



FUSION OF IMAGING AND INERTIAL SENSORS FOR NAVIGATION

DISSERTATION

Michael J. Veth, Major, USAF

AFIT/DS/ENG/06-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

The views expressed in this dissertation 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/DS/ENG/06-09

FUSION OF IMAGING AND INERTIAL SENSORS FOR NAVIGATION

DISSERTATION

Presented to the Faculty
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 Doctor of Philosophy

Michael J. Veth, B.S.E.E., M.S.E.E.
Major, USAF

September, 2006

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

FUSION OF IMAGING AND INERTIAL SENSORS FOR NAVIGATION

Michael J. Veth, B.S.E.E., M.S.E.E.

Major, USAF

Approved:

Date

John F. Raquet (Chairman)

Gary B. Lamont (Dean's Representative)

Mark E. Oxley (Member)

Meir Pachter (Member)

Gilbert L. Peterson (Member)

Accepted:

M. U. Thomas

Dean, Graduate School of Engineering and Management

Date

Abstract

The introduction of the Global Positioning System changed the way the United States Air Force fights, by delivering world-wide, precision navigation capability to even the smallest platforms. Unfortunately, the Global Positioning System signal is not available in all combat environments (e.g., under tree cover, indoors, or underground). Thus, operations in these environments are limited to non-precision tactics. The motivation of this research is to address the limitations of the current precision navigation methods by fusing imaging and inertial systems, which is inspired by observing the navigation capabilities of animals. The research begins by rigorously describing the imaging and navigation problem and developing practical models of the sensors, then presenting a transformation technique to detect features within an image. Given a set of features, a rigorous, statistical feature projection technique is developed which utilizes inertial measurements to predict vectors in the feature space between images. This coupling of the imaging and inertial sensors at a deep level is then used to aid the statistical feature matching function. The feature matches and inertial measurements are then used to estimate the navigation trajectory online using an extended Kalman filter. After accomplishing a proper calibration, the image-aided inertial navigation algorithm is then tested using a combination of simulation and ground tests using both tactical and consumer-grade inertial sensors. While limitations of the extended Kalman filter are identified, the experimental results demonstrate a navigation performance improvement of at least two orders of magnitude over the respective inertial-only solutions.

Table of Contents

	Page
Abstract	iv
List of Figures	viii
List of Tables	xii
List of Symbols	xiii
List of Abbreviations	xv
 I. Introduction	 1
1.1 Problem Definition	2
1.1.1 Accelerometers	2
1.1.2 Gyroscopes	3
1.1.3 Imaging Sensors	3
1.1.4 Trajectory Estimation	6
1.2 Research Contributions	7
 II. Mathematical Background	 9
2.1 Mathematical Notation	9
2.2 Reference Frames	10
2.3 Coordinate Transformations	16
2.4 Error Models and Perturbation Analysis	17
2.5 Inertial Navigation	19
2.5.1 Accelerometer Sensor Model	19
2.5.2 Gyroscope Sensor Model	20
2.5.3 Inertial Navigation Dynamics	22
2.5.4 Attitude Dynamics	23
2.5.5 Position and Velocity Dynamics	24
2.5.6 Development of Observation Equations	25
2.6 Inertial Navigation Error Model	27
2.6.1 Inertial Sensor Error Model	28
2.6.2 Attitude Error Development	28
2.6.3 Velocity Error Development	30
2.6.4 State-space Model	32
2.7 Digital Imaging	34
2.7.1 Optical Sensor Model	34
2.7.2 Projection Theory	37

