



ERROR CHARACTERIZATION OF FLIGHT TRAJECTORIES
RECONSTRUCTED USING STRUCTURE FROM MOTION

THESIS

Daniel Alix, Captain, USAF

AFIT-ENY-MS-15-M-214

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT-ENY-MS-15-M-214

ERROR CHARACTERIZATION OF FLIGHT TRAJECTORIES
RECONSTRUCTED USING STRUCTURE FROM MOTION

THESIS

Presented to the Faculty

Department of Aeronautical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautical Engineering

Daniel Alix, B.S.A.E.

Captain, USAF

March 2015

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

AFIT-ENY-MS-15-M-214

ERROR CHARACTERIZATION OF FLIGHT TRAJECTORIES RECONSTRUCTED USING
STRUCTURE FROM MOTION

Daniel C. Alix, B.S.A.E
Captain, USAF

Committee Membership:

Col. Karl Walli, PhD
Chair

Dr. John Raquet
Member

Dr. Kyle Kauffman
Member

Lt. Col. Brian Neff, PhD
Member

Abstract

This research effort assessed the accuracy of Structure from Motion (SFM) algorithms in replicating aircraft flight trajectories. Structure from Motion techniques can be used to estimate an aircraft trajectory by determining the position and pose of an aircraft mounted camera from a sequential series of images taken during flight. As a result, Structure from Motion techniques hold great promise for use in image based Simultaneous Localization and Mapping (SLAM); however, the error associated with these techniques must be understood for incorporation into a robust navigation system. An algorithm is proposed and implemented that successfully reconstructed aircraft trajectories using only a known starting position and a sequential series of images. Theoretical analysis, simulation and flight test data were used to evaluate the performance of the algorithm under a variety of conditions. The error in and reliability of the algorithm was found to be a function of image resolution as well as the amount of overlap and angular separation between sequential images. The trajectory estimated by the algorithm drifted from the true trajectory as a function of distance traveled. The drift was dominated by uncertainty in the scale of the reconstruction as well as angular errors in estimated camera orientations. It was shown that constraining the algorithm with periodic scale and attitude updates significantly improved the solution. A proposed system architecture that incorporated scale and attitude updates was tested on actual flight test data. The architecture successfully reconstructed a variety of trajectories but drift rates were highly variable due to limited perspective change between sequential images and noisy attitude constraints.

Acknowledgements

I would first and foremost like to thank my advisor, Col Karl Walli, for his help in developing the ideas for this project and his guidance in helping me learn the basic concepts of computer vision to which I was completely unfamiliar at the start of this effort. Additionally, I would like to thank the staff of the Autonomy and Navigation Technology Center at AFIT and, in particular, Dr John Raquet, Dr Kyle Kauffman and Mark Smearcheck for their support. The analysis of actual flight test data for this project was made possible by the support of both the Air Force Research Lab Sensors Directorate and the Air Force Test Pilot School. At AFRL, the support of Richard Van Hook, Todd Rovito, Don Venable and Jacob Campbell was invaluable. At the Test Pilot School, the team that executed the MAMI-II flight test, which collected a large amount of data for this research, consisted of Capt Jeremiah Johnson, Capt Matt McCormack, Capt Denise Michaels, Capt Michael Heiman, Capt Derek Marchlewicz and Lt Col Brian Neff. Their efforts collected data that provided much additional insight to the problems analyzed in this research.

Daniel Alix

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	ix
List of Tables	xii
List of Symbols	xiii
List of Abbreviations	xv
I. Introduction	1
1.1 Applications	1
1.1.1 Navigation	1
1.1.2 Reconnaissance	3
1.2 Problem Statement	3
1.3 Research Objectives	3
II. Background	5
2.1 Coordinate Frames	5
2.2 Camera Model	8
2.3 Epipolar Geometry	16
2.4 Structure from Motion Overview	18
2.4.1 Image Correspondences	19
2.4.2 Determining initial sparse 3-D structure	26
2.4.3 Bundle Adjustment	26
III. Methodology	29
3.1 Algorithm Inputs	29
3.2 Algorithm Overview	30
3.2.1 Collect images	30
3.2.2 Apply SFM to Series of Images	36
3.2.3 Real World Transformation	39
3.2.4 Post-VSFM Bundle Adjustment	41
3.2.5 Use Calculated Trajectory for Navigation	45
3.3 Software Tools	48
3.3.1 Algorithm Speed	49

	Page	
3.4	Limitations	50
3.5	Theoretical Error Sources	51
3.5.1	SFM Parameter Dependent Errors and Numerical Errors	51
3.5.2	Accuracy of Bundle Adjustment	52
3.5.3	Factors that that Affect Error through the Jacobian	53
3.5.4	Factors that Affect Error through Measurement Noise	59
3.5.5	Real World Transformation - Angular Errors	60
3.5.6	Real World Transformation - Scale Errors	62
3.5.7	Combining Bundles	64
3.6	Testing Methodology	66
3.6.1	Testing Ideal Cases with Simulation	66
3.6.2	Testing Noisy Cases with Simulation	67
3.6.3	Real World Data	67
IV.	Results and Analysis	69
4.1	Algorithm Verification	69
4.2	Effect of Bundle Adjustment Constraints	71
4.3	Predicting Camera Position Errors	79
4.4	Effect of Pixel Noise	81
4.5	Effect of Resolution, Angular Separation and Image Overlap	83
4.5.1	Effect of Image Resolution	83
4.5.2	Effect of Image Overlap	84
4.5.3	Effect of Camera Angular Separation	86
4.5.4	Interactions between Image Resolution, Angular Separation and GSD	89
4.6	Focal Length Noise	92
4.7	IMU Noise	94
4.8	Model Fit to Simulation Data	96
4.9	Combining Bundles	97
4.10	Summary of Simulation Results	101
4.11	ASPN Data	102
4.12	Angel Fire Data	106
4.13	MAMI-I Data	109
4.14	C-12 Data	113
4.15	MAMI-II Data	113
4.15.1	Trajectory 1	114
4.15.2	Trajectory 2	120
4.15.3	Trajectory 3	124
4.16	Summary of Real World Data	125

	Page
V. Conclusions and Recommendations	127
5.1 Summary of Results	127
5.1.1 Proposed Operational Concept	127
5.2 Major Research Contributions	130
5.3 Recommendations	131
Bibliography	133

List of Figures

Figure		Page
2.1.	Aircraft Body Frame	6
2.2.	ECEF	6
2.3.	East North Up Frame	7
2.4.	Camera Frame	8
2.5.	Pinhole Camera	12
2.6.	Tangential Distortions	15
2.7.	Radial Distortions	16
2.8.	Epipolar Geometry	17
2.9.	Image Correlation	20
2.10.	Difference of Gaussian	21
2.11.	Scale Space in SIFT	21
2.12.	SIFT matching without geometry constraints	24
2.13.	SIFT matching with geometry constraints	25
2.14.	SIFT on two similar images	25
2.15.	Combined A and B Jacobian Matrices	28
3.1.	F-16 with MAMI-II System	35
3.2.	MAMI-II Hardware	36
3.3.	Combined A and B Jacobian Matrices	43
3.4.	Combined A and B Jacobian Matrices with camera rotation and camera parameters fixed	43
3.5.	Navigation Algorithm	46
3.6.	Navigation Algorithm	47
3.7.	Navigation Algorithm	47
3.8.	Navigation Algorithm	48
3.9.	Software Tools	49

Figure		Page
3.10.	Angular Separation	55
3.11.	Angular Separation	56
3.12.	Angular Separation	57
4.1.	3-D View of Simulated Straight and Level Trajectory	69
4.2.	Un-constrained Camera Position Errors - No Noise	70
4.3.	Un-constrained Camera Attitude Errors - No Noise	70
4.4.	Un-constrained Position Errors - . 5 Pixel Noise	72
4.5.	Un-constrained Attitude Errors - . 5 Pixel Noise	72
4.6.	Calibration Constrained Position Errors - . 5 Pixel Noise	73
4.7.	Calibration Constrained Attitude Errors - . 5 Pixel Noise	73
4.8.	Calibration, Attitude Constrained Position Errors - . 5 Pixel Noise	74
4.9.	Calibration, Attitude Constrained Attitude Errors - . 5 Pixel Noise	74
4.10.	Calibration, Attitude, Altitude Constrained Position Errors - . 5 Pixel Noise	75
4.11.	Calibration, Attitude, Altitude Constrained Attitude Errors - . 5 Pixel Noise	76
4.12.	North Error in Final Frame	77
4.13.	Normal Probability Plot of North Error in Final Frame	77
4.14.	Predicted Camera Position Errors	80
4.15.	Actual Camera East Position Errors	81
4.16.	Actual Camera North Position Errors	81
4.17.	Actual Camera Vertical Position Errors	82
4.18.	Three Different Methods for Combining Bundles	99
4.19.	North Error Propagation Using Stereo System	101
4.20.	ASPN Errors - Constant Scale	103
4.21.	ASPN Errors - Updating Scale	104
4.22.	ASPN Errors - Updating Scale	105

Figure		Page
4.23.	ASPN Errors - Updating Scale, Unconstrained rotation	105
4.24.	Angel Fire Errors - Constant Scale	107
4.25.	Angel Fire Errors - Updating Scale	108
4.26.	Angel Fire Errors - Updating Scale	108
4.27.	MAMI I Errors - Constant Scale	110
4.28.	MAMI I Errors - Updating Scale	110
4.29.	MAMI I Errors - Updating Scale	111
4.30.	MAMI-I Segment 1 Actual and Calculated Trajectory	112
4.31.	MAMI-I Segment 1 Errors	112
4.32.	Aircraft Attitude	115
4.33.	MAMI II Errors - Constant Scale	116
4.34.	MAMI II Errors - Updating Scale	116
4.35.	MAMI II Errors - Updating Scale	117
4.36.	MAMI II Errors	118
4.37.	MAMI II Errors	120
4.38.	Aircraft Attitude	121
4.39.	MAMI II Errors - Constant Scale	122
4.40.	MAMI II Errors - Updating Scale	122
4.41.	MAMI II Errors - Updating Scale	123
4.42.	MAMI II Errors - Loop Trajectory	125

List of Tables

Table		Page
3.1.	ASPN Camera Parameters	31
3.2.	Angel Fire Camera Parameters	32
3.3.	MAMI-I Camera Parameters	33
3.4.	MAMI-II Camera Parameters	35
4.1.	Effect of Bundle Adjustment Constraints	78
4.2.	Effect of Pixel Noise	82
4.3.	Effect of GSD	84
4.4.	Effect of Image Overlap	85
4.5.	Effect of Camera Angular	86
4.6.	Effect of Camera Angular	88
4.7.	Effect of Altitude	89
4.8.	Effect of Camera Mounting	90
4.9.	Effect of Trajectory	91
4.10.	Effect of Focal Length Calibration Noise	94
4.11.	Effect of IMU Noise	95
4.12.	Interaction of IMU Noise and Focal Length Constraint	96
4.13.	Combining Bundles	100
4.14.	Combining Bundles with IMU noise	101
5.1.	Recommended System Guidelines	128
5.2.	Operational Parameters and Limitations	129
5.3.	Operational Performance	129

List of Symbols

Symbol		Page
C_n^a	NED to aircraft DCM	9
C_a^c	Aircraft to Camera DCM	9
ψ	Yaw	9
Θ	Pitch	9
ϕ	Roll	9
T^n	Vector between NED frame origin and Earth's Center . . .	10
C_e^n	ECEF to NED frame DCM	10
L	Longitude	10
λ	Latitude	10
f	Focal Length	11
s_x	Pixel size in x direction	12
s_y	Pixel size in y direction	12
s_Θ	Pixel Skew	12
o_x	x position of image center	12
o_y	y position of image center	12
R_{c1}^{c2}	Camera 1 to Camera 2 Frame DCM	17
T^{c2}	Translation between camera 1 and camera 2	17
$\hat{T}R$	Essential Matrix	18
$\hat{T}'K RK^{-1}$	Fundamental Matrix	18
H	Homography Matrix	23
$(C_m^c)^N$	VSFM Model to Camera Frame DCM for Nth Camera . .	40
C_r^m	Reference Camera to VSFM Model Frame DCM	40
C_{a1}^r	First aircraft frame to reference camera frame DCM . . .	40
C_{NED}^{a1}	NED to First Aircraft Frame DCM	40
C_{ENU}^{NED}	ENU to NED Frame DCM	40

Symbol		Page
$(x^m)^N$	Position of Nth Camera in Model Frame	40
$(t^c)^N$	Translation of Nth Camera in Camera Frame	40
$(C_c^m)^N$	Camera to Model frame DCM for Nth camera	40
$(x^m)^1$	Reference camera position in model frame	40
C_{a1}^{NED}	First aircraft frame to NED frame DCM	41
C_r^{a1}	Reference camera frame to first aircraft frame DCM	41
K	Model to World Scale Factor	41
$(C_{NED}^a)^N$	NED to Aircraft Frame DCM for Nth camera	44
$(C_a^c)^N$	Aircraft to Camera Frame DCM for Nth camera	44
P_x	State covariance matrix	52
P_z	Measurement covariance matrix	52
H	Camera AGL Altitude	53
X_2	Position of second camera	63

List of Abbreviations

Abbreviation		Page
SFM	Structure from Motion	1
SLAM	Simultaneous Localization and Mapping	1
INS	Inertial Navigation System	1
GPS	Global Positioning System	1
ECEF	Earth Centered Earth Fixed	5
WGS	World Geodetic System	5
NED	North-East-Down	5
VSFM	Visual Structure from Motion	7
DCM	Direction Cosine Matrix	9
SIFT	Scale Invariant Feature Transform	20
DOG	Difference of Gaussian	20
RANSAC	Random Sample Consensus	23
IMU	Inertial Measurement Unit	30
AFRL	Air Force Research Lab	30
AFIT	Air Force Institute of Technology	31
ANT	Autonomy and Navigation Technology	31
DTED	Digital Terrain Elevation Data	31
NGA	National Geospatial-Intelligence Agency	31
AFRL/RYS	Air Force Research Lab Sensors Directorate	31
ASPN	All Source Positioning and Navigation	31
SPAN	Synchronous Position, Attitude and Navigation	31
SPAN	Synchronous Position, Attitude and Navigation	32
USAF	United States Air Force	33
TPS	Test Pilot School	33
TMP	Test Management Project	33

Abbreviation		Page
AFB	Air Force Base	33
MAMI	Minor Area Motion Imagery	33
NASA	National Aeronautics and Space Administration	33
SPAN	Synchronous Position, Attitude and Navigation	33
RASCAL	Reconfigurable Airborne Sensor, Communications and Laser	34
AGL	Above Ground Level	34
SPAN	Synchronous Position, Attitude and Navigation	35
VSMF	Visual Structure from Motion	36
EXIF	Exchangeable Image File Format	38
ENU	East-North-Up	40
UCLA	University of California Los Angeles	48
VLFeat	Vision Lab Feature	48
VLG	Vision Lab Geometry	48
OpenCV	Open Computer Vision	48
DOF	Degrees of Freedom	52
GSD	Ground Sampling Distance	53
DOP	Dilution of Precision	54
MSL	Mean Sea Level	66
STD	Standard Deviation	78

ERROR CHARACTERIZATION OF FLIGHT TRAJECTORIES RECONSTRUCTED USING STRUCTURE FROM MOTION

I. Introduction

This research assesses the accuracy of a Structure from Motion algorithm in reconstructing aircraft flight trajectories using images taken from a camera system mounted on an aircraft. Structure from Motion (SFM) techniques use a set of overlapping images to reconstruct a three dimensional scene while simultaneously estimating the relative geometry between the target scene and the cameras used to take the images. This process can be used to determine an aircraft's trajectory and attitude relative to underlying terrain by estimating the position and pose of a camera mounted on the aircraft as that camera takes a sequential series of images during flight. Structure from Motion techniques hold great promise for use in vision based Simultaneous Localization and Mapping (SLAM) on aircraft; however, a major challenge for a Structure from Motion based SLAM algorithm is to transform the relative geometry of the reconstructed scene to a real world coordinate system and scale. This paper proposes a method for conducting this transformation in a way that is useful for aerial navigation. The approach is based on existing computer vision techniques and tools but is novel in the way in which these techniques are used. The errors associated with the method are characterized using both simulation and flight test data. This error characterization provides a basis for future work toward the goal of using Structure from Motion for aerial navigation and reconnaissance applications.

1.1 Applications

1.1.1 Navigation. Most military aircraft (autonomous and manned) use information from both an Inertial Navigation System (INS) and a Global Positioning System (GPS) to provide an integrated navigation solution. An INS is a stand alone system of accelerometers and gyroscopes that integrates accelerations to determine a

vehicle's position after a certain amount of time given a known starting point. This does not require external signals which are often susceptible to jamming and denial in wartime scenarios. An INS position solution drifts as a function of time due to errors in the gyroscope and accelerometer measurements [30]. High quality gyroscopes and accelerometers are expensive and therefore it is often cost prohibitive to reduce the drift rate to an acceptable level for most modern navigation applications. In order to solve this problem, most systems use external inputs from GPS to periodically update the INS solution and eliminate drift. GPS uses satellite signals to determine the position of the vehicle at a given moment in time; however, GPS is susceptible to jamming and may not be reliable in all scenarios. Current research therefore focuses on reducing the reliance on GPS or other external signals while maintaining acceptable navigation accuracy for operational missions.

Computer vision algorithms provide several approaches to this problem. The first approach uses feature recognition techniques to automatically recognize known landmarks in the navigation space and update the navigation solution based on the known location of these landmarks. This approach requires a database of known world features that can be recognized and matched by the on-board vision system. A second approach to this problem, and the one pursued in this project, does not require known features in the navigation space but only requires a known starting point and a sequential series of images. The sequential series of images can be used in a Structure from Motion algorithm so that the vision system can estimate trajectory independent of the INS. Both of these approaches can also be used in concert to develop a navigation system completely independent from both INS and GPS. In such a system, sequential images are matched together to form an INS-like trajectory solution while images matched to a database yield a GPS-like position update. In order to do this, it must be possible to accurately reconstruct an aircraft's trajectory and relate that trajectory to a world navigation frame while understanding the errors inherent in the process.

1.1.2 Reconnaissance. The military utility of geolocated aerial imagery of both man-made structures and natural terrain is obvious. Aerial images correlated with geographic information allow for effective intelligence, targeting and navigation operations. The next step in aerial reconnaissance technology is to create geolocated three dimensional reconstructions of targets and terrain from available imagery. This allows access to a three dimensional model of a target without the use of active ranging systems. Although this research effort primarily focuses on aerial navigation, the Structure from Motion process not only estimates the trajectory of the aircraft but simultaneously creates a three dimensional map of the location over which the aircraft flew. The method analyzed in this paper for transforming the reconstructed scene geometry to a real world coordinate system is also useful for geolocating targets in the scene and understanding the target location errors.

1.2 Problem Statement

The primary computer vision techniques used to support the goals of this research involve automatic feature matching and recognition, pose estimation and three dimensional scene reconstruction from multiple view geometry. These are fundamental computer vision techniques and they are currently implemented in many software packages [38] [10] [37] [20] [25] [13]; however their application to aerial navigation is relatively new and presents some unique challenges. The main problem is to determine how these recent advancements in computer vision can be leveraged to aid aerial navigation in realistic operational situations where traditional navigation techniques are limited.

1.3 Research Objectives

There are three primary objectives for this research. The first objective is to develop a prototype algorithm based on Structure from Motion that can successfully reconstruct the trajectory of an aircraft to determine the aircraft's current position using only a known starting point and images taken from a camera or cameras mounted

on the aircraft. The algorithm is designed in such a way that it can be realistically implemented on future systems. The second objective is to demonstrate the operation of this algorithm through both simulation and flight test with a variety of different cameras and flight profiles. The final objective is to characterize the error in the reconstructed trajectory and identify the dominant error sources using both simulation and flight test data.

II. Background

This chapter provides an overview of the relevant concepts in computer vision, navigation and mathematics necessary to develop the approach used in this research.

2.1 *Coordinate Frames*

There are six coordinate frames that are of interest in this research. The first frame is the Earth Centered Earth Fixed (ECEF) Frame [30]. This frame has origin at the center of the Earth and has a z-axis pointing through the North pole. The ECEF frame is fixed to the Earth and rotates with the Earth. The ECEF frame can be easily converted to a latitude, longitude and altitude on the Earth using the World Geodetic System (WGS)-84 Earth model, which is the model used in this research. The next frame is fixed to the aircraft center of gravity and is the aircraft body frame. This frame is defined with the x-axis pointing through the aircraft nose, the y-axis out the right wing and the z-axis down through the aircraft belly. It is fixed to the aircraft structure as the aircraft maneuvers. The aircraft North-East-Down (NED) frame shares the same origin as the aircraft body frame; however this frame remains in the same NED orientation as the aircraft maneuvers. The x-axis remains North, the y-axis points East and the z-axis points down (ie. perpendicular to local surface of the Earth). The relationship between the aircraft NED and aircraft body frame is described by Euler angle rotations using the order of ‘ZYX’ or ‘yaw, pitch, roll’ from NED to body. The aircraft body and NED frames are depicted below in Figure 2.1 and Figure 2.2 depicts the ECEF frame’s relationship to various body and local level frames [30] [32] [6] [35].

It is also useful to describe a local level navigation frame for the purposes of analyzing trajectories. This will be defined as an East-North-Up frame with the origin at the center of gravity of the aircraft at it’s first position in the trajectory. For example, at time zero the aircraft will be at the origin of the ENU local level frame and then will move away from the origin as time progresses and the trajectory is flown. See Figure 2.3 below showing the ENU frame for a given trajectory.

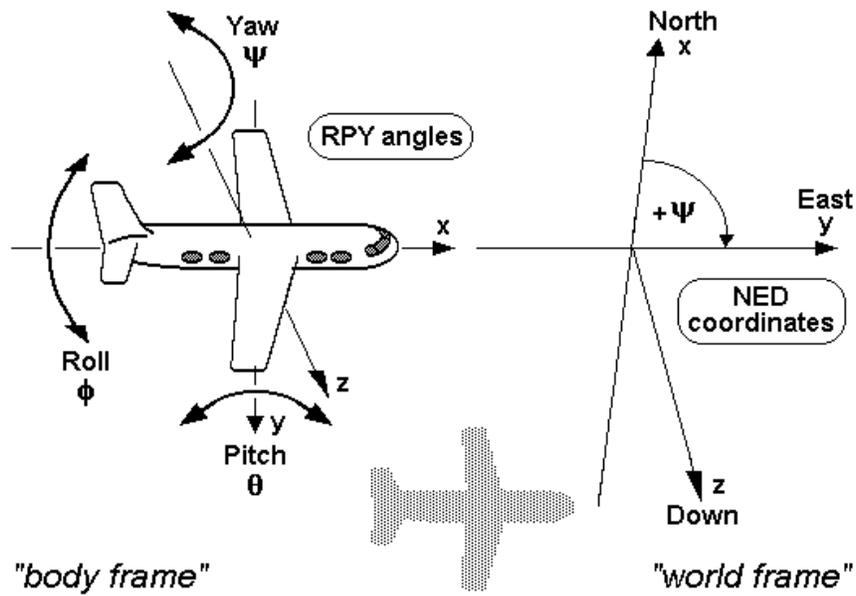


Figure 2.1: Aircraft Body Frame. The relationship between the aircraft body frame and the NED frame is shown. Figure adopted from [35].

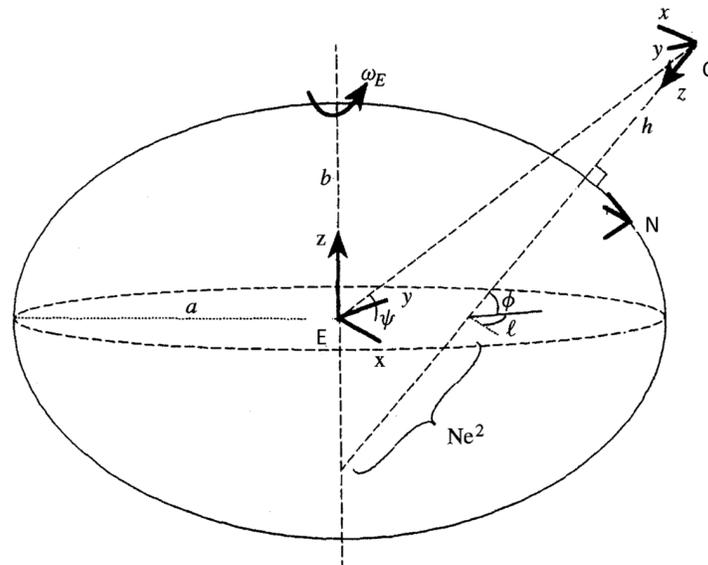


Figure 2.2: Earth Centered Earth Fixed Frame. The ECEF Frame is shown. Also shown is an aircraft body frame and a local level NED frame fixed to the surface of the Earth. Figure adopted from [35].

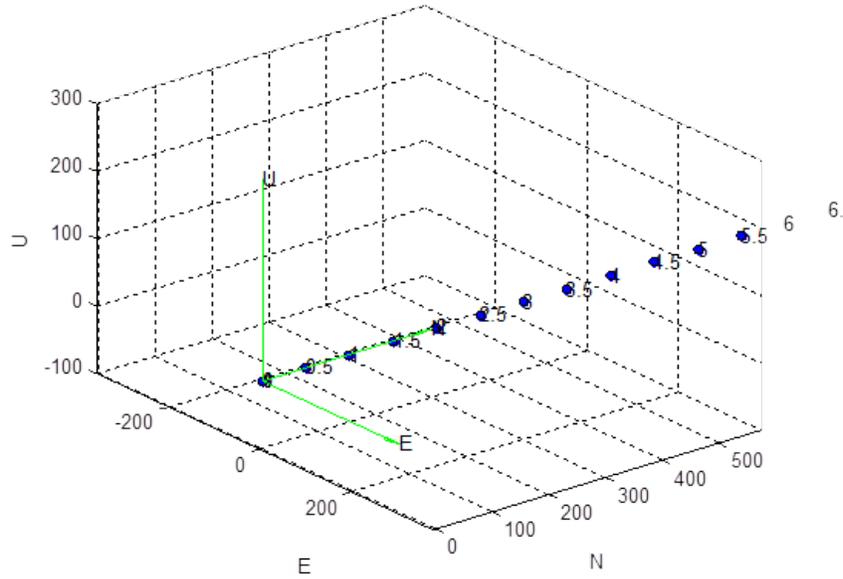


Figure 2.3: The local ENU frame with origin at the position of the aircraft at the initial time in the trajectory segment ($t=0$).

In this research a camera is fixed to the aircraft body. The camera frame has its origin at the optical center of the camera and z-axis parallel to the camera bore-sight pointing toward the scene. The x-axis of the camera frame points to the right when looking toward the scene and the y-axis points down. Note that this frame is the same camera frame used in the Visual Structure from Motion (VSFM) software package [37] and is shown in Figure 2.4.

The standard computer vision convention for a pixel coordinate frame is used. The pixel frame is a two dimensional frame with x and y axes parallel to the camera body x and y axis. The origin of the frame is the top left corner of the image as depicted below. Some of the derivations below describe an image coordinate frame. This is the same as the pixel frame; however, it has an origin that is in the center of the image plane instead of the top left corner. The x and y axes are the same direction. In Figure 2.4 the image coordinate frame location is denoted as (x,y) while the pixel coordinate frame location is denoted (u,v) [13].

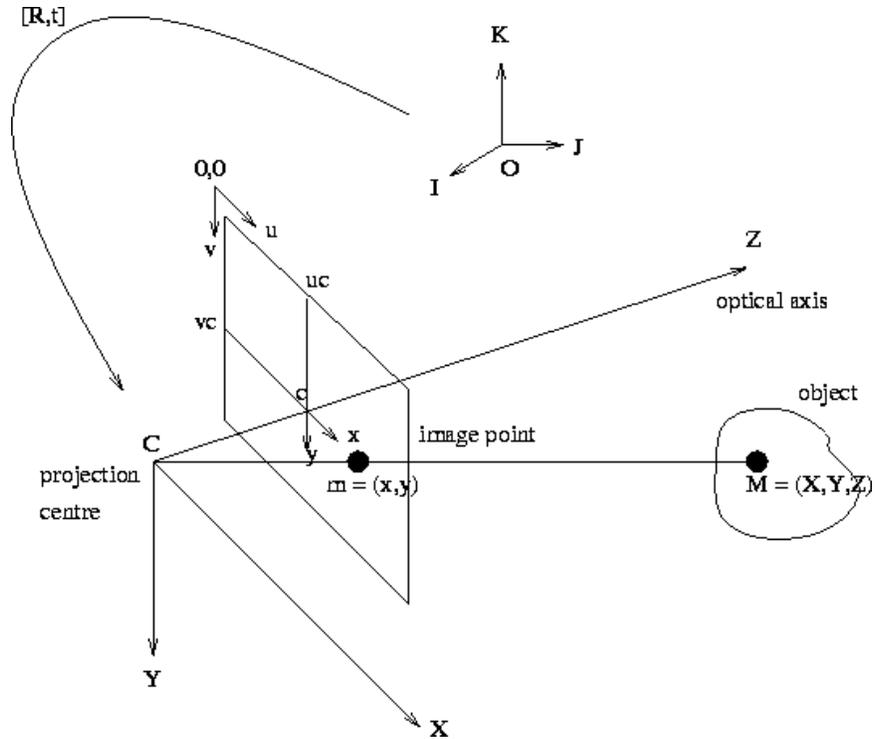


Figure 2.4: The camera frame as well as the corresponding image and pixel frames. Figure adopted from [27].

2.2 Camera Model

On the most basic level a camera is a device that focuses incoming light rays onto a photo-detector surface. The pattern formed by the photons that impact the photo-detector create an image. The image can be thought of as a two dimensional projection of the three dimensional scene towards which the camera is pointed. It is important to understand the relationship between an object in the three dimensional scene and its projection in the image. The basic mathematics behind an ideal pinhole camera are developed in this section.

As a convention for this paper, a superscript will denote the frame in which the vector is realized while a subscript will denote the type of vector (position of camera or position of target, etc). The superscript ‘e’ is the ECEF frame, ‘c’ is the camera frame, ‘a’ is the aircraft body frame, ‘n’ is the NED frame, ‘im’ is the image frame and ‘p’ is the pixel frame. Suppose the position of a target in the ECEF frame is

given as (note subscript ‘target’ denotes the target position vector and superscript ‘e’ denotes that ECEF frame):

$$x_{target}^e = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.1)$$

This location is easily converted to latitude, longitude and altitude of the target; however, for computation purposes we will leave the location in the above form. The location of the target in the camera frame can also be expressed as

$$x_{target}^c = C_a^c C_n^a x_{target}^n \quad (2.2)$$

where C_n^a is the direction cosine matrix (DCM) going from NED frame to aircraft body frame and C_a^c is the DCM going from aircraft body frame to camera frame. Since the NED and camera frames share an origin, only a rotation is required. In this example we assume that the camera frame also shares an origin with the aircraft frame. The DCM for NED to aircraft body is given below as

$$C_n^a = \begin{pmatrix} c(\Theta)c(\psi) & c(\Theta)s(\psi) & -s(\Theta) \\ s(\phi)s(\Theta)c(\psi) - c(\phi)s(\psi) & c(\phi)c(\psi) + s(\phi)s(\Theta)s(\psi) & s(\phi)\cos(\Theta) \\ s(\phi)s(\psi) + c(\phi)c(\psi)\sin(\Theta) & c(\phi)s(\psi)s(\Theta) - s(\phi)c(\psi) & c(\phi)c(\Theta) \end{pmatrix} \quad (2.3)$$

where ψ , Θ , ϕ are the yaw, pitch and roll of the camera relative to the NED frame (ie. Euler angles relative to local horizon) [30]. Unfortunately, we only know the position of the target in the ECEF frame. This is related to the position of the camera in the NED frame by the following rotation and translation:

$$x_{target}^n = C_e^n x_{target}^e + T^n \quad (2.4)$$

where T^n is the translation of the origin of the NED frame from the Earth's center to the camera center and where C_e^n is the direction cosine matrix rotating the ECEF frame to the NED frame. This DCM is a function of position on the Earth (lat, long) and is given as:

$$C_e^m = \begin{pmatrix} -\sin(\lambda)\cos(L) & -\sin(\lambda)\sin(L) & \cos(\lambda) \\ -\sin(L) & \cos(L) & 0 \\ -\cos(\lambda)\cos(L) & -\cos(\lambda)\sin(L) & -\sin(\lambda) \end{pmatrix} \quad (2.5)$$

where L is camera longitude and λ is camera latitude [30]. We can now relate the position of the target and the position of the camera origin by vector addition in the camera frame. If we know the position of the camera's center in the ECEF frame (from GPS or other navigation aid) we can rewrite Equation 2.4 as:

$$x_{target}^n = C_e^m x_{target}^e + C_e^m T^e \quad (2.6)$$

Finally, we multiply the above equation by the DCMs from NED to camera frame to obtain the relationship between the camera location, target location and camera to target vector expressed in the camera frame

$$C_n^c = C_a^c C_n^a \quad (2.7)$$

$$C_n^c x_{target}^n = x_{target}^c = C_n^c C_e^m x_{target}^e + C_n^c C_e^m x_{camera}^e \quad (2.8)$$

Since we know the location of the camera and the target in the ECEF frame we therefore have everything we need to calculate the vector between the camera center and the target x_{target}^c .

Instead of representing the above transformation as a rotation and translation in 3-D space it will be convenient to represent the above transformation as a single transformation in 4-D space. This can be done using homogeneous coordinates. For a more detailed discussion of homogeneous coordinates please see [23] [14] [15] but

for the purposes of this paper we know that the homogeneous coordinate for a vector is obtained by adding a 4th row to the vector with value of 1. We can re-write the above expression as:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{target}^c = \begin{pmatrix} C_n^c C_e^n & C_n^c C_e^n x_{camera}^e \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{target}^e \quad (2.9)$$

This expression is helpful and will give us a general idea of the target location since we can directly relate the camera and target position. We would like to go one step further and relate the target position to an individual pixel on the image plane. In order to do this, we need to make some assumptions about the camera. We will first say that we are using a perfect pinhole camera with no lens distortion. Although this is not generally a good assumption, there are known models to correct for lens distortion that can be easily incorporated. These distortion models will be discussed at the end of this section. In this case, we know from basic optical theory [23] that the 2-D projection of the target point on the camera image plane is given as:

$$\begin{pmatrix} x \\ y \end{pmatrix}_{target}^{im} = f/Z \begin{pmatrix} X \\ Y \end{pmatrix}_{target}^c \quad (2.10)$$

where f is the camera focal length, X, Y, Z are the components of the x_{target}^c vector derived in Equation 2.9. The superscript ‘im’ denotes the image plane frame as outlined earlier. A visual depiction of the above equation is shown in Figure 2.5.

As before, we can re-write this relationship in homogeneous coordinates as:

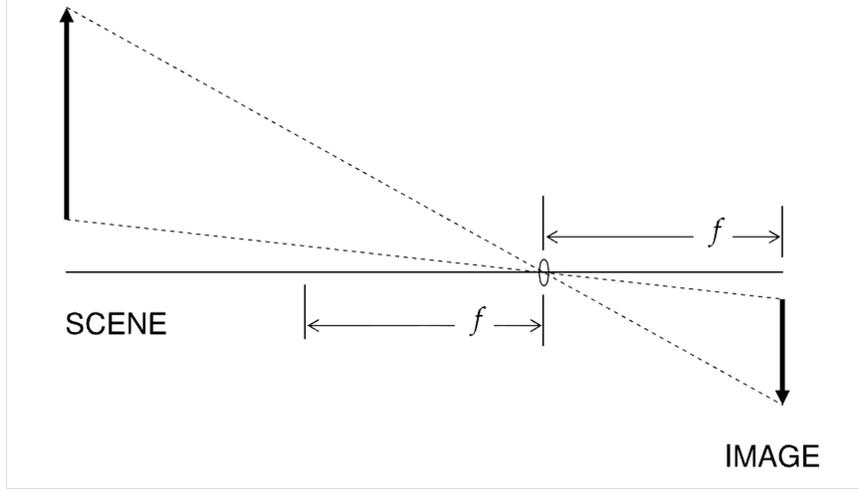


Figure 2.5: Geometry of an ideal pinhole camera. Figure adopted from [32].

