde{\mathbf{R}}_k)
\end{aligned} \tag{3.28}$$

wherein the symbol  $\sim$  denotes the augmented vectors and matrices. With the above formulation, the standard Kalman filter equations can be employed. The variance-covariance matrices of the new process noise and measurement noise vectors are

$$\begin{aligned}
\tilde{\mathbf{Q}}_{k-1} &= E[\tilde{\mathbf{w}}_{k-1} \tilde{\mathbf{w}}_{k-1}^T] \\
&= E\left( \begin{pmatrix} \mathbf{w}_{k-1} \\ \boldsymbol{\eta}_{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{w}_{k-1}^T & \boldsymbol{\eta}_{k-1}^T \end{pmatrix} \right) \\
&= \begin{pmatrix} \mathbf{Q}_{k-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}_{\boldsymbol{\eta}_{k-1}} \end{pmatrix} \\
\tilde{\mathbf{R}}_k &= E[\tilde{\mathbf{v}}_k \tilde{\mathbf{v}}_k^T] = \mathbf{O}
\end{aligned} \tag{3.29}$$

Notice that the measurement covariance is now zero. Theoretically, it is possible to run the Kalman filter with zero measurement errors [Simon, 2006]. Practically, the covariance matrix of the system innovation vector  $\tilde{\mathbf{S}}_k = \tilde{\mathbf{H}}_k \tilde{\mathbf{P}}_k \tilde{\mathbf{H}}_k^T + \tilde{\mathbf{R}}_k = \tilde{\mathbf{H}}_k \tilde{\mathbf{P}}_k \tilde{\mathbf{H}}_k^T$  has to be invertible [Grewal and Andrews, 2001]. However, the VCV matrix of the augmented state vector can become singular, especially when the transition matrix is close to identity [Bryson and Henrikson, 1968; Gelb, 1974]. To mitigate these numerical problems, [Wang et al, 2012] proposed two algorithms, the Tikhonov KF and the Perturbed-P. The former regularizes the gain matrix  $\mathbf{K}$  so that the VCV matrix of the augmented state vector becomes invertible. The later adds a small quantity to the diagonal elements of the

singular VCV matrix so that it is made positive definite. However, these two methods are not optimal.

The formulation of the time correlated measurement noise model in Equation (3.25),

$$\mathbf{v}_k = \Psi_{k-1}\mathbf{v}_{k-1} + \boldsymbol{\eta}_{k-1} \quad (3.30)$$

assumes that measurement noise vector  $\mathbf{v}_k$  at epoch  $k$  is not only correlated with the measurement noise vector  $\mathbf{v}_{k-1}$  at epoch  $k-1$  but also with the measurement noise vectors before the epoch  $k-1$ . For example, the correlation between  $\mathbf{v}_k$  and  $\mathbf{v}_{k-2}$  can be derived as

$$\begin{aligned} \mathbf{R}_{k,k-2} &= E[\mathbf{v}_k \mathbf{v}_{k-2}^T] \\ &= E[(\Psi_{k-1}\mathbf{v}_{k-1} + \boldsymbol{\eta}_{k-1})\mathbf{v}_{k-2}^T] \\ &= E[(\Psi_{k-1}(\Psi_{k-2}\mathbf{v}_{k-2} + \boldsymbol{\eta}_{k-2}) + \boldsymbol{\eta}_{k-1})\mathbf{v}_{k-2}^T] \\ &= E[(\Psi_{k-1}\Psi_{k-2}\mathbf{v}_{k-2} + \Psi_{k-1}\boldsymbol{\eta}_{k-2} + \boldsymbol{\eta}_{k-1})\mathbf{v}_{k-2}^T] \\ &= \Psi_{k-1}\Psi_{k-2}E[\mathbf{v}_{k-2}\mathbf{v}_{k-2}^T] + \Psi_{k-1}E[\boldsymbol{\eta}_{k-2}\mathbf{v}_{k-2}^T] \\ &\quad + E[\boldsymbol{\eta}_{k-1}\mathbf{v}_{k-2}^T] \\ &= \Psi_{k-1}\Psi_{k-2}\mathbf{R}_{k-2} + \mathbf{O} + \mathbf{O} \\ &= \Psi_{k-1}\Psi_{k-2}\mathbf{R}_{k-2} \neq \mathbf{O} \end{aligned} \quad (3.31)$$

If vector  $\mathbf{v}_k$  is *pairwise* time-correlated (i.e.  $\mathbf{v}_k$  is not correlated with  $\mathbf{v}_j$  for  $j < k-1$ ) and the shaping filter in (3.25) is used, then the state estimates will not be optimal. Furthermore, the solution performance can be degraded. Accordingly, Section 5.2.2 develops a method to process measurements that are *pairwise* time correlated.

### 3.2.3. Outlier detection via statistic tests

Outlier detection is important for quality assurance and quality control of the estimates from both least squares and Kalman filter. Assuming that the system and measurement models are correct, measurement outliers can be identified by performing statistic tests on

the measurement residuals and the system innovations [Wang, 1997, 2008]. The statistic test procedure can be summarized as follows [Caspary, 2000]: (a) formulate a null hypothesis  $H_0$  against its alternative  $H_a$ , (b) construct a suitable statistic based the known distribution of the random variable or vector for which  $H_0$  is true, and (c) select the risk level (a.k.a. significance level)  $\alpha$  to accept or reject  $H_0$ .

Consider the statistic tests on the individual measurement residuals. The null hypothesis is  $H_0 : v_i = 0$  and the alternative is  $H_a : v_i \neq 0$  where  $v_i$  is the  $i^{\text{th}}$  measurement residual of the residual vector  $\mathbf{v}$ . For the case where the a-priori variance factor  $\sigma_0^2$  is known, the test statistic follows the standardized normal distribution [Caspary, 2000]:

$$y = \frac{v_i}{\sigma_0 \sqrt{\sum v_i v_i}} \sim N(0,1). \quad (3.32)$$

If  $|y| < \xi_{N(0,1), 1-\alpha/2}$ ,  $\xi_{N(0,1), 1-\alpha/2}$  is the corresponding two-tails critical value, the null hypothesis  $H_0$  will be accepted. For the case where the a-priori variance factor  $\sigma_0^2$  is unknown or unreliable and the a-posteriori variance factor  $\hat{\sigma}_0^2$  is available, then the test statistic follows the  $\tau$  distribution [Caspary, 2000]:

$$y_\tau = \frac{v_i}{\hat{\sigma}_0 \sqrt{\sum v_i v_i}} \sim \tau_u. \quad (3.33)$$

If  $|y| < \xi_{\tau, u, 1-\alpha/2}$ ,  $u$  is the d.o.f. and  $\xi_{\tau, u, 1-\alpha/2}$  is the corresponding two-tails critical value, the null hypothesis  $H_0$  will be accepted. The  $\tau$  statistic is derived from the Student's  $t$  statistic as follows [Pope. 1976]:

$$\tau_u = \frac{t_{u-1}\sqrt{u}}{\sqrt{u-1+t_{u-1}^2}}. \quad (3.34)$$

Consider the statistic tests on the system innovation vector  $\mathbf{d}_k$  as a global test at epoch  $k$  [Wang, 2008]. The null hypothesis is  $H_0 : \mathbf{d}_k = \mathbf{0}$  and the alternative is  $H_a : \mathbf{d}_k \neq \mathbf{0}$ . The test statistics is

$$y_{\chi^2} = \mathbf{d}_k^T \mathbf{S}_k^{-1} \mathbf{d}_k \sim \chi_u^2. \quad (3.35)$$

If  $y < \xi_{\chi^2, u, 1-\alpha}$ , where  $u$  is the d.o.f. and  $\xi_{\chi^2, u, 1-\alpha}$  is the corresponding critical value, then the null hypothesis will be accepted. The individual elements of  $\mathbf{d}_k$  can also be tested by the following normal test statistics:

$$y_n = \frac{d_{k,i}}{\sqrt{S_{k,ii}}} \sim N(0,1) \quad (3.36)$$

where  $d_{k,i}$  is the  $i^{\text{th}}$  element in  $\mathbf{d}_k$ . The null hypothesis  $H_0$  will be accepted if  $|y| < \xi_{N(0,1), 1-\alpha/2}$ . The  $t$ - and  $F$ - tests can also be applied to the system innovation. However, these tests are not common [Wang, 2008]. Details and discussion on statistic test in least squares and Kalman filtering can be found in [Casparly, 2000] and [Wang, 1997, 2008] respectively.

### 3.3. Coordinate frames and transformations

In navigation and surveying applications, the positioning solution is typically expressed in geodetic coordinates and in roll, pitch and heading for attitude. The measurements from the sensors in a navigation system are usually given in their own specific coordinates forms. These measurements should be modeled as the functions of

the selected navigation parameters that are relevant to the application and readable to users. Therefore, it is important to define all coordinates that are involved in the development of a navigation system. This section overviews the definition of such coordinates systems applied in this dissertation and their transformations.

### 3.3.1. Inertial frame

An inertial frame is a frame which does not rotate or accelerate [Salyshev, 2004. p9]. However, it is impossible to realize. Instead a right ascension system as an approximation to the ‘true’ inertial frame is employed which is more suitable for surveying and navigation. The inertial frame (i-frame) defined here is a stationary frame with respect to the distant stars. It has its origin at the Earth’s center of mass, the  $x$ -axis points towards the mean vernal equinox,  $z$ -axis points toward to the North-pole and  $y$ -axis is perpendicular to the  $z$ -axis forming a right-hand orthogonal coordinate system.

### 3.3.2. Earth frame

The Earth-centered and Earth-fixed (e-frame) shares the same origin as the inertial frame i.e. the center of mass of the Earth. The  $x$ -axis points towards the Greenwich meridian in the equatorial plane. The  $z$ -axis points toward to the North-pole. The  $y$ -axis is perpendicular to the  $z$ -axis forming a right-hand coordinate system. The e-frame can be transformed to the i-frame by a negative rotation about the  $z$ -axis by the amount of the Greenwich Mean Sidereal Time (GMST) [Salyshev, 2004, p11]. The rotation rate vector of the e-frame with respect to the i-frame projected on to the e-frame is

$$\boldsymbol{\omega}_{ie}^e = (0 \quad 0 \quad \omega_e)^T \quad (3.37)$$

where  $\omega_e$  is the magnitude of the rotation rate of the Earth ( $7.2921158 \times 10^{-5}$  rad/s). The relationship between the e-frame position vector  $\mathbf{X}^e$  [m] and the geodetic coordinates is as follows

$$\mathbf{X}^e = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}^e = \begin{pmatrix} (R_N + h)c\varphi c\lambda \\ (R_N + h)c\varphi s\lambda \\ (R_N(1 - e^2) + h)s\varphi \end{pmatrix} \quad (3.38)$$

where  $\varphi$  is the geodetic latitude,  $\lambda$  is the geodetic longitude,  $h$  is the geodetic height,  $R_N$  is the curvature radius in the prime vertical and  $e$  is the first eccentricity of the reference ellipsoid. The WGS84 reference ellipsoid is used in this dissertation.

### 3.3.3. Navigation frame

The navigation frame (n-frame) is a local level frame that moves with the vehicle with its origin located at a predefined point on the vehicle. Its  $z$ -axis is normal to the reference ellipsoid and points downwards while its  $x$  and  $y$  axes point towards the geodetic North and East, respectively, to complete a right-handed Cartesian coordinate system. The DCM matrix from the n-frame to the e-frame is

$$\mathbf{C}_n^e = \begin{pmatrix} -s\varphi c\lambda & -s\lambda & -c\varphi c\lambda \\ -s\varphi s\lambda & c\lambda & -c\varphi s\lambda \\ c\varphi & 0 & -s\varphi \end{pmatrix} \quad (3.39)$$

The Earth's rotation rate vector in the n-frame can be described as

$$\boldsymbol{\omega}_{ie}^n = \mathbf{C}_n^e \boldsymbol{\omega}_{ie}^e = (\omega_e c\varphi \quad 0 \quad -\omega_e s\varphi)^T \quad (3.40)$$

The transport rate vector  $\boldsymbol{\omega}_{en}^n$  is the rotation rate vector of the n-frame with respect to the e-frame

$$\mathbf{\omega}_{en}^n = \left( \frac{v_E^n}{(R_N+h)} \quad -\frac{v_N^n}{(R_E+h)} \quad -\frac{v_E^n \tan \varphi}{(R_N+h)} \right)^T \quad (3.41)$$

where  $v_N^n$  and  $v_E^n$  are the North and East velocity components in [m/s], respectively, and  $R_E$  is the curvature radius of the meridian [m].

### 3.3.4. Computer frame

The computer frame ( $n^c$ -frame) is local level frame that has the origin at the computed inertial navigation position. Its  $z$ -axis is normal to the reference ellipsoid and points downwards while its  $x$  and  $y$  axes point towards the geodetic North and East, respectively, forming a right-handed Cartesian coordinate system. The misalignment vector of the  $n^c$ -frame w.r.t the  $n$ -frame is given by (Benson, 1975):

$$\mathbf{\delta\theta} = (c\varphi\delta\lambda \quad -\delta\varphi \quad -s\varphi\delta\lambda)^T \quad (3.42)$$

where  $\delta\varphi$  and  $\delta\lambda$  are the latitude and longitude errors, respectively. The DCM matrix from the  $n$ -frame to the  $n^c$ -frame can be written as

$$\mathbf{C}_n^{n^c} = \mathbf{I} - \mathbf{\delta\theta} \times \quad (3.43)$$

### 3.3.5. Body frame

The body frame ( $b$ -frame) shares the same origin with the  $n$ -frame. Its  $x$ -axis points along the vehicle's longitudinal axis and the  $z$ -axis points down while its  $y$ -axis forms a right-handed coordinate system. The DCM matrix from the  $b$ -frame to the  $n$ -frame is

$$\mathbf{C}_b^n = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \quad (3.44)$$

where  $\phi$  is the roll,  $\theta$  is the pitch and  $\psi$  is the heading.

### 3.3.6. Camera frame

The camera frame (c-frame) is the frame in which the image measurements are taken. Its origin is at the perspective center of the reference camera. Its  $x$ -axis and  $y$ -axis are parallel to the columns and rows of the CCD sensor while its  $z$ -axis points away from the CCD sensor to form a right-handed coordinate system. The camera system is assumed to be rigidly mounted on the vehicle. Hereafter, the left camera is set as the reference camera in the stereo system. The transformation from the c-frame to the b-frame via the known lever arms and boresight angles is given by

$$\mathbf{X}^b = \mathbf{C}_c^b \mathbf{X}^c + \mathbf{I}_{bc}^b \quad (3.45)$$

where  $\mathbf{X}^b$  and  $\mathbf{X}^c$  are the position vectors in the b-frame and the c-frame respectively,  $\mathbf{I}_{bc}^b$  is the lever-arm vector of the reference camera in the body frame and  $\mathbf{C}_c^b$  is the DCM between the c-frame to the b-frame. The DCM  $\mathbf{C}_c^b$  can be defined by the boresight angle vector  $\boldsymbol{\theta}_c^b$ .

### 3.3.7. n'-frame

The (n'-frame) has the same origin as the n-frame, but has an arbitrary fixed orientation with respect to the n-frame. The DCM matrix from the n'-frame to the n-frame is

$$\mathbf{C}_{n'}^n(\boldsymbol{\theta}_{n'}^n) \quad (3.46)$$

where  $\boldsymbol{\theta}_{n'}^n$  is the vector of Euler angles defining  $\mathbf{C}_{n'}^n$ . This frame is used when the roll, pitch and heading angles are not available. For example, when navigating with camera-

only systems, the object space can be defined in the  $n'$ -frame and the  $C_c^{n'}$  DCM parameterizes the orientation angles.

### 3.3.8. Summary

Figure 3.1 illustrates  $i$ -,  $e$ -,  $n$ -,  $n^c$ -,  $b$ -,  $c$ - and  $n'$ - frames.

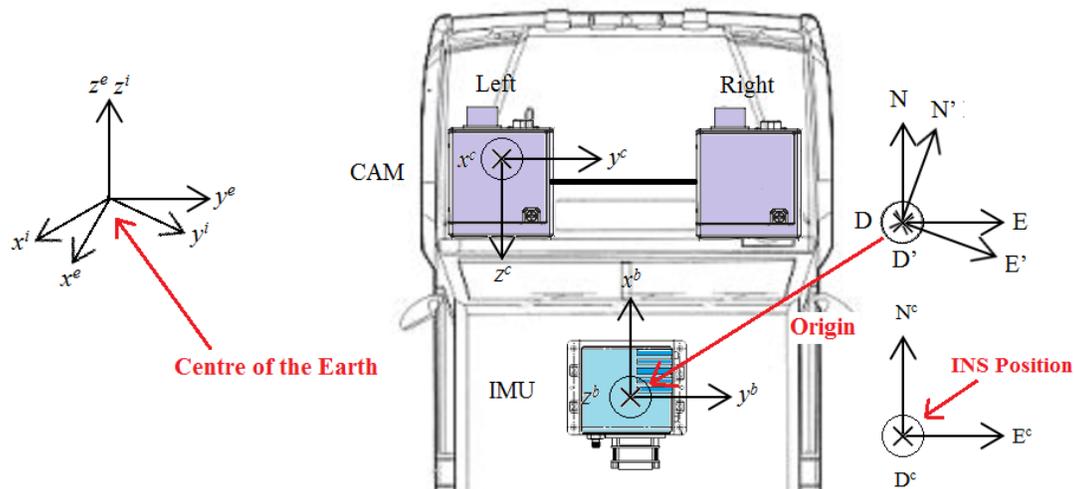


Figure 3.1 The  $i$ -,  $e$ -,  $n$ -,  $n^c$ -,  $b$ -,  $c$ - and  $n'$ - frames.

## 3.4. Global Navigation Satellite System

This section introduces the basic concepts of GNSS. A GNSS is a radio-based satellite system that is used to determine accurate position, velocity, attitude and time estimates worldwide under all weather conditions. As of May 2018 only two GNSS systems are fully operational: Global Positioning System (GPS) and Global Orbiting Navigation Satellite System (GLONASS). Europe's Galileo and China's Beidou satellite systems are currently in development and are expected to be fully operational by 2020. More details on GNSS can be found in [Parkinson and Spilker, 1996; Hofmann-Wellenhof et al., 2008].

### 3.4.1. GNSS position

A GNSS receiver computes the user's position by simple trilateration process using ranges or range differences measured to satellites [Hofmann-Wellenhof et al., 2008]. The basic principles of GNSS positioning is as follows. First the satellite ECEF position vector is computed from the ephemerides broadcast by the satellite. Secondly, the range to each satellite is measured by recording the time-of-flight of the coded satellite signal to reach the receiver. GNSS receivers typically use an inexpensive crystal clock and are not synchronized to the true system time. The synchronization error causes an offset in a measured distance. The measured distance is called the code pseudorange, which can be modelled as

$$P^j = \|\mathbf{X}_j^e - \mathbf{X}_r^e\| + c\Delta t_r \quad (3.47)$$

where  $P^j$  [m] is the pseudorange to satellite  $j$ ,  $\mathbf{X}_j^e$  [m] is the known satellite position vector,  $\mathbf{X}_r^e$  [m] is the position vector of the unknown receiver,  $c$  [ $\text{ms}^{-1}$ ] is the speed of light and  $\Delta t_r$  [s] is the receiver clock offset. Figure 3.2 illustrates the standard point positioning using pseudoranges.

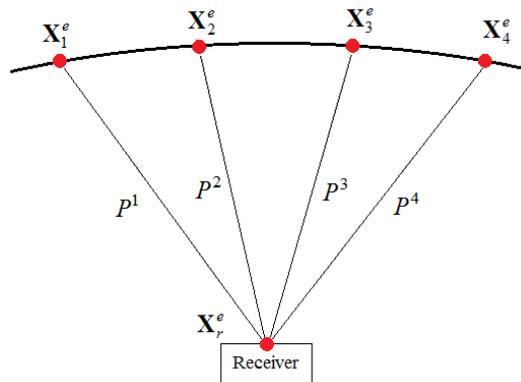


Figure 3.2 GNSS Pseudorange positioning

Finally, at least four simultaneous pseudoranges are required to solve for the four unknowns, i.e. the three components of receiver position and the clock offset. Typically these unknowns are estimated via least-squares or a Kalman filter algorithm. The accuracy of the position solution mainly depends on

- (a) the accuracy of each satellite position and clock error,
- (b) the accuracy of the pseudoranges,
- (c) the accuracy of the corrected ionosphere and troposphere delays, and
- (d) the geometry of the observed satellites.

The accuracy in autonomous standard point positioning (SPP) is within a few meters. The most accurate GNSS position can be achieved through relative positioning. This involves simultaneous observation from two receivers to same satellites. Differencing the code pseudoranges and carrier phases from the two locations to form a baseline, reduce or eliminate systematic errors when the baseline length is relatively short (typically less than 20 km). In this setup, the relative positioning accuracy between the two receivers can reach centimeter level once the phase ambiguities, inherent in the phase pseudorange measurements, are resolved to an integer [Hofmann-Wellenhof et al., 2008].

Other GNSS position positioning modes including network RTK [Vollath et al, 2000; Wübbena, 2001] and precise point positioning [Zumberge et al., 1997; Gao and Shen, 2002] will not be necessarily summarized here.

### **3.4.2. Velocity and track angle**

GNSS can determine the instantaneous velocity of a moving vehicle using the Doppler principle of radio signals. Because of the relative motion of the satellites with respect to a moving vehicle, the frequency of a signal broadcasted by the satellites is shifted when it

is received at the vehicle. This measurable Doppler shift is proportional to the relative radial velocity or range rate and can be expressed as [Hofmann-Wellenhof et al., 2008]

$$D^j = \frac{\mathbf{X}_j^e - \mathbf{X}_r^e}{\|\mathbf{X}_j^e - \mathbf{X}_r^e\|} \bullet (\mathbf{v}_j^e - \mathbf{v}_r^e) + c\Delta\dot{t}_r \quad (3.48)$$

where  $\mathbf{v}_j^e$  [ $\text{ms}^{-1}$ ] is the known satellite velocity vector,  $\mathbf{v}_r^e$  [ $\text{ms}^{-1}$ ] is the unknown receiver velocity vector and  $\Delta\dot{t}_r$  [unitless] is the change rate of the clock error. In case, the position vector of a receiver is known, a minimum of four Doppler measurements is required to solve for the receiver's velocity vector and the change rate of the clock error. The Doppler measurements can also be derived from carrier phases by computing first order central difference, which is given at epoch  $k$  as follows [Serrano et al, 2004]

$$D_k^j \approx \dot{\Phi}_k^j = \frac{1}{2\Delta t} (\Phi_{k+1}^j - \Phi_{k-1}^j) \quad (3.49)$$

where  $\Phi_{k-1}^j$  and  $\Phi_{k+1}^j$  are the phase measurements in meters at epochs  $k-1$  and  $k+1$  respectively, and  $\Delta t$  is the data sampling interval. It is simple to implement and can achieve an accuracy of  $0.005 \text{ ms}^{-1}$  [Serrano et al, 2004].

The track angle of a moving receiver can be derived from the velocity vector as follows [Grewal et al, 2001]

$$\psi = \tan^{-1} \left( \frac{v_E^n}{v_N^n} \right) \quad (3.50)$$

This model can deliver a valid heading when the absolute velocity is more significant than its uncertainty.

### 3.4.3. GNSS compass

The heading of a moving platform can be determined from a moving baseline [Graas and Braasch, 1991; Li, 1996]. This method requires two receivers whose baseline vector  $\Delta\mathbf{X}^b$  [m] in the body frame is known and fixed. Using the relative positioning method, the baseline vector  $\Delta\mathbf{X}^n$  [m] between the two GNSS receivers can be determined within an accuracy of a few centimeters. Then the heading angle in a longitudinal setup can be computed by [Reis et al, 2010]:

$$\psi = \tan^{-1}\left(\frac{\Delta E^n}{\Delta N^n}\right). \quad (3.51)$$

An accuracy of 1mm in the relative positioning of the dual antennas corresponds to 0.057 degrees in heading for a baseline length of 1m [Hofmann-Wellenhof et al., 2008]. The longer the baseline is, the higher the heading accuracy. GNSS can also be used to determine all the three attitude angles [McMillan, 1994; Giorgi, 2010], which requires a minimum of two independent baselines from at least three receivers. However this is beyond the scope of the research in this dissertation.

### 3.4.4. Summary

In this dissertation, the following GNSS positioning solution shall be applied: position derived from the relative positioning (b) velocity from carrier phase measurements (c) heading derived from the velocity vector and (d) DGNSS heading from a baseline between two antennas mounted on the moving platform. By the way, only the GNSS measurements have been applied in order to avoid any potential complication and under the considering of our time limit and the available GNSS receivers without loss of the generality with possibility of applying other GNSS measurements at large.

### **3.5. Inertial Navigation**

This section introduces the fundamental concepts of inertial navigation. The inertial navigation system (INS) is a self-contained dead-reckoning navigation system that uses accelerometers, gyroscopes and a computer to obtain the position, velocity and attitude of a moving object. The strapdown inertial navigation is widely used on vehicles such as aircrafts, space crafts, land vehicles, marine vehicles and so on. The accelerometers and gyroscopes are mounted in orthogonal triad clusters and enclosed within an inertial measurement unit (IMU) to provide three components of sensed acceleration and angular rate outputs, respectively [Rogers, 2007]. Typically, an IMU is rigidly mounted on the vehicle such that the IMU body axes are coincident with the axes of the moving body frame. The initial position, velocity and heading information are transferred to the low-cost INS from an external source, for example GNSS and GNSS compass. A computer then performs the numerical integration of the IMU outputs to yield the navigation solution, i.e., the position, velocity and attitude.

#### **3.5.1. Navigation equations**

The navigation equations with the outputs of the inertial sensors describe the dynamics of a vehicle in the specified coordinate frame. In a strap-down INS, the IMU measurements are resolved in the body frame. In order to navigate on or near Earth, the navigation solution expressed in the navigation frame is more convenient and meaningful to the user [Bekir, 2007]. The navigation equations in the navigation frame are given the following differential equations [Farrell and Barth, 1999; Bekir, 2007]:

$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{D}_r \mathbf{v}^n \\
\dot{\mathbf{v}}^n &= \mathbf{C}_b^n \mathbf{f}^b - (\boldsymbol{\omega}_{en}^n + 2\boldsymbol{\omega}_{ie}^n) \times \mathbf{v}^n + \mathbf{g}^n \\
\dot{\mathbf{C}}_b^n &= \mathbf{C}_b^n \boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^n \mathbf{C}_b^n
\end{aligned} \tag{3.52}$$

where  $\mathbf{r}$  is the geodetic position vector with latitude  $\phi$  [rad], longitude  $\lambda$  [rad] and height  $h$  [m],

$\mathbf{v}^n$  is the velocity vector [m],

$\mathbf{C}_b^n$  is the body frame to navigation frame DCM defined by attitude angles roll  $\phi$ , pitch  $\theta$  and heading  $\psi$ ,

$\mathbf{f}^b$  is the accelerometer measurement vector [ $\text{ms}^{-2}$ ],

$$\mathbf{D}_r = \text{diag}\left(\frac{1}{(R_N+h)}, \frac{1}{(R_e+h)\cos\phi}, -1\right),$$

$R_N$  is the radius of curvature of prime Meridian,

$R_E$  is the radius of curvature of the prime vertical,

$\boldsymbol{\omega}_{en}^n$  is the transport rate vector [ $\text{rads}^{-1}$ ],

$\boldsymbol{\omega}_{ie}^n$  is the Earth rotation rate vector [ $\text{rads}^{-1}$ ],

$\mathbf{g}^n$  is the gravity vector [ $\text{ms}^{-2}$ ],

$\boldsymbol{\Omega}_{ib}^b$  is the skew-symmetric form of the angular rate measurement vector  $\boldsymbol{\omega}_{ib}^b$  [ $\text{rads}^{-1}$ ], and

$\boldsymbol{\Omega}_{in}^n$  is the skew-symmetric form of rotation rate vector  $\boldsymbol{\omega}_{in}^n$  [ $\text{rads}^{-1}$ ] which is the sum of the transport rate and Earth rotation rate vectors [rad] i.e.  $\boldsymbol{\omega}_{in}^n = \boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n$ .

In (3.52)  $\mathbf{f}^b$  and  $\boldsymbol{\omega}_{ib}^b$  are the sensed specific force vector and angular rate vector of the IMU, respectively, in the body frame. The navigation solution is the position vector  $\mathbf{r}^n$ , velocity vector  $\mathbf{v}^n$  and the roll, pitch and heading that define  $\mathbf{C}_b^n$ . The transport rate vector  $\boldsymbol{\omega}_{en}^n$  is the vehicle's angular rate as it moves about a spherical Earth. The Earth rotation rate vector  $\boldsymbol{\omega}_{ie}^n$  is known. The gravity vector  $\mathbf{g}^n$  is normal to the Geoid and obtained from a gravity model. The derivation of these equations can be found in [Bekir, 2007; Rogers, 2007]. The navigation equation in (3.52) requires the heading to be known. For scenarios where the heading is not available, the navigation equation involving the wander frame and wander angle can be used. The wander frame is a local level frame where its x-axis is not slaved to the North. The wander angle is the angle between the x-axis of the wander frame and the North. Details on the wander frame and wander angle can be found in [Salyshev, 2004; Rogers, 2007].

### 3.5.2. INS error models

The INS error models are employed to analyze the systematic errors of the IMU and are used in the implementation of aided-INS data fusion. The navigation equations, which are used to generate the navigation solution, are non-linear. When the INS is aided with independent measurements via a Kalman filter algorithm, the algorithm requires a linearized error formulation of the navigation equations [Rogers, 2007]. The two common error models are the phi angle approach and the psi angle approach. The phi-error model and psi-error model are resolved the n-frame and  $n^c$ -frame respectively. The phi-error model can be written as [Scherzinger, 1994]:

$$\begin{aligned}
\delta\dot{\mathbf{X}}^n &= -\boldsymbol{\omega}_{en}^n \times \delta\mathbf{X}^n + \delta\boldsymbol{\theta} \times \mathbf{v}^n + \delta\mathbf{v}^n \\
\delta\dot{\mathbf{v}}^n &= -(\boldsymbol{\omega}_{en}^n + 2\boldsymbol{\omega}_{ie}^n) \times \delta\mathbf{v}^n + \mathbf{C}_b^n \mathbf{f}^b \times \boldsymbol{\varphi} - (\delta\boldsymbol{\omega}_{en}^n + 2\delta\boldsymbol{\omega}_{ie}^n) \times \mathbf{v}^n + \delta\mathbf{g}^n + \mathbf{C}_b^n \delta\mathbf{f}^b \\
\dot{\boldsymbol{\varphi}} &= -\boldsymbol{\omega}_{in}^n \times \boldsymbol{\varphi} + \delta\boldsymbol{\omega}_{in}^n - \mathbf{C}_b^n \delta\boldsymbol{\omega}_{ib}^b
\end{aligned} \tag{3.53}$$

wherein  $\boldsymbol{\varphi}$  is the misalignment vector of  $\mathbf{C}_b^n$ ,  $\delta\mathbf{X}^n$  is the position error vector in meters,  $\delta\mathbf{f}^b$  is the accelerometer sensor error vector and  $\delta\boldsymbol{\omega}_{ib}^b$  is the gyroscope sensor error vector. The geodetic position error vector can be express as

$$\delta\mathbf{r}^n = \mathbf{D}_r \delta\mathbf{X}^n \tag{3.54}$$

and the gravity perturbation  $\delta\mathbf{g}^n$  can be written as [Rogers, 2007]

$$\delta\mathbf{g}^n = (0 \quad 0 \quad \frac{2g}{R} \delta Z^n)^T \tag{3.55}$$

where  $g$  is the local gravity [ $\text{ms}^{-2}$ ] and  $R$  is the radius [m] of a spherical Earth. The psi-error model is given by [Scherzinger, 1994]:

$$\begin{aligned}
\delta\dot{\mathbf{X}}^{n^c} &= -\boldsymbol{\omega}_{en^c}^{n^c} \times \delta\mathbf{X}^{n^c} + \delta\mathbf{v}^{n^c} \\
\delta\dot{\mathbf{v}}^{n^c} &= -(\boldsymbol{\omega}_{en^c}^{n^c} + 2\boldsymbol{\omega}_{ie}^{n^c}) \times \delta\mathbf{v}^{n^c} + \mathbf{C}_b^{n^c} \mathbf{f}^b \times \boldsymbol{\psi} + \delta\mathbf{g}^{n^c} + \mathbf{C}_b^{n^c} \delta\mathbf{f}^b \\
\dot{\boldsymbol{\psi}} &= -\boldsymbol{\omega}_{in^c}^{n^c} \times \boldsymbol{\psi} - \mathbf{C}_b^{n^c} \delta\boldsymbol{\omega}_{ib}^b
\end{aligned} \tag{3.56}$$

wherein  $\boldsymbol{\psi}$  is the misalignment vector of  $\mathbf{C}_b^{n^c}$  and  $\delta\mathbf{g}^{n^c} = \delta\mathbf{g}^n - \mathbf{g}^n \times \delta\boldsymbol{\theta}$  [Rogers, 2007].

Both error models are equivalent [Benson, 1975]. As the position error becomes smaller, the psi-error model converges to the phi-error model. In this dissertation the psi-error model is employed since the position errors are expected to be small due the availability of GNSS position information. Furthermore, the psi-error equations contain fewer terms than the phi-error equations and therefore it is easier to be implemented in a Kalman filter.

### 3.5.3. Sensor errors and sensor error model

The accelerometer and gyroscope outputs are corrupted by a number of systematic errors, for example, biases, scale factor errors, nonlinearities, non-orthogonality of sensor axes, temperature variations and random measurement noises [Naranjo, 2008]. The biases and scale factor errors are the dominant ones in general. A bias is an additive error to the sensor output. It consists of a constant offset, a bias instability which is random in nature and temperature varying part. A scale factor is the ratio of the change in output to the input and is also contaminated by an offset, the drift rate and the temperature dependent part. The constant offsets and temperature varying part can be obtained through laboratory calibrations. The stochastic properties of the bias instability and measurement noises can be identified by two popular methods, namely the power spectral density [Yi, 2007] and the Allan variance [Allan, 1966; El-Sheimy et al, 2008]. The latter method is preferred due to its simplicity and efficiency.

Considering only the bias and scale factor, the accelerometer and gyroscope sensor errors can be modelled as [Roger et al, 2007]:

$$\begin{aligned}\delta\mathbf{x}^b &= \mathbf{b}_a + \text{diag}(\mathbf{f}^b)\mathbf{s}_b + \mathbf{w}_a \\ \delta\boldsymbol{\omega}_{ib}^b &= \mathbf{b}_g + \text{diag}(\boldsymbol{\omega}_{ib}^b)\mathbf{s}_g + \mathbf{w}_g\end{aligned}\tag{3.57}$$

where  $\mathbf{b}$  is the bias vector,  $\mathbf{s}$  is the scale factor error vector,  $\mathbf{w}$  is the measurement noise vector,  $a$  denotes the accelerometers and  $g$  denotes the gyroscopes. The biases and scale factors can be modelled as random constant or Gaussian-Markov processes. The measurement noise can be considered to be white with zero mean.

#### **3.5.4. Mechanization Equations**

The inertial navigation mechanization equations generate the position, velocity and attitude navigation states by the numerical integration of the outputs of the inertial sensors. The INS mechanization can be performed in the coordinate system required by the user, for instance, in the i-frame, e-frame or n-frame. The n-frame mechanization is commonly employed and is more meaningful to the user when navigating on or near the surface of the Earth [Titterton and Weston, 2004; Bekir, 2007]. The mechanization equation in the n-frame can be summarized in Figure 3.3. First the sensor biases and scale factor errors are compensated from the IMU raw measurements. These errors can be obtained from laboratory calibrations and/or estimated by a Kalman filter. The attitude is then updated followed by the transformation of the sensed accelerations (specific forces) from the body frame to the navigation frame. The accelerations due to gravity and Coriolis force are removed from the transformed sensed accelerations to obtain the total body accelerations with respect to the Earth. Finally the velocity and position vectors are computed by single and double integration of the total body acceleration vector, respectively.

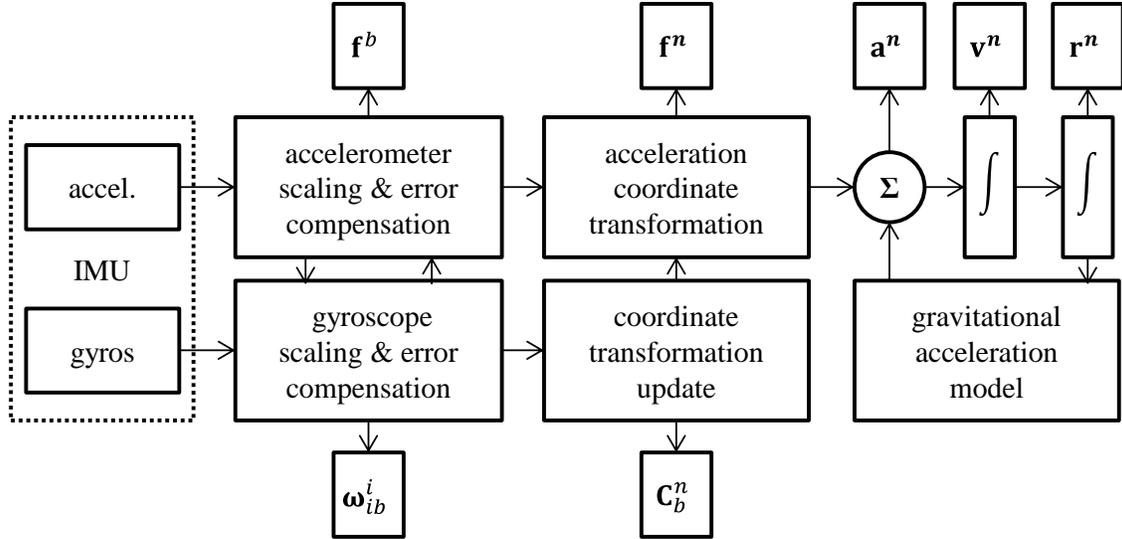


Figure 3.3 Inertial navigation mechanization [Grewal et al, 2013]

Details on the mechanization equations can be found in [Salyshev, 2004; Titterton and Weston, 2004].

### 3.5.5. Alignment

Alignment is the process of determining the initial attitude parameters. The alignment procedure typically consists of three stages (a) coarse horizontal alignment (a.k.a. coarse leveling), (b) coarse heading alignment, and (c) fine alignment [Salyshev, 2004].

In the coarse leveling stage, roll and pitch angles are obtained by accelerometer leveling while the vehicle is stationary [Bekir, 2007]:

$$\begin{aligned} \begin{pmatrix} s\phi \\ c\phi \end{pmatrix} &= \frac{1}{\sqrt{(f_y^b)^2 + (f_z^b)^2}} \begin{pmatrix} -f_y^b \\ -f_z^b \end{pmatrix} \\ \begin{pmatrix} s\theta \\ c\theta \end{pmatrix} &= \frac{1}{g_z^n} \begin{pmatrix} f_x^b \\ -(f_y^b s\phi + f_z^b c\phi) \end{pmatrix} \end{aligned} \quad (3.58)$$

where  $\mathbf{f}^b = (f_x^b, f_y^b, f_z^b)^T$  and  $g_z^n$  is the gravity in downward direction.

In practice, the accelerometer measurements from multiple epochs in stationary mode are averaged to reduce the measurement noise. However, the roll and pitch estimates are corrupted due to the biases in the accelerometer measurements.

In the coarse heading alignment stage, the heading can be obtained by gyro-compassing. Gyro-compassing uses the rotation rate of the Earth sensed by the gyroscopes to determine the heading. However, low-cost IMUs cannot observe the Earth's rotation rate due to large biases and noise in the angular rate measurements. The heading must be transferred from other sensors such as from a GNSS compass, GNSS track angle or a magnetic compass.

The fine alignment phase is performed using a Kalman filter. In this stage the attitude angles and together with the sensor error estimates are refined [Roger, 2007].

#### **3.5.6. GNSS aided-inertial navigation**

The Kalman filter is widely used for fusing GNSS and IMU data. There are two popular integration architectures, namely *loosely-coupled* integration and *tightly-coupled* integration [Scherzinger, 2000; Salyshev, 2004; Rogers, 2007]. The former consists of two parallel estimation processes. One filter processes the GNSS measurements to deliver the GNSS position, velocity and/or heading solutions, while the other Kalman filter integrates the GNSS navigation solution of the first filter as the aiding measurements and the inertial navigation solution from the inertial mechanization (Figure 3.4).

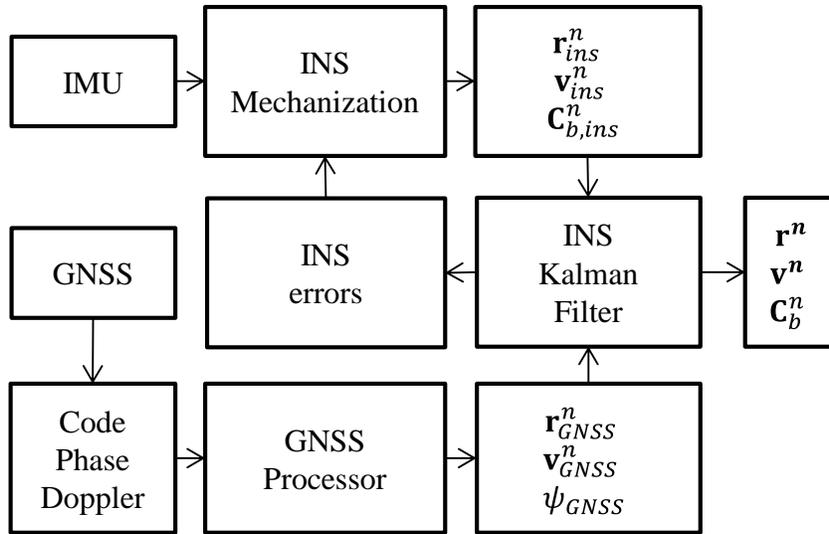


Figure 3.4 Loosely coupled GNSS/IMU integration architecture

The tightly-coupled integration architecture, described in Figure 3.5, directly integrates the GNSS raw observables to aid the inertial navigation solution [Scherzinger, 2000].

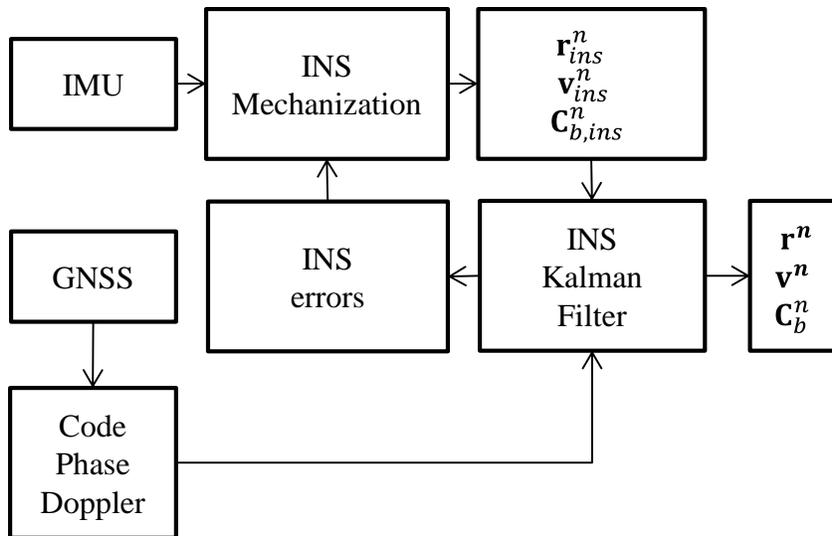


Figure 3.5 Tightly coupled GNSS/IMU integration architecture

The loosely-coupled method is simpler to implement and can easily integrate GNSS-only software in the system. It yields accurate navigation solution under full satellite coverage and low multipath environments. The tightly-coupled method is more difficult

to implement. Unlike the loosely-coupled method, however, it can still operate when fewer than four satellites are available [Moon, 2000].

There are two strategies to integrate IMU with an aiding sensor, namely the direct (total state) and indirect (error state) approaches. The state vector in the direct approach consists of the state variables and the measurements are the IMU outputs and the aiding sensor. In the indirect approach, the state vector consists of the errors of estimated variables and the measurements are the differences between the computed INS solution and the aiding sensor. For instance, the measurement equation for the indirect loosely-coupled GNSS-aided inertial navigation is:

$$\delta \mathbf{z}_{GNSS,k} = \begin{pmatrix} \mathbf{r}_{INS,k}^n - \mathbf{r}_{GNSS,k}^n \\ \mathbf{v}_{INS,k}^n - \mathbf{v}_{GNSS,k}^n \\ \psi_{INS,k}^n - \psi_{GNSS,k}^n \end{pmatrix} = \mathbf{H} \delta \mathbf{x}_{INS,k} + \mathbf{v}_{GNSS,k}. \quad (3.59)$$

In this dissertation, the indirect method is employed because (a) it works well with linear/linearized models (b) it can be modified to work with multiple types of measurements e.g. visual odometry and GNSS, and (c) uses less computation resources than the direct approach.

Actually, a generic multisensor integration strategy (GMIS) has also been developed [Wang and Sternberg, 2000; Qian et al, 2015, 2016; Wang et al, 2015 and Qian, 2017] where the kinematics of a rigid body was considered in integration GNSS and IMU measurements. The GMIS method uses a kinematic trajectory model as the system model in the navigation Kalman filter and the measurements from all of the individual sensors, including IMU measurements, are directly involved in the estimation of the navigation states. The advantages are: (a) the IMU measurements are directly applied through the measurement updates in Kalman filter instead of their application in inertial

mechanization and error measurements, (b) the system observation redundancy is increased due to angular rate and acceleration of a rigid body model, (c) the novel filter structure allows the easier fusion of multiple inertial sensors and all types of measurements, e.g., relative measurements, and (d) direct error analysis can be performed on raw sensor data and virtual zero-mean process noise measurements. The GMIS has shown its high potential to work with low-cost IMUs in particular [Wang et al, 2015 and Qian, 2017]. Readers may have noticed that the comprehensive research activities with the mentioned GMIS, however, have been run in parallel with my PhD studies so that it has not been adapted for my research due to the time limit of my PhD studies.

### **3.6. Photogrammetry and image-based navigation**

Photogrammetry is the science of making measurements from photographs and images for making inferences about the size, shape, and spatial attributes of the objects in images [Bethel, 2003]. Image-based navigation (IBN), on the other hand, refers to the determination of the navigation parameters of a camera system derived from a successive sequence of images. IBN employs the theories and techniques in computer vision and in photogrammetry to convert off-the-shelf digital cameras to a navigation sensor. This section presents the relevant background of IBN that will be used in the later development.

#### **3.6.1. Mathematical photogrammetry**

The equations in photogrammetry relate the coordinates of 3D objects to the 2D coordinates of the objects in images. The equations employed in the research are (a) the

collinearity equations, (b) the coplanarity equation, and (c) the scale-restraint equation. The choice of condition equations depends on the particular problem to be solved.

### 3.6.1.1. Collinearity equations

The fundamental characteristic of a perfect perspective camera model is that the object point, its corresponding image point, and the lens perspective center all lie on a line in space. The collinearity equations describe the perspective transformation between the image space and the object space [Bethel, 2003; Ghosh, 2005]:

$$\mathbf{x}_i^c = \begin{pmatrix} x_i - x_o \\ y_i - y_o \\ -f \end{pmatrix} = s(\mathbf{C}_c^n)^T \begin{pmatrix} X_i^n - X^n \\ Y_i^n - Y^n \\ Z_i^n - Z^n \end{pmatrix} \quad (3.60)$$

wherein  $(x_i, y_i)$  [px] is the coordinates of an image point,  $(x_o, y_o)$  [px] is the coordinates of the principal point and  $f$  [px] is the focal length,  $s$  is the scale factor,  $\mathbf{C}_c^n$  is the DCM from the camera frame to the navigation frame,  $\mathbf{X}_i^n$  [m] are the 3D object coordinates and  $\mathbf{X}^n$  [m] the perspective center coordinates in the navigation frame, respectively. The elimination of the scale factor  $s$  yields the following equations

$$\begin{aligned} x_i &= x_o - f \left( \frac{[C_{c,11}^n, C_{c,21}^n, C_{c,31}^n](\mathbf{X}_i^n - \mathbf{X}^n)}{[C_{c,13}^n, C_{c,23}^n, C_{c,33}^n](\mathbf{X}_i^n - \mathbf{X}^n)} \right) \\ y_i &= y_o - f \left( \frac{[C_{c,12}^n, C_{c,22}^n, C_{c,32}^n](\mathbf{X}_i^n - \mathbf{X}^n)}{[C_{c,13}^n, C_{c,23}^n, C_{c,33}^n](\mathbf{X}_i^n - \mathbf{X}^n)} \right). \end{aligned} \quad (3.61)$$

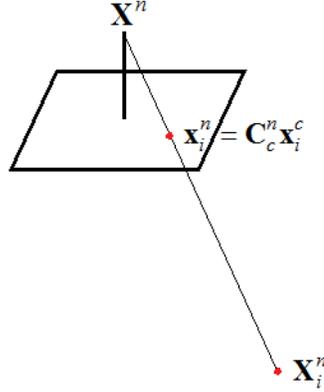


Figure 3.6 The collinearity condition

The corresponding short-hand notation can simplify to

$$\begin{aligned} x_i &= x_o - fg(\mathbf{X}^n, \boldsymbol{\theta}_c^n, \mathbf{X}_i^n) \\ y_i &= y_o - fh(\mathbf{X}^n, \boldsymbol{\theta}_c^n, \mathbf{X}_i^n) \end{aligned} \quad (3.62)$$

where  $\boldsymbol{\theta}_c^n$  [rad] is the vector of Euler angles,

$$g(\mathbf{X}^n, \boldsymbol{\theta}_c^n, \mathbf{X}_i^n) = \frac{[C_{c,11}^n, C_{c,21}^n, C_{c,31}^n] \Delta \mathbf{X}^n}{[C_{c,13}^n, C_{c,23}^n, C_{c,33}^n] \Delta \mathbf{X}^n} \quad (3.63)$$

and

$$h(\mathbf{X}^n, \boldsymbol{\theta}_c^n, \mathbf{X}_i^n) = \frac{[C_{c,12}^n, C_{c,22}^n, C_{c,32}^n] \Delta \mathbf{X}^n}{[C_{c,13}^n, C_{c,23}^n, C_{c,33}^n] \Delta \mathbf{X}^n}. \quad (3.64)$$

If the coordinates of the object and the perspective center are relatively close (i.e. a few hundred meters) and are not near the polar regions, then position difference between the object and the perspective center  $\Delta \mathbf{X}^n$  can be written as [Gade, 2010]

$$\Delta \mathbf{X}^n = \mathbf{X}_i^n - \mathbf{X}^n = \mathbf{D}_r^{-1}(\mathbf{r}_i - \mathbf{r}) \quad (3.65)$$

where  $\mathbf{r}_i$  and  $\mathbf{r}$  are the geodetic coordinates of the object and the perspective center respectively.

### 3.6.1.2. Coplanarity equation

The coplanarity condition, as shown in Figure 3.7, implies that the two perspective centers, any object point and the corresponding conjugate image points on a pair of images, must all lie in a common plane [Ghosh, 2005].

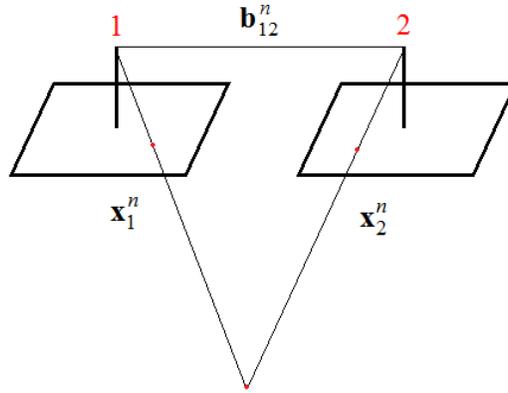


Figure 3.7 Two view geometry

Since the baseline vector  $\mathbf{b}_{12}^n$  between the two perspective centers and the position vectors  $\mathbf{x}_1^n = \mathbf{C}_{c,1}^n \mathbf{x}_1^c$  [px] and  $\mathbf{x}_2^n = \mathbf{C}_{c,2}^n \mathbf{x}_2^c$  [px] are coplanar, their scalar triple product is zero:

$$\mathbf{b}_{12}^n \bullet \mathbf{x}_1^n \times \mathbf{x}_2^n = 0 \quad (3.66)$$

The coplanarity equation is mainly used in determining the relative orientation between two views.

### 3.6.1.3. Scale restraint equation

The scale-restraint equation is a three-view constraint equation. Consider image point vectors  $\mathbf{x}_1^n = \mathbf{C}_{c,1}^n \mathbf{x}_1^c$  [px],  $\mathbf{x}_2^n = \mathbf{C}_{c,2}^n \mathbf{x}_2^c$  [px] and  $\mathbf{x}_3^n = \mathbf{C}_{c,3}^n \mathbf{x}_3^c$  [px] shown in Figure 3.8:

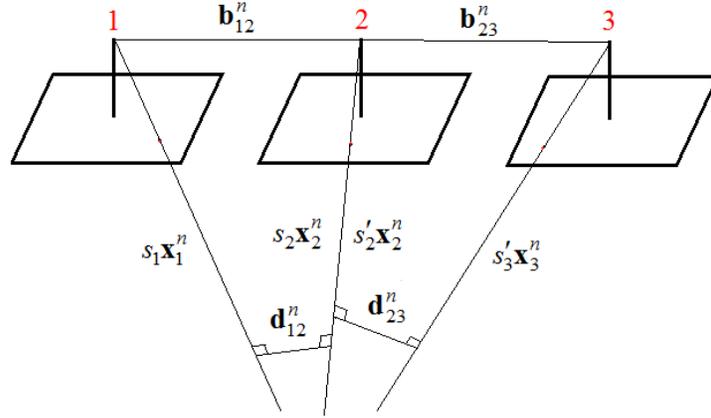


Figure 3.8 Three view geometry

wherein  $\mathbf{b}_{12}^n$  and  $\mathbf{b}_{23}^n$  are the baselines between images 1 and 2, and between images 2 and 3, respectively.  $s_1$  and  $s_2$  are the scale factors in stereomodel 1-2. Similarly,  $s'_2$  and  $s'_3$  are the scale factors in stereomodel 2-3. The three image point vectors  $\mathbf{x}_1^n$ ,  $\mathbf{x}_2^n$  and  $\mathbf{x}_3^n$  may fail to intersect at a common point due to scale variations [Ghosh, 2005]. The ‘mismatch’ vector  $\mathbf{d}_{12}^n$  is perpendicular to both of  $\mathbf{x}_1^n$  and  $\mathbf{x}_2^n$ . It is computed as  $\mathbf{d}_{12}^n = \mathbf{x}_1^n \times \mathbf{x}_2^n$ , while the other ‘mismatch’ vector is given as  $\mathbf{d}_{23}^n = \mathbf{x}_2^n \times \mathbf{x}_3^n$ .  $s_{12}$  and  $s'_{23}$  are the scale factors of the vectors  $\mathbf{d}_{12}^n$  and  $\mathbf{d}_{23}^n$  respectively. For the stereomodels 1-2 and 2-3 to intersect at the same point, the scale factors  $s_2$  and  $s'_2$  must be equal in magnitude and opposite in sign, i.e.

$$s_2 + s'_2 = 0. \quad (3.67)$$

From the principles of vector analysis, any four vectors  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$  in three-dimensional space can be related to each other through

$$\begin{aligned} (\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) &= (\mathbf{a} \cdot \mathbf{c} \times \mathbf{d}) \mathbf{b} - (\mathbf{b} \cdot \mathbf{c} \times \mathbf{d}) \mathbf{a} \\ &= (\mathbf{a} \cdot \mathbf{b} \times \mathbf{d}) \mathbf{c} - (\mathbf{a} \cdot \mathbf{b} \times \mathbf{c}) \mathbf{d} \end{aligned} \quad (3.68)$$

This implies,

$$\begin{aligned}
(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})\mathbf{c} &= (\mathbf{a}\cdot\mathbf{c}\times\mathbf{d})\mathbf{b} - (\mathbf{b}\cdot\mathbf{c}\times\mathbf{d})\mathbf{a} + (\mathbf{a}\cdot\mathbf{b}\times\mathbf{c})\mathbf{d} \\
\mathbf{c} &= \frac{(\mathbf{a}\cdot\mathbf{c}\times\mathbf{d})}{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})}\mathbf{b} - \frac{(\mathbf{b}\cdot\mathbf{c}\times\mathbf{d})}{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})}\mathbf{a} + \frac{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{c})}{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})}\mathbf{d} \\
\mathbf{c} &= b\mathbf{b} - a\mathbf{a} + d\mathbf{d}
\end{aligned} \tag{3.69}$$

where the scalar multipliers  $a$ ,  $b$  and  $d$  are

$$\begin{aligned}
a &= \frac{(\mathbf{b}\cdot\mathbf{c}\times\mathbf{d})}{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})} \\
b &= \frac{(\mathbf{a}\cdot\mathbf{c}\times\mathbf{d})}{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})} \\
d &= \frac{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{c})}{(\mathbf{a}\cdot\mathbf{b}\times\mathbf{d})}
\end{aligned} \tag{3.70}$$

Consider equating  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$  in (3.69) with  $\mathbf{x}_2^n, \mathbf{x}_1^n$ ,  $\mathbf{b}_{12}^n$  and  $\mathbf{d}_{12}^n$ , respectively, and also

with  $\mathbf{x}_3^n, \mathbf{x}_2^n$ ,  $\mathbf{b}_{23}^n$  and  $\mathbf{d}_{23}^n$ , respectively. One obtains:

$$\begin{aligned}
\mathbf{b}_{12}^n &= s_1\mathbf{x}_1^n - s_2\mathbf{x}_2^n + s_{12}\mathbf{d}_{12}^n \\
\mathbf{b}_{23}^n &= s'_2\mathbf{x}_2^n - s'_3\mathbf{x}_3^n + s'_{23}\mathbf{d}_{23}^n
\end{aligned} \tag{3.71}$$

From (3.70)  $s_2$  and  $s'_2$  can be computed as

$$\begin{aligned}
s_2 &= \frac{\mathbf{x}_1^n \cdot \mathbf{d}_{12}^n \times \mathbf{b}_{12}^n}{\mathbf{x}_1^n \cdot \mathbf{d}_{12}^n \times \mathbf{x}_2^n} \\
s'_2 &= -\frac{\mathbf{x}_2^n \cdot \mathbf{d}_{23}^n \times \mathbf{b}_{23}^n}{\mathbf{x}_2^n \cdot \mathbf{d}_{23}^n \times \mathbf{x}_3^n}
\end{aligned} \tag{3.72}$$

Substitute (3.72) in (3.67)

$$\frac{\mathbf{x}_1^n \cdot \mathbf{d}_{12}^n \times \mathbf{b}_{12}^n}{\mathbf{x}_1^n \cdot \mathbf{d}_{12}^n \times \mathbf{x}_2^n} - \frac{\mathbf{x}_2^n \cdot \mathbf{d}_{23}^n \times \mathbf{b}_{23}^n}{\mathbf{x}_2^n \cdot \mathbf{d}_{23}^n \times \mathbf{x}_3^n} = 0. \tag{3.73}$$

Equation (3.73) is the scale restraint equation [Bethel, 2003; Ghosh, 2005]. It is mainly used for solving successive relative orientations and for transferring the scale in a sequence of images.