	Page
2.7.3 Nonlinear Optical Distortion	42
2.7.4 Fixed Object Tracking	44
2.8 Kalman Filtering	46
2.8.1 Linear Kalman Filter	46
2.8.2 Extended Kalman Filter	49
2.9 Optimal Batch Estimation	52
2.9.1 Linear Mathematical Models	52
2.9.2 Nonlinear Mathematical Models	53
2.10 Direction Cosine Perturbation Methods	55
III. Navigation Using Imaging and Inertial Sensors - An Analysis of the State of the Art	58
3.1 Early Methods	58
3.1.1 Celestial Navigation	58
3.1.2 Driftmeter Navigation	61
3.2 Non-inertial Optical Navigation Methods	61
3.3 Inertially-aided Optical Navigation	64
3.3.1 Bhanu, Roberts, and Ming	65
3.3.2 Roumeliotis, Johnson, and Montgomery	66
3.3.3 Pachter, Polat, and Porter	67
3.3.4 Raquet and Giebner	68
3.3.5 StreLOW, Singh, and Adams	70
IV. Image and Inertial Fusion Navigation Algorithm	78
4.1 Proposed Image and Inertial Fusion Algorithm Walkthrough	78
4.2 Detailed Discussion of Image and Inertial Fusion Algorithm	82
4.2.1 Stochastic Feature Transformation	82
4.2.2 Statistical Predictive Transformation of Feature Space	90
4.2.3 Statistical Feature Matching	97
4.2.4 Navigation State Error Estimation	99
4.3 Image and Inertial Fusion Algorithm Extended Kalman Filter Implementation	100
4.3.1 Kalman Filter Description	100
4.3.2 Feature Tracking Algorithm	102
4.3.3 Measurement Models	103
4.4 Absolute Measurements	120
4.4.1 Reference Landmark	120
4.4.2 Revisitation of Previously Tracked Landmark	122
4.4.3 Celestial Measurements	123

	Page
V. Experimental Navigation System Design and Calibration	125
5.1 Experimental Image-Aided Navigation System Overview	125
5.1.1 Sensors	125
5.2 Calibration and Alignment	128
5.2.1 Camera Calibration	129
5.2.2 Sensor Alignment	133
VI. Simulation and Experimental Results	138
6.1 Stochastic Projection Algorithm Tests	138
6.1.1 Monte Carlo Simulation	138
6.1.2 Flight Test Experiment	142
6.2 Image and Inertial Fusion Algorithm Ground Tests . . .	143
6.2.1 Simulation	145
6.2.2 Experiment	156
6.3 Image and Inertial Fusion Algorithm Flight Tests	160
6.3.1 Simulation	166
VII. Conclusions	175
7.1 Conclusions	175
7.2 Future Work	177
7.3 Summary	182
Bibliography	183
Index	189

List of Figures

Figure		Page
1.1.	General Accelerometer Block Diagram.	3
1.2.	General Rate Gyroscope Block Diagram.	4
1.3.	General Imaging System Block Diagram.	5
1.4.	General Trajectory Estimation Block Diagram.	6
2.1.	Inertial, Earth and vehicle-fixed navigation frame illustration. .	12
2.2.	Earth-fixed navigation frame illustration.	13
2.3.	Aircraft body frame illustration.	14
2.4.	Camera frame illustration.	15
2.5.	Binocular disparity frame illustration.	15
2.6.	Imaging system model.	36
2.7.	Thin lens camera model.	38
2.8.	Pinhole camera model.	39
2.9.	Camera projection model.	40
2.10.	Camera image array.	41
2.11.	Target to image transformation geometry.	43
3.1.	Celestial Observation Geometry.	59
3.2.	Feed-forward Navigation Kalman Filter Block Diagram.	69
3.3.	Epipolar Geometry Between Two Images.	76
3.4.	Correspondence Search Constraint Using Epipolar Lines.	77
4.1.	Overview of Image-Aided Inertial Algorithm	79
4.2.	Stochastic Feature Projection.	84
4.3.	Frequency Response of the Gaussian Filter.	84
4.4.	Impulse Response of the Difference of Gaussian Filter.	85
4.5.	Frequency Response of the Difference of Gaussian Filter.	85
4.6.	Reference Image	86