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}_{target}^{im} = 1/Z \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{target}^c \quad (2.11)$$

The values of x_{target}^{im} and y_{target}^{im} found above are not yet in pixels. These are the coordinates of the target projection on the image plane. In order to find these values in pixel space we must first transform these values from distance units to pixel units and change the origin of the values since pixels are referenced from the top left corner of the image plane instead of the center point of the image plane. These two transformations are encompassed in the following equation:

$$\begin{pmatrix} x \\ y \end{pmatrix}_{target}^p = \begin{pmatrix} s_x & s_\Theta \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_{target}^{im} + \begin{pmatrix} o_x \\ o_y \end{pmatrix}^p \quad (2.12)$$

where s_x, s_y represent the scale of the pixel (ie. physical length of pixels in the x and y directions), s_Θ is the skew of the pixel (in case it is not a perfect square) and o_x, o_y are the pixel coordinates of the center of the image plane. The superscript ‘p’

indicates the pixel frame. As before, we can write the above equation in homogeneous coordinates as:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^p = \begin{pmatrix} s_x & s_\Theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^{im} \quad (2.13)$$

The three transformations outlined above are a transformation of the camera to target vector from world coordinates to camera coordinates (ECEF to camera frame), a projection of the target point from camera frame to the 2-D image plane and a transformation from image plane to pixel coordinates. These three transformations can be combined with homogeneous matrix multiplication to form the following equation:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}_{target}^p = \begin{pmatrix} s_x & s_\Theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} 1/Z \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{target}^e \quad (2.14)$$

$$\begin{pmatrix} C_n^c C_e^n & C_n^c C_e^n x_{camera}^e \\ 0 & 1 \end{pmatrix}$$

It will be useful to re-arrange this equation to the following form with matrix math [23]:

$$\begin{aligned}
\begin{pmatrix} x \\ y \end{pmatrix}_{target}^p &= \begin{pmatrix} fs_x & fs_\Theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{pmatrix} 1/Z \\
&\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} C_n^c C_e^n & C_n^c C_e^n x_{camera}^e \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}_{target}^e \quad (2.15)
\end{aligned}$$

Using the symbols K , Π and G respectively for the matrices above we re-write the equation as [23]:

$$x_{target}^p = \frac{1}{Z} K \Pi G x_{target}^e \quad (2.16)$$

Equation 2.16 represents the overall transformation from target coordinates to pixel values. The matrices in this equation contain the information about the camera position, camera orientation and camera internal parameters (focal length, pixel size, optical center and pixel skew).

The above analysis assumes a camera with a perfect lens. In the real world, camera lenses have distortions which cause the pixel location of a given feature to differ from the pixel location predicted by the pinhole camera model. The most common method for accounting for these distortions defines two types of lens distortions: radial and tangential. These distortions can be determined empirically for a given camera/lens. Empirical determination of the lens distortions yields a 5x1 vector of distortion coefficients. The first and second coefficients represent radial distortion which governs how lens distortions change depending on the radial distance from the center of the image plane. The third and fourth terms are tangential distortion and these terms describe lens distortions in a direction perpendicular to the radial distortion. The fifth term accounts for pixel skew and will be considered zero for this analysis. The following equations are used to correct the pinhole camera model [32] [21] [7]:

$$r = \sqrt{x^2 + y^2} \quad (2.17)$$

$$x_{undistorted}^{im} = (1 + k(1)r^2 + k(2)r^4 + k(5)r^6)x_{distorted}^{im} + \begin{pmatrix} 2k(3)xy + k(4)(r^2 + 2x^2) \\ k(3)(r^2 + 2y^2) + 2k(4)xy \end{pmatrix} \quad (2.18)$$

where x and y are components of $x_{distorted}^{im}$ and are the locations of the feature in the image plane in pixels (i.e. referenced to the center of the image plane not the top-left corner) and where k is the 5x1 distortion coefficient vector.

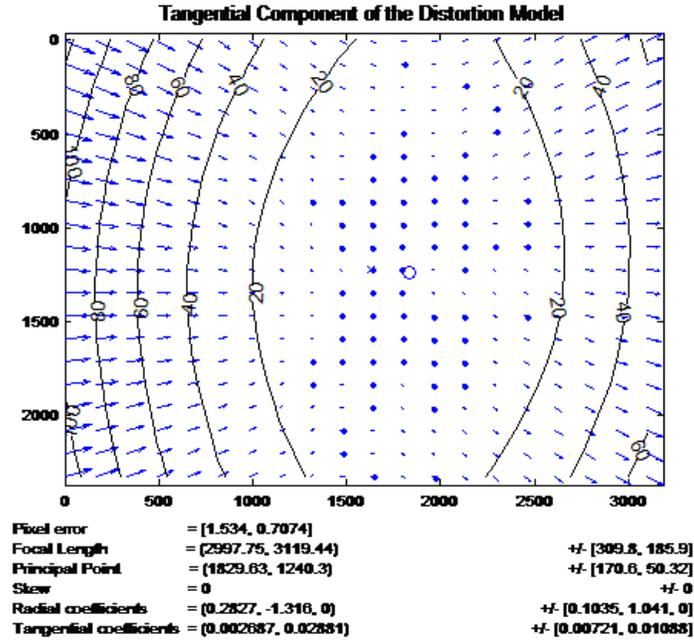


Figure 2.6: Sample map of tangential distortions on an image plane. [7]

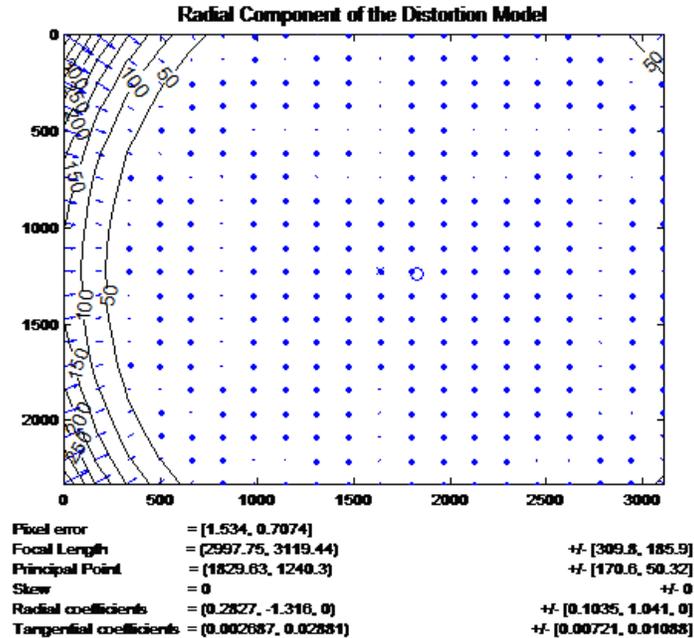


Figure 2.7: Sample map of radial distortions on an image plane. [7]

2.3 Epipolar Geometry

The previous section developed the basic equations relating a feature in a world coordinate frame to that feature's projection in an image. This was only possible with knowledge of the depth between the camera and the feature. This depth is equivalent to the length of the vector that starts at the camera center, passes through the image plane and ends at the target feature. In the case of navigation, this depth is uncertain since one may not know the exact location of the camera or the features. Fortunately, it is possible to determine depth by using multiple images and Epipolar geometry. Consider Figure 2.8 where two cameras image a scene with a common feature point.

Assuming that the same feature can be recognized in each image then it is possible to use geometry to determine a relationship between the two images. We know from basic vector addition that the locations of the feature in the camera 1 and the camera 2 frames are related as follows:

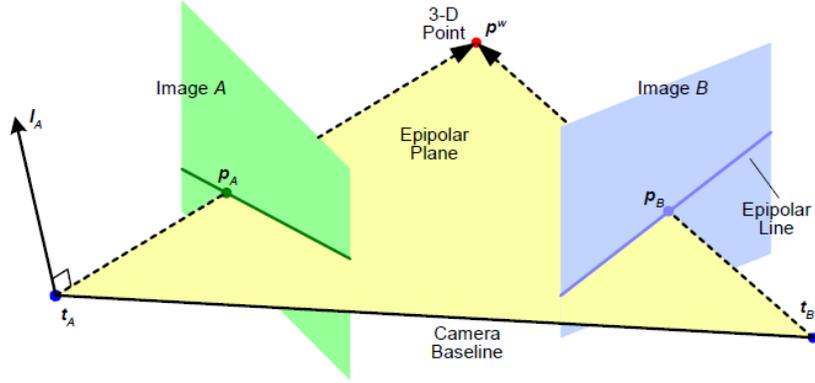


Figure 2.8: When two cameras view the same scene, the cameras and the 3-D points they view can all be related with Epipolar Geometry. Figure adopted from [15].

$$x_{feature}^{c2} = R_{c1}^{c2} x^{c1} + T^{c2} \quad (2.19)$$

where R_{c1}^{c2} is the rotation from camera 1 to camera 2 and T^{c2} is the translation between camera 1 and camera 2 expressed in the camera 2 frame. The locations of the feature in the camera frame can be then be replaced by the location of the feature in the image frame multiplied by an unknown depth, λ , if it is assumed that the cameras have perfect calibration so that $K=I$ [23] [15]. This yields:

$$\lambda_2 x^{im2} = R_{c1}^{c2} \lambda_2 x^{im1} + T^{c2} \quad (2.20)$$

Multiplying both sides of the equation by the skew symmetric form of T which is \hat{T} gives:

$$\lambda_2 \hat{T} x^{im2} = \hat{T} R_{c1}^{c2} \lambda_2 x^{im1} \quad (2.21)$$

This equation can be pre-multiplied by $(x^{im2})^T$ and taking advantage of the fact that $(x^{im2})^T \hat{T} x^{im2}$ is zero yields the epipolar constraint:

$$(x^{im2})^T \hat{T} R x^{im1} = 0 \quad (2.22)$$

where the quantity $\hat{T}R$ is defined as the essential matrix. Note that this matrix contains information about the relative rotation and translation between the two cameras and that the matrix can be found solely using corresponding feature points between two images. Once the essential matrix is found from feature matches then it can be decomposed to determine the relative rotation and translation between the two cameras. An eight point algorithm to decompose the essential matrix is proposed in [15].

Even though the derivation of the essential matrix does not consider camera calibration information (focal length and pixel size), this case can be easily generalize to account for the effects of camera calibration parameters. When the camera calibration matrix is given as K , then it can be shown that the epipolar constraint now becomes [23] [15]:

$$(x^{p2})^T \hat{T}' K R K^{-1} x^{p1} = 0 \quad (2.23)$$

where $\hat{T}' K R K^{-1}$ is defined as the fundamental matrix and contains information about relative translation, rotation as well as camera calibration. As with the essential matrix, the fundamental matrix can be calculated using corresponding features between two images and can also be decomposed with a similar eight point algorithm to determine relative translation, rotation and camera calibration [23] [15] [12] [34].

2.4 Structure from Motion Overview

The previous sections described the basic mathematics relating an image taken from one or more cameras to the three dimensional structure of a scene. This math forms the theoretical basis for a set of algorithms called Structure from Motion (SFM) in which corresponding features in multiple images of the same scene are used to construct a three dimensional representation of the scene as well as to estimate camera

pose and even calibration. There are many different implementations and variations of the Structure from Motion algorithm; however, this section will introduce the generic procedure used by SFM given a collection of images. The four main steps to any SFM process are as follows [38] [10] [25] [31]:

- Determine feature correspondences between images.
- Determine fundamental matrices for each pair of images.
- Use the fundamental matrices to determine an initial sparse 3-D structure.
- Minimize the re-projection errors in the initial sparse structure to create an accurate but relative 3-D reconstruction.

An optional fifth step exists to transform the sparse structure into a dense structure of 3-D points. This is an important step when building accurate 3-D models of an imaged scene for qualitative analysis; however, for the purposes of navigation the main interest is the recovery of camera pose which is not significantly affected by a dense reconstruction of 3-D points. Therefore, this fifth step will not be discussed.

2.4.1 Image Correspondences. The first challenge in implementing SFM is to identify the same features in multiple images. This is easy for a human observer but programming a computer to automatically recognize certain image features is a difficult task that is the subject of significant computer vision research. There are two main approaches to image correspondence. The first approach is an area based approach where the pixel patterns of two entire images are compared against each other. Areas of pixel patterns with high correlation are matched against each other and the images can be aligned so that pixels in one image are mapped to pixels in another image. In this method, a smaller image is convolved across a larger template image. Peaks in the correlation result represent potential matches. Figure 2.9 shows the correlation of a picture of the Ohio University football stadium taken from an aircraft, with a larger satellite image of the Athens, Ohio area. The resulting correlation result shows a distinct peak where the images match. This method can

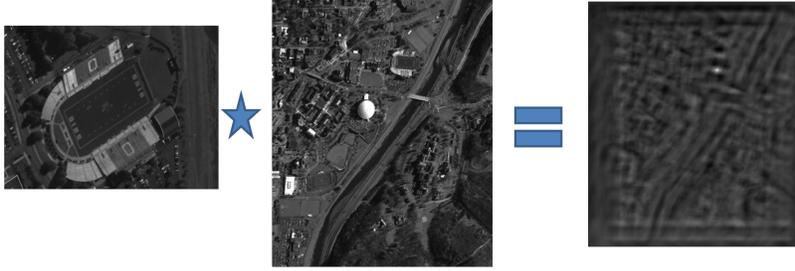


Figure 2.9: Aircraft image correlated with satellite image. The images must be rotated in the same direction and images must be of equal scale. Note the white peak in the resulting image denoting the region of maximum correlation (ie. the location where the images match).

be extremely effective; however, precise matching of distinct individual features in the images is difficult. Additionally, scale, rotation and perspective changes between images can greatly affect the correlation process resulting in false or no matches.

The second approach to image correspondence is a feature based method. In this approach the first step is to identify the locations of distinct features in an image and then describe those features in a unique way so that they can be matched to the same feature descriptions that appear in other images. Research has shown that distinct features in an image tend to be the result of corners or other sharp gradients in pixel values. The general approach is then to look for and describe areas that have these distinct gradients. There are several methods to do this but the most established and best performing method is Scale Invariant Feature Transform (SIFT) developed by David Lowe [22]. SIFT is a powerful feature detection and matching tool because it is invariant to changes in scale and rotation and can also handle some perspective changes. SIFT is the primary feature detecting tool used in this research. The first step in SIFT matching is to convolve each image with a Difference of Gaussian (DOG) filter. This filtering process is known to produce responses along edges and corners of an image. A difference of Gaussian filter is a filter comprised of two Gaussian filters with different variance (σ^2) values subtracted from one another. This filter is illustrated graphically in Figure 2.10.

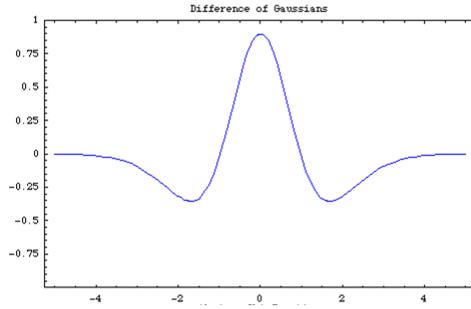


Figure 2.10: The difference of Gaussian filter is created by subtracting two Gaussian filters with different variances. When convolved with an input image this filter highlights edges in an image. [22] [13]

The key to SIFT’s scale invariance lies in the fact that this DOG filter is not just applied to the original image but it is applied to several down sampled versions of the original image. In other words, multiple low pass filters are applied to the original image so that edges can be detected in different levels of scale space. SIFT then looks for features that have a strong DOG response over different scale spaces and concludes that such features must be relatively invariant to changes in scale.

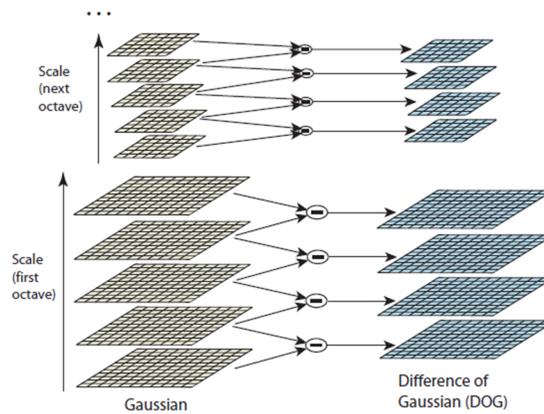


Figure 2.11: The DOG filter is applied to various down sampled versions of the original image so as to look for features that have responses throughout a variety of image scales. The number of octaves determines how many down sampled versions of the original image are used. Figure adopted from [22].

The next step is to localize and describe the areas with strong DOG responses throughout each scale space. This is done by calculating the pixel value gradients in the vicinity of the potential feature. Using this method, SIFT thresholds and localizes the strongest and most distinct gradients. Once weak gradients are eliminated, remaining gradients are analyzed to determine a feature magnitude, orientation and descriptor vector. The magnitude and orientation of the feature is calculated as:

$$m(x, y) = \sqrt{(I(x + 1, y) - I(x - 1, y))^2 + (I(x, y + 1) - I(x, y - 1))^2} \quad (2.24)$$

$$\Theta(x, y) = \arctan (I(x, y + 1) - I(x, y - 1)) / (I(x + 1, y) - I(x - 1, y)) \quad (2.25)$$

Assigning this orientation as the reference for a given feature allows for rotation invariance. In other words, the feature descriptor will be referenced to this orientation regardless of the rotation in the image. Finally, a 128 dimension descriptor vector is calculated for each feature. The descriptor vector is a unique description of the pixel gradients around the feature point. In theory, this 128-d vector should always show up with its particular feature gradient regardless of changes in lighting, scale and rotation.

The output of running SIFT on a group of images is then a set of 128-d descriptor vectors as well as x-y pixel locations for these vectors in each image. The next challenge is to effectively search through the set of vectors from each image and match those that are the same. This is done by comparing the Euclidean distance between a descriptor vector and its potential match. Euclidean distance between two vectors is calculated with a dot product. The higher the value of the dot product then the more similar the vectors. For example, the dot product of two identical vectors is 1. Once dot products between every combination of vectors are calculated, it is possible to analyze the dot product results and declare a potential match. There are several methods to analyze dot products. The simplest but least accurate is to pick

a non-adaptive threshold value. The most commonly used technique and one that generally produces a low rate of false matches is the 2nd-nearest neighbor method. In this method, a given feature is combined via dot product with every other feature that is a potential match. The magnitude of the dot products are ordered from highest to lowest and the ratio of the highest dot product to the second highest dot product is taken. The higher the ratio the better the match. A high ratio essentially means that the two features are most like each other and not like other features in the set. In general the threshold ratio is set between .6 and .8 but can vary and can be experimentally determined for a given image set. The result of this process produces a set of corresponding features.

The matching process is not perfect and can result in some outlier matches. These outliers can be eliminated by using geometric constraints and random sample consensus (RANSAC) [15] [34]. As discussed earlier, any two images will be related via a fundamental matrix that maps features in one image to features in another image. The matches generated by SIFT matching must be consistent with some common fundamental matrix. Any matches that are not consistent with a common fundamental matrix must be outliers. The concept of random sampling is used to determine the common fundamental matrix by randomly selecting at least 8 matching features [15]. From these features a fundamental matrix is calculated. This fundamental matrix then is used to project all the features in one image to features in the other image. The projections are compared to the actual feature locations and if the majority line up within a certain threshold then this must be a good fundamental matrix. This process is repeated multiple times with several random samples until a fundamental matrix is found that minimizes the total re-projection error. Any feature matches that don't align with this final fundamental matrix are considered outliers and disregarded.

In addition to using RANSAC with a fundamental matrix constraint, the same process can be used but with a simpler matrix constraint called a homography constraint. Unlike a fundamental matrix which relates cameras that are viewing a 3-D scene through epipolar geometry, a homography matrix (H) is a simple planar trans-

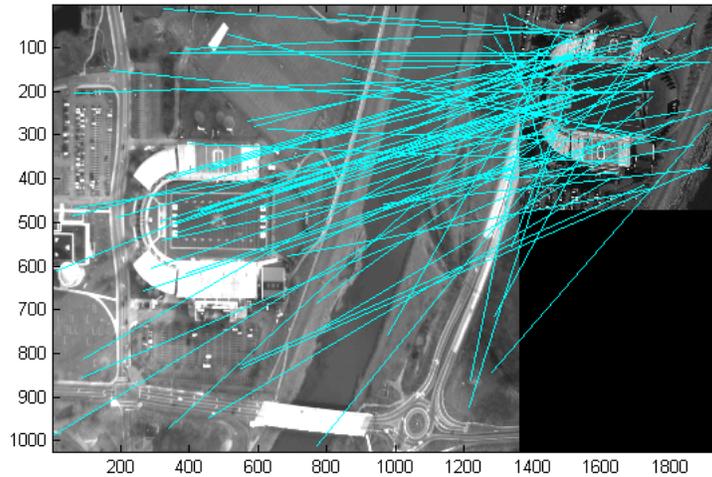


Figure 2.12: The results of nearest neighbor SIFT matching without applying geometry constraints to reject outliers. Note the large number of false matches.

formation between pixels in two images. In other words, the following constraint must be satisfied between two 2-D images:

$$x^{im2} = Hx^{im1} \tag{2.26}$$

where H is a 3×3 matrix and x^{im1} and x^{im2} are $3 \times n$ matrices where each column is an x,y feature location in homogeneous coordinates and n is the number of features. Solving for H is a linear, least squares solution to the above equation. The homography constraint and the fundamental matrix constraint can be used in conjunction to eliminate as many outliers as possible. Other logical constraints can also be applied to further improve outlier rejection (ie. one feature can only match to one other feature, etc). Figures 2.12 and 2.13 show matching between two very dissimilar images (ie. different cameras, scales, perspectives, times of year, etc) with and without using geometry constraints. In this case there are initially a lot of false matches; however, these are effectively removed using geometric and logical constraints.

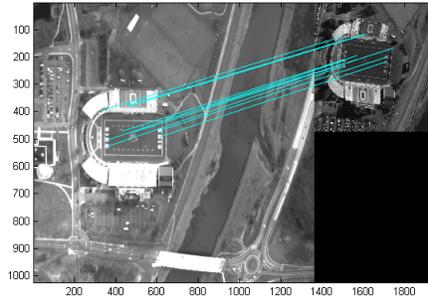


Figure 2.13: The results of nearest neighbor SIFT matching after rejecting outliers by using fundamental and homography matrix constraints.

Even though dissimilar images may have a lot of outliers and end up with only a few good SIFT matches, similar images taken from the same camera often have thousands of good matches. Figure 2.14 shows SIFT matching between two sequential images taken from the same camera of the same target. This is the type of result expected when attempting to match images in a sequence taken from an airborne camera.

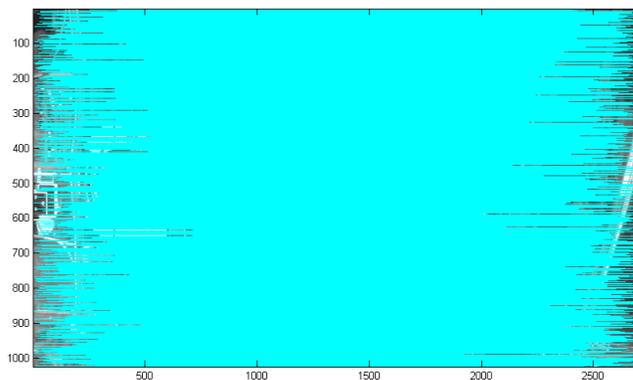


Figure 2.14: SIFT matching between two images taken within a few seconds of each other and from the same camera mounted on an aircraft. The thousands of matches in this case allow for robust estimation of the fundamental matrix even with noise present.

2.4.2 Determining initial sparse 3-D structure. The second step of the SFM process is to calculate a fundamental matrix for each image pair. This was likely already done as part of the outlier rejection step in SIFT matching. The fundamental matrices for each image pair can then be used to calculate camera projection matrices and the relative positions of each camera. Using this information, the 3-D locations of the matching features can be triangulated to form an initial 3-D structure that is self consistent. Note that even though the 3-D structure found from triangulation is consistent with itself, it is not and cannot be tied to any real world reference system without some other knowledge about the location of features or cameras in the real world.

Generating an initial 3-D structure relies on the fact that certain 3-D features can be tracked through several images. The SFM algorithm picks a first initialization pair of images as the pair of images with the most matches. This initial pair is then used to generate a set of 3-D features. From this initial pair the next image in the set is added based off which image sees the most number of 3-D points generated by the first two images. Images are then added one at a time in this same manner to construct the initial 3-D structure and camera pose estimates [38] [25] [4] [11] [16]. Note that this process is heavily dependent on an accurate guess of focal length for each camera and enough angular separation between cameras for triangulation. An inaccurate guess of focal length may lead to a highly inaccurate initial 3-D structure and the inability to generate tracks which will lead to reconstruction failure.

2.4.3 Bundle Adjustment. Once an initial estimate of 3-D structure and camera pose is available, the final step is to run a Sparse Bundle Adjustment (SBA) on the model to further refine the 3-D reconstruction. The basic concept of SBA is to minimize the pixel re-projection error between the measured feature pixel locations and the predicted feature pixel locations given the estimated 3-D point positions and estimated camera poses. This minimization cost function is defined as [23]:

$$\sum_{i=1}^n \sum_{j=1}^m w_{ij} \|q_{ij} - P(K_i, G_i, X_j)\| \quad (2.27)$$

for n 3-D points, m cameras and where w_{ij} is a weighting function that is 1 if a point is viewed in a given camera or zero if the point is not viewed in a given camera. The symbol q_{ij} is the actual projection of the i th point in the j th camera and P is the predicted projection of the i th point in the j th camera given the camera intrinsics (K), camera pose (G) and 3-D point location (X) [23] [15].

The camera intrinsics, camera poses and the 3-D points that minimize this cost function represent the final solution for the 3-D structure. This minimization problem is most commonly solved with the Levenberg-Marquardt non-linear least squares estimation algorithm [23]. The problem is simplified by the sparse nature of the Jacobian matrices involved. The solution method uses two Jacobian matrices, A and B , where A describes the change in reprojection error due to changes in camera location, orientation and intrinsic parameters while B describes the change in reprojection error due to changes in 3-D point location. The general structure of the two Jacobian matrices is shown in Figure 2.15.

Note that since reprojection errors are only affected by the cameras and points involved (and not other cameras in the bundle) then both Jacobian matrices have a block diagonal sparse structure. In other words, changes in the reprojection error of a given 3-D point are only caused by changes in the location of that point and changes to the camera that is viewing that point. Note that the Bundle Adjustment on a given 3-D structure can be constrained in various ways by changing elements of the Jacobian matrices. For example, if the intrinsic camera parameters are known from a previous calibration then the elements of the Jacobian matrix corresponding to changes in camera parameters can be set to zero. This will result in a solution that keeps the initial camera parameters and adjusts the optimal structure to keep these constraints. Constraining the bundle adjustment with various known parameters by zeroing the appropriate terms in the Jacobian matrix was an important part of

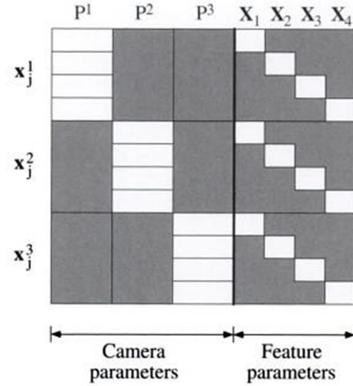


Figure 2.15: This is the Jacobian structure for a system with three cameras and four 3-D points. The A and B Jacobian matrices have been combined into one matrix. The un-shaded regions represent non-zero elements of the Jacobian. The pixel re-projection error is x , the pose of each camera, p , is made of roll, pitch, yaw, 3-D location as well as camera intrinsics (if K is not known) and X is the 3-D point location. Figure adopted from [15].

generating navigation solutions for this research and the specific effects of various constraints will be discussed in Chapter 4.

The Levenberg-Marquardt technique for minimization has the potential to converge on local minima and therefore success of the technique is highly dependent on the initial parameters. As a result, the initial 3-D structure is used as an input to the SBA process but if this structure is significantly inaccurate then SBA will likely fail to converge. Bundle Adjustment is often run iteratively during the SFM process and is therefore interwoven with the previous step above. Every time a new camera is added to the structure then a Bundle Adjustment can be run before another camera is added. A final Bundle Adjustment is run as the last step to any reconstruction [23] [10] [31].

III. Methodology

The chapter provides an overview of the SFM based algorithm developed to reconstruct an aircraft's flight trajectory using images captured from a camera or cameras mounted on the aircraft. The algorithm was developed in a way that is useful for and falls within the limitations of a system designed for aerial navigation. Additionally, this chapter will describe the methods used to implement the process and will assess the major sources of error predicted by theory.

3.1 *Algorithm Inputs*

The minimum inputs required for the algorithm are as follows:

1. Sequential series of images taken from the airborne platform.
2. Known real world camera locations for two of the images.
3. Known real world camera orientation for one of the images.
4. Known transformation from camera frame to aircraft body frame.

For a navigation application, it is appropriate to assume that the camera location and orientation is known for the first image in the sequence (ie. known starting point) and that the location of the same camera when taking the another image is also known. The camera location for the second image relative to the first image can be obtained through separate means (ie. GPS, INS, dead reckoning, stereo camera etc).

The transformation between the the camera frame and the aircraft frame can either be fixed for every image in the trajectory or it can change with each image. In other words, the camera can be rigidly mounted to the aircraft body or it can slew relative to the aircraft frame. In either case, it is possible and realistic to measure the transformation between frames using hardware methods. For simplicity, the simulations and tests in this research utilized designs with a fixed transformation between the aircraft frame and camera frame.

Even though this algorithm only requires the above four inputs, if available, it is useful to record a real world camera orientation and altitude for every image taken

in the sequence. Utilizing this added information is, in fact, realistic since most air vehicles are equipped with an Inertial Measurement Unit (IMU) to measure camera orientation and an altimeter to measure altitude. Intrinsic camera calibration parameters (measured from previous camera calibration) are also useful to the algorithm. There are several techniques for camera calibration that can be utilized prior to and during flight to accurately determine camera intrinsic parameters.

3.2 Algorithm Overview

The algorithm developed in this research uses the following steps:

1. Collect sequential images from aircraft mounted camera system.
2. Apply SFM to sequential images.
3. Transform resulting 3-D reconstruction to real world coordinate system.
4. Apply Sparse Bundle Adjustment to the model in the world coordinate system using available constraints.
5. Use calculated trajectory for navigation.

3.2.1 Collect images. This research used both real images collected from flight test as well as simulated images. This section describes the various image data sets used and how they were incorporated.

3.2.1.1 Simulated Images. In order to effectively analyze the errors associated with this approach it was necessary to develop a simulation in which all parameters could be completely controlled to determine their effect on navigation accuracy. This was done by simulating flight of an aircraft mounted camera over a set of simulated feature points on the Earth's surface. The software package ProfGen was developed by the Air Force Research Lab (AFRL) in order to generate various aircraft trajectories in a world coordinate system given a starting point and various aircraft maneuvers. This software was used to generate aircraft flight profiles of in-

terest. These profiles were then input into the SIFT feature simulator developed by the AFIT Autonomy and Navigation Technology (ANT) center. The SIFT feature simulator uses the flight profile, camera calibration information and camera mounting parameters to randomly generate features for the camera to see as it flies along the given trajectory. These features are randomly distributed along the Earth’s surface as modeled by a Digital Terrain Elevation Data (DTED) map uploaded from National Geospatial-Intelligence Agency (NGA) data at the location of the flight trajectory. The simulator uses the camera projection equations developed in Chapter 2 to calculate the location of each feature in every image taken along the trajectory. Each feature is given a unique identifier so it is possible to tell which features match between each simulated image. Overall, the simulator outputs a set of feature locations (x,y pixel locations) in each image and the unique identifier associated with each simulated feature.

3.2.1.2 ASPN Images. The Air Force Research Lab Sensors Directorate (AFRL/RV) provided imagery data from the All Source Positioning and Navigation (ASPN) program flight test. In this program a camera was mounted to a DC-3 and flown over Athens, Ohio. An on-board Novatel Synchronous Position, Attitude and Navigation (SPAN) INS/GPS system provided truth navigation data for every image frame. The IMU used with the system was a Novatel SPAN HG1700-58. The camera used in the ASPN test was rigidly mounted to the bottom of the aircraft fuselage and was pointed straight down toward the ground. The camera parameters for the ASPN flight test are detailed in Table 3.1.

Fx	1346.08 pixels
Fy	1340.66 pixels
Cx	720.12 pixels
Cy	500.72 pixels
Radial	[-0.243301683932734, 0.307959145479999, -0.000863739790749, 0.001362991052415, 0.0]
Image Size	1360 x 1024 pixels

Table 3.1: ASPN Camera Parameters

The transformation matrix from the ASPN reference camera frame to the IMU/aircraft body frame was:

$$C_r^a = \begin{pmatrix} 0.012077 & -0.996814 & -0.078842 \\ 0.999923 & 0.011805 & 0.003917 \\ -0.002974 & -0.078883 & 0.996879 \end{pmatrix} \quad (3.1)$$

3.2.1.3 Angel Fire Images. Angel Fire was an operational flight program developed by AFRL for wide area, high resolution surveillance capability. The Angel Fire camera system was mounted on a manned aircraft but unlike the ASPN system the camera was gimballed and could actively move with respect to the aircraft frame in order to track a target area. In general the camera looked out the side of the aircraft and downward. An on-board Novatel Synchronous Position, Attitude and Navigation (SPAN) INS/GPS system provided truth navigation data for every image frame. The IMU used with the system was a Novatel SPAN HG1700-58. The camera parameters for the Angel Fire test are listed below in Table 3.2. Data from an Angel Fire test mission over Athens, Ohio was used for this analysis.

Fx	14086.874 pixels
Fy	14086.874 pixels
Cx	2436 pixels
Cy	1624 pixels
Radial	[0.0, 0.0, 0.0, 0.0, 0.0]
Image Size	4872 x 3248 pixels

Table 3.2: Angel Fire Camera Parameters

The transformation matrix from the Angel Fire reference camera frame to the IMU frame was:

$$C_r^a = \begin{pmatrix} -0.00063662724 & 0.999997642 & 0.00207616071 \\ -0.986126585 & -0.000283163805 & -0.165994813 \\ -0.165993834 & -0.00215303409 & 0.98612444 \end{pmatrix} \quad (3.2)$$

3.2.1.4 *C-12 Images.* Several sets of image data were obtained from two different flight tests conducted by the USAF Test Pilot School (TPS) using a camera mounted on a C-12 aircraft. These tests were flown as part of two different Test Management Projects (TMP): Have SURF and Have Shuttermatch. In these tests, data was collected flying over various types of terrain and on different trajectories at Edwards Air Force Base (AFB), California. The camera was rigidly mounted to the aircraft and an on-board INS/GPS system provided truth navigation data for analysis. The camera mounting and calibration parameters for these tests were not available, so only a qualitative analysis was performed on this data as an aid to developing the SFM algorithm.