### 3.6.2. Camera error sources

This subsection reviews the systematic and random errors associated with a stereo camera system.

#### 3.6.2.1. Image distortion

Image distortion is due to the imperfections of the camera hardware system during manufacture and causes the image coordinates of a point to deviate from its true location. There are five types of image distortions: (a) the focal length error, (b) the principal point error, (c) the radial lens distortion, (d) the decentering lens distortion, and (e) the affine distortion.

There have been many variations in modelling of the image distortions [Ebner, 1976, El-Hakim and Faig, 1977, Grün, 1978, Tang, 2012]. The model developed by Brown [1971] is widely accepted in close range photogrammetry for camera self-calibration.

The image distortions  $(\Delta x_{d,i}, \Delta y_{d,i})$  corresponding to a specific point are given by

$$\begin{aligned}\Delta x_{d,i} &= -\Delta x_0 - \frac{\bar{x}}{f} \Delta f + \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + p_1(r^2 + 2\bar{x}^2) + 2p_2 \bar{x}\bar{y} \\ &\quad - A_1 \bar{x} + A_2 \bar{y} \\ \Delta y_{d,i} &= -\Delta y_0 - \frac{\bar{y}}{f} \Delta f + \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + 2p_1 \bar{x}\bar{y} + p_2(r^2 + 2\bar{y}^2)\end{aligned}\tag{3.74}$$

where  $r$  is the radial distance from the principal point to the image point ( $r^2 = \bar{x}^2 + \bar{y}^2 = (x_i - x_o)^2 + (y_i - y_o)^2$ ),  $\Delta f$  is the error of the focal length,  $(\Delta x_0, \Delta y_0)$  is the principal

point error,  $k_i$  and  $p_i$  are the coefficients of the radial and decentering distortions, respectively, and  $A_i$  ( $i = 1, 2$ ) are the coefficients of the affine deformation.

Other causes of the image distortion are atmospheric refraction and Earth's curvature. The atmospheric refraction bends light rays from the object to the camera and causes the image point to be shifted similar to the radial distortions [Wolf and Dewitt, 2000]. The collinearity equations assume that the objects are defined in a 3-D Cartesian coordinate system [Ghosh, 2005] and do not take account into the Earth's curvature. The corrections for the atmospheric refraction and Earth's curvature are employed in aerial photogrammetry since they can be significant. However, these errors are insignificant in close range photogrammetry since the objects are expected to have a range of shorter than 100 m.

### 3.6.2.2. *Lever-arms, boresight angles, stereo baseline and relative orientation*

The lever-arms can be the position vector between two sensors or between a sensor and a reference point. Once the sensors are assembled on the vehicle, the translational offsets between the individual sensors are assumed to be constant. In a stereo image-aided inertial navigation system, there are two lever-arm vectors as shown in Figure 3.9. The first one is the baseline vector  $\mathbf{b}_{LR}^c$  [m] between the perspective centers of two cameras. The second lever-arm vector is the vector  $\mathbf{la}_L^b$  [m] between the reference camera (in this example it is the left camera) and a reference point on the vehicle. For practical purposes,  $\mathbf{b}_{LR}^c$  and  $\mathbf{la}_L^b$  vectors are expressed in the camera and body frames respectively. The lever-arms can be determined directly using a measuring tape or a total

station. Alternatively, they can be estimated as part of the states in the data fusion using least-squares or a Kalman filter.

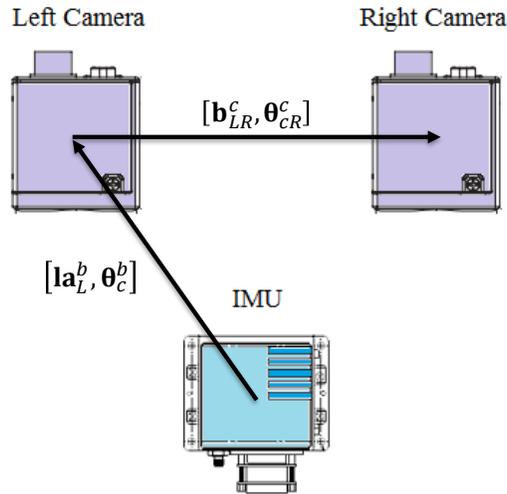


Figure 3.9 The geometrical relationship between an IMU and a stereo camera system: Lever-arms ( $\mathbf{la}_L^b$ ), boresight angles ( $\boldsymbol{\theta}_c^b$ ), stereo baseline ( $\mathbf{b}_{LR}^c$ ) and relative orientation ( $\boldsymbol{\theta}_{CR}^c$ ).

The boresight angles  $\boldsymbol{\theta}_c^b$  [rad] are the mounting angles between the reference camera system and the body frame. They are also assumed to be fixed after the sensors are assembled. Unlike the lever-arms, these angles cannot be measured directly and can only be estimated indirectly. In a stereo camera system, the relative orientation  $\boldsymbol{\theta}_{CR}^c$  [rad] of the right camera with respect to the left is also unknown and can only be obtained indirectly.

### 3.6.2.3. Image Motion Blur

Image motion is the smearing or blurring of an image due to the vibration and the relative movement of a camera with respect to the scene during the opening of its shutter (i.e. exposure). This may result in resolution degradation and displacement of the image point [Ghosh, 2005] and therefore decrease the accuracy of the image measurements. The longer the exposure time, the closer 3D objects, the higher the velocity and/or the higher

the attitude rate, the larger the motion blur will be. To mitigate the motion blur, the exposure time needs to be reduced or one needs to avoid high dynamics. The images can be deblurred by deconvolution using a blur kernel (or point spread function), which can enhance the visual quality of the image so that image features can be easily identified. However, these conventional methods do not reverse any image blur in a photogrammetrically correct and precise way [Sieberth et al, 2014].

#### *3.6.2.4. Image noise and filtering*

The visual quality of an image is important for detecting and matching point features. Image noise causes random variations in the brightness of each pixel. It degrades the quality of the image and appears as grains. The main types of image noise are (a) Gaussian noise, (b) speckle noise, (b) salt-and-pepper, and (d) shot noise [Farooque and Rohankar, 2013]. Gaussian noise (amplifier noise) is the additive random noise and independent between pixels and independent of the signal intensity. Speckle noise is the random multiplicative noise of the signal intensity. Salt-and-pepper noise (impulse noise) is the dark pixels in bright regions and the bright pixels in dark regions. Shot noise (Poisson Noise) is caused by the variation in the number of photons sensed at a given exposure level.

#### **3.6.3. Camera auto-calibration considerations**

The camera auto-calibration is performed via a bundle adjustment. There are several factors to be considered for achieving a successful and accurate auto-calibration. They are summarized below [Brown, 1989; Fraser, 2006; Tang, 2012]:

- The accuracy and geometric stability of a photogrammetric network increases with increasing convergence angles between the images (i.e. the angle between two point correspondences to an object). This implies the accuracy increases with longer baselines between the images or closer distance of the object or both. Furthermore, orthogonal roll angles must be present in order to decouple the IO and EO parameters.
- The accuracy increases with increasing number of image points per object up to a maximum of 8 points per object. The accuracy does not significantly improve after 8 points per object.
- The accuracy enhances with the increasing number of image points per image. However, the accuracy does not significantly improve after a few tens of points per image.
- The image points should be evenly distributed points throughout the image format to avoid degeneracy of a least-squares solution and to obtain reliable image distortion estimates. This is can be accomplished by the bucketing technique [Zhang et al, 1995] where the image is evenly divided into non-overlapping regions (buckets) and constant number of point matches are chosen from each bucket.
- The objects in the scene should be stationary (strictly w.r.t. the e-frame), stable and varying in distances (w.r.t. the c-frame).

The accuracy and repeatability assessment of the calibration parameters is not performed in the image space but in the object space since the 3D measurements are most

important to the user [Fraser, 2012]. This is done by comparing the computed object space coordinates with GCPs.

The number of the lens distortion terms in the model is important. If it is too small, the least-squares solution becomes biased. If it is too large, the solution will be weakened due to the over-parameterization [Fraser, 2006]. Most of the radial lens distortion is generally accounted by the second term  $k_2 r^4$  [Barazzetti et al, 2011]. The terms with  $k_3$  and even with  $k_4$  are typically included in high-accuracy applications and wide-angle lenses. The decentering distortion parameters  $p_i$  are highly correlated with the principal point offset. Furthermore, they are generally small. When they are omitted in the auto-calibration, the decentering error gets absorbed by the principal point offset error [Fraser, 2012]. The affine distortion parameters  $A_i$  were originally included in the self-calibration model by Brown [1971] because they could be significant in film cameras. However, this distortion is rarely significant in the modern digital cameras [Tang, 2013]. The number of the lens distortion terms can be empirically determined by performing the calibration with various lens distortion models and the most accurate combination is chosen.

#### **3.6.4. Image-based navigation**

This section describes image-based navigation (IBN) methods. The modern IBN consists of three main steps (a) feature detection, (b) feature matching, and (c) pose or pose change estimation. There are two major IBN approaches namely the visual odometry (VO) and the visual SLAM. The former uses the tracked features only from the consecutive frames while the latter applies the well-established SLAM technique

[Murcott et al, 2011]. It is assumed that a stereo camera system is employed and the cameras have been calibrated.

### 3.6.4.1. Feature detection

Feature detection is an essential step of many computer vision applications [Szeliski, 2011]. At this step, the feature detector locates those interesting points such as corners and blobs on an image. A corner is defined as a point at the intersection of two or more edges while a blob is an image pattern that differs in properties, such as brightness or color and texture in comparison with the surrounding regions [Scaramuzza and Fraundorfer, 2011].

A point feature detector consists of two steps. First, an image is filtered using a mask. Figure 3.10 shows an example of a 5x5 blob mask and a 5x5 corner mask. Then non-maxima suppression is applied on the filtered image where all the local minima (or maxima) locations are identified. The resulted outputs are the locations of the detected features [Scaramuzza and Fraundorfer, 2011].

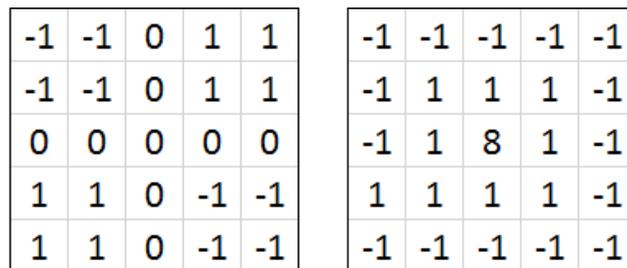


Figure 3.10 The corner (left) and blob (right) masks

The desirable qualities of a feature detector are its repeatability, localization accuracy, computational efficiency, robustness distinctiveness, photometric invariance and geometric invariance. Examples of the corner detectors are Harris [Harris et al., 1988],

Shi-Tomasi [Tomasi and Shi, 1994] and FAST [Rosten and Drummond, 2006]. And examples of the blob detectors include SIFT [Lowe, 1999], PCBR [Deng et al., 2007] and SURF [Bay et al., 2008].

#### 3.6.4.2. *Feature description and matching*

In the feature description step, a feature descriptor is constructed using the neighboring pixels around the feature points. Then the feature descriptor can be matched against descriptors from another image frame in order to obtain the correspondences. The simplest method to generate a descriptor is to extract an  $m \times m$  square patch surrounding each feature point, where  $m$  is odd, and then construct an  $m \times m - 1$  descriptor vector from the pixels in the window excluding the feature point itself [Howard, 2008]. The next step is to construct a score matrix between the feature points of the two frames to be matched. This is achieved by computing simple error metrics [Szeliski, 2011] such as the sum-of-absolute differences (SAD), sum-of-squared differences (SSD) or normalized cross-correlation (NCC) using the descriptor vectors between the two frames. If the values in the score matrix for the SAD and SSD are close to 0 or if the NCC value is close to 1, it indicates that the features between the frames are similar. The descriptors based on the local appearance are not invariant to the orientation, scale and illumination changes [Scaramuzza and Fraundorfer, 2011]. On the other hand, the descriptors, such as SIFT or SURF, are invariant to the orientation, scaling and illumination changes and may improve the matching accuracy. However, they are computationally expensive. In IBN, the local appearance of the features is not expected to change between consecutive and stereo image frames. This is due to the fact that video cameras can capture images at a high rate, the stereo cameras are usually parallel with each other, the stereo baseline

length is relatively short (less than 1m) and the camera motion are typically smooth [Geiger, 2011]. Hence, the simple SAD, SSD or NCC methods are usually sufficient for IBN systems.

Searching and matching for feature points between two frames can be inefficient since all the points are searched for potential matches. This can be time consuming if the number of the being detected features is large, for example, in the thousands. In order to reduce the searching radius, one can constrain the matching in a region where a potential match is expected. Between stereo pairs, the search can be performed along the epipolar lines [Bin Rais, et al, 2003]. Between the consecutive frames, the locations of the features on the next frame can be predicted using a motion model [Scaramuzza and Fraundorfer, 2011] or any available external sensor, for example, an IMU [Veth et al, 2006]. To increase the robustness of the matching process, one can employ the **random sample consensus** (RANSAC) algorithm [Fischler et al, 1981], which was overviewed in Section 3.2.1.1.

#### 3.6.4.3. Stereo Space Intersection

Stereo space intersection (a.k.a. stereo triangulation) is a technique that is commonly used to determine the 3D coordinates of an object from 2D image correspondences based on the known interior orientation and exterior orientation parameters. There are two methods to determine the 3D object coordinates: (a) the *simplified* triangulation equation and (b) the collinearity equations. The *simplified* model [Sibley et al, 2005] is

$$\begin{aligned} \mathbf{I}_i^c &= h_{stereo}(\mathbf{x}_{L,i}, \mathbf{x}_{R,i}) \\ &= \begin{pmatrix} x_{L,i} - x_{L,0} \\ y_{L,i} - y_{L,0} \\ -f \end{pmatrix} \frac{b_{LR}}{x_{L,i} - x_{R,i}} \end{aligned} \quad (3.75)$$

where  $\mathbf{I}_i^c$  [m] is the 3D position vector of the object w.r.t. the left camera,  $b_{LR}$  is the baseline length between the two cameras, and  $\mathbf{x}_{L,i}$  and  $\mathbf{x}_{R,i}$  are the image position vectors of the  $i$ -th object in left and right image, respectively. This model assumes that the stereo images have been rectified and has a unique solution (i.e. 3 equations and 3 unknowns). The space intersection expressed by collinearity equations is an over-determined system (i.e. 4 equations and 3 unknowns) so that the 3D object coordinates are estimated using least-squares. The four equations for space intersection by collinearity (corrected for image distortion) are given as follows [Bayoud et al, 2004]

$$\begin{aligned}
x_{L,i} &= x_{L,o} - f_L g_L(\mathbf{0}, \mathbf{0}, \mathbf{I}_i^c) + v_{x_{L,i}} \\
y_{L,i} &= y_{L,o} - f_L h_L(\mathbf{0}, \mathbf{0}, \mathbf{I}_i^c) + v_{y_{L,i}} \\
x_{R,i} &= x_{R,o} - f_R g_R(\mathbf{b}_{LR}^c, \boldsymbol{\theta}_{cR}^c, \mathbf{I}_i^c) + v_{x_{R,i}} \\
y_{R,i} &= y_{R,o} - f_R h_R(\mathbf{b}_{LR}^c, \boldsymbol{\theta}_{cR}^c, \mathbf{I}_i^c) + v_{y_{R,i}}
\end{aligned} \tag{3.76}$$

#### 3.6.4.4. Visual SLAM

Simultaneous localization and mapping (SLAM) is a navigation technique widely used in robotics. It incrementally builds a consistent map of landmarks in an unknown environment while simultaneously determining the location of the mobile system within this map [Dissanayake et al, 2006; Durrant-Whyte and T. Bailey, 2006a; Thrun et al. 2008]. The Extended Kalman filter (EKF) can be used to solve the SLAM problem and is referred as the EKF-SLAM algorithm. The system and measurement model for EKF-SLAM at epoch  $k$  is given as [Durrant-Whyte and Bailey, 2006a]:

$$\begin{aligned}
\begin{pmatrix} \mathbf{x}_k \\ \tilde{\mathbf{m}}_k \end{pmatrix} &= \begin{pmatrix} f(\mathbf{x}_k) + \mathbf{w}_{k-1} \\ \tilde{\mathbf{m}}_{k-1} \end{pmatrix}, \mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\
\mathbf{z}_{k,i} &= \mathbf{h}_{k,i}(\mathbf{x}_k, \mathbf{m}_i) + \mathbf{v}_{k,i}, \mathbf{v}_{k,i} \sim N(\mathbf{0}, \mathbf{R}_{k,i})
\end{aligned} \tag{3.77}$$

where  $\mathbf{x}_k$  is the vehicle's pose state vector,  $\tilde{\mathbf{m}}_k = [\mathbf{m}_1^T, \dots, \mathbf{m}_i^T, \dots, \mathbf{m}_N^T]^T$  is the map which consists of all the landmark position states,  $\mathbf{w}_{k-1}$  is the process noise vector that conforms to  $N(\mathbf{0}, \mathbf{Q}_{k-1})$ ,  $\mathbf{z}_{k,i}$  is the measurement vector of feature point  $i$ ,  $\mathbf{h}_{k,i}(\mathbf{x}_k, \mathbf{m}_i)$  is the non-linear function of the vehicle pose states and the landmark position states and  $\mathbf{v}_{k,i}$  is measurement noise vector that conforms to  $N(\mathbf{0}, \mathbf{R}_{k,i})$ .

The stereo visual SLAM measurement equation can be formulated based on 2D [Bayoud et al, 2004; Lategahn et al, 2011; Sazdovski et al, 2011; Alcantarilla et al, 2012] or 3D [Davison and Kita, 2001; Davison and Murray 2002; Paz et al, 2008] image measurements. The stereo visual SLAM with 2D image measurements can use the collinearity equations [Bayoud et al, 2004]

$$\begin{aligned}
x_{L,i} &= x_{L,o} - f_L g_L(\mathbf{X}_k^n, \boldsymbol{\theta}_{c(k)}^n, \mathbf{m}_i^n) + v_{x_{L,i}} \\
y_{L,i} &= y_{L,o} - f_L h_L(\mathbf{X}_k^n, \boldsymbol{\theta}_{c(k)}^n, \mathbf{m}_i^n) + v_{y_{L,i}} \\
x_{R,i} &= x_{R,o} - f_R g_R(\mathbf{X}_k^n + \mathbf{b}_{LR}^n, \boldsymbol{\theta}_{c(k)}^n, \mathbf{m}_i^n) + v_{x_{R,i}} \\
y_{R,i} &= y_{R,o} - f_R h_R(\mathbf{X}_k^n + \mathbf{b}_{LR}^n, \boldsymbol{\theta}_{c(k)}^n, \mathbf{m}_i^n) + v_{y_{R,i}}
\end{aligned} \tag{3.78}$$

where the navigation frame here is the global frame and  $\mathbf{X}_k^n$  and  $\boldsymbol{\theta}_{c(k)}^n$  are the position and orientation vectors of the camera system, respectively. The stereo visual SLAM measurement equation with 3D image measurements can be written as [Paz et al, 2008]

$$\mathbf{I}_{k,i}^c = (\mathbf{C}_{c(k)}^n)^T (\mathbf{m}_i^n - \mathbf{X}_k^n) + \mathbf{v}_{\mathbf{I}_{k,i}^c} \tag{3.79}$$

The 2D method can include the camera calibration parameters and is referred as simultaneous calibration localization and mapping (SCLAM) [Kümmerle et al, 2011].

The 3D method assumes that the cameras are calibrated and is easier to be implemented.

### 3.6.4.5. Visual odometry

The visual odometry algorithm computes the position and orientation change between two consecutive frames. Then by concatenating the successive pose changes, the trajectory of the camera system can be obtained. The matched features between the current frame  $k$  and the previous frame  $k-1$  yield two sets of 3D object positions. The relationship between the pair of the 3D vectors for each object is given by the following 3D rigid body transformation [Matthies and Shafer, 1987; Kelly et al, 2008]:

$$\mathbf{I}_{k,i}^c = \mathbf{C}_{c(k-1)}^{c(k)} \mathbf{I}_{k-1,i}^c + \Delta \mathbf{X}_{k,k-1}^c \quad (3.80)$$

where  $\mathbf{C}_{c(k-1)}^{c(k)}$  and  $\Delta \mathbf{X}_{k,k-1}^c$  are the DCM and position difference respectively.

$\Delta \mathbf{X}_{k,k-1}^c$  and  $\mathbf{C}_{c(k-1)}^{c(k)}$  can be estimated using the weighted least squares method. Another approach to estimate the pose change is to use the 2D image coordinates [Badino et al, 2013]. By substituting (3.75) in (3.80), one obtains:

$$h_{stereo}(\mathbf{x}_{L,k,i}, \mathbf{x}_{R,k,i}) = \mathbf{C}_{c(k-1)}^{c(k)} h_{stereo}(\mathbf{x}_{L,k-1,i}, \mathbf{x}_{R,k-1,i}) + \Delta \mathbf{X}_{k,k-1}^c \quad (3.81)$$

However, this method only applies to the rectified stereo images.

### 3.6.4.6. Accuracy assessment of IBN

The accuracy of an IBN solution can be evaluated using an absolute reference such as the corresponding GNSS-aided inertial navigation solution. The translation and rotation drift rates, i.e. the error with respect to distance travelled, are typically used to describe the performance. Suppose that  $(\mathbf{X}_{IBN,k}^n, \mathbf{C}_{c(k),IBN}^n)$  and  $(\mathbf{X}_{REF,k}^n, \mathbf{C}_{c(k),REF}^n)$  are the IBN and reference solutions, respectively, at epoch  $k$ . Then the percentage translation drift rate is computed by

$$\delta T = \frac{\|\mathbf{X}_{IBN,k}^n - \mathbf{X}_{REF,k}^n\|}{d_{REF,k}} \times 100 \quad (3.82)$$

and the rotational drift rate in  $radm^{-1}$  is

$$\delta \theta = \frac{2 \cos^{-1} \left( \frac{1}{2} \sqrt{\text{tr} \left( (\mathbf{C}_{c(k),IBN}^n)^T \mathbf{C}_{c(k),REF}^n \right) + 1} \right)}{d_{REF,k}} \quad (3.83)$$

where  $d_{REF}$  is the total path length, which is computed as shown in Algorithm 3.1.

Algorithm 3.1 Computation of total path length

$d_{REF,k} = 0$ $\text{for } i = 2 : k \{$ $d_{REF,k} = d_{REF,k} + \ \mathbf{X}_{REF,i}^n - \mathbf{X}_{REF,i-1}^n\ $ $\}$
---

In practice, the translational and rotational drift rates are computed for multiple lengths in the trajectory (e.g. 20m, 40m and so on). Then all the possible translational and rotational drift rates are averaged to obtain the global performance metrics.

## **4. Structureless stereo camera auto-calibration and system calibration**

This chapter is mainly based on the following publication:

Gopaul, Nilesh S.; Wang, Jianguo and Hu, Baoxin (2016): *Camera auto-calibration in GPS/INS/stereo camera integrated kinematic positioning and navigation system*, **Journal of Global Positioning Systems**, 2016, 14:3 <https://doi.org/10.1186/s41445-016-0003-7>, Springer, ISSN 1446-3164.

### **4.1. Introduction**

This chapter presents a two-step camera calibration method for multisensor integrated kinematic positioning and navigation systems with GNSS receivers, IMU and stereo cameras. The stereo camera auto-calibration employs two scale-restraint equations for each consecutive and stereo pair [Gopaul et al, 2016]. In comparison with the collinearity equations, the proposed method is structureless, i.e., free of the object positions. First, a camera auto-calibration is performed to obtain the lens distortion parameters, the up-to-scale baseline length and the relative orientation between the two cameras. Then, the system calibration is introduced to recover the camera lever-arms, and the bore-sight angles with respect to the IMU, and the absolute scale of the camera using the GNSS-aided inertial integrated navigation solution.

This chapter is organized as follows: Section 4.2 describes the proposed method for stereo camera auto-calibration and camera system calibration. The comparison and analysis of the computational complexity of the auto-calibration methods using the collinearity equations and the scale restraint equation through the simulated data is shown

in Section 4.3. Section 4.4 contains the pre-analysis to determine the best scale-restraint equations combination and the expected accuracy of the collinearity equations and the scale restraint equation using the simulated data. Section 4.5 presents lab test results. The chapter ends with a summary and the research contributions in Section 4.6. The field test results and analysis are presented in Section 7.3.

## 4.2. Camera calibration

The assumptions prior to the development are as follows:

- The object points in the scene are stationary w.r.t. the e-frame,
- The raw measurements from the sensors are synchronized,
- The GNSS-aided inertial integrated navigation solution is available,
- The GNSS-aided inertial integrated navigation solution is referenced at the center of the IMU, and
- The left camera is specified as the reference camera of the stereo camera system.

### 4.2.1. Structureless camera auto-calibration

The algorithmic development of the stereo camera auto-calibration begins with the well-known extended collinearity equations (COL). The roll, pitch and heading angles are assumed to be unavailable at this point. Therefore, EOPs are resolved in the n'-frame.

The extended collinearity equations for a stereo system at the current epoch  $k$  is:

$$\begin{aligned}
 x_{L,k,i} &= x_{0,L} - f_L g_L(\mathbf{X}'_{L,k}, \boldsymbol{\theta}'_{c(k)}, \mathbf{X}'_i) + \Delta x_{d,L,k,i} + v_{x_{L,k,i}} \\
 y_{L,k,i} &= y_{0,L} - f_L h_L(\mathbf{X}'_{L,k}, \boldsymbol{\theta}'_{c(k)}, \mathbf{X}'_i) + \Delta y_{d,L,k,i} + v_{y_{L,k,i}} \\
 x_{R,k,i} &= x_{0,R} - f_R g_R(\mathbf{X}'_{L,k} + \mathbf{C}'_{c(k)} \mathbf{b}^c_{LR}, \boldsymbol{\theta}'_{c(k)}, \mathbf{X}'_i, \boldsymbol{\theta}^c_{cR}) + \Delta x_{d,R,k,i} + v_{x_{R,k,i}} \\
 y_{R,k,i} &= y_{0,R} - f_R h_R(\mathbf{X}'_{L,k} + \mathbf{C}'_{c(k)} \mathbf{b}^c_{LR}, \boldsymbol{\theta}'_{c(k)}, \mathbf{X}'_i, \boldsymbol{\theta}^c_{cR}) + \Delta y_{d,R,k,i} + v_{y_{R,k,i}}
 \end{aligned} \tag{4.1}$$

wherein  $L$  and  $R$  denote the left and right cameras, respectively,  $(x_{k,i}, y_{k,i})$  [px] are the 2D coordinates of the image point  $i$ ,  $\mathbf{X}_i^{n'}$  [m] is the position vector of the object point,  $\mathbf{X}_{L,k}^{n'}$  [m] is the camera's perspective center of the left camera,  $\boldsymbol{\theta}_{c(k)}^{n'}$  [rad] is the orientation vector of the left camera,  $(x_0, y_0)$  [px] is the position vector of the principal point,  $f$  [px] is the focal length,  $(\Delta x_{d,k,i}, \Delta y_{d,k,i})$  [px] are the 2D image distortion,  $(v_{x_{k,i}}, v_{y_{k,i}})$  [px] are the measurement noises in x and y directions,  $\mathbf{b}_{LR}^c$  [m] is the baseline vector between two cameras, and  $\boldsymbol{\theta}_{cR}^c$  [rad] is the orientation vector of the right camera w.r.t. the left one. Figure 4.1 shows the image points, the object point and the camera EOP in a stereo camera system.

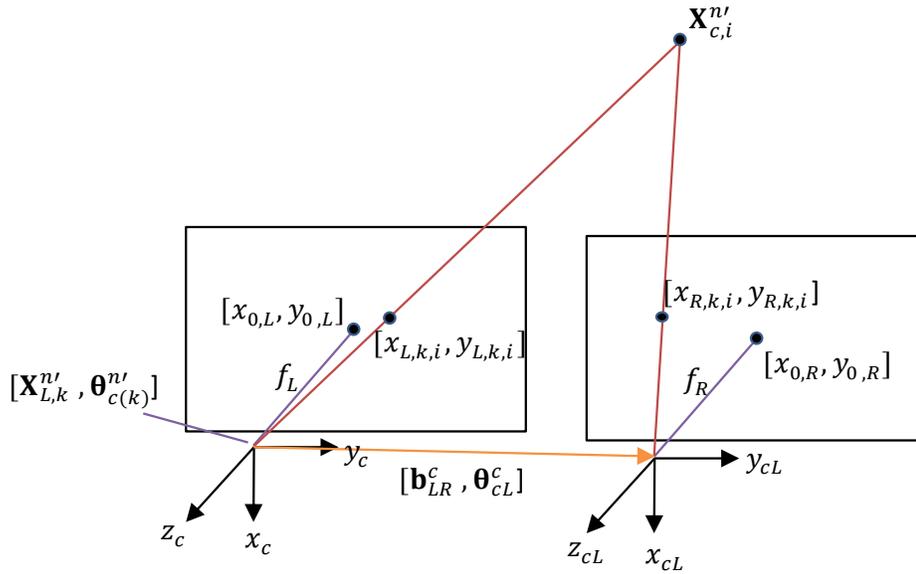


Figure 4.1 Stereo Vision

The auto-calibration image distortion model given by Brown [1971] is as follows

$$\begin{aligned}
x_{d,i} &= -\Delta x_0 - \frac{\bar{x}}{f} \Delta f + \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + p_1(r^2 + 2\bar{x}^2) + 2p_2\bar{x}\bar{y} \\
&\quad - A_1\bar{x} + A_2\bar{y} \\
\Delta y_{d,i} &= -\Delta y_0 - \frac{\bar{y}}{f} \Delta f + \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + 2p_1\bar{x}\bar{y} + p_2(r^2 + 2\bar{y}^2)
\end{aligned} \tag{4.2}$$

where  $r$  is the radial distance from the principal point to the image point ( $r^2 = \bar{x}^2 + \bar{y}^2 = (x_i - x_o)^2 + (y_i - y_o)^2$ ),  $\Delta f$  is the focal length error,  $(\Delta x_0, \Delta y_0)^T$  is the error vector of the principal point position,  $k_i$  and  $p_i$  are the coefficients of the radial and decentering distortions, respectively, and  $A_i$  is the coefficient of the affine deformation ( $i = 1, 2$ ).

Given the definition of  $n^2$ -frame and stationary object points are fixed in the e-frame, it is noted that (4.1) does not account for the transport rate  $\boldsymbol{\omega}_{en'}^{n'}$ . On the surface of the Earth, the transport rate in degrees per meter is approximately  $\tan^{-1}(\frac{1m}{R}) \times \frac{180}{\pi} \approx 9.0 \times 10^{-6}$  [deg/m], where  $R$  is the radius of a spherical Earth. It has been demonstrated that image-based navigation systems can generate orientation solutions with orientation drift rates as low as 0.003[deg/m] [Cvišić et al, 2017]. Given that the transport rate in degrees per meter is atleast 2 to 3 orders of magnitude smaller than the orientation drift rates of modern image-based navigation systems, it can be ignored.

Auto-calibration algorithms suffer from a rank deficiency of order seven (i.e. 3D position, 3D orientation and scale). In order to enable a free-network adjustment, one component of the baseline vector  $\mathbf{b}_{LR}^c$  must be free (i.e. 2 d.o.f.), one of the position  $\mathbf{X}_{L,k}^{n'}$  and one of the orientation parameters in  $\boldsymbol{\theta}_{c(k)}^{n'}$  must be fixed, which resolve the deficiencies due to the missing scale and orientation as part of the adjustment datum. Practically, one can set  $\mathbf{X}_{L,1}^{n'} = \mathbf{0}$  and  $\boldsymbol{\theta}_{c(1)}^{n'} = \mathbf{0}$  at the first epoch. The remaining

parameters, including the stereo baseline vector, the orientation vector of the right camera and the image distortion parameters can be solved in a least-squares bundle-adjustment.

Now, consider the scale restraint equation (SRE) [Bethel, 2003; Ghosh, 2005], which relates the three arbitrary views  $i$ ,  $j$  and  $k$  :

$$\frac{\mathbf{x}_i^{n'} \bullet (\mathbf{x}_i^{n'} \times \mathbf{x}_j^{n'}) \times \mathbf{b}_{ij}^{n'}}{\mathbf{x}_i^{n'} \bullet (\mathbf{x}_i^{n'} \times \mathbf{x}_j^{n'}) \times \mathbf{x}_j^{n'}} - \frac{\mathbf{x}_j^{n'} \bullet (\mathbf{x}_j^{n'} \times \mathbf{x}_k^{n'}) \times \mathbf{b}_{jk}^{n'}}{\mathbf{x}_j^{n'} \bullet (\mathbf{x}_j^{n'} \times \mathbf{x}_k^{n'}) \times \mathbf{x}_k^{n'}} = 0 \quad (4.3)$$

wherein

$$\mathbf{x}_{a=i,j,k}^{n'} = \mathbf{C}_{c,a}^{n'} \begin{pmatrix} x_a - x_o - \Delta x_{d,a} - v_{x_a} \\ y_a - y_o - \Delta y_{d,a} - v_{y_a} \\ -f \end{pmatrix}, \quad (4.4)$$

$\mathbf{b}_{ij}^{n'}$  and  $\mathbf{b}_{jk}^{n'}$  are the baselines [m] between images  $i$  and  $j$ , and between images  $j$  and  $k$ , respectively. For the stereo auto-calibration, the proposed algorithm uses point features matched from four views which are stereo pairs from two consecutive epochs, i.e.,  $\mathbf{x}_{L,k}^c$ ,  $\mathbf{x}_{R,k}^c$ ,  $\mathbf{x}_{L,k-1}^c$  and  $\mathbf{x}_{R,k-1}^c$ . Figure 4.2 (left) depicts the quad-matches.

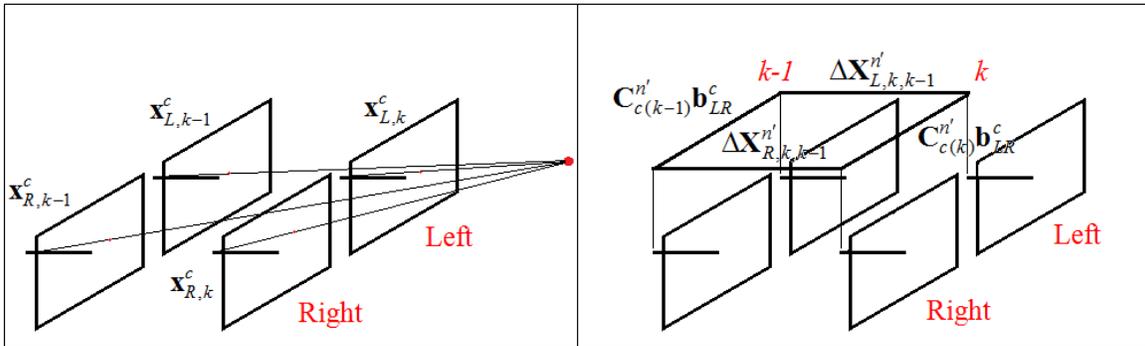


Figure 4.2 Four view feature matching (left) and translational offsets between the four views (right)

Figure 4.2 (right) shows the translational offsets between the four views while the parameters are listed in Table 4.1. Since the scale restraint equation constrains three views, there are four possible combinations in a four-view configuration. The four combinations are listed in Table 4.2.

Table 4.1 Camera auto-calibration parameters (L and R denote the left and right camera)

Parameter	Description
$\Delta f_L, \Delta f_R$	Correction for focal lengths [px]
$\Delta x_{L,0}, \Delta y_{L,0}, \Delta x_{R,0}, \Delta y_{R,0}$	Correction for principal points [px]
$k_{L,1}, k_{L,2}, k_{L,3}, k_{R,1}, k_{R,2}, k_{R,3}$	Radial lens distortion parameters [px <sup>-2</sup> , px <sup>-4</sup> , px <sup>-6</sup> , px <sup>-2</sup> , px <sup>-4</sup> , px <sup>-6</sup> ]
$p_{L,1}, p_{L,2}, p_{R,1}, p_{R,2}$	Decentering distortion parameters [px <sup>-2</sup> , px <sup>-2</sup> , px <sup>-2</sup> , px <sup>-2</sup> ]
$A_{L,1}, A_{L,2}, A_{R,1}, A_{R,2}$	Affine deformation parameters [px <sup>-1</sup> , px <sup>-1</sup> , px <sup>-1</sup> , px <sup>-1</sup> ]
$\mathbf{b}_{LR}^c$	Stereo baseline vector [m]
$\boldsymbol{\theta}_{cR}^c$	Orientation vector of the right camera w.r.t. left camera [rad]
$\Delta \mathbf{X}_{L,k,k-1}^{n'}$	Position difference of the left camera between two consecutive frames [m]
$\boldsymbol{\theta}_{c(k)}^{n'}$	Orientation vector of the left camera [rad]

Table 4.2 Four combinations of the scale restraint equation in four views

Combination	Image $i$	Image $j$	Image $k$
a	$\mathbf{x}_{L,k}^c$	$\mathbf{x}_{R,k}^c$	$\mathbf{x}_{R,k-1}^c$
b	$\mathbf{x}_{L,k}^c$	$\mathbf{x}_{R,k}^c$	$\mathbf{x}_{L,k-1}^c$
c	$\mathbf{x}_{L,k}^c$	$\mathbf{x}_{L,k-1}^c$	$\mathbf{x}_{R,k-1}^c$
d	$\mathbf{x}_{R,k}^c$	$\mathbf{x}_{R,k-1}^c$	$\mathbf{x}_{L,k-1}^c$

In order to relate all four views, two scale restraint equations are required. This leads to the following six possible combinations: (1) {a,b}, (2) {a,c}, (3) {a,d}, (4) {b,c}, (5){b,d}, and (6) {c,d} as shown in Figure 4.3.

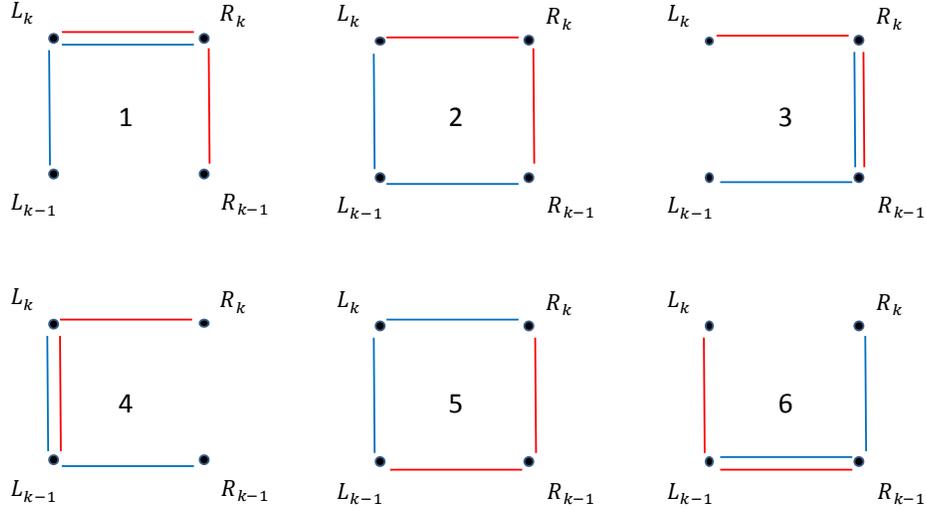


Figure 4.3 The six possible combinations to constrain four views with two SREs

The combination that outputs the minimum trace of the variance-covariance matrix of the parameter vector when performing the auto-calibration can be considered the most accurate and therefore the best solution. Simulated data in a typical land vehicle scenario were used to determine the best combination. The corresponding test results can be found in Section 4.4.1. The test results showed that combination (5) performed the best and the corresponding equations can be expressed as

$$\begin{aligned}
 & \frac{\mathbf{x}_{R,k}^{n'} \bullet (\mathbf{x}_{R,k}^{n'} \times \mathbf{x}_{L,k}^{n'}) \times \mathbf{C}_{c(k)}^{n'} \mathbf{b}_{LR}^c}{\mathbf{x}_{R,k}^{n'} \bullet (\mathbf{x}_{R,k}^{n'} \times \mathbf{x}_{L,k}^{n'}) \times \mathbf{x}_{L,k}^{n'}} - \frac{\mathbf{x}_{L,k}^{n'} \bullet (\mathbf{x}_{L,k}^{n'} \times \mathbf{x}_{L,k-1}^{n'}) \times \Delta \mathbf{X}_{L,k,k-1}^{n'}}{\mathbf{x}_{L,k}^{n'} \bullet (\mathbf{x}_{L,k}^{n'} \times \mathbf{x}_{L,k-1}^{n'}) \times \mathbf{x}_{L,k-1}^{n'}} = 0 \\
 & \frac{\mathbf{x}_{L,k-1}^{n'} \bullet (\mathbf{x}_{L,k-1}^{n'} \times \mathbf{x}_{R,k-1}^{n'}) \times \mathbf{C}_{c(k-1)}^{n'} \mathbf{b}_{LR}^c}{\mathbf{x}_{L,k-1}^{n'} \bullet (\mathbf{x}_{L,k-1}^{n'} \times \mathbf{x}_{R,k-1}^{n'}) \times \mathbf{x}_{R,k-1}^{n'}} - \frac{\mathbf{x}_{R,k-1}^{n'} \bullet (\mathbf{x}_{R,k-1}^{n'} \times \mathbf{x}_{R,k}^{n'}) \times \Delta \mathbf{X}_{R,k,k-1}^{n'}}{\mathbf{x}_{R,k-1}^{n'} \bullet (\mathbf{x}_{R,k-1}^{n'} \times \mathbf{x}_{R,k}^{n'}) \times \mathbf{x}_{R,k}^{n'}} = 0
 \end{aligned}
 \tag{4.5}$$

where  $\mathbf{x}_{L,k}^{n'} = \mathbf{C}_{c(k)}^{n'} \mathbf{x}_{L,k}^c$ ,

$\mathbf{x}_{R,k}^{n'} = \mathbf{C}_{c(k)}^{n'} \mathbf{C}_{cR}^c \mathbf{x}_{R,k}^{cR}$ , and

$\Delta \mathbf{X}_{R,k,k-1}^{n'} = \Delta \mathbf{X}_{L,k,k-1}^{n'} + (\mathbf{C}_{c(k)}^{n'} - \mathbf{C}_{c(k-1)}^{n'}) \mathbf{b}_{LR}^c$ .

In short form, (4.5) can be written as,

$$\begin{aligned}
h_{k,k-1,i}(\mathbf{x}_{L,k}^c, \mathbf{x}_{R,k}^c, \mathbf{x}_{L,k-1}^c, \Delta \mathbf{X}_{L,k,k-1}^{n'}, \boldsymbol{\theta}_{c(k)}^{n'}, \boldsymbol{\theta}_{c(k-1)}^{n'}, \mathbf{b}_{LR}^c, \boldsymbol{\theta}_{cR}^c) &= 0 \\
g_{k,k-1,i}(\mathbf{x}_{L,k}^c, \mathbf{x}_{L,k-1}^c, \mathbf{x}_{R,k-1}^c, \Delta \mathbf{X}_{L,k,k-1}^{n'}, \boldsymbol{\theta}_{c(k)}^{n'}, \boldsymbol{\theta}_{c(k-1)}^{n'}, \mathbf{b}_{LR}^c, \boldsymbol{\theta}_{cR}^c) &= 0
\end{aligned} \tag{4.6}$$

In order to introduce a free-network adjustment, one component of the baseline vector  $\mathbf{b}_{LR}^c$  must be free (i.e. 2 d.o.f.) and one of the orientation parameters in  $\boldsymbol{\theta}_{c(k)}^{n'}$  must be fixed (ideally  $\boldsymbol{\theta}_{c(1)}^{n'} = \mathbf{0}$ ). Since the measurement equation in (4.5) is the implicit form (i.e.,  $\mathbf{h}(\mathbf{x}, \mathbf{z}, \mathbf{v}) = \mathbf{0}$ ), the parameters can be solved by bundle adjustment with the conditional least-squares method with parameters described in Section 3.2.1.

When the objects are close and the stereo baseline is long or both, the stereo overlapping region reduces. To fill the entire image frame, the point features in the non-overlapping areas can be constrained by one SRE equation if the object point can be viewed in 3 images or by the coplanarity equation (CP) between 3 images, which is illustrated in Figure 4.4.

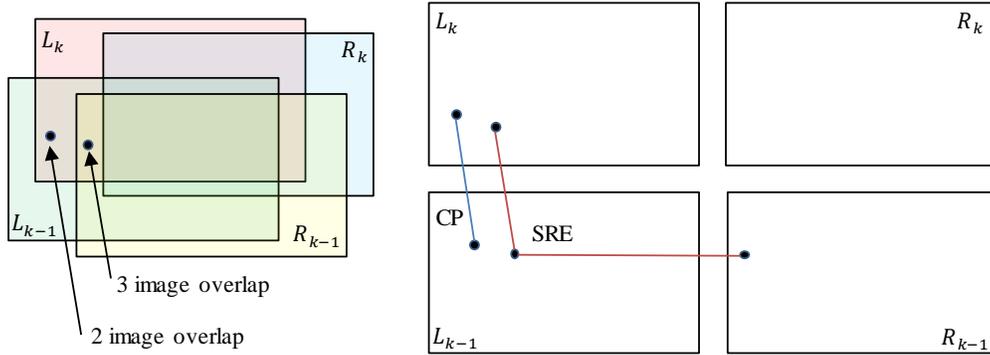


Figure 4.4 Two and three overlapping regions.

#### 4.2.2. Camera system calibration

The camera system calibration is to determine the lever-arm vector  $\mathbf{Ia}_L^b$ , the absolute scale  $s_c$  of the camera and the bore-sight angle vector  $\boldsymbol{\theta}_c^b$ . The relationship between the

integrated GNSS aided-INS solution and left camera can be expressed as the following seven parameter Helmert transformation:

$$\mathbf{X}_{GNSSINS,k}^n = s_c \mathbf{C}_{n'}^n \mathbf{X}_{L,k}^{n'} - \mathbf{C}_{b(k),GNSSINS}^n \mathbf{1a}_L^b \quad (4.7)$$

wherein  $\mathbf{X}_{GNSSINS,k}^n$  and  $\mathbf{C}_{b(k),GNSSINS}^n$  are the integrated GNSS aided-INS position vector and attitude DCM respectively.  $\mathbf{C}_{n'}^n$  is determined by the orientation of the camera system with respect to the n-frame at the first epoch

$$\mathbf{C}_{n'}^n = \mathbf{C}_{c(1)}^n = \mathbf{C}_{b(1),GNSSINS}^n \mathbf{C}_c^b \quad (4.8)$$

where  $\mathbf{C}_c^b$  is the camera-to-body DCM represented by the bore-sight angle vector  $\theta_c^b$ .

Differencing (4.7) between epoch  $k-1$  and epoch  $k$  gives

$$\Delta \mathbf{X}_{GNSSINS,k}^n = s_c \mathbf{C}_{n'}^n \Delta \mathbf{X}_{L,k,k-1}^{n'} - (\mathbf{C}_{b(k),GNSSINS}^n - \mathbf{C}_{b(k-1),GNSSINS}^n) \mathbf{1a}_L^b \quad (4.9)$$

wherein  $\Delta \mathbf{X}_{GNSSINS,k}^n = \mathbf{D}_r^{-1}(\mathbf{r}_{GNSSINS,k}^n - \mathbf{r}_{GNSSINS,k-1}^n)$ . From (4.5), the relationship between

$\mathbf{C}_{b(k),GNSSINS}^n$  and the camera DCM  $\mathbf{C}_{c(k)}^{n'}$  can be written as

$$\mathbf{C}_{b(k),GNSSINS}^n \mathbf{C}_c^b = \mathbf{C}_{n'}^n \mathbf{C}_{c(k)}^{n'}. \quad (4.10)$$

Equations (4.9) and (4.10) equate the GNSS/IMU information ( $\Delta \mathbf{X}_{GNSSINS,k}^n$ ,  $\mathbf{C}_{b(k),GNSSINS}^n$ ) and the up-to-scale auto-calibration  $\Delta \mathbf{X}_{L,k,k-1}^{n'}$  and  $\mathbf{C}_{c(k)}^{n'}$  parameters. All of 7 system calibration parameters can be solved by using the least-squares.

### 4.2.3. Parameter Initialization

The least-squares method requires good approximation of the parameter vector (i.e.  $\mathbf{x}^{(0)}$  in (3.15)) when the measurement equations are non-linear. This can reduce the

effects of non-linearity and prevent the LS solution from diverging. This section presents a practical method for initializing the parameters in the structureless camera auto-calibration and system calibration.

The up-to-scale stereo baseline vector  $\mathbf{b}_{LR}^c$ , the orientation vector  $\boldsymbol{\theta}_{cR}^c$  of the right camera w.r.t. the left camera and the coefficients  $k_i$  of the radial lens distortions can be initialized using the coplanarity equations. The focal length error  $\Delta f$ , the principal point error  $(\Delta x_0, \Delta y_0)$ , the decentering distortion  $p_i$  and the affine deformation parameters  $A_i$  are relatively small and can be ignored during the initialization process. The coplanarity equation is given by

$$\mathbf{b}_{LR}^c \bullet \mathbf{x}_{L,k,i}^c \times \mathbf{C}_{cR}^c \mathbf{x}_{R,k,i}^{cR} = 0 \quad (4.11)$$

where

$$\mathbf{x}_{(a=L,R),k,i}^c = \begin{pmatrix} x_{a,k,i} - x_{0,a} - \bar{x}_{a,k,i}(k_{a,1}r^2 + k_{a,2}r^4 + k_{a,3}r^6 + \dots) - v_{x_{a,k,i}} \\ y_{L,a,i} - y_{0,a} - \bar{x}_{a,k,i}(k_{a,1}r^2 + k_{a,2}r^4 + k_{a,3}r^6 + \dots) - v_{y_{L,a,i}} \\ -f \end{pmatrix}. \quad (4.12)$$

Typically, a stereo system is assembled such that the cameras are near parallel and left-right adjacent to each other. Hence,  $\mathbf{b}_{LR}^c$  is approximately equal to  $(0, b, 0)^T$  where  $b$  is the baseline length. Since (4.11) is rank defect by one, one of the components in  $\mathbf{b}_{LR}^c$  must be set free. Here  $b_{LR,y}^c = b$  is set. Therefore, the parameter vector becomes

$$\mathbf{x} = (b_{LR,x}^c, b_{LR,z}^c, (\boldsymbol{\theta}_{cR}^c)^T, k_{L,1}, k_{L,2}, \dots, k_{R,1}, k_{R,2}, \dots)^T. \quad (4.13)$$

It is recommended to run the adjustment at two or three epochs in order to check for consistency.

As mentioned in the Section 4.2.2, the approximate values of the lever-arm vector  $\mathbf{la}_L^b$  can be obtained directly by external means, for example, a measuring tape. However, the bore-sight angles  $\theta_c^b$  can only be obtained indirectly. One way to determine these angles is to compare the integrated GNSS/IMU solution and the camera orientation solutions for a given time interval. Since only the camera orientation information is required and the scale is not important at this stage, the coplanarity equation can be employed. Consider the coplanarity equation for the left (reference) camera at two consecutive epochs  $k-1$  and  $k$ :

$$\Delta \mathbf{X}_{L,k,k-1}^c \bullet \mathbf{x}_{L,k,i}^c \times \mathbf{C}_{c(k-1)}^{c(k)} \mathbf{x}_{L,k-1,i}^c = 0 \quad (4.14)$$

where the image vectors  $\mathbf{x}_{L,k,i}^c$  and  $\mathbf{x}_{L,k-1,i}^c$  have been corrected for the radial distortion obtained from (4.11). Similar to (4.12), (4.14) is also rank defect by one, thus one of the components in  $\Delta \mathbf{X}_{L,k,k-1}^c$  must be set as a free parameter. For example, set  $\Delta Z_{L,k,k-1}^c$  to

1. Thus, parameter vector becomes

$$\mathbf{x} = (\Delta X_{L,k,k-1}^c, \Delta Y_{L,k,k-1}^c, (\theta_{c(k-1)}^{c(k)})^T)^T \quad (4.15)$$

Next,  $\mathbf{C}_{c(k)}^{n'}$  can be obtained by chaining the  $\mathbf{C}_{c(k-1)}^{c(k)}$  solution from (4.14) for epochs up to and including  $k$ :

$$\mathbf{C}_{c(k)}^{n'} = \mathbf{C}_{c(k-1)}^{n'} (\mathbf{C}_{c(k-1)}^{c(k)})^T \quad (4.16)$$

with  $\mathbf{C}_{c(1)}^{n'} = \mathbf{C}_{c(1)}^n = \mathbf{I}$ . Then, compare the GNSS/IMU and the camera orientations to obtain the bore-sight angles. The substitution of (4.8) in (4.10) gives

$$\mathbf{C}_{b(k),GNSSINS}^n \mathbf{C}_c^b = \mathbf{C}_{b(1),GNSSINS}^n \mathbf{C}_c^b \mathbf{C}_{c(k)}^{n'} \quad (4.17)$$

which relates the GNSS/INS and the reference camera orientation solution. The boresight angles  $\theta_c^b$  in (4.17) can be obtained via the linearized least-squares method.

### 4.3. Computational complexity

This section compares the number of parameters and the number of floating point operations (flops) required in a least squares bundle adjustment between the stereo auto-calibration algorithms based on collinearity equations (COL) (Equation (4.1)) and the scale restraint equations (SRE) (Equation (4.5)).

Table 4.3 shows the number of parameters based on COL and SRE auto-calibration approaches, in which  $n_x$  is the number of the stereo image frames and  $n_{obj}$  is the number of the visible 3D objects. One component in the stereo baseline vector is left free (i.e., only two stereo baseline parameters). The number of the camera position and orientation parameters is  $6(n_x - 1)$  in total because the first camera position and orientation is fixed (practically they are set to zero). The advantage of employing SRE is to have the number of the estimated parameters far less than the one with COL.

Table 4.3 Dimension of the parameter vector (COL vs. SRE)

	<b>COL</b>	<b>SRE</b>
Number of image frames	$n_x$	$n_x$
Number of observed objects	$n_{obj}$	$n_{obj}$
Focal length, principal point	$2 \times 3$	$2 \times 3$
Lens distortion ( $k_1, k_2, k_3$ )	$2 \times 3$	$2 \times 3$
Stereo baseline and relative orientation	$2 + 3$	$2 + 3$
Camera position and orientation	$6(n_x - 1)$	$6(n_x - 1)$
Object parameters	$3n_{obj}$	0
Total	$11 + 6n_x + 3n_{obj}$	$11 + 6n_x$

The flop count is the total number of multiplication and addition operations required to obtain a least squares (LS) solution. Table 4.4 shows the flops counts for the common matrix operations used in least squares.