Figure		Page
4.7.	Sample Image Scale Decomposition	87
4.8.	Image-aided Inertial Navigation Filter Block Diagram.	101
4.9.	Landmark Track State Replacement.	105
4.10.	Monocular Imaging Geometry.	105
4.11.	Binocular Imaging Geometry.	110
4.12.	Binocular Feature Prediction	114
4.13.	Monocular Egomotion Imaging Geometry.	117
5.1.	Image-Aided Inertial Navigation System.	126
5.2.	Test Navigation System Block Diagram.	127
5.3.	Image of Optical Calibration Array.	130
5.4.	Binocular Camera Calibration	134
5.5.	Typical Alignment Accuracy of Optical and Inertial Sensors . .	137
6.1.	Simulated Feature Prediction Flight Path	140
6.2.	Landmark Pixel Location Error and Uncertainty Bound	141
6.3.	Landmark Pixel Location Error and Uncertainty Bound Showing Relationship to Epipolar Search Area	141
6.4.	T-38 Flight Data Collection System	142
6.5.	Three Image Sequence Recorded During T-38 Flight	143
6.6.	Predicted Landmark Location Using Stochastic Projection Method	144
6.7.	Indoor Profile Simulated Position Error Using Tactical-Grade In- ertial Sensor.	147
6.8.	Indoor Profile Simulated Velocity Error Using Tactical-Grade In- ertial Sensor.	148
6.9.	Indoor Profile Simulated Attitude Error Using Tactical-Grade Inertial Sensor.	149
6.10.	Indoor Profile Simulated Position Error Using Consumer-Grade Inertial Sensor.	150
6.11.	Indoor Profile Simulated Velocity Error Using Consumer-Grade Inertial Sensor.	151

Figure		Page
6.12.	Indoor Profile Simulated Attitude Error Using Consumer-Grade Inertial Sensor.	152
6.13.	Indoor Profile Simulated Horizontal Position Errors for Consumer and Tactical-grade Inertial Sensors.	153
6.14.	Indoor Profile Simulated Velocity Errors for Consumer and Tactical-grade Inertial Sensors.	154
6.15.	Indoor Profile Simulated Attitude Errors for Consumer and Tactical-grade Inertial Sensors.	155
6.16.	Outdoor Ground Data Collection Profile.	156
6.17.	Sample Image from Outdoor Data Collection.	157
6.18.	Sample Image from Indoor Data Collection.	159
6.19.	Estimated Path from Indoor Data Collection Using Tactical-grade Inertial Sensor.	161
6.20.	Estimated Path from Indoor Data Collection Using Consumer-grade Inertial Sensor.	162
6.21.	Performance Comparison for Image-aided Tactical and Consumer-grade Inertial Sensors for Indoor Profile.	163
6.22.	Indoor Data Collection Start/Stop Area.	164
6.23.	Estimated Path and Feature Location from Indoor Data Collection for Consumer-grade Inertial Sensor.	165
6.24.	Flight Profile Simulated Horizontal Position Error Using Tactical-Grade Inertial Sensor.	168
6.25.	Flight Profile Simulated Velocity Error Using Tactical-Grade Inertial Sensor.	169
6.26.	Flight Profile Simulated Attitude Error Using Tactical-Grade Inertial Sensor.	170
6.27.	Flight Profile Simulated Horizontal Position Error Using Consumer-Grade Inertial Sensor.	172
6.28.	Flight Profile Simulated Velocity Error Using Consumer-Grade Inertial Sensor.	173

Figure		Page
6.29.	Flight Profile Simulated Attitude Error Using Consumer-Grade Inertial Sensor.	174
7.1.	Multi-sensor imaging from solo vehicle.	180
7.2.	Multi-vehicle cooperative navigation scenario.	181

List of Tables

Table		Page
5.1.	Inertial Measurement Sensor Specifications	129
5.2.	Camera to Inertial Alignment Sensitivity Analysis	136