3.2.1.5 *Minor Area Motion Imagery (MAMI).* The AFRL sponsored Minor Area Motion Imagery (MAMI) project consisted of two phases: MAMI-I and MAMI-II. In MAMI-I a high resolution, wide area camera array was mounted on a NASA Twin Otter aircraft and flown over Wright Patterson AFB, Ohio. As with Angel Fire, the camera system looked out the side of the aircraft and downward and was gimbaled so that it could move with respect to the aircraft in order to track a target area. An on-board Novatel Synchronous Position, Attitude and Navigation (SPAN) INS/GPS system provided truth navigation data for every image frame. The IMU used with the system was a Novatel SPAN HG1700-58. The camera parameters for the MAMI-I test are listed in Table 3.3.

Fx	7418.37 pixels
Fy	7418.37 pixels
Cx	1024 pixels
Cy	1024 pixels
Radial	[0.0, 0.0, 0.0, 0.0, 0.0]
Image Size	2048 x 2048 pixels

Table 3.3: MAMI-I Camera Parameters

The transformation matrix from the MAMI-I reference camera frame to the IMU frame was:

$$C_r^a = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (3.3)$$

The data from MAMI-II was also used to support this research. MAMI-II was a radically different system than any of those previously discussed. The goal of MAMI-II was to collect imagery for research into image navigation and target three dimensional reconstruction from a high performance aircraft at higher altitudes, airspeeds and longer ranges than was possible with previous programs. To accomplish this goal, the MAMI-II system was designed to be self contained within a USAF Test Pilot School (TPS) Reconfigurable Airborne Sensor, Communications and Laser (RASCAL) Pod. The RASCAL pod was built by the USAF Test Pilot School and designed to host a variety of experimental payloads. Payloads could be quickly integrated and the RASCAL pod could be mounted on a variety of compatible aircraft. The data used in this research was from a series of flight tests sorties at Edwards AFB with the MAMI-II/Rascal Pod system mounted on an F-16. The system is shown in Figure 3.1.

The MAMI-II data was unique and important because of the large operating envelope of the F-16. The data were collected at speeds ranging from 200 to 600 knots and at altitudes between 500 feet and 30,000 feet above ground level (AGL). Previous data were limited by lower performance aircraft that could only achieve maximum speeds of 200 knots and maximum altitudes of 15,000 feet. The extreme altitudes and airspeeds allowed testing of the trajectory reconstruction algorithm in a realistic operational environment as any future visual navigation systems may have to operate on a high performance aircraft under these types of conditions. The MAMI-II hardware design was not optimal for this algorithm. The system was designed to satisfy multiple research objectives for AFRL that needed high resolution imagery at high altitudes and long ranges. This requirement drove the use of a lens with a much



Figure 3.1: The RASCAL pod (yellow and blue pod under the wing) containing the MAMI-II payload is shown mounted on an F-16.

longer focal length (400 millimeters) and narrower field of view ($.81^\circ$) than desired for this algorithm. The camera parameters for the MAMI-II test are shown in Table 3.4.

Fx	73676.8 pixels
Fy	73676.8 pixels
Cx	512 pixels
Cy	512 pixels
Radial	[0.0, 0.0, 0.0, 0.0, 0.0]
Image Size	1024 x 1024 pixels

Table 3.4: MAMI-II Camera Parameters

Figure 3.2 shows the components of the MAMI-II system within the RASCAL pod. An on-board Novatel Synchronous Position, Attitude and Navigation (SPAN) INS/GPS system provided truth navigation data for every image frame. The IMU used with the system was a Novatel SPAN HG1700-58. In order to fit the long focal length lens into the pod, a mirror was used so that the field of view of the camera pointed straight down out the bottom of the pod. This design complicated the determination of the transformation matrix between the camera and IMU frame

since the mirror essentially changed the camera frame orientation by reflecting the scene. The transformation matrix was computed by the MAMI-II test team using the process outlined in [5]. The transformation matrix between the reference camera frame to the IMU frame for the MAMI-II system was:

$$C_r^a = \begin{pmatrix} .0222 & -.9935 & -.1121 \\ .9988 & .0270 & -.0406 \\ .0433 & -.1106 & .9924 \end{pmatrix} \quad (3.4)$$

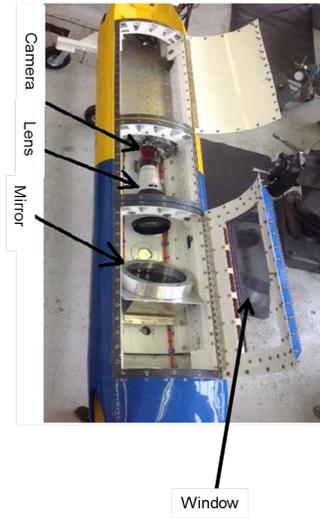


Figure 3.2: The MAMI-II camera payload is shown inside the RASCAL pod. This view is from the bottom of the pod and shows the mirror used to reflect the image so that the camera field of view looks out the bottom of the pod. In this image the panels are opened for maintenance but during flight the pod was sealed and the window under the mirror was the viewing portal.

3.2.2 Apply SFM to Series of Images. This project used the software package Visual Structure from Motion (VSFM) developed by Changchang Wu to implement the majority of the Structure from Motion process described in Chapter 2 [37]. VSFM is closed source software but there are multiple paths whereby the user can ad-

just settings and manipulate the algorithm to achieve desired results. These features make VSFM the ideal tool for this research. The only required inputs to VSFM are a set of images. Using the input images, VSFM runs SIFT on each image, matches SIFT features, builds a sparse 3-D reconstruction and computes sparse bundle adjustment to refine the solution. The 3-D reconstruction is displayed to the user and an output text file is generated with the positions and orientations of all the reconstructed model points and cameras in an arbitrary VSFM model reference frame.

3.2.2.1 Simulator Interface to VSFM. VSFM generally required input JPEG images; however, the SIFT simulation used in this research only generated data files with a list of feature locations and feature identifiers in each simulated frame and not actual images. Fortunately, VSFM allowed the user to input defined features and matches instead of having to use the built in SIFT based matching process. To do this the user must write and input a .SIFT file that contains feature locations for each simulated image and a text file that specifies feature matching between simulated images. In order to input the simulated feature locations and matches, two Matlab scripts were written: writefeat.m and makematches.m. The writefeat.m script takes the simulated feature locations and writes them to a .SIFT file that can be input to VSFM in lieu of having VSFM run its own SIFT matching. In the process it is possible to inject noise into the simulated feature locations as desired. The makematches.m script uses the simulator generated feature identifiers to match features across simulated frames. The feature matches are written to a text file in the format specified by VSFM documentation. Finally, even though the SIFT matching is already specified by the input files, actual JPEG images are still needed by VSFM to complete the graphical display of the reconstruction. Therefore, completely black (arrays of zero) JPEG images were generated and sized appropriately for the simulated camera. This allowed VSFM to run as if it were processing normal images.

3.2.2.2 VSFM Parameters. When using actual imagery collected from flight test data the only step that needs to be accomplished is to load the images into

VSFM. From this point a 3-D reconstruction can be generated without any additional data; however, there are several parameters that can be modified to influence the quality and speed of the reconstruction. The first option that can significantly affect the reconstruction is related to lens distortion. As discussed in Chapter 2, radial and tangential lens distortion can have a significant impact on the pixel location of a given feature point. VSFM can estimate lens distortion in the process; however the results are not as reliable as an independent measurement of lens distortion. If available, the known distortion parameters for each set of flight test data were used to undistort images prior to inputting them into VSFM. This often improved results and having access to lens distortion parameters is a realistic assumption for a navigation system.

The next set of parameters are the SIFT matching parameters. VSFM matching uses a nearest neighbor criteria as well as homography and fundamental matrix constraints as described in Chapter 2. The user can modify the distance ratio for nearest neighbor matching, the pixel re-projection thresholds for the two geometry constraints, the maximum number of matches and the RANSAC sample size. In the default setting VSFM will attempt to match every image in the set to every other image. For image sequences (as used in this research), the user can specify what images should be attempted to match to other images. For example, one can choose to only attempt to match an image with the X images immediately before and after it in the sequence. This can significantly speed up reconstruction as time is not wasted trying to match images that may not share common features. Even though all data were post-processed for this research, processing time will be an important consideration for any future operational implementation of this algorithm. Finally, since this research deals with sequences of images from the same camera, the reconstructions used a 1 pixel homography and fundamental matrix reprojection error constraint to reject outlier matches. This ensured sub-pixel registration between images in the sequence.

Once feature matching is complete there are several parameters that influence the 3-D reconstruction. Perhaps the most important parameter is the initial guess of focal length. VSFM generally uses Exchangeable Image File Format (EXIF) tags from

digital cameras for this initial estimate; however, this EXIF data was not included in the flight test images so a rough estimate of the camera focal length must be input manually. A guess of focal length that is wildly inaccurate may lead to convergence on an incorrect local minimum and failure of the reconstruction. If the actual camera calibration is known (as is the case for this research) then the camera matrix can be input and VSFM will constrain the reconstruction to fit this known calibration. This is currently the only way in which VSFM will constrain the bundle adjustment (ie. the user is not able to input known camera rotations, etc). Finally, there are several thresholds describing how often bundle adjustment is done in the reconstruction process and when to add new cameras or tracks to the reconstruction. Many of these settings are already optimized for the best reconstruction so this research will only focus on adjusting the parameters that may be important in navigation applications.

3.2.3 Real World Transformation. The VSFM software gives outputs that are referenced to an arbitrary frame and scale. It is therefore necessary to convert the VSFM output model to a real world frame prior to the final bundle adjustment step. While relatively straightforward, the development of this process and the understanding of the associated errors is a major contribution of this research. Much current work in 3-D reconstruction is only concerned with the qualitative result of the reconstruction. In other words, does the reconstruction look like it is supposed to look on a relative scale? Military applications, including navigation and targeting, require that the reconstruction can be accurately associated with a real world coordinate system and scale. This is done by first converting the camera orientations output by VSFM in the arbitrary VSFM model frame (denoted ‘m’ for model frame) to the frame of the first camera in the sequence called the reference camera frame (denoted ‘r’). The known IMU orientation of the reference camera is then used to associate the model frame with a real world East-North-Up frame and, from there, each VSFM camera orientation can be converted to an orientation relative to ENU. The overall equation for this process is as follows:

$$C_{ENU}^c = (C_m^c)^N C_r^m C_{a1}^r C_{NED}^{a1} C_{ENU}^{NED} \quad (3.5)$$

where $(C_m^c)^N$ is the VSFM model frame to camera frame rotation as output by VSFM for the Nth camera, C_r^m is the rotation of the reference camera (first camera in sequence) frame to the VSFM model frame as output by VSFM, C_{a1}^r is the rotation from the first aircraft frame to the reference camera frame as determined by camera mounting parameters, C_{NED}^{a1} is the rotation from NED frame to aircraft frame as given by the IMU for the first camera (assuming IMU is mounted in aircraft frame) and C_{ENU}^{NED} is the conversion from East-North-Up (ENU) to North-East-Down (NED). This equation represents the rotation of each camera relative to the ENU frame as determined by VSFM.

Camera positions must also be converted from the VSFM model frame to the ENU frame prior to the final bundle adjustment. VSFM outputs the translation of each camera from the arbitrary model frame origin to the camera center as expressed in the camera frame. The position of the Nth camera in the VSFM model frame is then given as:

$$(x^m)^N = -(C_c^m)^N (t^c)^N \quad (3.6)$$

where $(x^m)^N$ is the position of the Nth camera in the model frame, $(t^c)^N$ is the translation in the Nth camera frame output by VSFM and $(C_c^m)^N$ is the rotation of the Nth camera to the model frame as output by VSFM. The position of each camera in the model frame is then converted to a position relative to the first camera in the sequence (the reference camera) by

$$(x^{ref})^N = C_m^r ((x^m)^N - (x^m)^1) \quad (3.7)$$

where $(x^m)^1$ is the position of the reference camera in the model frame. Finally, the known orientation of the reference camera is used to compute the position of each

camera in the ENU world frame. In this case the ENU world frame has an origin at the position of the first camera.

$$x^{ENU} = C_{NED}^{ENU} C_{a1}^{NED} C_r^{a1} (x^{ref})^N \quad (3.8)$$

where C_{a1}^{NED} and C_r^{a1} come from the mounting parameters and IMU orientation of the reference camera. The VSFM outputs have now been successfully converted to a real world coordinate system; however, the data still needs to be scaled appropriately.

A second known reference location is used to assign a scale to the model. Although any model point can be used, as described in Chapter 2, for navigation purposes the second camera in the sequence is chosen. This means that the first two camera measurements must be accompanied by some sort of independent relative measurement in position. Although the first camera requires a real world coordinate measurement (GPS, surveyed starting point, etc.) the second camera position just needs to be known relative to the first. This can be done either with another absolute measurement (GPS, surveyed point) or by a relative method (INS, dead reckoning, stereo camera, etc.). A scale factor, K , is then determined using the ratio of the distance between the two reference cameras in the VSFM model frame and the independently measured distance. Multiplying the scale factor by Equation 3.7 gives the position of all cameras in an ENU frame with origin at the first camera.

If it is not desired to run any further bundle adjustment steps using more constraints, then this is the final step and the solution can be compared to truth data. However, it might be possible to improve the quality of the solution by incorporating known constraints measured from other reliable sources into a final bundle adjustment.

3.2.4 Post-VSFM Bundle Adjustment. As mentioned earlier, the only parameter that can be fixed in the VSFM bundle adjustment is the camera calibration. For navigation applications, it may be desirable to further constrain the bundle ad-

justment by incorporating known camera orientations from an IMU measurement or even incorporating known altitudes from altimeter measurements. Finally, the positions of the first two reference cameras are known and serve as additional constraints. In theory, more known constraints incorporated into the bundle adjustment should improve reconstruction accuracy. Unfortunately, VSFM can only constrain bundle adjustment using camera calibration. A Matlab based software package called Vision Lab Geometry (VLG) Library developed by the Vision Lab at UCLA provides an open source bundle adjustment implementation with more flexibility [20]. This software requires an initial guess of the 3-d structure and camera locations that must be relatively close to the actual structure. Using this initial guess, a Levenburg-Marquardt algorithm is implemented to minimize reprojection errors and refine the reconstruction. The VSFM reconstruction is used as the initial guess and is input to this software. When no parameters are changed from the VSFM reconstruction then the output is the same result as given by VSFM. However, the VLG software allows the user to fix not only camera calibration but also camera position. Additionally, modifications to this software made by the author provide the ability to fix the orientation of each camera based off known IMU data and each camera's altitude based off altimeter data. This is done by setting the terms in the Jacobian matrix that represent the change in reprojection error due to variations in camera orientation, altitude, position or calibration to zero. Recall the structure of the Jacobian matrix A discussed in Chapter 2 and repeated in Figure 3.3.

This Jacobian represents unconstrained bundle adjustment; however, setting the terms dealing with camera orientation, altitude, position and calibration to zero fixes calibration, altitude, position and camera rotation to the values input in the initial guess. The new structure of the Jacobian matrix is shown in Figure 3.4:

In order to constrain with respect to camera orientation, the camera orientation input needs to be the actual camera orientation as measured by the IMU. The IMU measures camera orientation in the real world. In other words, it gives the transformation from the NED frame to the body frame. Therefore true orientations of each

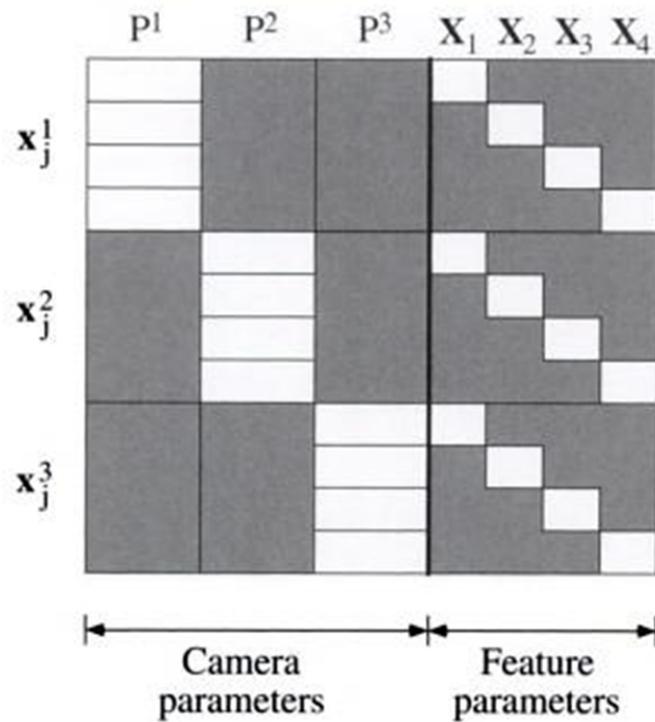


Figure 3.3: Original Unconstrained Jacobian Matrix

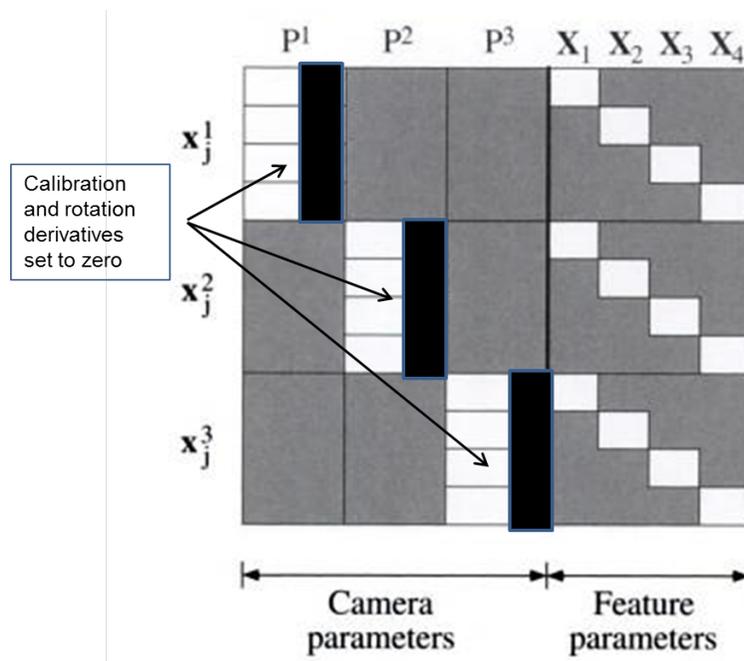


Figure 3.4: Desired parameters are constrained to their initial guess by setting the appropriate Jacobian terms to zero.

camera are calculated using the following equation as opposed to Equation 3.4 which gave the VSFM estimate.

$$C_{ENU}^c = (C_a^c)^N (C_{NED}^a)^N C_{ENU}^{NED} \quad (3.9)$$

where C_{NED}^{a1} from Equation 3.4 is replaced by $(C_{NED}^a)^N$ which is the IMU orientation of the Nth aircraft in the sequence and $(C_a^c)^N$ is the transformation from the aircraft body to the Nth camera, which is generally constant but can vary for a gimbaled camera system.

Equation 3.8 gives the orientation of the Nth camera with respect to the ENU frame centered on that camera. However, the bundle adjustment must be constrained with the orientation of the Nth camera with respect to the ENU frame centered on the first camera. Since the aircraft is traveling over a ellipsoidal Earth, there will be a difference in pitch between these two orientations. This “transport” pitch is determined by the distance traveled over the Earth between the first and Nth camera. This must be incorporated into the pitch measurement of the IMU prior to applying Equation 3.8.

In addition to IMU orientation data, accurate altitude measurements are often available from a barometric altimeter. This measurement corresponds to the Up component of the ENU frame. In this case the Jacobian term relating changes in reprojection error to variations in Up position for each camera is set to zero and the Up position for each camera (H) is fixed as the difference between the altitude of the reference camera and the measured altimeter of the Nth camera:

$$H = Altitude^N - Altitude^r \quad (3.10)$$

Other variations on the constrained bundle adjustment are certainly available; however, the most useful for navigation purposes are fixed camera calibration, fixed camera orientation, fixed camera altitude, and fixed reference camera positions. A

major contribution of this research is exploring the implications of using these different types of constrained bundle adjustments on the navigation solution.

3.2.5 Use Calculated Trajectory for Navigation. Once a trajectory has been calculated in this manner there are several ways to use the information for effective aerial navigation. In the simplest approach, the last image in the sequence was taken at the current position and therefore determining the position of the last image gives the user's current location. In order for this approach to be useful the entire algorithm must be able to run fast enough to process all images and determine location before the user moves significantly from the position where the final image was taken. In a high speed aircraft this may be a difficult task; however, lagging position updates can still be useful when incorporated with other navigation systems via a Kalman Filter. Additionally, information about a past trajectory is useful because the SFM derived trajectory can be compared to other estimates of trajectory to identify errors. For example, suppose the GPS solution is being "spoofed" by an adversary in an attempt to steer the navigation system in the wrong direction. In this case, the false GPS trajectory would not match the SFM derived trajectory thereby alerting the system of a problem.

Irregardless of the exact application, the most useful way for an SFM trajectory reconstruction system to operate is to continuously calculate the current position relative to a known starting point as sequential images are added. The sequential images are processed in bundles. Each bundle consists of N-images that are processed together where the last image in the bundle was taken at the current time (minus processing delay). The minimum information needed to navigate using this scheme is a known starting point, a known starting camera orientation and a known distance between the first two camera positions in the sequence. Figure 3.5 gives a visual depiction of this process.

The process in Figure 3.5 assumes that no other scale updates are used and the scale of the reconstruction is completely set by the first two camera positions. This is

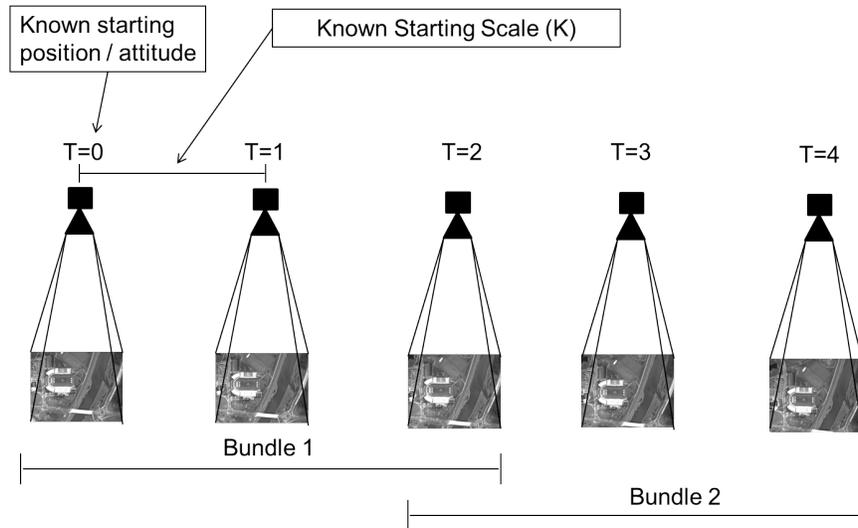


Figure 3.5: A proposed architecture for navigation that requires a known starting point and a known reference distance between the first two camera positions.

the case when using a single camera and no other position updates are available. In many cases, it may be possible to continuously update the scale of the reconstruction using other information. The simplest scheme for continuously updating scale is to use a stereo camera system where two images are taken simultaneously and the distance between the two cameras is known and fixed. This is depicted in Figure 3.6.

If only a monocular camera system is available, the distance between sequential images can be continuously estimated using an independent speed measurement and dead-reckoning over short time intervals as depicted in Figure 3.7.

Finally, the scale of the reconstruction can be set using feature 3-D positions. For example, suppose the altitude of the aircraft above the ground is known from a radar altimeter. In this case, the scale of the reconstruction can be continuously updated with the known position between a camera and a feature point on the ground. This situation is depicted in Figure 3.8.

No matter the method used for setting the reconstruction scale, the reconstruction can be further constrained within each bundle using other known parameters such as camera orientation from IMU data, camera altitude from an altimeter and

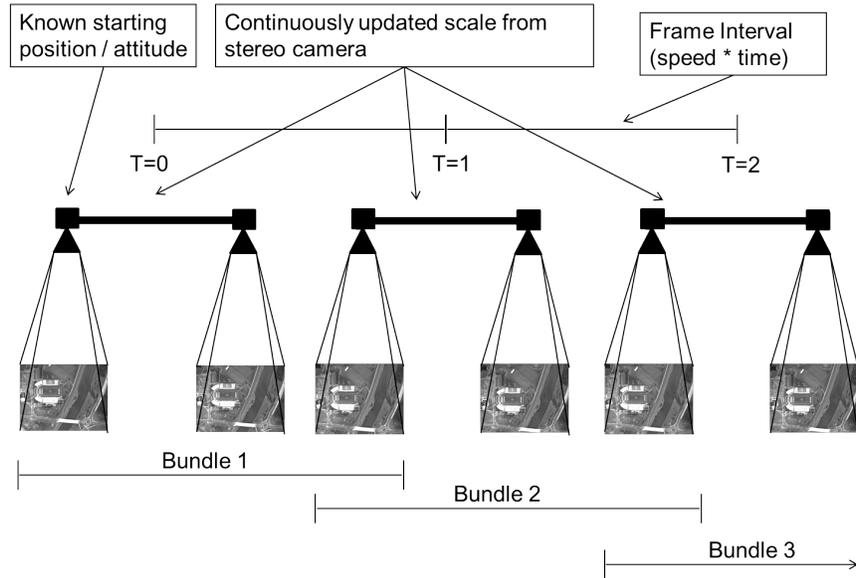


Figure 3.6: A proposed architecture for navigation that requires a known starting point and a stereo camera system.

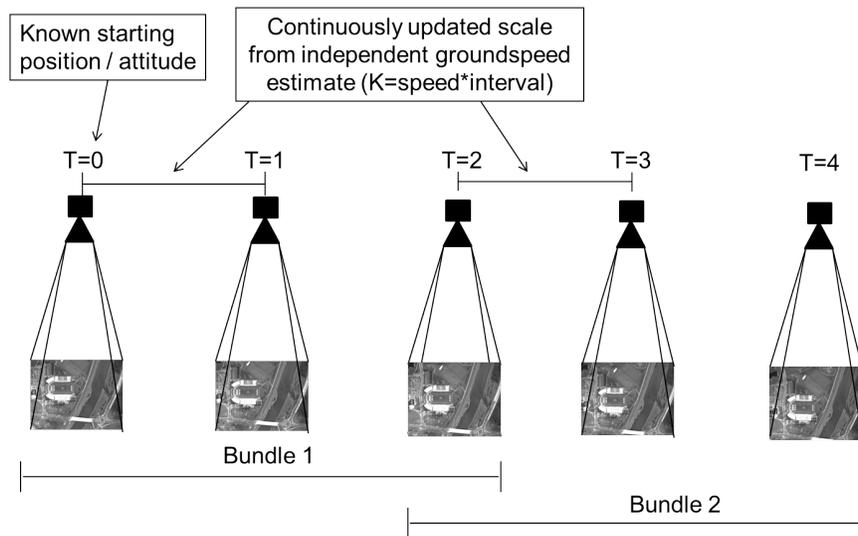


Figure 3.7: A proposed architecture for navigation that requires a known starting point and an independent estimate of groundspeed for each bundle.

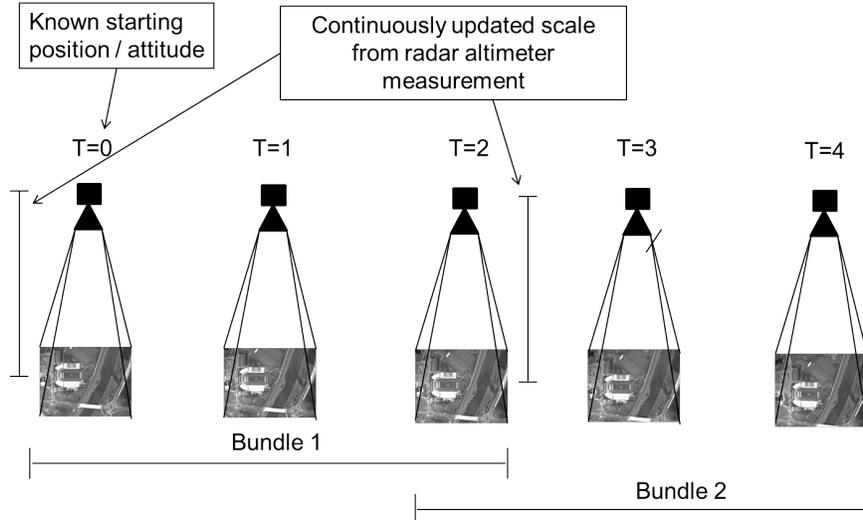


Figure 3.8: A proposed architecture for navigation that requires a known starting point and an altitude estimate for each bundle.

camera calibration. The exact implementation of the algorithm therefore depends on what other information is available to aid in navigation, but, in its most basic form, the system only requires sequential images, a known scale and known starting point.

3.3 Software Tools

A combination of commercial, government and open source academic software tools were used to implement and test this navigation algorithm. The algorithm itself is implemented in a Matlab wrapper written by the author that in turn calls functions from Visual Structure from Motion (VSFM), the UCLA Vision Lab Feature (VLFeat) and Vision Lab Geometry (VLG) function libraries, the Open Computer Vision (OpenCV) function library and the Matlab image processing toolbox [37] [20] [39]. The input to the overall Matlab wrapper is a set of images and the associated truth position data for each image. As part of development and testing, simulated images and position data were generated using the the AFIT ANT Lab SIFT simulator and an Air Force Research Lab (AFRL) computer program called ProfGen. ProfGen generates a user specified aircraft trajectory and a simulated IMU/GPS data file for that trajectory. The AFIT ANT Lab SIFT simulator uses a ProfGen trajectory and

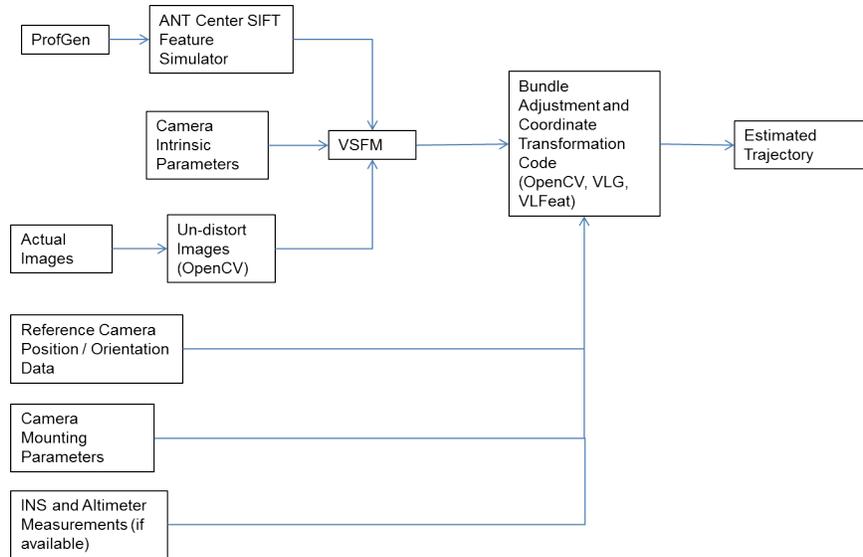


Figure 3.9: The implementation and testing of the algorithm is done using a combination of the software tools outlined above.

an Earth model to generate simulated SIFT features on the ground below the aircraft. The program determines which features would be seen by a camera mounted on the aircraft. Matlab code written by the author uses this output to generate simulated images and data files that are input to the algorithm in the same way that real images and data files are input. An overview of the interaction of software tools is shown in Figure 3.9. All the code except VSFM and ProfGen is implemented within Matlab. In the cases of the VLG, OpenCV and VLFeat libraries some functions are implemented via MEX files that in turn call compiled versions of C++ functions from these libraries. This is necessary since calculation in Matlab on large image data sets can be very slow.

3.3.1 Algorithm Speed. The focus of this research effort was on developing the navigation algorithm and characterizing the associated errors so the software tools were optimized for post flight error analysis and not algorithm speed. However, in order for the algorithm to be useful it must be able to operate in realtime with only small delays between the time an image is taken and a position estimate is calculated. The delay between the last image and the formation of a position estimate is a function

of the time required to run SIFT on each image, the frame rate and the time required to complete SFM on each image bundle. The time required to complete SIFT on each image is a function of the size of the image and the number of features in the image. The time required to complete SFM on each image bundle is a function of the number of images in the bundle, the number of features per image and the constraints used in the bundle adjustment. For the scheme depicted in Figure 3.5, the total delay between the last image and the position estimate is the time required to do SIFT on the last image plus the time required to do SFM on the bundle. Although speed was not closely tracked during this research, the processing delay for a three image bundle of 1200x1200 JPEG images was about .6 seconds in C++ code plus 6 seconds for the supporting Matlab code. All data was processed on a Windows 7 laptop with 16 GB of RAM and a 2.2HGz Intel(R) Core(TM) i7-2670QM CPU. The Matlab code performs a second bundle adjustment that is only necessary because VSFM is closed source. Most of the functions performed in the Matlab code would therefore not be necessary in an operational system so the delay of an operational system could probably be reduced to less than one second.

3.4 Limitations

The use of both simulated and real world data to test the algorithm provided greater flexibility in assessing and controlling sources of error; however both these approaches were not without their limitations. The primary limitation of the simulation approach is that the simulator is unable to accurately render simulated features for aggressively maneuvering trajectories, camera mounting angles close to horizontal or very high altitude flight. In these cases, the simulator outputs far too many visible features per image than SIFT would actually generate. This is because the simulator thinks that the camera can see features that are very far away instead of limiting the total number of features. The very large number of features is too high to effectively run the process in a reasonable amount of time and this large number of features does not represent real world images. This limitation can be fixed in future versions of the

SIFT simulator but for the purposes of this research the test cases were designed to avoid this limitation. Additionally, the simulated features are distributed randomly and uniformly across the Earth surface. There are no feature clusters or areas that lack features as might be seen in the real world.

The use of pre-compiled MEX libraries as well as some closed source software also provided two main limitations. The first is that the ability to constrain altitude in the bundle adjustment is limited to the straight and level flight profile with a straight downward looking camera. Although, this effectively limits the testing of altitude constrained bundle adjustment to one test case, this test case is enough to demonstrate the concept. Second, when constraining bundle adjustment using IMU rotation information, the accuracy of the yaw, pitch and roll is limited and numerical errors can be as high as $1e-4$ depending on camera configuration. This is due to the process used to convert between rotation vectors and direction cosine matrices. Although significant in some cases, this error is small enough for the purposes of this analysis.

3.5 Theoretical Error Sources

This section outlines the major sources of error as predicted by theory in the proposed algorithm. The goal of this research is to investigate the effect of these error sources on the final reconstruction and the ability to use that reconstruction for aerial navigation.

3.5.1 SFM Parameter Dependent Errors and Numerical Errors. As discussed in Chapter 2, SFM is essentially a non-linear least squares estimation process and therefore is subject to limitations depending on the estimation routines and parameters used. For example, an inaccurate guess of initial focal length can wreak havoc on the solution by converging on a false local minimum. Additionally, parameters in the bundle adjustment process control the number of iterations used when converging on a solution. These types of parameters can be varied and often are a

tradeoff between speed and accuracy. For the purposes of this research, these parameters will be left at the default settings used in the VSFM and VLG software packages. Additionally, the errors associated with varying these parameters are purely process errors and are small compared to the other sources of error considered.

Inevitably, numerical errors in computation are present. In general, these errors are very small (1e-15); however, there are some instances in this research where numerical errors become as high as 1e-4 due to inherent limitations in the software used as well as the combination of multiple sequential computations. The results presented will distinguish between numerical errors and other sources.