Table 4.4 Matrix operations and the corresponding number of flops

<b>Operation</b>	<b>Description</b>	<b>Flops</b>
$\mathbf{A}_{m \times n} \mathbf{B}_{n \times l}$	Matrix-Matrix Product	$2mnl - ml$
$\mathbf{A}_{m \times n} \mathbf{b}_{n \times 1}$	Matrix-Vector Product	$2mn - m$
$\mathbf{A}_{n \times n}^{-1}$	Symmetric Matrix Inversion	$n^3 + n^2 + n$
$\sum_{i=1}^k \mathbf{A}_{m \times n, i}$	Matrix Summation	$mn(k-1)$
$\sum_{i=1}^k \mathbf{b}_{m \times 1, i}$	Vector Summation	$m(k-1)$

The factors considered in the analysis are the number of the matched stereo points (i.e. number of measurements), the number of the image frames, the number of the objects in view and the overlapping percentage between the consecutive frames. The overlap percentage encompasses the frame rate, the velocity and the attitude rate of the camera system. Furthermore, COL system has its measurements as the functions of the unknowns in the explicit form (i.e.  $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$ ) and is solved by applying the parametric LS adjustment method while the SRE system has its measurements as the functions of the unknowns in the implicit form (i.e.,  $\mathbf{h}(\mathbf{x}, \mathbf{z}, \mathbf{v}) = \mathbf{0}$ ) and is solved by applying the conditional LS adjustment with parameters. The flop counts between the two will be different under a given number of the measurements and parameters.

To simplify the analysis, the number of the processed image frames is kept constant (here it is set to 92 and the same number is used the simulation tests in Section 4.4).

Furthermore, the number of the measurements and objects and the overlapping percentage are assumed to have the following predictive relationship

$$m' = \frac{m}{n_{obj}} = \frac{1}{1-p} \quad (4.18)$$

where  $m'$  is the average number of the matched stereo pairs per object,  $m$  is the total number of the matched stereo pairs and  $p$  is the average overlapping percentage. For instance, if  $p = 75\%$  then  $m' = 4$ , which means that, on average, an object is viewed on 4 images. By keeping the average overlap percentage constant, the number of the objects in the system can be predicted with a given number of the stereo points. The number of the measurements and parameters in the LS are now known. Therefore, the flop count can be predicted. Under the assumption that the measurement variance-covariance matrix  $\mathbf{R}$  is homoscedastic, i.e.  $\mathbf{R} = \sigma^2 \mathbf{I}$ , the least-squares bundle adjustment solution for COL is given as follows

$$\begin{aligned} \mathbf{N} &= \sum_{i=1}^m (\mathbf{H}_{x,i}^T \mathbf{H}_{x,i}) \\ \mathbf{P} &= \mathbf{N}^{-1} \\ \delta \hat{\mathbf{x}} &= \mathbf{P} \sum_{i=1}^m (\mathbf{H}_{x,i}^T \delta \mathbf{z}_i) \end{aligned} \quad (4.19)$$

Since there are 4 equations per stereo pair (Equation (4.1)), the dimension of  $\mathbf{H}_{x,i}$  is  $4 \times [11 + 6(1) + 3(1)] = 4 \times 26$ . Furthermore, the dimension of  $\delta \hat{\mathbf{x}}$  is  $n = 11 + 6n_x + 3n_m$  (Table 4.3). The total flops count required for computation in (4.19) is summarized in Table 4.5.

Table 4.5 Flops count required for computing COL

Operation	Flops
$\mathbf{N} = \sum_{i=1}^m (\mathbf{H}_{x,i}^T \mathbf{H}_{x,i})$	$[(2(26 \times 4 \times 26) - 26 \times 26) + 26 \times 26]m + 26 \times 26(m-1)$ $= 6084m - 676$
$\mathbf{P} = \mathbf{N}^{-1}$	$n^3 + n^2 + n$
$\mathbf{l} = \sum_{i=1}^m (\mathbf{H}_{x,i}^T \delta \mathbf{z}_i)$	$[(2(26 \times 4) - 26) + 26]m + 26(m-1)$ $= 234m - 26$
$\delta \hat{\mathbf{x}} = \mathbf{P} \mathbf{l}$	$2n^2 - n$
Total	$n^3 + 3n^2 + 6318m - 702$

The least-squares bundle adjustment solution for SRE can be written as

$$\begin{aligned}
 \mathbf{N} &= \sum_{i=1}^{n_{obj}} (\mathbf{H}_{x,i}^T (\mathbf{H}_{z,i} \mathbf{H}_{z,i}^T)^{-1} \mathbf{H}_{x,i}) \\
 \mathbf{P} &= \mathbf{N}^{-1} \\
 \delta \hat{\mathbf{x}} &= \mathbf{P} \sum_{i=1}^{n_{obj}} (\mathbf{H}_{x,i}^T (\mathbf{H}_{z,i} \mathbf{H}_{z,i}^T)^{-1} \delta \mathbf{z}_i)
 \end{aligned} \tag{4.20}$$

The dimension of  $\delta \mathbf{z}_i$  in (4.20) varies as it depends on the number of the matched stereo pairs for a specific object. For example, if an object is viewed in three consecutive stereo pairs, the SRE equations will be

$$\begin{aligned}
 f_{k,k-1,i}(\mathbf{x}_{L,k}^c, \mathbf{x}_{R,k}^c, \mathbf{x}_{L,k-1}^c, \dots) &= 0 \\
 g_{k,k-1,i}(\mathbf{x}_{L,k}^c, \mathbf{x}_{L,k-1}^c, \mathbf{x}_{R,k-1}^c, \dots) &= 0 \\
 f_{k-1,k-2,i}(\mathbf{x}_{L,k-1}^c, \mathbf{x}_{R,k-1}^c, \mathbf{x}_{L,k-2}^c, \dots) &= 0 \\
 g_{k-1,k-2,i}(\mathbf{x}_{L,k-1}^c, \mathbf{x}_{L,k-2}^c, \mathbf{x}_{R,k-2}^c, \dots) &= 0
 \end{aligned} \tag{4.21}$$

Table 4.6 Flops count required for computing SRE

Operation	Flops
$\mathbf{M} = (\mathbf{H}_{z,i} \mathbf{H}_{z,i}^T), n_{obj}$ times	$[2(2m')(4m' + 4)(2m') - (2m')(2m')]n_{obj}$ $= (32m'^3 + 28m'^2)n_{obj}$
$\mathbf{M}^{-1}, n_{obj}$ times	$[(2m')^3 + (2m')^2 + (2m')]n_{obj}$ $= (8m'^3 + 4m'^2 + 2m')n_{obj}$
$\mathbf{H}_{x,i}^T \mathbf{M}^{-1}, n_{obj}$ times	$[2(6m' + 17)(2m')(2m') - (6m' + 17)(2m')]n_{obj}$ $= (48m'^3 + 124m'^2 - 34m')n_{obj}$
$[\mathbf{H}_{x,i}^T \mathbf{M}^{-1}] \mathbf{H}_{x,i}, n_{obj}$ times	$[2(6m' + 17)(2m')(6m' + 17) - (6m' + 17)(6m' + 17)]n_{obj}$ $= (144m'^3 + 780m'^2 + 952m' - 289)n_{obj}$
$\mathbf{N} = \sum_{i=1}^{n_{obj}} (\mathbf{H}_{x,i}^T (\mathbf{H}_{z,i} \mathbf{H}_{z,i}^T)^{-1} \mathbf{H}_{x,i})$	$[(6m' + 17)(6m' + 17)](n_{obj} - 1)$ $= (36m'^2 + 204m' + 289)(n_{obj} - 1)$
$\mathbf{P} = \mathbf{N}^{-1}$	$n^3 + n^2 + n$
$[\mathbf{H}_{x,i}^T \mathbf{M}^{-1}] \delta \mathbf{z}_i, n_{obj}$ times	$[2(6m' + 17)(2m') - (6m' + 17)]n_{obj}$ $= (24m'^2 + 62m' - 17)n_{obj}$
$\mathbf{l} = \sum_{i=1}^{n_{obj}} (\mathbf{H}_{x,i}^T (\mathbf{H}_{z,i} \mathbf{H}_{z,i}^T)^{-1} \delta \mathbf{z}_i)$	$6m' + 17)(n_{lm} - 1)$
$\tilde{\delta \mathbf{x}} = \mathbf{P} \mathbf{l}$	$2n^2 - n$
<i>Total</i>	$n^3 + 3n^2 + (232m'^3 + 1202m'^2 + 1192m')n_{obj} \dots$ $- (36m'^2 + 210m' + 306)$ $= n^3 + 3n^2 + (232m'^2 + 1202m' + 1192)m \dots$ $- (36m'^2 + 210m' + 306)$

If the average number of the matched stereo pairs per object is  $m'$ , the dimension of  $\delta \mathbf{z}_i$  is  $2m'$ , the dimension of  $\mathbf{H}_{z,i}$  is  $2m' \times (4m' + 4)$ , and the dimension of  $\mathbf{H}_{x,i}$  is  $2m' \times (6m' + n_{cal}) = 2m' \times (6m' + 17)$ , where  $n_{cal}$  corresponds to the number of the calibration parameters (the focal length, the principal point, the image distortion parameters, the stereo baseline and the relative orientation in Table 4.3). Furthermore, the dimension of  $\tilde{\delta \mathbf{x}}$  is  $n = 11 + 6n_x$  as shown in Table 4.3. The total flops count required to

solve (4.20) is summarized in. Figure 4.5 shows the number of flops vs. number of stereo points required in COL and SRE with the overlapping percentages of 70%, 80% and 90%.

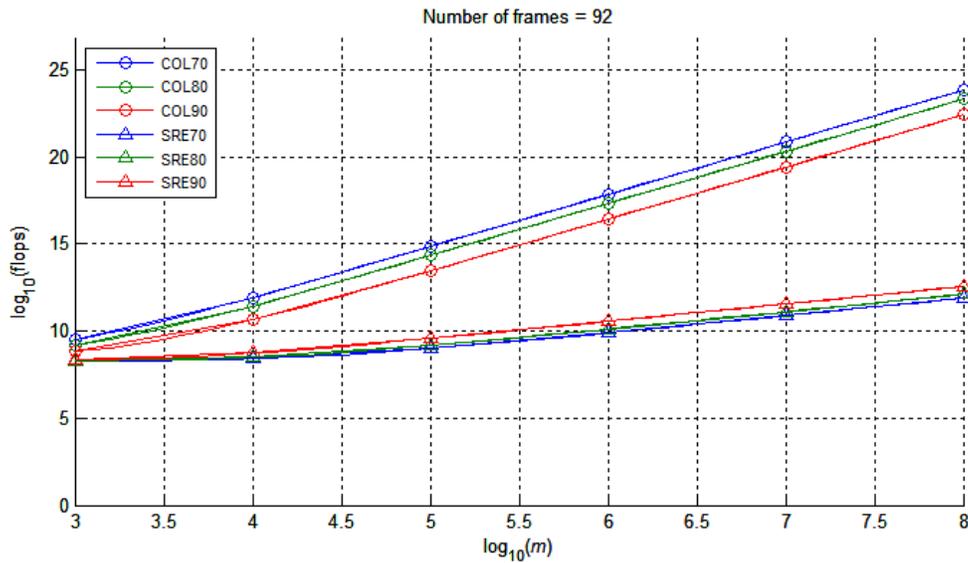


Figure 4.5 Flops vs. number of stereo points ( $m$ )

As expected, the plot shows that COL uses more flops than SRE does. As the overlap percentage increases, the number of the flops in COL decreases because the number of the matched stereo pairs per object becomes larger. Therefore, given the same number of the measurements, the number of the object parameters becomes smaller. As the overlap percentage increases, the number of the flops in SRE increases because more matrix inversion operations are needed in the conditional LS adjustment with parameters.

#### 4.4. Pre-analysis using the simulated data

This section presents the results from the pre-analysis of COL and SRE auto-calibration algorithms via least-squares bundle adjustment. Pre-analysis is typically performed in geodetic control network design in order to predict the achievable accuracy

of the estimated parameters and appropriately adjust the network design or so. Here, the pre-analysis is applied to (a) determine the best SRE combination described in Figure 4.3 in Section 4.2.1 and (b) compare the performance of COL and SRE auto-calibration algorithms in terms of their accuracy and computation complexity. The simulated data were used for the following reasons: (a) they contain no correspondence errors, (b) the measurement model is exact and does not contain un-modelled errors, and (c) it provides flexibility in terms of number of measurements for the analysis.

A typical land vehicle trajectory was simulated for the tests, i.e., it consisted of large horizontal motion and heading variation. Figure 4.6 (left) shows the vehicle's trajectory and the landmarks. The vehicle's height and attitude profiles are given in Figure 4.6 (right).

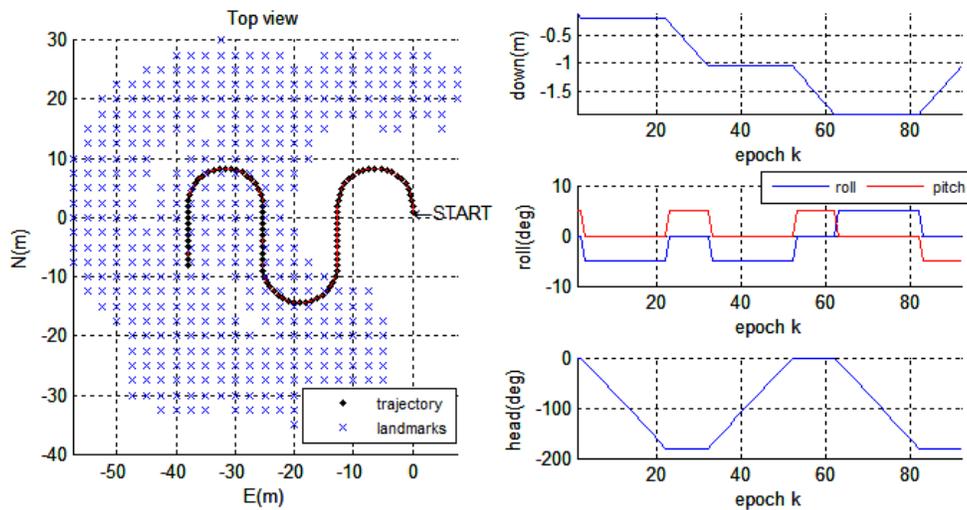


Figure 4.6 The 2D trajectory, landmarks (left) and the height, roll, pitch and heading profiles (right)

The camera resolution and its field of view (FOV) were set to 640x480 pixels and 50 degrees (equivalent to 686.2 pixels), respectively. The baseline length between the two cameras was 0.65m. Algorithm 4.1 below describes the simulation parameters. The

number of the epochs is 92. To simulate the urban scenario, the landmarks between the ranges 15m and 25m were selected to be in the view of the camera. Furthermore, the measurement noise was set to zero mean Gaussian noise with the standard deviation of  $\pm 0.5\text{px}$ . The initial value for each camera calibration parameter was set to zero except the baseline component  $b_{LR,y}^c$  as the free parameter and equal to 0.65m.

#### Algorithm 4.1 Simulation Algorithm

1. Simulate trajectory (i.e. position & attitude)
2. Simulate landmark position (i.e. map)
3. For each position and attitude
  - a. Find landmarks in view
  - b. Find landmarks with ranges [15,25] m
  - c. For each landmark
    - i. Get image coordinates
    - ii. Add lens distortion
    - iii. Add measurement noise ( $0.0 \pm 0.5\text{px}$ )

Figure 4.7 shows a 3D view of the trajectory, landmarks in view and the corresponding left and right images at epoch 46.

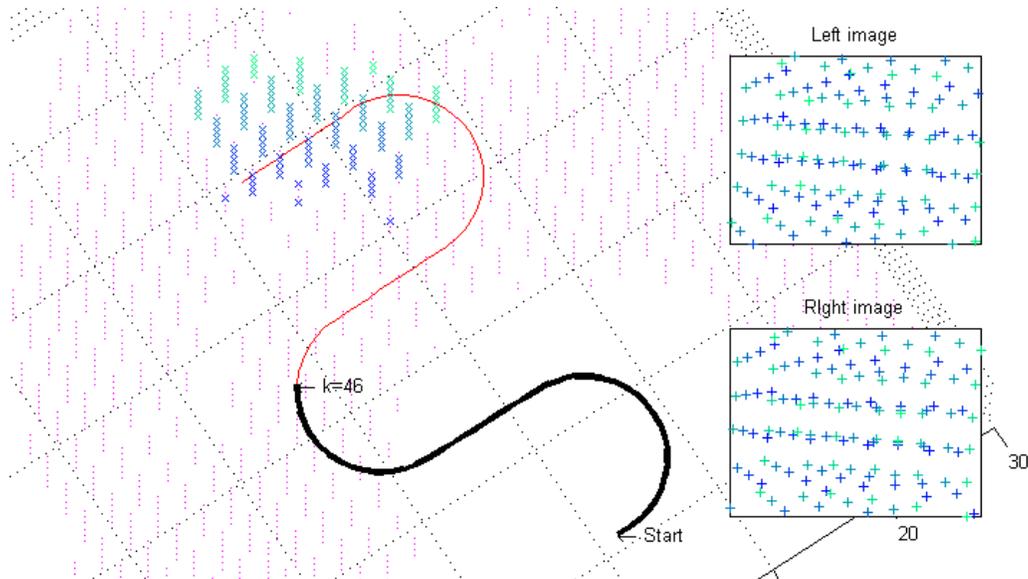


Figure 4.7 3D trajectory, landmarks in view, left image and right image at epoch 46

#### 4.4.1. SRE combination test

This section determines the best SRE combination as described in Figure 4.3. For each combination, the least-squares bundle adjustment SRE solution was computed. Table 4.7 lists the parameters and gives the size of the parameter vector. The number of stereo points per frame is shown Figure 4.8, in which the total number of stereo pair measurements is 17074 . Then, the trace of the parameter variance-covariance matrix was computed for each of the SRE combination. Finally, the results are presented in Table 4.8, from which the Combination 5 performed the best with the minimum trace.

Table 4.7 SRE stereo auto-calibration parameter

Parameter	Size
Image distortion (6 parameter model: $\Delta f, \Delta x_0, \Delta y_0, k_1, k_2, k_3$ )	2×6
Stereo baseline and relative orientation (one baseline component is fixed)	2+3
Exterior Orientation (one EO parameter is fixed)	6×(92-1)
<i>Total parameter vector size</i>	<i>563</i>

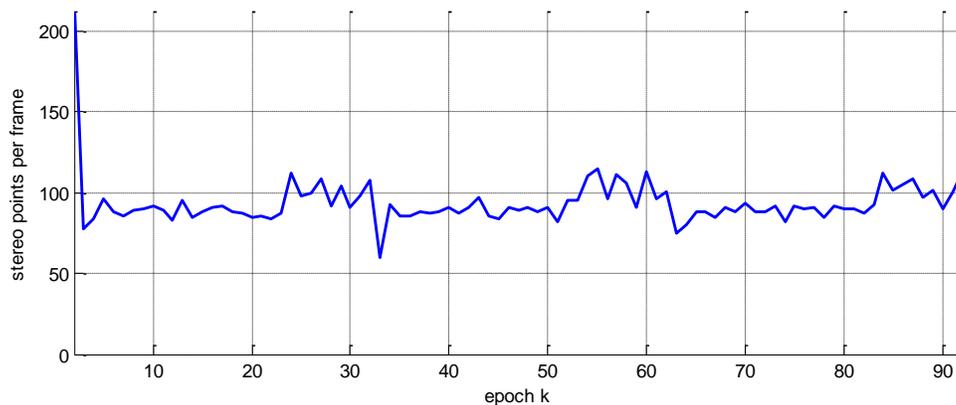


Figure 4.8 Number of stereo points per frame vs. epoch

Table 4.8 Square-root of the trace of the parameter variance-covariance matrix for each of the SRE combination shown in Figure 4.3

SRE Combination Number	SRE Equation 1 Image Points	SRE Equation 2 Image Points	$tr(\mathbf{P})$
1	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c$	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{R,k}^c \ \mathbf{x}_{L,k-1}^c$	8.2305
2	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c$	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{L,k-1}^c \ \mathbf{x}_{R,k-1}^c$	7.7353
3	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c$	$\mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c \ \mathbf{x}_{L,k-1}^c$	55.211
4	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{R,k}^c \ \mathbf{x}_{L,k-1}^c$	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{L,k-1}^c \ \mathbf{x}_{R,k-1}^c$	74.896
5	$\mathbf{x}_{L,k}^c \ \mathbf{x}_{R,k}^c \ \mathbf{x}_{L,k-1}^c$	$\mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c \ \mathbf{x}_{L,k-1}^c$	6.1466
6	$\mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c \ \mathbf{x}_{L,k-1}^c$	$\mathbf{x}_{R,k}^c \ \mathbf{x}_{R,k-1}^c \ \mathbf{x}_{L,k-1}^c$	56.589

#### 4.4.2. Comparison of COL and SRE auto-calibration algorithms

The accuracy analysis on COL and SRE algorithms is presented in this section. Auto-calibration results, one from COL and two from SRE (as SRE1 and SRE2), are presented here. The estimates from COL and SRE1 were obtained using the same number of the measurements, i.e., 17074 stereo points ( $m$ ) as SRE2 used 4.5 times more measurements. The average overlap percentage for all three cases was 74%. Figure 4.9 shows the number of stereo points per frame and the number of the accumulated stereo points.

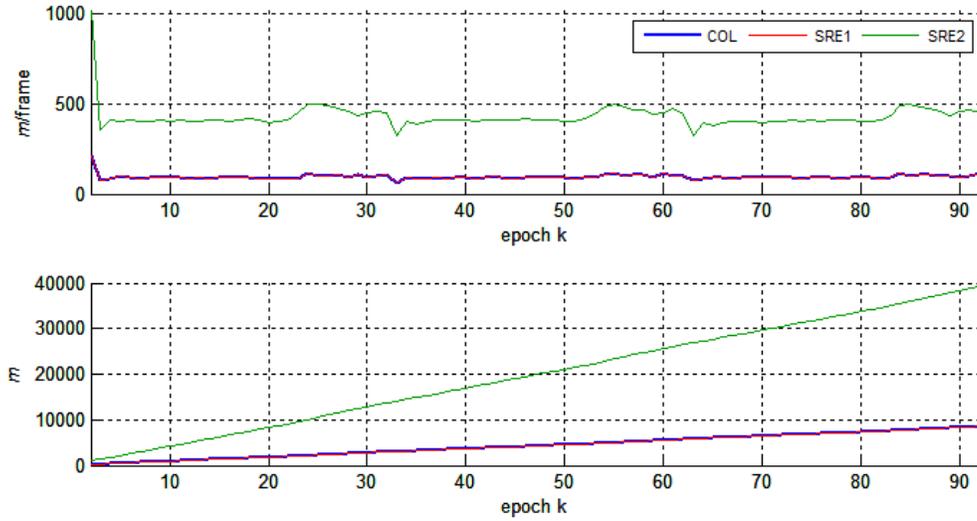


Figure 4.9 The number of the stereo points per frame (top) and the number of the cumulative stereo points (bottom).

Figure 4.10 shows the standard deviations of the estimated focal length and the coordinates of the principal point obtained from the three cases w.r.t. the number of the epochs.

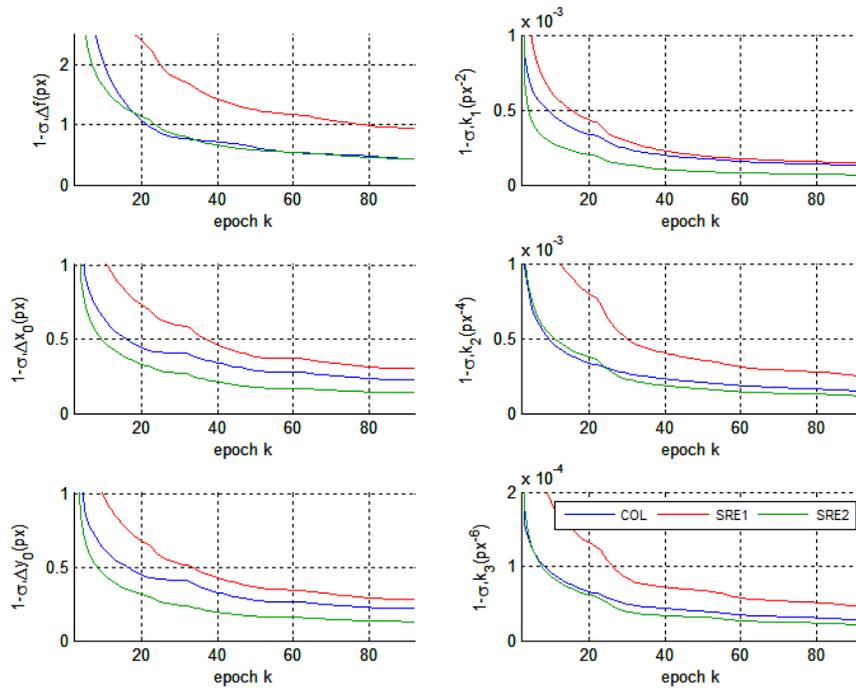


Figure 4.10 The standard deviations of  $\Delta f_L$ ,  $\Delta x_{L,0}$ ,  $\Delta y_{L,0}$ ,  $k_{L,1}$ ,  $k_{L,2}$  and  $k_{L,3}$  for the left camera

The results showed that, with the same given number of measurements, SRE1 performed worse than COL did. However, SRE2 required 4.5 times more measurements to reach the equivalent or better performance than the COL algorithm did. Figure 4.11 shows the standard deviations w.r.t. the number of flops. The results showed that both SRE1 and SRE2 required less computation resources to achieve the same level of the accuracy as COL did.

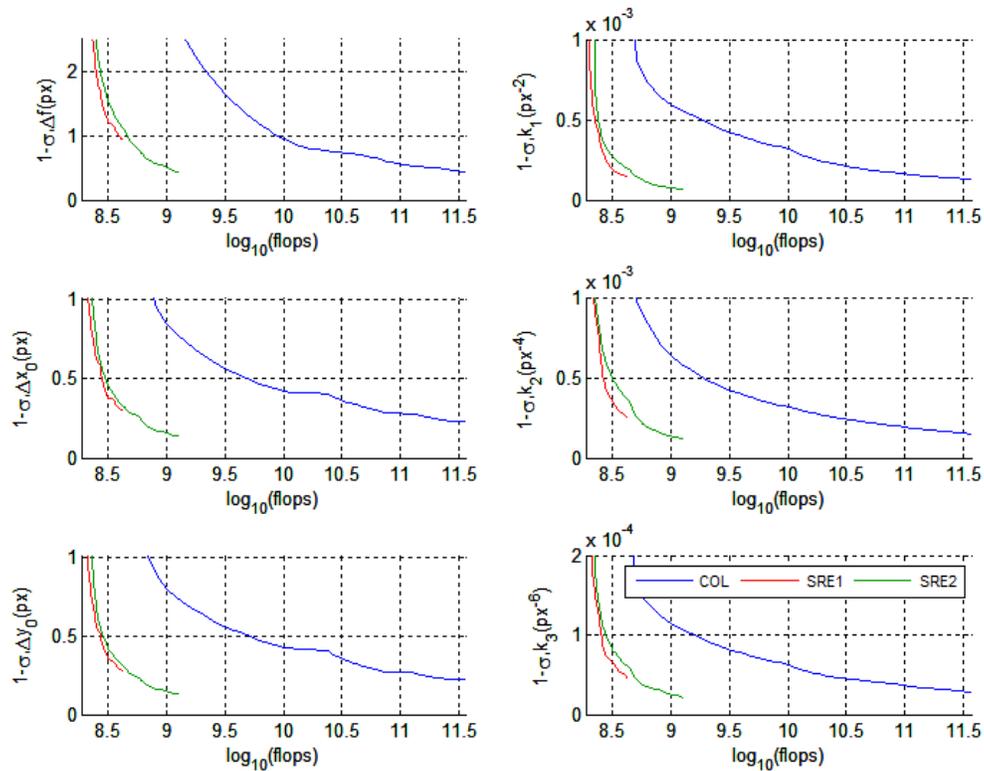


Figure 4.11 The standard deviations of  $\Delta f_L$ ,  $\Delta x_{L,0}$ ,  $\Delta y_{L,0}$ ,  $k_{L,1}$ ,  $k_{L,2}$  and  $k_{L,3}$  for the left camera as the function of flops

The final bundle adjustment solution from COL, SRE1 and SRE2 are presented next. Table 4.9, Table 4.10 and Table 4.11 show the true values, estimates and their standard deviations for the left, right and relative camera orientation calibration parameters, respectively. The results showed that the accuracies of the focal length error from COL

and SRE2 were similar and better than the one from SRE1. SRE2 delivered the smallest principal point error, but SRE1 behaved the worst. The radial distortion coefficients from COL and SRE1 were similar. However, SRE2 delivered the best results.

Table 4.9 Left camera lens distortion parameters

Parameter	True value	COL		SRE1		SRE2	
		mean	stdev	mean	stdev	mean	stdev
$\Delta f$ (px)	-2	-1.825	0.438	-2.713	0.875	-2.318	0.417
$\Delta x_0$ (px)	2.5	2.393	0.230	2.665	0.309	2.432	0.142
$\Delta y_0$ (px)	-3	-3.232	0.219	-2.851	0.266	-3.061	0.122
$k_1$ (px <sup>-2</sup> )	5.0e <sup>-7</sup>	5.03e <sup>-07</sup>	6.47e <sup>-09</sup>	5.07e <sup>-07</sup>	6.97e <sup>-09</sup>	5.06e <sup>-07</sup>	3.21e <sup>-09</sup>
$k_2$ (px <sup>-4</sup> )	4.0e <sup>-13</sup>	7.16e <sup>-14</sup>	1.03e <sup>-13</sup>	6.49e <sup>-13</sup>	1.04e <sup>-13</sup>	3.92e <sup>-13</sup>	4.75e <sup>-14</sup>
$k_3$ (px <sup>-6</sup> )	4.5e <sup>-19</sup>	1.78e <sup>-18</sup>	4.86e <sup>-19</sup>	4.89e <sup>-18</sup>	4.89e <sup>-19</sup>	3.72e <sup>-18</sup>	2.20e <sup>-19</sup>

Table 4.10 Right camera lens distortion parameters

Parameter	True value	COL		SRE1		SRE2	
		mean	stdev	mean	stdev	mean	stdev
$\Delta f$ (px)	+2	2.082	0.441	2.534	0.884	2.078	0.421
$\Delta x_0$ (px)	-2	-2.003	0.226	-2.250	0.319	-2.029	0.146
$\Delta y_0$ (px)	1	0.391	0.217	1.217	0.243	1.233	0.112
$k_1$ (px <sup>-2</sup> )	5.0e <sup>-07</sup>	5.08e <sup>-07</sup>	6.72e <sup>-09</sup>	5.04e <sup>-07</sup>	8.02e <sup>-09</sup>	5.06e <sup>-07</sup>	3.76e <sup>-09</sup>
$k_2$ (px <sup>-4</sup> )	4.0e <sup>-13</sup>	2.85e <sup>-13</sup>	1.08e <sup>-13</sup>	2.38e <sup>-13</sup>	1.28e <sup>-13</sup>	3.60e <sup>-13</sup>	6.03e <sup>-14</sup>
$k_3$ (px <sup>-6</sup> )	4.5e <sup>-19</sup>	9.01e <sup>-19</sup>	5.21e <sup>-19</sup>	5.89e <sup>-18</sup>	6.22e <sup>-19</sup>	5.82e <sup>-18</sup>	2.93e <sup>-19</sup>

The relative orientation parameters from SRE2 were the best overall as the ones from COL and SRE1 were similar to each other. The total number of the used stereo points, the number of parameters and the flop count are shown in Table 4.12 for each of the auto-calibration algorithms. COL and SRE1 employed the same number of measurements, as SRE2 used 4.5 times more measurements. SRE1 and SRE2 estimated the same number parameters, but COL estimated 7854 more parameters. Even though, SRE2 processed more measurements than COL, it still used 1000 times less flops and achieved a higher accuracy.

Table 4.11 Relative orientation of right camera w.r.t left camera (†Free parameter)

Parameter	True value	COL		SRE1		SRE2	
		mean	stdev	mean	stdev	mean	stdev
$b_{LR,x}^c$ (m)	0.01	0.011	0.001	0.007	0.002	0.007	0.001
$b_{LR,y}^c$ (m) †	0.65	0.650	-	0.650	-	0.650	-
$b_{LR,y}^c$ (m)	-0.01	-0.013	0.004	-0.013	0.004	-0.012	0.002
$\theta_{cR,x}^c$ (deg)	-0.25	-0.257	0.007	-0.266	0.008	-0.260	0.004
$\theta_{cR,y}^c$ (deg)	0.5	0.504	0.008	0.500	0.014	0.494	0.006
$\theta_{cR,z}^c$ (deg)	0	0.002	0.002	-0.003	0.002	-0.002	0.001

Table 4.12 Number of points and parameters

	COL	SRE1	SRE2
Number of stereo points	17074	17074	77945
Number of parameters	8417	563	563
$\log_{10}$ (flops)	12.4	8.8	9.4

## 4.5. Laboratory tests and results

This section presents the calibration results of a stereo camera system in a laboratory setting. The first set of tests determined the number of the image distortion terms through trial and error. The second test compared the performance of SRE and COL auto-calibration algorithms.

### 4.5.1. System description

The stereo camera system consisted of two Point Grey Flea3 cameras separated by a baseline of 0.65cm long as shown in Figure 4.12 (left). The camera resolution and the FOV were set to 640x480 pixels and 50 degrees (equivalent to 686.2 pixels), respectively. The size of each pixel is 4.65  $\mu\text{m}$  [Point Grey Research Inc, 2011]. An 8x12

checkerboard with the dimensions of 90x60cm (Figure 4.12 (right)) was used as the reference target field.



Figure 4.12 Stereo Camera System with IMU (Left). 90x60cm Checkerboard (right)

17 stereo pairs of images, 19 left-only images and 20 right-only images were captured. The left-only and right-only images were taken to fill the entire image format. The corners of the checkerboard were extracted using the Camera Calibration Toolbox in Matlab [Bouquet, 2015]. The total number of the extracted points was 5621. Figure 4.13 shows the extracted image measurements.

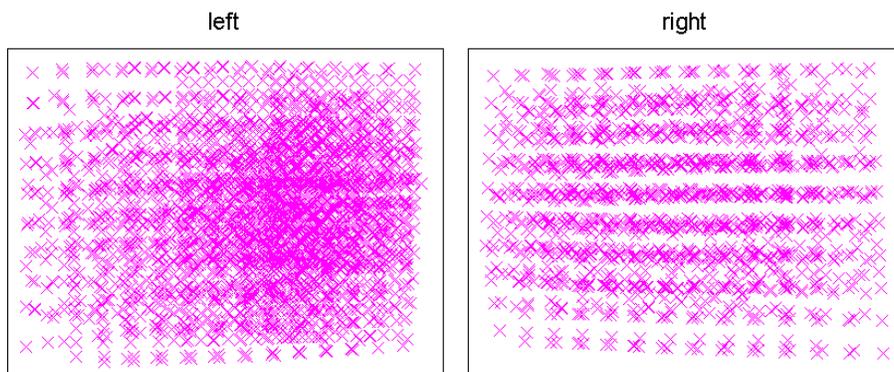


Figure 4.13 Image measurements for the calibration

The minimum, average and maximum distances from the mid-point of the stereo system to the center of the checkerboard were 1.2m, 1.6m and 2.5m, respectively. In

order to obtain a reliable geometry, the image network consisted of orthogonal roll angles and highly convergent images.

#### 4.5.2. Determination of lens distortion parameters

In this section the significant image distortion terms are determined experimentally. The calibration is performed with various sets of image distortion coefficients. Then, the most accurate combination is chosen as the *correct* one. Table 4.13 lists the combinations specifically employed in the completed tests.

Table 4.13 Image distortion coefficients combinations

Combination ID	Combination
1	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2, k_3, k_4, k_5, p_1, p_2, A_1, A_2$
2	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2, k_3, k_4, k_5, A_1, A_2$
3	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2, k_3, k_4, k_5, p_1, p_2$
4	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2, k_3, k_4, k_5$
5	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2, k_3, k_4$
6	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2, k_3$
7	$\Delta x_0, \Delta y_0, \Delta f, k_1, k_2$
8	$\Delta x_0, \Delta y_0, \Delta f, k_1$

The combinations are chosen for the following reasons: the radial distortion is dominant and therefore is included in all the combinations. The decentering distortion can be small enough that it can be absorbed by the position error of the principal point [Fraser, 2012] and the affine distortion may not exist in the digital cameras [Tang, 2013]. Thus, they only appear in combinations 1, 2 and 3. The combinations of the image distortion coefficients were computed and the ranging accuracy of the stereo system was analyzed. For each combination, the followings were conducted:

1. The camera system was calibrated using the traditional calibration method (i.e. the checkerboard coordinates were set as constants) so that the exterior orientation of the cameras can be obtained.
2. Auto-calibration was performed using the collinearity equations.
3. By using the estimated position and orientation of the stereo system from the tradition calibration results in Step 1 and the auto calibration results in Step 2, the 3D position of the corners of the checkerboard were estimated by space intersection.
4. The differences of the estimated 3D positions and the *true* checkerboard corner positions were computed. Then the mean and RMS of the differences was determined.

Figure 4.14 shows the X, Y, Z and the 3D ranging errors for combinations 1 to 8.

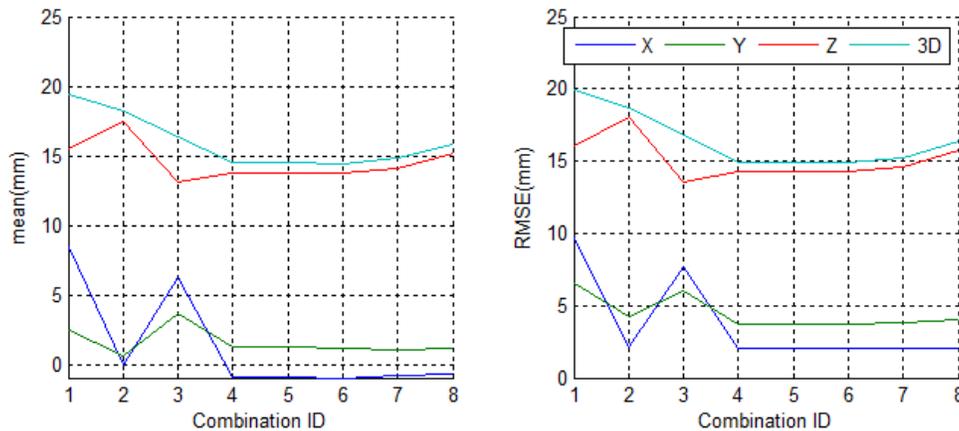


Figure 4.14 X, Y, Z and 3D ranging errors for combinations 1 to 8

The results from the combinations 1, 2 and 3 showed that the inclusion of the decentering distortion and the affine distortion terms degraded the overall ranging accuracy in comparison with the models that contained only the radial distortion terms.

The results were degraded due to the high correlation of the decentering distortion and the affine distortion with other parameters. Table 4.14 shows the correlation coefficients between the focal length error and affine parameters for Combination 2.

Table 4.14 The correlation coefficients of the focal length error and affine parameters for Combination 2

	Left Camera				Right Camera		
	$\Delta f$	$A_1$	$A_2$		$\Delta f$	$A_1$	$A_2$
$\Delta f$	-	<b>0.82</b>	0.17	$\Delta f$	-	<b>0.79</b>	0.39
$A_1$		-	0.18	$A_1$		-	-0.18
$A_2$			-	$A_2$			-

The correlation coefficients between principal point error and decentering distortion parameters for Combination 3 are shown in Table 4.15.

Table 4.15 The correlation coefficients of the principal point error and decentering distortion parameters for Combination 3

	Left Camera					Right Camera			
	$\Delta x_0$	$\Delta y_0$	$p_1$	$p_2$		$\Delta x_0$	$\Delta y_0$	$p_1$	$p_2$
$\Delta x_0$	-	-0.12	<b>0.92</b>	0.23	$\Delta x_0$	-	-0.22	<b>0.89</b>	-0.17
$\Delta y_0$		-	0.09	<b>0.89</b>	$\Delta y_0$		-	0.21	<b>0.86</b>
$p_1$			-	-0.16	$p_1$			-	-0.18
$p_2$				-	$p_2$				-

Table 4.14 shows that the correlation coefficient between  $\Delta f$  and the affine parameter  $A_2$  are 0.82 and 0.79 for the left and right cameras respectively. Furthermore, Table 4.15 shows that  $\Delta x_0$  and  $\Delta y_0$  are highly correlated with  $p_1$  and  $p_2$ , respectively, with correlation coefficients greater than 0.86. A high correlation mathematically implies a significant linear relation between two parameters in question and can cause instability in the solving the bundle adjustment solution. The presence of these high correlation

degraded the overall calibration solution. (Note: the full correlation analysis of all the results were omitted; only relevant ones were presented)

Other than the principal point error and the focal length error, the combinations 4 to 8 contain only radial distortion terms. The RMSE decreased from the combinations 8 to 6 and then levels off from 6 to 4. This showed that most of the radial distortion is accounted up to the term  $k_3$  and the inclusion of the coefficients  $k_4$  and  $k_5$  does not improve the overall solution accuracy. Furthermore, the mean errors slightly increased when  $k_4$  and  $k_5$  were included. The test results showed that the combination 6 performed the best, i.e.

$$\begin{aligned}\Delta x_{d,i} &= -\Delta x_0 - \frac{\bar{x}}{f} \Delta f + \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6) \\ \Delta y_{d,i} &= -\Delta y_0 - \frac{\bar{y}}{f} \Delta f + \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6)\end{aligned}\tag{4.22}$$

### 4.5.3. Comparison between COL and SRE

The SRE and COL auto-calibration algorithms were tested and their performances were compared using the auto-calibration model in (4.22). The calibration results using the traditional method via collinearity equations were used as the reference solution. The standard deviation of the measurement for the image points was assumed to be  $\pm 0.3$ px. Table 4.16, Table 4.17 and Table 4.18 show the estimates and their standard deviations for the left, right and relative orientation camera calibration parameters, respectively

Table 4.16 Left camera lens distortion parameters

Parameter	Collinearity Equations (Traditional)		Collinearity Equations (Auto)		Scale Restraint Equations (Auto)	
	mean	stdev	mean	stdev	mean	stdev
$\Delta f_L$ (px)	-1.72	0.40	2.49	0.56	2.50	0.79
$\Delta x_{L,0}$ (px)	-0.49	0.33	-0.24	0.37	0.04	0.57
$\Delta y_{L,0}$ (px)	0.47	0.32	-0.17	0.36	0.15	0.73
$k_{L,1}$ (px <sup>-2</sup> )	-4.80e <sup>-07</sup>	1.35e <sup>-08</sup>	-4.93e <sup>-07</sup>	1.30e <sup>-08</sup>	-4.79e <sup>-07</sup>	1.11e <sup>-08</sup>
$k_{L,2}$ (px <sup>-4</sup> )	1.09e <sup>-12</sup>	2.64e <sup>-13</sup>	4.91e <sup>-13</sup>	2.57e <sup>-13</sup>	4.93e <sup>-13</sup>	2.12e <sup>-13</sup>
$k_{L,3}$ (px <sup>-6</sup> )	-3.35e <sup>-18</sup>	1.55e <sup>-18</sup>	-5.37e <sup>-20</sup>	1.52e <sup>-18</sup>	7.11e <sup>-19</sup>	1.27e <sup>-18</sup>

Table 4.17 Right camera lens distortion parameters.

Parameter	Collinearity Equations (Traditional)		Collinearity Equations (Auto)		Scale Restraint Equations (Auto)	
	mean	stdev	mean	stdev	mean	stdev
$\Delta f_R$ (px)	2.02	0.24	5.60±	0.54	6.58	0.84
$\Delta x_{R,0}$ (px)	1.02	0.24	-0.63±	0.37	0.49	0.64
$\Delta y_{R,0}$ (px)	2.80	0.22	1.77±	0.36	1.88	0.76
$k_{R,1}$ (px <sup>-2</sup> )	-4.97e <sup>-07</sup>	1.24e <sup>-08</sup>	-4.94e <sup>-07</sup>	1.31e <sup>-08</sup>	-4.88e <sup>-07</sup>	1.10e <sup>-08</sup>
$k_{R,2}$ (px <sup>-4</sup> )	9.99e <sup>-13</sup>	2.49e <sup>-13</sup>	3.92e <sup>-13</sup>	2.63e <sup>-13</sup>	6.69e <sup>-13</sup>	2.20e <sup>-13</sup>
$k_{R,3}$ (px <sup>-6</sup> )	-1.82e <sup>-18</sup>	1.50e <sup>-18</sup>	9.88e <sup>-19</sup>	1.56e <sup>-18</sup>	-3.15e <sup>-19</sup>	1.39e <sup>-18</sup>

Table 4.18 Baseline and relative orientation estimates. †Free parameter.

Parameter	Collinearity Equations (Traditional)		Collinearity Equations (Auto)		Scale Restraint Equations (Auto)	
	mean	stdev	mean	stdev	mean	stdev
$b_{LR,x}^c$ (mm)	3.11	0.43	3.15	0.60	8.36	2.99
$b_{LR,y}^c$ (mm)	650.19	0.59	†650.19	-	†650.19	-
$b_{LR,z}^c$ (mm)	2.62	2.34	6.73	3.63	3.81	3.45
$\theta_{cR,x}^c$ (deg)	0.32	0.06	0.36	0.10	0.29	0.04
$\theta_{cR,y}^c$ (deg)	-0.23	0.07	-0.41	0.12	-0.30	0.05
$\theta_{cR,z}^c$ (deg)	-0.06	0.01	-0.07	0.02	-0.04	0.01

Table 4.19 Number of the points and parameters

	<b>Collinearity Equations (Traditional)</b>	<b>Collinearity Equations (Auto)</b>	<b>Scale Restraint Equations (Auto)</b>
Number of points	5621	5621	5621
Number of parameters	354	578	347

Table 4.19 shows the total number of the used feature points and the number of the estimated parameters. As expected, the results showed that the estimated standard deviations in SRE are worse than the ones in COL with SRE containing less number of the estimated parameters. The accuracy of the auto-calibration parameters was evaluated as follows

1. The exterior orientation of the cameras was obtained using the traditional calibration method.
2. Using the estimated position and orientation of the stereo system from the tradition calibration results in Step 1 and the auto calibration results of COL and SRE, the 3D position of the corners of the checkerboard were estimated by using space intersection.
3. The differences of the estimated 3D positions and the *true* checkerboard corner positions were computed. Then the mean, standard deviations and RMS of the differences were determined.

Table 4.20 shows the mean, the standard deviations and the RMS of the position errors. The results showed that the SRE algorithm performed worse than the COL algorithm. This is expected since the same measurements were used.

Table 4.20 Means, standard deviations and RMS of the deviations between the corresponding 3D reference points and estimated points.

	Collinearity Equations (Auto)			Scale Restraint Equations (Auto)		
	dx(mm)	dy(mm)	dz(mm)	dx(mm)	dy(mm)	dz(mm)
mean	-0.94	1.22	13.76	0.74	1.85	16.94
std	1.81	3.54	3.71	1.65	4.18	4.83
rms	2.04	3.75	14.25	1.81	4.57	17.61

Table 4.21 shows the means, the standard deviations and the RMS of the image residuals. The residual RMSes are at subpixel levels. This shows a tight fit of the measurement models with the measurements.

Table 4.21 Means, standard deviations and RMS of the image residuals.

Camera	Collinearity Equations (Auto)				Scale Restraint Equations (Auto)			
	Left		Right		Left		Right	
	$v_x$ (px)	$v_y$ (px)	$v_x$ (px)	$v_y$ (px)	$v_x$ (px)	$v_y$ (px)	$v_x$ (px)	$v_y$ (px)
mean	-0.003	0.005	-0.005	-0.007	0.002	0.003	-0.004	-0.006
std	0.140	0.149	0.142	0.156	0.158	0.154	0.177	0.162
rms	0.140	0.149	0.142	0.157	0.158	0.154	0.177	0.163

## 4.6. Summary

This chapter presented the design and implementation of a structureless camera auto-calibration method in a GNSS, IMU and Stereo camera integrated kinematic positioning and navigation system. The most widely used mathematical model for camera auto-calibration is the well-known extended collinearity equations. The vast majority of the parameters to be solved in the bundle adjustment are the object coordinates. However, they are not directly part of the calibration parameters. By removing or omitting these parameters from the system of equations, the memory and computational usage can drastically be reduced. Structureless auto-calibration methods, which are free of object

coordinates, have been proposed. For instance, Faig [1975] and Cefalu et al [2016] utilized the two-view coplanarity constraint in their bundle adjustment. However, they did not consider the scale-consistency between the image positions in their calibration algorithm. Scale-consistency is important when measurements from external sensors, such as a direct-georeferencing system with GNSS receivers and IMU, are incorporated in an auto-calibration.

The realized auto-calibration here in this research employs the scale-restraint equation and consists of two-stages. The first step determines the lens distortion parameters, the up-to-scale baseline length and the relative orientation between the two cameras by employing two scale-restraint equations to constrain the matched features from two consecutive stereo pairs. Then, the system calibration is introduced to recover the camera lever-arms, and the bore-sight angles with respect to the IMU, and the absolute scale of the camera using the GNSS-aided inertial navigation solution. The main advantage of the proposed method is that it is free of the object positions in the parameters vector and this results in computation and memory savings. The contributions of the research in this chapter can be summarized as follows:

- The development of a camera auto-calibration and system calibration algorithm for a GNSS, IMU and stereo camera system that is based on the scale-restraint equation.
- The pre-analysis and comparison of the calibration accuracy of a self-calibrating bundle adjustment based on the collinearity equations and the proposed method. Test results using simulated data for a land vehicle scenario showed (a) the collinearity equations performed better than the scale-restraint equations when the

same measurements were used and (b) the scale-restraint equation required approximately four times more measurements to produce comparable results to the collinearity equations.

- The analysis and comparison of the computational complexity for solving the self-calibrating bundle adjustment based on the collinearity equations and the scale-restraint equation. The test results with simulated data showed that the scale-restraint equation required 4.5 times more measurements than the collinearity equations to achieve comparable calibration accuracy while using only 0.1% of the computational resources.

The real road tests and their results will be presented in Section 7.3.

## 5. Loosely-coupled visual odometry aided inertial navigation

This chapter is mainly based on the following publication:

Gopaul, Nilesh S.; Wang, Jianguo and Hu, Baoxin (2017): *Loosely coupled visual odometry aided inertial navigation system using discrete extended Kalman filter with pairwise time correlated measurements*, 2017 CPGPS Forum on Cooperative Positioning and Service, 19-21 May 2017, Harbin, China, IEEE EI-Indexed Proceedings, Electronic ISBN: 978-1-5090-5022-2/17©2017 IEEE, DOI: 10.1109/CPGPS.2017.8075140.

### 5.1. Introduction

A loosely-coupled (LC) visual odometry (VO) aided-INS implies that pose changes estimated from the VO engine are used to aid the inertial navigation system. The Kalman filter algorithm is generally preferred for the fusion of the inertial measurements and the measurements from other aiding sensors. The Kalman filter in the standard form assumes that the process and the measurement noise vectors are white and conform to normal distributions with zero means. However, the assumption about white noise is not satisfied with the VO measurements since the VO estimates are pairwise time-correlated, i.e. the measurements at the current epoch are correlated with the measurements from the previous epoch in the stochastic sense. Neglecting significant time-correlated errors can degrade the performance of the filter and produce suboptimal solution.

This chapter develops an algorithm for processing pairwise time-correlated measurements in a Kalman filter. Time-correlated errors are usually modelled by a shaping filter. The shaping filter presented in this chapter uses Cholesky factors as the

coefficients derived from the variance and covariance matrices of the measurement noise vectors. The advantage of using Cholesky factors is that they are derived sequentially epoch-by-epoch and therefore it is suitable to be implemented in a Kalman filter. The results using the simulated data showed that the proposed algorithm performed better than the existing Kalman filter algorithms and provided more realistic covariance estimates. Furthermore, because VO measurements are relative in nature, the Kalman filter formulation for processing relative measurements is also represented.

This chapter is organized as follows: Section 5.2 constructs the Kalman filter for processing the relative measurements and dealing with their pairwise time-correlation. Then, the simulation tests results are shown in Section 5.3 to validate the development presented in Section 5.2. Section 5.4 presents the system and measurement models for the VO-aided inertial navigation in loosely-coupled mode. Finally, a summary is given in Section 5.5. The field test results and analysis are presented in Section 7.4.

## **5.2. Design of the Kalman filter**

The assumptions prior to constructing the Kalman filter are

- the stereo camera system is calibrated;
- the visual odometry is derived from two consecutive image frames;
- the IMU data rate is higher than the one of the camera system;
- the IMU has a constant output rate;
- the camera system output rate may not necessarily be constant;
- the image and IMU measurements are synchronized.

### 5.2.1. Relative measurements

The VO estimates are *relative* in nature because they are the differences of the positions and attitudes of a moving vehicle between two consecutive image frames. In order to process the relative measurements, the equations in the standard Kalman Filter have to be modified. This section summarizes the *stochastic cloning* algorithm for processing relative measurements which was developed by [Roumeliotis et al, 2002].

Consider the following linear/linearized discrete system and measurement model:

$$\begin{aligned} \mathbf{x}_k &= \Phi_{k-1} \mathbf{x}_{k-1} + \Psi_{k-1} \mathbf{w}_{k-1}, \mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{H}_{k-m} \mathbf{x}_{k-m} + \mathbf{v}_k, \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k), m > 1 \end{aligned} \quad (5.1)$$

wherein the measurements are the functions of the state vectors at the current epoch  $k$  and the past epoch  $k-m$ . Furthermore, there are no aiding measurements available between  $k-m$  and  $k$ , i.e. only the system model is used to predict the state vectors from epoch  $k-m$  to epoch  $k$ , and conducts the measurement update at epoch  $k$ . Figure 5.1 depicts the relationship between the states and measurements between the epoch  $k-m$  and the epoch  $k$ .

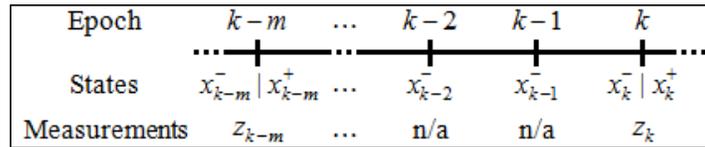


Figure 5.1 Timeline between epochs  $k-m$  and  $k$ .