3.5.2 Accuracy of Bundle Adjustment. As outlined above, the final step of SFM is the minimization of feature reprojection errors known as bundle adjustment. After linearization, bundle adjustment is a least squares minimization problem in the form of:

$$z = Jx \tag{3.11}$$

where z is the measurement vector of pixel locations, x is the state vector of camera and 3-D point positions and J is a Jacobian [29]. Assuming that the noise in the measurement is zero mean and Gaussian distributed, the covariance of each estimated state can be found as:

$$P_x = (J^T P_z^{-1} J)^{-1} \tag{3.12}$$

where P_x and P_z are the covariance matrices of the states and measurements, respectively [29]. In the case of bundle adjustment, however, this calculation quickly becomes impractical due to the number of cameras and the number of 3-D points. The length of the state vector is six times the number of cameras (6 degrees of freedom DOF) per camera, assuming camera calibration is known) plus 3 times the number of 3-D points (3 DOF per point). The length of the z vector is two times the total

number of features seen by all the cameras (x,y pixel location of each feature gives 2 DOF). Fortunately, the block diagonal Jacobian structure allows for a more efficient calculation of covariance. Hartley and Zisserman outline this process and provide algorithm A6.4 in [15] for calculating the covariance of the states estimated using bundle adjustment. The algorithm was implemented for this research in order to estimate the accuracy of the trajectory reconstructions. Note that the covariance of the solution states depends on the Jacobians and the magnitude of the measurement noise. The nature of the Jacobian matrix is determined by the camera intrinsic parameters and the geometry of the cameras as related to the scene features. The nature of the measurement noise is related to camera intrinsics, feature matching methods and the system's operating environment.

3.5.3 Factors that that Affect Error through the Jacobian.

3.5.3.1 *Camera Intrinsic Parameters - Image Resolution.* The Jacobian structure contains information about the following camera intrinsic parameters: the camera pixel size, focal length and image plane size. The combined effect of pixel size and focal length determines the maximum image resolution as limited by digital quantization effects (ie. not considering optical diffraction limitations on resolution) for a given distance between the target and the camera. This maximum resolution is measured as the ground sampling distance (GSD). The GSD is the ratio of meters to pixels in the image. For an image taken by a camera pointing straight down from an aircraft toward flat terrain below, the GSD throughout the image is constant and is given by the following equation:

$$GSD = \frac{H}{fs} \quad (3.13)$$

where f is the focal length in meters, s is the quantity of pixels/meter on the actual camera focal plane and H is the camera AGL altitude [23]. Clearly, higher resolution images (small pixels, long focal lengths and shorter distances) minimize pixel quanti-

zation error by having a higher number of pixels per meter in an image. Large pixel quantization errors (large pixels, short focal lengths and longer distances) mean that the pixel locations of features input to SFM from SIFT are less precise which in turn leads to less precise estimation of the fundamental matrices between cameras. Note that that GSD within an image can vary if the camera is not looking straight down on flat terrain. For example, a camera that is looking down but slightly forward or to the side of the aircraft is capturing ground targets at different distances within its field of view so H must be replaced with slant range from camera to feature and this range is no longer constant for each part of the image.

3.5.3.2 Camera and Scene Geometry Affects - Maximum Separation Angle. The number of cameras used and the relative positions of those cameras to each other and to scene features are also important considerations in reconstruction accuracy that are manifested through the Jacobian. Ekholm showed in [12] that at least three cameras and a minimum convergence angle of 6° is generally required for an accurate target 3-D reconstruction. In other words, cameras without sufficient angular separation lead to poorly conditioned matrices that do not produce accurate results. In a sequence of images taken from an aircraft this means that camera mounting angles, camera frame rate, camera field of view, altitude and aircraft trajectory will play an important part in the accuracy of the navigation solution. Mounting angles, frame rates, fields of view, altitudes and trajectories that allow for angular separation between images will likely perform better than those that do not. This is similar to the concept of GPS Dilution of Precision (DOP) where the geometry used in triangulation of points has an effect on the estimation error [29]. Imagine trying to triangulate a 3-D point from two cameras that are only 3° apart versus two cameras separated by 30° . Figure 3.10 illustrates this concept with three cases.

The cone emanating from each camera illustrates the feature position uncertainty caused by pixel noise and image quantization error. This uncertainty (quantified by GSD) increases with distance between camera and feature. The maximum

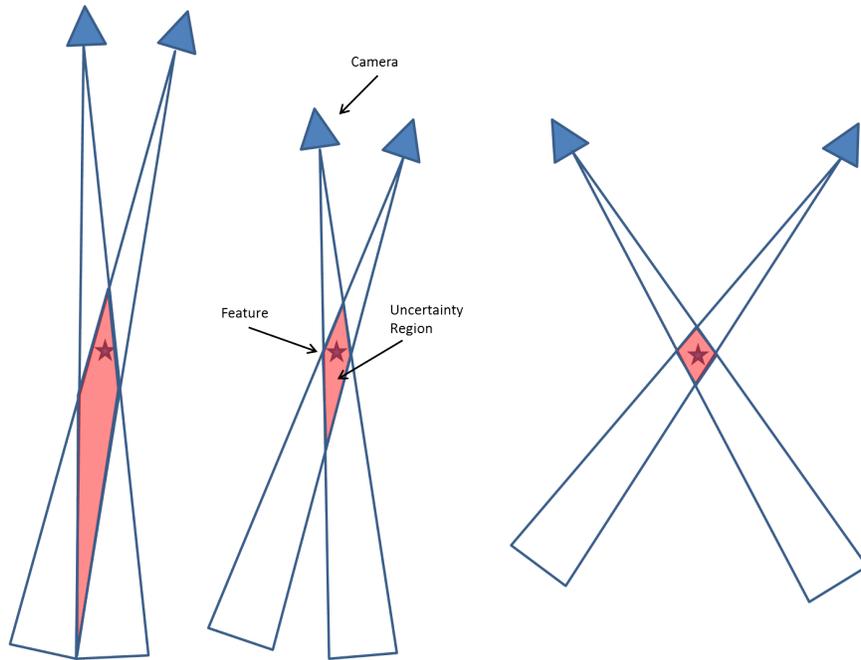


Figure 3.10: Angular separation between cameras and features changes the uncertainty in position estimates. The situation on the left is the worst case since the cameras are close together and far away from the feature.

angular separation between two images is a function of the distance between the two cameras that took the images, camera field of view, camera orientations and the distance between the cameras and the common target feature. Clearly triangulation, in the presence of noise, using angular separations near 90° will be more accurate than separations near 0° , but the relationship between accuracy and angular separation is not linear and is tightly coupled with pixel noise and GSD. The accuracy of the calculated feature and camera positions in a given axis is also variable. For example, in the case on the left hand side of Figure 3.10, the vertical position of the feature is far more uncertain than the left/right feature position.

For the situation of a straight and level flight profile with a downward pointing camera, the triangulation angle varies for each overlapping feature. For example, a feature directly between the two cameras has higher triangulation angle than a feature to the left or right of both cameras. This situation is shown in Figure 3.11.

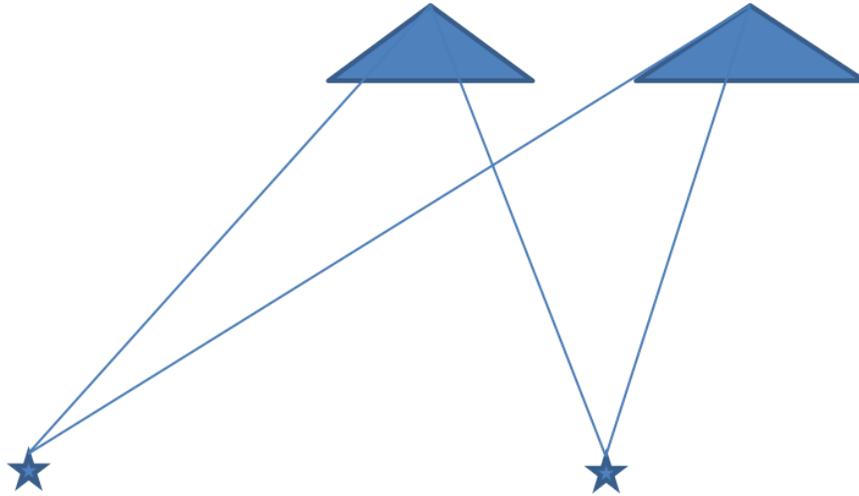


Figure 3.11: For cameras looking straight down, triangulation angle varies depending on feature location relative to the two cameras.

Assuming features are evenly distributed on the ground below the two cameras, the triangulation angles used in SFM between the two cameras will have a distribution as shown in Figure 3.12.

The maximum angular separation occurs between the two overlapping camera positions is given by the following equation derived from trigonometric relationships:

$$\Phi = 2 \tan^{-1}\left(\frac{d}{2H}\right) \quad (3.14)$$

where d is the distance between cameras when each image is taken and H is the camera altitude. Note that this equation only applies to the straight and level case with a downward pointing camera flying over flat terrain. The equation assumes that the camera field of view is wide enough so that images overlap. This is the maximum possible angular separation for triangulating a feature that sits exactly between between the two images; however, there will be many overlapping features with significantly less angular separation as shown in Figure 3.12. The equation shows that larger camera altitudes lead to smaller angles and larger distances between cameras lead to larger angles for triangulation. For an aircraft mounted system, this

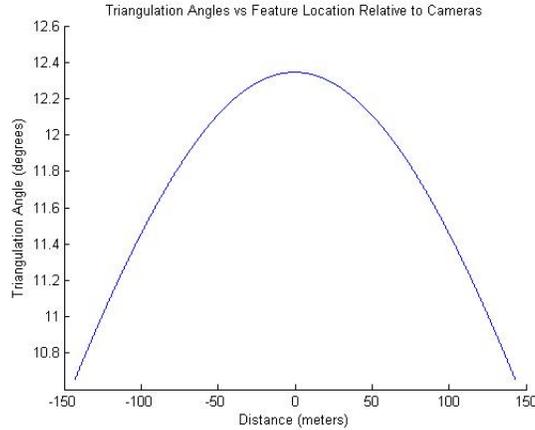


Figure 3.12: The triangulation angle for a given feature varies as a function of the feature position relative to the cameras. In the above plot, the $x = 0$ point is the center between two cameras. As features get farther away from that center, the triangulation angles decrease as a tangent function that depends on the altitude of and the distance between the two cameras. Note that the x distance is limited by the overlap between the two camera images. In this case the total overlap length is about 300 meters ($\pm 150\text{m}$).

means that angular separation is affected by altitude, speed, frame rate, camera mounting and aircraft trajectory.

3.5.3.3 Feature Distribution and Image Overlap. Figure 3.10 illustrated the effects of angular separation using one feature; however, a given image may have thousands of SIFT detected features. These features may be evenly distributed or clustered in certain parts of an image. Clearly, more matching feature points between images gives more measurements for the least squares minimization. Since less overlap between images means less matching features, it is expected that decreasing overlap will reduce accuracy. Additionally, the distribution of these features and the amount of overlap between images changes the nature of the Jacobian matrix. Imagine matching two images where features are clustered in only one small part of each image or two images that only have a small amount of overlap. In this case, the Jacobian may be poorly conditioned leading to a less accurate estimate of

the fundamental matrix between images. In an airborne system, is not possible to control feature distribution as this is a function of the type of terrain overflown; however, image overlap is a function of camera frame rate, altitude, speed, field of view and trajectory. For a straight and level trajectory with a downward pointing camera, the percent overlap between two images is the overlapping area on the ground divided by the total ground footprint of one image. From trigonometric relationships and the camera model equations, percent overlap is given as:

$$Overlap = \frac{2H \tan(\frac{\Theta}{2})(2H \tan(\frac{\Theta}{2}) - d)}{(2H \tan(\frac{\Theta}{2}))^2} \times 100 \quad (3.15)$$

which simplifies to

$$Overlap = \frac{(2H \tan(\frac{\Theta}{2}) - d)}{2H \tan(\frac{\Theta}{2})} \times 100 \quad (3.16)$$

where d is the distance between cameras when each image is taken, H is the camera altitude and Θ is the total camera angular field of view. This equation also assumes that the image plane is square (ie. equal number of pixels on each side).

3.5.3.4 Jacobian Constraints. As outlined above, the four main factors that should influence the accuracy of bundle adjustment and whose information is encoded in the Jacobian are: image resolution, maximum separation angle, feature distribution and image overlap. These four factors are in turn determined by many different camera and geometry parameters. This leads to large Jacobian matrices with many degrees of freedom and un-observability between certain parameters. For example, since bundle adjustment simultaneously solves for the position and attitude of each camera there is some un-observability between errors in camera orientation and errors in camera position. Additionally, since image GSD is a function of both camera intrinsic parameters and the distance between the camera and the target, there is some un-observability between errors caused by distance effects or by camera intrinsics. A technique for improving the bundle adjustment solution by constraining

various parameters in the Jacobian was presented earlier. When using this technique the accuracy of the constraints may have an effect on the overall reconstruction. The primary constraints used in this research are the camera calibration parameters and IMU measured camera orientations. Camera calibration errors are assumed to be in the form of a bias in focal length or principal point due to lens distortion. Angular errors in the orientation of each frame are due to either zero mean, Gaussian noise and/or bias in the IMU measured yaw, pitch and roll angles. It is possible that when errors in these constraints become too large, the reconstruction may fail or be grossly inaccurate.

3.5.4 Factors that Affect Error through Measurement Noise. Quantization error is not the only reason that there may be errors in the pixel locations used for fundamental matrix estimation. The SIFT process itself may be limited in how accurately it can locate features based on either image geometry or feature type. For example, less distinct features (ie. weak corners) are not as easy for SIFT to find and describe and therefore are not as accurately located within an image. There also exists the possibility of false matches. The matching geometry constraints discussed in Chapter 2 can be tuned to eliminate false matches and reduce the feature location pixel error to a given threshold. This pixel noise threshold has a strong effect on the accuracy of the final reconstruction. Additionally the number of successfully matched features will influence the reconstruction. More matching features between images means a more robust least squares estimation of the fundamental matrix so images with more feature matches should produce better reconstructions in the presence of noise. The assumption is made in this research that the pixel errors due to SIFT matching after geometry constraints are applied can be modeled as zero mean, uncorrelated Gaussian noise [28]. Finally, radial and tangential lens distortions that are not properly removed will lead to pixel errors by injecting error into the measured pixel location. When available, camera lens coefficients are used to remove distortion prior to the SFM process.

3.5.5 Real World Transformation - Angular Errors. The process of assigning a real world reference frame to the model produced by SFM introduces error to the reconstruction. This transformation requires both the information on how the camera is mounted to the IMU/aircraft and the measured yaw, pitch and roll at the time of the first image frame. Since the first camera of an image bundle is used to fix real world orientation for that bundle, any orientation error in that camera will be linearly related to position error as a function of distance by the following equation derived from trigonometric relationships:

$$\delta x = r\delta\theta \tag{3.17}$$

where r is the total distance flown and $\delta\theta$ is the angular error (assuming small angles). Inaccuracies in the orientation of the first camera have a clear and dramatic effect on the overall solution since the error is linearly increasing and can become very large over long distances.

3.5.5.1 Calculating the Transformation between Camera and IMU. As discussed in the previous section, the transformation between the camera frame and the IMU frame must be known accurately in order for the navigation algorithm to work with minimum error. The primary method for determining this transformation is with hardware measurements of the camera and IMU. This research effort found that it is possible to use bundle adjustment to calculate this transformation matrix if accurate position data for a number of sequential image frames is known. This can be done by calculating the relative positions of the sequential images with respect to the first camera frame position using the SFM process. These positions can then be compared to the known truth positions of each image frame (ie. from accurate GPS or other truth data) and a rotation matrix can be derived that best minimizes the errors between each calculated and known camera position. This type of problem is known in literature as Wahba's problem and several solutions exist [33]. The task is to minimize the following cost function:

$$J = \frac{1}{2} \sum_{k=1}^N a_k \|x_k^{a1} - C_r^{a1} x_k^r\|^2 \quad (3.18)$$

where x_k^{a1} is the known position of each camera in the aircraft 1 reference frame (from GPS), x_k^r is the calculated position of each camera in the reference camera 1 frame (from SFM process), a is a weighting factor and C_r^{a1} is the DCM between the IMU and the reference camera. A solution to this minimization is adopted from [33]:

$$B = \sum_{k=1}^N a_k x_k^{a1} (x_k^r)^T \quad (3.19)$$

$$B = USV^T \quad (3.20)$$

$$C_r^{a1} = UMV^T \quad (3.21)$$

where

$$M = \text{diag}([11\det(U)\det(V)]) \quad (3.22)$$

The result of this method is a transformation matrix between the camera frame and the IMU frame. This was determined only with truth position data about the trajectory and images taken from the camera. Other methods for determining this transformation matrix require hardware measurements or surveyed ground targets. The application of Wahba's method to determining camera mounting parameters is novel and is a small, but important contribution of this research because camera mounting parameters are important in a variety of applications aside from just navigation. This method may therefore be a useful alternative to hardware measurements or surveyed ground targets in some situations. The method is demonstrated on simulation and flight test data in Chapter 4.

3.5.6 *Real World Transformation - Scale Errors.* Structure from motion is a sequential process where the position of the Nth image frame in a sequence is calculated relative to the previous image in the sequence. The new relative position is then added to the position of the previous frame [38]. This process is based on a known starting point and the sequentially calculated positions. When the reconstruction is done in an arbitrary model frame, with origin at the first camera, then every time a relative camera position is estimated, that relative estimate is independent from previous estimates. Suppose that the position estimate of the Nth camera has an associated random error that is Gaussian distributed and zero mean with standard deviation σ . The value of σ is dependent on the previously discussed factors of overlap, GSD and maximum triangulation angle but is independent for each estimated camera since cameras are added sequentially. Independent random variables add by summing variances [36] so the standard deviation of the total error (in the model frame) at the end of the sequence is then the sum of N independent random variables and is given as:

$$\sigma_N = \sqrt{N}\sigma_{BA} \quad (3.23)$$

where N is the number of images starting at zero, σ_{BA} is the error in position estimate due to bundle adjustment in the model frame. Note that σ_{BA} and σ_N are vectors of the form:

$$\sigma = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{pmatrix} \quad (3.24)$$

since the standard deviation may differ in each direction of the model frame. The image at $N = 0$ is called the anchor point and, for a moving aircraft mounted camera, N is proportional to distance traveled. By definition, the model frame origin is an

anchor point so there is no error in this position in the model frame. This suggests that error propagates as a random walk in the model frame.

When the reconstruction is transformed from the arbitrary model frame and scale to a real world frame and scale, the scale is set using a second anchor point. As discussed earlier, a real world scale is assigned to the model by using the distance between the first two cameras in the reconstructed model compared with the distance between the first two cameras in the real world. The computed scale factor (K) is then given as:

$$K = \frac{X_2^{World}}{X_2^{Model}} \quad (3.25)$$

where X_2 is the position of the second camera in either the world or model frame. The position of the N th camera in the sequence in the world frame is found by applying the calculated scale factor to the model frame reconstruction:

$$X_N^{World} = K(X_N^{Model}) \quad (3.26)$$

When using a scale factor in the above form, the error in the N th camera position is still independent of the previous camera but it is now dependent on the total distance from the origin. Therefore, it is proposed that the standard deviation of the total error at the N th camera in the world frame is given as:

$$\sigma_N \propto X_N^{World} \sqrt{N} \sigma_{BA} \quad (3.27)$$

since a scalar can be multiplied by a standard deviation [36]. This can also be thought of as adding linearly correlated random variables where the number of correlated random variables is proportional to X_N^{World} and the standard deviation of each variable is $\sqrt{N} \sigma_{BA}$. Dependent random variables can be summed as described in [36]. Note that in the case when there is a constant distance interval between frames (R), the above equation becomes:

$$\sigma_N \propto X_N^{World} \sqrt{\frac{X_N^{World}}{R}} \sigma_{BA} \quad (3.28)$$

which can be re-written as:

$$\sigma_N \propto \frac{(X_N^{World})^{1.5}}{\sqrt{R}} \sigma_{BA} \quad (3.29)$$

This is an extremely important result that governs the behavior of the scale error. The scale error grows proportional the total distance traveled raised to the power of 1.5. The scale error will only grow in this form in directions that the aircraft flies. For other directions, the error is only dependent on the number of frames used and is governed by Equation 3.23. Finally, if the errors in bundle adjustment between two frames are zero mean and Gaussian, then the total error at the end of the sequence will also be zero mean and Gaussian.

3.5.7 Combining Bundles. Thus far the error analysis has been limited to describing the errors within one bundle of images with a constant scale set by the first two images. For the proposed navigation scheme, image bundles are combined in series so that the algorithm continuously calculates the current position. As discussed in Section 3.2.5, there are two ways to combine bundles: fixed scale factor or continuously updated scale factor. When bundles are combined using a fixed scale factor, the error is dependent on the total distance traveled. Equation 3.28 is adopted to the multiple bundle case by simply increasing the distance between the origin and the Nth camera by a factor of B, where B is the number of bundles:

$$\sigma_N \propto (BX_N^{World}) \sqrt{\frac{BX_N^{World}}{R}} \sigma_{BA} \quad (3.30)$$

which can be re-written as:

$$\sigma_N \propto \frac{(BX_N^{World})^{1.5}}{\sqrt{R}} \sigma_{BA} \quad (3.31)$$

This is same as Equation 3.29 except that the total distance between the origin and the Nth camera has increased. When bundles are combined using a constantly updating scale factor (ie. scale is updated in each bundle) then the total error in each bundle is independent of previous bundles and is given as:

$$\sigma_N \propto \sqrt{B} (X_N^{World} \sqrt{\frac{X_N^{World}}{R}} \sigma_{BA}) \quad (3.32)$$

or

$$\sigma_N \propto \sqrt{B} \frac{(X_N^{World})^{1.5}}{\sqrt{R}} \sigma_{BA} \quad (3.33)$$

Note that X_N^{World} in Equations 3.32 and 3.33 is the distance from the Nth camera of a given bundle to the first camera of the bundle not the total distance to the Nth camera in the sequence. The error now propagates as a random walk proportional to the number of bundles used in the total sequence. For a camera mounted on a moving aircraft, the number of bundles is proportional to the distance flown. All else being equal (same total distance, same frame rate, etc), the total error standard deviation for the updating scale case is less than the constant scale case by a factor of B. This is found by dividing Equations 3.31 and 3.33. This result is intuitive since the more scale updates used means there will be less error. The total scale error for any given situation, however, is a complex relationship between the number of bundles used, the frame rate, the distance covered in each bundle and the scheme used to update scale information. The above equations will be used to make predictions in specific simulation and real world test cases.

3.5.7.1 Quantifying Drift rate. When scale error is the dominant error, Equation 3.29 shows that the standard deviation of the error is proportional to the total distance traveled raised to the power of 1.5. In this case, the appropriate performance metric to use is a drift rate with units of $\frac{m}{km^{1.5}}$. This gives the standard deviation of position error in meters for a given distance flown. This metric applies only to the case of a fixed scale factor. When scale error is not the dominant error,

then total error is proportional to the square root of the distance traveled. In this case, the appropriate performance metric to use is a drift rate with units of $\frac{m}{\sqrt{km}}$. This metric applies only to the case of a continuous updating scale factor.

3.6 Testing Methodology

The previous sections gave an overview of the proposed navigation algorithm as well as the predicted sources of error. Although there are several different parameters that will likely affect the accuracy of the algorithm, this research focuses on those parameters most relevant to aerial navigation and those most likely to have significant impact on the solution. These critical parameters and error sources are the basis for experimentation in this project. The testing of this algorithm is broken down into three phases: Testing Ideal Cases with Simulation, Testing Noisy Cases with Simulation, Testing Real World Data. The intent of this testing process is to first prove that the algorithm works in the ideal case and to determine the upper limits of performance. Next, testing with controlled noise and other imperfections will give insight into real world performance, optimal parameters and error characteristics. Finally, real world flight test data is analyzed to validate simulation results.

3.6.1 Testing Ideal Cases with Simulation. An ideal test case was generated for use in algorithm debugging and to determine the upper limits of performance of the algorithm. It is expected that the navigation solution from this test will match very closely with the simulated trajectory. The only sources of error expected here are numerical errors, unavoidable process noise due to the estimation algorithms used (ie. Levenburg-Marquardt) and camera geometry/Jacobian effects. The baseline test case was an aircraft flying straight and level at 300 knots and 500 meters Mean Seal Level (MSL) over level terrain with an average altitude of 144 meters MSL. This profile used a sequence of 31 images taken every .5 seconds so as to cover 15 seconds of flight which is 2310 meters of motion. The camera has a focal length of 6675 pixels and an image size of 6800 x 6800 pixels. This large image size was chosen to minimize pixel

quantization error. This simulation was run as a single bundle with a fixed scale that was set by the first two cameras.

3.6.2 Testing Noisy Cases with Simulation. This phase of testing was used to inject controlled pixel noise and other variations into the straight and level simulation described above. These tests include variations in the following parameters:

- Bundle Adjustment Constraints
- Pixel Noise
- Image Resolution (GSD)
- Angular Separation Between Frames
- Image Overlap
- Camera Mounting
- Camera Calibration Noise
- IMU Noise
- Multiple Bundles with Scale Updates

Additionally, in order to study the effect of trajectory, there were four simulated 300 knot trajectories used: straight and level, straight climb, level turn and climbing turn. In all simulation tests, the camera flew over the same set of simulated terrain features and only the camera parameters or aircraft parameters were changed. The features were evenly distributed so that there were about 1000 features captured in every image.

3.6.3 Real World Data. Real world data from a variety of flight test sources was used and compared to simulated results. The real world data provided an opportunity to evaluate the accuracy of the algorithm when using actual SIFT features over different types of terrain. Additionally, real world flight test data allowed analysis of more aggressive trajectories than were possible with simulation. The real world

data was processed by dividing the data into bundles with three images each and then combining bundles sequentially using either a fixed or constantly updating scale factor. This process simulated a real-time navigation system that processed a bundle and gave a position update every time an new image was taken.

IV. Results and Analysis

4.1 Algorithm Verification

A first simulation was run without adding any pixel noise to verify that the navigation algorithm had been correctly implemented. In this case, the only constraint to the bundle adjustment was the known position of the first two cameras (the reference cameras). This simulation served as a baseline to show the upper limit of performance of the unconstrained algorithm. Even in the ideal scenario, pixel quantization error was unavoidable as feature points were projected into the quantized pixel space. Therefore a high resolution camera was simulated to minimize this effect. The profile consisted of a straight and level trajectory of 31 images taken every .5 seconds flown at 300 knots with a downward pointing camera as depicted in Figure 4.1. Figures 4.2 and 4.3 show the resulting errors for this ideal simulation.

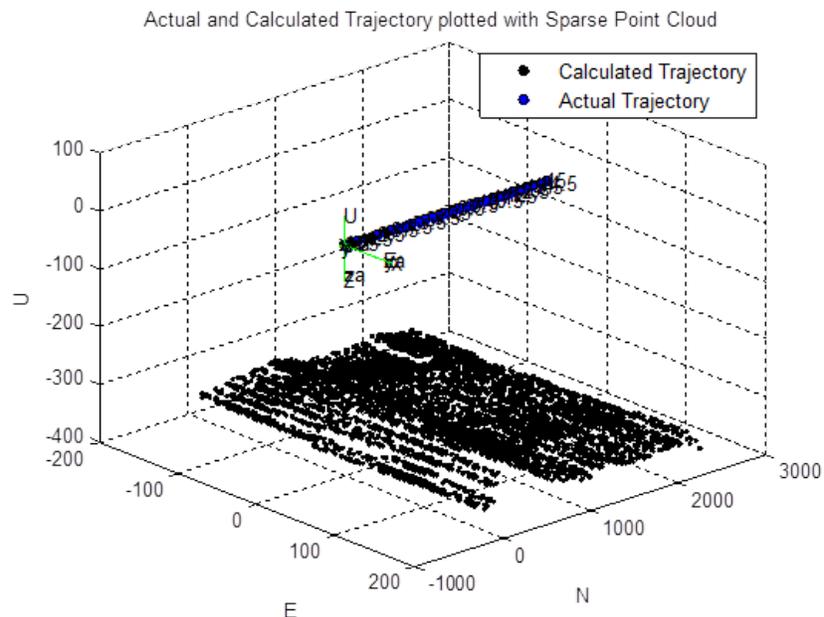


Figure 4.1: 3-D view of the actual and estimated trajectory along with the sparse point cloud

In this case, SFM estimated the focal length to be 6120.224 pixels for each camera instead of the actual value of 6675 pixels. The initial guess for focal length was 6675 pixels. The total radial error at the end of the 2310 meter trajectory is 1.2×10^{-3}

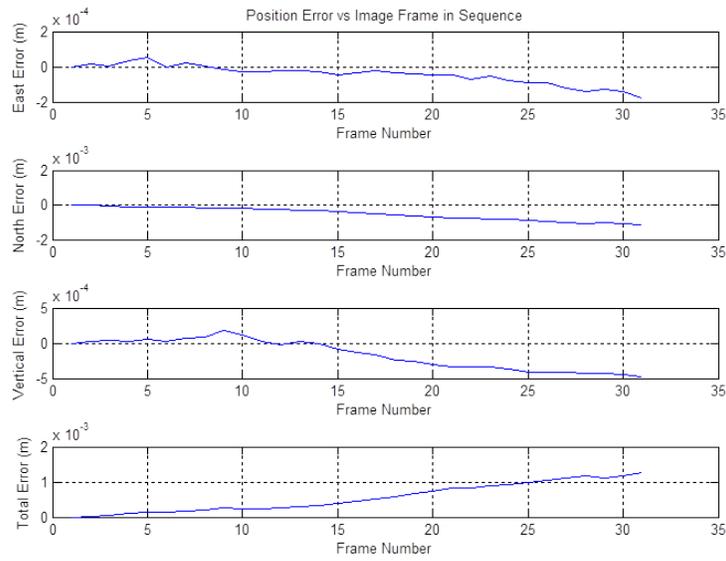


Figure 4.2: Un-constrained Sim Camera Position Errors - No Noise

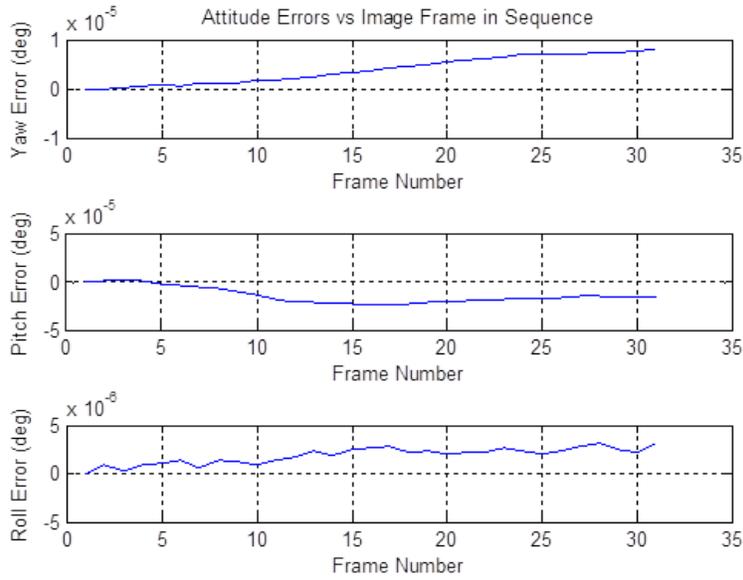


Figure 4.3: Un-constrained Sim Camera Attitude Errors - No Noise

meters which yields a drift rate of $.00033 \frac{m}{km^{1.5}}$. This drift is due to the small pixel

quantization error as well as estimation error in the numerical routines. Overall, this ideal simulation verifies that the methodology works and is implemented correctly.

4.2 Effect of Bundle Adjustment Constraints

The same straight and level simulated flight profile was used to test the method of constraining calibration, altitude and rotation in bundle adjustment. In order to illustrate the effects of the constraints, noise was added to the measured pixel locations of the features in each image taken by the camera. The noise was Gaussian distributed with zero mean and a standard deviation of .5 pixels. This approximates the amount of noise expected in the SIFT matching process after geometry constraints are applied to eliminate outlier matches. Since the images used are sequential and taken from the same camera at a relatively high frame rate, it is expected that sub-pixel registration is possible. If the images were from different cameras and non-sequential, the SIFT process would likely be less accurate and a higher standard deviation would be required to appropriately model the noise. Figures 4.4 and 4.5 show the effect of adding this noise to the same simulated profile introduced in the previous section.

Note the dramatic increase in error caused by the addition of pixel noise as opposed to the ideal, noiseless profile. In an attempt to improve this solution, the bundle adjustment was constrained with the known camera calibration parameters (focal length, principal point, pixel size). Figures 4.6 and 4.7 show the results.

As expected, constraining bundle adjustment with the known camera matrix significantly improved the accuracy in both attitude and position. It is clear that there is some un-observability between attitude errors and position errors since both the attitude and position for each camera is being estimated simultaneously. In other words, an error in the attitude estimate will affect the position estimate and it is impossible to decouple the two without further knowledge. Fortunately, as explained previously, most aircraft have an inertial measurement system to provide the yaw, pitch and roll of the aircraft to the pilot or flight control system. Therefore it is

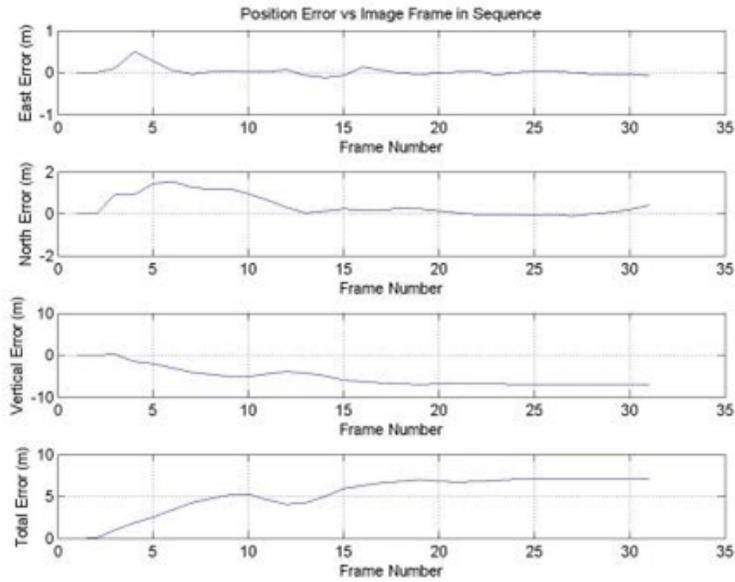


Figure 4.4: Un-constrained Position Errors - .5 Pixel Noise

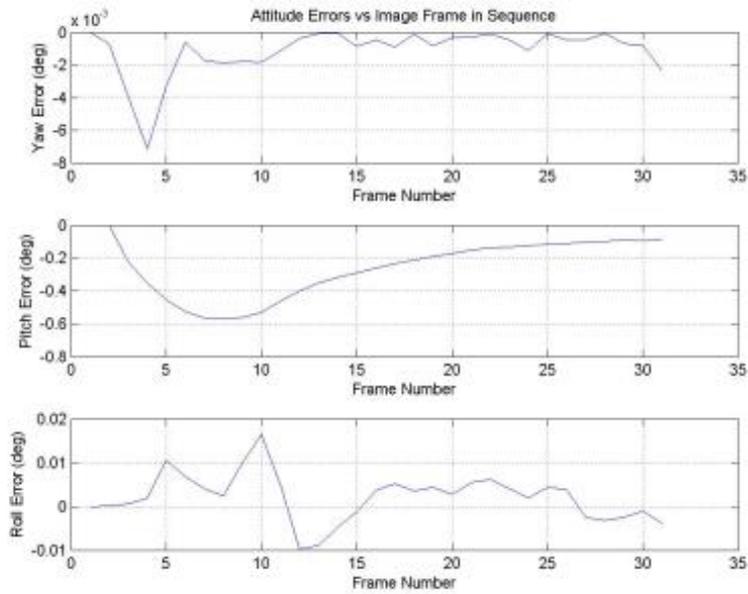


Figure 4.5: Un-constrained Attitude Errors - .5 Pixel Noise

possible to further constrain the bundle adjustment with the known rotations of each camera. In this case, the simulation assumes that yaw, pitch and roll are known

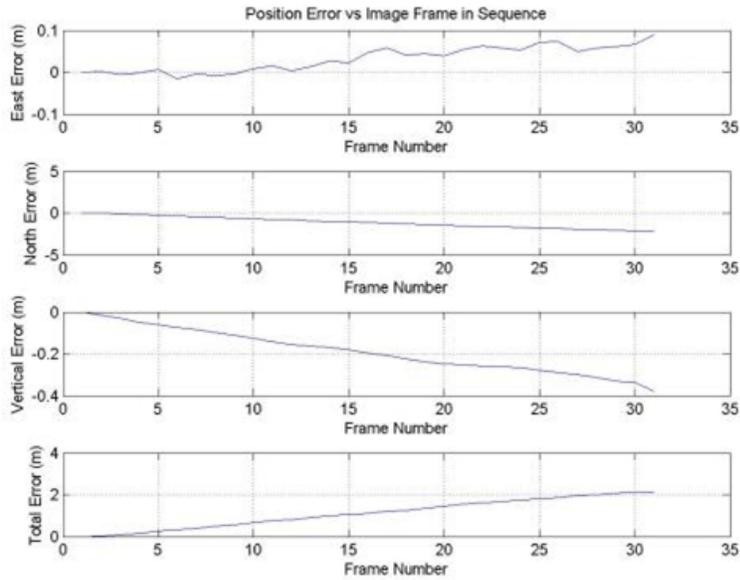


Figure 4.6: Calibration Constrained Position Errors - .5 Pixel Noise

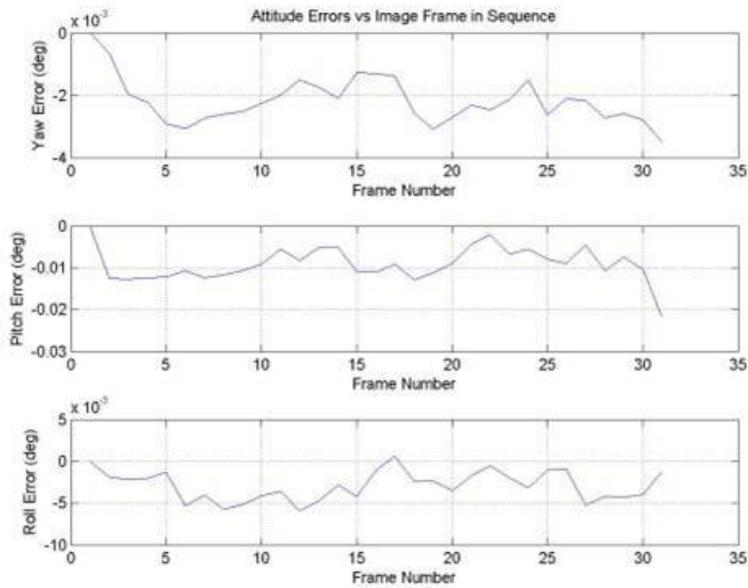


Figure 4.7: Calibration Constrained Attitude Errors - .5 Pixel Noise

perfectly (ie. no error in IMU data). Figures 4.8 and 4.9 show the result when the bundle adjustment is further constrained with known camera attitude.

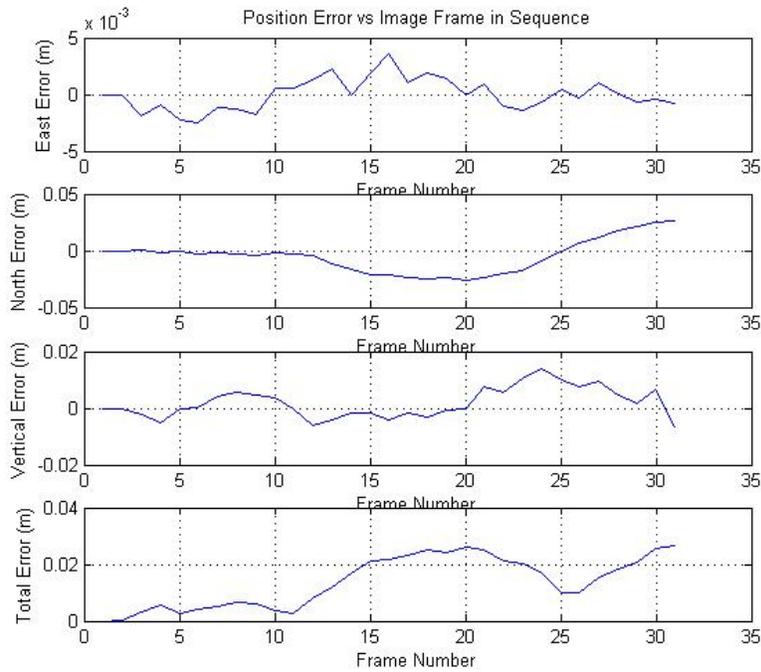


Figure 4.8: Calibration, Attitude Constrained Position Errors
 - .5 Pixel Noise

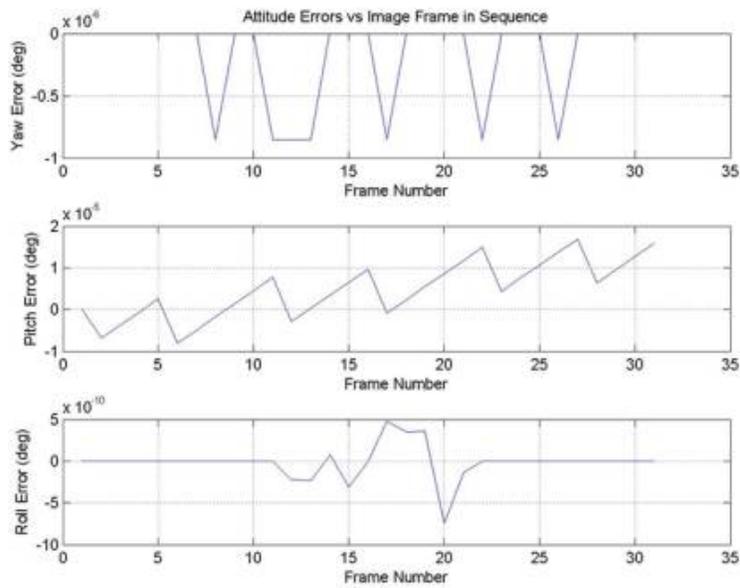


Figure 4.9: Calibration, Attitude Constrained Attitude Errors
 - .5 Pixel Noise

The effect of adding known rotation to the bundle adjustment was dramatic. The errors in attitude are now simply numerical errors and can be treated as zero. It is clear that the error in position in all axes has been reduced due to this added constraint. The estimation problem is now significantly constrained providing a much more accurate solution. It is now possible to add one more constraint: camera altitude as measured from an ideal barometric altimeter. Figures 4.10 and 4.11 show the result of adding the altitude constraint:

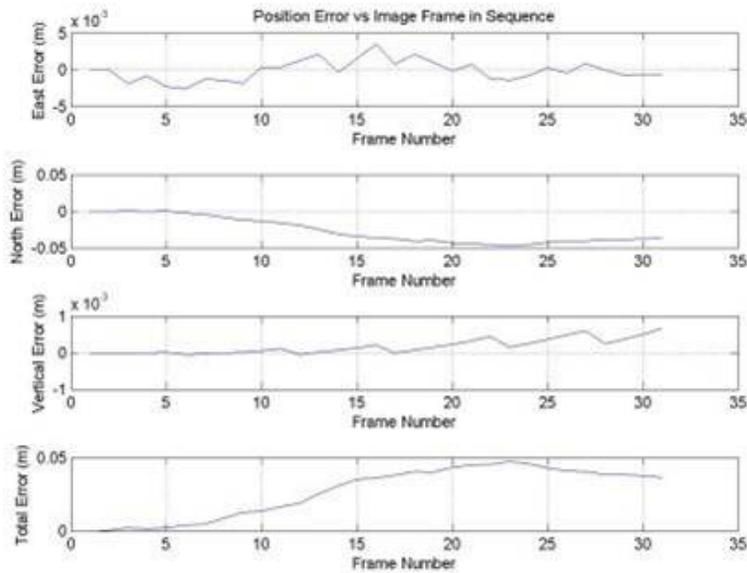


Figure 4.10: Calibration, Attitude, Altitude Constrained Position Errors - . 5 Pixel Noise

As expected, the error in the vertical axis was driven to zero (ignoring numerical errors) and, additionally, constraining altitude also improved the accuracy of the other axes although the effect was not as dramatic as the attitude constraint. Unfortunately, due to code limitations, it was only possible to implement this altitude constraint for the straight and level flight profile with a camera looking straight down. Therefore, for consistency, all further tests will only utilize the calibration and rotation constraints and not the altitude constraint.

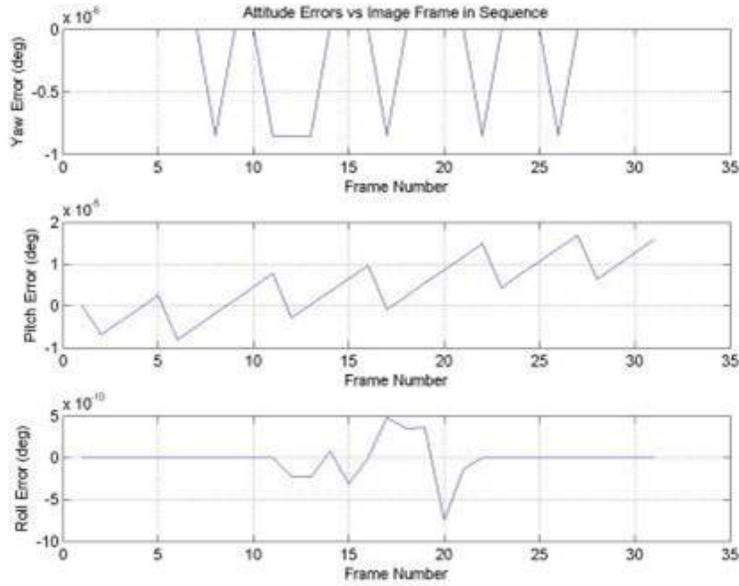


Figure 4.11: Calibration, Attitude, Altitude Constrained Attitude Errors - .5 Pixel Noise

The above figures are from one particular run of the straight and level trajectory with a particular set of randomly generated pixel noise for all the measurements. Since the pixel noise is random, in order to verify that the constraints are actually having a substantive effect and that this one run was not an outlier, multiple runs of the same test were done and random noise was generated for each set. Note that the noise level remains the same (ie. .5 pixel standard deviation) but each run has its own unique noise set. These simulations revealed that, in the presence of noise, the total position error of the final frame was random in both magnitude and direction. As predicted, the position errors of the last frame in each direction (East, North and Up) follow a Gaussian normal distribution. As an example, Figure 4.12 shows the error of the final frame in the North direction for each of the 100 simulator runs of the fully constrained case. Figure 4.13 shows a normal probability plot of the same data. The linear nature of the normal probability plot confirmed that the data followed a Gaussian, normal distribution [24].

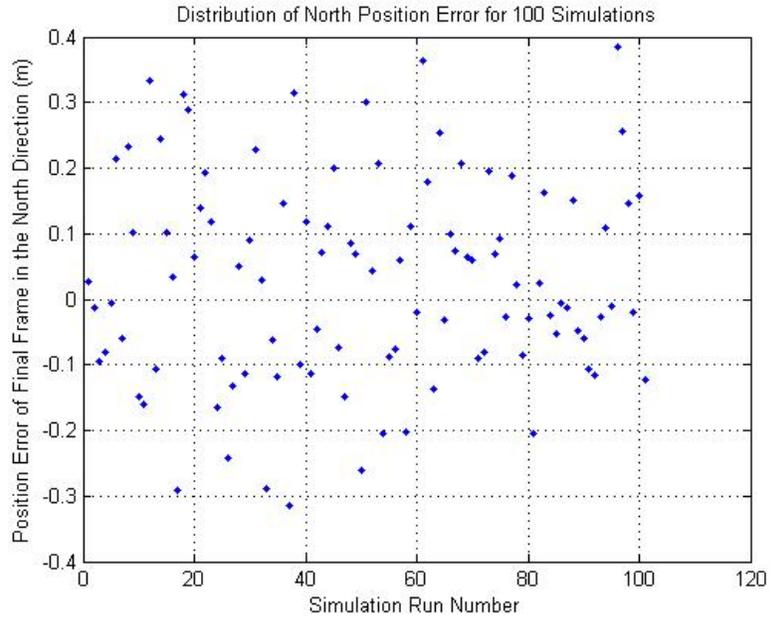


Figure 4.12: North error in final frame for each simulation run - .5 Pixel Noise

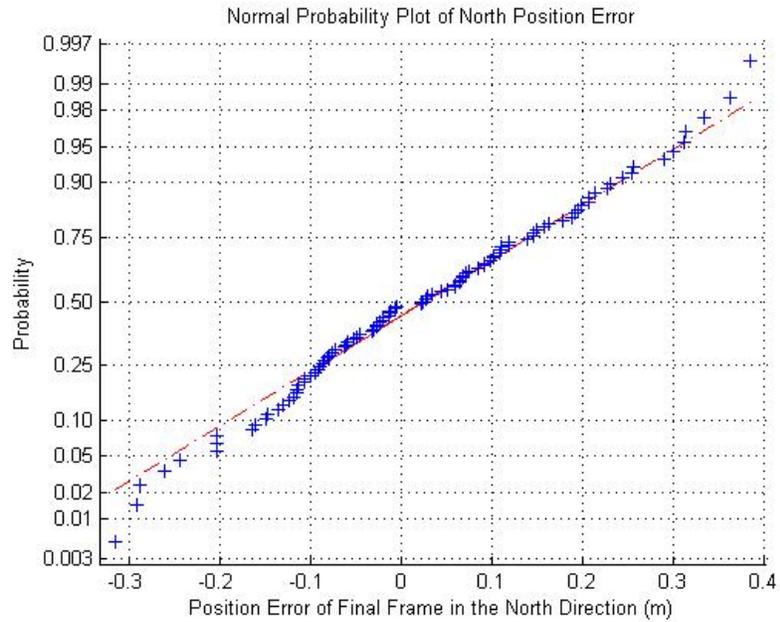


Figure 4.13: Normal Probability Plot of North error in final frame for each simulation run - .5 Pixel Noise

Since the position error at the final frame is random in magnitude and direction, the effect of various constraints on the algorithm can be seen by analyzing the statistics

from multiple simulation runs. The results for these simulations are summarized in Table 4.1.

Statistics for position error in the final image frame (all units are meters)										
Constraint	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
Unconstrained	-0.01500	0.11800	0.87000	1.00500	-8.71000	3.43000	8.84000	3.34000	3.57615	1.01859
Constrained Focal Length	0.01130	0.12700	0.11220	1.15000	0.11400	0.43500	1.06000	0.65400	1.23606	0.35207
Constrained Attitude	-0.02640	0.05330	1.39800	1.09500	-0.00470	0.16300	1.59800	0.78900	1.10835	0.31569
Constrained Focal Length and Attitude	0.00073	0.00400	0.02760	0.15610	-0.00110	0.01360	0.14400	0.11260	0.15674	0.04464

*Setup: 100 runs per configuration, Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise (Std Dev=.5 pixels), Focal Length: 6675 pixels, Image Size: 6800x6800 pixels

Table 4.1: Effect of BA Constraints on Solution Accuracy, .5 pixel noise

The table shows errors in the position of the final frame of the image sequence as constraints are progressively added. The first line represents 100 runs done with only the positions of the first two cameras used as constraints. This is termed the unconstrained case since the first two camera positions are always used as part of the algorithm. Next, the known focal length and camera attitude are added separately and, finally, all constraints are used together. The table shows both the mean and standard deviations of errors at the final frame for all 100 runs in each direction (East, North, Vertical). The columns labeled “Radial Error Mean” and “Radial Error Standard Deviation (STD)” are the mean and standard deviations of the 2-norm of the final error of each run while the column “Norm of E,N,V STDs” is the 2-norm of the standard deviations (East, North, Vertical) of all the runs combined. The norm of the East, North and Vertical standard deviations gives the best overall view of the expected error for any given trial. Therefore, the “Drift Rate Standard Deviation” column is the 2-norm of the East, North and Vertical standard deviations divided by the total distance flown raised to the power of 1.5 to give a sense of the standard deviation of the drift rate for a given trial. This number was chosen as the primary metric because it gives a sense of how much, in total, the trajectory will drift given a certain set of parameters.

The drift rate standard deviation for a particular simulation set was compared to the drift rate standard deviation of each of the other sets using an F-test to determine if there was a statistically significant difference between any two simulation

sets. For a complete description of using the F-test to compare the variances of two populations see [24]. This analysis showed with a 95% certainty that the drift rate standard deviation with constraints applied was less than the drift rate standard deviation without constraints applied. Also, with 95% certainty, the drift rate standard deviation with both constraints applied was less than the drift rate with only one constraint applied. It also appeared that constraining attitude had a larger effect than constraining focal length but this conclusion can only be stated with 85% certainty.

4.3 Predicting Camera Position Errors

Algorithm A6.4 from [15] was implemented to try to predict the accuracy of the fully constrained bundle adjustment process (ie. camera orientation and focal length known). The algorithm outputs a covariance matrix for each camera in the set and this covariance matrix can be expressed in the East, North, Up frame so that a variance for the camera position error in each direction can be obtained and compared to simulation results. The diagonal terms of this covariance matrix represent the variances in the East, North and Up directions and the total variance is the 2-norm of these three values. Figure 4.14 shows the predicted standard deviation of the position estimates of the cameras when the measurement covariance matrix is Identity for the straight and level test trajectory. The Identity measurement matrix corresponds to a pixel noise of variance 1 pixel. Since the measurement noise is assumed to be zero mean, Gaussian and uncorrelated, then the resulting variance estimates for any other value of measurement noise input to this trajectory are simply scalar multiples of the below plot.

These predicted errors reveal a number of interesting trends. First, note that the uncertainty in the first two cameras is zero, as expected, since these cameras are constrained with their actual positions during the final bundle adjustment. Also note that in the North direction, the standard deviation of the error in the camera positions increases proportional to distance raised to the power of 1.5. This prediction agrees with the behavior predicted by the scale error equations derived in Chapter 3. In the

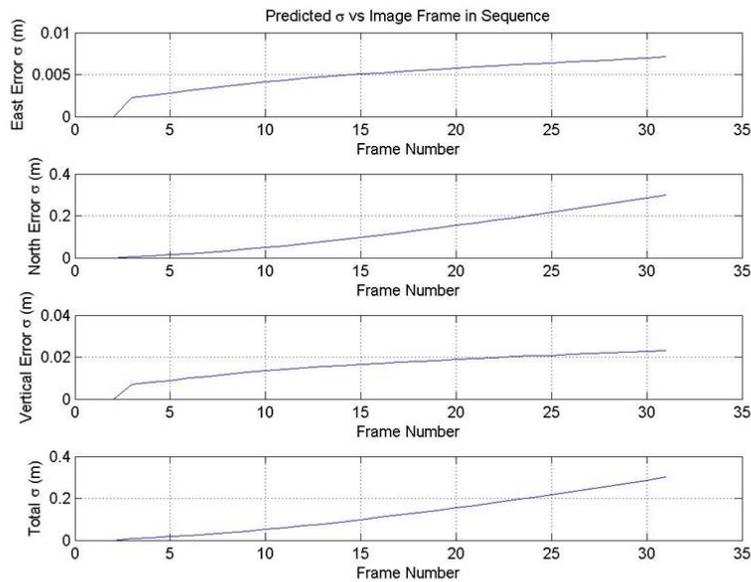


Figure 4.14: Predicted camera position errors when the measurement covariance matrix is identity (1 pixel noise). Straight and level trajectory with downward looking camera.

East and Vertical directions, there is no movement away from the origin with each successive camera so the scale error equation does not apply. Instead, since frames are added sequentially in the bundle, the final error is the result of a random walk and is proportional to \sqrt{N} , where N is the number of frames.

Figures 4.15, 4.16, 4.17 show the standard deviations of the actual East, North and Vertical error at each frame for 100 runs of the fully constrained algorithm. The error trends and magnitudes closely match the predicted results above and the trends predicted by the error equations developed in the previous chapter. The data are fit with curves in the form of the error equations.

This result validates the form of the error equations developed in Chapter 3. It shows that the error in the direction of travel is dominated by scale error and increases proportional to $x^{1.5}$.

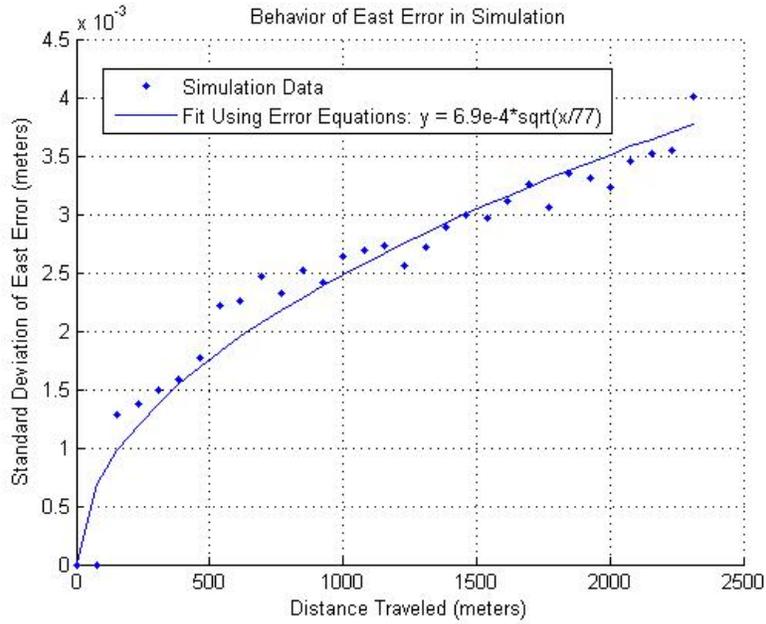


Figure 4.15: Actual East position errors from a Northbound, straight and level trajectory with downward looking camera.

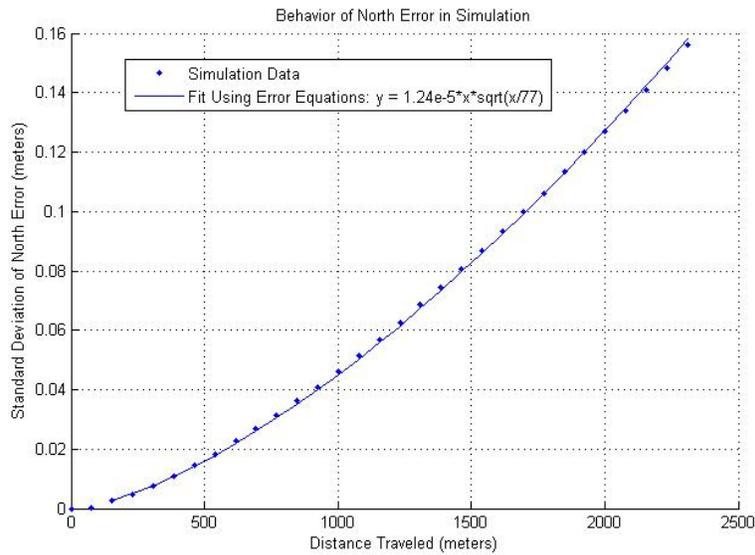


Figure 4.16: Actual North position errors from a Northbound, straight and level trajectory with downward looking camera.

4.4 Effect of Pixel Noise

Even though SIFT does a good job of matching sequential images to a sub-pixel level, it is still useful to examine the effect of increasing the standard deviation of

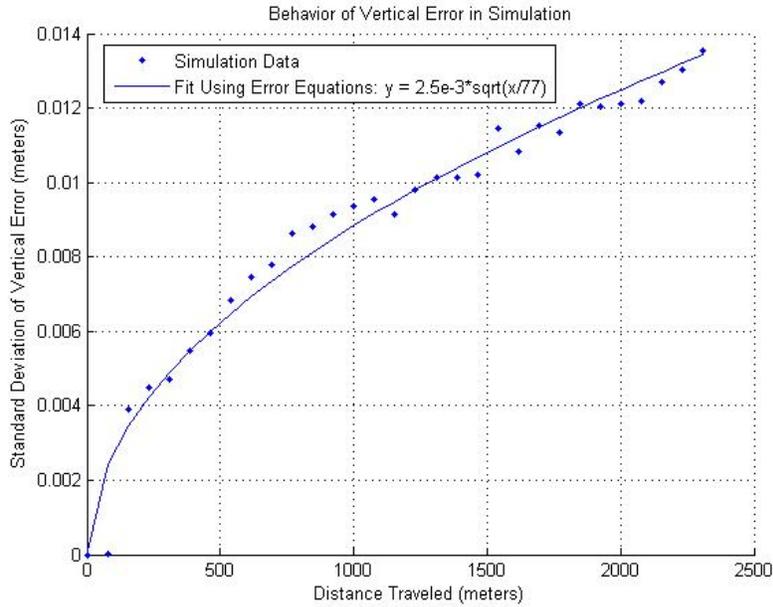


Figure 4.17: Actual Vertical position errors from a North-bound, straight and level trajectory with downward looking camera.

the pixel noise injected into the simulation. Table 4.2 shows the results of simulation runs with a pixel noise standard deviation .5 pixel and 1 pixel.

Standard Deviation (STD) of Pixel Noise	East Error Mean	East Error STD	Predicted East Error STD	North Error Mean	North Error STD	Predicted North Error STD	Vertical Error Mean	Vertical Error STD	Predicted Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Predicted Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
Noise STD: .5 Pixel (100 runs)	0.00073	0.00400	0.00350	0.02760	0.15610	0.15000	-0.00110	0.01360	0.01200	0.13000	0.09090	0.15674	0.15052	0.04464
Noise STD: 1 Pixel (10 runs)	-0.00120	0.00870	0.00700	0.24750	0.42400	0.30000	0.00830	0.04090	0.02400	0.35730	0.32950	0.42606	0.30104	0.12135

*Straight/Level Flight with Fixed Downward Pointing Camera, GSD=.054m/pixel, Altitude=356m, Focal Length=6675 pixels, Image Size: 6800x6800 pixels

Table 4.2: Effect of Pixel Noise Level on Solution Accuracy

These data show a clear and near linear relationship between the standard deviation of the pixel noise added to the feature location measurements and the amount of drift in the solution. An F-Test confirms, with 95% certainty, a statistically significant difference in drift rates between the two simulation sets. Clearly, more noise in feature matching leads to less accurate estimation of the fundamental matrices which in turn leads to less accurate position estimates. The predicted error for each axis is listed in Table 4.2 and was calculated with the techniques described earlier. Note that the predictions follow the appropriate trend of increasing linearly with pixel

noise. Also note that, as predicted by the scale error equations, the standard deviation of the error is largest in the direction of travel (North in this case).

4.5 Effect of Resolution, Angular Separation and Image Overlap

Theory predicts that the nature of the Jacobian matrix in the bundle adjustment will have a strong effect on the final navigation solution error. The nature of the Jacobian matrix is determined by a complex interaction of image resolution, camera angular separation and image overlap as discussed in Chapter 3. These factors are in turn governed by the camera pixel size, image plane size, camera focal length, camera mounting angles, camera frame rate, aircraft altitude, aircraft speed, aircraft trajectory and terrain overflow. The effects of image resolution, angular separation and image overlap were studied individually by isolating each effect and running multiple simulations. The simulations were all run with the same simulated feature set to ensure that feature geometry was not a factor. Each simulation was run using white measurement noise with a .5 pixel standard deviation. For each run the total number of features viewed in an image was kept constant between 800-1300 features. The straight and level profile at various altitudes and with various camera intrinsic parameters was first utilized to understand the effects of resolution, angular separation and image overlap on the solution. This allowed the use of the equations developed in Chapter 3 to determine the GSD, maximum angular separation and image overlap for a particular test. Various trajectories and camera mountings were then simulated to show how different trajectories and camera mountings can affect resolution, angular separation and image overlap and hence can change the solution results.

4.5.1 Effect of Image Resolution. The effect of image resolution was isolated by running several simulations of the straight and level trajectory at the same altitude while varying only camera focal length and camera field of view to change GSD while keeping constant the amount of overlap between successive images. The constant altitude, speed and frame rate between simulations meant that the maximum

angular separation between sequential images was also constant. It was expected that increased GSD would lead to less accurate reconstructions since the size of a given pixel is increasing. Increased pixel size increases quantization error and the same amount of pixel noise will correspond to higher error in meters. The prediction was confirmed by the results reported in Table 4.3

Statistics for position error in the final frame (all units are meters)														
Test Configuration	East Error Mean	East Error STD	Predicted East Error STD	North Error Mean	North Error STD	Predicted North Error STD	Vertical Error Mean	Vertical Error STD	Predicted Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Predicted Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{0.15})
GSD=.054 m/pixel (100 runs)	0.00073	0.00400	0.00350	0.02760	0.15610	0.15000	-0.00110	0.01360	0.01200	0.13000	0.09090	0.15674	0.15052	0.04464
GSD=.278 m/pixel (10 runs)	0.01300	0.01390	0.01700	-0.00640	0.91330	0.68700	0.00520	0.06570	0.05500	0.62380	0.64100	0.91577	0.68941	0.26084
GSD=.618 m/pixel (100 runs, 36% successful)	0.13040	0.81470	0.04200	-0.48460	2.63450	1.81000	0.00520	0.33160	0.14000	1.92040	2.04400	2.77746	1.81589	0.79110

*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise with Std Dev = .5 Pixel, 356 meters altitude, 79% overlap and 12.4° angular separation between each frame

Table 4.3: Effect of GSD on Solution Accuracy, .5 pixel noise

There is a clear, near linear relationship that is statistically significant between the predicted/observed error and the value of GSD. It is therefore likely that GSD is a major and dominant source of error in the trajectory reconstruction. Once again the error in the direction of travel (North) is the largest error due to scale effects. Note that the reconstructions with a GSD of .618 meter/pixel only had a 36% success rate. In this case the reported error statistics apply only to the successful reconstructions. The low success rate is likely due to the increased GSD combined with only 1000 features per images and 78% overlap. Reconstructions likely failed because there were too few features to effectively calculate fundamental matrices in the presence of noise with this higher GSD. The effect of increasing the number of overlapping features with a constant GSD will be discussed in the next section.

4.5.2 Effect of Image Overlap. The effect of image overlap was isolated by running simulations of the straight and level trajectory but only changing camera field of view by increasing the number of pixels on the image plane while keeping pixel size and GSD constant. It was expected that increased overlap between sequential images would lead to more accurate reconstructions and less overall drift. Two different camera footprint sizes were run and the results are summarized in Table 4.4.

The results show a four fold increase in both predicted and observed error for a 12% decrease in overlap between sequential images. Additionally, reconstruction

Statistics for position error in the final frame (all units are meters)																	
Test Configuration	East Error		Predicted East Error		North Error		Predicted North Error		Vertical Error		Predicted Vertical Error		Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Predicted Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km*1.5)
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD							
Image Overlap = 90.8% (99° FOV) (10 runs, 100% successful)	-0.01650	0.01710	0.01800	0.28200	0.47300	0.46300	0.09920	0.02650	0.02500	0.42390	0.47403	0.46303	0.1352				
Image Overlap = 78.4% (53° FOV) (100 runs, 36% successful)	0.13040	0.81470	0.04200	-0.48460	2.63450	1.81000	0.00520	0.33160	0.14000	1.92040	2.04400	2.77746	1.81589	0.79110			

Table 4.4: Effect of Image Overlap on Solution Accuracy, .5 pixel noise

reliability increased from a 36% to 100% success rate with increased overlap. In this situation, the 12% increase in overlap corresponds to doubling the camera field of view. Note a non-linear relationship between the total area of the camera ground footprint and the drift rate although only two test points were run due to system and time limitations. The dramatic increase in accuracy and reliability is expected since increased overlap between images leads to more matches between any given image pair. It is important to note that in this test the total number of features in each image was held constant and features were uniformly distributed throughout the image. Even though the number of features in each image stays the same, the number of matches between a given image pair increases due to increased overlap. Additionally, the matching features are more widely distributed throughout the images leading to a better conditioned Jacobian that reduces ambiguity between camera rotation and translation. Both of these factors lead to better estimation of the fundamental matrix and a more accurate camera position estimate. Also, note that these two simulations were run with a high GSD (.62 meters/pixel). There may be an interaction between GSD and image overlap so the results of this same test might change with different GSD. These interactions will be discussed shortly.

In addition to increasing the number of matches for a given image pair, the number of camera pairs with shared features increases. This also improves accuracy and it is not known whether the effect of more matching cameras or more matches between two sequential cameras is the driving factor in error. This can be tested by only allowing matching between sequential images as oppose to allowing any image to match to any other image in the set (sequential matching vs full matching). A limitation with the setup of the simulation did not allow for this restriction to be

implemented; however this restriction was implemented on select real world data sets and no change between sequential matching and full matching was observed although its effect may have been masked by other, larger errors in the real data. Further testing of this is recommended in future research as sequential matching significantly speeds up the reconstruction process because less image pairs need to be matched and feature matching requires significant processing time. Overall, increased overlap allows for more matches between image pairs and therefore less error.

4.5.3 Effect of Camera Angular Separation. The effect of camera angular separation was isolated by running simulations of the straight and level trajectory at varying altitudes while changing camera focal length to maintain constant image resolution and overlap. The speed of the aircraft and the camera frame rate were not changed so increasing altitude corresponded directly to decreasing angular separation between sequential image frames relative to the terrain below. It was expected that reduced angular separation would decrease accuracy but that the effects might be highly non-linear and directional. Table 4.5 shows the results of the simulation and the associated predictions.

Statistics for position error in the final frame (all units are meters)														
Test Configuration	East Error Mean	East Error STD	Predicted East Error STD	North Error Mean	North Error STD	Predicted North Error STD	Vertical Error Mean	Vertical Error STD	Predicted Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Predicted Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
Alt = 356m, Angular Separation=12.4° (100 runs)	0.00073	0.00400	0.00350	0.02760	0.15610	0.15000	-0.00110	0.01360	0.01200	0.13000	0.09090	0.15674	0.15052	0.04464
Alt = 856m, Angular Separation=5.2° (10 runs)	-0.00055	0.00320	0.00340	0.04340	0.14030	0.13620	-0.01140	0.02500	0.02800	0.11310	0.09200	0.14255	0.13909	0.04060
Alt = 1856m, Angular Separation=2.4° (10 runs)	-0.00150	0.00420	0.00340	0.03060	0.16690	0.12700	0.01760	0.08040	0.05970	0.14400	0.11260	0.18530	0.14037	0.05278
Alt = 4156m, Angular Separation=1.1° (10 runs)	Failed Reconstructions													

*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise with Std Dev = .5 Pixel, 79% Overlap, GSD=.054m/pixel

Table 4.5: Effect of Camera Angular Separation on Solution Accuracy, .5 pixel noise

The results suggest that there is no statistically significant difference in total drift rate with angle. However, further insight is gained by looking at the error in the individual directions: East, North and Vertical. As expected, the North error is dominant due to scale effects and follows the same trend as the total drift; however, the North error was predicted to continually decrease with angle and it does not. There was no predicted change in the East error and there was no statistically significant change in East error observed. The Vertical error was predicted to continually

increase as angular separation decreases and the observations follow this prediction. This error behavior can be explained using Figure 3.10. In a situation when the cameras are looking straight down, decreasing angular separation continually increases uncertainty in the vertical axis. In this geometry the same decreasing angular separation may have no effect or a non-linear effect on error in other axes. Also note that reconstructions attempted at 1.1° failed altogether. This suggests a minimum angular separation of $1 - 2^\circ$ for a successful reconstruction in this case. However, there is an interaction between angular separation and GSD as higher GSD values led to failed reconstructions at higher angles. For a GSD of .278 meters/pixel, only 60% of the reconstructions were successful at 2.4° and for a GSD of .618 meters/pixel, reconstructions below 5.2° were not possible. Additionally, when overlap was increased from 78% to 98%, reconstructions were possible down to 2.4° with a GSD of .618 meters/pixel. The success rate of reconstructions was therefore not simply dependent on angular separation but was dependent on the interaction of angular separation, GSD and overlap.