The derivation starts with the estimated state vector  $\mathbf{x}_{k-m}^+$  at the epoch  $k-m$ . The augmented state vector and its associated variance-covariance matrix are

$$\begin{aligned}\tilde{\mathbf{x}}_{k-m}^+ &= \begin{pmatrix} \mathbf{x}_{1,k-m}^+ \\ \mathbf{x}_{2,k-m}^+ \end{pmatrix} \\ \tilde{\mathbf{P}}_{k-m}^+ &= \begin{pmatrix} \mathbf{P}_{k-m}^+ & \mathbf{P}_{k-m}^+ \\ \mathbf{P}_{k-m}^+ & \mathbf{P}_{k-m}^+ \end{pmatrix}\end{aligned}\quad (5.2)$$

Where  $\sim$  denotes the augmented vectors and matrices, and the subscripts 1 and 2 differentiate between two copies of the state vector  $\mathbf{x}_{k-m}^+$ . Between two epochs,  $k-m$  and epoch  $k$ , only the first copy of  $\mathbf{x}_k^+$  is propagated while the second one has to remain in stationary. Therefore, the augmented system becomes

$$\begin{pmatrix} \mathbf{x}_{1,k-m+1}^- \\ \mathbf{x}_{2,k-m+1}^- \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi}_{k-m} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{1,k-m}^+ \\ \mathbf{x}_{2,k-m}^+ \end{pmatrix} + \begin{pmatrix} \mathbf{\Psi}_{k-m} \\ \mathbf{O} \end{pmatrix} \mathbf{w}_{k-m}. \quad (5.3)$$

Now, the propagation of the augmented variance-covariance matrix is discussed. At epoch  $k-m+1$ , the variance-covariance matrix of the predicted augmented state vector becomes

$$\begin{aligned}\tilde{\mathbf{P}}_{k-m+1}^- &= \begin{pmatrix} \mathbf{\Phi}_{k-m} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{P}_{k-m}^+ & \mathbf{P}_{k-m}^+ \\ \mathbf{P}_{k-m}^+ & \mathbf{P}_{k-m}^+ \end{pmatrix} \begin{pmatrix} \mathbf{\Phi}_{k-m} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{pmatrix}^T \\ &\quad + \begin{pmatrix} \mathbf{\Psi}_{k-m} \\ \mathbf{O} \end{pmatrix} \mathbf{Q}_{k-m} \begin{pmatrix} \mathbf{\Psi}_{k-m} \\ \mathbf{O} \end{pmatrix}^T \\ &= \begin{pmatrix} \mathbf{\Phi}_{k-m} \mathbf{P}_{k-m}^+ \mathbf{\Phi}_{k-m}^T + \mathbf{\Psi}_{k-m} \mathbf{Q}_{k-m} \mathbf{\Psi}_{k-m}^T & \mathbf{\Phi}_{k-m} \mathbf{P}_{k-m}^+ \\ \mathbf{P}_{k-m}^+ \mathbf{\Phi}_{k-m}^T & \mathbf{P}_{k-m}^+ \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{P}_{k-m+1}^- & \mathbf{\Phi}_{k-m} \mathbf{P}_{k-m}^+ \\ \mathbf{P}_{k-m}^+ \mathbf{\Phi}_{k-m}^T & \mathbf{P}_{k-m}^+ \end{pmatrix}\end{aligned}\quad (5.4)$$

Similarly, the variance-covariance matrix of the predicted augmented state vector at epoch  $k-m+2$  is

$$\tilde{\mathbf{P}}_{k-m+2}^- = \begin{pmatrix} \mathbf{P}_{k-m+2}^- & \mathbf{\Phi}_{k-m+1} \mathbf{\Phi}_{k-m} \mathbf{P}_{k-m}^+ \\ \mathbf{P}_{k-m}^+ \mathbf{\Phi}_{k-m}^T \mathbf{\Phi}_{k-m+1}^T & \mathbf{P}_{k-m}^+ \end{pmatrix}. \quad (5.5)$$

Finally, at epoch  $k$ , the variance-covariance matrix of the predicted augmented state vector becomes

$$\tilde{\mathbf{P}}_k^- = \begin{pmatrix} \mathbf{P}_k^- & \mathbf{\Phi}'_{k-1,k-m} \mathbf{P}_{k-m}^+ \\ \mathbf{P}_{k-m}^+ \mathbf{\Phi}'_{k-1,k-m}{}^T & \mathbf{P}_{k-m}^+ \end{pmatrix} \quad (5.6)$$

with  $\mathbf{\Phi}'_{k-1,k-m} = \mathbf{\Phi}_{k-1} \mathbf{\Phi}_{k-2} \cdots \mathbf{\Phi}_{k-m}$ . The variance-covariance matrix of the system innovation vector at epoch  $k$  is given by

$$\begin{aligned} \mathbf{S}_k &= (\mathbf{H}_k \quad \mathbf{H}_{k-m}) \tilde{\mathbf{P}}_k^- (\mathbf{H}_k \quad \mathbf{H}_{k-m})^T + \mathbf{R}_k \\ &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \mathbf{\Phi}'_{k-1,k-m}{}^T \mathbf{H}_k^T \cdots \\ &\quad + \mathbf{H}_k \mathbf{\Phi}'_{k-1,k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T + \mathbf{R}_k \end{aligned} \quad (5.7)$$

After the measurement update, the variance-covariance matrix of the augmented state vector can be obtained as follows

$$\begin{aligned} \tilde{\mathbf{P}}_k^+ &= \tilde{\mathbf{P}}_k^- - \tilde{\mathbf{P}}_k^- (\mathbf{H}_k \quad \mathbf{H}_{k-m})^T \mathbf{S}_k^{-1} (\mathbf{H}_k \quad \mathbf{H}_{k-m}) \tilde{\mathbf{P}}_k^- \\ &= \tilde{\mathbf{P}}_k^- - \begin{pmatrix} \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{\Phi}'_{k-1,k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T \\ \mathbf{P}_{k-m}^+ \mathbf{\Phi}'_{k-1,k-m}{}^T \mathbf{H}_k^T + \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T \end{pmatrix} \mathbf{S}_k^{-1} \cdots \\ &\quad \left( \mathbf{H}_k \mathbf{P}_k^- + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \mathbf{\Phi}'_{k-1,k-m}{}^T \quad \mathbf{H}_k \mathbf{\Phi}'_{k-1,k-m} \mathbf{P}_{k-m}^+ + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \right) \end{aligned} \quad (5.8)$$

The corresponding Kalman gain matrix is given by

$$\begin{aligned} \tilde{\mathbf{K}}_k &= \begin{pmatrix} \mathbf{K}_{1,k} \\ \mathbf{K}_{2,k} \end{pmatrix} = \tilde{\mathbf{P}}_k^- (\mathbf{H}_k \quad \mathbf{H}_{k-m})^T \mathbf{S}_k^{-1} \\ &= \begin{pmatrix} (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{\Phi}'_{k-1,k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T) \mathbf{S}_k^{-1} \\ (\mathbf{P}_{k-m}^+ \mathbf{\Phi}'_{k-1,k-m}{}^T \mathbf{H}_k^T + \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T) \mathbf{S}_k^{-1} \end{pmatrix} \end{aligned} \quad (5.9)$$

The solution of the augmented state vector after the measurement update is equal to

$$\tilde{\mathbf{x}}_k^+ = \tilde{\mathbf{x}}_k^- + \tilde{\mathbf{K}}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- - \mathbf{H}_{k-m} \mathbf{x}_{k-m}^+). \quad (5.10)$$

In summary, the basic equations are:

$$\begin{aligned}
\Phi'_{k-1,k-m} &= \Phi_{k-1} \Phi_{k-2} \cdots \Phi_{k-m} \\
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \Phi_{k-1,k-m}'^T \mathbf{H}_k^T \cdots \\
&\quad + \mathbf{H}_k \Phi'_{k-1,k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T + \mathbf{R}_k \\
\mathbf{K}_k &= (\mathbf{P}_k^- \mathbf{H}_k^T + \Phi'_{k-1,k-m} \mathbf{P}_{k-m}^+ \mathbf{H}_{k-m}^T) \mathbf{S}_k^{-1} \\
\mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- - \mathbf{H}_{k-m} \mathbf{x}_{k-m}^+) \\
\mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{H}_{k-m} \mathbf{P}_{k-m}^+ \Phi_{k-1,k-m}'^T)
\end{aligned} \tag{5.11}$$

### 5.2.2. Pairwise time correlated measurements

This section develops a shaping filter to handle pairwise time-correlated measurement noises in a Kalman filter [Gopaul et al, 2014, 2017]. The goal is to derive the optimal Kalman filter solution and run the filter sequentially in practice. Since the measurement noise vector is independent of whether the measurement type is relative or not, the relative measurement states described in Section 5.2.1 are not included in the derivation here. They can be added later. The derivation starts with the following measurement model

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{5.12}$$

If  $\mathbf{R}_k$  is positive definite, it can be decomposed using the Cholesky factorization [Grewal and Andrews, 2001; Bierman, 2006, etc.] as follows:

$$\mathbf{R}_k = \mathbf{C}_k \mathbf{C}_k^T \tag{5.13}$$

wherein the Cholesky factor matrix  $\mathbf{C}_k$  is unique and real lower triangular with the strictly positive diagonal elements. By multiplying (5.12) with its inverse, i.e.  $\mathbf{C}_k^{-1}$ , one obtains

$$\mathbf{z}'_k = \mathbf{C}_k^{-1} \mathbf{z}_k = \mathbf{C}_k^{-1} \mathbf{H}_k \mathbf{x}_k + \mathbf{C}_k^{-1} \mathbf{v}_k = \mathbf{H}'_k \mathbf{x}_k + \mathbf{v}' \tag{5.14}$$

wherein the individual components in the derived measurement noise vector  $\mathbf{v}'_k$  are uncorrelated and have the identity matrix as their variance and covariance matrix:

$$\begin{aligned}\mathbf{R}'_k &= E[(\mathbf{v}'_k)(\mathbf{v}'_k)^T] = \mathbf{C}_k^{-1}E[\mathbf{v}_k\mathbf{v}_k^T]\mathbf{C}_k^{-T} \\ &= \mathbf{C}_k^{-1}\mathbf{R}_k\mathbf{C}_k^{-T} = \mathbf{C}_k^{-1}\mathbf{C}_k\mathbf{C}_k^T\mathbf{C}_k^{-T} \\ &= \mathbf{I}\end{aligned}\quad (5.15)$$

The substitution of  $\mathbf{v}_k = \mathbf{C}_k\mathbf{v}'_k$  into (5.12) yields

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{C}_k\mathbf{v}'_k, \mathbf{v}'_k \sim N(\mathbf{0}, \mathbf{I}) \quad (5.16)$$

Now suppose that the measurement noises are pairwise-time correlated, i.e.  $E[\mathbf{z}_k\mathbf{z}_{k-1}^T] \neq \mathbf{0}$  and  $E[\mathbf{z}_k\mathbf{z}_j^T] = \mathbf{0}$  for  $j \neq k-1$  and  $j \neq k$ , which are equivalent to  $E[\mathbf{v}_k\mathbf{v}_{k-1}^T] \neq \mathbf{0}$  and  $E[\mathbf{v}_k\mathbf{v}_j^T] = \mathbf{0}$  in terms of their noise vectors. After having concatenated all the measurement noise vectors from epoch 1 to epoch  $N$  into a vector  $\mathbf{v}$ , one obtains its corresponding variance-covariance matrix  $\mathbf{R}$

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \vdots \\ \mathbf{v}_N \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{32} & \mathbf{R}_{33} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{R}_{NN} \end{pmatrix}. \quad (5.17)$$

By applying the Cholesky factorization to the matrix  $\mathbf{R}$ , one has

$$\begin{aligned}
\mathbf{R} &= \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{32} & \mathbf{R}_{33} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{R}_{NN} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{C}_{11} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_{32} & \mathbf{C}_{33} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{C}_{NN} \end{pmatrix} \begin{pmatrix} \mathbf{C}_{11} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_{32} & \mathbf{C}_{33} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{C}_{NN} \end{pmatrix}^T \\
&= \begin{pmatrix} \mathbf{C}_{11}\mathbf{C}_{11}^T & \mathbf{C}_{11}\mathbf{C}_{21}^T & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{C}_{21}\mathbf{C}_{11}^T & \mathbf{C}_{21}\mathbf{C}_{21}^T + \mathbf{C}_{22}\mathbf{C}_{22}^T & \mathbf{C}_{22}\mathbf{C}_{32}^T & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_{32}\mathbf{C}_{22}^T & \mathbf{C}_{32}\mathbf{C}_{32}^T + \mathbf{C}_{33}\mathbf{C}_{33}^T & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{C}_{N-1,N}\mathbf{C}_{N-1,N}^T + \mathbf{C}_{NN}\mathbf{C}_{NN}^T \end{pmatrix}
\end{aligned} \tag{5.18}$$

The original global measurement noise vector  $\mathbf{v}$  is related to the de-correlated global measurement noise vector  $\mathbf{v}'$  as follows

$$\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \vdots \\ \mathbf{v}_N \end{pmatrix} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_{32} & \mathbf{C}_{33} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{C}_{NN} \end{pmatrix} \begin{pmatrix} \mathbf{v}'_1 \\ \mathbf{v}'_2 \\ \mathbf{v}'_3 \\ \vdots \\ \mathbf{v}'_N \end{pmatrix} \tag{5.19}$$

In short form, the measurement noise vector at an arbitrary epoch  $k$  can be written as

$$\mathbf{v}_k = \mathbf{C}_{k,k} \mathbf{v}'_k + \mathbf{C}_{k,k-1} \mathbf{v}'_{k-1} \tag{5.20}$$

with  $\mathbf{v}'_k \sim N(\mathbf{0}, \mathbf{I})$  and  $\mathbf{C}_{1,0} = \mathbf{O}$ . Based on (5.18), the Cholesky factor matrices  $\mathbf{C}_{k,k-1}$  and  $\mathbf{C}_{k,k}$  can be obtained sequentially from epoch to epoch as described in Algorithm

5.1.

Algorithm 5.1: Sequential Cholesky factorization of the pairwise time correlated variance-covariance matrix

$$\begin{array}{l}
 \mathbf{C}_{0,0} = \mathbf{O} \\
 \mathbf{C}_{1,0} = \mathbf{O} \\
 \text{for } k = 1, \dots, N \{ \\
 \quad \mathbf{C}_{k,k-1} = (\mathbf{C}_{k-1,k-1}^{-1} \mathbf{R}_{k-1,k})^T \\
 \quad \mathbf{C}_{k,k} = \text{chol}(\mathbf{R}_{k,k} - \mathbf{C}_{k,k-1} \mathbf{C}_{k,k-1}^T) \\
 \}
 \end{array}$$

By substituting  $\mathbf{v}_k$  in (5.20) into (5.12), the measurement model becomes

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{C}_{k,k} \mathbf{v}'_k + \mathbf{C}_{k,k-1} \mathbf{v}'_{k-1}, \mathbf{v}'_k \sim N(\mathbf{0}, \mathbf{I}) \quad (5.21)$$

With  $\mathbf{C}_{1,0} = \mathbf{O}$ . Based on (3.27), the augmented system and measurement models

become:

$$\begin{pmatrix} \mathbf{x}_k \\ \mathbf{v}'_k \\ \mathbf{v}'_{k-1} \end{pmatrix} = \begin{pmatrix} \Phi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{v}'_{k-1} \\ \mathbf{v}'_{k-2} \end{pmatrix} + \begin{pmatrix} \Psi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{w}_{k-1} \\ \boldsymbol{\eta}_{k-1} \\ \mathbf{0} \end{pmatrix} \quad (5.22)$$

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{H}_k & \mathbf{C}_{k,k} & \mathbf{C}_{k,k-1} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{v}'_k \\ \mathbf{v}'_{k-1} \end{pmatrix} + \mathbf{0}$$

Since  $\mathbf{v}'_k$  is now part of the state vector, its corresponding noise is modelled as the process noise vector  $\boldsymbol{\eta}_{k-1}$ ,  $\mathbf{v}'_k = \boldsymbol{\eta}_{k-1}$ . Therefore,  $\boldsymbol{\eta}_{k-1} \sim N(\mathbf{0}, \mathbf{I})$ . In short form, the augmented system (5.22) can be written as

$$\begin{aligned}
 \tilde{\mathbf{x}}_k &= \tilde{\Phi}_{k-1} \tilde{\mathbf{x}}_{k-1} + \tilde{\Psi}_{k-1} \tilde{\mathbf{w}}_{k-1}, \tilde{\mathbf{w}}_{k-1} \sim N(\mathbf{0}, \tilde{\mathbf{Q}}_{k-1}) \\
 \mathbf{z}_k &= \tilde{\mathbf{H}}_k \tilde{\mathbf{x}}_k + \tilde{\mathbf{v}}_k, \tilde{\mathbf{v}}_k \sim N(\mathbf{0}, \mathbf{O})
 \end{aligned} \quad (5.23)$$

where  $\tilde{\cdot}$  denotes the augmented vectors and matrices. Notice that (5.23) is in the form of the standard Kalman filter with zero measurement variance matrix i.e.  $\tilde{\mathbf{R}}_k = \mathbf{O}$ .

Theoretically, the Kalman filter algorithm can handle measurement noises with zero variance-covariance [Simon, 2006] since it only requires the system innovation variance-covariance matrix  $\tilde{\mathbf{H}}_k \tilde{\mathbf{P}}_k^- \tilde{\mathbf{H}}_k^T$  to be non-singular [Grewal and Andrews, 2001]. Since (5.23) is in the form of the standard Kalman filter defined in (3.21) and (3.22), any smoother associated with the Kalman filter can easily be employed without any modifications. Next, (5.23) is simplified. The time update of the augmented state vector runs

$$\tilde{\mathbf{x}}_k^- = \begin{pmatrix} \Phi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{k-1}^+ \\ \mathbf{v}_{k-1}'^+ \\ \mathbf{v}_{k-2}'^+ \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k^- \\ \mathbf{v}_k'^- \\ \mathbf{v}_{k-1}'^- \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k^- \\ \mathbf{0} \\ \mathbf{v}_{k-1}'^- \end{pmatrix} \quad (5.24)$$

with the associated variance matrix update

$$\begin{aligned} \tilde{\mathbf{P}}_k^- &= \tilde{\Phi}_{k-1} \tilde{\mathbf{P}}_{k-1}^+ \tilde{\Phi}_{k-1}^T + \tilde{\Psi}_{k-1} \tilde{\mathbf{Q}}_{k-1} \tilde{\Psi}_{k-1}^T \\ &= \begin{pmatrix} \Phi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{P}_{k-1}^+ & \mathbf{P}_{k-1, \mathbf{x}_{k-1} \mathbf{v}'_{k-1}}^+ & \mathbf{P}_{k-1, \mathbf{x}_{k-1} \mathbf{v}'_{k-2}}^+ \\ \mathbf{P}_{k-1, \mathbf{v}'_{k-1} \mathbf{x}_{k-1}}^+ & \mathbf{P}_{k-1, \mathbf{v}'_{k-1} \mathbf{v}'_{k-1}}^+ & \mathbf{P}_{k-1, \mathbf{v}'_{k-1} \mathbf{v}'_{k-2}}^+ \\ \mathbf{P}_{k-1, \mathbf{v}'_{k-2} \mathbf{x}_{k-1}}^+ & \mathbf{P}_{k-1, \mathbf{v}'_{k-2} \mathbf{v}'_{k-1}}^+ & \mathbf{P}_{k-1, \mathbf{v}'_{k-2} \mathbf{v}'_{k-2}}^+ \end{pmatrix} \begin{pmatrix} \Phi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \end{pmatrix}^T \dots \\ &\quad + \begin{pmatrix} \Psi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \Psi_{k-1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \end{pmatrix}^T \\ &= \begin{pmatrix} \Phi_{k-1} \mathbf{P}_{k-1}^+ \Phi_{k-1}^T + \Psi_{k-1} \mathbf{Q}_{k-1} \Psi_{k-1}^T & \mathbf{O} & \Phi_{k-1} \mathbf{P}_{k-1, \mathbf{x}_{k-1} \mathbf{v}'_{k-1}}^+ \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{P}_{k-1, \mathbf{v}'_{k-1} \mathbf{x}_{k-1}}^+ \Phi_{k-1}^T & \mathbf{O} & \mathbf{P}_{k, \mathbf{v}'_{k-1} \mathbf{v}'_{k-1}}^- \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{P}_k^- & \mathbf{O} & \mathbf{P}_{k, \mathbf{x}_k \mathbf{v}'_{k-1}}^- \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{P}_{k, \mathbf{v}'_{k-1} \mathbf{x}_k}^- & \mathbf{O} & \mathbf{P}_{k, \mathbf{v}'_{k-1} \mathbf{v}'_{k-1}}^- \end{pmatrix} \end{aligned} \quad (5.25)$$

The gain matrix of the augmented system can be derived as follows

$$\begin{aligned}
\tilde{\mathbf{K}}_k &= \begin{pmatrix} \mathbf{K}_k \\ \mathbf{K}_{k,\mathbf{v}'_k} \\ \mathbf{K}_{k,\mathbf{v}'_{k-1}} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{P}_k^- & \mathbf{O} & \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- & \mathbf{O} & \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \end{pmatrix} \begin{pmatrix} \mathbf{H}_k^T \\ \mathbf{C}_{k,k}^T \\ \mathbf{C}_{k,k-1}^T \end{pmatrix} \cdots \\
&\quad \left[ \begin{pmatrix} \mathbf{H}_k & \mathbf{C}_{k,k} & \mathbf{C}_{k,k-1} \end{pmatrix} \begin{pmatrix} \mathbf{P}_k^- & \mathbf{O} & \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \\ \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- & \mathbf{O} & \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \end{pmatrix} \begin{pmatrix} \mathbf{H}_k^T \\ \mathbf{C}_{k,k}^T \\ \mathbf{C}_{k,k-1}^T \end{pmatrix} \right]^{-1} \\
&= \begin{pmatrix} \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T & \mathbf{C}_{k,k}^T \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T & \mathbf{C}_{k,k}^T \end{pmatrix} \begin{pmatrix} \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{C}_{k,k} \mathbf{C}_{k,k}^T \cdots \\ + \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \cdots \\ + \mathbf{H}_k \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \cdots \\ + \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \mathbf{H}_k^T \end{pmatrix}^{-1} \\
&= \begin{pmatrix} (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T)(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \boldsymbol{\Sigma}_k)^{-1} \\ \mathbf{C}_{k,k}^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \boldsymbol{\Sigma}_k)^{-1} \\ (\mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T)(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \boldsymbol{\Sigma}_k)^{-1} \end{pmatrix} \tag{5.26}
\end{aligned}$$

wherein

$$\begin{aligned}
\boldsymbol{\Sigma}_k &= \mathbf{C}_{k,k} \mathbf{C}_{k,k}^T + \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \cdots \\
&\quad + \mathbf{H}_k \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T + \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \mathbf{H}_k^T \\
&= (\mathbf{C}_{k,k} \quad \mathbf{C}_{k,k-1}) \begin{pmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \end{pmatrix} \begin{pmatrix} \mathbf{C}_{k,k}^T \\ \mathbf{C}_{k,k-1}^T \end{pmatrix} \tag{5.27}
\end{aligned}$$

The measurement update further runs

$$\tilde{\mathbf{x}}_k^+ = \begin{pmatrix} \mathbf{x}_k^+ \\ \mathbf{v}'_k^+ \\ \mathbf{v}'_{k-1}^+ \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k^- \\ \mathbf{v}'_k^- \\ \mathbf{v}'_{k-1}^- \end{pmatrix} + \begin{pmatrix} \mathbf{K}_k \\ \mathbf{K}_{k,\mathbf{v}'_k} \\ \mathbf{K}_{k,\mathbf{v}'_{k-1}} \end{pmatrix} (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- - \mathbf{C}_{k,k} \mathbf{v}'_k^- - \mathbf{C}_{k,k-1} \mathbf{v}'_{k-1}^-). \tag{5.28}$$

However, from (5.24),  $\mathbf{v}'_k^- = \mathbf{0}$ . So, (5.28) becomes

$$\tilde{\mathbf{x}}_k^+ = \begin{pmatrix} \mathbf{x}_k^+ \\ \mathbf{v}_k^+ \\ \mathbf{v}_{k-1}^+ \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k^- \\ \mathbf{0} \\ \mathbf{v}_{k-1}^- \end{pmatrix} + \begin{pmatrix} \mathbf{K}_k \\ \mathbf{K}_{k,\mathbf{v}'_k} \\ \mathbf{K}_{k,\mathbf{v}'_{k-1}} \end{pmatrix} (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- - \mathbf{C}_{k,k-1} \mathbf{v}_{k-1}^-). \quad (5.29)$$

The corresponding variance-covariance matrix of  $\tilde{\mathbf{x}}_k^+$  is

$$\begin{aligned} \tilde{\mathbf{P}}_k^+ &= \begin{pmatrix} \mathbf{P}_k^+ & \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_k}^+ & \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^+ \\ \mathbf{P}_{k,\mathbf{v}'_k\mathbf{x}_k}^+ & \mathbf{P}_{k,\mathbf{v}'_k\mathbf{v}'_k}^+ & \mathbf{P}_{k,\mathbf{v}'_k\mathbf{v}'_{k-1}}^+ \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^+ & \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_k}^+ & \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^+ \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} - \begin{pmatrix} \mathbf{K}_k \\ \mathbf{K}_{k,\mathbf{v}'_k} \\ \mathbf{K}_{k,\mathbf{v}'_{k-1}} \end{pmatrix} \begin{pmatrix} \mathbf{H}_k & \mathbf{C}_{k,k} & \mathbf{C}_{k,k-1} \end{pmatrix} \begin{pmatrix} \mathbf{P}_k^- & \mathbf{0} & \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- & \mathbf{0} & \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \end{pmatrix} \\ &= \begin{pmatrix} \begin{pmatrix} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \cdots \\ -\mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \end{pmatrix} & -\mathbf{K}_k \mathbf{C}_{k,k} & \begin{pmatrix} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \cdots \\ -\mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \end{pmatrix} \\ \begin{pmatrix} -\mathbf{K}_{k,\mathbf{v}'_k} \mathbf{H}_k \mathbf{P}_k^- \cdots \\ -\mathbf{K}_{k,\mathbf{v}'_k} \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \end{pmatrix} & (\mathbf{I} - \mathbf{K}_{k,\mathbf{v}'_k} \mathbf{C}_{k,k}) & \begin{pmatrix} \mathbf{K}_{k,\mathbf{v}'_k} \mathbf{H}_k \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \cdots \\ -\mathbf{K}_{k,\mathbf{v}'_k} \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \end{pmatrix} \\ \begin{pmatrix} (\mathbf{I} - \mathbf{K}_{k,\mathbf{v}'_{k-1}} \mathbf{C}_{k,k-1}) \cdots \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- - \mathbf{K}_{k,\mathbf{v}'_{k-1}} \mathbf{H}_k \end{pmatrix} & -\mathbf{K}_{k,\mathbf{v}'_{k-1}} \mathbf{C}_{k,k} & \begin{pmatrix} (\mathbf{I} - \mathbf{K}_{k,\mathbf{v}'_{k-1}} \mathbf{C}_{k,k-1}) \cdots \\ \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \\ -\mathbf{K}_{k,\mathbf{v}'_{k-1}} \mathbf{H}_k \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \end{pmatrix} \end{pmatrix} \quad (5.30) \end{aligned}$$

From (5.24) and (5.25), the time update of  $\mathbf{x}_k$  is summarized as

$$\begin{aligned} \mathbf{x}_k^- &= \Phi_{k-1} \mathbf{x}_{k-1}^+ \\ \mathbf{P}_k^- &= \Phi_{k-1} \mathbf{P}_{k-1}^+ \Phi_{k-1}^T + \Psi_{k-1} \mathbf{Q}_{k-1} \Psi_{k-1}^T. \end{aligned} \quad (5.31)$$

From (5.26), (5.29) and (5.30), the corresponding measurement update is further summarized

$$\begin{aligned}
\mathbf{d}_k &= \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- - \mathbf{C}_{k,k-1} \mathbf{v}'_{k-1} \\
\boldsymbol{\Sigma}_k &= \mathbf{C}_{k,k} \mathbf{C}_{k,k}^T + \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \\
&\quad \cdots + \mathbf{H}_k \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T + \mathbf{C}_{k,k-1} \mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{x}_k}^- \mathbf{H}_k^T \\
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \boldsymbol{\Sigma}_k \\
\mathbf{K}_k &= (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{S}_k^{-1} \\
\mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{K}_k \mathbf{d}_k \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-
\end{aligned} \tag{5.32}$$

Equation (5.32) requires the estimate  $\mathbf{v}'_{k-1}$ , its variance matrix  $\mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^-$  and the covariance matrix  $\mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^-$ . Based on (5.24) at epoch  $k$  and (5.29) at epoch  $k-1$ ,  $\mathbf{v}'_{k-1}$  can be computed by

$$\mathbf{v}'_{k-1} = \mathbf{v}'_{k-1}^+ = \mathbf{K}_{k-1,\mathbf{v}'_{k-1}} \mathbf{d}_{k-1} \tag{5.33}$$

Similarly, from (5.25) at epoch  $k$  and (5.30) at epoch  $k-1$ ,  $\mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^-$  and  $\mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^-$  can further be obtained by

$$\begin{aligned}
\mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- &= \mathbf{P}_{k-1,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^+ = (\mathbf{I} - \mathbf{K}_{k-1,\mathbf{v}'_k} \mathbf{C}_{k-1,k-1}) \\
\mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- &= \boldsymbol{\Phi}_{k-1} \mathbf{P}_{k-1,\mathbf{x}_{k-1}\mathbf{v}'_{k-1}}^+ = -\boldsymbol{\Phi}_{k-1} \mathbf{K}_{k-1} \mathbf{C}_{k-1,k-1}
\end{aligned} \tag{5.34}$$

By substituting  $\mathbf{K}_{k-1,\mathbf{v}'_{k-1}} = \mathbf{C}_{k-1,k-1}^T \mathbf{S}_{k-1}^{-1}$  ((5.26) at epoch  $k-1$ ) into (5.33) and (5.34), one obtains

$$\begin{aligned}
\mathbf{v}'_{k-1} &= \mathbf{C}_{k-1,k-1}^T \mathbf{S}_{k-1}^{-1} \mathbf{d}_{k-1} \\
\mathbf{P}_{k,\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- &= (\mathbf{I} - \mathbf{C}_{k-1,k-1}^T \mathbf{S}_{k-1}^{-1} \mathbf{C}_{k-1,k-1}) \\
\mathbf{P}_{k,\mathbf{x}_k\mathbf{v}'_{k-1}}^- &= -\boldsymbol{\Phi}_{k-1} \mathbf{K}_{k-1} \mathbf{C}_{k-1,k-1}
\end{aligned} \tag{5.35}$$

In summary, the time update can be written as

$$\begin{aligned}
\mathbf{x}_k^- &= \boldsymbol{\Phi}_{k-1} \mathbf{x}_{k-1}^+ \\
\mathbf{P}_k^- &= \boldsymbol{\Phi}_{k-1} \mathbf{P}_k^+ \boldsymbol{\Phi}_{k-1}^T + \boldsymbol{\Psi}_{k-1} \mathbf{Q}_{k-1} \boldsymbol{\Psi}_{k-1}^T
\end{aligned} \tag{5.36}$$

and so can the measurement update:

$$\begin{aligned}
\mathbf{v}'_{k-1} &= \mathbf{C}_{k-1,k-1}^T \mathbf{S}_{k-1}^{-1} \mathbf{d}_{k-1} \\
\mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- &= (\mathbf{I} - \mathbf{C}_{k-1,k-1}^T \mathbf{S}_{k-1}^{-1} \mathbf{C}_{k-1,k-1}) \\
\mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- &= -\Phi_{k-1} \mathbf{K}_{k-1} \mathbf{C}_{k-1,k-1} \\
\mathbf{C}_{k,k-1} &= (\mathbf{C}_{k-1,k-1}^{-1} \mathbf{R}_{k-1,k})^T \\
\mathbf{C}_{k,k} &= \text{chol}(\mathbf{R}_{k,k} - \mathbf{C}_{k,k-1} \mathbf{C}_{k,k-1}^T) \\
\mathbf{d}_k &= \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k^- - \mathbf{C}_{k,k-1} \mathbf{v}'_{k-1} \\
\boldsymbol{\Sigma}_k &= \mathbf{C}_{k,k} \mathbf{C}_{k,k}^T + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \cdots \\
&\quad + \mathbf{H}_k \mathbf{P}_{\mathbf{x}_k \mathbf{x}_k}^- \mathbf{C}_{k,k-1}^T + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{H}_k^T \\
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \boldsymbol{\Sigma}_k \\
\mathbf{K}_k &= (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{S}_k^{-1} \\
\mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{K}_k \mathbf{d}_k \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-
\end{aligned} \tag{5.37}$$

The final state covariance in Joseph stabilized form [Simon, 2006] is as follows

$$\begin{aligned}
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \boldsymbol{\Sigma}_k \mathbf{K}_k^T \cdots \\
&\quad - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^- - \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \mathbf{K}_k^T
\end{aligned} \tag{5.38}$$

The derivation of the state covariance in Joseph stabilized form can be found in Appendix 0. The estimator will run as long as the system innovation variance-covariance matrix  $\mathbf{S}_k$  is invertible [Grewal and Andrews, 2001] and the global measurement noise variance-covariance matrix  $\mathbf{R}$  in (5.17) is positive definite. To check for correctness, assume that the time-correlation is zero, i.e. set  $\mathbf{R}_{k-1,k} = \mathbf{O}$  in (5.37). The matrix  $\mathbf{C}_{k,k-1}$  becomes zero and the equations will be the same as in the standard Kalman filter.

### 5.2.3. Pairwise time-correlation in stereo visual odometry

This section derives the pairwise time correlation for the stereo visual odometry and also shows that the corresponding global variance-covariance matrix  $\mathbf{R}$  is always positive definite. The visual odometry measurement equation described in (3.80) at epoch  $k$  is

$$\mathbf{l}_{k,i}^c - \mathbf{C}_{c(k-1)}^{c(k)} \mathbf{l}_{k-1,i}^c - \Delta \mathbf{X}_{k,k-1}^c = \mathbf{0} \quad (5.39)$$

for  $i = 1, 2, \dots, N$  features. In short form (5.39) can be written as

$$\mathbf{h}_k(\Delta \mathbf{x}_{k,k-1}, \mathbf{l}_k, \mathbf{l}_{k-1}) = \mathbf{0} \quad (5.40)$$

wherein  $\Delta \mathbf{x}_{k,k-1} = [\Delta \mathbf{X}_{k,k-1}^c, \boldsymbol{\theta}_{c(k-1)}^{c(k)}]$  is the pose change between epoch  $k-1$  and epoch  $k$ ,

$\mathbf{l}_{k-1} = [\mathbf{l}_{k-1,1}^c, \mathbf{l}_{k-1,2}^c, \dots, \mathbf{l}_{k-1,N}^c]$  and  $\mathbf{l}_k = [\mathbf{l}_{k,1}^c, \mathbf{l}_{k,2}^c, \dots, \mathbf{l}_{k,N}^c]$  are the image measurement

vectors at epochs  $k-1$  and  $k$ , respectively. The pose change can be estimated as in

(3.25) using the conditional LS adjustment with parameters:

$$\begin{aligned} \delta \Delta \hat{\mathbf{x}}_{k,k-1} &= -\mathbf{G}_k \mathbf{h}_k(\Delta \mathbf{x}_{k,k-1}^{(0)}, \mathbf{l}_k, \mathbf{l}_{k-1}) \\ \mathbf{P}_{\delta \Delta \hat{\mathbf{x}}_{k,k-1}} &= (\mathbf{H}_{x,k}^T (\mathbf{H}_{z,k} \mathbf{R}_{\parallel,k} \mathbf{H}_{z,k}^T)^{-1} \mathbf{H}_{x,k})^{-1} \\ \mathbf{G}_k &= \mathbf{P}_{\delta \Delta \hat{\mathbf{x}}_{k,k-1}} \mathbf{H}_{x,k}^T (\mathbf{H}_{z,k} \mathbf{R}_{\parallel,k} \mathbf{H}_{z,k}^T)^{-1} \\ \mathbf{R}_{\parallel,k} &= \begin{pmatrix} \mathbf{R}_{\mathbf{l}_k \mathbf{l}_k} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{\mathbf{l}_{k-1} \mathbf{l}_{k-1}} \end{pmatrix} \end{aligned} \quad (5.41)$$

where  $\mathbf{P}_{\delta \Delta \hat{\mathbf{x}}_{k,k-1}}$ ,  $\mathbf{H}_{z,k}$ ,  $\mathbf{H}_{x,k}$ ,  $\mathbf{R}_{\parallel,k}$  and  $\Delta \mathbf{x}_{k,k-1}^{(0)}$  are the variance-covariance matrix for the estimated parameters, the Jacobian matrix associated with the image measurements in (5.40), the Jacobian matrix associated with the parameters in (5.40), the variance-covariance matrix of the image measurements and the initial approximation of the pose change vector, respectively. Through error propagation, the parameter variance-covariance matrix  $\mathbf{P}_{\delta \Delta \hat{\mathbf{x}}_{k,k-1}}$  can also be expressed in the form of

$$\begin{aligned}
\mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{k,k-1}} &= \mathbf{G}_k \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} & \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix} \mathbf{R}_{\mathbf{l},k} \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} & \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix}^T \mathbf{G}_k^T \\
&= \mathbf{G}_k \left( \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} \end{pmatrix} \mathbf{R}_{\mathbf{l}_k} \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} \end{pmatrix}^T + \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix} \mathbf{R}_{\mathbf{l}_{k-1}} \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix}^T \right) \mathbf{G}_k^T \\
&= \mathbf{G}_k \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} \end{pmatrix} \mathbf{R}_{\mathbf{l}_k} \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} \end{pmatrix}^T \mathbf{G}_k^T + \mathbf{G}_k \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix} \mathbf{R}_{\mathbf{l}_{k-1}} \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix}^T \mathbf{G}_k^T \\
&= \mathbf{B}_{k,k} \mathbf{R}_{\mathbf{l}_k} \mathbf{B}_{k,k}^T + \mathbf{B}_{k,k-1} \mathbf{R}_{\mathbf{l}_{k-1}} \mathbf{B}_{k,k-1}^T
\end{aligned} \tag{5.42}$$

where  $\mathbf{B}_{k,k} = \mathbf{G}_k \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_k} \end{pmatrix}$  and  $\mathbf{B}_{k,k-1} = \mathbf{G}_k \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix}$ .

Consider two consecutive VO estimates at epochs  $k-1$  and  $k$ :

$$\begin{aligned}
\delta\Delta\hat{\mathbf{x}}_{k-1,k-2} &= -\mathbf{G}_{k-1} \mathbf{h}_{k-1} (\Delta\mathbf{x}_{k-1,k-2}^{(0)}, \mathbf{l}_{k-1}, \mathbf{l}_{k-2}) \\
\delta\Delta\hat{\mathbf{x}}_{k,k-1} &= -\mathbf{G}_k \mathbf{h}_k (\Delta\mathbf{x}_{k,k-1}^{(0)}, \mathbf{l}_k, \mathbf{l}_{k-1})
\end{aligned} \tag{5.43}$$

Notice that the image measurement vector  $\mathbf{l}_{k-1}$  in (5.43) appears at both epochs.

Therefore,  $\delta\Delta\hat{\mathbf{x}}_{k-1,k-2}$  and  $\delta\Delta\hat{\mathbf{x}}_{k,k-1}$  are stochastically correlated due to  $\mathbf{l}_{k-1}$ . The cross-

covariance matrix between two successive epochs  $k-1$  and  $k$  can be computed as:

$$\begin{aligned}
\mathbf{R}_{k-1,k} &= E[\delta\Delta\hat{\mathbf{x}}_{k-1,k-2} \delta\Delta\hat{\mathbf{x}}_{k,k-1}^T] \\
&= \mathbf{G}_{k-1} \begin{pmatrix} \frac{\partial \mathbf{h}_{k-1}}{\partial \mathbf{l}_{k-1}} \end{pmatrix} \mathbf{R}_{\mathbf{l}_{k-1}} \begin{pmatrix} \frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \end{pmatrix}^T \mathbf{G}_k^T \\
&= \mathbf{B}_{k-1,k} \mathbf{R}_{\mathbf{l}_{k-1}} \mathbf{B}_{k,k}^T
\end{aligned} \tag{5.44}$$

Furthermore, for all  $j \neq k-1$  and  $j \neq k$ , one has

$$\mathbf{R}_{j,k} = E[\delta\Delta\hat{\mathbf{x}}_{j,j-1} \delta\Delta\hat{\mathbf{x}}_{k,k-1}^T] = \mathbf{O} \tag{5.45}$$

The degree of the correlation between two successive epochs  $k-1$  and  $k$  depends on the number of the shared image measurements between the estimates  $\delta\Delta\hat{\mathbf{x}}_{k-1,k-2}$  and

$\delta\Delta\hat{\mathbf{x}}_{k,k-1}$ . This directly depends on the percentage overlap between the image frames

$k-2$ ,  $k-1$  and  $k$ , as shown in Figure 5.2.

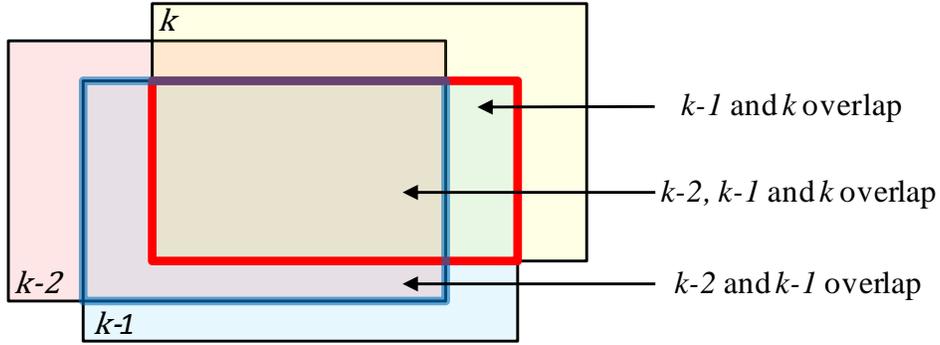


Figure 5.2 Overlap of frames  $k-2$ ,  $k-1$  and  $k$ .

If the percentage overlap is zero, then the correlation between the estimates  $\delta\Delta\hat{\mathbf{x}}_{k-1,k-2}$  and  $\delta\Delta\hat{\mathbf{x}}_{k,k-1}$  is obviously zero. If the overlap is 100% (i.e. when the vehicle is stationary), then  $\delta\Delta\hat{\mathbf{x}}_{k-1,k-2}$  and  $\delta\Delta\hat{\mathbf{x}}_{k,k-1}$  share half of the measurements and the correlation coefficient is approximately -0.50 (the negative sign comes from the fact that in (5.44),  $\frac{\partial \mathbf{h}_{k-1}}{\partial \mathbf{l}_{k-1}} = \mathbf{I}$  and  $\frac{\partial \mathbf{h}_k}{\partial \mathbf{l}_{k-1}} \approx -\mathbf{I}$  for small orientation change). This correlation coefficient depends on multiple factors, such as the velocity of the vehicle, frame rate, texture of the images and the number visible features. In reality, one can expect the value to be between -0.45 and -0.25. This is significant and if the correlation is ignored the Kalman filter then position and orientation errors can accumulate faster over long distances.

The global variance-covariance matrix  $\mathbf{R}$  is positive definite if and only if  $\mathbf{y}^T \mathbf{R} \mathbf{y} > 0$  for any non-zero vector  $\mathbf{y}$ . Another definition of the positive definiteness says that  $\mathbf{R}$  is positive definite if and only if  $\mathbf{R}$  can be expressed as  $\mathbf{R} = \mathbf{A} \mathbf{A}^T$ , where  $\mathbf{A}$  is an  $m \times p$  matrix, where  $m \leq p$ , and has independent columns [Strang, 2009]. The substitution of (5.42) and (5.44) into the global variance-covariance matrix  $\mathbf{R}$  in (5.17) yields

$$\mathbf{R} = \begin{pmatrix} \mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{1,0}} & \mathbf{R}_{12} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{R}_{21} & \mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{2,1}} & \mathbf{R}_{23} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{32} & \mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{3,1}} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{N,N}} \end{pmatrix} = \mathbf{B}\mathbf{R}_{\parallel}\mathbf{B}^T \quad (5.46)$$

where

$$\mathbf{R}_{\parallel} = \text{diag}(\mathbf{R}_{1_0 1_0}, \mathbf{R}_{1_1 1_1}, \dots, \mathbf{R}_{1_N 1_N}) \quad (5.47)$$

and

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{1,0} & \mathbf{B}_{1,1} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}_{3,2} & \mathbf{B}_{3,3} & \cdots & \mathbf{O} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{B}_{N,N} \end{pmatrix}. \quad (5.48)$$

Since  $\mathbf{R}_{\parallel}$  is diagonal, it can be factorized into two equal diagonal matrices

$$\begin{aligned} \mathbf{R} &= \mathbf{B}\mathbf{R}_{\parallel}\mathbf{B}^T \\ &= \mathbf{B}(\sqrt{\mathbf{R}_{\parallel}})(\sqrt{\mathbf{R}_{\parallel}})^T \mathbf{B}^T. \\ &= \mathbf{B}'\mathbf{B}'^T \end{aligned} \quad (5.49)$$

Since the columns in  $\mathbf{B}$  are independent and  $\mathbf{R}_{\parallel}$  is diagonal,  $\mathbf{B}'$  also has its columns independent. Therefore  $\mathbf{R}$  always positive definite.

In summary, the VO estimates can be computed as

$$\begin{aligned} \delta\Delta\hat{\mathbf{x}}_{k,k-1} &= -\mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{k,k-1}} \sum_{i=1}^N [\mathbf{H}_{x,k,i}^T (\mathbf{H}_{z,k,i} \mathbf{R}_{\parallel,k,i} \mathbf{H}_{z,k,i}^T)^{-1} \mathbf{h}_k(\Delta\mathbf{x}_{k,k-1}^{(0)}, \mathbf{l}_{k,i}, \mathbf{l}_{k-1,i})] \\ \mathbf{P}_{\delta\Delta\hat{\mathbf{x}}_{k,k-1}} &= \left( \sum_{i=1}^N [\mathbf{H}_{x,k,i}^T (\mathbf{H}_{z,k,i} \mathbf{R}_{\parallel,k,i} \mathbf{H}_{z,k,i}^T)^{-1} \mathbf{H}_{x,k,i}] \right)^{-1} \\ \mathbf{R}_{\parallel,k,i} &= \text{diag}(\mathbf{R}_{1_{k,i} 1_{k,i}}, \mathbf{R}_{1_{k-1,i} 1_{k-1,i}}) \end{aligned} \quad (5.50)$$

and the corresponding cross-covariance matrix of the parameters with the one from the previous epoch can be obtained by

$$\begin{aligned}
\mathbf{R}_{k-1,k} &= \sum_{i=1}^N [\mathbf{G}_{k-1,i} \left( \frac{\partial \mathbf{h}_{k-1,i}}{\partial \mathbf{l}_{k-1,i}} \right) \mathbf{R}_{k-1,i} \mathbf{l}_{k-1,i} \left( \frac{\partial \mathbf{h}_{k-1,i}}{\partial \mathbf{l}_{k-1,i}} \right)^T \mathbf{G}_{k,i}^T] \\
&= \sum_{i=1}^N [\mathbf{B}_{k-1,k,i} \mathbf{R}_{k-1,i} \mathbf{l}_{k-1,i} \mathbf{B}_{k,k,i}^T] \\
\mathbf{G}_{k,i} &= \mathbf{P}_{\delta \Delta \hat{\mathbf{x}}_{k,k-1}} \mathbf{H}_{x,k,i}^T (\mathbf{H}_{z,k,i} \mathbf{R}_{\parallel,k,i} \mathbf{H}_{z,k,i}^T)^{-1}
\end{aligned} \tag{5.51}$$

wherein  $N$  is the number of the features.

### 5.3. Test results with the simulated data

A series of simulations were conducted to compare the performance of the standard Kalman filter (KF), the Kalman filter with the standard shaping filter for time-correlated measurements (KF-TC) and the proposed Kalman filter with *pairwise* time-correlated measurements (KF-PTC). Furthermore, the measurements are relative. The simulated data were used for the following reasons: (a) the measurement noise characteristics are known (b) the true state errors can be computed and (c) the proposed KF-PTC can be validated. The three models being studied can be summarized as

1. The standard Kalman filter [Kalman, 1960; Kalman-Bucy, 1961]

$$\begin{aligned}
\mathbf{x}_k &= \Phi_{k-1} \mathbf{x}_{k-1} + \Psi_{k-1} \mathbf{w}_{k-1}, \mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\
\mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{H}_{k-m} \mathbf{x}_{k-m} + \mathbf{v}_k, \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)
\end{aligned} \tag{5.52}$$

2. The Kalman filter with the time-correlated measurements [Bryson and Henrikson, 1968]

$$\begin{aligned}
\mathbf{x}_k &= \Phi_{k-1} \mathbf{x}_{k-1} + \Psi_{k-1} \mathbf{w}_{k-1}, \mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\
\mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{H}_{k-m} \mathbf{x}_{k-m} + \mathbf{v}_k \\
\mathbf{v}_k &= \psi_{k-m} \mathbf{v}_{k-m} + \boldsymbol{\eta}_{k-m}, \boldsymbol{\eta}_{k-m} \sim N(\mathbf{0}, \mathbf{Q}_{\boldsymbol{\eta}_{k-m}}) \\
E[\mathbf{v}_k \mathbf{v}_{k-m}^T] &= \mathbf{R}_{k,k-m} = \psi_{k-m} \mathbf{R}_{k-m}
\end{aligned} \tag{5.53}$$

3. The Kalman filter with the pairwise time-correlated measurements [Gopaul, 2014; Gopaul, 2017]

$$\begin{aligned}
\mathbf{x}_k &= \Phi_{k-1}\mathbf{x}_{k-1} + \Psi_{k-1}\mathbf{w}_{k-1}, \mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\
\mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{H}_{k-m}\mathbf{x}_{k-m} + \mathbf{C}_{k,k}\mathbf{v}'_k + \mathbf{C}_{k,k-m}\mathbf{v}'_{k-m}, \mathbf{v}'_k \sim N(\mathbf{0}, \mathbf{I}) \\
\mathbf{C}_{k,k-m} &= (\mathbf{C}_{k-m,k-m}^{-1}\mathbf{R}_{k-m,k})^T, \mathbf{C}_{1,0} = \mathbf{0} \\
\mathbf{C}_{k,k} &= chol(\mathbf{R}_{k,k} - \mathbf{C}_{k,k-m}\mathbf{C}_{k,k-m}^T), \mathbf{C}_{0,0} = \mathbf{0} \\
E[\mathbf{v}_k\mathbf{v}_{k-m}^T] &= \mathbf{R}_{k,k-m}
\end{aligned} \tag{5.54}$$

In this example, a 2D planar trajectory shown in Figure 5.3(left) was simulated. Figure 5.3 (right) shows the velocity and heading profiles. Furthermore, 2D image ranging and IMU measurements were constructed. The 2D n- and b- frames were considered here.

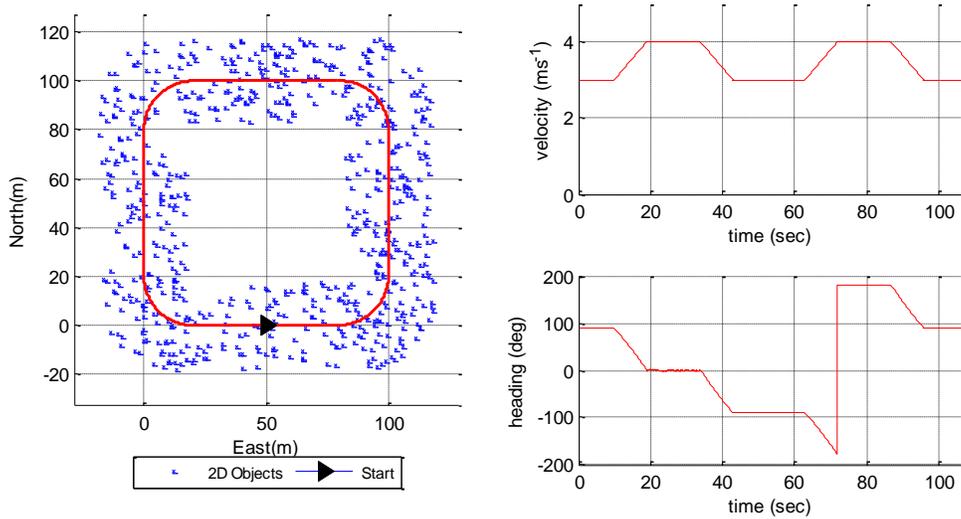


Figure 5.3 The top view of the vehicle trajectory with visible landmarks (left) and the velocity and heading profiles (right)

The IMU measurements (two accelerometers and one gyroscope) were acquired at 100Hz. The power spectral density of the process noises of the accelerometers and gyroscope were set to 1.0 m/s/ $\sqrt{\text{hr}}$  and 4.5 deg/ $\sqrt{\text{hr}}$ , respectively. The data rate of the

image measurements was 10Hz and the standard deviation of the 2D feature points in (5.41) was set to 0.10m. Both the image and IMU measurements are resolved in the b-frame. The 2D VO were computed using the visible features in Figure 5.3 (left). The 2D VO measurement equation is given by

$$\mathbf{l}_{k,i}^b - \mathbf{v}_{k,i}^b = \mathbf{C}_{b(k-m)}^{b(k)} (\mathbf{l}_{k-m,i}^b - \mathbf{v}_{k-m,i}^b) + \Delta \mathbf{X}_{k,k-m}^b \quad (5.55)$$

where  $\mathbf{l}_{k,i}^b$  [m] and  $\mathbf{l}_{k-m,i}^b$  [m] are the measurement vectors of object  $i$  in epochs  $k$  and  $k-m$  respectively,  $\mathbf{v}_{k,i}^b$  [m] and  $\mathbf{v}_{k-m,i}^b$  [m] are the measurement noise vectors,  $\mathbf{C}_{b(k-m)}^{b(k)}$  is the DCM between  $k$  and  $k-m$ , and  $\Delta \mathbf{X}_{k,k-m}^b$  [m] is the position difference. Figure 5.4 shows the standard deviations of the 2D VO estimates and the number of the used features.

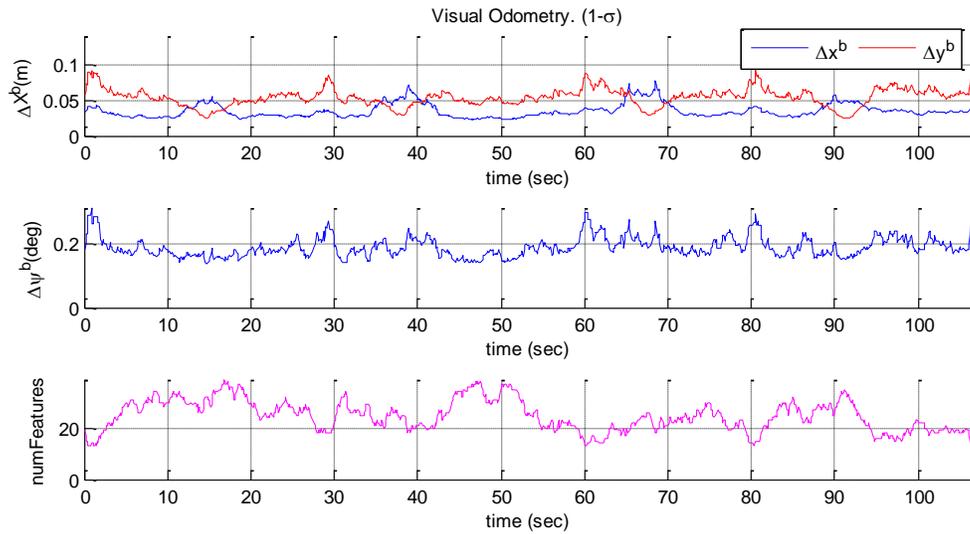


Figure 5.4 The 2D VO standard deviations and the numbers of the features

At epoch  $k$ , the system model in the n-frame is

$$\begin{pmatrix} \delta\dot{X}^n \\ \delta\dot{Y}^n \\ \delta v_x^n \\ \delta v_y^n \\ \delta\dot{\psi}_z^n \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & a_y^n \\ 0 & 0 & 0 & 0 & -a_x^n \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta X^n \\ \delta Y^n \\ \delta v_x^n \\ \delta v_y^n \\ \delta\psi_z^n \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^n & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} w_{a_y^b} \\ w_{a_x^b} \\ w_{g_z^b} \end{pmatrix} \quad (5.56)$$

where  $(\delta X^n, \delta Y^n)^T$  [m],  $(\delta v_x^n, \delta v_y^n)^T$  [m] and  $\delta\psi_z^n$  [rad] are the vehicle's position, velocity and heading error states, respectively,  $(a_x^n, a_y^n)^T$  [ms<sup>-2</sup>] are the accelerations,  $(w_{a_x^b}, w_{a_y^b})^T$  [ms<sup>-2</sup>] are the accelerometer process noises,  $w_{g_z^b}$  [rads<sup>-1</sup>] is the gyroscope process noise, and

$$\mathbf{C}_b^n = \begin{pmatrix} \cos(\psi_z^n) & -\sin(\psi_z^n) \\ \sin(\psi_z^n) & \cos(\psi_z^n) \end{pmatrix}. \quad (5.57)$$

Equation (5.56) was derived from (3.56) by (a) setting the  $\boldsymbol{\omega}_{en^c}^{n^c}$ ,  $\boldsymbol{\omega}_{ie}^{n^c}$ ,  $\boldsymbol{\omega}_{in^c}^{n^c}$  and  $\delta\mathbf{g}^{n^c}$  vectors to zero and (b) removing the  $\delta Z^{n^c}$ ,  $\delta v_z^{n^c}$ ,  $\delta\psi_x^{n^c}$  and  $\delta\psi_y^{n^c}$  states from the state vector. The measurement model is given by

$$\begin{pmatrix} \Delta X_{VO,k,k-m}^b \\ \Delta Y_{VO,k,k-m}^b \\ \Delta\psi_{VO,k,k-m}^b \end{pmatrix} = \begin{pmatrix} (\mathbf{C}_{b(k),INS}^n)^T \begin{pmatrix} X_{INS,k}^n - X_{INS,k-m}^n \\ Y_{INS,k}^n - Y_{INS,k-m}^n \\ \psi_{INS,k}^n - \psi_{INS,k-m}^n \end{pmatrix} \\ \psi_{INS,k}^n - \psi_{INS,k-m}^n \end{pmatrix} + \begin{pmatrix} v_{\Delta X_{VO,k,k-m}^b} \\ v_{\Delta Y_{VO,k,k-m}^b} \\ v_{\Delta\psi_{VO,k,k-m}^b} \end{pmatrix} \quad (5.58)$$

wherein  $m = 10$ , specifically in this simulation.

Monte Carlo (MC) simulations were used to compare the performance of the three implemented Kalman filters. Each algorithm was run for 100 times. The true position and heading errors were computed for each run. Then the root-mean-square errors (RMSE) across the 100 runs were computed at every epoch. The resulting error bounds were then compared with the estimated standard deviations. Figure 5.5 (left) compares the pose

RMS errors computed against the true poses in (a) KF, (b) KF-TC, and (c) the proposed KF-PTC. Clearly the true errors from KF-PTC are smaller than the ones from KF and KF-TC.

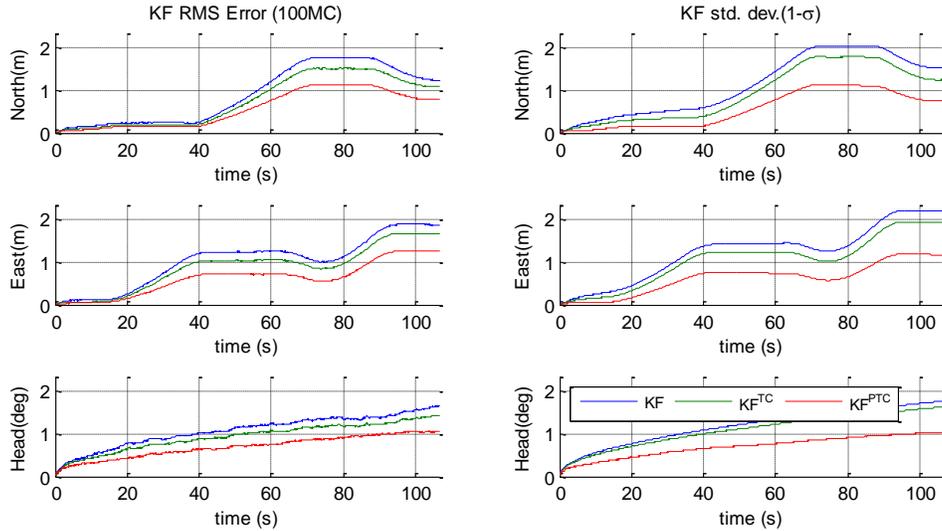


Figure 5.5 True position and heading RMSE (left) and the estimated position and heading standard deviations (right)

Table 5.1 The dimensions of the state vectors in the KF, KF-ST and KF-PTC

States	KF	KF-TC	KF-PTC
Navigation	5	5	5
Relative measurements	3	3	3
Shaping filter	0	3	6
<i>Total</i>	<i>8</i>	<i>11</i>	<i>14</i>

The Figure 5.5 (right) shows that the corresponding standard deviations estimated from the three filters and can conclude that KF-PTC standard deviations are smaller than the ones from KF and KF-ST. Figure 5.5 also shows that the standard deviations of KF and KF-TC are larger than the true errors, while the standard deviations of KF-PTC closely match the corresponding true errors. Therefore, the solutions from KF and KF-TC are optimistic, while KF-PTC solutions are reasonably optimal. Table 5.1 compares the

dimensions of the state vectors in three Kalman filters. As expected, the dimension of the state vector in KF-PTC is larger than the ones in KF-ST and KF.