This angular effect on reconstruction success poses a problem for high altitude flight. It can be overcome by using larger distances between cameras at higher altitudes to increase angular separation; however this means that the update rate of any navigation system will be slower since more time is required between images. Additionally larger baseline distances may not be practical in a stereo camera system. In an attempt to overcome these problems, a method was successfully developed to allow for successful reconstructions at very low angular separations $< 1^\circ$. Since the initial reconstruction can be performed in any scale, a smaller scale was chosen that allows for sufficient angular separation. Once the reconstruction was completed on this smaller scale, the result was multiplied by the appropriate scale factor to return to the real world scale. This was accomplished by constraining the bundle adjustment with an artificially low focal length. For example, a scene viewed by two cameras separated by 100 feet with focal lengths of 10,000 pixels at an altitude of 10,000 feet is the same as a scene reconstructed by cameras separated by 100 feet with focal

lengths of 1,000 pixels at an altitude of 1,000 feet. In other words, the scene GSD and image overlap have not changed because the decrease in altitude is offset by the decrease in focal length. The only difference is that the angular separation at the lower altitude is larger than the angular separation at the higher altitude. This larger angular separation may allow for a successful reconstruction at the lower altitude and focal length. The successful reconstruction can then be scaled back to the original scale. This process was attempted for the fourth case in Table 4.5, the case of a 1.1° that originally failed. The focal length was artificially constrained to 1,500 pixels instead of the actual value of 6675 pixels resulting in an artificial maximum angular separation of 175° . This method allowed for a successful reconstruction. Table 4.6 shows the resulting error statistics as compared to the other three cases.

Statistics for position error in the final frame (all units are meters)															
Test Configuration	East Error		Predicted East Error		North Error		Predicted North Error		Vertical Error		Predicted Vertical Error		Norm of E,N,V STDs	Predicted Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km*1.5)
	Mean	STD	East Error STD	North Error Mean	North Error STD	Predicted North Error Mean	Predicted North Error STD	Vertical Error Mean	Vertical Error STD	Predicted Vertical Error Mean	Predicted Vertical Error STD				
Alt = 356m, Angular Separation=12.4° (100 runs)	0.00073	0.00400	0.00350	0.02760	0.15610	0.15000	-0.00110	0.01360	0.01200	0.13000	0.09090	0.15674	0.15052	0.04464	
Alt = 856m, Angular Separation=5.2° (10 runs)	-0.00055	0.00320	0.00340	0.04340	0.14030	0.13620	-0.01140	0.02500	0.02800	0.11310	0.09200	0.14255	0.13909	0.04060	
Alt = 1856m, Angular Separation=2.4° (10 runs)	-0.00150	0.00420	0.00340	0.03060	0.16690	0.12700	0.01760	0.08040	0.05970	0.14400	0.11260	0.18530	0.14037	0.05278	
Alt = 4156m, Angular Separation=1.1°, Artificially Constrained to 175° (10 runs)	0.00052	0.00590		0.03960	0.15380		0.00120	0.00530		0.12990	0.08290	0.15400		0.04386	

*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise with Std Dev = 5 Pixel, 79% Overlap, GSD=0.54m/pixel

Table 4.6: Effect of Camera Angular Separation on Solution Accuracy, .5 pixel noise

There was no statistically significant difference between drift rate of the final case with any of the other cases. In fact, the Vertical error of the final case showed a significant decrease as compared to the other cases. This was expected since the maximum triangulation angle was 175° meaning that the uncertainty in the vertical axis had significantly decreased. It will be shown later with further simulation that constraining the bundle adjustment with an artificial focal length that was lower than the actual focal length had no statistically significant adverse effect on drift rate. This is therefore a viable method of generating successful reconstructions at very small angular separations. Flight at high altitudes and high camera frame rates will require this type of method as these situations will lead to low triangulation angles between frames.

4.5.4 Interactions between Image Resolution, Angular Separation and GSD.

4.5.4.1 Altitude Effects.

In order to study the interactions between image resolution, camera angular separation and image overlap as altitude changes, the straight and level profile was flown four times at different altitudes with the same camera. The camera parameters and camera frame rate were not changed for each run so the ground sample distance (GSD), camera angular separation and image overlap all changed solely as a result of increasing altitude. According to the pinhole camera model, an increase in altitude causes an increase in GSD as well as a decrease in camera angular separation relative the terrain features. Increasing GSD decreases accuracy while the decrease in angle was shown to not effect accuracy so long as the angle is large enough for a successful reconstruction. However, an increase in altitude also increases image footprint and overlap which increases accuracy. Table 4.7 shows both the predicted and observed errors as altitude was increased while GSD, angular separation and footprint were allowed to vary accordingly.

Statistics for position error in the final frame (all units are meters)														
Test Configuration	East Error	East Error	Predicted	North Error	North Error	Predicted	Vertical Error	Vertical Error	Predicted	Radial Error	Radial Error	Norm of	Predicted	Drift Rate
	Mean	STD	East Error	Mean	STD	North Error	Mean	STD	Vertical Error	Mean	STD	E,N,V STDs	Norm of	Standard Deviation
Alt=336m, GSD=054m/pix, Overlap=78.7%, Angular Sep=12.4°, 100 runs	0.00073	0.00400	0.00350	0.02760	0.15610	0.15000	-0.00110	0.01360	0.01200	0.13000	0.09090	0.15676	0.15052	0.04464
Alt=856m, GSD=128m/pix, Overlap=91.2%, Angular Sep=5.2°, 10 runs	-0.00008	0.00190	0.00190	-0.01440	0.08790	0.09800	0.00170	0.01760	0.01320	0.07530	0.04480	0.08963	0.08996	0.02553
Alt=1856m, GSD=278m/pix, Overlap=95.9%, Angular Sep=2.4°, 10 runs	-0.00024	0.00440	0.00360	0.01690	0.21820	0.16340	0.00100	0.02390	0.01760	0.17500	0.12010	0.21938	0.16343	0.06249
Alt=4156m, GSD=618m/pix, Overlap=88.2%, Angular Sep=1.1° (constrained to 21°), 10 runs	-0.00540	0.00530	0.01200	0.71110	0.39930	0.19900	-0.00470	0.00900	0.03300	0.71130	0.39910	0.39944	0.20207	0.11377

Table 4.7: Effect of Altitude on Solution Accuracy (constant camera parameters), .5 pixel noise

The error changes observed in these tests were a combination of GSD effects and footprint effects. As altitude increased, both GSD and overlap increased meaning that GSD error effects and overlap error effects were working in opposite directions. The only statistically significant change in overall drift rate was between the trajectory flown at 4156 meters and the trajectory flown at 1856 meters. This suggests that, for large altitude changes, the effect of GSD dominates the effect of overlap in this set of simulations but that the changes in overlap and GSD tend to balance each other out for small altitude changes. It is therefore important to note that when the camera parameters are unchanged, flying at higher altitudes may provide less accurate nav-

igation results. This certainly has operational implications for any future navigation system but this effect may be mitigated by changing camera parameters as altitude is increased. From a navigation perspective, systems that minimize GSD while maximizing footprint are best suited for accuracy but clearly require more processing power, time, system cost and complexity. A tradeoff between speed and accuracy will be necessary for any future system. Additionally, adaptive focal lengths and fields of view may be necessary to compensate for the changes in GSD and footprint that arise with altitude if consistent performance is required across a wide range of altitudes.

4.5.4.2 *Camera Mounting Variations.* In order to study the interactions between image resolution, camera angular separation and image overlap with changes in camera mounting, the straight and level flight trajectory was flown with five different camera mountings: downward, right, left, forward and backward. The forward, back, right and left cameras were all 20° up from the vertical or, equivalently, 70° degrees down from the aircraft local level and pointed in the specified direction with respect to the aircraft body frame. For example the left camera position is found by starting with the camera frame in the straight downward position and then rotating the camera 90° left and 20° up in that order. Higher camera angles (> 20°) were not practical due to simulation limitations. Table 4.8 shows results from the five test cases.

Statistics for position error in the final image frame (all units are meters)										
Camera Line of Sight Mounting	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
Down	-0.00015	0.00420	-0.00170	0.11640	-0.00025	0.00150	0.09110	0.06660	0.11649	0.03318
20° Right	-0.00022	0.00450	-0.06210	0.14110	-0.00069	0.00150	0.12900	0.07670	0.14118	0.04021
20° Left	-0.00009	0.00340	-0.03550	0.14890	-0.00009	0.00130	0.11860	0.09000	0.14894	0.04242
20° Forward	0.05960	0.10750	0.74370	1.05000	1.24000	0.34780	1.65990	0.70360	1.11132	0.31653
20° Backward	-0.07350	0.07760	0.12380	0.84820	-0.44550	0.22040	0.89310	0.37130	0.87980	0.25059

*Setup: 10 runs per configuration, Straight/Level Flight, Altitude=356m, focal length=6675 pixels, Image Size: 6800x6800 pixels, Focal Length Artificially Constrained to 667.5 pixels

Table 4.8: Effect of Camera Mounting on Solution Accuracy, .5 pixel noise

Simulation showed that there was no statistically significant change in drift between the downward camera and the side cameras nor between the forward and backward cameras (at least for small mounting angle changes). There was a statically

significant difference between the side/down cameras and the forward/backward cameras. Additionally, these reconstruction were made by reducing the focal length by a factor of 10. This was required because the forward and backward camera cases failed to reconstruct due to lack of angular separation. Angular separation is reduced when cameras look forward or aft relative to the trajectory. The increase in error for the forward/backward cases was likely due the decreased footprint overlap for forward/aft facing cameras relative to side facing cameras.

The lack of error change between the down and side cameras is likely due to the interaction of GSD and overlap since both GSD and overlap increase for side mounted cameras relative to a downward mounted camera. These results only apply to small mounting angles. Even though extreme cases we not able to be simulated, the simulation results suggest that a camera pointing near straight down is useable for navigation and that a camera pointing slightly forward or aft may degrade algorithm performance.

4.5.4.3 Trajectory Variations. The effect of aircraft trajectory on reconstruction accuracy was tested using four flight profiles with a downward looking camera: straight and level, 25° level turn, 10° straight climb, 25° turn with a simultaneous 10° climb. Due to software limitations these trajectories were reconstructed without constraining camera orientation. The results are summarized in Table 4.9.

Statistics for position error in the final image frame (all units are meters)										
Trajectory	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
Straight/Level	0.0113	0.1270	0.1122	1.1500	0.1140	0.4350	1.0600	0.6540	1.2361	0.35207
Straight Climb (10° Pitch)	0.0154	0.1219	0.1537	1.5556	0.0480	0.6844	1.4080	0.9621	1.7039	0.48530
Level 25° Bank Turn	0.0946	0.3307	0.0772	1.3845	0.1325	0.4958	1.2958	0.7803	1.5073	0.42933
Climbing Turn (25° Bank, 10° Pitch)	0.0115	0.3249	-0.0081	1.5436	0.0383	0.6934	1.3228	1.0970	1.7231	0.49078

*Setup: 100 runs per configuration, Starting Altitude=356m, Focal Length=6675 pixels, Random Pixel Noise with Std Dev = .5 Pixel, Un-constrained Rotation

Table 4.9: Effect of Trajectory on Solution Accuracy, .5 pixel noise

As expected, trajectories that incorporate a climbing maneuver showed increased drift rates most likely due to the change in altitude and resulting changes in GSD and footprint. In this case the 25° level turn performed worse than the

straight and level profile. This may be due to the increased range to features caused by the aircraft bank; however, this begs the question: why did the turn perform worse than the level case but the 20° sideways mounted camera had the same drift rate as the level case? A possible answer to this question is that image overlap is related to both angular rate as well as angle. The sideways mounted camera had a constant angle so overlap and GSD were constant. In the case of the sideways camera, that overlap/GSD relationship stayed constant and produced the same error as the downward looking camera. During the level turn there was a changing roll rate as the aircraft rolled into the turn and then a constant heading rate as the aircraft continued the turn at a constant bank. This led to continual variations in GSD and overlap making it more likely that error will change relative to the straight and level downward looking camera.

Finally, notice the increased error in the East direction for both the turning trajectories and the increase in error in the vertical direction for both climbing trajectories. These increases are due to scale error. In the turning trajectories the aircraft moves East and gets farther away from the origin in the Easterly direction. In the climbing trajectories the aircraft moves up and gets farther away from the origin in the vertical direction. Recall that scale error is proportional to the distance from the origin so these results match the expected trend.

Overall, trajectory variations that increase altitude and include banking maneuvers can cause significant variability in error due to the complex interactions between GSD and image overlap. Additionally, trajectories that decrease angular separation may cause the reconstruction to fail depending on the GSD and overlap. These result suggests that the algorithm performs best in a straight and level trajectory with downward looking camera.

4.6 Focal Length Noise

Thus far it has been assumed that the focal length of the camera is known perfectly and is used to constrain the bundle adjustment. In reality, there will be

some noise in the calibration that is used to constrain the bundle adjustment. Several methods exist to calibrate airborne cameras. The first option is to do a manual calibration with a checkerboard prior to flight. This method may be limited if the focal length of the camera is very long (ie. optimized for high altitude imagery). In this case, the distance between the checkerboard and the camera will be prohibitively long for ground operations. In flight calibration can be done by comparing images to known surveyed landmarks and using the information about camera position from a separate positioning source (ie. GPS) to back out calibration parameters. Without surveyed points, bundle adjustment can be used to determine camera calibration given known camera positions from a separate positioning source. This technique was evaluated by [21] with promising results. Finally, the camera calibration can be simultaneously estimated within the navigation algorithm but this was already shown to degrade the navigation solution. As an additional challenge, calibration parameters may change during flight due to thermal and vibration effects. However, since the time periods in each bundle used in the navigation algorithm are short, it is assumed that the calibration will not significantly change over that short time period and any inaccuracies in calibrated focal length can be modeled as a constant bias for each image.

In order to examine the effects of noise in the focal length constraint, simulations were run with varying levels of focal length noise. Table 4.10 summarizes the results of simulations run after injecting noise into the focal length constraint. These simulations constrained the solution using the inaccurate focal length and the accurate camera orientation parameters. Note that all cameras in the bundle had the same actual focal length and were constrained with the same incorrect focal length.

There was no statistically significant degradation observed as focal length errors were introduced except when the focal length became too large and the reconstruction failed. The explanation for this behavior is that a constant focal length bias should only change the SFM process by a constant scale factor that is later removed in the transformation to real world scale. The scale factor has no effect on the error unless

Statistics for position error in the final image frame (all units are meters)										
Focal Length Noise Standard Deviation	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
Correct Focal Length (No Error) (100 runs)	0.0007	0.0040	0.0276	0.1561	-0.0011	0.0136	0.1440	0.1126	0.1567	0.04464
1000 pixels (10 runs)	0.0003	0.0047	0.0217	0.1305	-0.0006	0.0183	0.0948	0.0896	0.1319	0.03756
2000 pixels (10 runs)	0.0007	0.0042	0.0204	0.1214	0.0017	0.0214	0.0978	0.0718	0.1233	0.03513
Focal length / 10 (10 runs)	-0.00015	0.00420	-0.00170	0.11640	-0.00025	0.00150	0.09110	0.06660	0.1165	0.03318
Focal length / 49 (10 runs)	-0.0005	0.0044	-0.0722	0.1277	-0.0001	0.0008	0.1332	0.0505	0.1278	0.03639
Focal Length * 10 (0/10 runs successful)	Unstable Reconstructions									
*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Actual Focal Length: 6675 pixels, Random Pixel Noise (Std Dev=.5 pixels)										

Table 4.10: Effect of Focal Length Calibration Noise on Solution Accuracy, .5 pixel noise

the scale factor becomes too large. In this case, the triangulation angles become too small and the reconstruction fails. It is therefore not critical to accurately calibrate the focal length of a camera used for navigation. Additionally, the algorithm can still operate in environments where angular separation is low (high altitude or high frame rate) by constraining the bundle adjustment with a focal length that is lower than the actual focal length.

4.7 IMU Noise

The simulations thus far have also assumed that the attitude measurements from the IMU are perfect. In reality the gyroscopes used to measure yaw, pitch and roll have both biases and random errors depending on their quality and type. In general these errors increase with time and can also vary with g levels and operating temperatures [30]. In addition to noise in the IMU itself, there may be noise in the measurement of IMU to camera mounting angles. These angles can be determined through manual measurement or through constrained bundle adjustment techniques as outlined earlier and as explored by [21]. Finally, the ability to synchronize the camera measurement time with the IMU measurement time is a major challenge (especially if GPS timing is not available). Any lead or lag in these times will cause error in the associated orientation measurement for each image. The error in individual IMU measurements was modeled as zero mean, random Gaussian noise with a constant standard deviation. This was a first order model of IMU behavior over short time periods and neglected many important IMU error sources; however, to first order, this model provides insight

to the effect of IMU errors on the solution. Table 4.11 summarizes the results of added levels of IMU noise to the straight and level test case.

Statistics for position error in the final image frame (all units are meters)										
IMU Angular Noise Standard Deviation (2-Norm of Yaw, Pitch and Roll Std Devs)	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
No IMU noise (100 runs)	0.00073	0.00400	0.02760	0.15610	-0.00110	0.01360	0.14400	0.11260	0.15674	0.04464
0.001° (10 runs)	-0.0035	0.007	0.0951	0.2162	0.0035	0.0218	0.1624	0.168	0.21741	0.06192
.1° (10 runs)	0.0159	0.3787	-1.1399	16.2699	0.0012	0.4159	11.9804	10.4199	16.27962	4.63688

*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise (Std Dev=.5 pixels)

Table 4.11: Effect of IMU Noise on Solution Accuracy, .5 pixel noise

It is clear that levels of IMU noise below about $.001^\circ$ have a negligible affect on the solution as compared to other error sources. However, as IMU noise is increased, it quickly becomes the main driver of error leading to the largest errors observed in any of the simulations run during this research. This is expected since small angle errors lead to large position errors over large distances. The effect is not completely linear and primarily increases the variability of the error as opposed to changing the mean of the error. This is because the errors are still random in direction. Random IMU error is different than an IMU or mounting bias which would give predictable divergence in one direction so as to change the mean of the error in a given direction. The value of $.1^\circ$ is significant as this is approximately the 2-norm of the yaw, pitch and roll error standard deviations in a typical tactical grade IMU after 1 hour of operation without GPS aiding [30].

It was previously shown that constraining the reconstruction with an artificially low focal length did not significantly effect the total error of the reconstruction. This allowed reconstructions at low triangulation angles by artificially increasing the triangulation angle during bundle adjustment; however, this analysis was done assuming that the IMU measured camera angles were known perfectly. When constraining the reconstruction with an artificially low focal length in the presence of IMU noise, then any IMU angular errors are also amplified. Table 4.12 shows the combined effect of IMU error and an artificially low focal length constraint.

Statistics for position error in the final image frame (all units are meters)										
IMU Angular Noise Standard Deviation (2-Norm of Yaw, Pitch and Roll Std Devs)	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{1.5})
.1°, Actual Focal Length (10 runs)	0.02	0.38	-1.14	16.27	0.00	0.42	11.98	10.42	16.28	4.64
.1°, Focal Length / 49 (10 runs)	-0.32	1.48	572.50	313.20	-0.33	1.38	576.70	303.50	313.21	89.21
4.9°, Actual Focal Length (10 runs)	27.40	55.50	1877.40	412.34	810.15	249.74	1996.10	626.90	485.26	138.21

*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise (Std Dev=.5 pixels)

Table 4.12: Interaction of IMU noise and focal length constraint.

These results suggest that constraining the focal length to an artificially low value in the presence of IMU noise has a similar effect as increasing the IMU noise. The effect is not exactly the same and may depend largely on the specific geometry of a given trajectory. Note that in the above case, increasing IMU noise by a factor of 49 increased error dramatically in all three axis whereas reducing the focal length constraint by a factor of 49 primarily increased error in the direction of travel (North in this case). The interaction between focal length constraints and IMU errors has not been fully characterized in this research, but it is clear that when using an artificially low focal length to allow SFM with low triangulation angles, there will be a tradeoff between an increase in IMU errors and the amount the focal length is decreased. The focal length should therefore only be decreased by the minimum amount necessary to allow for a successful reconstruction.

4.8 Model Fit to Simulation Data

Simulation revealed that the the drift rate of the algorithm was influenced by interactions between image GSD, image overlap and error in IMU measurements. These three factors influenced drift rate by changing the nature of the Jacobian in the bundle adjustment. A mathematical model based on theory for how GSD, image overlap and IMU error influence the Jacobian was not developed in this research but the following linear model was fit to the data collected during simulation using the Matlab linear model fit routine:

$$\begin{aligned}
\sigma_{\text{Drift Rate}} = & -.028 + 36.11\text{GSD} - 75.48(\text{GSD})(\text{Overlap}) + 39.67(\text{GSD})(\text{Overlap})^2 \\
& -147.72\sigma_{IMU} - 11641(\text{GSD})\sigma_{IMU} + 45854(\text{GSD})(\text{Overlap})\sigma_{IMU} \\
& -33796(\text{GSD})(\text{Overlap})^2\sigma_{IMU}
\end{aligned} \tag{4.1}$$

where $\sigma_{\text{Drift Rate}}$ is the standard deviation of the total drift rate in $\frac{m}{km^{1.5}}$, GSD is expressed in meters/pixel, Overlap is the ratio defined in Equation 3.14 (expressed as a ratio not a percentage) and σ_{IMU} is the 2-norm of the standard deviation of angular error in the IMU's yaw, pitch and roll measurements. The exact form of the model was chosen to minimize the residuals in the model fit. Although all terms in the above equation were statistically significant, the model is not to be interpreted as an exact physical representation of how GSD, overlap and IMU noise drive drift rate. Instead, the model was fit to the simulation data so that a first order estimate of drift rate under certain real world conditions could be predicted and compared by analogy to actual data collected during flight test. This analogy based prediction only applies to trajectories reconstructed using a constant scale factor determined by the first two images in the sequence, for GSD values between .05-1 meters/pixel, for overlap ratios between .79-1 and for IMU angular error standard deviations of less than $.1^\circ$. Additionally, the total distance traveled in each simulation was 2.3 km, the scale in all simulation data was set with a reference baseline distance of 77 meters and a total of 31 images were used for each simulation. Despite the significant limitations on using this model to predict real world data, it is still useful for a first order analysis to compare simulation results by analogy to actual data collected under similar conditions.

4.9 Combining Bundles

The analysis thus far has focused on the behavior of error within a single bundle of images. In a navigation system operating in real time, bundles need to be

combined sequentially. As discussed in Chapter 3, this can be done either using a fixed constant scale (set by the first two cameras) or a continuously updated scale using outside information (stereo cameras, groundspeed estimate, etc). In order to test the effects of combining bundles the same simulated straight and level trajectory was processed three different ways using both calibration and rotation constraints. The three different methods are listed below and depicted in Figure 4.18.

1. *Constant Scale*: The trajectory was processed either as a single bundle or multiple sequential bundles of three images each. Either way, the process used a constant scale set by the first two images of the entire sequence. This is equivalent to using a monocular camera with no scale updates (this is how all analysis in the previous sections was done).
2. *Continuously Updated Scale*: The trajectory was broken up into bundles of three images where the last image of one bundle was the first image of the next bundle. The scale was set during each bundle by assuming the distance between the first two cameras of each bundle was known. This is equivalent to using a stereo camera system where the stereo cameras are mounted parallel to the direction of motion and the distance covered by the aircraft between frames is twice the stereo distance. This is also equivalent to the case of using a very accurate outside estimate of groundspeed for each bundle. When using groundspeed the scale is found by taking the groundspeed and multiplying by the time between the first two frames of each bundle. In test cases presented here, the scale was set by GPS truth data simulating either a stereo camera or a groundspeed update.
3. *Continuously Updated Scale with Double the Baseline Distance*: This was the same as the second case except the reference scale distance was increased by a factor of two. This was equivalent to using a stereo camera system where the stereo cameras are mounted parallel to the direction of motion and the distance covered by the aircraft between frames is half the stereo distance. This is also

equivalent to the case of using a very accurate outside estimate of groundspeed for each bundle. When using groundspeed the scale is found by taking the groundspeed and multiplying by the time between the first and third frames of each bundle. In test cases presented here, the scale was set by GPS truth data simulating either a stereo camera or a groundspeed update.

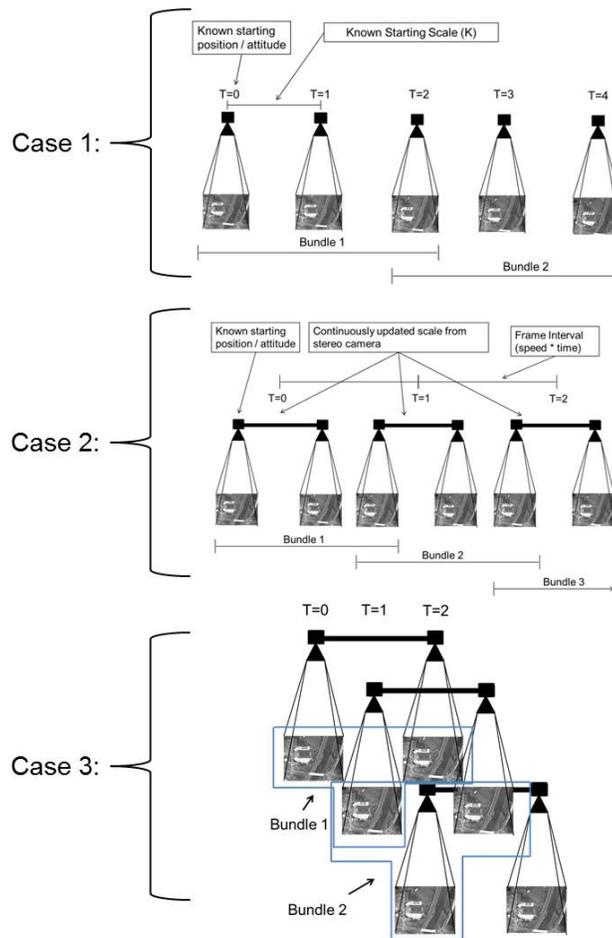


Figure 4.18: The three different tested methods for combining bundles.

Table 4.13 shows the results of simulations run on the straight and level trajectory using all three schemes.

The observed results agree very closely with the results predicted using the error equations. In the case of a fixed scale, the scale error for all the bundles combined

Statistics for position error in the final image frame (all units are meters)														
Method for Combining Bundles	East Error Mean	East Error STD	Predicted East Error STD	North Error Mean	North Error STD	Predicted North Error STD	Vertical Error Mean	Vertical Error STD	Predicted Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Predicted Norm of E,N,V STDs	Drift Rate Standard Deviation
Case 1 (100 Runs)	0.00073	0.00400	0.00350	0.02766	0.15610	0.15000	-0.00110	0.01360	0.01200	0.13000	0.09090	0.15674	0.15052	0.044 m/km ^{1.5}
Case 2 (10 Runs)	-0.00048	0.00441	0.00350	0.00049	0.00882	0.01000	-0.00235	0.01109	0.01200	0.01335	0.00609	0.01484	0.01601	0.098 m/km ^{1.5}
Case 3 (10 Runs)	-0.00020	0.00081	0.00175	0.00140	0.00373	0.00500	-0.00078	0.00670	0.00600	0.00680	0.00344	0.00771	0.00800	0.051 m/km ^{1.5}

Table 4.13: Three different methods used to process the straight and level trajectory. In the first two methods, 15 bundles of three images were combined to form the full trajectory. In the third method, 32 bundles of three images each were used.

is only present in the North direction and propagates as a distance increases. The error equations correctly predicted that incorporating a known scale for every bundle (method 2) eliminates the scale error and reduces error by a factor of 15 (the number of bundles that incorporated scale updates). Once scale error was eliminated, error in the North direction propagated as a random walk and was proportional to $\sqrt{(B)}$, where B is the number of bundles and is proportional to distance. This is verified in Figure 4.19 where the North error in method 2 is shown to increase as a function of square root of distance.

Since the drift rates for methods 2 and 3 are no longer dominated by scale error, they are reported with units of $\frac{m}{\sqrt{km}}$ as opposed to $\frac{m}{km^{1.5}}$. There was no change in the error of the other directions between method 1 and 2 as these directions were not effected by scale error since the trajectory was moving purely to the North.

For method 3, the overlap between each bundle was increased from 1 frame to 2 frames of overlap and 32 bundles were used. Therefore, the error equations predict that error will be further reduced from method 2 by a factor of two. The results agree with the predictions. Error in all directions for method 3 still propagates as a random walk (like method 2) but the overall magnitude is reduced by half.

These results show that using the algorithm implemented in method 2 and method 3 significantly decreases scale error, which was the dominant error in most simulations. A final set of simulations was run to see the results of using method 2 in the presence of IMU noise for a tactical grade IMU. IMU noise was previously found to significantly increase error when compared to the ideal scenario. Table 4.14

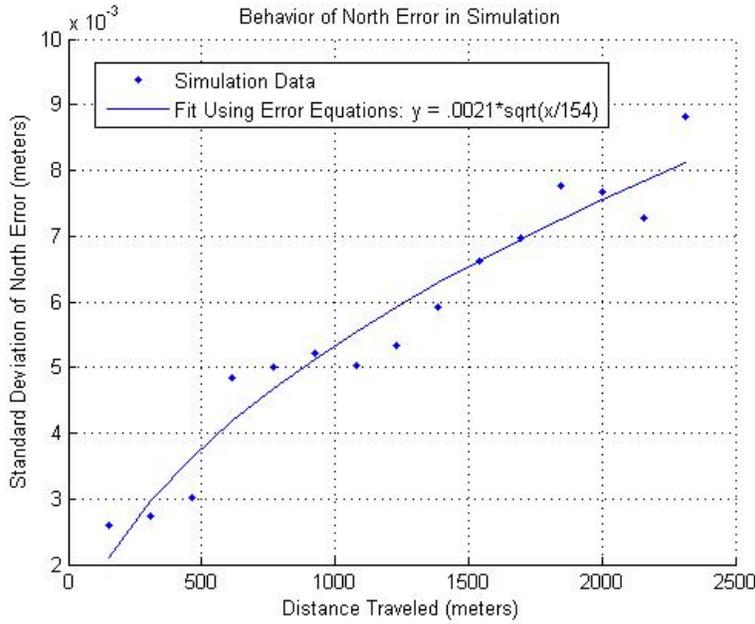


Figure 4.19: The scale error was eliminated by using a simulated stereo camera system. The total error now increases proportional to the square root of distance traveled as opposed to the distance traveled raised to the power of 1.5

shows that the error still increases in the presence of IMU noise but the drift rate with method 2 is still better than the drift rate for method 1 (Table 4.11) given the same level of noise.

Statistics for position error in the final image frame (all units are meters)										
IMU Angular Noise Standard Deviation (2-Norm of Yaw, Pitch and Roll Std Devs)	East Error Mean	East Error STD	North Error Mean	North Error STD	Vertical Error Mean	Vertical Error STD	Radial Error Mean	Radial Error STD	Norm of E,N,V STDs	Drift Rate Standard Deviation (m/km ^{0.5})
Case 2: No IMU Noise (10 Runs)	-0.00048	0.00441	0.00049	0.00882	-0.00335	0.01109	0.01335	0.00609	0.01484	0.00977
Case 2: .01° (10 runs)	-0.0583	0.1233	-0.0689	0.3169	-0.0074	0.0295	0.30590	0.1502	0.34132	0.22457
Case 2: .1° (16 runs)	0.2342	1.7331	0.1518	3.4972	-0.0128	0.2303	3.403	1.7371	3.90987	2.57251

*Setup: Straight/Level Flight with Fixed Downward Pointing Camera, Random Pixel Noise (Std Dev=.5 pixels)

Table 4.14: Improvement gained by using method 2 scheme in the presence of IMU noise.

4.10 Summary of Simulation Results

Overall, the simulation experiments provide a basis for understanding the basic sources of error in this navigation algorithm as well as verifying that the algorithm worked as designed. Simulation verified the theoretical predictions that solution accu-

racy was a largely a function of feature measurement noise, image resolution, image overlap, angular separation and IMU measurement errors. Simulation also showed how complicated interactions between these factors change the overall error depending on aircraft flight parameters and camera mounting parameters. Considering all error sources, it is likely that the real world data will be dominated by IMU errors and scale errors since these produced the largest errors observed in simulation. Additionally simulation showed that the predictions for scale error were correct and that the dominant error was in the direction(s) of travel for the constant scale case. The performance was significantly improved when scale error was eliminated by using a processing scheme that continuously updated scale (method 2 or method 3). A model was fit to the simulation data that will allow a first order comparison to select real world experiments. The performance of the algorithm on a variety of real world flight test data will now be examined.

4.11 ASPN Data

The first set of real world data analyzed were 144 images from the ASPN flight test. These 144 images were processed using the three methods described above in Section 4.9 (constant scale, updating scale with short and long baselines). Three images were used in each bundle. Each ASPN image was separated by .2 seconds. The camera was mounted so that it looked straight down from the aircraft. The aircraft flew on a straight and level trajectory at 146 knots and approximately 484 meters above ground level (AGL). Since this was a straight and level profile with a downward looking camera, the GSD, image overlap and maximum angular separation for this image set were determined using Equations 3.13-3.16. The GSD was calculated to be .37 meters/pixel and this matched an estimated GSD of .38 meters/pixel which was measured directly from the imagery by counting the number of pixels on a football field that the aircraft overflow. The percent overlap for each image was calculated to be 96% and the maximum angular separation between sequential images was 1.76° . The ASPN IMU was tactical grade and the 2-norm of the standard deviation of yaw,

pitch and roll angular errors was estimated to be $.025^\circ$. Equation 4.1 predicts that the standard deviation of the drift rate for the constant scale case (method 1) under these conditions will be $7.86 \frac{m}{km^{1.5}}$. Figure 4.20 shows position errors of the estimated trajectory for the constant scale case.

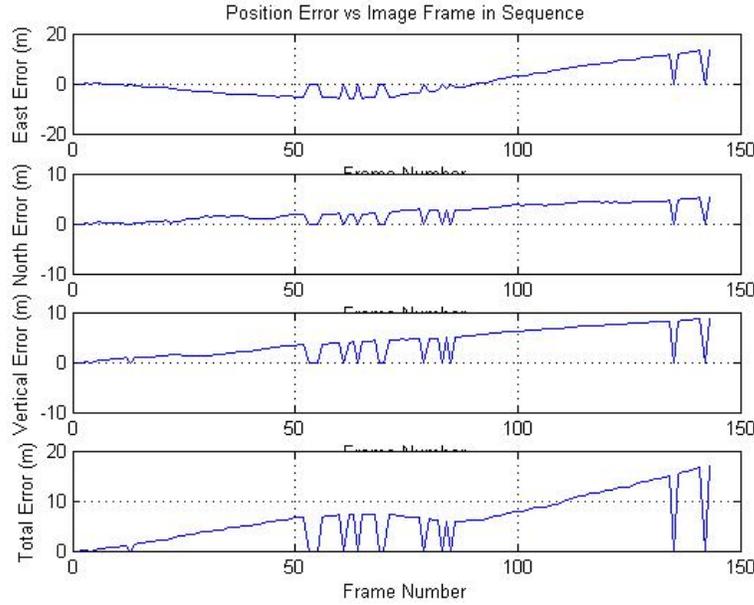


Figure 4.20: ASPN Errors Method 1: Constant scale set by the first two images.