#### 5.4. Loosely-coupled stereo VO-aided inertial navigation

This section derives the measurement model for the LC VO-aided inertial navigation. Here psi-error model (3.56) is used to model the measurements. It assumed that (a) the initial position and velocity are obtained by external means (e.g. GNSS) and (b) the alignment procedure has been completed. Furthermore, the VO solution is generated in the b-frame. The VO measurement equations are given by

$$\begin{aligned}\Delta \mathbf{X}_{VO,k,k-m}^b &= (\mathbf{C}_{b(k),INS}^{n^c})^T \mathbf{D}_r^{-1} (\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m}) \\ \mathbf{C}_{b(k-m),VO}^{b(k)} &= (\mathbf{C}_{b(k),INS}^{n^c})^T \mathbf{C}_{b(k-m),INS}^{n^c}\end{aligned}\quad (5.59)$$

where  $\mathbf{D}_r^{-1} = \text{diag}((R_N + h), (R_e + h)c\varphi, -1)$ . The linearization of the first equation in (5.59) gives

$$\begin{aligned}\delta \Delta \mathbf{X}_{VO,k}^b &= (\mathbf{C}_{b(k),INS}^{n^c})^T \mathbf{D}_r^{-1} \delta \mathbf{r}_{INS,k} - (\mathbf{C}_{b(k),INS}^{n^c})^T \mathbf{D}_r^{-1} \delta \mathbf{r}_{INS,k-m} \cdots \\ &\quad - (\mathbf{C}_{b(k),INS}^{n^c})^T (\mathbf{D}_r^{-1} (\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m})) \times \boldsymbol{\Psi}_{INS,k} \cdots \\ &\quad + (\mathbf{C}_{b(k),INS}^{n^c})^T (\delta \mathbf{D}_r^{-1} (\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m}))\end{aligned}\quad (5.60)$$

where

$$\delta \mathbf{D}_r^{-1} \approx \begin{pmatrix} \delta h_k & 0 & 0 \\ 0 & c\varphi_k \delta h_k - h_k s\varphi_k \delta \varphi_k & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (5.61)$$

The position difference  $(\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m})$  between two consecutive frames is expected to be small i.e. less than 3.0 m. Hence, the term  $\delta \mathbf{D}_r^{-1} (\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m})$  in (5.60) will be negligibly small i.e.,

$$\begin{aligned}
\delta \mathbf{D}_r^{-1}(\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m}) &= \begin{pmatrix} \delta h_k & 0 & 0 \\ 0 & c\varphi_k \delta h_k - h_k s\varphi_k \delta \varphi_k & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta \varphi_{k,k-m} \\ \Delta \lambda_{k,k-m} \\ \Delta h_{k,k-m} \end{pmatrix} \\
&= \begin{pmatrix} \delta h_k \Delta \varphi_{k,k-m} \\ (c\varphi_k \delta h_k - h_k s\varphi_k \delta \varphi_k) \Delta \lambda_{k,k-m} \\ 0 \end{pmatrix} \approx \mathbf{0}
\end{aligned} \tag{5.62}$$

Therefore,  $(\mathbf{C}_{b(k),INS}^{n^c})^T (\delta \mathbf{D}_r^{-1}(\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m}))$  is omitted.  $\mathbf{D}_r^{-1} \delta \mathbf{r}_{INS,k}$  can be expressed as  $\delta \mathbf{X}_{INS,k}^{n^c}$ . (5.60) can be simplified to

$$\begin{aligned}
\delta \Delta \mathbf{X}_{VO,k}^b &= (\mathbf{C}_{b(k),INS}^{n^c})^T \delta \mathbf{X}_{INS,k}^{n^c} - (\mathbf{C}_{b(k),INS}^{n^c})^T \delta \mathbf{X}_{INS,k-m}^{n^c} \dots \\
&\quad - (\mathbf{C}_{b(k),INS}^{n^c})^T (\mathbf{D}_r^{-1}(\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-m})) \times \boldsymbol{\Psi}_{INS,k}
\end{aligned} \tag{5.63}$$

Further, the linearization of the 2<sup>nd</sup> equation (the orientation change measurement equation) in (5.59) gives

$$\boldsymbol{\Phi}_{VO,k} = -(\mathbf{C}_{b(k),INS}^{n^c})^T (\boldsymbol{\Psi}_{INS,k} - \boldsymbol{\Psi}_{INS,k-m}) \tag{5.64}$$

The system model in discrete time is written as

$$\begin{pmatrix} \delta \mathbf{x}_{INS,k} \\ \delta \mathbf{x}'_{INS,k-m} \\ \mathbf{v}'_k \\ \mathbf{v}'_{k-m} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}_{k-1} & \mathbf{O}_{15 \times 6} & \mathbf{O}_{15 \times 6} & \mathbf{O}_{15 \times 6} \\ \mathbf{O}_{6 \times 15} & \mathbf{I}_{6 \times 6} & \mathbf{O}_{6 \times 6} & \mathbf{O}_{6 \times 6} \\ \mathbf{O}_{6 \times 15} & \mathbf{O}_{6 \times 6} & \mathbf{O}_{6 \times 6} & \mathbf{O}_{6 \times 6} \\ \mathbf{O}_{6 \times 15} & \mathbf{O}_{6 \times 6} & \mathbf{I}_{6 \times 6} & \mathbf{O}_{6 \times 6} \end{pmatrix} \begin{pmatrix} \delta \mathbf{x}_{INS,k-1} \\ \delta \mathbf{x}'_{INS,k-m} \\ \mathbf{v}'_{k-m} \\ \mathbf{v}'_{k-2m} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\Psi}_{k-1} \mathbf{w}_{k-1} \\ \mathbf{0}_{6 \times 1} \\ \boldsymbol{\eta}_{k-1} \\ \mathbf{0}_{6 \times 1} \end{pmatrix} \tag{5.65}$$

wherein  $\delta \mathbf{x}_{INS,k} = (\delta \mathbf{X}_{INS,k}^{n^c}, \delta \mathbf{v}_{INS,k}^{n^c}, \boldsymbol{\Psi}_{INS,k}, \mathbf{b}_a^b, \mathbf{b}_g^b)^T$ ,  $\delta \mathbf{x}'_{k-m,INS} = (\delta \mathbf{X}_{INS,k-m}^{n^c}, \boldsymbol{\Psi}_{INS,k-m})^T$ ,

$\mathbf{w}_{k-1} \sim N(\mathbf{0}, \mathbf{Q}_{k-1})$  and  $\boldsymbol{\eta}_{k-1} \sim N(\mathbf{0}, \mathbf{I})$ . The measurement model is given as

$$\delta \mathbf{z}_{VO,k} = \begin{pmatrix} \tilde{\mathbf{H}}_k & \tilde{\mathbf{H}}_{k-m} & \mathbf{C}_{k,k} & \mathbf{C}_{k,k-m} \end{pmatrix} \begin{pmatrix} \delta \mathbf{x}_{INS,k} \\ \delta \mathbf{x}'_{INS,k-m} \\ \mathbf{v}'_k \\ \mathbf{v}'_{k-m} \end{pmatrix} + \mathbf{0}_{6 \times 1} \tag{5.66}$$

wherein

$$\delta \mathbf{z}_{VO,k} = \begin{pmatrix} \delta \Delta \mathbf{X}_{VO,k}^b \\ \boldsymbol{\Psi}_{VO,k} \end{pmatrix}, \quad (5.67)$$

$$\tilde{\mathbf{H}}_k = \begin{pmatrix} \mathbf{C}_{n^c}^{b(k)} & \mathbf{O}_{3 \times 3} & -\mathbf{C}_{n^c}^{b(k)} (\mathbf{D}_r^{-1} (\mathbf{r}_k - \mathbf{r}_{k-m})) \times & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & -\mathbf{C}_{n^c}^{b(k)} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \end{pmatrix}, \quad (5.68)$$

and

$$\tilde{\mathbf{H}}_{k-m} = \begin{pmatrix} -\mathbf{C}_{n^c}^{b(k)} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{C}_{n^c}^{b(k)} \end{pmatrix}. \quad (5.69)$$

With the formulation of the system and measurement models in (5.65) and (5.66), the VO measurements can be easily integrated with an existing GNSS/IMU integration and smoothers without any modifications to the architecture. For instance, in a LC GNSS (3.59) and LC VO integrated inertial navigation scheme, the measurement equation can be written as:

$$\begin{pmatrix} \delta \mathbf{z}_{GNSS,k} \\ \delta \mathbf{z}_{VO,k} \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{GNSS,k} & \mathbf{O}_{7 \times 6} & \mathbf{O}_{7 \times 6} & \mathbf{O}_{7 \times 6} \\ \tilde{\mathbf{H}}_k & \tilde{\mathbf{H}}_{k-m} & \mathbf{C}_{k,k} & \mathbf{C}_{k,k-m} \end{pmatrix} \begin{pmatrix} \delta \mathbf{x}_{INS,k} \\ \delta \mathbf{x}'_{INS,k-m} \\ \mathbf{v}'_k \\ \mathbf{v}'_{k-m} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_{GNSS,k} \\ \mathbf{0}_{6 \times 1} \end{pmatrix} \quad (5.70)$$

The LC stereo VO aided-inertial navigation with pairwise time correlated measurements is outlined in Figure 5.6.

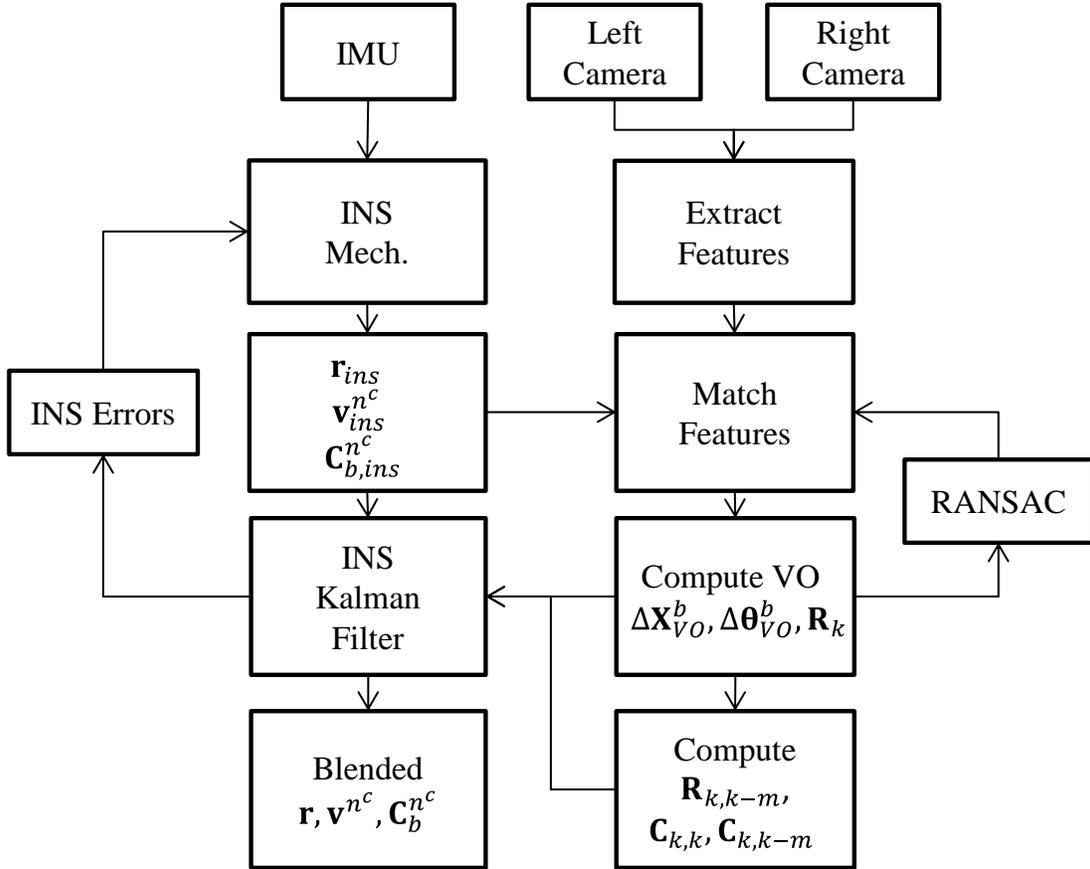


Figure 5.6 Loosely-coupled VO aided-inertial navigation with pairwise time correlated measurements.

The LC stereo VO-aided inertial navigation field test results and analysis are presented in Section 7.4.

## 5.5. Summary

This chapter presented a novel Kalman filter algorithm for processing pairwise time-correlated measurements and specifically applied it to the visual odometry-aided inertial navigation in a loosely-coupled manner. The coefficients of the constructed shaping filter for modeling the time-correlation are Cholesky factors obtained from the measurement variance-covariance matrices. Results using a simulated dataset showed that the proposed Kalman filter with the pairwise time-correlated measurement noise performed better than

the standard Kalman filter and the Kalman filter with the conventional shaping filter. Furthermore, the covariance matrix of a state vector provides an accurate description of the uncertainty of the state vector. The Kalman filter is in the standard form and therefore can easily be adapted in the current GNSS-aided inertial navigation integration architecture. This method is primarily designed for the loosely-coupled VO-aided inertial navigation. But, it can also be applied to the tightly coupled counterpart. However, the dimension of the state vector will be augmented by the measurement noise vector for each individual image measurement and become impractically very large. Two problems may appear with the algorithm: firstly, its benefit may not be apparent if the correlation between the two consecutive pose estimates is weak and secondly, the solution can be significantly degraded if the measurement noise is non-Gaussian which can be the case with real world applications. Results with real datasets are presented in Section 7.4.

## 6. Tightly-coupled stereo multi-frame visual odometry aided-inertial navigation

This chapter is mainly based on the following publication:

Gopaul, Nilesh S.; Wang, Jianguo and Hu, Baoxin (2015), *Multi-frame Visual Odometry in Image-Aided Inertial Navigation System*, In: Sun J., Liu J., Fan S., Lu X. (eds) China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III, Lecture Notes in Electrical Engineering, vol 342, Springer, Berlin, Heidelberg, pp. 649-658, DOI: [https://doi.org/10.1007/978-3-662-46632-2\\_57](https://doi.org/10.1007/978-3-662-46632-2_57).

### 6.1. Introduction

This chapter presents a novel stereo multi-frame aided inertial navigation algorithm for reducing drifts in position and orientation in poor GNSS and/or GNSS-denied environments. Usually, the image aided inertial navigation based on the visual odometry uses the tracked features only from a pair of the consecutive image frames. The proposed method integrates the features tracked from multiple overlapping image frames to obtain solution accuracy improvement, which is referred to as stereo Multi-Frame Visual Odometry (MFVO) with respect to pairwise VO in previous chapter. Basically, the MFVO measurement model is derived from the SLAM measurement equation system. In particular, MFVO algebraically eliminates the landmark position parameters included in SLAM by timely-differencing the measurements between two consecutive epochs. Furthermore, the measurement updates in the Kalman filter can be performed using a sequential de-correlation mechanism, since the time-differenced measurements are timely-correlated. Monte Carlo simulations show that the pose estimates from the MFVO

are identical with the ones from SLAM. In comparison with the SLAM, the proposed MFVO method uses less computation resources, especially when the number of features in view becomes large.

This chapter is organized as follows: Section 6.2 develops the system and measurement models in a Kalman filter for the proposed MFVO method. Then, the results from simulation tests are shown in Section 6.3 to validate the development presented in Section 6.2. Section 6.4 presents the system and measurement models for MFVO aided-inertial navigation in the tightly coupled integration architecture. At the end, Section 6.5 concludes the chapter. The field test results and analysis are presented in Section 7.5 .

## 6.2. Stereo Multi-frame visual odometry

The goal of the proposed MFVO algorithm is to develop a measurement model that can be run in the navigation Kalman filter sequentially to obtain optimal solutions. The term ‘visual odometry’ is used here because the focus is on image measurements. However, the algorithm is a generic one applicable to any EKF (extended Kalman filter) based SLAM application.

The derivation begins with the EKF-SLAM model. The system model for  $N$  landmark states at epoch  $k$  is given as follows

$$\begin{pmatrix} \mathbf{x}_k \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{pmatrix} \quad (6.1)$$

wherein  $\mathbf{x}_k$  is the vehicle's position and orientation state vector,  $\mathbf{m}_i$  ( $i = 1, 2, \dots, N$ ) are the position states of the landmarks and  $\mathbf{w}_{k-1}$  is the process noise vector and conforms to  $N(\mathbf{0}, \mathbf{Q}_{k-1})$ . The measurement model is given as

$$\begin{pmatrix} \mathbf{z}_{k,1} \\ \vdots \\ \mathbf{z}_{k,N} \end{pmatrix} = \begin{pmatrix} \mathbf{h}_{k,1}(\mathbf{x}_k, \mathbf{m}_1) \\ \vdots \\ \mathbf{h}_{k,N}(\mathbf{x}_k, \mathbf{m}_N) \end{pmatrix} + \begin{pmatrix} \mathbf{v}_{k,1} \\ \vdots \\ \mathbf{v}_{k,N} \end{pmatrix} \quad (6.2)$$

where  $\mathbf{z}_{k,i}$  is the measurement vector for landmark  $i$ ,  $\mathbf{h}_{k,i}(\mathbf{x}_k, \mathbf{m}_i)$  is the non-linear measurement model of the vehicle's trajectory states and the landmark states, and  $\mathbf{v}_{k,i}$  is the measurement noise vector and conforms to  $N(\mathbf{0}, \mathbf{R}_{k,i})$  for  $i = 1, 2, \dots, N$ . The stereo visual SLAM measurement equation (described in Section 3.6.4.4) is

$$\mathbf{z}_{k,i}^b = (\mathbf{C}_{b(k)}^n)^T (\mathbf{m}_i^n - \mathbf{X}_k^n) + \mathbf{v}_{k,i} \quad (6.3)$$

Now consider the measurement model for landmark  $i$  at two consecutive epochs  $k-1$  and  $k$ :

$$\begin{aligned} \mathbf{z}_{k-1,i} &= (\mathbf{C}_{b(k-1)}^n)^T (\mathbf{m}_i^n - \mathbf{X}_{k-1}^n) + \mathbf{v}_{k-1,i} \\ \mathbf{z}_{k,i} &= (\mathbf{C}_{b(k)}^n)^T (\mathbf{m}_i^n - \mathbf{X}_k^n) + \mathbf{v}_{k,i} \end{aligned} \quad (6.4)$$

Next, algebraically cancel the landmark position vector  $\mathbf{m}_i$  in (6.4) to obtain the following implicit measurement equation:

$$\mathbf{C}_{b(k)}^n (\mathbf{z}_{k,i} - \mathbf{v}_{k,i}) - \mathbf{C}_{b(k-1)}^n (\mathbf{z}_{k-1,i} - \mathbf{v}_{k-1,i}) + \mathbf{X}_k^n - \mathbf{X}_{k-1}^n = \mathbf{0}. \quad (6.5)$$

In short form (6.5) can be expressed as

$$\tilde{\mathbf{h}}_{k,i}(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{z}_{k,i}, \mathbf{z}_{k-1,i}, \mathbf{v}_{k,i}, \mathbf{v}_{k-1,i}) = \mathbf{0}. \quad (6.6)$$

The linearization of (6.6) using first order Taylor series expansion gives:

$$\begin{aligned}
-\tilde{\mathbf{h}}_{k,i}(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{z}_{k,i}, \mathbf{z}_{k-1,i}, \mathbf{0}, \mathbf{0}) &= \mathbf{H}_{k,k,i} \delta \mathbf{x}_k + \mathbf{H}_{k,k-1,i} \delta \mathbf{x}_{k-1} \dots \\
&+ \mathbf{M}_{k,k,i} \mathbf{v}_{k,i} + \mathbf{M}_{k,k-1,i} \mathbf{v}_{k-1,i}
\end{aligned} \tag{6.7}$$

where  $\mathbf{H}_{k,k,i}$ ,  $\mathbf{H}_{k,k-1,i}$ ,  $\mathbf{M}_{k,k,i}$  and  $\mathbf{M}_{k,k-1,i}$  are the Jacobian matrices associated with  $\mathbf{x}_k$ ,  $\mathbf{x}_{k-1}$ ,  $\mathbf{v}_{k,i}$  and  $\mathbf{v}_{k-1,i}$ , respectively. For simplicity, write  $\delta \tilde{\mathbf{z}}_k = -\tilde{\mathbf{h}}_k(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{z}_{k,i}, \mathbf{z}_{k-1,i}, \mathbf{0}, \mathbf{0})$  and  $\tilde{\mathbf{v}}_{k,i} = \mathbf{M}_{k,k,i} \mathbf{v}_{k,i} + \mathbf{M}_{k,k-1,i} \mathbf{v}_{k-1,i}$ . Then, one has the linearized measurement model as follows

$$\delta \tilde{\mathbf{z}}_{k,i} = \mathbf{H}_{k,k,i} \delta \mathbf{x}_k + \mathbf{H}_{k,k-1,i} \delta \mathbf{x}_{k-1} + \tilde{\mathbf{v}}_{k,i}, \tilde{\mathbf{v}}_{k,i} \sim N(\mathbf{0}, \tilde{\mathbf{R}}_{k,k,i}) \tag{6.8}$$

and the variance-covariance matrix of the measurement noise vector  $\tilde{\mathbf{v}}_{k,i}$

$$\tilde{\mathbf{R}}_{k,k,i} = \mathbf{M}_{k,k,i} \mathbf{R}_{k,i} \mathbf{M}_{k,k,i}^T + \mathbf{M}_{k,k-1,i} \mathbf{R}_{k-1,i} \mathbf{M}_{k,k-1,i}^T. \tag{6.9}$$

Consider (6.6) at two consecutive epochs  $k-1$  and  $k$ :

$$\begin{aligned}
\tilde{\mathbf{h}}_{k-1,i}(\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \mathbf{z}_{k-1,i}, \mathbf{z}_{k-2,i}, \mathbf{v}_{k-1,i}, \mathbf{v}_{k-2,i}) &= \mathbf{0} \\
\tilde{\mathbf{h}}_{k,i}(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{z}_{k,i}, \mathbf{z}_{k-1,i}, \mathbf{v}_{k,i}, \mathbf{v}_{k-1,i}) &= \mathbf{0}
\end{aligned} \tag{6.10}$$

which has involved the measurement noise vector  $\mathbf{v}_{k-1,i}$  in both  $\tilde{\mathbf{h}}_{k-1,i}(\cdot)$  and  $\tilde{\mathbf{h}}_{k,i}(\cdot)$ .

Therefore, the derived measurement noise vectors  $\tilde{\mathbf{v}}_{k,i}$  and  $\tilde{\mathbf{v}}_{k-1,i}$  are correlated with respect to time

$$\begin{aligned}
E[\tilde{\mathbf{v}}_{k,i} \tilde{\mathbf{v}}_{k-1,i}^T] &= E[(\mathbf{M}_{k,k,i} \mathbf{v}_{k,i} + \mathbf{M}_{k,k-1,i} \mathbf{v}_{k-1,i})(\mathbf{M}_{k-1,k-1,i} \mathbf{v}_{k-1,i} + \mathbf{M}_{k-1,k-2,i} \mathbf{v}_{k-2,i})^T] \\
&= \mathbf{M}_{k,k,i} E[\mathbf{v}_{k,i} \mathbf{v}_{k-1,i}^T] \mathbf{M}_{k-1,i,k-1,i}^T + \mathbf{M}_{k,k-1,i} E[\mathbf{v}_{k-1,i} \mathbf{v}_{k-1,i}^T] \mathbf{M}_{k-1,i,k-1,i}^T \dots \\
&\quad + \mathbf{M}_{k,k,i} E[\mathbf{v}_{k,i} \mathbf{v}_{k-2,i}^T] \mathbf{M}_{k-1,i,k-2,i}^T \dots \\
&\quad + \mathbf{M}_{k,k-1,i} E[\mathbf{v}_{k-1,i} \mathbf{v}_{k-2,i}^T] \mathbf{M}_{k-1,i,k-2,i}^T \dots \\
&= \mathbf{O} + \mathbf{M}_{k,k-1,i} E[\mathbf{v}_{k-1,i} \mathbf{v}_{k-1,i}^T] \mathbf{M}_{k-1,i,k-1,i}^T + \mathbf{O} + \mathbf{O} \\
&= \mathbf{M}_{k,k-1,i} E[\mathbf{v}_{k-1,i} \mathbf{v}_{k-1,i}^T] \mathbf{M}_{k-1,i,k-1,i}^T \\
&= \mathbf{M}_{k,k-1,i} \mathbf{R}_{k-1,i} \mathbf{M}_{k-1,i,k-1,i}^T \\
&= \tilde{\mathbf{R}}_{k,k-1,i}
\end{aligned} \tag{6.11}$$

If the feature  $i$  is first observed at epoch  $k - j$ , then all the linearized measurement equations up to and including the epoch  $k$  can be written as

$$\begin{aligned}
\delta\tilde{\mathbf{z}}_{k-j+1,i} &= \mathbf{H}_{k-j+1,k-j+1,i}\delta\mathbf{x}_{k-j+1} + \mathbf{H}_{k-j+1,k-j,i}\delta\mathbf{x}_{k-j} + \tilde{\mathbf{v}}_{k-j+1,i}, \dots \\
&\quad \tilde{\mathbf{v}}_{k-j+1,i} \sim N(\mathbf{0}, \tilde{\mathbf{R}}_{k-j+1,k-j+1,i}) \\
&\quad \vdots \\
\delta\tilde{\mathbf{z}}_{k-2,i} &= \mathbf{H}_{k-2,k-2,i}\delta\mathbf{x}_{k-2} + \mathbf{H}_{k-2,k-3,i}\delta\mathbf{x}_{k-3} + \tilde{\mathbf{v}}_{k-2,i}, \tilde{\mathbf{v}}_{k-2,i} \sim N(\mathbf{0}, \tilde{\mathbf{R}}_{k-2,k-2,i}) \\
\delta\tilde{\mathbf{z}}_{k-1,i} &= \mathbf{H}_{k-1,k-1,i}\delta\mathbf{x}_{k-1} + \mathbf{H}_{k-1,k-2,i}\delta\mathbf{x}_{k-2} + \tilde{\mathbf{v}}_{k-1,i}, \tilde{\mathbf{v}}_{k-1,i} \sim N(\mathbf{0}, \tilde{\mathbf{R}}_{k-1,k-1,i}) \\
\delta\tilde{\mathbf{z}}_{k,i} &= \mathbf{H}_{k,k,i}\delta\mathbf{x}_k + \mathbf{H}_{k,k-1,i}\delta\mathbf{x}_{k-1} + \tilde{\mathbf{v}}_{k,i}, \tilde{\mathbf{v}}_{k,i} \sim N(\mathbf{0}, \tilde{\mathbf{R}}_{k,k,i})
\end{aligned} \tag{6.12}$$

Now concatenate all the measurement noise vector as time series into a vector  $\tilde{\mathbf{v}}_i$  and construct its corresponding variance-covariance matrix  $\tilde{\mathbf{R}}_i$ :

$$\tilde{\mathbf{v}}_i = \begin{pmatrix} \tilde{\mathbf{v}}_{k-j+1,i} \\ \vdots \\ \tilde{\mathbf{v}}_{k-2,i} \\ \tilde{\mathbf{v}}_{k-1,i} \\ \tilde{\mathbf{v}}_{k,i} \end{pmatrix}, \tilde{\mathbf{R}} = \begin{pmatrix} \tilde{\mathbf{R}}_{k-j+1,k-j+1,i} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \tilde{\mathbf{R}}_{k-2,k-2,i} & \tilde{\mathbf{R}}_{k-2,k-1,i} & \mathbf{0} \\ \mathbf{0} & \cdots & \tilde{\mathbf{R}}_{k-1,k-2,i} & \tilde{\mathbf{R}}_{k-1,k-1,i} & \tilde{\mathbf{R}}_{k-1,k,i} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{R}}_{k,k-1,i} & \tilde{\mathbf{R}}_{k,k,i} \end{pmatrix}. \tag{6.13}$$

The standard Kalman filter runs under the assumption that the measurement noise vector is white, i.e. normally distributed with zero means, and independent to each other from epoch to epoch. However, the measurement noises in (6.13) are pairwise time-correlated between two consecutive epochs. One way to de-correlate the measurement noise vector is to use the inverse matrix of the Cholesky factors of the variance-covariance matrix. For simplicity, all the vectors and matrices in (6.12) are concatenated to obtain the following short-form with dropping out the feature index  $i$ :

$$\delta\tilde{\mathbf{z}} = \mathbf{H}\delta\mathbf{x} + \tilde{\mathbf{v}}, \tilde{\mathbf{v}} \sim N(\mathbf{0}, \tilde{\mathbf{R}}) \tag{6.14}$$

If  $\tilde{\mathbf{R}}$  is positive definite, it can be decomposed using the Cholesky factorization [Grewal and Andrews, 2001; Bierman, 2006; etc.]:

$$\tilde{\mathbf{R}} = \mathbf{C}\mathbf{C}^T \quad (6.15)$$

wherein the Cholesky factor matrix  $\mathbf{C}$  is a unique, real lower triangular matrix with its diagonal elements strictly positive. By multiplying (6.14) with the matrix  $\mathbf{L}$ , the inverse of  $\mathbf{C}$ , one obtains

$$\begin{aligned} \delta\tilde{\mathbf{z}} &= \mathbf{H}\delta\mathbf{x} + \tilde{\mathbf{v}} \\ \mathbf{C}^{-1}\delta\tilde{\mathbf{z}} &= \mathbf{C}^{-1}\mathbf{H}\delta\mathbf{x} + \mathbf{C}^{-1}\tilde{\mathbf{v}} \\ \mathbf{L}\delta\tilde{\mathbf{z}} &= \mathbf{LH}\delta\mathbf{x} + \mathbf{L}\tilde{\mathbf{v}} \\ \delta\mathbf{z}' &= \mathbf{H}'\delta\mathbf{x} + \mathbf{v}' \end{aligned} \quad (6.16)$$

wherein the derived measurement vector  $\mathbf{v}'$  becomes un-correlated and has the identity matrix as its variance matrix

$$\begin{aligned} \mathbf{R}' &= E[(\mathbf{v}')(\mathbf{v}')^T] = \mathbf{C}^{-1}E[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^T]\mathbf{C}^{-T} \\ &= \mathbf{C}^{-1}\tilde{\mathbf{R}}\mathbf{C}^{-T} = \mathbf{C}^{-1}\mathbf{C}\mathbf{C}^T\mathbf{C}^{-T} \\ &= \mathbf{I} \end{aligned} \quad (6.17)$$

In this case, the variance-covariance matrix  $\tilde{\mathbf{R}}$  in (6.13) is sparse, actually a lower bidiagonal block matrix, i.e. only its first lower off-diagonal blocks are non-zero. Therefore, the Cholesky factor matrix  $\mathbf{C}$  is also in the same lower bidiagonal block structure

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{O} & \cdots & \mathbf{C}_{k-2,k-2} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \cdots & \mathbf{C}_{k-1,k-2} & \mathbf{C}_{k-1,k-1} & \mathbf{O} \\ \mathbf{O} & \cdots & \mathbf{O} & \mathbf{C}_{k,k-1} & \mathbf{C}_{k,k} \end{pmatrix} \quad (6.18)$$

Furthermore, the block factor matrices  $\mathbf{C}_{k,k}$  and  $\mathbf{C}_{k,k-1}$  can be obtained sequentially as described in Algorithm 6.1.

Algorithm 6.1 Sequential Cholesky factorization of  $\tilde{\mathbf{R}}$  at epoch  $k$

$$\begin{array}{l}
\mathbf{C}_{k-j,k-j} = \mathbf{O} \\
\mathbf{C}_{k-j+1,k-j} = \mathbf{O} \\
\text{for } i = (k-j+1), \dots, k \{ \\
\quad \mathbf{C}_{i,i-1} = (\mathbf{C}_{i-1,i-1}^{-1} \tilde{\mathbf{R}}_{i-1,i})^T \\
\quad \mathbf{C}_{i,i} = \text{chol}(\tilde{\mathbf{R}}_{i,i} - \mathbf{C}_{i,i-1} \mathbf{C}_{i,i-1}^T) \\
\}
\end{array}$$

However, the matrix  $\mathbf{L}$  is a full lower block (triangular) matrix:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{L}_{k-2,k-j+1} & \cdots & \mathbf{L}_{k-2,k-2} & \mathbf{O} & \mathbf{O} \\ \mathbf{L}_{k-1,k-j+1} & \cdots & \mathbf{L}_{k-1,k-2} & \mathbf{L}_{k-1,k-1} & \mathbf{O} \\ \mathbf{L}_{k,k-j+1} & \cdots & \mathbf{L}_{k,k-2} & \mathbf{L}_{k,k-1} & \mathbf{L}_{k,k} \end{pmatrix}. \quad (6.19)$$

Since the matrix  $\mathbf{C}$  is lower triangular, the block matrices  $\mathbf{L}_{k,k-j+1} \dots \mathbf{L}_{k,k}$  at epoch  $k$  can be obtained sequentially. Therefore, the process to obtain  $\mathbf{L}_{k,k-j+1} \dots \mathbf{L}_{k,k}$  can be simplified. Algorithm 6.2 describes the algorithm for computing  $\mathbf{L}_{k,k-j+1} \dots \mathbf{L}_{k,k}$  sequentially from  $\tilde{\mathbf{R}}$  at epoch  $k$ , whose derivation can be found in Section 0.

Algorithm 6.2 Inverse of the Cholesky factor matrix  $\mathbf{C}$  at epoch  $k$

$$\begin{array}{l}
\mathbf{L}_{k,k} = \mathbf{C}_{k,k}^{-1} \\
\text{for } i = (k-j+1), \dots, (k-1) \{ \\
\quad \mathbf{L}_{k,i} = -\mathbf{L}_{k,k} \mathbf{C}_{k,k-1} \sum_{j=i}^{k-1} \mathbf{L}_{j,i} \\
\}
\end{array}$$

Besides, (6.12) can be written as

$$\begin{pmatrix} \tilde{\delta \mathbf{z}}_{k-j+1} \\ \vdots \\ \tilde{\delta \mathbf{z}}_{k-1} \\ \tilde{\delta \mathbf{z}}_k \end{pmatrix} = \mathbf{H} \begin{pmatrix} \delta \mathbf{x}_{k-j} \\ \delta \mathbf{x}_{k-j+1} \\ \vdots \\ \delta \mathbf{x}_{k-1} \\ \delta \mathbf{x}_k \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{v}}_{k-j+1} \\ \vdots \\ \tilde{\mathbf{v}}_{k-1} \\ \tilde{\mathbf{v}}_k \end{pmatrix} \quad (6.20)$$

$$\text{where } \mathbf{H} = \begin{pmatrix} \mathbf{H}_{k-j+1,k-j} & \mathbf{H}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{H}_{k-1,k-2} & \mathbf{H}_{k-1,k-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{H}_{k,k-1} & \mathbf{H}_{k,k} \end{pmatrix}.$$

The de-correlation of (6.20) by multiplying by  $\mathbf{L}$  goes as follows

$$\mathbf{L} \begin{pmatrix} \tilde{\delta \mathbf{z}}_{k-j+1} \\ \vdots \\ \tilde{\delta \mathbf{z}}_{k-1} \\ \tilde{\delta \mathbf{z}}_k \end{pmatrix} = \mathbf{LH} \begin{pmatrix} \delta \mathbf{x}_{k-j} \\ \delta \mathbf{x}_{k-j+1} \\ \vdots \\ \delta \mathbf{x}_{k-1} \\ \delta \mathbf{x}_k \end{pmatrix} + \mathbf{L} \begin{pmatrix} \tilde{\mathbf{v}}_{k-j+1} \\ \vdots \\ \tilde{\mathbf{v}}_{k-1} \\ \tilde{\mathbf{v}}_k \end{pmatrix} \quad (6.21)$$

$$\begin{pmatrix} \delta \mathbf{z}'_{k-j+1} \\ \vdots \\ \delta \mathbf{z}'_{k-1} \\ \delta \mathbf{z}'_k \end{pmatrix} = \mathbf{H}' \begin{pmatrix} \delta \mathbf{x}_{k-j} \\ \delta \mathbf{x}_{k-j+1} \\ \vdots \\ \delta \mathbf{x}_{k-1} \\ \delta \mathbf{x}_k \end{pmatrix} + \begin{pmatrix} \mathbf{v}'_{k-j+1} \\ \vdots \\ \mathbf{v}'_{k-1} \\ \mathbf{v}'_k \end{pmatrix}.$$

Thus, the de-correlated measurement vector  $\delta \mathbf{z}'$  is given as

$$\begin{pmatrix} \delta \mathbf{z}'_{k-j+1} \\ \vdots \\ \delta \mathbf{z}'_{k-1} \\ \delta \mathbf{z}'_k \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{L}_{k-1,k-j+1} & \cdots & \mathbf{L}_{k-1,k-1} & \mathbf{O} \\ \mathbf{L}_{k,k-j+1} & \cdots & \mathbf{L}_{k,k-1} & \mathbf{L}_{k,k} \end{pmatrix} \begin{pmatrix} \tilde{\delta \mathbf{z}}_{k-j+1} \\ \vdots \\ \tilde{\delta \mathbf{z}}_{k-1} \\ \tilde{\delta \mathbf{z}}_k \end{pmatrix}. \quad (6.22)$$

In short form,  $\delta \mathbf{z}'_k$  can be computed as

$$\delta \mathbf{z}'_k = \sum_{i=k-j+1}^k (\mathbf{L}_{k,i} \tilde{\delta \mathbf{z}}_i). \quad (6.23)$$

The de-correlated output matrix  $\mathbf{H}'$  is given by

$$\begin{aligned}
& \begin{pmatrix} \mathbf{H}'_{k-j+1,k-j} & \mathbf{H}'_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{H}'_{k-1,k-j} & \mathbf{H}'_{k-1,k-j+1} & \cdots & \mathbf{H}'_{k-1,k-2} & \mathbf{H}'_{k-1,k-1} & \mathbf{O} \\ \mathbf{H}'_{k,k-j} & \mathbf{H}'_{k,k-j+1} & \cdots & \mathbf{H}'_{k,k-2} & \mathbf{H}'_{k,k-1} & \mathbf{H}'_{k,k} \end{pmatrix} \\
& = \begin{pmatrix} \mathbf{L}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{L}_{k-1,k-j+1} & \cdots & \mathbf{L}_{k-1,k-1} & \mathbf{O} \\ \mathbf{L}_{k,k-j+1} & \cdots & \mathbf{L}_{k,k-1} & \mathbf{L}_{k,k} \end{pmatrix} \begin{pmatrix} \mathbf{H}_{k-j+1,k-j} & \mathbf{H}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{H}_{k-1,k-2} & \mathbf{H}_{k-1,k-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{H}_{k,k-1} & \mathbf{H}_{k,k} \end{pmatrix} \\
& = \begin{pmatrix} \mathbf{L}_{k-j+1,k-j+1} \mathbf{H}_{k-j+1,k-j} & \mathbf{L}_{k-j+1,k-j+1} \mathbf{H}_{k-j+1,k-j+1} & \cdots & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{L}_{k-1,k-j+1} \mathbf{H}_{k-j+1,k-j} & \mathbf{L}_{k-1,k-j+1} \mathbf{H}_{k-j+1,k-j+1} & \cdots & \mathbf{L}_{k-1,k-2} \mathbf{H}_{k-2,k-2} & \cdots & \mathbf{L}_{k-1,k-1} \mathbf{H}_{k-1,k-1} \\ \mathbf{L}_{k,k-j+1} \mathbf{H}_{k-j+1,k-j} & \mathbf{L}_{k,k-j+1} \mathbf{H}_{k-j+1,k-j+1} & \cdots & \mathbf{L}_{k,k-2} \mathbf{H}_{k-2,k-2} & \cdots & \mathbf{L}_{k,k-1} \mathbf{H}_{k-1,k-1} \\ \mathbf{L}_{k,k-j+1} \mathbf{H}_{k-j+1,k-j} & \mathbf{L}_{k,k-j+1} \mathbf{H}_{k-j+1,k-j+1} & \cdots & \mathbf{L}_{k,k-1} \mathbf{H}_{k-1,k-2} & \mathbf{L}_{k,k} \mathbf{H}_{k,k-1} & \mathbf{L}_{k,k} \mathbf{H}_{k,k} \end{pmatrix}
\end{aligned} \tag{6.24}$$

The block matrix  $\mathbf{H}'_{k,i}$  for  $i = (k-j) \leq i \leq k$  can be computed by

$$\mathbf{H}'_{k,i} = \begin{cases} \mathbf{L}_{k,i+1} \mathbf{H}_{i+1,i}, & \text{for } i = (k-j) \\ \mathbf{L}_{k,i} \mathbf{H}_{i,i} + \mathbf{L}_{k,i+1} \mathbf{H}_{i+1,i}, & \text{for } i = (k-j+1) \leq i \leq (k-1). \\ \mathbf{L}_{k,i} \mathbf{H}_{k,i}, & \text{for } i = k \end{cases} \tag{6.25}$$

At epoch  $k$ , the final measurement equation for feature point  $i$  becomes

$$\delta \mathbf{z}'_k = \mathbf{H}'_{k,k} \delta \mathbf{x}_k + \mathbf{H}'_{k,k-1} \delta \mathbf{x}_{k-1} + \cdots + \mathbf{H}'_{k,k-j} \delta \mathbf{x}_{k-j} + \mathbf{v}'_k, \mathbf{v}'_k \sim N(\mathbf{0}, \mathbf{I}). \tag{6.26}$$

The system model is given by

$$\begin{pmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{x}_{k-1} \\ \vdots \\ \delta \mathbf{x}_{k-j} \end{pmatrix} = \begin{pmatrix} \Phi_{k-1} \delta \mathbf{x}_{k-1} \\ \delta \mathbf{x}_{k-1} \\ \vdots \\ \delta \mathbf{x}_{k-j} \end{pmatrix} + \begin{pmatrix} \Psi_{k-1} \mathbf{w}_{k-1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}. \tag{6.27}$$

The state augmentation and time update algorithm has been described in Section 5.2.1 above.

### 6.2.1. Discussions

The system model in (6.27) and the measurement models in (6.26) are in the form of standard Kalman filter and therefore can be easily implemented in the current GNSS/IMU integration architecture and smoothers. For the Cholesky factors  $\mathbf{C}$  is unique, real and strictly positive with its diagonal block submatrices, the variance-covariance matrix  $\tilde{\mathbf{R}}$  must be positive definite.  $\tilde{\mathbf{R}}$  is positive definite if it can be written as  $\tilde{\mathbf{R}} = \mathbf{A}\mathbf{A}^T$ , wherein  $\mathbf{A}$  is possibly rectangular with the independent columns [Strang, 2009]. In theory, (6.9) and (6.11) satisfy this condition and therefore  $\tilde{\mathbf{R}}$  is always positive definite. However, in practice,  $\tilde{\mathbf{R}}$  can become semi-definite or indefinite if the feature matcher finds incorrect correspondences.

The computational complexity of the SLAM measurement updates is in the order of  $O((6+3N)\times(6+3N))$  where  $N$  is the number of landmarks in the map. On the other hand, the MFVO computational complexity is in the order of  $O((6+6j)\times(6+6j))$  where  $k-j$  is the epoch at which the feature in the current feature list at epoch  $k$  was first observed. Because matrix  $\tilde{\mathbf{R}}$  is sparse, the computation of  $\mathbf{L}$  can be simplified as described in Algorithm 6.1 and Algorithm 6.2, and does not consume significant computational resources. The computational complexity of computing  $\mathbf{L}_k$ ,  $\delta\mathbf{z}'_k$  and  $\mathbf{H}'_k$  is in the order of  $O((3N)\times(6+6j))$ . This shows that MFVO is more efficient than SLAM when  $N$  is large and  $j$  is small. The value of  $j$  depends on the overlap percentage of the image frames which, in turns, depends on the velocity of the camera system. Based on (4.18) and under the assumptions in the analysis presented in Section 4.3, the value  $j$  as a function of the overlap percentage  $p$  can be predicted as

$$j = \text{ceil} \left( \frac{1}{1-p} - 1 \right). \quad (6.28)$$

wherein  $\text{ceil}(x)$  is the operator that rounds  $x$  up and returns the smallest integral value, which is not less than  $x$ . For example, if  $p=30\%$ , a feature can only be observed at epochs  $k$  and  $k-1$ , but not again at  $k-2$ , i.e.,  $j=1$ . In case  $p=60\%$ , the overlap percentage between  $k$  and  $k-2$  will be 20%, and a feature can be observed at epochs  $k$ ,  $k-1$  and  $k-2$ , but not further at  $k-3$ , now  $j=2$ . As the overlap percentage increases, the value  $j$  increases as graphically presented in Figure 6.1.

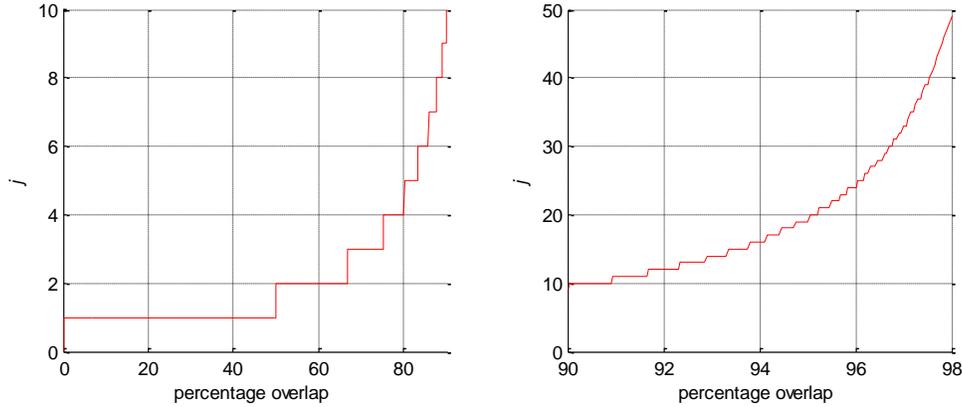


Figure 6.1  $j$  as a function of overlap percentage

When a vehicle is stationary, the image overlap percentage is 100% and  $j = \infty$ . This implies that the size of the state vector increases without bounds. For practical purposes,  $j$  can be limited to a predefined threshold  $j_{\max}$  and realized in three ways. Firstly, set an appropriate frame distance, i.e., the distance between two consecutive image frames. If the distance between the frames are too short, the overlap percentage and  $j$  may become too large. If the distance between the frames is too long, there could be too few correspondences to compute an odometry solution. Secondly, the extracted feature points

can be treated as new landmark when  $j$  exceeds  $j_{\max}$ . Thirdly, the measurement vector  $\tilde{\mathbf{z}}_{k-j+1}$  can be omitted in the derivation of  $\delta\mathbf{z}'_k$ , once  $j$  exceeds  $j_{\max}$ . However, this requires the re-computation of the corresponding matrices  $\mathbf{C}$  and  $\mathbf{L}$  from  $(k-j+2)$  to  $k$ .

### 6.3. Test results with simulated data

Monte Carlo (MC) simulations were conducted to illustrate and compare the performance of the SLAM and the proposed MFVO. The same 2D trajectory, measurements and simulation parameters were employed as in Section 5.3. Figure 6.2 shows the vehicle's trajectory with the visible landmarks, velocity and heading profiles.

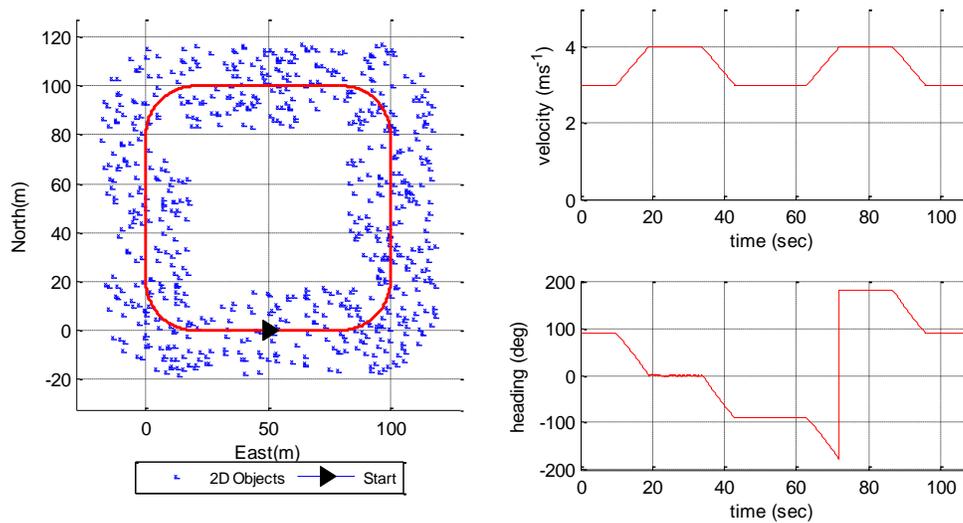


Figure 6.2 The top view of the vehicle trajectory with the visible landmarks (left) and the velocity and heading profiles (right)

The system model in the navigation frame is given by:

$$\begin{pmatrix} \delta\dot{X}^n \\ \delta\dot{Y}^n \\ \delta v_x^n \\ \delta v_y^n \\ \delta\dot{\psi}_z^n \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & a_y^n \\ 0 & 0 & 0 & 0 & -a_x^n \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta X^n \\ \delta Y^n \\ \delta v_x^n \\ \delta v_y^n \\ \delta\psi_z^n \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^n & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} w_{a_y^b} \\ w_{a_x^b} \\ w_{g_z^b} \end{pmatrix}. \quad (6.29)$$

wherein  $(\delta X^n, \delta Y^n)^T$  [m],  $(\delta v_x^n, \delta v_y^n)^T$  [m] and  $\delta\psi_z^n$  [rad] are the error states of the vehicle's position, velocity and heading, respectively,  $(a_x^n, a_y^n)^T$  [ms<sup>-2</sup>] are the accelerations;  $(w_{a_x^b}, w_{a_y^b})^T$  [ms<sup>-2</sup>] are the accelerometer process noises,  $w_{g_z^b}$  [rads<sup>-1</sup>] is the gyroscope process noise, and

$$\mathbf{C}_b^n = \begin{pmatrix} \cos(\psi_z^n) & -\sin(\psi_z^n) \\ \sin(\psi_z^n) & \cos(\psi_z^n) \end{pmatrix}. \quad (6.30)$$

The 2D SLAM measurement equations for landmark  $i$  are given as

$$\begin{pmatrix} l_{x,k,i}^b \\ l_{y,k,i}^b \end{pmatrix} = (\mathbf{C}_{b,k}^n)^T \begin{pmatrix} X_{k,i}^n - m_{x,i}^n \\ Y_{k,i}^n - m_{y,i}^n \end{pmatrix} + \begin{pmatrix} v_{x,k,i}^b \\ v_{y,k,i}^b \end{pmatrix} \quad (6.31)$$

where  $(l_{x,k,i}^b, l_{y,k,i}^b)^T$  [m] is the measurement vector,  $(m_{x,i}^n, m_{y,i}^n)^T$  [m] is the landmark position vector, and  $(v_{x,k,i}^b, v_{y,k,i}^b)^T$  is the measurement noise vector. Using the 2D SLAM (6.31) at epochs  $k$  and  $k-m$  and the landmark position vector  $(m_{x,i}^n, m_{y,i}^n)^T$  are algebraically eliminated in order to derive the following 2D MFVO measurements

$$(\mathbf{C}_{b,k}^n)^T \begin{pmatrix} l_{x,k,i}^b - v_{x,k,i}^b \\ l_{y,k,i}^b - v_{y,k,i}^b \end{pmatrix} - (\mathbf{C}_b^n)^T \begin{pmatrix} l_{x,k-m,i}^b - v_{x,k-m,i}^b \\ l_{y,k-m,i}^b - v_{y,k-m,i}^b \end{pmatrix} + \begin{pmatrix} X_{k,i}^n \\ Y_{k,i}^n \end{pmatrix} - \begin{pmatrix} X_{k-m,i}^n \\ Y_{k-m,i}^n \end{pmatrix} = 0. \quad (6.32)$$

The SLAM and MFVO were run for 100 times, respectively. The true position and heading errors were computed for each run. Then the root-mean-square errors (RMSE) across the 100 runs were computed every epoch. The resulting error bounds were then

compared with the estimated standard deviations. The true pose RMS errors from SLAM and MFVO are plotted in Figure 6.3 (left) while the corresponding estimated standard deviations are plotted in Figure 6.3 (right). Clearly, the position and heading solutions are identical.

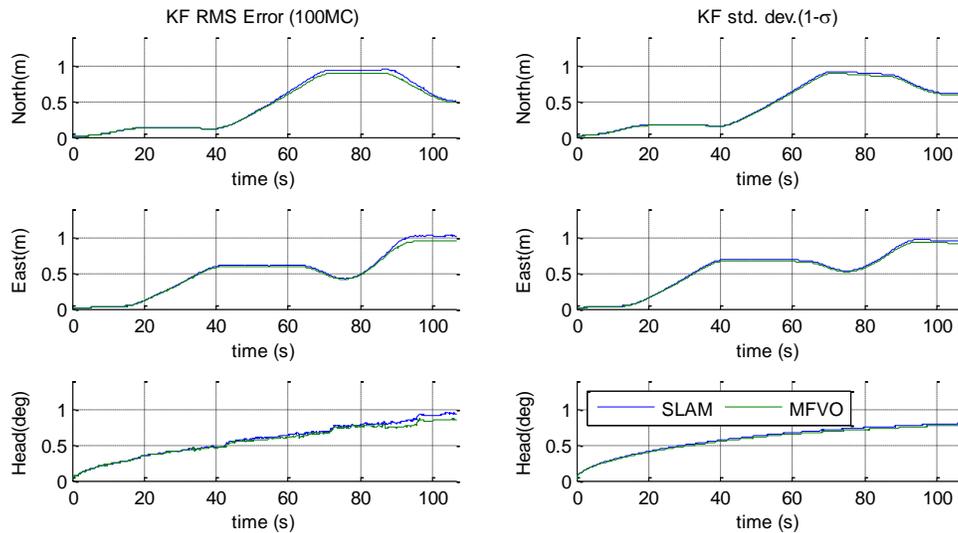


Figure 6.3 The true position and heading RMS errors (left) and the estimated position and heading standard deviations (right)

Figure 6.4 shows the dimensions of the state vectors with SLAM and MFVO, the number of features in view and the value  $j$ . With the same number of features, the number of the states in MFVO is generally smaller than the one in SLAM, except around the 60th second as there were less features in view around that instant.

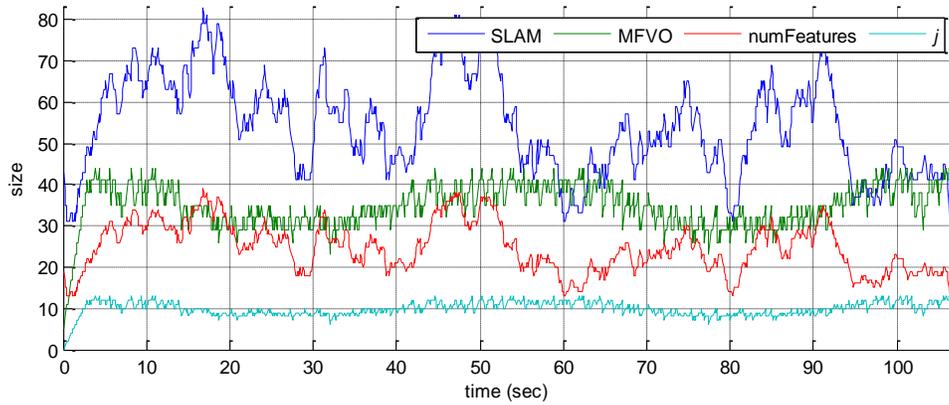


Figure 6.4 The dimensions of the state vectors with SLAM and MFVO, the number of features and  $j$

Notice that the dimension of the MFVO state vector and the number  $j$  vary inversely with the velocity profile (in Figure 6.2), because the slower a vehicle travels, the more time a camera spends to view the same set of landmarks and therefore the value of  $j$  increases.

### 6.3.1. VO KF-PTC versus MFVO

The performance comparison between the VO with KF-PTC presented in Section 5.3 and the MFVO is shown in Figure 6.5, which shows that the MFVO approach performed better than the VO/KF-PTC with the same set of measurements. Unsurprisingly, the MFVO approach works with more states and requires more computational resources.

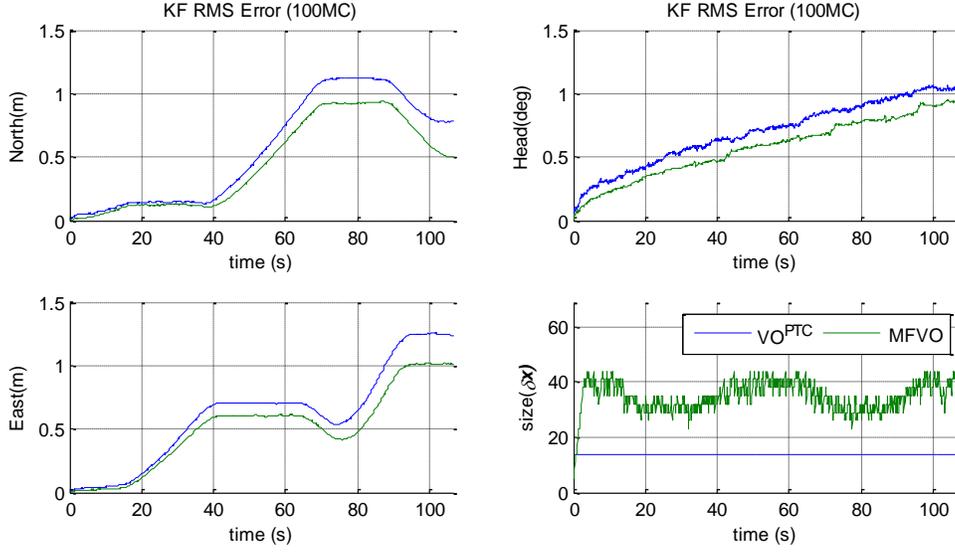


Figure 6.5 The comparison between the VO/KF-PTC and MFVO approaches

#### 6.4. Tightly-coupled stereo MFVO aided inertial navigation

This section derives the measurement model to develop the tightly-coupled MFVO-aided inertial navigation. The derivation starts with the visual SLAM measurement model for feature  $i$  at epoch  $k$

$$\mathbf{I}_{k,i}^b = (\mathbf{C}_{b(k),INS}^n)^T (\mathbf{m}_i^n - \mathbf{X}_{INS,k}^n) + \mathbf{v}_{k,i}^b, \mathbf{v}_{k,i}^b \sim N(\mathbf{0}, \mathbf{R}_{\mathbf{v}_{k,i}^b \mathbf{v}_{k,i}^b}^b) \quad (6.33)$$

where  $\mathbf{I}_{k,i}^b$  is the 3D image measurement vector. Using the visual SLAM as in (6.33) at epochs  $k$  and  $k-1$ ,  $\mathbf{m}_i^n$  can be algebraically eliminated to yield the MFVO model:

$$\mathbf{C}_{b(k),INS}^n (\mathbf{l}_{k,i}^b - \mathbf{v}_{k,i}^b) - \mathbf{C}_{b(k-1),INS}^n (\mathbf{l}_{k-1,i}^b - \mathbf{v}_{k-1,i}^b) + \mathbf{X}_{INS,k}^n - \mathbf{X}_{INS,k-1}^n = \mathbf{0}. \quad (6.34)$$

in which  $\mathbf{X}_{INS,k}^n - \mathbf{X}_{INS,k-1}^n$  can be expressed as  $\mathbf{D}_r^{-1}(\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-1})$  since it is expected to be small i.e. less than 3.0 m. Thus, (6.34) becomes

$$\mathbf{C}_{b(k),INS}^n (\mathbf{l}_{k,i}^b - \mathbf{v}_{k,i}^b) - \mathbf{C}_{b(k-1),INS}^n (\mathbf{l}_{k-1,i}^b - \mathbf{v}_{k-1,i}^b) + \mathbf{D}_r^{-1}(\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-1}) = \mathbf{0}. \quad (6.35)$$

The linearization of (6.35) gives

$$\begin{aligned}
\tilde{\delta \mathbf{z}}_{k,i} = & -\mathbf{C}_{b(k),INS}^n \mathbf{v}_{k,i}^b + (\mathbf{C}_{b(k),INS}^n \mathbf{l}_{k,i}^b) \times \boldsymbol{\Psi}_{INS,k} + \mathbf{C}_{b(k-1),INS}^n \mathbf{v}_{k-1,i}^b \cdots \\
& - (\mathbf{C}_{b(k-1),INS}^n \mathbf{l}_{k-1,i}^b) \times \boldsymbol{\Psi}_{INS,k-1} + \delta \mathbf{D}_r^{-1} (\mathbf{r}_{INS,k}^n - \mathbf{r}_{INS,k-1}^n) \cdots \\
& + \mathbf{D}_r^{-1} (\delta \mathbf{r}_{INS,k} - \delta \mathbf{r}_{INS,k-1})
\end{aligned} \tag{6.36}$$

wherein  $\tilde{\delta \mathbf{z}}_{k,i} = -(\mathbf{C}_{b(k),INS}^n \mathbf{l}_{k,i}^b - \mathbf{C}_{b(k-1),INS}^n \mathbf{l}_{k-1,i}^b + \mathbf{D}_r^{-1} (\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-1}))$ . Since the position change between two consecutive frames is expected to be relatively small, i.e. less than 3.0 m, the term  $\delta \mathbf{D}_r^{-1} (\mathbf{r}_{INS,k} - \mathbf{r}_{INS,k-1})$  will be significantly small and can be omitted in (6.36) (see Section 5.4 for more details).  $\mathbf{D}_r^{-1} \delta \mathbf{r}_{INS,k}$  can be expressed as  $\delta \mathbf{X}_{k,INS}^n$  and (6.36) can further be simplified to

$$\begin{aligned}
\tilde{\delta \mathbf{z}}_{k,i} = & \delta \mathbf{X}_{INS,k}^n - \delta \mathbf{X}_{INS,k-1}^n + (\mathbf{C}_{b(k),INS}^n \mathbf{l}_{k,i}^b) \times \boldsymbol{\Psi}_{INS,k} \cdots \\
& - (\mathbf{C}_{b(k-1),INS}^n \mathbf{l}_{k-1,i}^b) \times \boldsymbol{\Psi}_{INS,k-1} - \mathbf{C}_{b(k),INS}^n \mathbf{v}_{k,i}^b + \mathbf{C}_{b(k-1),INS}^n \mathbf{v}_{k-1,i}^b .
\end{aligned} \tag{6.37}$$

The measurement noise vector  $\tilde{\mathbf{v}}_{k,i}$  is expressed as follows

$$\tilde{\mathbf{v}}_{k,i} = -\mathbf{C}_{b(k),INS}^n \mathbf{v}_{k,i}^b + \mathbf{C}_{b(k-1),INS}^n \mathbf{v}_{k-1,i}^b . \tag{6.38}$$

With the corresponding variance-covariance matrix  $\tilde{\mathbf{R}}_{k,k,i}$  and the cross-covariance matrix  $\tilde{\mathbf{R}}_{k,k-1,i}$

$$\begin{aligned}
\tilde{\mathbf{R}}_{k,k,i} = & (\mathbf{C}_{b(k),INS}^n) \mathbf{R}_{\mathbf{v}_{k,i}^b \mathbf{v}_{k,i}^b} (\mathbf{C}_{b(k),INS}^n)^T \cdots \\
& + (\mathbf{C}_{b(k-1),INS}^n) \mathbf{R}_{\mathbf{v}_{k-1,i}^b \mathbf{v}_{k-1,i}^b} (\mathbf{C}_{b(k-1),INS}^n)^T . \\
\tilde{\mathbf{R}}_{k,k-1,i} = & -(\mathbf{C}_{b(k-1),INS}^n) \mathbf{R}_{\mathbf{v}_{k-1,i}^b \mathbf{v}_{k-1,i}^b} (\mathbf{C}_{b(k-1),INS}^n)^T
\end{aligned} \tag{6.39}$$

The coefficient matrices  $\mathbf{H}_{k,k,i}$  and  $\mathbf{H}_{k,k-1,i}$  of the measurement model are

$$\begin{aligned}
\mathbf{H}_{k,k,i} = & \begin{pmatrix} \mathbf{I}_{3 \times 3} & (\mathbf{C}_{b(k),INS}^n \mathbf{l}_{k,i}^b) \times \end{pmatrix} \\
\mathbf{H}_{k,k-1,i} = & \begin{pmatrix} -\mathbf{I}_{3 \times 3} & -(\mathbf{C}_{b(k-1),INS}^n \mathbf{l}_{k-1,i}^b) \times \end{pmatrix} .
\end{aligned} \tag{6.40}$$

Hence, the discrete system model can be given as follows

$$\begin{pmatrix} \delta \mathbf{x}_{k,INS} \\ \delta \mathbf{x}'_{k-1,INS} \\ \vdots \\ \delta \mathbf{x}'_{k-j,INS} \end{pmatrix} = \begin{pmatrix} \Phi_{k-1} \delta \mathbf{x}_{k-1,INS} \\ \delta \mathbf{x}'_{k-1} \\ \vdots \\ \delta \mathbf{x}'_{k-j} \end{pmatrix} + \begin{pmatrix} \Psi_{k-1} \mathbf{w}_{k-1} \\ \mathbf{0}_{6 \times 1} \\ \vdots \\ \mathbf{0}_{6 \times 1} \end{pmatrix} \quad (6.41)$$

wherein  $\mathbf{x}_{k,INS} = (\delta \mathbf{X}_{k,INS}^n, \delta \mathbf{v}_{k,INS}^n, \boldsymbol{\psi}_{k,INS}^n, \mathbf{b}_a^b, \mathbf{b}_g^b)^T$  and  $\delta \mathbf{x}'_{k'} = (\delta \mathbf{X}_{k',INS}^n, \boldsymbol{\psi}_{k',INS}^n)^T$  for  $k' = (k-j) \cdots (k-1)$ . For each feature in view, one can compute the corresponding  $\mathbf{C}_{k,k,i}$ ,  $\mathbf{C}_{k,k-1,i}$ ,  $\mathbf{L}_{k,k-j+1,i}, \dots, \mathbf{L}_{k,k,i}$  and  $\mathbf{H}'_{k,k-j,i}, \dots, \mathbf{H}'_{k,k,i}$  and  $\delta \mathbf{z}'_{k,i}$ . The measurement model can also be given as

$$\mathbf{z}'_{k,i} = \mathbf{H}'_{k,k,i} \delta \mathbf{x}_k + \mathbf{H}'_{k,k-1,i} \delta \mathbf{x}_{k-1} + \dots + \mathbf{H}'_{k,k-j,i} \delta \mathbf{x}_{k-j} + \mathbf{v}'_{k,i}, \mathbf{v}'_{k,i} \sim N(\mathbf{0}, \mathbf{I}). \quad (6.42)$$

With the formulation of the system and measurement models as in (6.41) and (6.42), respectively, the MFVO measurements can be easily integrated into the current GNSS/IMU Kalman filters and smoothers without any further modification to the architecture (see Section 5.4 for more details). The tightly coupled MFVO aided-inertial navigation algorithm is outlined in Figure 6.6.

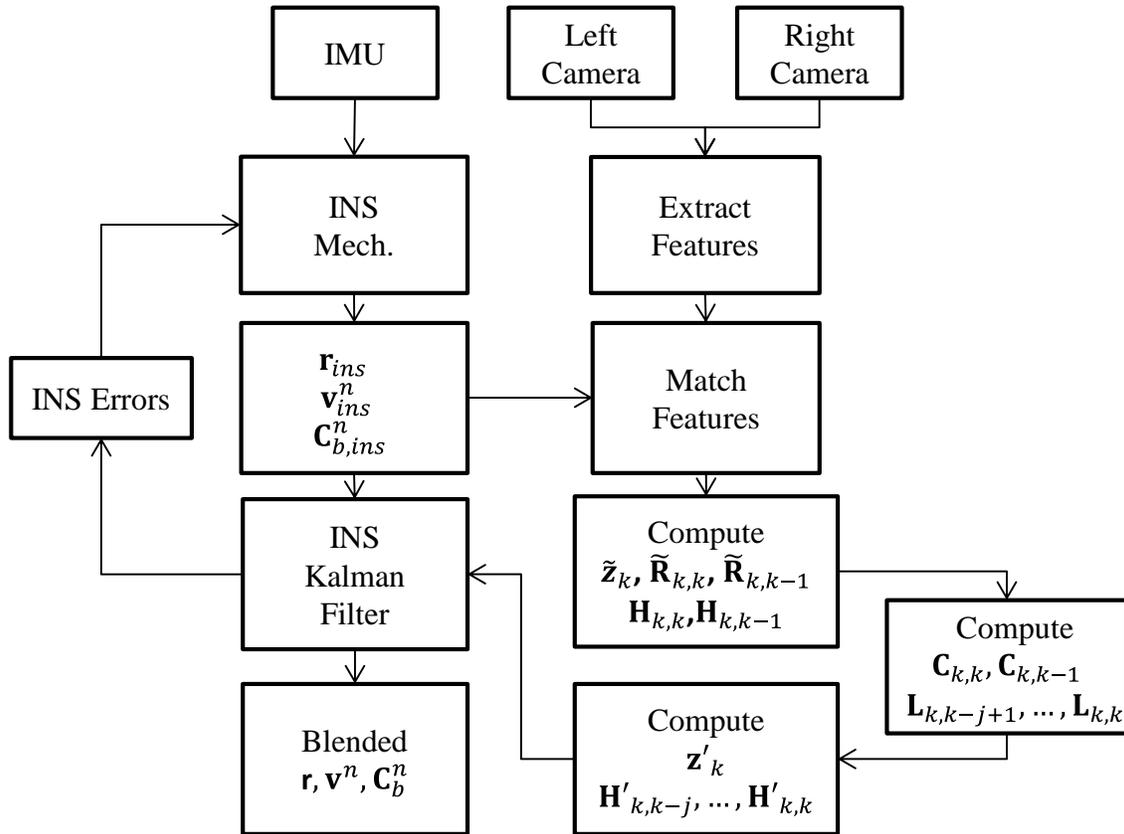


Figure 6.6 Tightly coupled MFVO aided-inertial navigation

The field test results from the TC stereo MFVO-aided inertial navigation and their analysis are presented in Section 7.5 .

## 6.5. Summary

This chapter presented a novel stereo based MFVO aided inertial navigation which integrates features tracked from multiple overlapping image frames sequentially and optimally. The existing algorithms such as [Fraundorfer et al, 2010; Clement et al, 2015; Wen et al, 2016] employed batch processing estimators and jointly estimated the vehicle's pose and feature positions at the local level and have two major drawbacks: (a) the number of parameters in the system quickly increases as more features per frame are

being observed and (b) it is impractical to integrate them into the an existing GNSS/INS integration architecture since they do not employ a Kalman filter.

The proposed MFVO method can process features from multiple overlapping frames without having to estimate the landmark positions. This is achieved by algebraically eliminating the time-invariant landmark position parameters in the SLAM measurement equation system at two consecutive epochs. Consequently, the derived measurements is timely-correlated. Through a sequential de-correlation algorithm, the measurement updates in Kalman filter can strictly be performed sequentially without loss of the solution's optimality. Monte Carlo simulations showed that the solution of the MFVO positions and orientations is identical to the one directly solved using the SLAM algorithm.

Although, this method is here specifically developed for the tightly coupled multi-frame aided-inertial integrated navigation, it can certainly be developed for the loosely-coupled multi-frame aided-inertial integrated navigation. However, its solution may be sub-optimal since the measurement time correlation may not be easily modeled appropriately.

Loop closure techniques can improve the navigation accuracy in image-based navigation system. However, these techniques were not applied in the research. Loop closure detection algorithms and the measurement update for SLAM application are well established. The challenge in applying loop closures in MFVO is how to perform the measurement update after a scene has been revisited given that the landmark position has been cancelled and the map is non-existent. This requires further research and

implementation in order to compare both SLAM and MFVO with loop closures. MFVO with loop closures is suggested for the future work.

The advantages of the proposed algorithm can be summarized as follows.

- The MFVO measurement updates in Kalman filter can be performed sequentially without loss of the optimality under the appropriate consideration of the measurement time correlation from epoch to epoch.
- The Kalman filter is in the standard form and therefore can easily be adapted in an existing GNSS-aided inertial navigation integration architecture.
- The MFVO and SLAM pose accuracy are the similar. MFVO allocates less memory and computation resources than SLAM when the number of features in view becomes large.
- MFVO can perform better than VO especially when the number of frames per landmark is high i.e. 3 or higher.
- The algorithm assumes that the landmark position vector can be algebraically eliminated. This can be accomplished with other SLAM applications and measurement models. For instance, in 2D Lidar SLAM [Wang et al, 2013], 3D Lidar SLAM [Hewitt and Marshall, 2016], radar SLAM [Callmer et al, 2011] and Sonar SLAM [Siantidis, 2016].

The disadvantages of MFVO are:

- Matrix  $\tilde{\mathbf{R}}$  can become semi-definite or indefinite if an incorrect match occurs and the Cholesky factorization can fail. In this case the measurements should be rejected.

- Conventional loop closure techniques cannot be applied in the current design since the landmark position vector is cancelled.

The selected results from visual SLAM aided-inertial integrated navigation and MFVO aided-inertial integrated navigation with real datasets and their comparison are presented in Section 7.5.

## **7. Experiments: road tests and results**

### **7.1. Introduction**

This chapter overviews the test results and analysis of the proposed algorithms developed in chapters 4, 5 and 6 using the data collected from the York University Multisensor Integrated System (YUMIS) [Qian et al, 2012; Qian, 2017]. The algorithms were implemented as software utilities in the MATLAB script language and their performances were evaluated in post-mission.

This chapter is organized as follows: Section 7.2 provides the YUMIS hardware specification and dataset information. Section 7.3 presents the results and analysis for the structureless stereo camera auto-calibration algorithm and system calibration algorithms. Then, the loosely-coupled visual odometry aided-inertial navigation analysis and results are shown in Section 7.4 followed by the one for the tightly-coupled multi-frame visual odometry aided-inertial navigation in Section 7.5.

### **7.2. YUMIS system and dataset information**

The data were collected by the YUMIS navigation system developed at the Earth Observation Laboratory of York University on a land vehicle [Qian et al, 2012; Qian, 2017]. This system provides a low-cost alternative to the expensive commercial navigation systems such as the Applanix POS system []. The YUMIS navigation system consists of two NovAtel OEM5 GNSS receivers, one Crossbow IMU440CA and two PointGrey Flea3 cameras. They are all connected to a central 1.6 GHz Intel Atom N270 CPU on a Jetway motherboard (NF94-270-LF). The GNSS receivers and the IMU are connected to the system via the RS232 serial ports, while cameras use the IEEE firewire

1394 interface. The system was built on the Ubuntu Linux (10.04 LTS) with Real Time Application Interface (Linux/RTAI) operating system (OS). The OS environment was set to hard real-time so that it can handle multiple tasks and to avoid time latency. The software system consists of four major components, namely data collectors, time-tagging module using GPS time, the data buffers, and data processors. The data collector grabs the raw data from the various sensors and loads them into a data buffer that the data processors can access. The time-tagging module catches the PPS (pulse per second) pulse train from the GNSS receivers to time tag and synchronize the various data. Figure 7.1 shows the hardware configuration of YUMIS system during a van test. For more details on the design and development of the YUMIS system, refer to [Qian et al, 2012; Qian, 2017].



Figure 7.1 GPS and IMU, and Controller in YUMIS system [Qian, 2017]

Multiple datasets collected using the YUMIS system have been processed and analysed. The results from one of them were selected to demonstrate the relevant research for the following reasons: (a) it contains relatively *ideal* GPS observables, (i.e. open sky conditions, no datagaps, minimal cycle slips and low multipath) so that a reliable cm-

level RTK position solution could be obtained and (b) the images were highly textured and therefore many point features could be extracted.

Two GPS receivers, separated by 1.8m, on the vehicle’s roof were used for the GNSS compass to estimate the absolute heading measurements and a third GPS receiver was set up as the base station for providing RTK level of positioning solution. The stereo baseline was 65cm and the resolution of the images was set to 640x480 pixels with the field of view of 50 degrees. The size of each pixel is 4.65  $\mu\text{m}$  [Point Grey Research Inc, 2011]. The lever arms of the GNSS receivers and the cameras with respect to the IMU were measured using a measuring tape at the accuracy of about 0.5cm. The individual sensor data rates were set to 5.0Hz, 100Hz and 7.5Hz for GPS receivers, IMU and cameras, respectively. The technical specification of Crossbow IMU440CA is summarized in Table 7.1 [Crossbow Technology Inc, 2010].

Table 7.1 Crossbow IMU440CA technical specification (partial)

Acceleration	Bias Stability[mg]	<1.0
	Velocity Random Walk[m/s/ $\sqrt{\text{hr}}$ ]	<1.0
Angular Rate	Bias Stability [deg/hour]	<10.0
	Angle Random Walk [deg/ $\sqrt{\text{hr}}$ ]	< 4.5

The test ran in City of Vaughan, Ontario, on 2 November 2014. The top view, the velocity profile and the attitude profile of the trajectory with the dataset can be overviewed in Figure 7.2. As the test was conducted in a residential area, the speed limit was 40km/h (11.1m/s).

Table 7.2 lists the length of the dataset, the traverse length, the maximum velocity and the maximum rover-base baseline length.

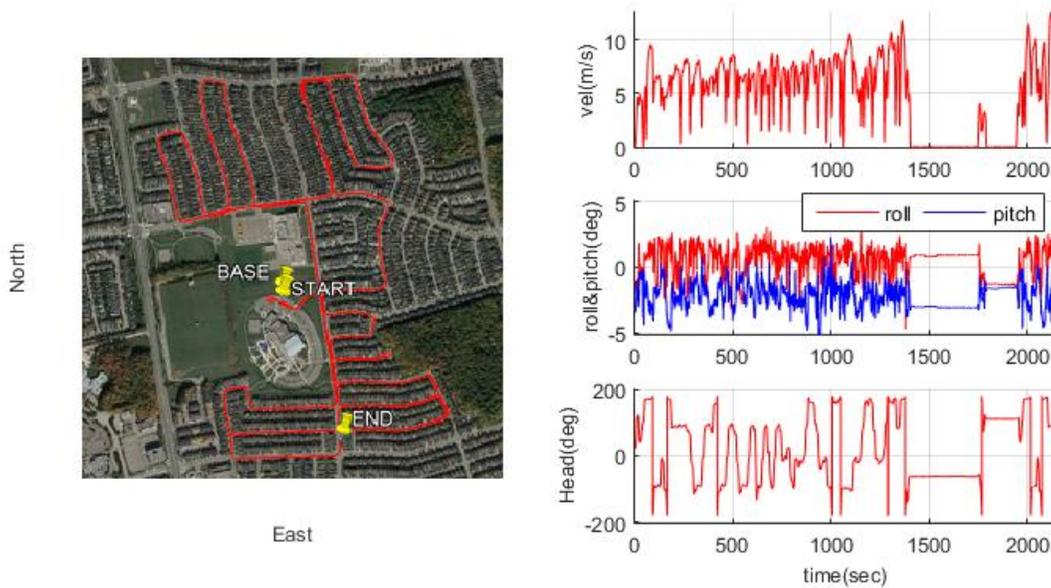


Figure 7.2 The 2D overview of the trajectory with starting point, end point and base position (left) and the velocity and attitude profiles (right)

Table 7.2 Dataset properties

Dataset Property	Value
Dataset length (sec)	2147
Traverse length (m)	10436
Maximum velocity (m/s)	12.6
Maximum baseline length (km)	0.93

### 7.2.1. Loosely-coupled GNSS aided inertial navigation

The loosely-coupled GNSS aided inertial navigation solution was generated and used as the reference trajectory for the subsequent tests. This sub-section summarizes the results.

The rover position was obtained using the third party POSGNSS (Grafnav) software. A MATLAB based GPS compass module was developed to obtain the GPS heading measurement. The GPS compass module consists of a sequential least squares estimator that processes C1 code and L1 phase measurements, *integer ambiguity resolution* using the LAMBDA method [Teunissen, 1995], a carrier phase cycle-slip detector [Bisnath,

2000] and a measurement outlier detector. Figure 7.3 shows the estimated position accuracy ( $1-\sigma$ ), estimated heading accuracy ( $1-\sigma$ ), the rover-base baseline length and the number of satellite used for computing the GPS position and heading.

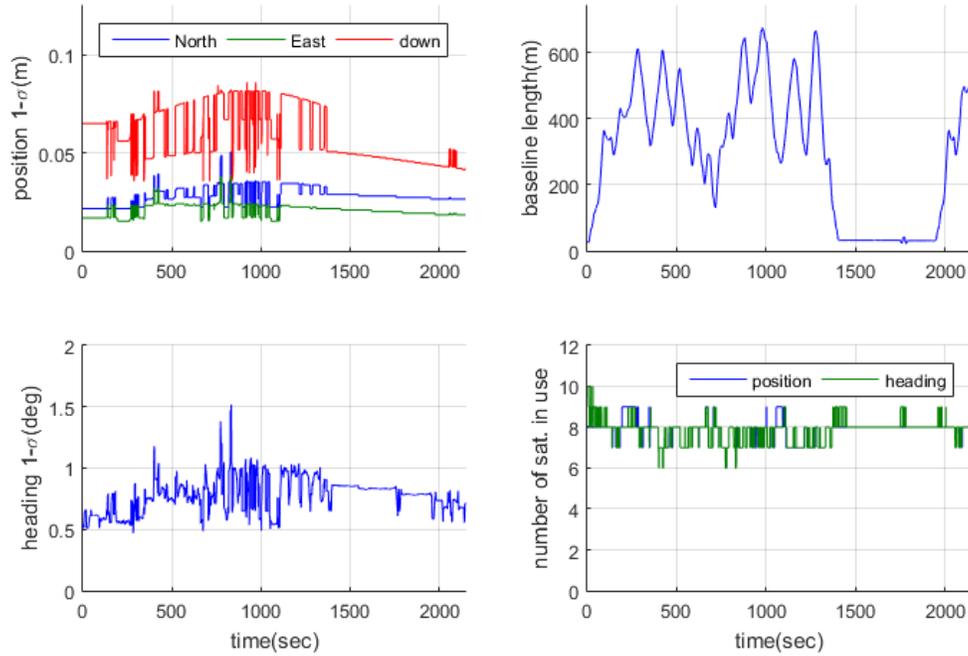


Figure 7.3 The GPS position standard deviation ( $1-\sigma$ ) (top-left). The GPS heading accuracy (bottom-left). The baseline length (top-right). The number of satellite used for computing the GPS position and heading (bottom-right)

Table 7.3 lists the IMU sensor error parameters. The parameters were obtained using the Allan variance technique [Allan, 1966] with 18 hours of static data (the detailed Allan variance results are omitted here).

The loosely-coupled integration was performed next. The psi-error model described in Section 3.5.2 was employed and the navigation state vector is

$$\mathbf{x}_k = (\delta\mathbf{X}^{n^c} \quad \delta\mathbf{v}^{n^c} \quad \boldsymbol{\psi} \quad \mathbf{b}_a^b \quad \mathbf{b}_g^b)^T \quad (7.1)$$

wherein  $\delta\mathbf{X}^{n^c}$ ,  $\delta\mathbf{v}^{n^c}$ ,  $\boldsymbol{\psi}$ ,  $\mathbf{b}_a^b$  and  $\mathbf{b}_g^b$  are the 3x1 position error vector, 3x1 velocity error vector, 3x1 attitude misalignment vector, 3x1 accelerometer bias vector and gyroscope bias vector, respectively.

Table 7.3 IMU440CA sensor error model parameters

Parameter	Component	Power spectral density
Velocity random walk [m/s/ $\sqrt{\text{hr}}$ ]	x	0.62
	y	0.46
	z	0.43
Angle random walk [deg/ $\sqrt{\text{hr}}$ ]	x	1.73
	y	2.26
	z	1.94
Accelerometer bias stability [mg]	x	0.87
	y	0.73
	z	0.74
Gyroscope bias stability [deg/hour]	x	8.40
	y	8.70
	z	8.39

Firstly, the horizontal static alignment was performed using (3.58). Then, the GPS position and heading measurements were used to initialize the state vector together with its covariance matrix. Finally, the Kalman filter was employed in the loosely coupled integration architecture. Figure 7.4 shows the overall 1- $\sigma$  position and orientation accuracies. The North, East and down accuracies were about  $\pm 0.03\text{m}$ ,  $\pm 0.02\text{m}$  and  $\pm 0.05\text{m}$ , respectively. The roll, pitch and heading accuracies were approximately  $\pm 0.15\text{deg}$ ,  $\pm 0.15\text{deg}$  and  $\pm 0.20\text{deg}$ , respectively. The position and attitude results are accurate enough as the reference solution for the subsequent tests.

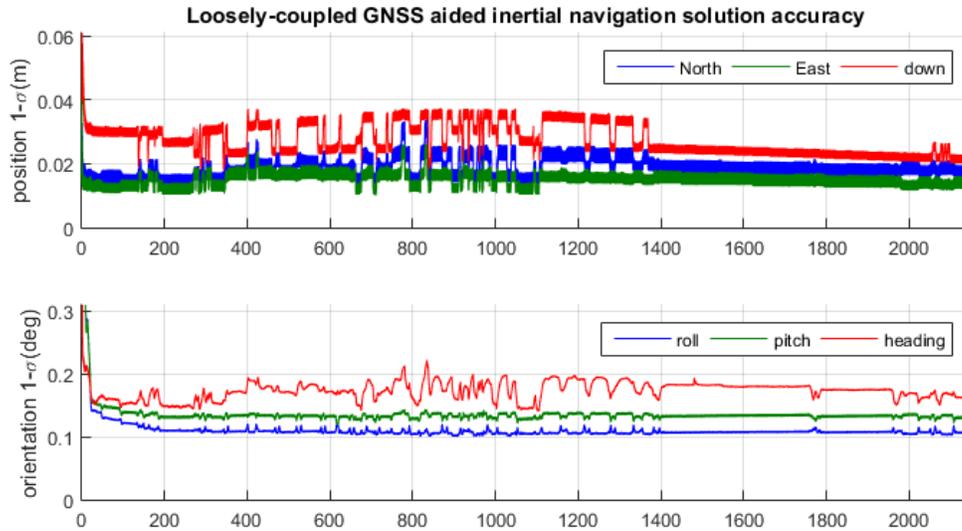


Figure 7.4 The estimated position accuracy (1- $\sigma$ ) and orientation accuracy (1- $\sigma$ ) of the loosely-coupled GNSS aided inertial navigation solution

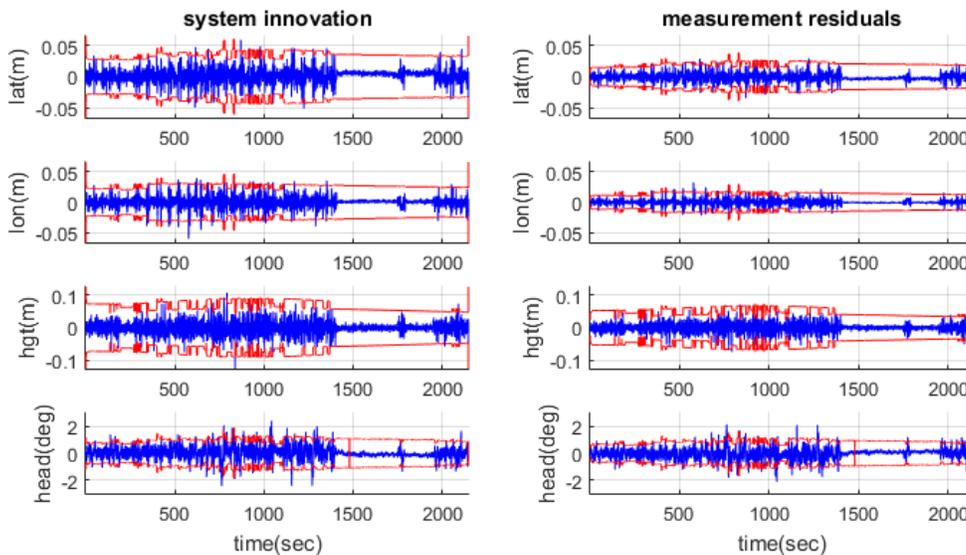


Figure 7.5 The GPS system innovations (left) and measurement residuals (right) with the corresponding 1- $\sigma$  envelope

Figure 7.5 shows the GNSS system innovations and measurement residuals with the corresponding 1- $\sigma$  envelope. The magnitude of the system innovations and measurement residuals for the GNSS position (<5cm) and GNSS heading (<0.5 deg) measurements

verify the correctness of the computed position and heading accuracy. Figure 7.6 shows the estimated accelerometer and gyroscope biases together with 1- $\sigma$  envelope.

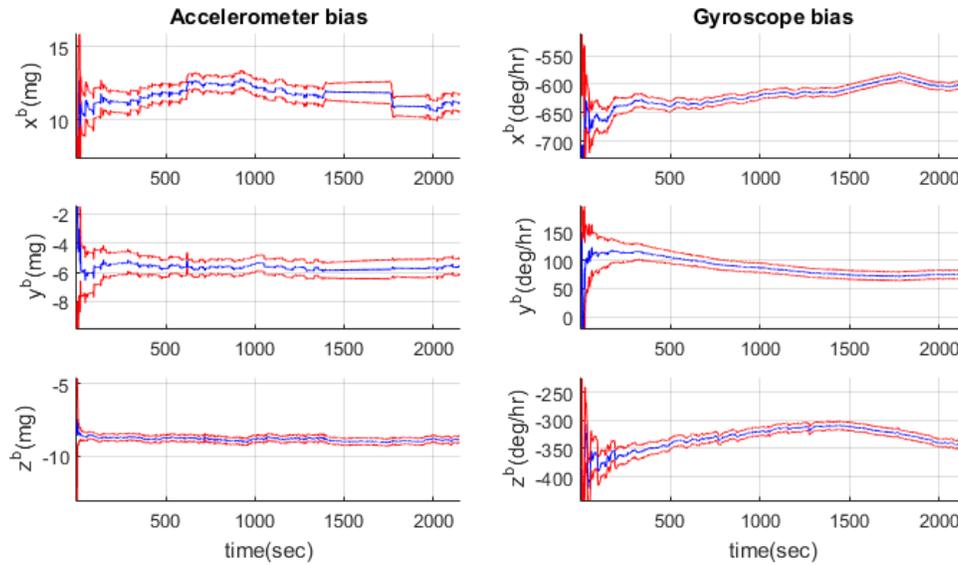


Figure 7.6 The estimated accelerometer and gyroscope biases with the corresponding 1- $\sigma$  envelope.

In summary, the loosely-coupled GNSS aided inertial navigation solution at the centimeter level position accuracy can provide an accurate reference trajectory.

### 7.3. Structureless stereo camera calibration

This section presents the results from the structureless stereo camera auto-calibration and system calibration using the algorithms developed in Chapter 4 and also compares them with the ones using the algorithm based on the collinearity equations. The analysis was conducted as follows:

- An interval of the trajectory was chosen for the calibration. Furthermore feature points within this interval were extracted and matched.
- The calibration parameters were computed by bundle adjustment using (a) the collinearity equations and (b) the scale-restraint equation. Both calibrations

were photo block-invariant i.e. the calibration parameters were assumed to be constant.

- The estimated calibration parameters were evaluated and compared.

This section compares the number of parameters and the number of floating point operations (flops) required in a least squares bundle adjustment between the stereo auto-calibration algorithms based on collinearity equations (COL) (Equation (4.1)) and the scale restraint equations (SRE) (Equation (4.5)).

### 7.3.1. Calibration interval and measurement information

One hundred and forty (140) stereo images within the interval shown in red (Figure 7.7 (left)) were used to test the camera calibration algorithms. This section of the trajectory contained turns and this favored the estimation of the lever-arm and bore-sight components. Furthermore, the images were highly textured and should be in favor of detecting point features.

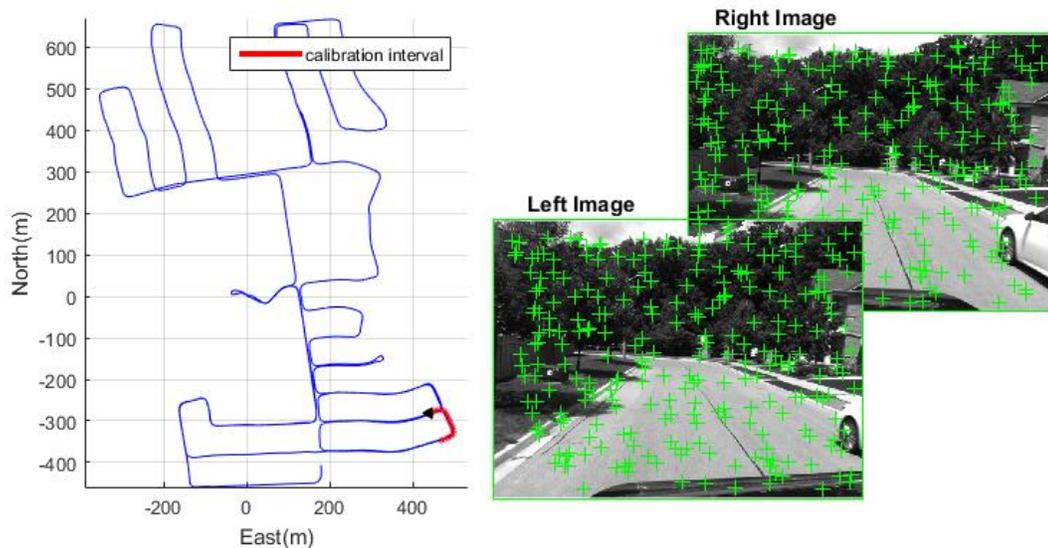


Figure 7.7 The calibration interval (left) and a stereo pair with the matched points (right)

The point features were extracted and matched using the LIBVISO2 library [Geiger et al, 2011], which consists of a corner and a blob detector. Its matcher employs the Sum of Absolute Differences (SAD) method on an 11x11 window. To improve the matching accuracy between stereo pairs, each search was constrained using the coplanarity equation. Between the consecutive frames, the feature locations on the following frame were predicted by using the GPS-aided inertial navigation solution. The standard deviation of the measurement for the extracted point features was assumed to be  $\pm 0.3$ px. Figure 7.7 (right) shows the 75<sup>th</sup> stereo image with the matched feature points.

Similar to the tests performed in Sections 4.3 and 4.4, the calibration results from one COL and two SREs are presented (i.e. still refer to as SRE1 and SRE2). The purpose of COL and SRE1 is to compare the performance of the collinearity equations and the scale-restraint equation when the same measurements are used. Simulation results in Section 4.4 suggest that the scale-restraint equation requires approximately 4 times more measurements to produce comparable calibration results to the collinearity equations. The number of extractable features in the real world depends on the texture of the images. With this set of images, SRE2 was able to obtain 3.6 more features than COL and SRE1. Figure 7.8 (top) shows the number of the used features and Figure 7.8 (bottom) shows the minimum, mean and maximum ranges of the 3D features in the test. The average overlapping percentage for all three cases was approximately 72%.

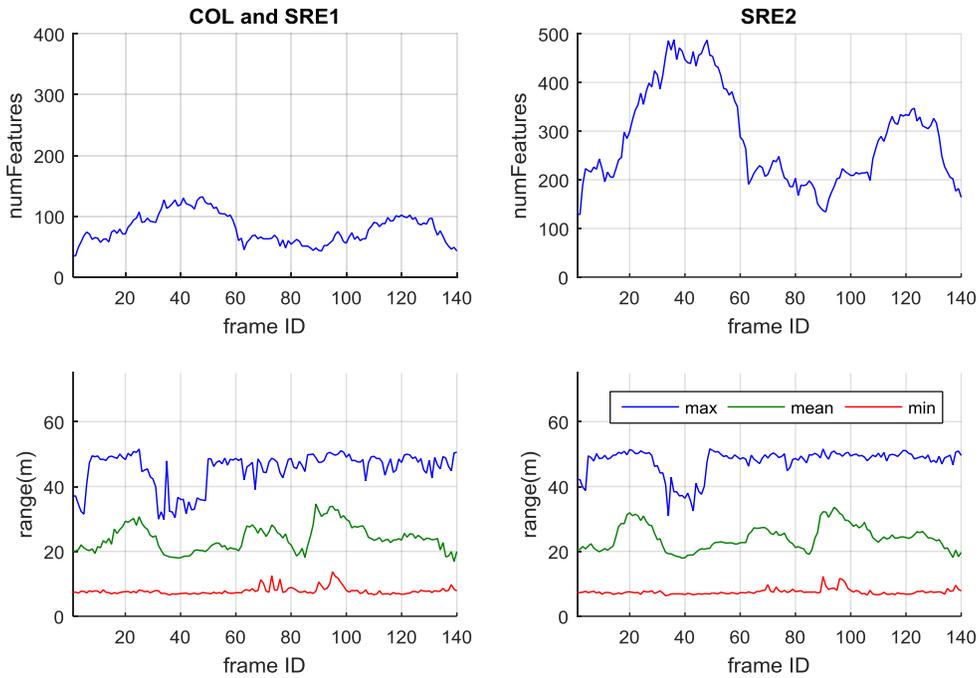


Figure 7.8 The number of the features (top), the minimum, mean and maximum ranges of the 3D features (bottom) for COL, SRE1 and SRE2.

The EO parameters for each image frame were obtained from the GNSS aided inertial navigation solution in Section 7.2.1. Their standard deviations are shown in Figure 7.9.

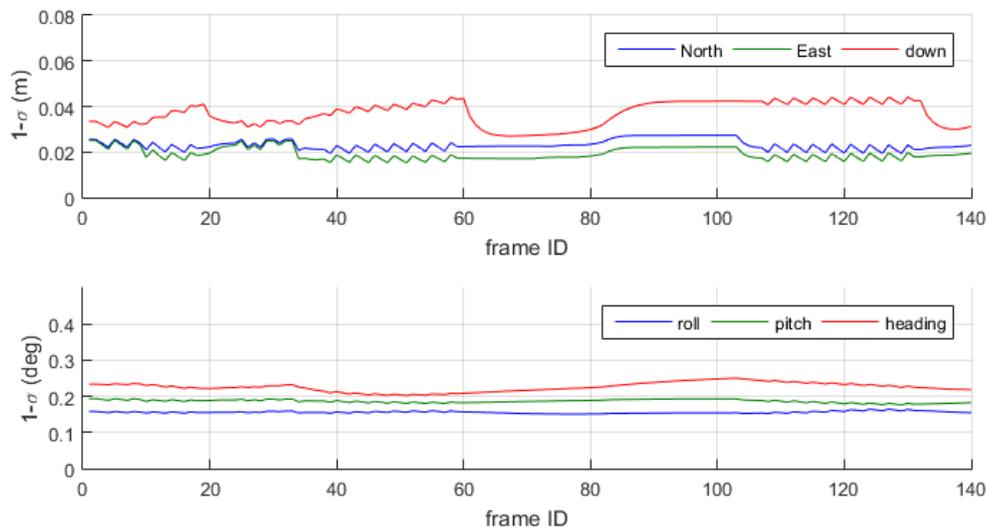


Figure 7.9 The standard deviations of the EO parameter for the image frames

The next subsections present and analyze the auto-calibration results associated with COL, SRE1 and SRE2.

### 7.3.2. Camera and system calibration results

The calibration parameters were initialized using the techniques described in Section 4.2.3. The initial lever-arm vector from the camera system to the IMU was measured using a measuring tape. The calibration parameters were then estimated using the least-squares method.

The estimated lens distortion parameters together with their a-posteriori standard deviations with the left and right cameras are given in Table 7.4 and Table 7.5, respectively. The results showed the similar results for the focal length error from COL and SRE2, but SRE1 performed the worst. The best estimated principal point errors came from SRE2, followed by the ones from COL and SRE1. The results showed that the coefficients  $k_1$  and  $k_2$  accounted for most of the radial distortions. More in details, the lowest standard deviations for them were achieved by SRE2, whilst they were similar with COL and SRE1.

Table 7.4 The lens distortion parameters with the left camera

Parameter	COL		SRE1		SRE2	
	mean	stdev	mean	stdev	mean	stdev
$\Delta f_L$ (px)	0.393	0.592	-0.694	0.967	0.393	0.549
$\Delta x_{L,0}$ (px)	-0.787	0.193	-0.938	0.323	-0.787	0.169
$\Delta y_{L,0}$ (px)	0.068	0.730	-0.326	0.923	0.068	0.501
$k_{L,1}$ (px <sup>-2</sup> )	-4.81e <sup>-07</sup>	9.32e <sup>-09</sup>	-4.84e <sup>-07</sup>	9.41e <sup>-09</sup>	-4.77e <sup>-07</sup>	4.92e <sup>-09</sup>
$k_{L,2}$ (px <sup>-4</sup> )	6.89e <sup>-13</sup>	1.26e <sup>-13</sup>	6.92e <sup>-13</sup>	1.45e <sup>-13</sup>	4.92e <sup>-13</sup>	7.40e <sup>-14</sup>
$k_{L,3}$ (px <sup>-6</sup> )	-0.98e <sup>-18</sup>	7.16e <sup>-19</sup>	-1.02e <sup>-18</sup>	7.35e <sup>-19</sup>	-9.77e <sup>-20</sup>	3.72e <sup>-19</sup>

Table 7.5 The lens distortion parameters with the right camera

Parameter	COL		SRE1		SRE2	
	mean	stdev	mean	stdev	mean	stdev
$\Delta f_R$ (px)	2.606	0.602	2.307	0.974	2.544	0.553
$\Delta x_{R,0}$ (px)	-0.545	0.210	-0.691	0.310	-0.495	0.164
$\Delta y_{R,0}$ (px)	2.045	0.805	1.930	0.877	2.009	0.477
$k_{R,1}$ (px <sup>-2</sup> )	-4.95e <sup>-07</sup>	9.36e <sup>-09</sup>	-5.06e <sup>-07</sup>	9.62e <sup>-09</sup>	-4.89e <sup>-07</sup>	5.03e <sup>-09</sup>
$k_{R,2}$ (px <sup>-4</sup> )	8.94e <sup>-13</sup>	1.32e <sup>-13</sup>	9.15e <sup>-13</sup>	1.47e <sup>-13</sup>	5.56e <sup>-13</sup>	7.60e <sup>-14</sup>
$k_{R,3}$ (px <sup>-6</sup> )	-1.75e <sup>-18</sup>	6.45e <sup>-19</sup>	-1.34e <sup>-18</sup>	7.40e <sup>-19</sup>	8.46e <sup>-20</sup>	3.82e <sup>-19</sup>

The estimated relative orientation parameters and their standard deviations are listed in Table 7.6. The estimated baseline vectors were similar from all of them, so were the boresight angles about  $y$  and  $z$ . Besides, the estimated boresight angles about  $x$  were similar from COL and SRE2. SRE1 estimated this same boresight angle at the lowest accuracy in comparison with the other two.

Table 7.6 The relative orientation of the right camera w.r.t the left camera.  
† free parameter

Parameter	COL		SRE1		SRE2	
	mean	stdev	mean	stdev	mean	stdev
$b_{LR,x}^c$ (m)	0.019	0.001	0.017	0.001	0.018	0.001
$b_{LR,y}^c$ (m) †	0.65	-	0.65	-	0.65	-
$b_{LR,z}^c$ (m)	-0.011	0.001	-0.009	0.001	-0.010	0.001
$\theta_{cR,x}^c$ (deg)	0.235	0.010	0.223	0.016	0.240	0.008
$\theta_{cR,y}^c$ (deg)	-0.571	0.004	-0.569	0.006	-0.594	0.003
$\theta_{cR,z}^c$ (deg)	-0.050	0.001	-0.054	0.002	-0.053	0.001

Table 7.7 summarizes the estimated lever-arms, the absolute scale factor and the boresight angles of the stereo camera system.

Table 7.7 The estimated lever-arms, scale and bore-sight angles

Parameter	COL		SRE1		SRE2	
	mean	stdev	mean	stdev	mean	stdev
$la_{L,x}^b$ (m)	0.103	0.015	0.122	0.021	0.102	0.015
$la_{L,y}^b$ (m)	-0.283	0.014	-0.310	0.021	-0.294	0.018
$la_{L,z}^b$ (m)	-0.142	0.055	-0.160	0.060	-0.151	0.061
$s_c$	1.012	0.002	1.013	0.002	1.011	0.002
$\theta_{c,x}^b$ (deg)	92.697	0.027	92.424	0.029	92.278	0.027
$\theta_{c,y}^b$ (deg)	0.391	0.029	0.495	0.029	0.454	0.025
$\theta_{c,z}^b$ (deg)	89.485	0.256	89.123	0.331	89.071	0.333

The estimated bore-sight angles and absolute scale factors were similar in all three calibration results. However, the estimated lever-arms were not consistent with each other and showed that their estimates could not be reliable. The standard deviation of the lever-arm in  $z$ -direction was approximately three times larger than the ones in  $x$  and  $y$  directions as there was little variation with the pitch angle and the vertical accuracy of the GPS position was normally two times worse than its horizontal accuracy. The lever-arms were measured beforehand using a measuring tape at the accuracy of 0.5cm. The difference between the estimated and the measured lever-arms, and the standard deviations are presented in Table 7.8.

Table 7.8 Difference between the estimated and measured lever arm components and with the corresponding standard deviations

Component	Measured lever-arm (m)	COL(m)		SRE1(m)		SRE2(m)	
		mean	stdev	mean	stdev	mean	stdev
$x$	0.060	0.043	0.015	0.062	0.021	0.042	0.015
$y$	-0.325	0.042	0.014	0.015	0.021	0.031	0.018
$z$	-0.050	-0.092	0.055	-0.110	0.060	-0.101	0.061

The results in Table 7.8 show that difference between the estimated and measured lever arm components are within three standard deviations ( $3\text{-}\sigma$ ). This implies that the lever-arms could be estimated and the measurements were not accurate enough to recover them reliably. Table 7.9 lists the number of points, number of objects, the size of the parameter, the number of iterations, the total theoretical flops and the a-posteriori variance of unit weight ( $\hat{\sigma}_0^2$ ) from the solutions using COL, SRE1 and SRE2. Furthermore, Table 7.10 lists the components of the parameter vector and their corresponding sizes.

Table 7.9 Number of points, objects, iterations, flops and  $\hat{\sigma}_0^2$

	<b>COL</b>	<b>SRE1</b>	<b>SRE2</b>
Number of stereo images	140	140	140
Number of stereo points	11137	11137	40021
Number of observed objects	2731	2731	9818
Parameter vector size	9051	858	858
Number of least-squares iterations	2	4	5
$\log_{10}(\text{flops})$ (theoretical) per iteration	12.027	8.9608	9.2163
$\log_{10}(\text{flops})$ (theoretical)	12.328	9.5629	9.9153
A-posteriori variance of unit weight $\hat{\sigma}_0^2$	1.54	1.69	1.76

Table 7.10 COL, SRE1 and SRE2 parameter list and size

<b>Parameter</b>	<b>COL</b>	<b>SRE1</b>	<b>SRE2</b>
Image distortion model (6 parameter model, see Section 4.5.2)	2×6	2×6	2×6
Stereo baseline and relative orientation (one baseline component is fixed)	2+3	2+3	2+3
Exterior orientation (one EO parameter is fixed)	6×(140-1)	6×(140-1)	6×(140-1)
Object position parameters	3×2731	0	0
Lever-arm, scale and boresight	3+1+3	3+1+3	3+1+3
<i>Total parameter vector size</i>	<i>9051</i>	<i>858</i>	<i>858</i>

The COL and SRE1 estimates used the same set of measurements. SRE2 employed 3.6 times more measurements. At the same time, SRE1 and SRE2 estimated the same

number of the parameters, but COL estimated 8193 more parameters. Even though, SRE2 processed more measurements than COL, it still used 259 less flops (or 647 times less flops per iteration). However, SRE1 and SRE2 required 2 and 3 more iterations than COL, respectively. This is due to the higher non-linearity of the scale-restraint equation in comparison with the collinearity equations. The a-posteriori variance of unit weight  $\hat{\sigma}_0^2$  for all solutions were greater than 1 (go to Section 7.3.4 for the discussion). The following section evaluates the three calibration results.

### **7.3.3. Evaluation**

To evaluate the calibration results from COL, SRE1 and SRE2, the VO solution based on (3.80) was computed with each of the calibration parameters and then compared with the GNSS aided inertial navigation solution. Since the estimated lever-arm vector was unreliable due to its significant standard error, the tape measured values were employed instead. The trajectory was parsed into thirty-five 200m sections, which all contained continuous image data at the specified data rate. Figure 7.10 shows the color coded sections together with their corresponding ID.

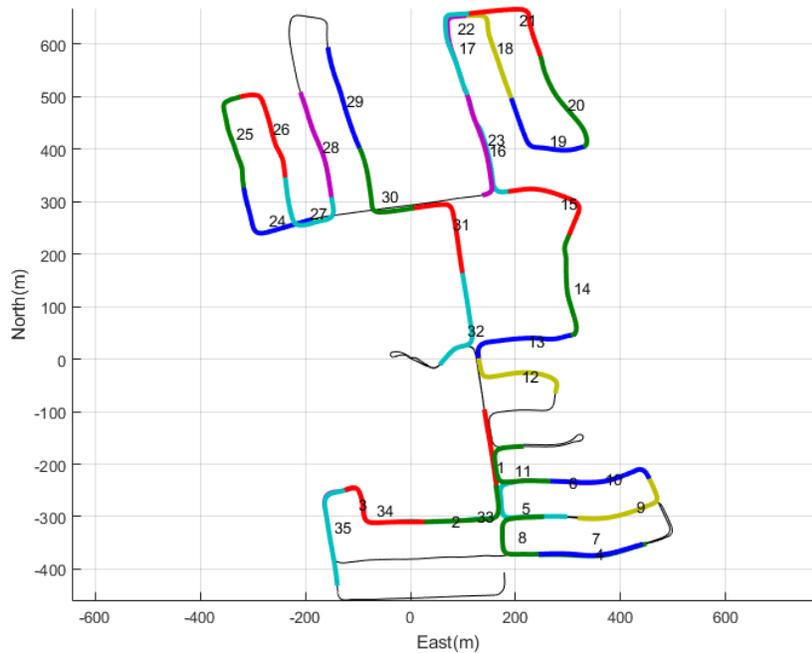


Figure 7.10 The thirty-five 200m sections with Section ID

For each set of calibration parameters, the VO translation and rotation errors were computed in the following way. First, the VO solution and trajectory for each section were computed. The VO frame distance was set to 1.0m and the significance level  $\alpha$  for measurement rejection was set to 0.5%. Then, for each section, the position and orientation differences between the VO path and the reference solution were computed. And finally the translation and rotation drift rates (i.e. error per meter) were computed using the computed position and orientation differences (see Section 3.6.4.6 for more details).

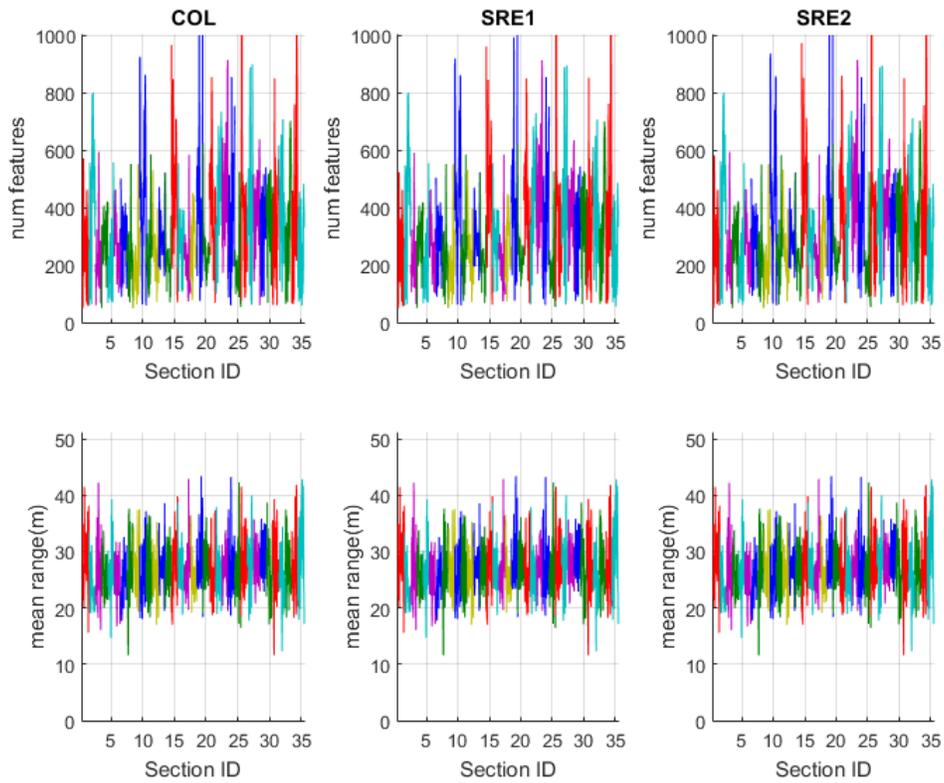


Figure 7.11 The numbers, and mean ranges of the 3D features for the divided sections. The colors of the plot corresponds the sections in Figure 7.10.

The number of the features and the mean range of the 3D features for each section are shown in Figure 7.11 while the number of the image frames, the time interval and the average number of the points per frame in each section are given in Table 7.11. The numbers of features in all three solutions were very similar. The small differences in number were due to the fact that the calibration parameters of COL, SRE1 and SRE2 were not exactly the same and therefore produced slightly different sets of measurements for the VO computation. The magnitudes of the measurement residuals between the solutions were close. Thus, the numbers of the detected outliers were slightly different. Figure 7.12 shows the VO 3D position and rotation drift of all the sections (the color of

the lines in Figure 7.12 corresponds to the section ID in Figure 7.10). Table 7.12 lists the VO translation and rotation drift rates for each section and their RMS errors of drift rates.