In this case the total error at the end of the trajectory was 17 meters yielding a drift rate of $5.45 \frac{m}{km^{1.5}}$. This single sample drift rate falls well within the predicted drift rate standard deviation. It is likely that the majority of the error was due to IMU errors which, in simulation, dominate the contribution to drift rate in otherwise high quality cameras. Note the data dropouts in Figure 4.20. These were caused by the failure of a given bundle to successfully reconstruct. There were 12 out of 71 bundles that failed to reconstruct resulting in a 17% bundle failure rate. The failure rate is likely due to the low angular separation between each image (1.76°). The same data was then processed using method 2 so as to continuously update the scale of the reconstruction. In this case, method 2 simulated using a stereo camera system with 15 meters between the two cameras. Figure 4.21 shows the resulting errors.

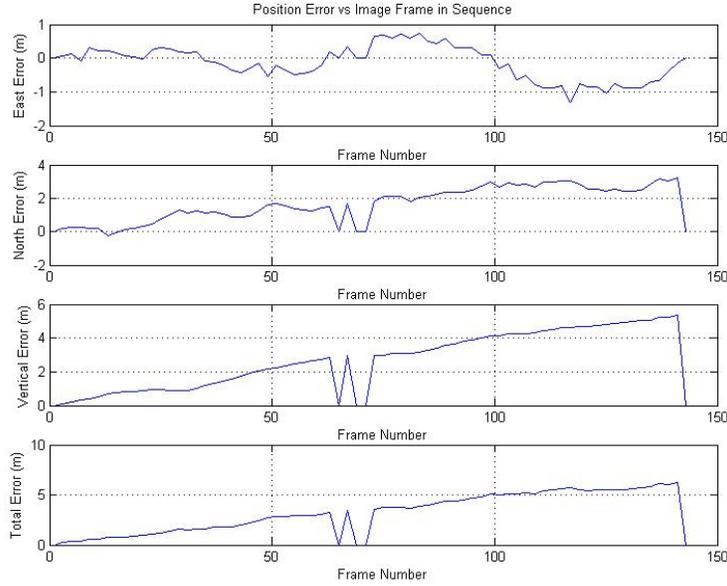


Figure 4.21: ASPN Errors Method 2: Updating scale set by first and second image of each bundle.

The total error is reduced to a rate of $4.29 \frac{m}{\sqrt{km}}$. The error is further reduced by using method 3. In this case, method 3 simulated using a stereo camera system with 30 meters between the two cameras. Figure 4.22 shows that the total error for method 3 is $.513 \frac{m}{\sqrt{km}}$. The error reductions are not as large as predicted by the scale error equations but still significant.

Overall, the errors observed in this data follow the error trends observed in simulation. In the constant scale case, scale error was dominant in the direction of travel. Once scale error was removed, the total error decreased drastically and was further decreased in method 3. The magnitude of the errors observed corresponds to the expected magnitude of error when considering IMU errors. It is important to note that the error plots are single samples and not plots of standard deviations. Therefore the individual plots of error do not have the shape of \sqrt{x} or $x^{1.5}$ even though the drift rate standard deviation follows these trends. Simulation suggested that in some cases it may be better to run the reconstruction without constraining camera attitude rather than constraining attitude with noisy IMU measurements. Figure 4.23

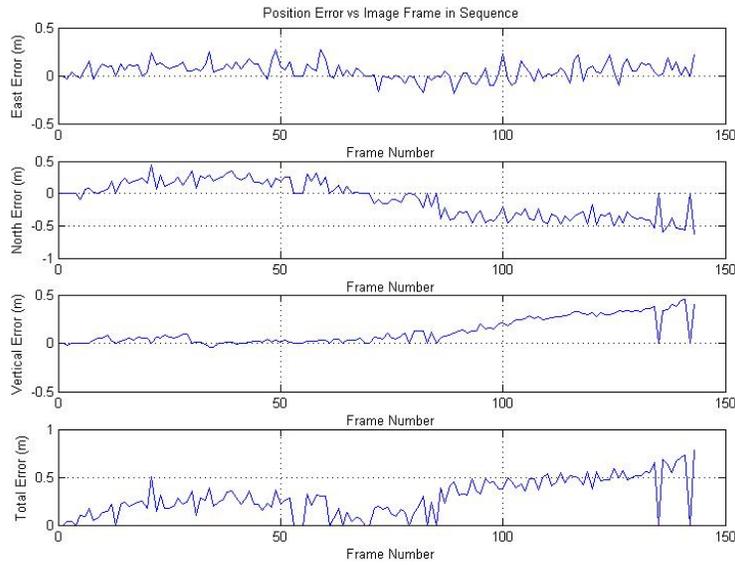


Figure 4.22: ASPN Errors Method 3: Updating scale set by first and third image of each bundle.

shows the results of the reconstruction when attitude was not constrained and scale was constantly updated using the same method as method 3.

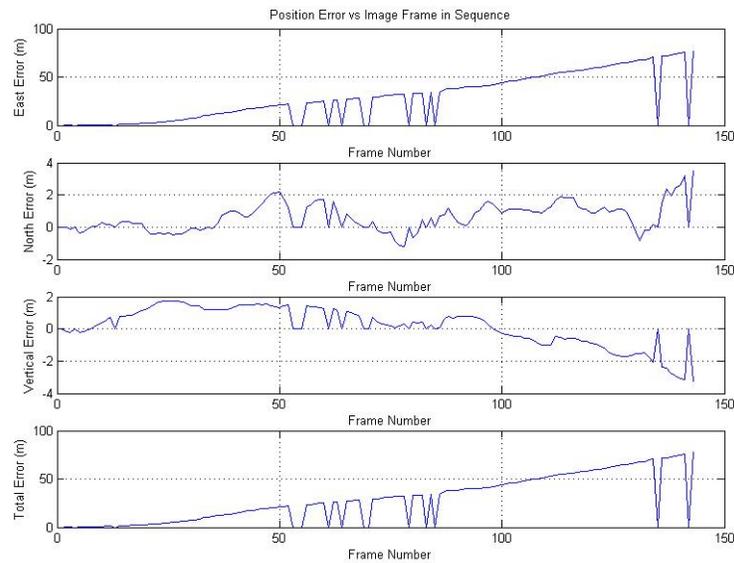


Figure 4.23: ASPN Errors Method 3: Updating scale set by first and third image of each bundle, camera rotation unconstrained.

The total drift rate was $52.89 \frac{m}{\sqrt{km}}$. This was larger by a factor of 100 than the previous use of method 3 when camera orientation was constrained. In this case, it was clearly better to constrain the solution with noisy IMU measurements than to run an unconstrained reconstruction. This suggests that the estimated attitudes of the unconstrained solution were more uncertain than the noise in the IMU ($.025^\circ$). In fact, for the last frame in the trajectory, the VSFM attitude estimates were off by -5.3° in yaw, 6.5° in pitch and 13.5° in roll. Finally, note that the majority of the error was in the East direction and that the error looks like scale error. This error was not scale error since scale error was removed by using the case 3 scheme. Instead this error was due to angular inaccuracies in each bundle. The total calculated path length to the East is shorter than actual because the orientation of each bundle is inaccurate meaning that path "wobbles" to the East instead of smoothly moving in that direction.

4.12 Angel Fire Data

The next set of real world data consisted of 217 frames taken from the Angel Fire system. This trajectory was flown at an altitude of 4900 meters AGL and a speed of 160 knots. The images have a measured GSD of .95 meters/pixel which is higher than the ASPN data. As with the ASPN data, the GSD was also measured by counting the number of pixels on a football field that was overflowed by the aircraft. The Angel Fire aircraft flew a circular trajectory and the camera was mounted on the inside of the turn looking sideways and about 45° down toward the ground. Images were captured every .8 seconds. The camera was on gimbals so that it moved with respect to the aircraft but was fixed with respect to the IMU. Due to the unique geometry of the camera setup it is not possible to use the exact form of equations 3.14-3.16 to find overlap and maximum angular separation; however these parameters were estimated as 96% overlap and $.57^\circ$ using rough calculations based on those equations. The main difference between the ASPN data and the Angel Fire data is that the Angel Fire GSD was greater than the ASPN GSD by a factor of 2.5. The Angel Fire system also

contained a tactical grade IMU with angular errors on the order of $.025^\circ$. Equation 4.1 predicts that the standard deviation of the drift rate in the constant scale case will be $26.09 \frac{m}{km^{1.5}}$. Figures 4.24, 4.25 and 4.26 show the position errors for the Angel Fire data processed using methods 1-3.

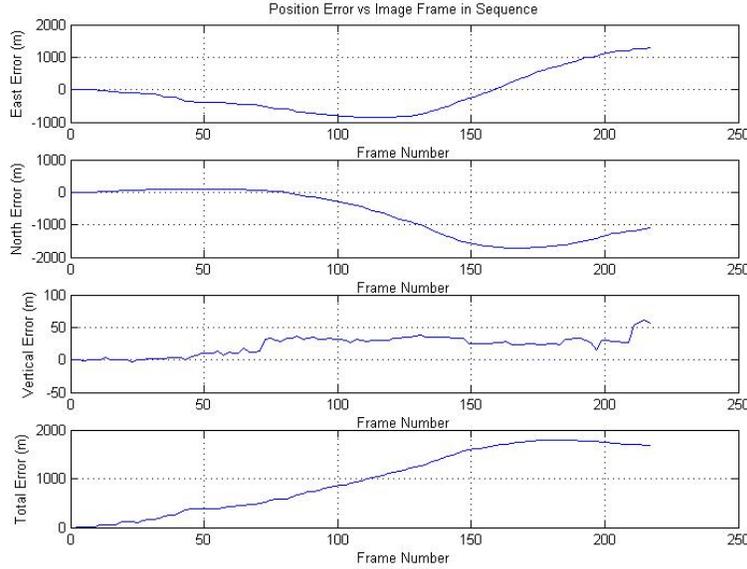


Figure 4.24: Angel Fire Errors Method 1: Constant scale set by the first two images.

The total drift rates were $30.24 \frac{m}{km^{1.5}}$, $21.88 \frac{m}{\sqrt{km}}$ and $8.4 \frac{m}{\sqrt{km}}$ for methods 1-3, respectively. Note that in the first case, the scale error was seen in both the North and East directions since the aircraft was flying a circular trajectory. The total drift of this constant scale case is well within two standard deviations of the predicted drift rate suggesting that the prediction was accurate. The simulation results also predicted the correct trend between ASPN and Angel Fire data.

As expected, the scale error was reduced for methods 2 and 3. In this case, methods 2 and 3 simulated using a stereo camera system with 67 meters and 134 meters, respectively, between the two cameras. The error when using these methods was also larger than the error in the ASPN data due to the higher GSD. Also note that successful Angel Fire reconstructions were possible at very small angular separations ($.57^\circ$) without the use of an artificial scale factor. In simulations with comparable

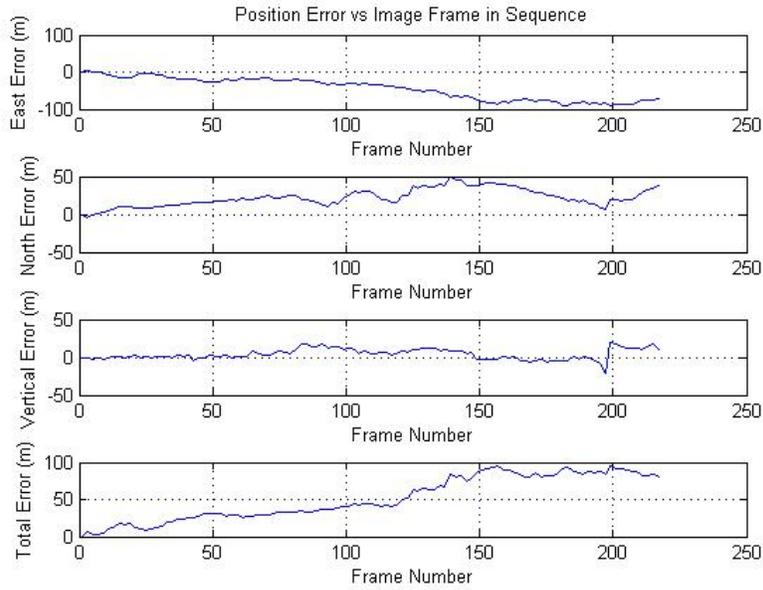


Figure 4.25: Angel Fire Errors Method 2: Updating scale set by first and second image of each bundle.

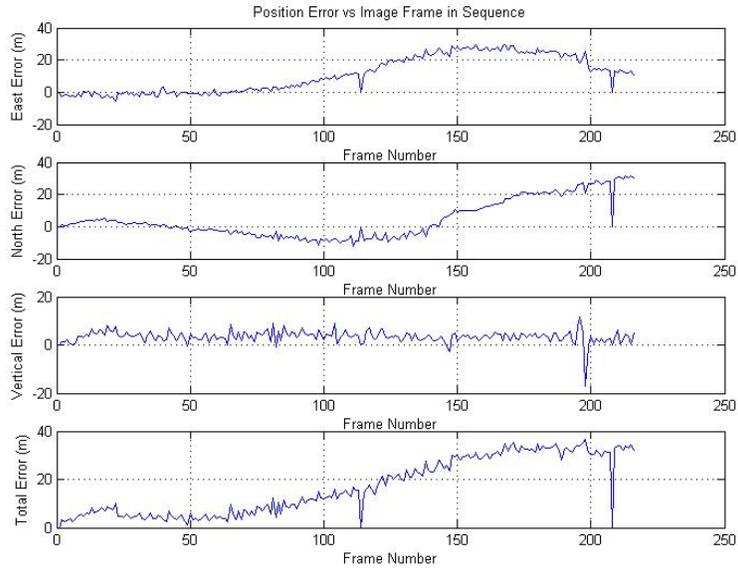


Figure 4.26: Angel Fire Errors Method 3: Updating scale set by first and third image of each bundle.

GSD and overlap, reconstructions failed below 1° until an artificial scale factor was used that enabled reconstructions below 1° . The added stability of the Angel Fire

reconstruction was likely due to the fact that the Angel Fire images had about 8,000 SIFT features per image. During simulation, the images only had about 1,000 features per image. The increased number of features may have contributed to increased reconstruction stability at smaller angles than was possible in simulation.

4.13 MAMI-I Data

The MAMI-I data set provided another opportunity to analyze data from a high resolution sensor. The camera was similar to the camera used in Angel Fire but was flown at a lower altitude and shallower look angle. In this case the aircraft flew a circular trajectory at 109 knots and 852 meters AGL but the camera looked sideways and down about 45° resulting in a slant range of about 1205 meters from camera to target. The resulting estimated GSD for the MAMI-I data was .16 meters / pixel with a total of 375 images taken at .5 second intervals. Each sequential image had about 92% overlap with the previous image and a maximum angular separation between images of less than 1.28° . As with the Angel Fire data, the camera was gimballed with respect to the aircraft but fixed with respect to the IMU. The IMU was also a tactical grade IMU with angular errors on the order of $.025^\circ$. Equation 4.1 predicts that the standard deviation of the drift rate in the constant scale case will be $3.7 \frac{m}{km^{1.5}}$. The predicted reduction in drift rate relative to ASPN and Angel Fire is due to the much lower GSD of the MAMI-I data. Figures 4.27, 4.28 and 4.29 show the position errors for methods 1-3.

The total drift rates were $28.6 \frac{m}{km^{1.5}}$, $29.97 \frac{m}{\sqrt{km}}$ and $6.37 \frac{m}{\sqrt{km}}$ for methods 1-3, respectively. In this case, methods 2 and 3 simulated using a stereo camera system with 27 meters and 54 meters, respectively, between the two cameras. The drift rate in this sample of data for the constant scale case is well above 3 standard deviations of the predicted drift rate. The drift rates were not lower than the ASPN and Angel Fire cases despite predictions. This suggests another source of error in the MAMI-I trajectory that was not predicted in simulation and was not seen in previous data. Closer examination of a segment of 25 frames showed that there was a constant angular

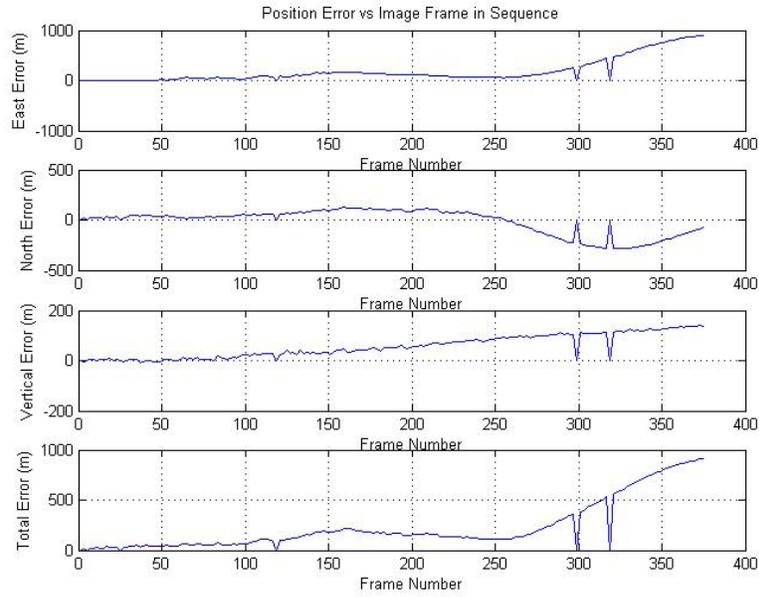


Figure 4.27: MAMI I Errors Method 1: Constant scale set by the first two images.

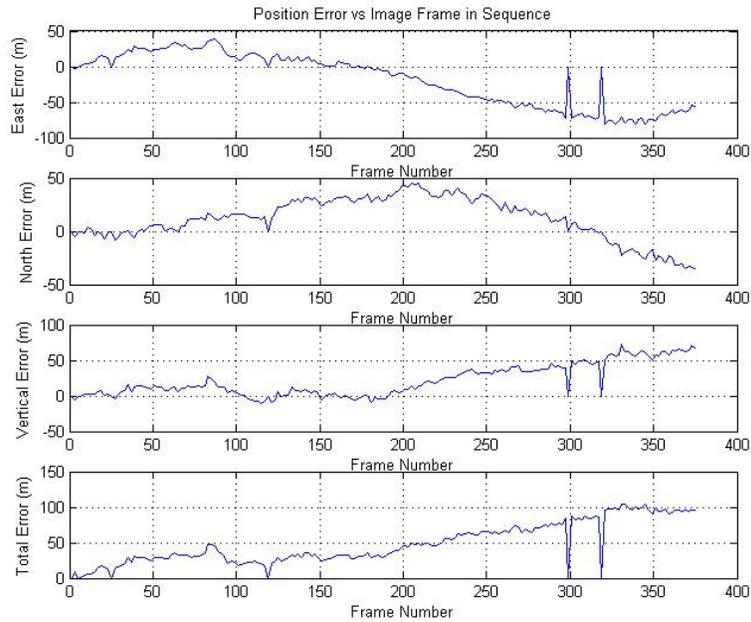


Figure 4.28: MAMI I Errors Method 2: Updating scale set by first and second image of each bundle.

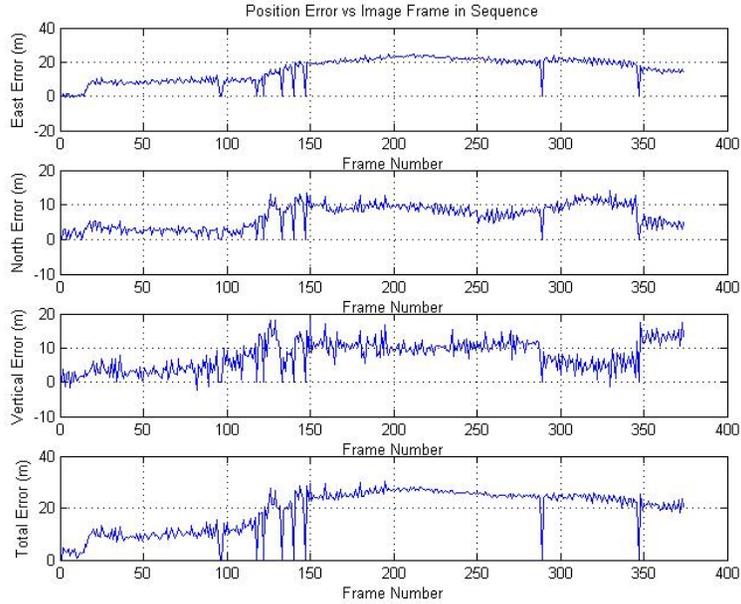


Figure 4.29: MAMI I Errors Method 3: Updating scale set by first and third image of each bundle.

bias in the trajectory. Figures 4.30 and 4.31 show a segment of 25 frames of the total MAMI-I trajectory. The calculated trajectory for the segment tends to move down and to the left with respect to the actual trajectory. This same error was observed in other segments of the trajectory. This was indicative of a bias error in the camera mounting parameters (ie. the Direction Cosine Matrix (DCM) going from IMU frame to camera frame).

This error was somewhat expected as the provided DCM between the IMU and the camera frame was exactly Identity, indicating that no true calibration had been done and that the transformation was assumed to be perfect which was likely not the case in the real world. The angular bias in the first camera was removed by calculating a direction cosine matrix (DCM) between the image and IMU frames that minimized the error between the calculated and actual trajectories. The DCM was calculated using the minimization routine known as Wahbas problem which was discussed in Chapter 3. Wahbas problem is a least squares minimization routine that finds a best fit DCM between two sets of data in different reference frames. In this

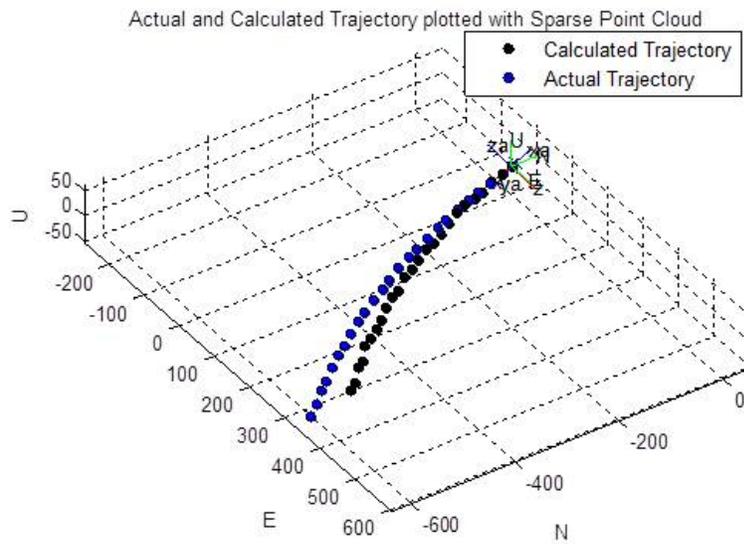


Figure 4.30: MAMI-I Segment 1 Actual and Calculated Trajectory

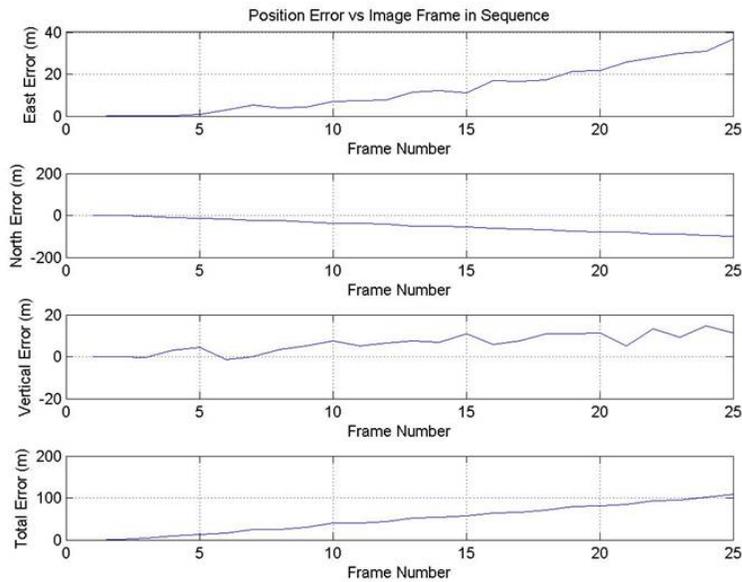


Figure 4.31: MAMI-I segment 1 errors showing evidence of an angular camera mounting error

case, the best fit DCM between the truth data and the calculated trajectory represents the best fit IMU to camera frame DCM that eliminates angular error bias error. For

the above segment of 25 images, this process reduced error by 10 meters. The total segment distance was 670.5 meters suggesting an angular bias on the order of $.85^\circ$ using Equation 3.17 to estimate the effect of angular bias. With this level of IMU error, Equation 4.1 predicts the standard deviation of the drift rate of the MAMI-I data in the constant scale case to be $123.9 \frac{m}{km^{1.5}}$ as opposed to the original prediction of $3.7 \frac{m}{km^{1.5}}$ which assumed angular errors on the order of $.025^\circ$. The observed drift rate now falls within the $1-\sigma$ bounds of this prediction. Note the dramatic effect that a small angular error had on the drift rate.

4.14 C-12 Data

Several sets of flight test data from cameras mounted on a C-12 aircraft at the USAF Test Pilot School were made available for this research. Unfortunately, the transformations between camera and IMU frame also contained significant errors similar to what was seen in the MAMI-I data. Reconstructions were possible but error was dominated by angular problems and not enough information was available to troubleshoot. Therefore, no quantitative results are listed for this data set.

Even though no quantitative data for the C-12 flight is presented, an interesting observation was made about flight over water. Two C-12 data sets were analyzed that flew a camera over the Pacific Ocean. Attempts to reconstruct these trajectories failed to produce any semblance of an accurate trajectory. This is likely due to the lack of distinct features in the ocean environment making it difficult to accurately register images. This is a significant limitation to any future SFM based operational system, although it may be possible to adjust the settings of the feature matcher to improve image registration over feature deficient terrain (ie ocean, clouds, etc).

4.15 MAMI-II Data

The MAMI-II project collected 8 terabytes of image and position data to support this research and other AFRL research efforts. For this the purposes of this paper, three trajectory segments from the MAMI-II data are analyzed in detail.

4.15.1 *Trajectory 1.* The first MAMI-II trajectory analyzed was a trajectory in which the aircraft flew straight and level for 1.6 km to the East and then started a banked level turn with variable roll while flying another 1.6km. The aircraft flew at 266 knots and at 2459 meters AGL altitude above desert terrain. The GSD of this data was .06 meters/pixel as calculated by equation 3.13 and this was verified by actual measurements of images taken of a resolution target. Images were taken every .04 seconds. The small camera field of view, coupled with the high groundspeed meant that the maximum angle between sequential images was $.12^\circ$ and that there was only 85% overlap between sequential images. The IMU was also a tactical grade IMU with angular errors on the order of $.025^\circ$.

The MAMI-II images initially failed to reconstruct any trajectory until the focal length was constrained to artificially increase angular separation using the method described in Section 4.5.3. This method was previously shown to allow successful reconstructions but, in the presence of IMU noise, this technique also dramatically increases error. The focal length was constrained to 1500 pixels which meant that angular separation was artificially increased by a factor of 49 to 5.9° . Angular errors were therefore also increased by a factor of 49 to 1.23° . Without this large angular error, the MAMI-II data was predicted to have a standard deviation of drift rate of about $.7 \frac{m}{km^{1.5}}$; however, with angular errors on the order of 1.23° , the drift rate standard deviation was predicted to be $33.7 \frac{m}{km^{1.5}}$.

This trajectory revealed several challenges to reconstruction. First, the reconstruction algorithm failed several times so a single, smooth reconstruction was not possible. The failure of the algorithm corresponded to changes in roll. Additionally, the drift rate of the reconstruction was variable and changes in this drift rate also corresponded to changes in roll. The attitude profile of the trajectory is shown in Figure 4.32.

The first 300 frames were successfully reconstructed using all three variations of the algorithm. The drift rates for the first 300 frames (1.6km) were $54.95 \frac{m}{km^{1.5}}$,

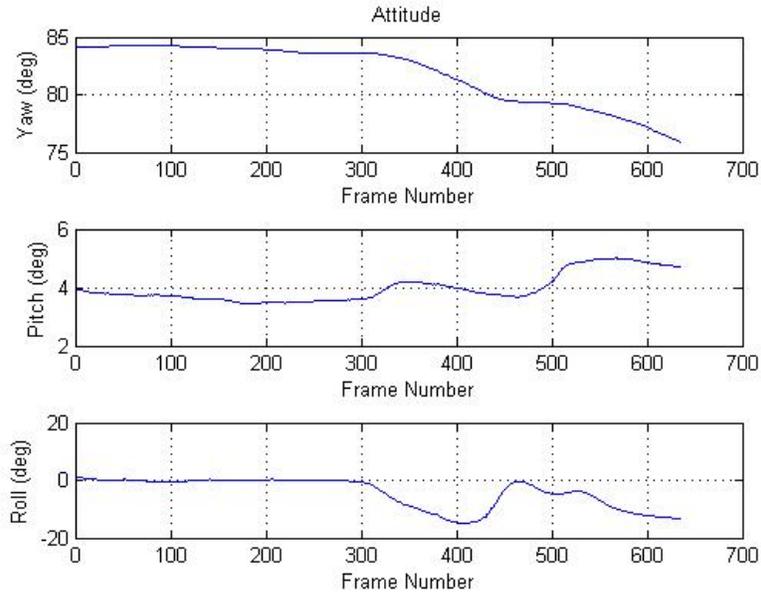


Figure 4.32: Truth aircraft attitude at each frame as recorded by the IMU. The aircraft flies straight for 300 frames and then begins a variable bank turn. The total distance flown is 3.2km.

$67.68 \frac{m}{\sqrt{km}}$ and $27.36 \frac{m}{\sqrt{km}}$ for method 1-3 respectively. In this case, methods 2 and 3 simulated using a stereo camera system with 5 meters and 10 meters, respectively, between the two cameras. The observed drift rate for method 1 was within two predicted standard deviations (when considering angular error) but the reduction in drift rate for methods 2 and 3 was not as large as was seen in previous data. There were a total of 12 out of the first 150 bundles that failed to reconstruct causing brief data interruptions in the calculated trajectory (8% failure rate). The errors in the first 169 frames are shown in Figures 4.33, 4.34 and 4.35.

Even though the error was reduced when using methods 2 and 3, the error was still much larger than expected and was dominated by a linearly increasing error in the vertical axis. This is indicative of angular error that is either a result of camera mounting errors or IMU errors amplified by the low focal length constraint. A further analysis of this angular error is discussed shortly.

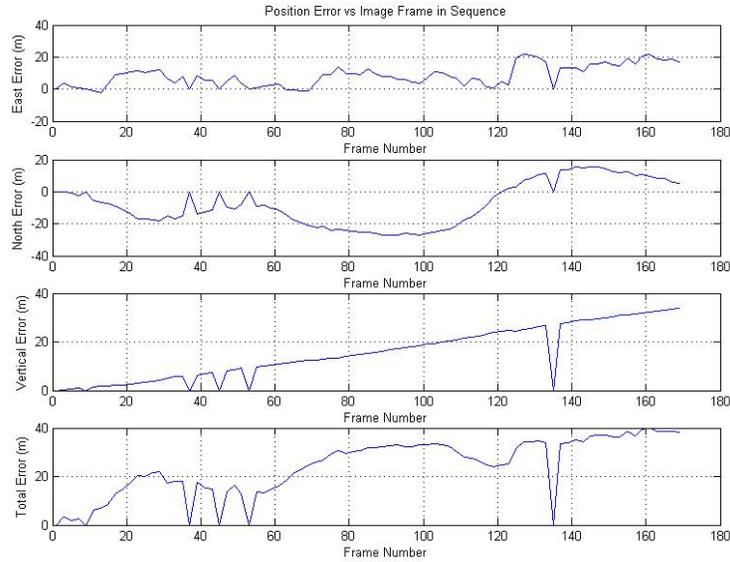


Figure 4.33: MAMI II Errors Method 1: Constant scale set by the first two images.

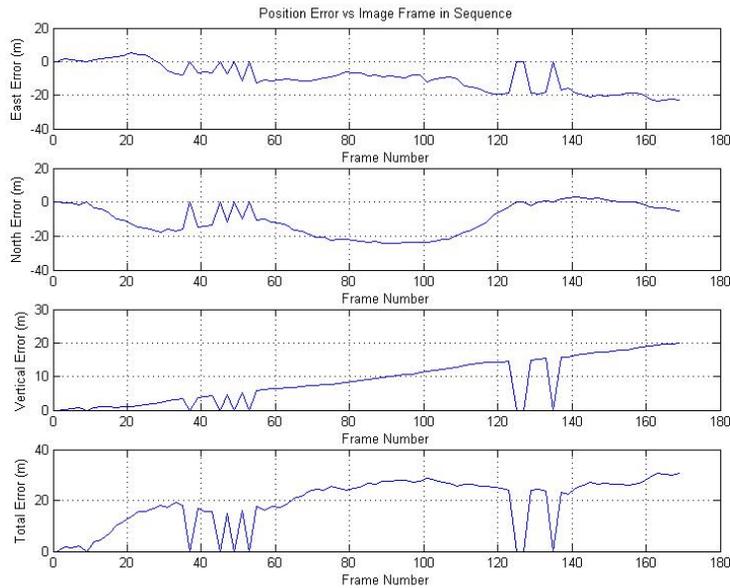


Figure 4.34: MAMI II Errors Method 2: Updating scale set by first and second image of each bundle.

In the remaining 325 frames of the trajectory there were 52 bundles that failed to reconstruct yielding a 32% failure rate and several data dropouts. The drift rate

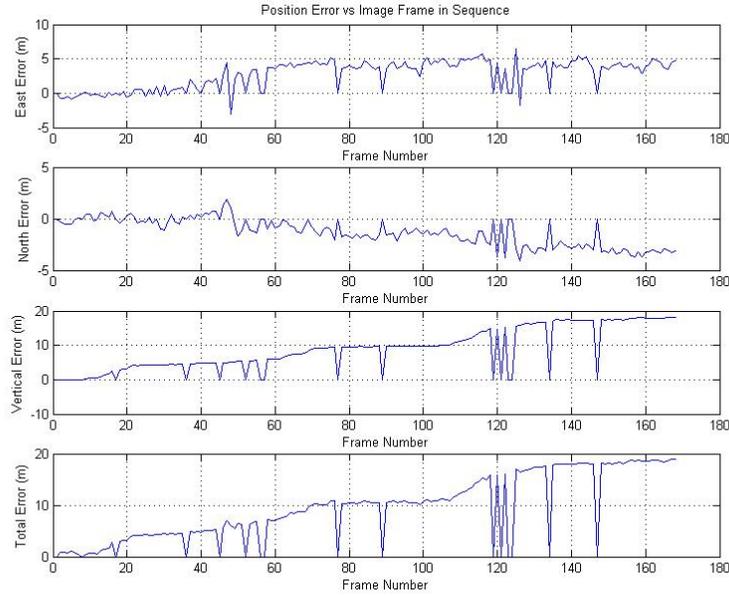


Figure 4.35: MAMI II Errors Method 3: Updating scale set by first and third image of each bundle.

in this portion of the trajectory was also highly variable and very large. The drift rate ranged from 73.5 to 714 $\frac{m}{km^{1.5}}$ in the constant scale case and the reconstruction often did not resemble anything that looked like a realistic trajectory. The use of different algorithm methods (methods 1-3) did not improve the solution. In some instances, sequential cameras were calculated to be over 1000 meters apart. These gross errors can be treated as failed reconstructions. These failures corresponded to rapid and large changes in aircraft attitude as the aircraft changed its bank angle. This behavior was not previously seen in simulation as the combination of rapid attitude changes and small camera fields of view were not simulated. However, it is known from simulation that turning trajectories increase error due to the interactions of GSD and image overlap as angular rates vary. Small fields of view combined with rapid attitude changes mean that overlap can be significantly decreased between images leading to poorly conditioned Jacobians in the bundle adjustment causing ambiguity between rotation and translation. Therefore any noise in the rotation constraint leads to noise in translation estimates and rapid attitude changes may be interpreted as as

large position changes. In other words, the roll in the trajectory was ambiguous with lateral translation and there was too much noise in the IMU attitude measurements to sufficiently constrain the bundle adjustment and resolve the ambiguity.