Table 7.12 shows that the RMS errors associated with the translation drift rates from COL, SRE1 and SRE2 were 1.78%, 1.86% and 1.70%, respectively. The auto-calibration parameters from SRE2 produced the lowest translation error and therefore were the most accurate solution. The second best and the worst auto-calibration parameters were given by COL and SRE1, respectively. The RMS errors of the estimated rotation drift rates were similar (approximately 0.024deg/m) for all three cases. Conclusively, the estimated VO position change was more sensitive to the variations of the camera calibration values than the estimated VO orientation change, which will further be discussed in the next section.

Table 7.11 Average number of points per frame used in each section.

Section ID	Number of frames	Interval (sec)	Average points per frame		
			COL	SRE1	SRE2
1	164	22.8	289.8	284.3	296.2
2	170	32.5	453.1	450.7	455.8
3	210	41.3	221.0	220.0	223.2
4	190	25.3	308.5	305.7	311.0
5	206	34.3	232.5	230.5	233.3
6	168	34.5	279.2	277.1	280.7
7	190	25.2	314.0	313.9	314.5
8	193	32.0	236.6	235.8	237.6
9	200	29.6	226.6	225.8	228.7
10	172	32.7	480.0	477.7	481.5
11	203	36.9	245.4	244.0	246.2
12	213	33.7	268.5	267.2	270.4
13	195	29.8	303.8	301.7	305.7
14	211	34.1	254.5	253.2	255.7
15	206	32.1	485.9	482.4	487.8
16	197	32.5	290.6	289.3	291.6
17	212	32.7	230.9	229.0	232.5
18	191	27.5	293.4	290.7	296.7
19	193	29.9	486.2	480.0	490.4
20	201	28.5	261.0	259.5	262.5
21	203	33.5	449.3	445.5	452.2
22	199	29.3	436.1	433.2	438.0
23	198	29.9	469.5	465.4	473.2
24	188	25.3	398.7	396.1	403.2
25	197	39.5	213.9	212.1	215.3
26	196	35.0	459.0	454.9	463.4
27	210	34.4	448.5	445.8	450.4
28	174	23.1	427.0	423.6	434.1
29	153	20.3	370.9	361.5	377.6
30	184	26.9	394.1	387.4	401.1
31	165	24.1	390.6	388.0	399.6
32	177	25.7	402.7	397.0	407.8
33	182	25.1	342.4	340.7	343.5
34	192	28.2	465.7	462.3	469.7
35	185	27.7	276.5	271.1	283.6
<i>mean</i>	<i>191.1</i>	<i>30.2</i>	<i>345.9</i>	<i>342.9</i>	<i>349.0</i>

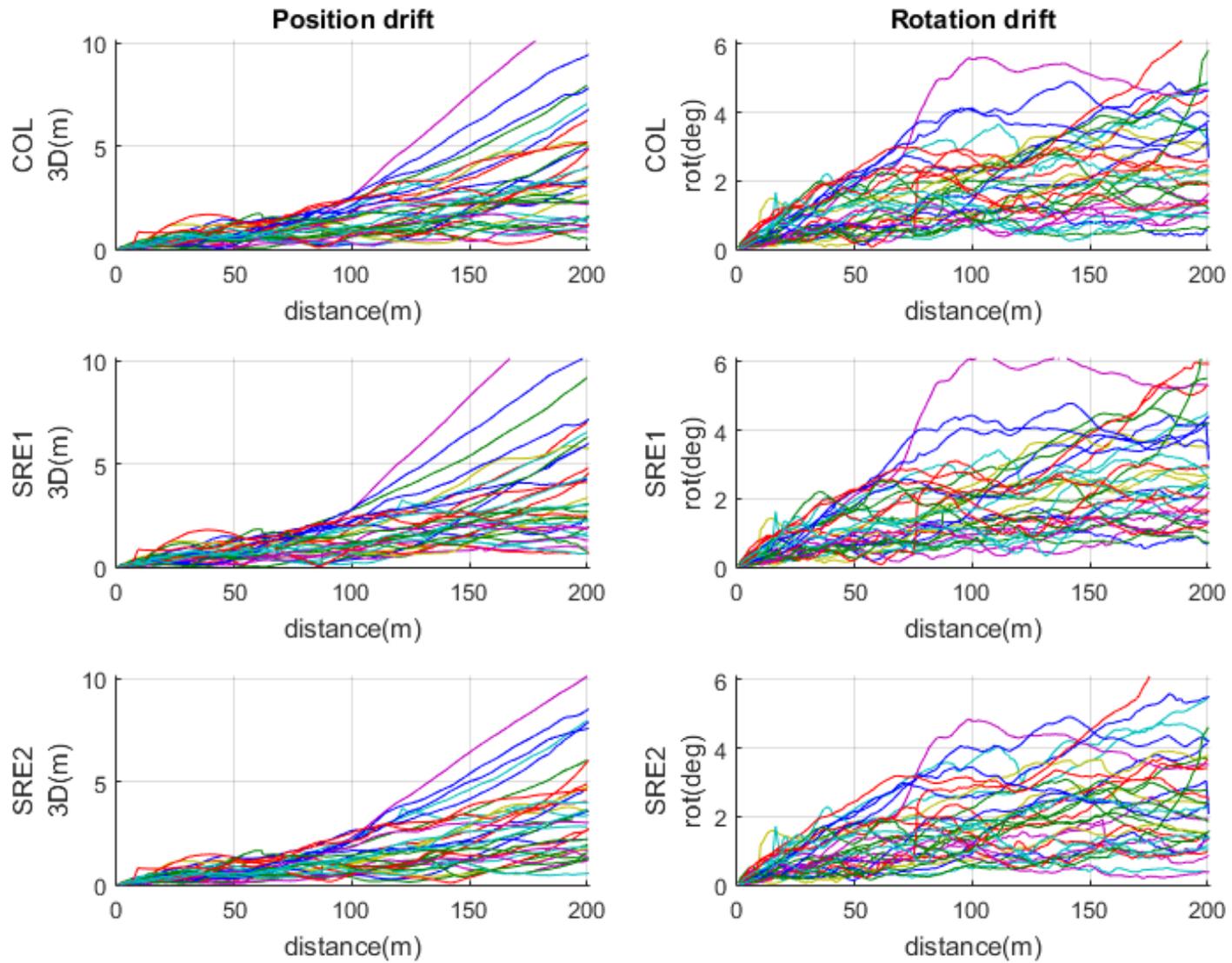


Figure 7.12 3D position and orientation drifts of the individual sections

Table 7.12 The VO translation and rotation drift rates

Section ID	Translation drift rate (%)			Rotation drift rate (deg/m)		
	COL	SRE1	SRE2	COL	SRE1	SRE2
1	2.21	2.28	1.88	0.030	0.034	0.025
2	1.74	1.70	1.89	0.026	0.025	0.029
3	1.70	1.74	1.57	0.015	0.015	0.016
4	1.74	1.96	1.47	0.024	0.027	0.019
5	1.28	1.30	1.34	0.025	0.023	0.028
6	2.77	3.06	2.40	0.039	0.043	0.034
7	1.49	1.46	1.58	0.026	0.025	0.030
8	1.40	1.35	1.49	0.026	0.027	0.025
9	1.81	2.00	1.52	0.026	0.028	0.022
10	1.89	1.83	2.02	0.024	0.021	0.028
11	1.27	1.29	1.25	0.024	0.025	0.024
12	1.24	1.32	1.20	0.021	0.020	0.023
13	1.79	2.05	1.45	0.026	0.029	0.023
14	1.44	1.67	1.23	0.014	0.013	0.017
15	1.66	1.70	1.63	0.022	0.020	0.026
16	1.10	1.18	1.10	0.009	0.011	0.009
17	1.14	1.12	1.23	0.021	0.021	0.021
18	1.66	1.79	1.56	0.019	0.021	0.016
19	2.32	2.42	2.10	0.015	0.015	0.015
20	1.62	1.77	1.43	0.026	0.029	0.020
21	2.06	2.13	2.12	0.039	0.038	0.044
22	1.31	1.45	1.09	0.024	0.026	0.020
23	1.48	1.69	1.19	0.012	0.014	0.011
24	2.04	1.94	2.21	0.028	0.027	0.028
25	2.02	2.06	1.95	0.030	0.033	0.025
26	1.47	1.52	1.43	0.022	0.023	0.021
27	1.98	2.12	1.82	0.027	0.029	0.023
28	1.39	1.32	1.43	0.021	0.021	0.018
29	2.57	2.70	2.31	0.029	0.032	0.025
30	1.95	2.01	1.93	0.017	0.017	0.019
31	1.56	1.47	1.57	0.016	0.015	0.016
32	1.53	1.58	1.52	0.019	0.019	0.020
33	1.20	1.24	1.29	0.020	0.019	0.022
34	2.05	2.11	1.92	0.019	0.019	0.020
35	2.63	2.65	2.63	0.019	0.027	0.015
<i>RMS</i>	<i>1.78</i>	<i>1.86</i>	<i>1.70</i>	<i>0.024</i>	<i>0.025</i>	<i>0.023</i>

#### 7.3.4. Discussion

The results showed that with a given number of measurements, the collinearity equations produced more accurate calibration parameters than the scale-restraint equation and disadvantageously utilized more computational and memory resources. Oppositely, the scale-restraint equation is highly non-linear in comparison with the collinearity equations and required more iterations when running the least-squares bundle adjustment algorithm. Despite the fact that more measurements were processed with more iterations, the scale-restraint equation used less computation resources and delivered more accurate calibration parameters than the collinearity equations.

The test results showed that both methods could not estimate the IMU-camera lever-arms at the expected accuracy. This could be due to (a) the measurements were not accurate enough to provide the absolute and relative positional information at an accuracy of better than 1cm and (b) the estimation was limited by the low dynamics of a land vehicle especially in the vertical. The former is the key to the problem.

The a-posteriori variances of unit weight  $\hat{\sigma}_0^2$  for all solutions were greater than 1. This suggests that (a) the measurement noise model are too optimistic or the weighting between the two sets of measurements were not optimal (b) the measurement vector contained undetected outliers, possibly due to the incorrect feature matching and (c) the measurement functional model may contain unknown errors.

The most essential finding in the test results is that the scale-restraint equation can estimate the camera calibration parameters more accurately than the collinearity equations at a fraction of the computational resources even though more measurements were employed.

## 7.4. Loosely-coupled stereo visual odometry aided-INS

This section presents the road test results with the loosely coupled VO aided-inertial integrated navigation developed in Chapter 5. The loosely coupled VO aided-inertial integrated navigation with the proposed Kalman filter with *pairwise* time-correlated measurements (KF-PTC) was tested and compared with two other versions of the Kalman filter: the standard Kalman filter (KF), the Kalman filter with the standard shaping filter for time-correlated measurements (KF-TC). The data processing was conducted as follows:

- Process the image data to obtain the stereo visual odometry solution for the entire trajectory.
- Compute the loosely coupled VO aided inertial-navigation solutions using the three Kalman filters.
- Evaluate the performance of the three VO aided-inertial solutions and compare them with each other.

### 7.4.1. Stereo visual odometry solution

The visual odometry-based pose changes together with their corresponding variance-covariance matrix and time-correlated covariance matrix were processed independently of the GPS and IMU data. The disadvantage is that the IMU data was not available to improve the feature matching between consecutive image frames. However, in order to ensure the identical VO solution to be used for the three Kalman filters, they were first

derived, saved and then used as measurements in aiding the inertial navigation. The visual odometry solution was computed as follows:

- The SRE2 image distortion parameters, the stereo baseline, orientation of the right camera, the bore-sight angles and the scale factor derived in Section 7.3 were used to calibrate the stereo camera system.
- Since the estimated IMU-camera lever-arm vector was not good enough to replace the measured one by using a steel tape, the measured values were employed instead.
- The VO frame distance was set to a minimum 1.0m or maximum angular change of 3.0deg. The standard deviation of the measurement noise for extracted point features was assumed to be  $\pm 0.3\text{px}$ . The maximum feature range accuracy was set to 3.0m. The significance level  $\alpha$  for measurement outlier rejection was set to 0.5% (approximately  $3\text{-}\sigma$ ).
- The point features were extracted and matched using the LIBVISO2 library [Geiger et al, 2011]. To improve the matching accuracy between the stereo pairs, the search was constrained along the epipolar lines using the coplanarity equations.
- The pose changes were computed using the least-squares estimator and RANSAC was applied to improve the robustness of the estimates.

The VO estimates and their corresponding standard deviations ( $1\text{-}\sigma$ ) are given in Figure 7.13 and Figure 7.14, respectively.

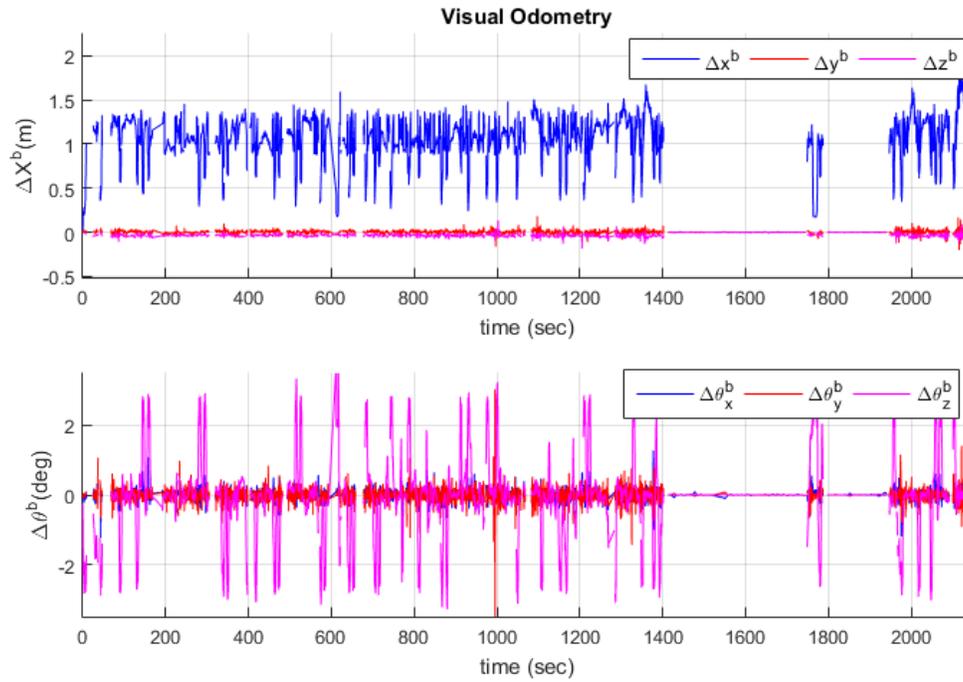


Figure 7.13 The visual odometry solution; the position change (top) and orientation change (bottom)

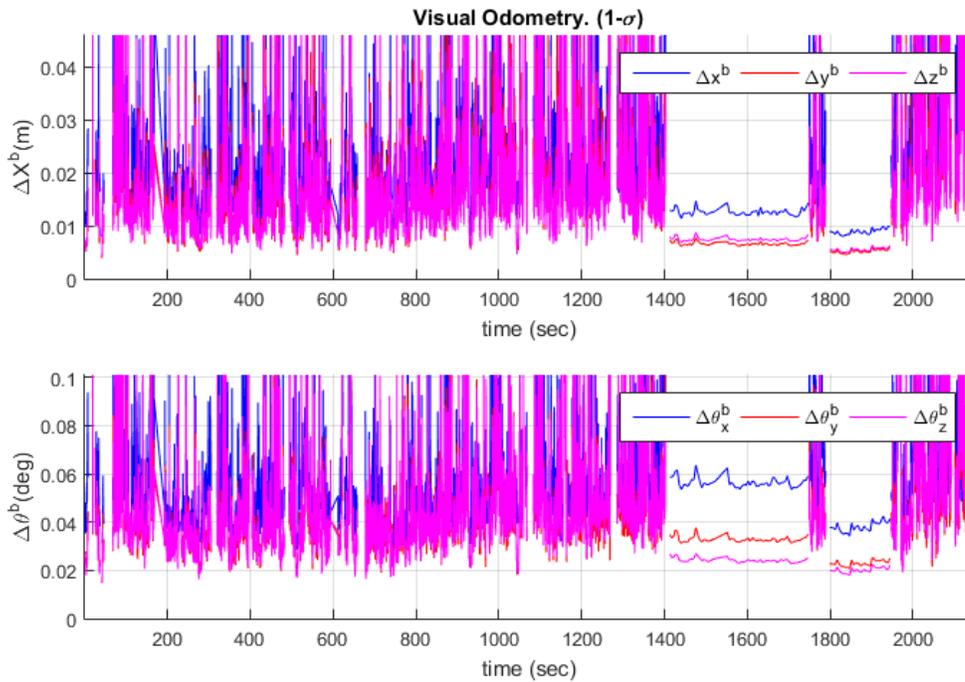


Figure 7.14 The standard deviation (1-σ) of the VO solution

Figure 7.15 shows the number of the features at each epoch, the minimum, average and maximum object feature ranges, and the percentages of the shared features between the current and previous VO estimates. The maximum range was approximately 52m since the objects with the ranges beyond 52m had their accuracy worse than 3.0m and were not included in the VO estimates.

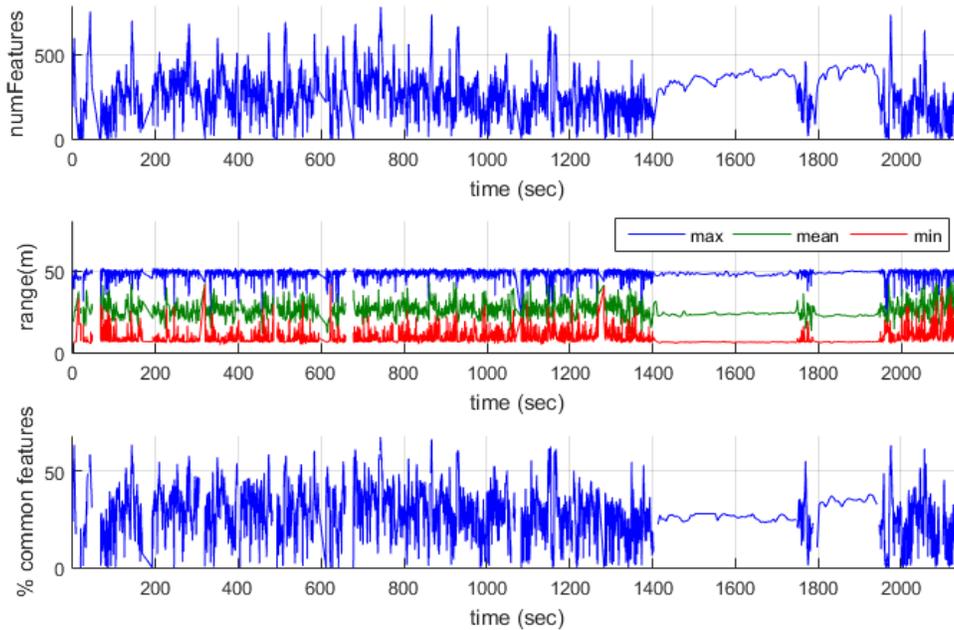


Figure 7.15 The number of the features at each epoch (top), the minimum, average and maximum object feature ranges (middle), and the percentage of the shared features between the current and previous VO solution epochs (bottom).

To check for the consistency, the VO solution was compared with the b-frame position and orientation changes derived from the reference solution (the GNSS-aided inertial integrated navigation solution in Section 7.2.1). Figure 7.16 plots the differences between the VO solution and the reference solution.

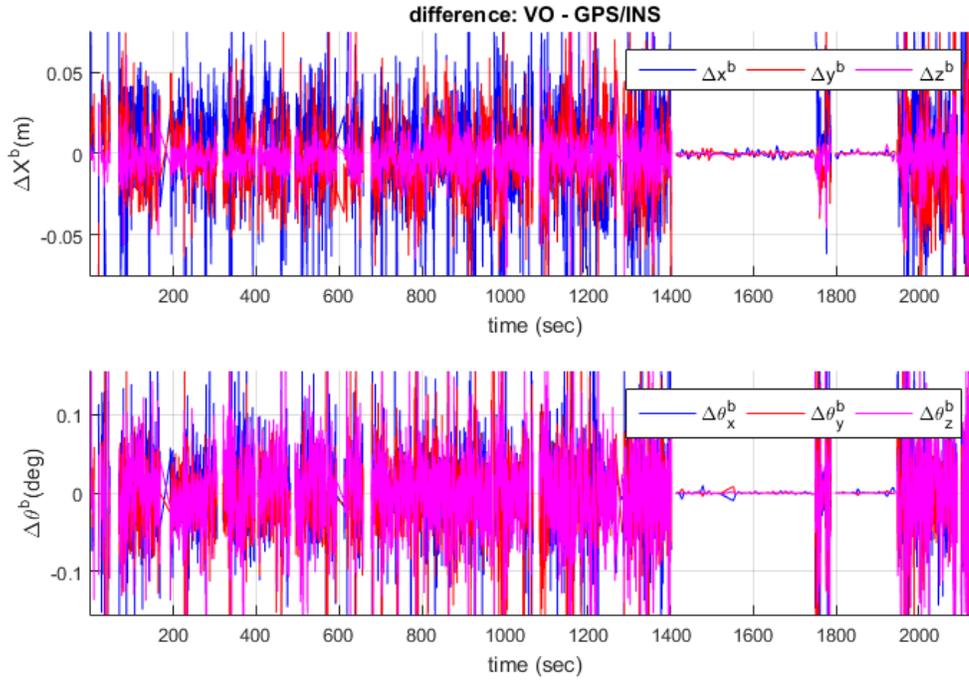


Figure 7.16 The differences between the VO and GNSS-aided inertial integrated navigation solution

Table 7.13 The mean, standard deviation and rms of the differences between the VO and GNSS-aided inertial integrated navigation solution

	$\Delta X^b$ (m)	$\Delta Y^b$ (m)	$\Delta Z^b$ (m)	$\Delta \theta_x^b$ (deg)	$\Delta \theta_y^b$ (deg)	$\Delta \theta_z^b$ (deg)
mean	-0.002	-0.004	-0.003	0.001	-0.007	0.002
1- $\sigma$	0.030	0.019	0.008	0.050	0.044	0.062
rms	0.030	0.020	0.008	0.050	0.044	0.062

The results in Table 7.13 show that the VO solution and the reference solution were consistent with each other with the RMS errors: [0.030, 0.020, 0.008] m and [0.050, 0.044, 0.062] deg for the position and orientation changes, respectively.

#### 7.4.2. VO-aided inertial navigation solution and evaluation

This section presents and compares the loosely coupled VO aided inertial-navigation solution using the three Kalman filters. For this series of tests, GPS position and heading

measurements were used for alignment and also for aiding in loosely-coupled mode. For simplicity, the VO measurements were only employed when the GPS measurements were turned off. The state vector for the KF is

$$\mathbf{x}_k = (\delta\mathbf{X}_k^{n^c}, \delta\mathbf{v}_k^{n^c}, \boldsymbol{\psi}_k, \mathbf{b}_a^b, \mathbf{b}_g^b, \delta\mathbf{X}_{k-m}^{n^c}, \boldsymbol{\psi}_{k-m}^{n^c})^T \quad (7.2)$$

wherein  $\delta\mathbf{X}_k^{n^c}$ ,  $\delta\mathbf{v}_k^{n^c}$ ,  $\boldsymbol{\psi}_k$ ,  $\mathbf{b}_a^b$  and  $\mathbf{b}_g^b$  are the 3x1 subvectors for position error, velocity error, attitude misalignment, accelerometer bias and gyroscope bias, respectively.  $k-m$  is the epoch of the previous image frame. The KF-TC state vector is

$$\mathbf{x}_k = (\delta\mathbf{X}_k^{n^c}, \delta\mathbf{v}_k^{n^c}, \boldsymbol{\psi}_k, \mathbf{b}_a^b, \mathbf{b}_g^b, \delta\mathbf{X}_{k-m}^{n^c}, \boldsymbol{\psi}_{k-m}^{n^c}, \mathbf{v}_k)^T \quad (7.3)$$

where  $\mathbf{v}_k$  is the 6x1 measurement noise vector. The KF-PTC state vector is

$$\mathbf{x}_k = (\delta\mathbf{X}_k^{n^c}, \delta\mathbf{v}_k^{n^c}, \boldsymbol{\psi}_k, \mathbf{b}_a^b, \mathbf{b}_g^b, \delta\mathbf{X}_{k-m}^{n^c}, \boldsymbol{\psi}_{k-m}^{n^c}, \mathbf{v}'_k, \mathbf{v}'_{k-m})^T \quad (7.4)$$

wherein  $\mathbf{v}'_k$  is the 6x1 de-correlated measurement noise vector.

The horizontal static alignment was performed using (3.58) whilst the GPS position and heading measurements were used to initialize the state vector together with its covariance matrix. Then, the respective Kalman filters became ready for navigation.

To assess the performance of three different Kalman filters, 35 GPS outages for 200m long, the same as described in Section 7.3.3, were simulated by excluding the GPS position and heading measurements. Figure 7.10 shows the color coded sections together with their corresponding IDs while Table 7.11 lists the number image frames and the time interval for each section.

For each Kalman filter, the translation and rotation errors for each section were computed in three steps. First, the VO-aided inertial navigation solution was computed. Then, the position and orientation differences between the VO-aided inertial navigation

solution and the reference solution were computed. And finally the translation and rotation drift rates (i.e. error per meter) were computed using the computed position and orientation differences. The drifts in 3D position and rotation, and their drift rates with the associated RMS errors for each section are given in Figure 7.18 and in Table 7.14, respectively.

Table 7.14 shows that the RMS errors of the position drifts rates of KF, KF-ST and KF-PTC were 1.48%, 1.28% and 1.18%. The results show that the position drifts with of KF-PTC improved the solution by 20% and 8% in comparison with the ones from KF and KF-ST, respectively. However, the rotation drift rates remained the same. It is noticed that sections 27 and 25 performed the best and the worst, respectively, whose 2D overviews are given in Figure 7.17.

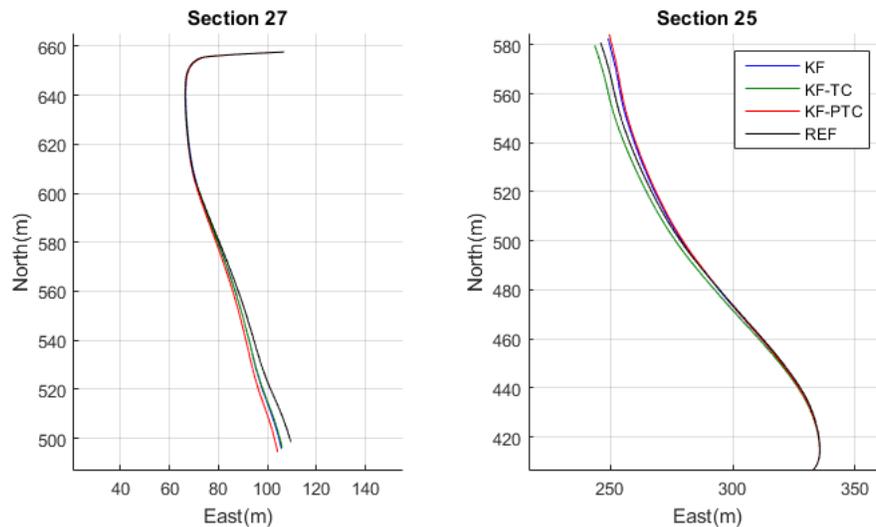


Figure 7.17 The 2D overview of Section 27 (best) and Section 25 (worst)

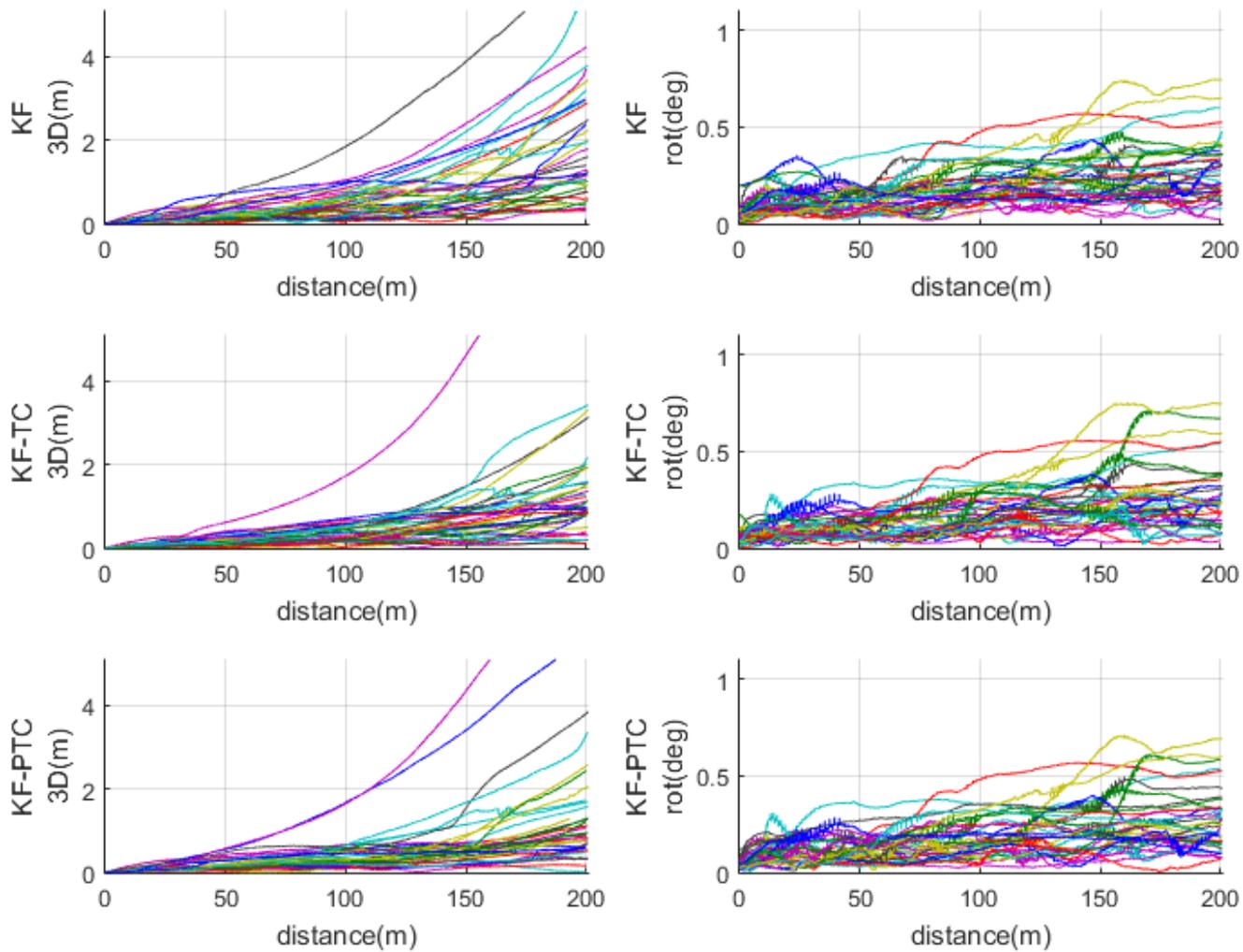


Figure 7.18 The drifts in 3D position and orientation from KF, KF-ST and KF-PT for the individual color coded sections

Table 7.14 The translation and rotation drift rates from the VO-aided inertial integrated navigation

Section ID	Translation error (%)			Rotation error (deg/m)		
	KF	KF-ST	KF-PTC	KF	KF-ST	KF-PTC
1	1.58	0.91	0.64	0.0156	0.0139	0.0141
2	1.01	0.48	0.46	0.0028	0.0021	0.0021
3	0.81	0.52	0.46	0.0066	0.0063	0.0065
4	0.60	1.03	1.01	0.0049	0.0070	0.0029
5	1.69	0.65	0.68	0.0137	0.0138	0.0144
6	0.75	0.47	0.68	0.0032	0.0026	0.0038
7	0.80	0.94	0.47	0.0048	0.0048	0.0051
8	1.87	2.70	1.13	0.0101	0.0122	0.0108
9	0.82	3.08	0.64	0.0055	0.0025	0.0037
10	1.00	0.50	0.93	0.0095	0.0089	0.0102
11	2.36	1.10	2.10	0.0028	0.0037	0.0030
12	1.78	0.36	0.40	0.0047	0.0059	0.0046
13	1.42	1.53	1.97	0.0192	0.0178	0.0192
14	3.43	0.73	1.66	0.0078	0.0071	0.0081
15	0.39	0.51	0.59	0.0101	0.0088	0.0085
16	0.87	1.82	1.21	0.0111	0.0160	0.0183
17	0.70	0.58	0.77	0.0086	0.0086	0.0091
18	3.00	1.86	1.16	0.0128	0.0088	0.0079
19	0.76	0.48	0.85	0.0076	0.0069	0.0071
20	1.62	1.23	0.57	0.0106	0.0065	0.0061
21	0.90	0.70	1.04	0.0027	0.0031	0.0031
22	1.59	0.43	0.75	0.0087	0.0050	0.0066
23	0.65	0.34	0.29	0.0044	0.0019	0.0018
24	2.22	0.47	0.40	0.0075	0.0038	0.0020
25	2.31	4.28	4.53	0.0053	0.0080	0.0059
26	1.15	0.78	1.35	0.0052	0.0049	0.0063
27	0.42	0.25	0.20	0.0077	0.0101	0.0079
28	0.54	0.50	0.47	0.0104	0.0098	0.0096
29	0.64	0.49	0.79	0.0046	0.0044	0.0051
30	0.90	0.44	0.37	0.0056	0.0052	0.0054
31	0.44	0.77	0.36	0.0007	0.0024	0.0011
32	2.29	0.55	0.84	0.0166	0.0152	0.0151
33	1.54	0.46	0.55	0.0042	0.0053	0.0057
34	0.41	0.82	0.50	0.0039	0.0042	0.0030
35	1.13	0.96	0.90	0.0020	0.0027	0.0017
rms	1.48	1.28	1.18	0.0086	0.0083	0.0084

The system innovations with the corresponding 1- $\sigma$  error bounds for section 27 are given in Figure 7.19 and Figure 7.20. The system innovation was small w.r.t. its standard deviation. This shows that VO and INS solution are in agreement and all the aiding measurements were of good quality. Figure 7.21 and Figure 7.22 show the system innovation and the corresponding 1- $\sigma$  error bounds for section 25. The system innovation was relatively large especially in the  $\Delta X^b$ ,  $\Delta Y^b$  and  $\Delta \theta_z^b$  components. As shown in Figure 7.15, the number of features available during this interval (from 995 to 1025 seconds) dropped to approximately 50 for most epochs. This degraded the VO solution and hence degraded the integrated solution.

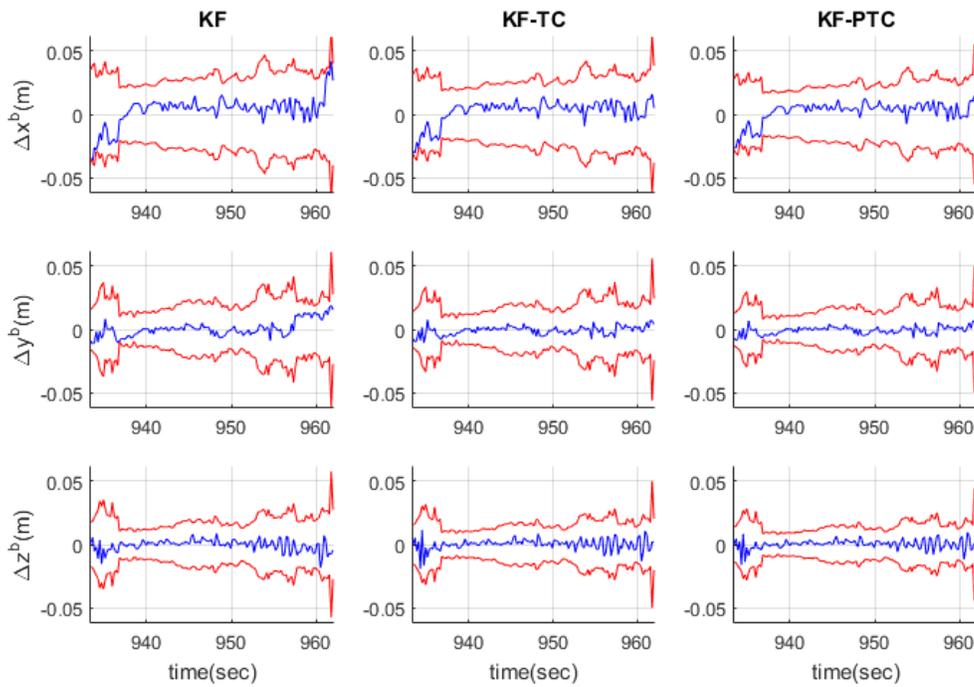


Figure 7.19 Section 27 position change system innovation and 1- $\sigma$  error bounds from KF, KF-ST and KF-PTC

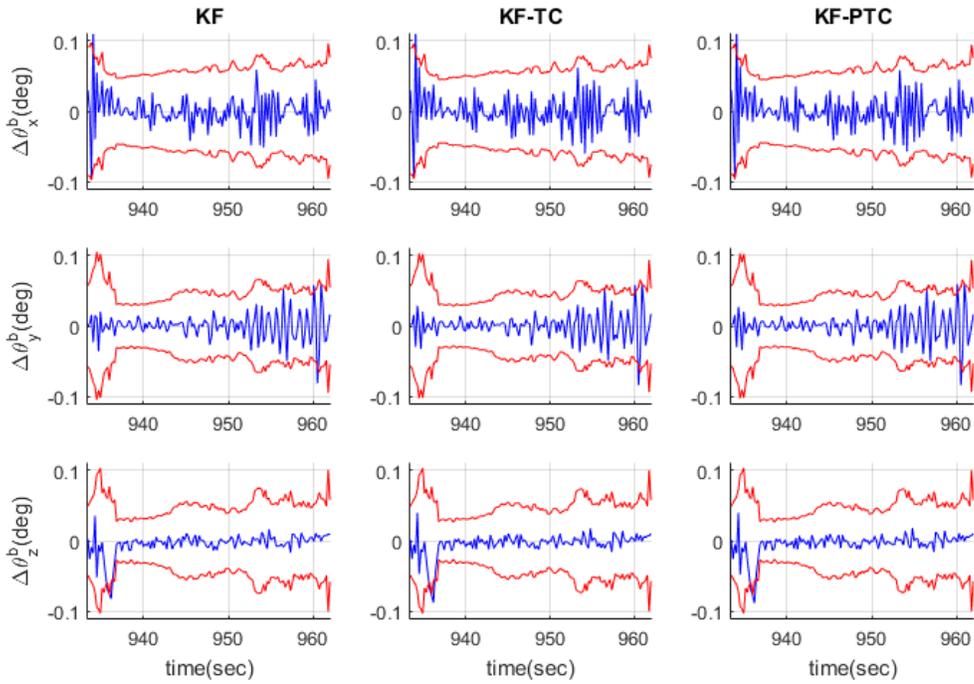


Figure 7.20 Section 27 orientation change system innovation and 1- $\sigma$  error bounds from KF, KF-ST and KF-PTC

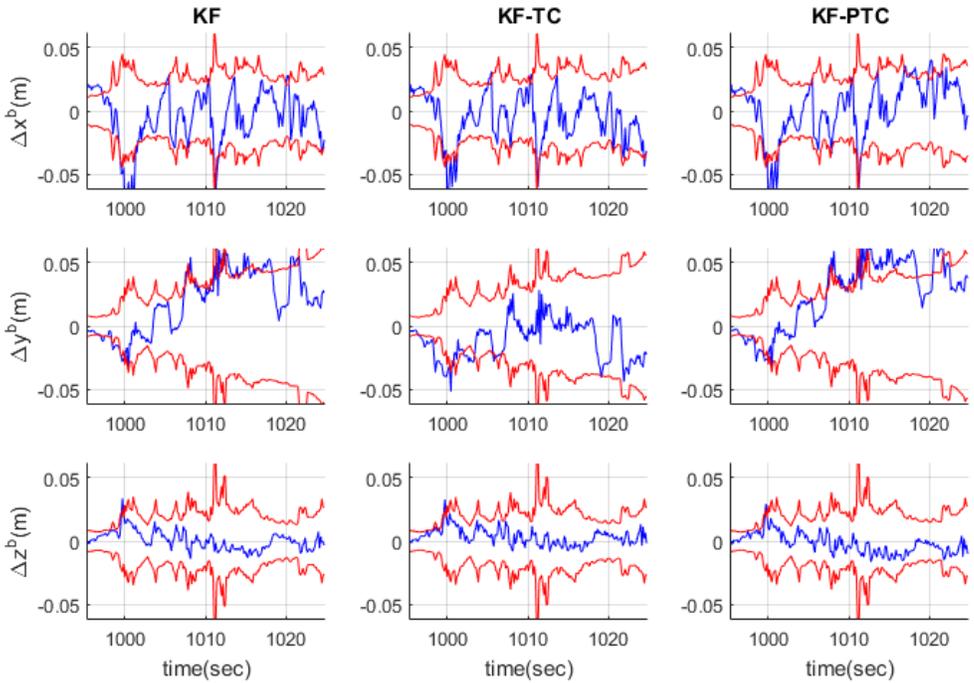


Figure 7.21 The system innovations of the position changes and 1- $\sigma$  error bounds from KF, KF-ST and KF-PTC (Section 25)

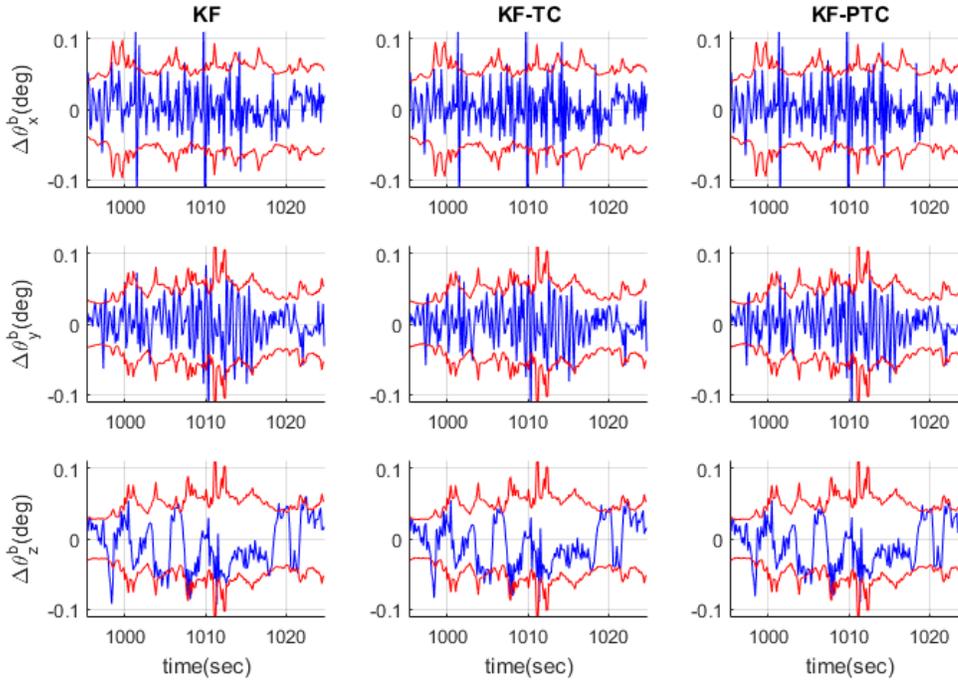


Figure 7.22 The system innovations of the orientation changes and 1- $\sigma$  error bounds from KF, KF-ST and KF-PTC (Section 27)

### 7.4.3. Discussion

The results showed that the measurement model under the consideration of their time-correlation in the VO IA-INS using the KF-TC has reduced the overall position error in comparison with the standard KF solution. The position performance of the VO IA-INS was further improved when the KF-PTC was employed. The position drifted linearly with distance. The position drifts were mainly caused by the scale and heading errors. The former caused the drifts in the along-track direction while the latter caused the drifts in the cross-track direction. Furthermore, these scale and heading errors amplified position error as the distance increased.

The results also showed that the orientation results were similar among the three Kalman filters. The VO aiding does not improve the performance of the roll and pitch

angle estimates. However, the improvement of the heading performance in VO aiding depends on the accuracy of the VO orientation change measurements and the quality of the gyroscopes. Figure 7.14 shows that the standard deviation of the VO orientation changes was, on average, 0.06 deg per frame. From Table 7.3, the random walk and bias stability with the z-gyroscope are 1.94 deg/ $\sqrt{\text{hr}}$  and 8.39 deg/hr respectively. Given that the image rate is 7.5 Hz (or 0.133 sec), the IMU heading error on the level ground in free-inertial mode is expected to be approximately 0.012 deg in 0.133 sec. This shows that the VO orientation change measurements were too noisy to improve the heading solution.

The most important finding in the completed tests was that, the quality of the navigation solution from the Kalman filter was improved by appropriately modelling the time-correlated measurements.

## 7.5. Tightly-coupled stereo MFVO aided-INS

In this section, the tightly-coupled (TC) stereo SLAM-aided INS will be analysed and compared with the TC stereo MFVO-aided INS developed in Chapter 6 using the aboved-utilized road test data. The camera calibration parameters used for LC VO-aided INS tests Section 7.4 was employed for the tests. The TC SLAM aided-INS state vector for  $N$  landmarks is given by

$$\mathbf{x}_k = (\delta\mathbf{X}_k^{n^c}, \delta\mathbf{v}_k^{n^c}, \boldsymbol{\psi}_k, \mathbf{b}_a^b, \mathbf{b}_g^b, \mathbf{m}_1^{n^c}, \mathbf{m}_2^{n^c}, \dots, \mathbf{m}_N^{n^c})^T \quad (7.5)$$

wherein  $\delta\mathbf{X}_k^{n^c}$ ,  $\delta\mathbf{v}_k^{n^c}$ ,  $\boldsymbol{\psi}_k$ ,  $\mathbf{b}_a^b$ ,  $\mathbf{b}_g^b$  and  $\mathbf{m}_i^{n^c}$  (for  $i=1, 2, \dots, N$ ) are the position error, velocity error, attitude misalignment, accelerometer bias, gyroscope bias and landmark position vectors respectively. The state vector for the TC MFVO aided-INS is given by

$$\mathbf{x}_k = (\delta\mathbf{X}_k^{n^c}, \delta\mathbf{v}_k^{n^c}, \boldsymbol{\psi}_k, \mathbf{b}_a^b, \mathbf{b}_g^b, \delta\mathbf{X}_{k-j_1}^{n^c}, \boldsymbol{\psi}_{k-j_1}^{n^c}, \delta\mathbf{X}_{k-j_2}^{n^c}, \boldsymbol{\psi}_{k-j_2}^{n^c}, \dots, \delta\mathbf{X}_{k-j_M}^{n^c}, \boldsymbol{\psi}_{k-j_M}^{n^c})^T \quad (7.6)$$

where  $k - j_i$  (for  $i=1, 2, \dots, M$  and  $0 < j_1 < \dots < j_{m-1} < j_m$ ) are the frame epochs and  $k - j_m$  is the epoch at which the feature in the current feature list at epoch  $k$  was first observed. The road test dataset was processed with the two algorithms and the results evaluated.

### 7.5.1. SLAM and MFVO-aided inertial navigation solution and evaluation

With this series of tests, the GPS position and heading measurements were used for alignment and also aiding in loosely-coupled mode. The horizontal static alignment was performed using (3.58). Then, the GPS position and heading measurements were used to initialize the state vector together with its covariance matrix. The computation of the 3D feature points for both algorithms is summarized as follows:

- The frame distance between consecutive frames was set to a minimum of 1.0m or maximum angular change of 3.0 deg.
- The point features were extracted and matched using the LIBVISO2 library [Geiger et al, 2011]. To improve the matching accuracy between stereo pairs the search was constrained along the epipolar lines using the coplanarity equations. Between consecutive frames, the locations of the features on the next frame were predicted by using the inertial navigator solution.
- The standard deviation of the measurements for the 2D features was assumed to be  $\pm 0.3\text{px}$ . The maximum accuracy for the triangulated 3D features was set to 3.0m.

The same method, which was used to assess the LC VO-aided INS algorithms in Section 7.4.2 , was employed to evaluate the TC SLAM-aided INS and TC MFVO-aided INS algorithms. That is, the same 35 GPS outages of 200m long were simulated by

excluding the GPS position and heading measurements. Then, during the GPS outages only the 3D image measurements were used for aiding. And finally, the translation and rotation errors for each section and for the two algorithms were computed, respectively. Figure 7.23 shows the drifts in 3D position and rotation for all the sections. Table 7.15 lists the translation and rotation drift rates for each section and the their RMS errors.

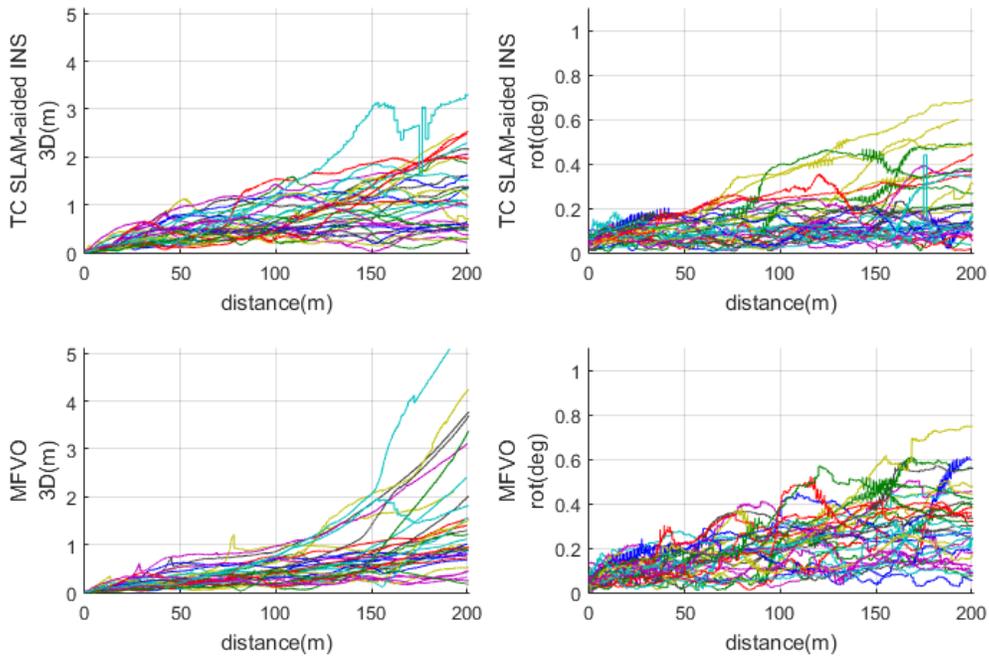


Figure 7.23 The drifts in 3D position and orientation from TC SLAM and TC MFVO for the individual color coded sections

Table 7.15 shows that the RMS errors of the translation drift rates of TC SLAM and TC MFVO were 1.21% and 1.27%, respectively. Furthermore, the changes of the rotation drifts were 0.0086 deg/m and 0.0093 deg/m, respectively. The results showed that the TC MFVO solution was slightly worse than the one from TC SLAM. It is noticed that the drifts with TC MFVO between 0m and 150m distance are smaller than the ones with TC SLAM (Figure 7.23). However, TC MFVO drifts between 150m and 200m distances,

were worse than the ones with TC SLAM, which contributed to the overall degradation of TC MFVO.

Figure 7.24 shows the size of the state vector, the number of the features and the average number of the features per frame from each respective section, and the color coded sections. The size of the state vector of TC SLAM varied approximately between 500 and 1100, as it was between 40 and 80 for MFVO. The average number of features per frame for both algorithms was generally similar for most sections. The small differences may be because (a) the measurement equations are different, (b) the estimated poses were not the same and therefore the feature match results were not identical, and (c) the unmodelled errors may have produced different sets of inliers. The number of TC SLAM measurements during intervals [1005, 1025] sec and [1206, 1232] sec were considerably lower than the ones with TC MVFO. During these two periods, there were numerous SLAM re-initializations due to the low number of features in view.

Table 7.15 The translation and rotation drift rates from the TC SLAM and TC MFVO algorithms

Section ID	Translation error (%)		Rotation error (deg/m)	
	TC SLAM	TC MFVO	TC SLAM	TC MFVO
1	1.16	0.56	0.0030	0.0075
2	1.13	0.59	0.0015	0.0049
3	1.01	0.72	0.0033	0.0107
4	1.20	0.70	0.0021	0.0082
5	1.55	0.94	0.0020	0.0101
6	1.07	0.46	0.0022	0.0031
7	2.09	1.92	0.0201	0.0057
8	1.03	2.54	0.0030	0.0155
9	0.93	0.51	0.0028	0.0014
10	0.91	2.14	0.0071	0.0113
11	1.24	3.27	0.0026	0.0063
12	0.56	0.43	0.0019	0.0069
13	0.64	0.50	0.0126	0.0193
14	1.75	1.89	0.0055	0.0076
15	0.66	0.47	0.0049	0.0049
16	1.27	0.97	0.0135	0.0154
17	1.65	0.94	0.0089	0.0088
18	1.24	0.71	0.0046	0.0066
19	0.90	0.54	0.0034	0.0022
20	1.26	0.78	0.0046	0.0124
21	0.58	1.16	0.0201	0.0085
22	0.96	0.68	0.0033	0.0162
23	1.12	0.61	0.0005	0.0097
24	1.17	1.51	0.0092	0.0075
25	0.78	1.54	0.0019	0.0119
26	1.21	1.38	0.0083	0.0044
27	0.47	0.57	0.0058	0.0063
28	0.80	0.66	0.0055	0.0021
29	1.78	1.03	0.0160	0.0087
30	0.99	1.29	0.0009	0.0108
31	0.94	0.56	0.0144	0.0027
32	1.70	2.42	0.0176	0.0113
33	0.82	0.62	0.0034	0.0069
34	1.21	0.60	0.0104	0.0109
35	2.14	1.18	0.0040	0.0024
<i>rms</i>	<i>1.21</i>	<i>1.27</i>	<i>0.0086</i>	<i>0.0093</i>

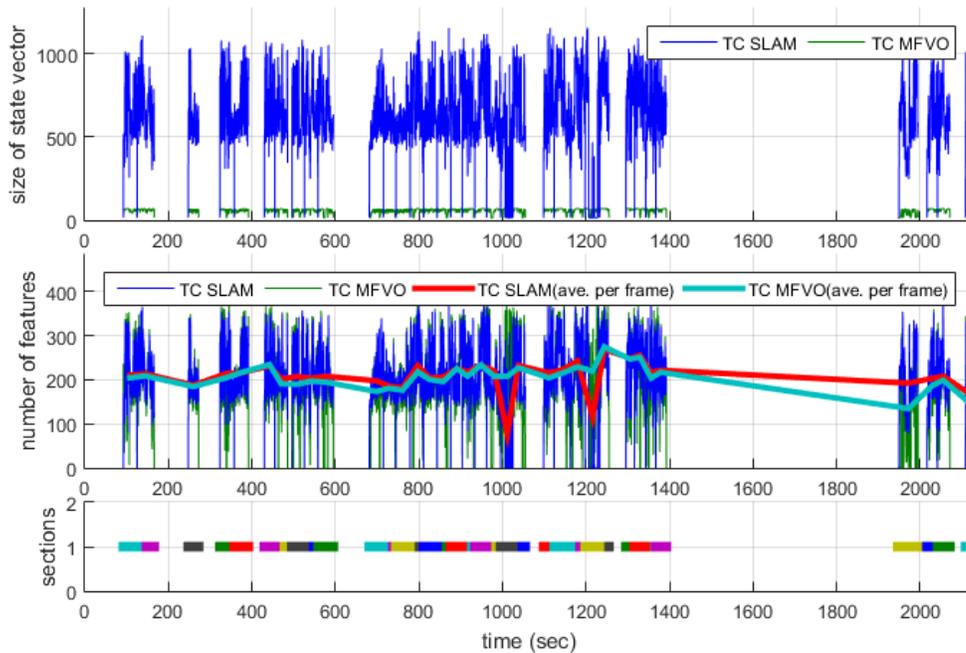


Figure 7.24 The size of the state vector (top). The number of features per frame and the average number of features per frame during the respective sections (middle). The color coded intervals for the section (bottom).

Figure 7.25 shows the histograms of the standardized system innovation vectors for TC SLAM and TC MFVO together with a standardized normal distribution curve. The histograms were narrow in comparison with the standardized normal distribution. This shows that the 3D image measurements and the INS solution were in agreement and the aiding measurements were of good quality.