The amount of error in the IMU attitude measurements was a function of the IMU specifications and was amplified by a factor of 49 due to the focal length constraint. A 30 frame subsection of the above trajectory was chosen for further analysis. This 30 frame section was processed as a single bundle with constant scale and attitude constrained with IMU data. The results are shown in Figure 4.36.

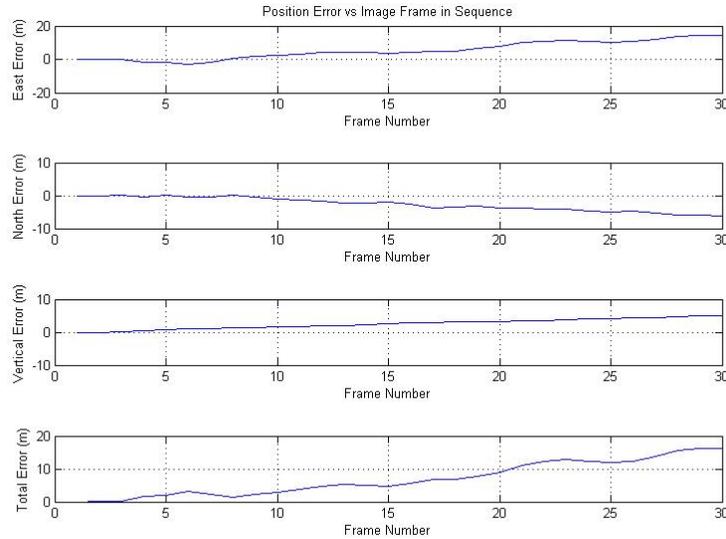


Figure 4.36: Position errors of a 30 frame segment processed using constant scale and IMU constrained attitude.

The total drift rate was $278.3 \frac{m}{km^{1.5}}$. Note that the error was dominated by scale error in the East direction (direction of travel); however there are also linearly increasing errors in the North and Vertical directions. This is indicative of an angular bias in the measured orientation of the first camera. The bias could be a result of camera mounting error or IMU error. The camera mounting parameters were calculated using laser measurements and were accurate to 0.001 degrees. Additionally, the error in other segments of the trajectory diverged in different directions than in

this segment indicating that the angular error observed was not due to a constant bias but rather to a randomly changing angular error. To first order, it is appropriate to assume that all the error in the vertical direction in Figure 4.36 was due to angular error. The final value of the linearly increasing vertical error was about 6 meters after flying 150 meters. Using trigonometry, this suggests an angular bias of 2.3 degrees at the first camera. This was consistent in magnitude with the angular errors expected after amplification due to the constrained focal length. In other words, the orientation of the first camera was likely incorrect by some random amount in each direction. This random error in the first camera orientation was propagated as a constant angular bias throughout the sequence.

In order to further demonstrate the effect of angular error on the calculated trajectory, the angular bias in the first camera was removed by calculating a direction cosine matrix (DCM) between the camera and IMU frames that minimized the error between the trajectories. The DCM was calculated using the minimization routine known as Wahbas problem which was discussed in Chapter 3. Wahbas problem is a least squares minimization routine that finds a best fit DCM between two sets of data in different reference frames. In this case, the best fit DCM between the truth data and the calculated trajectory had no physical meaning since it was a function of the random IMU error; however applying this DCM eliminated angular errors in this segment. The resulting errors are shown in Figure 4.37.

When the angular bias was removed using this method, the total drift rate was reduced to $214.72 \frac{m}{km^{1.5}}$. The primary reduction of error was in the vertical and north directions supporting the hypothesis that error in these directions was dominated by angular errors; however the total error was still much higher than predicted indicating that the process for removing angular bias does not work well in this case, possibly because of the artificially low focal length constraint. This is currently the only explanation for why such large drift rates remained in this set of MAMI-II data.

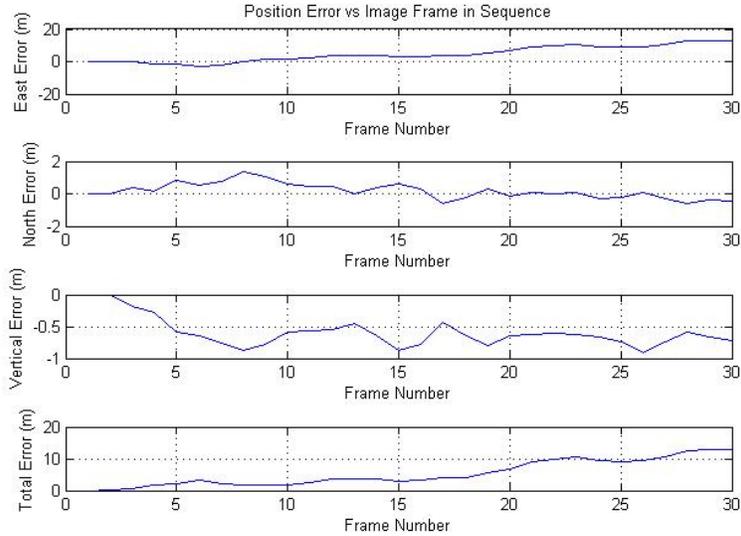


Figure 4.37: Position errors of a 30 frame segment processed using constant scale and a best fit DCM to remove angular errors.

Finally, the test in Figure 4.36 was repeated without constraining rotation with IMU data. This led a drift rate of $828.8 \frac{m}{km^{1.5}}$ indicating that constraining the MAMI II solution with inaccurate IMU data was still better than relying on SFM estimates of camera attitude.

4.15.2 Trajectory 2. The second MAMI trajectory analyzed consisted of 625 frames taken every .04 seconds for a total distance of 4.028 km as the aircraft flew to the Southwest. The aircraft flew straight for the first 400 frames and then rolled to 60° of bank in four seconds and held a 60° bank angle in a turn for 5 seconds. The aircraft maintained a slight descent in first 500 frames, losing a total of 50 meters. After rolling into the bank, the aircraft pitch dropped and the descent rate increased. In all, a total of 138 meters of altitude was lost throughout the entire trajectory. This trajectory was started at 3669 meters above the ground and at 329 knots. The GSD, overlap and maximum triangulation angle for the straight and level portion were .06 meters/pixel, 87% and $.12^\circ$. Once again, the reconstruction was constrained with an

artificial focal length of 1500 pixels to allow successful reconstructions. The attitude profile of the trajectory is shown in Figure 4.32.

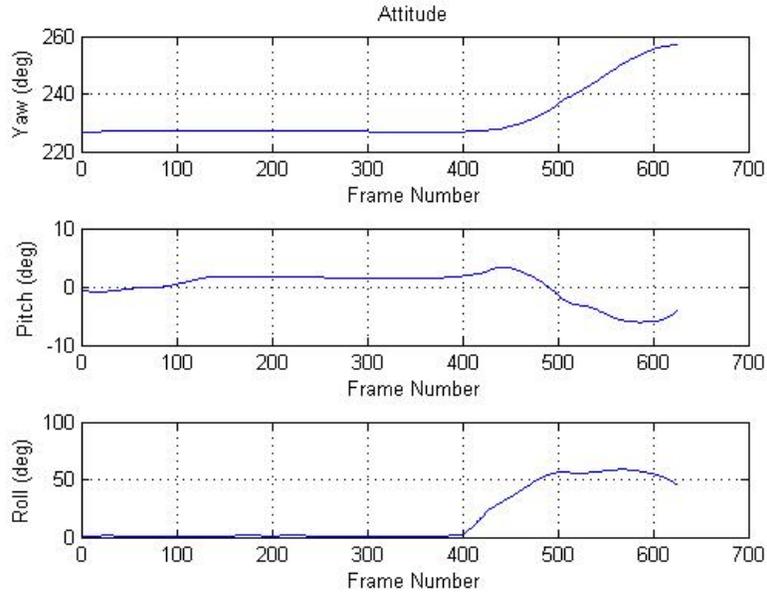


Figure 4.38: Truth aircraft attitude at each frame as recorded by the IMU. The aircraft flies straight for 400 frames and then begins a roll to 60° of bank. The total distance flown is 4.028km.

Since the GSD and overlap were the same as the first trajectory, and assuming that angular errors were still present, the drift rate standard deviation was predicted to be $33.7 \frac{m}{km^{1.5}}$. The resulting reconstructions using the three methods are shown in Figures 4.39, 4.40 and 4.41.

Note that all variations of the algorithm fail to successfully reconstruct the interval between frames 400 and 500 where the roll rate is large. When using method 3, the algorithm fails to reconstruct anything beyond frame 400. The drift rate for frames 0-400 is $59.27 \frac{m}{km^{1.5}}$, $124.2 \frac{m}{\sqrt{km}}$ and $79.88 \frac{m}{\sqrt{km}}$ for methods 1-3, respectively. The error for the constant scale case is within two standard deviations of the predicted result. As before, these errors are likely due to IMU errors amplified by the constrained focal length. The errors follow the appropriate trend in that the error decreases for each case. Note the large component of linearly increasing vertical error. Linearly

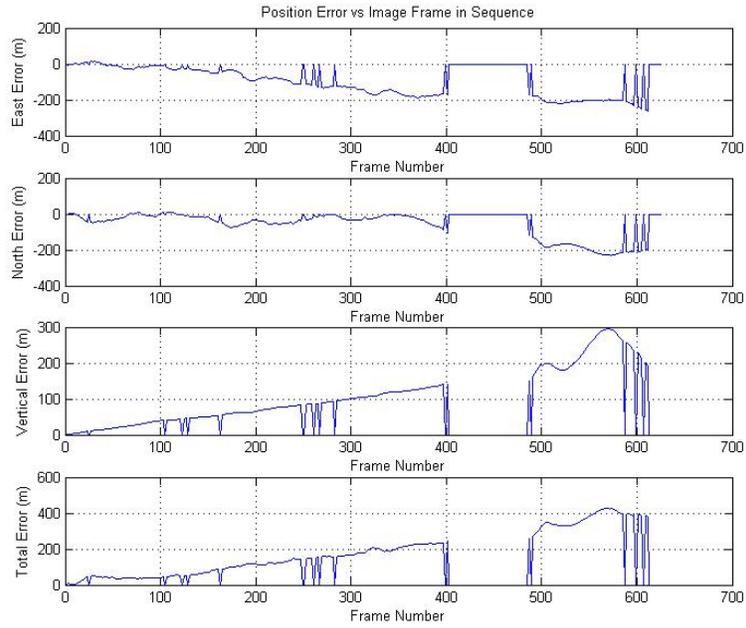


Figure 4.39: MAMI II Errors Method 1: Constant scale set by the first two images.

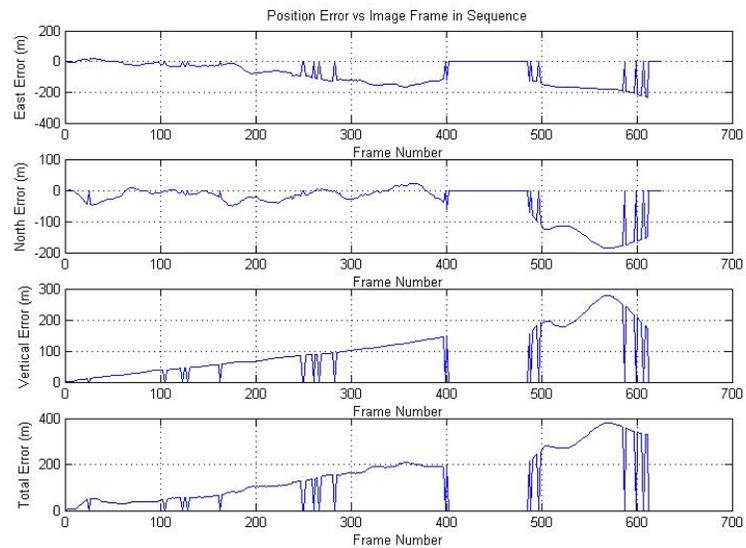


Figure 4.40: MAMI II Errors Method 2: Updating scale set by first and second image of each bundle.

increasing vertical error of this nature was also seen in the first trajectory and was

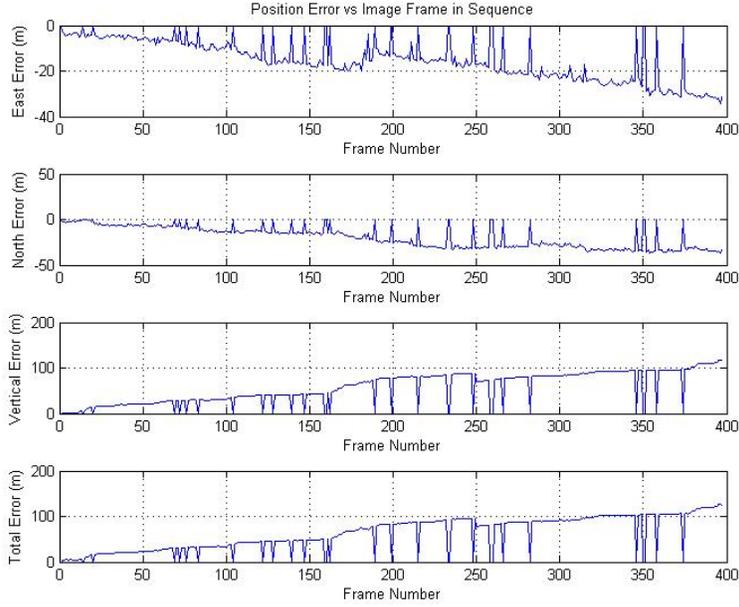


Figure 4.41: MAMI II Errors Method 3: Updating scale set by first and third image of each bundle.

thought to be the result of angular error from the IMU amplified by a factor of 49 (from the focal length constraint).

After frame 500, the error trends in the Vertical and North directions correspond with changes in roll. The drift rate from frame 500-625 is $80.1 \frac{m}{km^{1.5}}$ and $83.38 \frac{m}{\sqrt{km}}$ for cases 1 and 2 respectively. Case 3 failed completely above frame 400. In this regime, method 2 has a larger overall error than method 1. This suggests that the primary error is driven by the ambiguity between attitude and translation due to small field of view and low image overlap. This error was large enough that it dominated scale error effects which is why no improvement was seen from using method 2. This was the same effect as seen previously in the first trajectory; however, in the first trajectory, the effect was large enough to cause complete reconstruction failure. In this case, the effect remains bounded so it can be observed (between frames 500-625). The behavior suggests that the error corresponds with roll rate and not just roll angle, as was seen in simulation. The reconstruction failed at high angular rates (frames

400-500) but was successful at high angles and lower rates (frames 500-625). This makes sense since angular rate drives image overlap. High angular rates and small fields of view significantly reduce image overlap and lead to very poorly conditioned Jacobian matrices in the bundle adjustment process.

4.15.3 Trajectory 3. The final trajectory analyzed was part of a loop maneuver. A loop was initiated at 4,835 meters AGL and 591 knots over desert terrain heading East. This sequence contains 57 images spaced 0.01 seconds apart and represents only 0.6 seconds of the entire loop maneuver. The interval between sampled images was reduced when analyzing this sequence to obtain sufficient overlap between images. Even after reducing the interval between images, each image only contained about 30% overlap with neighboring images. This was reduced as compared to the previous trajectory due to the high aircraft speed and the aggressive pitch rate which caused the camera FOV to move faster along the ground. IMU truth data were only available for every fourth image so error analysis was only conducted every fourth image and camera attitude was not constrained. During this segment of the loop, the aircraft was approximately 45° nose high with pitch increasing at a rate of 7° per second.

The algorithm performed significantly worse than the straight and level case. The images were processed as a single bundle with constant scale factor. Figure 4.42 shows a plot of the position error indicating a total drift rate of $1285.4 \frac{m}{km^{1.5}}$.

The predicted scale error was clearly seen in the east and vertical directions, which were the primary directions of travel. Additionally there was angular bias error in the north direction. The error sources in this reconstruction matched the behavior of the predicted error sources previously discussed; however, the dynamic nature of this trajectory exacerbated the errors. The degraded performance of the algorithm during this maneuver was due to three main contributors. First, as the aircraft pitched up the distance to objects that the camera viewed increased. At 45 degrees pitch, the camera was aimed at the horizon as opposed to 0 degrees pitch when the camera

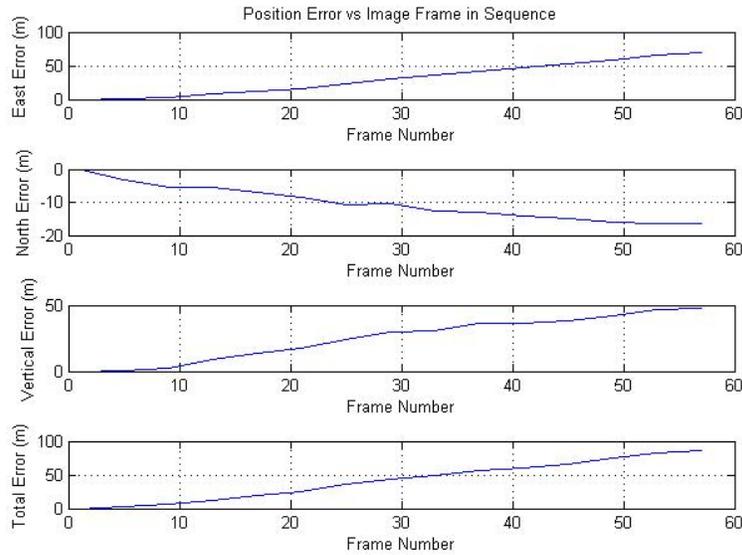


Figure 4.42: MAMI II Errors during loop trajectory. Constant scale case.

was aimed down. This decreased the image resolution and increased the amount of pixel error in the feature matching process. Second, the high pitch rate meant that the camera FOV was tracking much faster along the ground than when pitch rate was zero in level flight. In level flight the camera motion was only due to aircraft velocity. The increased camera motion meant that there was less overlap between successive images. Furthermore, the high pitch rate increased image smearing. This increased error in pixel location and contributed to error in reconstruction accuracy. Finally, the loop was flown at a higher airspeed than the straight and level trajectory (591 knots versus 329 knots). The higher speed had the same effect as the high pitch rate in reducing the overlap between successive images while also increasing image smearing.

4.16 Summary of Real World Data

Overall, the error trends predicted by simulation were observed in the real world data. The error magnitudes were successfully predicted to first order when all angular errors were taken into account. Additionally, the MAMI-II data revealed a type of

error behavior that was not directly observed in simulation. The combination of a small camera field of view and high rates of roll, pitch and yaw led to error that was highly non-linear, unpredictable and very large in magnitude. The ambiguity between camera attitude and camera translation was exacerbated in these situations leading to failed reconstructions or very large errors. Finally, analysis of real world data also showed that reconstructions could be successful with very small triangulation angles between successive cameras but when focal length is constrained to an artificially low value, angular error effects increase.

V. Conclusions and Recommendations

5.1 Summary of Results

This research developed a prototype algorithm based on Structure from Motion that can successfully reconstruct the trajectory of an aircraft to determine the aircraft's current position using only a known starting point and images taken from a camera or cameras mounted on the aircraft. The algorithm was not implemented in real time but could be adopted to a real time system with streamlined software. The error associated with the reconstructed trajectory was predicted using theoretical concepts and validated with both simulation and real data. In an ideal scenario, the overall drift rate and reliability of the navigation solution was shown to be a function of image GSD, overlap and triangulation angle. These factors were in turn determined by complex interactions between camera parameters and aircraft trajectory. The algorithm estimated trajectory drifted from true trajectory as a function of distance traveled. The drift was dominated by uncertainty in the scale of the reconstruction as well as angular errors in estimated camera orientations. It was shown that constraining the algorithm with periodic scale and attitude updates significantly improved the solution. Once constrained in this way, the overall drift rate and reliability was dominated by angular errors in the IMU data used to constrain the solution. These errors are the most important errors to consider in any future operational implementation of such a system since the overall drift rate is limited by the quality of the available IMU.

5.1.1 Proposed Operational Concept. Many current aircraft and airborne weapons already have embedded imaging, inertial and computer systems that can be used to implement the algorithm developed in this research. This section outlines a proposed operational concept for an aircraft or weapon that has these embedded systems and has implemented the algorithm as proposed in this research. The operational concept assumes that the performance of the system is the performance demonstrated in this study and no further improvements to the algorithm are made.

Additionally, the scenario assumes that the drift rates demonstrated in this research are representative of drift rates over longer distances.

The mission statement in this proposed concept is that the customer needs an airborne vehicle that can travel 30 nautical miles from a known starting point without GPS or INS guidance to arrive at a target as accurately as possible. The vehicle can have a stereo camera system as well as an IMU to measure yaw, pitch and roll angles. To first order (neglecting angular bias), the angular accuracy of the IMU after one hour of operation without GPS updates can be modeled as random noise with a standard deviation that depends on the quality of the gyroscopes. For current tactical and navigation grade IMUs, the standard deviation of angular error after one hour is on the order of $.01^\circ$ and $.001^\circ$, respectively.

Using the results of this study combined with engineering judgment, any system used to satisfy the above operational requirement should meet the guidelines in Table 5.1.

GSD	<.5 meters/pixel
Sequential Image Overlap	>90%
Sequential Image Angular Separation	>.5°

Table 5.1: Recommended system guidelines

The values for these parameters were chosen using engineering judgment based on first order analysis of the simulation and flight test results in this study. The recommended values should be used as a general guide to start system design and not hard constraints. Assuming the use of a stereo vision system operating under method 3 (proposed in Chapter 4) on an F-16 with one camera mounted near the nose and the other mounted near the tail (51 feet stereo distance), Table 5.2 shows the required operational parameters and limitations needed to meet the guidelines in Table 5.1.

With current camera technology it is relatively easy to meet the GSD and overlap requirements but the angle requirement is difficult to meet in such a scenario since large distances between stereo cameras are required at higher altitudes and stereo

Groundspeed	<500 knots
Altitude	<2600 ft AGL
Roll/Pitch Rate*	<33°/sec
Bank/Pitch Angle*	<10°
Frame Rate	>36 frames/s
FOV	>12°
Focal Length	>27mm
Pixel Size	<5 microns
*Assuming downward, body fixed cameras and only 12° FOV	

Table 5.2: Operational parameters and limitations required to meet recommended system guidelines.

distance is limited by aircraft length or wingspan. If a larger operating envelope is required, then it might be possible to extend the stereo distance by towing a camera behind the aircraft or incorporating data from cameras mounted on other aircraft in formation.

Table 5.3 shows the expected position error when the vehicle arrives at the target for various configurations demonstrated during this research. The best results were achieved using either algorithm method 2 or 3 while conforming to the guidelines in Table 5.1.

Research Trial	IMU Angular Error Standard Deviation (°)	Predicted Standard Deviation of Position Error on Target (meters)
Simulation	0.001	0.07
Simulation	0.01	1.67
Simulation	0.1	19.18
Research Trial	IMU Angular Error Standard Deviation (°)	Predicted Total Position Error based on Single Sample (meters)
Best ASPN Data	0.025	3.97
Best Angel Fire Data	0.025	62.61
Best MAMI 1 Data	0.025	47.48
Best MAMI 2 Data	0.025	203.94

Table 5.3: Operational performance predictions for a 30 km flight based on drift rates demonstrated during this research.

The results are variable depending on many factors but some of the results demonstrated during this research would be operational useful for many military

and civilian applications. Overall the system behaves much like an INS. Whereas an INS uses gyroscopes and accelerometers to determine trajectory, this system uses gyroscopes and cameras or only cameras. In some circumstances, it may be more cost effective to use a visual system based on this algorithm rather than a traditional INS.

5.2 Major Research Contributions

The following list summarizes the major contributions made by this research effort:

1. SFM based image navigation on airborne imagery: Previous research efforts demonstrated image based navigation using either SFM or Kalman filtering on image data collected from ground based robots or small UAVs under controlled circumstances [19] [18] [2] [8] [3] [1] [26] [9] [17] [28]. This research expanded on these previous efforts by demonstrating SFM based techniques on a much larger scale, in uncontrolled environments and on a wide variety aerial platforms.
2. Comprehensive error characterization and simulation: This research developed theory and simulation tools that can be used to predict errors in SFM based reconstruction on specific platforms in specific environments.
3. Techniques for transformation of SFM reconstructions to world scales and coordinate systems: This research developed a method for transforming SFM reconstructions from arbitrary to real world coordinate systems. This method can be used not only for navigation routines but also for geolocation of three dimensional targets reconstructed using SFM.
4. Techniques for constraining bundle adjustment: This research developed showed that SFM solutions can be improved with the use of various constraints that are often available on airborne systems.
5. Application of Wahba's problem to determine camera mounting parameters: A novel method for determining the mounting parameters of a camera on an aircraft was proposed. The method required GPS and image data collected

during flight and may be a useful alternative to hardware measurements in certain situations.

6. Demonstration of successful SFM in low triangulation angle situations: Successful reconstructions were demonstrated with maximum triangulation angles as low as $.12^\circ$. This was done on actual images by constraining the focal length in bundle adjustment to an artificially low value. It was shown that this method increased reconstruction reliability but also increased error in the presence of IMU noise. This method may allow for successful SFM in situations where it was not previously thought to be possible due to small distances between cameras.
7. Invariance of reconstruction accuracy to focal length noise: Reconstruction accuracy was not dependent on small amounts of focal length noise when accurate IMU data was available. This means that precise calibration of focal length is not necessary for SFM based navigation algorithms.

5.3 Recommendations

The following list summarizes the major recommendations for future research and operational systems:

1. Future operational systems built to implement this type of algorithm should use cameras that strive to achieve the lowest possible GSD while maintaining at least 90% image overlap between sequential images for the expected operational environment. Using multiple cameras to stitch images together allows for a wide field of view without sacrificing image GSD. The system should be constructed so that images of the ground are always taken no matter the attitude of the aircraft.
2. Future operational systems built to implement this type of algorithm should use stereo camera systems that ensure angular separation between the cameras of at least $.5^\circ$ (assuming low GSD and high overlap) throughout the system's operational envelope. If the aircraft length or wingspan is too small to support

the required stereo distance for a given envelope, consider designing a system to tow a camera behind the aircraft or use cameras mounted on multiple aircraft in a formation. Implementing this algorithm with a monocular camera system will require a continuous outside source of scale updates (ie. radar altimeter, INS, GPS).

3. Future operational systems built to implement this algorithm should incorporate systems capable of providing independent angular measurements of camera attitude. The angular error in attitude needs to be commensurate with the desired operational performance.
4. Conduct further work to make the simulation tools developed in this research more robust and capable of quickly predicting error with a wide variety of camera parameters and flight trajectories. Further validate simulation results with a larger sample of actual data.
5. Conduct further research on ways to mitigate the effect of IMU errors on trajectory reconstruction. In particular, conduct further research on the propagation of angular errors in Structure from Motion.
6. Conduct further research to study the scale and position errors of three dimensional target models created using SFM.
7. Conduct further research to determine ways of extracting features and performing SFM when overflying water, clouds or other feature deficient environments.
8. Develop an algorithm for incorporating the results of SFM trajectory reconstruction into a Kalman filter with other navigation updates.
9. Conduct further software development to make the proposed algorithm run in real time using government owned or contracted software that can be used on future operational systems.

Bibliography

1. A. Proctor, T. Apker, E. Johnson. “Vision-only Control and Guidance for Aircraft”. *Journal of Robotics*, 2006.
2. A. Watkins, R. Lind, J. Kehoe. “SLAM for Flight through Urban Environments using Dimensionality Reduction”. *AIAA GNC Exhibit*, 2006.
3. A. Wu, A. Proctor, E. Johnson. “Vision-Aided Inertial Navigation for Flight Control”. *AIAA GNC Conference*, 2005.
4. Aider, O. A., P. Hoppenot, and E. Colle. “A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences”. *Robotics and Autonomous Systems*, 52(2):229–246, 2005.
5. Alix, D. “MAMI-II Technical Information Memorandum”. 2014.
6. Beich, J. “Vision-Aided Autonomous Precision Weapon Terminal Guidance Using a Tightly-Coupled INS and Predictive Rendering Techniques”. *Master’s Thesis, AFIT*, 2011.
7. Bouguet, J.Y. “Camera Calibration Toolbox for Matlab”, 2013. URL http://www.vision.caltech.edu/bouguetj/calib_doc/.
8. C. De Wagter, J.A. Mulder. “Towards Vision-Based UAV Situational Awareness”. *AIAA GNC Conference*, 2005.
9. C. Olsen, M. Schoppers M. Maimone, L. Matthies. “Robust Stereo Ego-motion for Long Distance Navigation”. *NASA JPL*, 2013.
10. C. Wu, B. Curless, S. Agarwal and S. Seitz. “Multicore Bundle Adjustment”. *CVPR*, 2011.
11. Davison, A. J., I. D. Reid, N. D. Molton, and O. Stasse. “MonoSLAM: Real-Time Single Camera SLAM”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007. ID: 1.
12. Ekholm, J. “3-D Scene Reconstruction from Aerial Imagery”. *Master’s Thesis, AFIT*, 2012.
13. Gonzalez, R., R. Woods, and S. Eddins. *Digital Image Processing using Matlab*. Gatesmark, USA, 2 edition, 2009.
14. Goshtasby, A. Ardeshir. *2-D and 3-D Image Registration*. John Wiley and Sons, New Jersey, 1 edition, 2005.
15. Hartley, R. I. and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, 2 edition, 2003.

16. J. Dias, L. Almeida, J. Lobo. “Cooperation Between Visual and Inertial Information for 3-D Vision”. *Proceedings on the 10th Mediterranean Conference on Control and Automation*, 2002.
17. J. Kim, S. Sukkarieh. “Real-Time Implementation of Airborne Inertial-SLAM”. 2006.
18. J. Langelaan, S. Rock. “Passive GPS-Free Navigation for Small UAVs”. *IEEEAC*.
19. Konolige, K. and M. Agrawal. “FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping”. *Robotics, IEEE Transactions on*, 24(5):1066–1077, 2008. ID: 1.
20. Lee, T. “Vision Lab Geometry Library”. *www.vision.ucla.edu/vlg*, 2008.
21. Lorenzini, P. “Sensitivity Analysis of an Automated Calibration Routine for Airborne Cameras”. *Master’s Thesis, AFIT*, 2013.
22. Lowe, D. G. “Distinctive Image Features from Scale-Invariant Keypoints”. 2004.
23. Ma, Y., S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3-D Vision*. Springer-Verlag, New York, 1 edition, 2004.
24. Montgomery, D. *Design and Analysis of Experiments*. John Wiley and Sons, New Jersey, 8 edition, 2012.
25. N. Snavely, S. Seitz and R. Szeliski. “Photo Tourism: Exploring Photo Collections in 3D”. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 25(3):835–846, 2006.
26. N. Sunderhauf, S. Lacroix P. Protzel, K. Konolige. “Visual Odometry Using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle”. *Tagungsband Autonome Mobile Systeme*, 2013.
27. Owens, R. “Camera Calibration”, 1997. URL <http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCALCOPIES/OWENS/LECT9/node2.html>.
28. P. Saeedi, D. Lowe, P. Lawrence. “Vision-Based 3D Trajectory Tracking for Unknown Environments”. 2006.
29. Raquet, J. “An Introduction to GPS”. University Lecture Notes, 2012.
30. Titterton, D. H. and J. L. Weston. *Strapdown Inertial Navigation Technology*. AIAA and IEEE, 2 edition, 2004.
31. Triggs, B., P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. “Bundle Adjustment A Modern Synthesis”.
32. Veth, M. “Fusing of Imaging and Inertial Sensors for Navigation”. *P.h.D Thesis, AFIT*, 2006.
33. Wahba, G. “Problem 65-1: A Least Squares Estimate of Spacecraft Attitude”. *SIAM Review*, 1965.

34. Walli, K. “Relating Multimodal Imagery Data in 3D”. *P.h.D Thesis, Rochester Institute of Technology*, 2010.
35. Wikipedia. “Axes conventions — Wikipedia, The Free Encyclopedia”, 2014. URL [\url{http://en.wikipedia.org/w/index.php?title=Axes_conventions&oldid=640398769}](http://en.wikipedia.org/w/index.php?title=Axes_conventions&oldid=640398769).
36. Wikipedia. “Sum of normally distributed random variables — Wikipedia, The Free Encyclopedia”, 2015. URL http://en.wikipedia.org/w/index.php?title=Sum_of_normally_distributed_random_variables&oldid=645940193. [Online; accessed 18-February-2015].
37. Wu, C. “VisualSFM: A Visual Structure from Motion System”. <http://homes.cs.washington.edu/~ccwu/vsfm/>, 2011.
38. Wu, C. “Towards Linear-time Incremental Structure from Motion”. *3DV*, 2013.
39. Yamaguchi, K. “MEXOPENCV”, 2013. URL <http://www3.cs.stonybrook.edu/~kyamagu/mexopencv/>.

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 27-03-2015	2. REPORT TYPE Master's Thesis	3. DATES COVERED (From - To) Sept 2012 - March 2015
--	--	---

4. TITLE AND SUBTITLE Error Characterization of Flight Trajectories Reconstructed Using Structure from Motion	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S) Alix, Daniel C., Capt, USAF	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way Wright-Patterson AFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENY-MS-15-M-214
--	---

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Lab Mr. Richard Van Hook 2241 Avionics Circle Wright-Patterson Air Force Base, Ohio 45433 richard.van_hook.1@us.af.mil	10. SPONSOR/MONITOR'S ACRONYM(S) AFRL
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
This research effort assessed the accuracy of Structure from Motion (SFM) algorithms in replicating aircraft fight trajectories. Structure from Motion techniques can be used to estimate aircraft trajectory by determining the position and pose of an aircraft mounted camera from a sequential series of images taken during flight. An algorithm is proposed and implemented that successfully reconstructed aircraft trajectory using only a known starting position and a sequential series of images. The error in and reliability of the algorithm was found to be a function of image resolution as well as the amount of overlap and angular separation between sequential images. The trajectory estimated by the algorithm drifted from the true trajectory as a function of distance traveled. The drift was dominated by uncertainty in the scale of the reconstruction as well as angular errors in estimated camera orientations. A proposed system architecture that incorporated scale and attitude updates was tested on actual flight test data. The architecture successfully reconstructed a variety of trajectories but drift rates were highly variable.

15. SUBJECT TERMS
structure from motion, visual odometry, image based navigation, image registration, three dimensional reconstruction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 153	19a. NAME OF RESPONSIBLE PERSON Capt Daniel Alix
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (203)246-1471, daniel.alix@us.af.mil