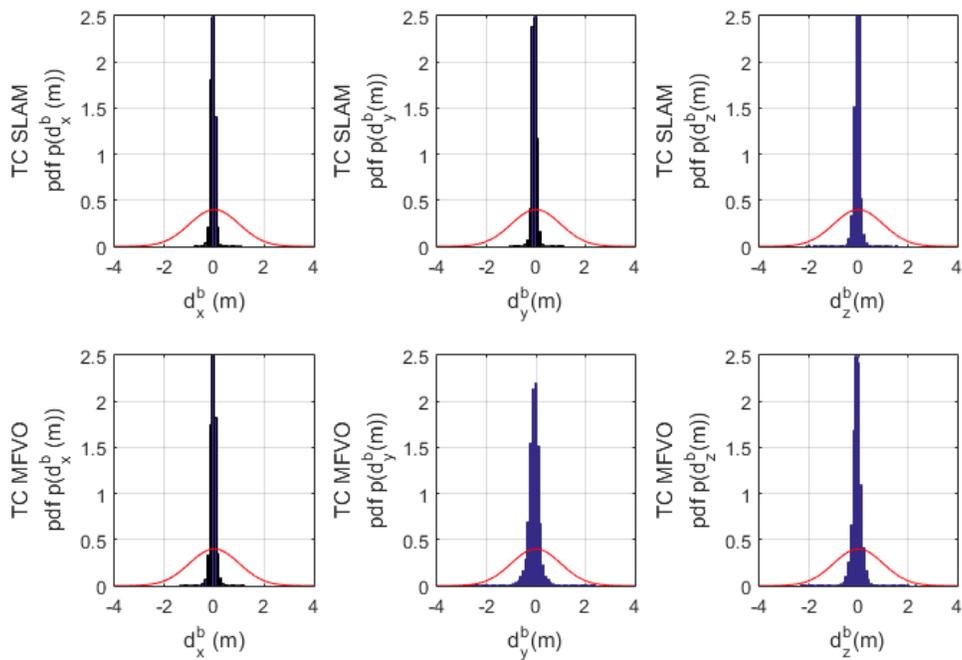


Figure 7.25 The histograms of the standardized system innovation vectors for TC SLAM and TC MFVO

### 7.5.2. Discussion

The results from the simulated data in Section 6.3 showed that the position and orientation solutions from TC SLAM and TC MFVO algorithms were similar. However, the results from the real data here showed that the overall translation drift rate of the TC MFVO translation was 5% worse than the one with the TC SLAM as the TC SLAM translation error drifted as expected, i.e., linearly with the distance. However, the drift associated with the TC MFVO translation increased quadratically with the distance. This error characteristic is not expected with image-based navigation systems. Furthermore, the drift with the TC MFVO was smaller for the first 150m of each section. During the last 50m, TC MFVO drift was much worse than the ones from the TC SLAM. A common understanding is that working with real data could be much more complicated than

working with the simulated data. The TC MFVO error characteristics were most likely due to the combination of the following factors: (a) The characteristics of the MFVO measurement errors was unexpectedly changed when the MFVO measurements were derived by time-differencing the SLAM measurements. For example, the effect of the focal length errors on the 3D feature range estimates were reduced by cancelling common errors in two consecutive SLAM measurements. (b) The MFVO measurements, as shown in (6.23), is the function the features from multiple epochs. This requires accurate timing from the hardware system. The timing errors could have amplified the measurement errors. Further analysis on the error characteristics of the derived MFVO measurements from real data is required. Furthermore, the hardware related investigations on the YUMIS prototype navigation system are needed to obtain definitive results.

Simulation results in Section 6.3.1 showed that TC SLAM and TC MFVO were more accurate than LC VO with KF-PTC. However, test results with the real data showed that the LC VO IA-INS with the KF-PTC in Section 7.4 performed better than the TC SLAM and TC MFVO. This verifies the fact that tightly coupled systems are generally more difficult to implement and requires extensive filter model tuning [Corke et al, 2007]. Furthermore, the research shows that the loosely coupled IA-INS is more robust and suitable for low-cost navigation systems.

As discussed in Section 6.5, the loop closure technique was not applied. The challenge in applying loop closures in MFVO is how to perform the measurement update after a scene has been revisited given that the landmark positions have been cancelled and the map is non-existent. This requires further research and implementation in order to

compare both SLAM and MFVO with loop closures. The MFVO with loop closures is suggested for the future work.

## **8. Conclusion and future work**

### **8.1. Conclusion and contributions**

The focus of this PhD research was on (a) designing and implementing a structureless camera auto-calibration camera and system calibration for a GNSS, IMU and stereo camera integrated navigation system, (b) developing the Kalman filter for processing *pairwise* time-correlated measurements for loosely coupled VO aided-INS, and (c) developing a tightly coupled multi-frame aided INS for drift reduction with the free inertial navigation calculation in poor GNSS and/or GNSS denied environment. Its major contributions are summarized below.

First, Chapter 4 presented the design and implementation along with the analysis of a camera auto-calibration and system calibration algorithm for a GNSS, IMU and stereo camera integrated system based on the scale-restraint equation. The first step determines lens distortion parameters, the up-to-scale baseline length and the relative orientation between the two cameras by employing two scale-restraint equations to constrain the matched features from two consecutive stereo image pairs. Then, the system calibration is introduced to recover the camera lever-arms, and the bore-sight angles with respect to the IMU, and the absolute scale of the camera system using the GNSS-aided inertial navigation solution. The scale-restraint equation has two important characteristics: (a) free from the object position and (a) ensuring the scale consistency between the image positions. For the stereo case, two scale restraint equations are employed per two consecutive stereo pairs, thus ensure that all the common image rays in the stereo-image pairs are operated on the same scale. In comparison with the collinearity equations, the scale-restraint equations use much less computation and memory resources with the same

number of measurements, but at the cost of the accuracy. This study showed that the proposed method required more measurements (approximately 4 times more) to reach a comparable auto-calibration accuracy as the collinearity equations, while using much less computation and memory resources.

Second, a shaping filter for processing of *pairwise* time-correlated measurements was developed in a Kalman filter in Chapter 5. The coefficients of the shaping filter are Cholesky factors obtained from the VCV matrices associated with the measurements. The analysis showed that the global measurement VCV of the VO solution is always positive definite and therefore the Cholesky factors are always unique, real and consists of strictly positive diagonal elements. The derived measurement equation can seamless and optimally integrate VO information into the state-of-the-art GNSS aided-INS system without any modifications to the integration architecture. The results using simulated and real data showed that the proposed method performed better than the standard Kalman filter and the Kalman filter with the conventional shaping filter. Furthermore, the state covariance matrix provides a realistic description of the uncertainty of the state vector.

Third, Chapter 6 presented the development of a tightly coupled stereo MFVO aided INS which integrated features tracked from multiple overlapping image frames for the better VO aiding measurements to efficiently restrict position and orientation drifts during GNSS outages. The MFVO equation was derived from the SLAM measurement equation by algebraically eliminating the landmark position vector over two consecutive epochs. However, the derived measurements are time-correlated. Through a sequential de-correlation algorithm, the Kalman filter measurement update can be performed sequentially and optimally. Furthermore, the Kalman filter in the standard form and

therefore can easily be integrated into the Kalman filter in the current GNSS/INS integration architectures and smoothers. Monte Carlo simulations showed that the solution of the MFVO position and orientation is similar to the ones estimated by the SLAM algorithm. In addition, MFVO requires less memory and computation resources when the number of features in view becomes larger and the percentage overlap becomes smaller.

## **8.2. Future work and recommendations**

Some suggestions for future works and recommendations are:

- Employ the measurement contribution index [Wang, 1997, 2008, 2009] to select measurements in camera auto-calibration. The measurement contribution index of a measurement indicates how much the measurement contributes to the accuracy of a parameter or parameter group. The larger the contribution index, the more effect it has on the parameter or parameter group. In camera-auto calibration, specifically in indoor and land vehicle navigation, the distribution of measurements extracted from the images depends on the environment the camera is sensing. For instance, in land vehicle applications, the objects tend to be far (5-100m) while indoor applications the objects tend to be closer. When auto-calibration is performed in different environments, the calibrated values and their accuracies will be different. Also, a group of parameters can be estimated more accurately when objects are far while others may require closer ones. For instance, bore-sight angles require far objects, internal orientation parameters require dynamics parallel to the image plane and lever-arms require closer objects and high vehicle dynamics. By performing the measurement index analysis, one

can determine quantitatively which conditions are more favorable to estimate each parameter individually. Furthermore, measurements that do not contribute significantly to the estimated parameter can be omitted from the system and thus reduces the computation loading.

- Develop Kalman filter equations for loosely coupled monocular VO aided-INS while considering the time-correlated measurements and scale transfer. The scale restraint equation can compute up-to-scale VO information using measurements from three consecutive image frames. The scale can be transferred by chaining images  $\{1, 2, 3\}$ , then images  $\{2, 3, 4\}$ , and so on and so forth. This ensures that all images operate on the same scale. However, the VO information at epoch  $k$  is correlated with the ones at epochs  $k-1$  and  $k-2$  since some of the same features is employed one and/or two epochs back. Based on the Kalman filter equations for *pairwise* time-correlated measurements presented in Chapter 5, the shaping filter for monocular vision aided-INS can be developed by augmenting the state vector with the pose and de-correlated measurement noise states at epoch  $k-2$ .
- Develop and apply loop-closure techniques for MFVO.
- Employ the York's generic multisensor integration strategy to fuse GNSS, IMU and image measurements [Wang et al, 2015; Qian et al, 2015, 2016; Qian, 2017].
- Utilize a-posteriori variance component estimation (VCE) technique [Wang, 1997, 2008, 2009; Gopaul et al, 2010, Wang et al, 2010, Qian et al, 2016] to tune VO measurements along with other measurements in the navigation Kalman filter. The VCE method simultaneously estimates the process noise and

measurement noise VCV matrices based on the measurement residuals and the process noise residuals.

## References

- Aggarwal, P.; Syed, Z.; Noureldin, A. and El-Sheimy, N. (2010): MEMS-Based Integrated Navigation, GNSS Technology and Application Series. Artech House Publishers, Nov. 2010.
- Alcantarilla, P. F.; Yebes, J. J.; Almazàn, J.; and Bergasa, L. M. (2012): On Combining Visual SLAM and Dense Scene Flow to Increase the Robustness of Localization and Mapping in Dynamic Environments. 2012 IEEE International Conference on Robotics and Automation, River Centre, Saint Paul, Minnesota, USA May 14-18, 2012.
- Allan, D.W. (1966): Statistics of Atomic Frequency Standards. Proceedings of IEEE, vol. 54 no. 2, pp. 221-230, Feb. 1966.
- Badino, H.; Yamamoto, A. and Kanade, T. (2013): Visual Odometry by Multi-frame Feature Integration. Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on DOI: 10.1109/ICCVW.2013.37 Publication Year: 2013 , pp. 222- 229.
- Bay, H.; Ess, A.; Tuytelaars, T. and Van Gool, L. (2008): SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.
- Bayoud F. A.; Skaloud J. and Merminod B. (2004): Photogrammetry-derived Navigation Parameters for INS Kalman Filter Updates. Proceedings of the XXth ISPRS Congress. XXth ISPRS Congress, Istanbul (Turkey), July 14-23.
- Bekir, E. (2007): Introduction to Modern Navigation Systems. World Scientific Publishing Co. Pte. Ltd.

- Bender, D.; Schikora, M.; Sturm, J. and Cremers, D. (2013): Graph-based bundle adjustment for INS-camera calibration. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XL-1/W2, 2013UAV-g2013, 4 – 6 September 2013, Rostock, Germany.
- Benson Jr, D. O. (1975): A Comparison of Two Approaches to Pure-Inertial and Doppler-Inertial Error Analysis. IEEE Trans. Aerospace and Electronic Systems, vol. AES-1 1, no. 4, July 1975.
- Bethel, J.S. (2003): The Civil Engineering Handbook. 2nd Edition, CRC Press, Chapter 56, Photogrammetry and Remote Sensing,
- Bierman, G. J. (2006): Factorization Methods for Discrete Sequential Estimation. Dover Publications (May 26 2006), ISBN 0-486-44981-5
- Bin Rais, N.; Khan, H.A.; Kamran, F. and Jamal, H. (2003): A new algorithm of stereo matching using epipolar geometry. Multi Topic Conference, INMIC 2003. 7th International DOI: 10.1109/ INMIC.2003.1416609, 2003, pp. 21- 24.
- Bisnath, S. B. and Langley, R. B. (2000): Efficient, automated cycle-slip correction of dual-frequency kinematic GPS data. In proceedings of ION GPS 2000 (pp. 145-154).
- Bloesch M., Omari S., Hutter M., and Siegwart R. (2015): Robust visual inertial odometry using a direct EKF-based approach. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 28 Sept.-2 Oct. 2015. DOI: 10.1109/IROS.2015.7353389
- Bouguet, J-Y (2015): Camera Calibration Toolbox for Matlab. Computer Vision Research Group, Dept. of Electrical Engineering, California Institute of

Technology. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/) (accessed on 15 March 2015).

Brown, D.C. (1971): Close-Range Camera Calibration. *Photogrammetric Engineering*, 37(8): 855–866.

Brown, D. (1989): A strategy for multi-camera on the job self-calibration. *Schriftenreihe 14, Institute für Photogrammetrie der Universität Stuttgart, Festschrift Friedrich Ackermann zum 60. Geburtstag*, (pp. 9-21).

Bryson, A.E. Jr and Henrikson, L.J. (1968): Estimation using sampled data containing sequentially correlated noise. *Journal of Spacecraft and Rockets*, 5(6), 662-665.

Callmer, J.; Törnqvist, D.; Gustafsson, F.; Svensson, H. and Carlbom, P. (2011): Radar SLAM using visual features. *EURASIP Journal on Advances in Signal Processing*, 2011:71, 23 September 2011. Springer. 2011. DOI: 10.1186/1687-6180-2011-71

Caspary, W. F. (2000): *Concepts of Network and Deformation Analysis*. Monograph 11. School of Geomatics Engineering, University of New South Wales, UNSW Sydney NSW 2052, Australia. 3<sup>rd</sup> Impression, August 2000.

Cefalu, A.; Haala, N. and Fritsch, D. (2016): Structureless Bundle Adjustment with Self-Calibration Using Accumulated Constraints. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, III-3, 3-9, doi:10.5194/isprs-annals-III-3-3-2016, 2016.

Choi, S.; Kim, T. and Yu, W. (2009): Performance evaluation of RANSAC family. *Proceedings of the British Machine Vision Conference (2009)*.

Chum O. and J. Matas (2008): Optimal Randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472 – 1482, 8 Aug. 2008. DOI: 10.1109/TPAMI.2007.70787

- Clarke, T. and Fryer, J. (1998): The development of camera calibration methods and models. *Photogrammetric Record*, 16(91), 51-66.
- Clement, L. E.; Peretroukhin, V.; Lambert, J. and Kelly J. (2015): The Battle for Filter Supremacy: A Comparative Study of the Multi-State Constraint Kalman Filter and the Sliding Window Filter. *Conference on Computer and Robot Vision (CRV)*, 2015 12<sup>th</sup>. Halifax, NS, Canada. 3-5 June 2015
- Corke, P.; Lobo, J. and Dias, J. (2007): An Introduction to Inertial and Visual Sensing. *The International Journal of Robotics Research* 2007; 26; 519 DOI: 10.1177/0278364907079279
- Crossbow Technology Inc. (2010): 440 Series User's Manual. Document 7430-0131-01, 2010, U.S.A.
- Cvišić, I.; Ćesić, J. ; Marković, I. and Petrović, I. (2017): SOFT-SLAM: Computationally Efficient Stereo Visual SLAM for Autonomous UAVs. *Journal of Field Robotics* 2017. vol. 13, no. 2, pp. 99, 2017. DOI:10.1002/rob.21762.
- Davison, A. and Kita, N. (2001): 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. *CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8-14 Dec. 2001. Kauai, HI, USA, USA. OI: 10.1109/CVPR.2001.990501
- Davison, A. and Murray, D.W. (2002): Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume: 24, Issue: 7, Jul 2002. DOI: 10.1109/TPAMI.2002.1017615.

- Davison, A. (2003): Real-time simultaneous localisation and mapping with a single camera. Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03) 13-16 Oct. 2003. DOI: 10.1109/ICCV.2003.1238654
- Dissanayake, G.; Durrant-Whyte, H. and Bailey, T. (2006): Computationally efficient Solution to the Simultaneous Localisation and Map Building SLAM Problem. Australian Centre for Field Robotics University of Sydney, Sydney, NSW, Australia, 2006.
- Durrant-Whyte, H. and Bailey, T. (2006a): Simultaneous localization and mapping (SLAM): Part I The essential algorithms. Robotics & Automation Magazine, IEEE In Robotics & Automation Magazine, IEEE, Vol. 13, No. 2 (2006), pp. 99–110.
- Durrant-Whyte, H. and Bailey, T. (2006b): Simultaneous localization and mapping (SLAM): Part II state of the art. Robotics & Automation Magazine, IEEE In Robotics & Automation Magazine, IEEE, Vol. 13, No. 3. (2006), pp. 108-117
- Ebner, H. (1976): Self-calibrating block adjustment. Bildmessung und Luftbildwesen, Vol.44, p. 128-139.
- El-Hakim, S. and Faig, W. (1977): Compensation of systematic image errors using spherical harmonics. Proceedings of the American Society of Photogrammetry, papers from the 1977 Fall Technical Meeting, pp. 492-499.
- El-Sheimy, N.; Hou, H. and Niu, X. (2008): Analysis and Modeling of Inertial Sensors Using Allan Variance. IEEE Transactions on Instrumentation and Measurement, VOL. 57, NO. 1, P140, Jan 2008.

- Faig, W. (1975): Calibration of Close-Range Photogrammetric Systems: Mathematical Formulation. XIII Congress of the International Society for Photogrammetry, Helinski, 1976, Commission V, Working Group V/I (Invited Paper)
- Farooque, M. A. and Rohankar, J.S. (2013): Survey on various noises and techniques for denoising the color images. International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 11, November 2013 ISSN 2319 – 4847.
- Farrell, J. A. and Barth, M. (1999): The Global Positioning System and Inertial Navigation. McGraw-Hill, New York, 1999.
- Fischler, M.A. and Bolles, R.C. (1981): Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, June 1981, Volume 24, Number 6, ACM 0001-0782/81/0600-0381, 1981.
- Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2017) On-manifold preintegration for real-time visual-inertial odometry. IEEE Transactions on Robotics. Vol 33(1) Feb. 2017. DOI: 10.1109/TRO.2016.2597321.
- Fraser, C. S. (2012): Automatic camera calibration in close-range photogrammetry. ASPRS 2012 Annual Conference, Sacramento, California, March 19-23, 2012.
- Fraundorfer, F., Scaramuzza, D. and Pollefeys, M. (2010): A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. International Conference on Robotics and Automation (ICRA), 2010 IEEE. 3-7 May 2010. DOI: 10.1109/ROBOT.2010.5509733.

- Gade, K. (2010): A Non-singular Horizontal Position Representation. The Journal of Navigation (2010), 63, p395–417. The Royal Institute of Navigation doi: 10.1017/S0373463309990415
- Gao, Y. and Shen, X. (2002): A New Method for Carrier Phase Based Precise Point Positioning. Navigation, Journal of the Institute of Navigation, Vol. 49, No. 2.
- Geiger, A.; Ziegler, J. and Stiller, C. (2011): StereoScan: Dense 3d Reconstruction in Real-time. Department of Measurement and Control, Karlsruhe Institute of Technology, 2011.
- Gelb, A. (1974): Applied Optimal Estimation. Analytic Sciences Corporation, the MIT press
- Ghosh, S.K. (1988): Analytical Photogrammetry. 2<sup>nd</sup> Edition. 1988, ISBN: 0-08-036103-X.
- Ghosh, S.K. (2005): Fundamentals of Computational Photogrammetry. 2005, ISBN: 81-8069-188-8.
- Giorgi, G.; Teunissen, P.J.G.; Odijk, D. and Buist, P. J. (2010): Enhancing the Time-To-Fix for the Unaided Single-Frequency Integer Ambiguity Resolution in GNSS Attitude Determination Applications. Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION, 4-6 May 2010, Indian Wells, CA, USA
- Gopaul, N.; Wang, J. and Guo. J. (2010): On Posteriori Variance-Covariance Component Estimation in GPS Relative Positioning, Proceedings of CPGPS 2010 Navigation and Location Services: Emerging Industry and International Exchanges, August 2010, Shanghai, China, ISBN 978-1-935068-44-0, Scientific Research Publishing, USA, 2010, pp. 141~148.

- Gopaul, N.; Wang, J. and Baoxin, H. (2014): Discrete EKF with pairwise time-correlated measurement noise for image-aided inertial integrated navigation. Joint ISPRS/IGU 2014 Conference, October 6-8, 2014, Toronto, Canada.
- Gopaul, N.; Wang, J. and Baoxin, H. (2015): Multi-frame Visual Odometry in Image-Aided Inertial Navigation System. Chapter 57, Vol. III, CSNC 2015 Proceedings, Vol. 342, Lecture Notes in Electrical Engineering, Springer, pp. 649-658.
- Gopaul, N.; Wang, J. and Baoxin, H. (2016): Camera auto-calibration in GPS/INS/stereo camera integrated kinematic positioning and navigation system. The Journal of Global Positioning Systems (2016) 14:3 DOI 10.1186/s41445-016-0003-7
- Gopaul, N.; Wang, J. and Baoxin, H. (2017): Loosely coupled visual odometry aided inertial navigation system using discrete extended Kalman filter with pairwise time correlated measurements. 2017 Forum on Cooperative Positioning and Service (CPGPS) , IEEE. 19 October 2017. DOI: 10.1109/CPGPS.2017.8075140.
- Graas, V. and Braasch, M. (1991): GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results. Navigation, Vol. 38, Fall 1991, pp 297-316.
- Grewal, M. S. and Andrews, A. P. (2001): Kalman filtering: Theory and Practice Using MATLAB. Second Edition
- Grewal, M. S.; Andrews A. P. and Bartone, C. G. (2013): Global Navigation Satellite Systems, Inertial Navigation, and Integration,
- Grün, A. (1978): Progress in photogrammetric point determination by compensation of systematic errors and detection of gross errors. Symposium of Comm. III of the ISP, Moscow, pp. 113-140.

- Guivant, J. E. and Nebot E. M. (2001): Optimization of the simultaneous localization and map-building algorithm for real-time implementation. IEEE Transactions on Robotics and Automation, Vol 17, Issue 3, Jun 2001, pp242-257. DOI: 10.1109/70.938382.
- Harris, C. and Stephens, M. (1988): A combined corner and edge Detection. Proceedings of The Fourth Alvey Vision Conference, pages 147–151, 1988.
- Hartley, R.I. (1997): Lines and points in three views and the trifocal tensor. IJCV 22,125–140 (1997)
- Hartley, R. and Zisserman, A. (2000): Multiple View Geometry in Computer Vision. Cambridge, 2000.
- Heipke, C.; Jacobsen, K. and Wegmann, H. (2002): Analysis of the results of the OEEPE test Integrated sensor orientation. Integrated sensor orientation - Test report and workshop proceedings, OEEPE Official Publications No., vol. 43, pp. 31–49, 2002.
- Hewitt, R.A. and Marshall, J. A. (2016): Towards intensity-augmented SLAM with LiDAR and ToF sensors. At the International Conference on Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ I. 28 Sept.-2 Oct. 2015, Hamburg, Germany DOI: 10.1109/IROS.2015.7353634.
- Hofmann-Wellenhof, B.; Lichtenegger, H. and Wasle, E. (2008): Global Navigation Satellite System, GPS, GLONASS, Galileo & More, Springer Wien NewYork, 2008
- Howard, A. (2008): Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems Acropolis Convention Center Nice, France, Sept, 22-26, 2008.

- Indelman, V.; Roberts, R.; Beall, C. and Dellaert, F. (2012): Incremental light bundle adjustment. In British Machine Vision Conf. (BMVC), September 2012.
- Indelman, V.; Melim, A. and Dellaert, F. (2013): Incremental Light Bundle Adjustment for Robotics Navigation. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) November 3-7, 2013, Tokyo, Japan.
- Jeong, Y.; Nister, D.; Steedly, D.; Szeliski, R.; and Kweon, I. (2012): Pushing the Envelope of Modern Methods for Bundle Adjustment. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 34, No. 8 (2012), pp. 1605-1617.
- Julier, S. J. (2001): A Sparse Weight Kalman Filter Approach to Simultaneous Localisation and Map Building. Proceedings of the 2001 IEEE/IRJ, International Conference on Intelligent Robots and Systems Maui, Hawaii, USA, Oct 29 – Nov. 03, 2001.
- Kaess, M.; Ranganathan, A. and Dellaert, F. (2008): iSAM: Incremental Smoothing and Mapping. IEEE Transactions on Robotics Manuscript, Sept 7, 2008
- Kaess, M.; Johannsson, H.; Roberts, R.; Ila, V.; Leonard, J. and Dellaert, F. (2011): iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering. 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center May 9-13, 2011, Shanghai, China,
- Keivan, N. and Sibley, G. (2014): Constant-time Monocular Self-Calibration. Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO) 2014.

- Kelly, J.; Saripalli, S. and Sukhatme, G. S. (2008): Combined Visual and Inertial Navigation for an Unmanned Aerial Vehicle. Field and Service Robotics, Volume 42 of the series Springer Tracts in Advanced Robotics pp 255-264, VI, 2008, 10.1007/978-3-540-75404-6\_24.
- Kelly, J. and Sukhatme, G. S. (2009): Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium.
- Kelly, J.; Matthies, L. H. and Sukhatme, G. S. (2011): Simultaneous Mapping and Stereo Extrinsic Parameter Calibration using GPS Measurement. 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center May 9-13, 2011, Shanghai, China.
- Kenefick, J.F.; Gyer, M.S. and Harp, B.F. (1972): Analytical self calibration. Photogrammetric engineering, Remote sensing, 1972, 38, 1117–1126.
- Konolige, K.; Agrawal, M. and Solà, J. (2007): Large scale visual odometry for rough terrain. In International Symposium on Research in Robotics, pp. 201–212, November 2007.
- Konolige, K and Agrawal M. (2008): FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. IEEE Transactions on Robotics (Vol: 24, Issue: 5, Oct. 2008) p1066-1077. DOI: 10.1109/TRO.2008.2004832.
- Konolige, K. (2010): Sparse sparse bundle adjustment. Proceedings of the British Machine Vision Conference (BMVC).
- Kümmerle, R.; Grisetti, G. and Burgard, W. (2011): Simultaneous calibration, localization, and mapping. 2011 IEEE/RSJ International Conference on Intelligent

Robots and Systems (IROS). 25-30 Sept. 2011. San Francisco, CA, USA. DOI: 10.1109/IROS.2011.6094817.

Lategahn, H.; Geiger, A. and Kitt, B. (2011): Visual SLAM for Autonomous Ground Vehicles. 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center May 9-13, 2011, Shanghai, China.

Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R. and Furgale P. T. (2015): Keyframe-based visual-inertial odometry using nonlinear optimization. The International Journal of Robotics Research. Vol. 34(3)314–334, 2015.

Levenberg, K. (1944): A Method for the Solution of Certain Non-Linear Problems in Least Squares. The Quarterly of Applied Math., Vol. 2 (1944), pp. 164-168.

Li, R. (1996): Super-fast integer ambiguity resolution approach to GPS interferometric heading determination for the application in a low cost integrated GPS/inertial navigation system. Proceedings of the 1996 National Technical Meeting, Santa Monica, CA, USA. Institute of Navigation. p 83-88.

Li, M. and Mourikis, A. I. (2013): High-precision, consistent EKF-based visual-inertial odometry. The International Journal of Robotics Research. Vol 32(6) 690–711, 2013.

Liu, Y. and Zhang, H. (2012): Visual Loop Closure Detection with a Compact Image Descriptor. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems October 7-12, 2012, Vilamoura, Algarve, Portugal.

Liu, Y. Xiong, R., Wang, Y., Huang, H., Xie, X., Liu, X., and Zhang, G. (2016) Stereo Visual-Inertial Odometry With Multiple Kalman Filters Ensemble. IEEE

Transactions on Industrial Electronics. Vol 63 (10). Oct. 2016. DOI:  
10.1109/TIE.2016.2573765

Lourakis, M. and Argyros, A.A. (2009): SBA: A software package for generic sparse bundle adjustment. TOMS 36 (2009) 2.

Lowe, D. G. (1999): Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.

Matthies, L. and Shafer, S. (1987): Error modeling in stereo navigation. IEEE J. Robotics and Automation, vol. RA-3, no. 3, pp. 239–250, June 1987.

McMillan, J.C.; Arden, D.A.G.; Lachapelle, G. and Lu G. (1994): Dynamic GPS Attitude Performance Using INS/GPS Reference. Proceedings of GPS-94, Salt Lake City, September 21-23, The Institute of Navigation, Alexandria, VA, pp.675-682.

Merhav, S. and Bresler, Y. (1986): On-Line Vehicle Motion Estimation from Visual Terrain Information Part 1: Recursive Image Registration. IEEE Transactions on Aerospace and Electronic Systems, Vol. 22, No. 5, 1986, pp. 583–587.

Miller, M. M; Soloviev, A.; de Haag M. U.; Veth, M.; Raquet, J.; Klausutis, T. J. and Touma, J. E. (2011): Navigation in GPS Denied Environments: Feature-Aided Inertial Systems. NATO Research and Technology Organization, 2011.

Mirzaei, F. M. and Roumeliotis, S. I. (2008): A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. IEEE Trans. Robotics, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.

Moe, D., Sampath, A., Christopherson, J., and Benson, M. (2010): Self Calibration of small and medium format digital cameras. ISPRS TC VII Symposium – 100 Years ISPRS, Vienna, Austria, July 5–7, 2010, IAPRS, Vol. XXXVIII, Part 7B.

- Moniwa, H. (1980): The concept of *Photo-variant* Self-calibration and its application in block adjustment with bundles, *Photogrammetria*, 36 (1981) 11-29.
- Moon, S.W. (2000): Design and Implementation of an Efficient Tightly-Coupled GPS/INS Integration Scheme. Proceedings of the 2000 ION GPS-2000 Conference. 159–165. Anaheim, CA: Institute of Navigation, January 2000.
- Moravec, H. (1980): Obstacle avoidance and navigation in the real world by a seeing robot rover. Ph.D. dissertation, Stanford Univ., Stanford, CA, 1980
- Mostafa, M. R. (2001): Boresight Calibration of Integrated Inertial/Camera Systems. Applanix Corporation, 2001.
- Mostafa, M. R. (2002): Camera/IMU boresight calibration: New advance and performance analysis. Proceedings of the American Society for Photogrammetry and Remote Sensing. Annual Meeting, Washington, DC, April 21-26, 2002
- Mourikis, A.I. and Roumeliotis, S.I (2007): A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. 2007 IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007.
- Mourikis, A.I.; Roumeliotis, S.I. and Burdick, J.W. (2007): SC-KF Mobile Robot Localization: A Stochastic Cloning-Kalman Filter for Processing Relative-State Measurements, *IEEE Transactions on Robotics*, 23(4), pp. 717-730, August 2007.
- Mourikis, A.I.; Trawny N.; Roumeliotis, S.I.; Johnson, A.E; Ansar, A and Matthies, L. (2009): Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. *IEEE Transactions on Robotics*, Vol. 25, No. 2, April 2009.
- Murcott, C.; Du Plessis, F. and Meyer, J. (2011): A critique on previous work in vision aided navigation, *Africon 2011*, DOI: 10.1109/AFRICON.2011.6072126.

- Naranjo, C. C. M. (2008): Analysis and Modeling of MEMS based Inertial Sensors, MSc thesis, Stockholm 2008, Signal Processing, School of Electrical Engineering, Kungliga Tekniska Hgskolan.
- Nister, D.; Naroditsky, O. and Bergen, J. (2004): Visual odometry. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. 27 June-2 July 2004. DOI: 10.1109/CVPR.2004.1315094
- Parkinson, B. W. and Spilker Jr., J.J. (1996): Global Positioning System: Theory and Applications Volumes 1&2 Progress in Astronautics and Aeronautics, Volume 163.
- Paz, L.M.; Piniés, P.; Tardos, J. D. and Neira, J. (2008): Large-Scale 6-DOF SLAM With Stereo-in-Hand. IEEE Transactions on Robotics, Vol 24, Issue: 5, Oct. 2008. DOI: 10.1109/TRO.2008.2004637.
- Petovello, M. G.; O’Keefe, K.; Lachapelle, G. and Cannon, M. E. (2009): Consideration of time-correlated errors in a Kalman filter applicable to GNSS. J Geod (2009) 83:51–56 DOI 10.1007/s00190-008-0231-z.
- Pinto, L. and Forlani, G. (2002): A single step calibration procedure for IMU/GPS in aerial photogrammetry. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,, vol. 34 Part B3, pp. 210–219, 2002.
- Point Grey Research® Inc. (2011): Flea3 FL3-FW Technical Reference Manual Version 1.6, Revised 8/29/2011.
- Pope, A. J. (1976): The statistics of residuals and the Detection of Outliers. NOAA Technical Report NOS 65 NGS 1, Rockville Md. 13.

- Prokos, A.; Kalisperakis, I.; Petsa, E. and Karras, G. (2012): Automatic calibration of stereo-cameras using ordinary chess-board patterns. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B5, 2012, XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia
- Qian, K.; Wang, J.; Gopaul, N. and Hu, B. (2012): Low Cost Multisensor Kinematic Positioning and Navigation System with Linux/RTAI, Journal of Sensors and Actuator Networks, 2012, 1(3), 166-182.
- Qian, K.; Wang, J. and Hu, B. (2015): Novel integration strategy for GNSS-aided inertial integrated navigation. Geomatica, 2015, 2, Vol. 69, pp. 217-230.
- Qian, K.; Wang, J. and Hu, B. (2016): A posteriori estimation of stochastic model for multisensor integrated inertial kinematic positioning and navigation on basis of variance component estimation. Journal of GPS, 2016.
- Qian, K. (2017): Generic multi-sensor integration strategy and innovative error analysis for integrated navigation. PhD Thesis, Earth and space science, York University, Canada.
- Reis, J.; Sanguino, J. and Rodrigues, A. (2010): Impact of Satellite Coverage in Single-Frequency Precise Heading Determination. Conference: Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION, Indian Wells, CA, USA, 4-6 May 2010, p 592 – 597.
- Remondino, F. and Fraser, C. (2006): Digital Camera Calibration Methods: Considerations and Comparisons. IAPRS Volume XXXVI, Part 5, Dresden 25-27 September 2006.

- Rodriguez, A. L.; López-de-Teruel, P. E. and Ruiz, A. (2011a): Reduced epipolar cost for accelerated incremental SfM. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on 20-25 June 2011, Providence, RI, USA.
- Rodriguez, A. L.; López-de-Teruel, P. E. and Ruiz, A. (2011b): GEA Optimization for live structureless motion estimation. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on 20-25 June 2011, Providence, RI, USA.
- Scaramuzza, D. and Fraundorfer, F. (2011): Visual Odometry Part I&II: The First 30 Years and Fundamentals. IEEE Robotics & Automation Magazine, December 2011, 18(4):80–92, 2011.
- Rogers, R. M. (2007): Applied Mathematics in Integrated Navigation Systems. Third Edition.
- Roumeliotis, S.I.; Johnson, A.E. and Montgomery, J.F. (2002): Augmenting inertial navigation with image-based motion estimation. Proceedings, IEEE International Conference on Robotics and Automation, 2002, vol.4, pp. 4326- 4333
- Salyshev, O. (2004): Applied Inertial Navigation: Problems and Solutions.
- Sazdovski, V. and Silson, P. M. G. (2011): Inertial Navigation Aided by Vision-Based Simultaneous Localization and Mapping. IEEE SENSORS JOURNAL, Vol. 11, No. 8 (2011), pp. 1646-1656.
- Scherzinger B. M. and Reid B.D. (1994): Modified Strapdown Inertial Navigator Error Models. Position Location and Navigation Symposium, 1994., IEEE, p426-430, Las Vegas, NV, 11-15 Apr 1994.
- Scherzinger, B. M., (2000): Precise Robust Positioning with Inertial/GPS RTK. Proceedings of the 13th International Technical Meeting of the Satellite Division of

The Institute of Navigation (ION GPS 2000), Salt Lake City, UT, September 2000, pp. 155-162

Serrano, L.; Kim, D.; Langley, R. B.; Itani, K. and Ueno, M. (2004): A GPS Velocity Sensor: How Accurate Can It Be? - A First Look. ION NTM 2004, 26-28 January 2004.

Shen, Y. and Liu, J. (2005): Vision based SLAM for Robot Navigation with Single Camera. Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

Siantidis, K. (2016): Side scan sonar based onboard SLAM system for autonomous underwater vehicles. Proceedings at the Autonomous Underwater Vehicles (AUV), 2016 IEEE/OES Conference. Tokyo, Japan. 6-9 Nov. 2016. DOI: 10.1109/AUV.2016.7778671

Sieberth, T.; Wackrow, R. and Chandler, J. H. (2014): Motion blur disturbs - The influence of motion-blurred images in photogrammetry. The Photogrammetric Record, 29(148), pp. 434-453.

Simon, D. (2006): Optimal State Estimation, Kalman,  $H_\infty$  and Nonlinear Approaches. Dan Simon, Cleveland State University.

Sirtkaya, S.; Seymen, B. and Alatan, A.A. (2013): Loosely Coupled Kalman Filtering for Fusion of Visual Odometry and Inertial Navigation. 16th International Conference on Information Fusion Istanbul, Turkey, July 9-12, 2013

Škaloud, J.; Cramer, M. and Schwarz, K.P. (1996): Exterior Orientation by Direct Measurement of Position and Attitude. International Archives of Photogrammetry and Remote Sensing, 31 (B3): 125-130.

- Subbarao, R. and Meer, P. (2006): Beyond RANSAC: User Independent Robust Regression. Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop, IEEE, 0-7695-2646-2/06, 2006
- Steffen, R.; Frahm J.-M. and Förstner, W. (2010): Relative bundle adjustment based on trifocal constraints. Trends and Topics in Computer Vision. Springer Berlin Heidelberg, 2012. pp. 282-295.
- Szeliski, R. (2011): Computer Vision: Algorithms and Applications. Springer-Verlag London Limited 2011, ISBN 978-1-84882-934-3.
- Tang, R (2012): A rigorous and flexible calibration method for digital airborne camera systems. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B1, 2012 XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia
- Tang, R. (2013): Mathematical Methods for Camera Self-Calibration in Photogrammetry and Computer Vision, PhD Thesis, Institute of Photogrammetry, University of Stuttgart.
- Tardif, J.P.; George, M. D.; Laverne, M.; Kelly, A. and Stentz, A. (2010): A new approach to vision-aided inertial navigation. IROS. IEEE, 2010, pp. 4161–4168.
- Teller, S.; Koch, O.; Walter, M. R. and Huang, A. S. (2010): Ground robot navigation using uncalibrated cameras. Proceedings of the 2010 IEEE International Conference on Robotics and Automation, pp.2423-2430, 2010 IEEE.
- Teunissen, P. J. (1995): The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. Journal of geodesy, 70(1-2), 65-82.

- Thrun, S. and Leonard, J. J. (2008): Simultaneous localization and mapping. Springer Handbook of Robotics; Chapter 37, Siciliano, Khatib Editors, ISBN: 978-3-540-23957-4, pp 871–886.
- Titterton, D. and Weston, J. L. (2004): Strapdown inertial navigation technology. 2nd edition, Co-published by the Institution of Engineering and Technology, London, UK and the American Institute of Aeronautics, Reston, Virginia, USA, 2004.
- Triggs, N.; McLauchlan, P.; Hartley, R. and Fitzgibbon, A. (2000); Bundle Adjustment - A Modern Synthesis, Proc. Int'l Workshop Vision Algorithms: Theory and Practice, 2000, pp. 298-372.
- Veth, M. and Raquet, J. (2006) "Two-Dimensional Stochastic Projections for Tight Integration of Optical and Inertial Sensors for Navigation," Proceedings of the 2006 National Technical Meeting of The Institute of Navigation, Monterey, CA, January 2006, pp. 587-596.
- Vollath, U.; Buecherl, A.; Landau, H.; Pagels, C. and Wagner, B. (2000): Multi-Base RTK Positioning Using Virtual Reference Stations. Proceedings of ION GPS 2000, 13th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, Utah, September 19-22:123-131.
- Wang, D.; Liang, H.; Mei, T.; Zhu, H.; Fu, J. and Tao, X. (2013): Lidar Scan matching EKF-SLAM using the differential model of vehicle motion. In the proceedings at the Intelligent Vehicles Symposium (IV), 2013 IEEE. 23-26 June 2013. Gold Coast, QLD, Australia DOI: 10.1109/IVS.2013.6629582.

- Wang, Jianguo (1997): Filtermethoden zur fehlertoleranten kinematischen Positionsbestimmung, Schriftenreihe Studiengang Vermessungswesen, Federal Arm-Forced University Munich, Germany, No. 52, Neubiberg, 1997.
- Wang, J. (2008): Test Statistics in Kalman Filtering, Vol. 7, No. 1, Journal of GPS, pp. 81-90.
- Wang, J. (2009): Reliability Analysis in Kalman Filtering, Vol. 8, No. 1, Journal of GPS, pp. 101-111.
- Wang, J.; Gopaul, N. and Guo, J. (2010): Adaptive Kalman Filter based on Posteriori Variance-Covariance Component Estimation, Proceedings of CPGPS 2010 Navigation and Location Services: Emerging Industry and International Exchanges, August 2010, Shanghai, China, ISBN 978-1-935068-44-0, Scientific Research Publishing, USA, 2010, pp. 115~125.
- Wang, J.; Qian, K. and Hu, B. (2015): An Unconventional Full Tightly-Coupled Multi-Sensor Integration for Kinematic Positioning and Navigation, Chapter 65, in J. Sun et al. (eds), China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III, Lecture Notes in Electrical Engineering 342, Springer-Verlag Berlin Heidelberg 2015.
- Wang, J. and Sternberg, H. (2000): Model Development for Kinematic Surveying of Land Vehicle Trajectories, No. 60-1, Schriftenreihe Studiengang Vermessungswesen der Universitaet der Bundeswehr Muenchen, ISSN 0173-1009, Neubiberg, Bayern, Germany 2000, pp. 317-331.
- Wang, K.; Li, Y. and Rizos, C. (2012): The practical approaches to Kalman filtering with time-correlated measurement errors. IEEE Transactions on Aerospace and

Electronic Systems. 48(2): 1669 - 1681, April 2012.  
DOI:10.1109/TAES.2012.6178086

Wen, K.; Wu, W.; Kong, X. and Liu, K. (2016): A Comparative Study of the Multi-state Constraint and the Multi-view Geometry Constraint Kalman Filter for Robot Ego-Motion Estimation. 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016. Hangzhou, China. 27-28 Aug. 2016.  
DOI: 10.1109/IHMSC.2016.185.

Williams, B. and Reid, I (2010): On Combining Visual SLAM and Visual Odometry. 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District, May 3-8, 2010, Anchorage, Alaska, USA.

Wolf, P. R. and Dewitt, B. A. (2000): Elements of Photogrammetry with Applications in GIS. 3rd Edition, 2000, ISBN 0-07-292454-3.

Wübbena, G.; Bagge, A. and Schmitz, M. (2001): Network-Based Techniques for RTK Applications. Presented at the GPS Symposium, GPS JIN 2001, GPS Society, Japan Institute of Navigation, November 14.-16., 2001, Tokyo, Japan.

Zhang, Z.; Deriche, R.; Faugeras, O. and Luong, Q. T. (1995): A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence, vol. 78, no. 1 - 2, pp. 87 - 119, 1995.

Zhang, Z. (2000): A flexible new technique for camera calibration. IEEE Trans. on PAMI, Vol. 22(11), pp. 1330-1334

Zhang, Z. (2004): Camera Calibration with One-Dimensional Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 26, NO. 7, JULY 2004

Zumberge, J. F.; Heflin, M. B.; Jefferson, D. C. and Watkins, M.M. (1997): Precise point positioning for the efficient and robust analysis of GPS data from large networks. *Journal of Geophysical Research*, Vol. 102, No. B3, pp. 5005—5017.

## Appendices

### Appendix A: KF-PTC state covariance update in Joseph stabilized form

This section derives the state covariance measurement update in Joseph stabilized form for the Kalman filter with pairwise time correlated measurements. The system innovation covariance matrix  $\mathbf{S}_k$ , the Kalman gain  $\mathbf{K}_k$  and the state covariance matrix  $\mathbf{P}_k^+$  for the Kalman filter with pairwise time correlated measurements are (i.e. from (5.37))

$$\begin{aligned}
 \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \boldsymbol{\Sigma}_k \\
 \mathbf{K}_k &= (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k, \mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k, k-1}^T) \mathbf{S}_k^{-1} \\
 \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_{k, k-1} \mathbf{P}_{\mathbf{v}'_{k-1} \mathbf{x}_k}^-
 \end{aligned} \tag{A.1}$$

Starting with  $\mathbf{P}_k^+$

$$\begin{aligned}
 \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_{k, k-1} \mathbf{P}_{\mathbf{v}'_{k-1} \mathbf{x}_k}^- \\
 &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_{k, k-1} \mathbf{P}_{\mathbf{v}'_{k-1} \mathbf{x}_k}^- \\
 &= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k, k-1} \mathbf{P}_{\mathbf{v}'_{k-1} \mathbf{x}_k}^-) \cdots \\
 &\quad - (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k, k-1}^T) \mathbf{K}_k^T + (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k, \mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k, k-1}^T) \mathbf{K}_k^T
 \end{aligned} \tag{A.2}$$

The transpose of  $\mathbf{K}_k$  in (A.1) can be written as

$$\mathbf{K}_k^T = \mathbf{S}_k^{-1} (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k, k-1} \mathbf{P}_{\mathbf{v}'_{k-1} \mathbf{x}_k}^-). \tag{A.3}$$

Substitute (A.3) with in the last  $\mathbf{K}_k^T$  in (A.2) one obtains

$$\begin{aligned}
\mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) - (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{K}_k^T \cdots \\
&\quad + (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{S}_k^{-1} (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) \\
&= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) - (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{K}_k^T \\
&\quad + (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) [\mathbf{S}_k^{-1} \mathbf{S}_k] \mathbf{S}_k^{-1} (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) \\
&= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) - (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{K}_k^T \\
&\quad + [(\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{S}_k^{-1}] \mathbf{S}_k [(\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{k,\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{S}_k^{-1}]^T \\
&= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) - (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{K}_k^T \\
&\quad + \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T
\end{aligned} \tag{A.4}$$

Substitute  $\mathbf{S}_k$  from (A.1) in (A.4)

$$\begin{aligned}
\mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- + \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^-) - (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T) \mathbf{K}_k^T \\
&\quad + \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{\Sigma}_k) \mathbf{K}_k^T \\
&= (\mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T) + \mathbf{K}_k \mathbf{\Sigma}_k \mathbf{K}_k^T \\
&\quad - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^- - \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \mathbf{K}_k^T \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{\Sigma}_k \mathbf{K}_k^T \\
&\quad - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^- - \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \mathbf{K}_k^T
\end{aligned} \tag{A.5}$$

Hence

$$\begin{aligned}
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^- \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{\Sigma}_k \mathbf{K}_k^T \\
&\quad - \mathbf{K}_k \mathbf{C}_{k,k-1} \mathbf{P}_{\mathbf{v}'_{k-1}\mathbf{x}_k}^- - \mathbf{P}_{\mathbf{x}_k \mathbf{v}'_{k-1}}^- \mathbf{C}_{k,k-1}^T \mathbf{K}_k^T
\end{aligned} \tag{A.6}$$

## Appendix B: Sequential block inversion of a lower triangle matrix

This section derives the sequential block inversion of a lower triangle matrix, i.e. the block elements  $\mathbf{L}_{k,1}, \dots, \mathbf{L}_{k,k}$  in Equation (A.7) are obtained from  $\mathbf{C}_{k,1}, \dots, \mathbf{C}_{k,k}$  and  $\mathbf{L}_{i,1}, \dots, \mathbf{L}_{i,i-1}, \mathbf{L}_{i,i}$  for  $1 \leq i \leq k-1$ .

$$\begin{pmatrix} \mathbf{C}_{1,1} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_{k-1,1} & \cdots & \mathbf{C}_{k-1,k-1} & \mathbf{O} \\ \mathbf{C}_{k,1} & \cdots & \mathbf{C}_{k,k-1} & \mathbf{C}_{k,k} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{L}_{1,1} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{L}_{k-1,1} & \cdots & \mathbf{L}_{k-1,k-1} & \mathbf{O} \\ \mathbf{L}_{k,1} & \cdots & \mathbf{L}_{k,k-1} & \mathbf{L}_{k,k} \end{pmatrix} \quad (\text{A.7})$$

Consider block inverse identity

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{F}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{F}_{22}^{-1} \\ -\mathbf{F}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{F}_{22}^{-1} \end{pmatrix} \quad (\text{A.8})$$

wherein  $\mathbf{F}_{11} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$  and  $\mathbf{F}_{22} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$ . With  $\mathbf{A}_{12} = \mathbf{O}$  Equation

(A.8) becomes

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}_{11}^{-1} & \mathbf{O} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{A}_{22}^{-1} \end{pmatrix}. \quad (\text{A.9})$$

The inverse for  $k = 1$

$$\mathbf{C}_{11}^{-1} = \mathbf{L}_{11}. \quad (\text{A.10})$$

The inverse for  $k = 2$

$$\begin{aligned} \begin{pmatrix} \mathbf{C}_{11} & \mathbf{O} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix}^{-1} &= \begin{pmatrix} \mathbf{C}_{11}^{-1} & \mathbf{O} \\ -\mathbf{C}_{22}^{-1}\mathbf{C}_{21}\mathbf{C}_{11}^{-1} & \mathbf{C}_{22}^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L}_{11} & \mathbf{O} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix}. \end{aligned} \quad (\text{A.11})$$

The inverse for  $k = 3$

$$\begin{aligned}
\begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{0} \\ \mathbf{C}_{31} & \mathbf{C}_{32} & \mathbf{C}_{33} \end{pmatrix}^{-1} &= \begin{pmatrix} \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix}^{-1} & \mathbf{0} \\ -\mathbf{C}_{33}^{-1}(\mathbf{C}_{31} & \mathbf{C}_{32}) \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix}^{-1} & \mathbf{C}_{33}^{-1} \end{pmatrix} \\
&= \begin{pmatrix} \begin{pmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix} & \mathbf{0} \\ -\mathbf{L}_{33}(\mathbf{C}_{31} & \mathbf{C}_{32}) \begin{pmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix} & \mathbf{L}_{33} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{L}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \mathbf{0} \\ -(\mathbf{L}_{33}\mathbf{C}_{31}\mathbf{L}_{11} + \mathbf{L}_{33}\mathbf{C}_{32}\mathbf{L}_{21}) & -\mathbf{L}_{33}\mathbf{C}_{32}\mathbf{L}_{22} & \mathbf{L}_{33} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{L}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} \end{pmatrix}
\end{aligned} \tag{A.12}$$

The inverse for  $k$

$$\begin{aligned}
& \begin{pmatrix} \mathbf{C}_{1,1} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_{k-1,1} & \cdots & \mathbf{C}_{k-1,k-1} & \mathbf{O} \\ \mathbf{C}_{k,1} & \cdots & \mathbf{C}_{k,k-1} & \mathbf{C}_{k,k} \end{pmatrix}^{-1} \\
&= \begin{pmatrix} \begin{pmatrix} \mathbf{C}_{1,1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{k-1,1} & \cdots & \mathbf{C}_{k-1,k-1} \end{pmatrix}^{-1} & \begin{pmatrix} \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix} \\ -\mathbf{C}_{k,k}^{-1} (\mathbf{C}_{k,1} \cdots \mathbf{C}_{k,k-1}) \begin{pmatrix} \mathbf{C}_{1,1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{k-1,1} & \cdots & \mathbf{C}_{k-1,k-1} \end{pmatrix}^{-1} & \mathbf{C}_{k,k}^{-1} \end{pmatrix} \\
&= \begin{pmatrix} \begin{pmatrix} \mathbf{L}_{1,1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{k-1,1} & \cdots & \mathbf{L}_{k-1,k-1} \end{pmatrix} & \begin{pmatrix} \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix} \\ -\mathbf{L}_{k,k} (\mathbf{C}_{k,1} \cdots \mathbf{C}_{k,k-1}) \begin{pmatrix} \mathbf{L}_{1,1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{k-1,1} & \cdots & \mathbf{L}_{k-1,k-1} \end{pmatrix} & \mathbf{L}_{k,k} \end{pmatrix} \\
&= \begin{pmatrix} \begin{pmatrix} \mathbf{L}_{1,1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{k-1,1} & \cdots & \mathbf{L}_{k-1,k-1} \end{pmatrix} & \begin{pmatrix} \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix} \\ -\mathbf{L}_{k,k} \sum_{j=1}^{k-1} (\mathbf{C}_{k,j} \mathbf{L}_{j,1}) \cdots -\mathbf{L}_{k,k} \sum_{j=k-1}^{k-1} (\mathbf{C}_{k,j} \mathbf{L}_{j,k-1}) & \mathbf{L}_{k,k} \end{pmatrix} \tag{A.13} \\
&= \begin{pmatrix} \mathbf{L}_{1,1} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{L}_{k-1,1} & \cdots & \mathbf{L}_{k-1,k-1} & \mathbf{O} \\ \mathbf{L}_{k,1} & \cdots & \mathbf{L}_{k,k-1} & \mathbf{L}_{k,k} \end{pmatrix}
\end{aligned}$$

$\mathbf{L}_{k,1}, \dots, \mathbf{L}_{k,k}$  in Equation (A.13) can be obtained sequentially by Algorithm A.1.

Algorithm A.1 Determination of  $\mathbf{L}_{k,1}, \dots, \mathbf{L}_{k,k}$  from the inverse of matrix  $\mathbf{C}$

$$\begin{array}{l}
 \mathbf{L}_{k,k} = \mathbf{C}_{k,k}^{-1} \\
 \text{for } i = 1 \dots (k-1) \{ \\
 \quad \mathbf{L}_{k,i} = -\mathbf{L}_{k,k} \sum_{j=i}^{k-1} (\mathbf{C}_{k,j} \mathbf{L}_{j,i}) \\
 \}
 \end{array}$$

Now consider the case where  $\mathbf{C}$  is an off diagonal lower triangular matrix:

$$\begin{pmatrix}
 \mathbf{C}_{1,1} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} \\
 \mathbf{C}_{2,1} & \mathbf{C}_{2,2} & \mathbf{O} & \dots & \mathbf{O} \\
 \mathbf{O} & \mathbf{C}_{3,2} & \mathbf{C}_{3,3} & \dots & \mathbf{O} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 \mathbf{O} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{C}_{k,k}
 \end{pmatrix}^{-1} = \begin{pmatrix}
 \mathbf{L}_{1,1} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} \\
 \mathbf{L}_{2,1} & \mathbf{L}_{2,2} & \mathbf{O} & \dots & \mathbf{O} \\
 \mathbf{L}_{3,1} & \mathbf{L}_{3,2} & \mathbf{L}_{3,3} & \dots & \mathbf{O} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 \mathbf{L}_{k,1} & \mathbf{L}_{k,2} & \mathbf{L}_{k,3} & \dots & \mathbf{L}_{k,k}
 \end{pmatrix}. \quad (\text{A.14})$$

Then Equation (A.13) can simplified to

$$\begin{aligned}
 & \begin{pmatrix}
 \mathbf{C}_{1,1} & \dots & \mathbf{O} & \mathbf{O} \\
 \vdots & \ddots & \vdots & \vdots \\
 \mathbf{O} & \dots & \mathbf{C}_{k-1,k-1} & \mathbf{O} \\
 \mathbf{O} & \dots & \mathbf{C}_{k,k-1} & \mathbf{C}_{k,k}
 \end{pmatrix}^{-1} \\
 &= \begin{pmatrix}
 \begin{pmatrix} \mathbf{L}_{11} & \dots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{k-1,1} & \dots & \mathbf{L}_{k-1,k-1} \end{pmatrix} & \begin{pmatrix} \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix} \\
 -\mathbf{L}_{k,k} \begin{pmatrix} \mathbf{O} & \dots & \mathbf{C}_{k,k-1} \end{pmatrix} & \begin{pmatrix} \mathbf{L}_{11} & \dots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{k-1,1} & \dots & \mathbf{L}_{k-1,k-1} \end{pmatrix} \mathbf{L}_{k,k}
 \end{pmatrix} \\
 &= \begin{pmatrix}
 \begin{pmatrix} \mathbf{L}_{11} & \dots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{k-1,1} & \dots & \mathbf{L}_{k-1,k-1} \end{pmatrix} & \begin{pmatrix} \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix} \\
 -\mathbf{L}_{k,k} \mathbf{C}_{k,k-1} \sum_{j=1}^{k-1} \mathbf{L}_{j,1} & \dots & -\mathbf{L}_{k,k} \mathbf{C}_{k,k-1} \sum_{j=k-1}^{k-1} \mathbf{L}_{j,k-1} & \mathbf{L}_{k,k}
 \end{pmatrix} \quad (\text{A.15})
 \end{aligned}$$

$\mathbf{L}_{k,1}, \dots, \mathbf{L}_{k,k}$  in Equation (A.15) can be obtained sequentially by Algorithm A.2.

Algorithm A.2 Determination of  $\mathbf{L}_{k,1}, \dots, \mathbf{L}_{k,k}$  from the inverse of off diagonal lower triangle matrix  $\mathbf{C}$

$$\begin{aligned} \mathbf{L}_{k,k} &= \mathbf{C}_{k,k}^{-1} \\ \text{for } i &= 1 \dots (k-1) \{ \\ \mathbf{L}_{k,i} &= -\mathbf{L}_{k,k} \mathbf{C}_{k,k-1} \sum_{j=i}^{k-1} \mathbf{L}_{j,i} \\ \} \end{aligned}$$