List of Symbols

Symbol		Page
$\mathbf{f}^b$	specific force vector in the body frame	20
$\mathbf{a}^b$	accelerometer bias vector	20
$\boldsymbol{\omega}_{ib}^b$	angular rate vector relative to inertial	20
$\boldsymbol{\Omega}_{ib}^b$	skew-symmetric angular rate matrix	21
$\mathbf{b}^b$	gyroscopic bias vector	22
$\mathbf{p}^n$	vehicle position vector in navigation frame	22
$\mathbf{v}^n$	vehicle velocity vector in navigation frame	23
$\mathbf{C}_b^n$	body to navigation frame direction cosine matrix	23
$\mathbf{p}_0^e$	Earth-fixed navigation frame origin vector	23
$\mathbf{C}_n^e$	navigation to earth frame orientation direction cosine matrix	23
$\boldsymbol{\Omega}_{nb}^b$	navigation to body skew-symmetric angular rate matrix	23
$\boldsymbol{\omega}_{ie}^e$	Earth's sidereal angular rate vector	23
L	geodetic latitude	25
λ	longitude	25
h	height above the WGS-84 ellipsoid	25
ϕ	roll angle	25
θ	pitch angle	25
ψ	heading	25
N	geodetic radius	26
a	semi-major axis	26
e	ellipsoidal eccentricity	26
$\mathbf{C}_e^{n'}$	Earth to navigation frame direction cosine matrix	26
$\mathbf{v}^{n'}$	velocity in navigation frame	27
$\mathbf{C}_{n'}^b$	navigation to body frame direction cosine matrix	27
$\mathbf{w}_a^b$	accelerometer measurement noise vector	28

Symbol		Page
$\mathbf{w}_b^b$	gyroscopic measurement noise vector	28
τ_a	accelerometer bias time constant	28
τ_b	gyroscopic bias time constant	28
$\mathbf{w}_{a_{bias}}^b$	accelerometer bias driving noise vector	28
$\mathbf{w}_{b_{bias}}^b$	gyroscopic bias driving noise vector	28
$\boldsymbol{\psi}$	attitude error vector	28
$W(\mathbf{p}^e)$	gravity potential function	31
GM	Earth's gravitational constant	31
$\delta \dot{\mathbf{v}}^n$	acceleration error vector	31
$\mathbf{G}$	gravity gradient tensor	32
λ_t	meridian angle	58
ϕ_t	declination angle	58
$\Phi(t_{i+1}, t_i)$	state transition matrix	94
$\mathbf{P}_{xx}(t_i)$	navigation state error covariance	96
σ_h^2	height uncertainty variance	96

List of Abbreviations

Abbreviation		Page
GPS	Global Positioning System	1
NED	North, East and Down	11
DCM	direction cosine matrix	16
WGS-84	World Geodetic System of 1984	25
SIFT	scale-invariant feature tracker	45
EKF	extended Kalman filter	49
TERCOM	terrain-contour matching	62
VLSI	very-large scale integration	63
SIFT	scale-invariant feature tracking	82
CMOS	complementary metal-oxide semiconductor	125
AGL	above ground level	166
PCA	principal component analysis	177

FUSION OF IMAGING AND INERTIAL SENSORS FOR NAVIGATION

I. Introduction

This dissertation outlines a research effort focused on fusing optical and inertial sensors for robust, self-contained, passive, autonomous navigation. This effort is motivated by the requirement for autonomous navigation in environments where external navigation reference sources (such as the Global Positioning System) are unavailable.

Precision navigation is the cornerstone to the modern concepts of precision attack and synergistic combat operations. The Global Positioning System (GPS) was fielded in the 1980's and first used for precision navigation and targeting in combat during the first Gulf War [70]. This new capability of precision attack and maneuver allowed the coalition forces to fight and win the fastest, lowest casualty, most devastatingly destructive one-sided war in recorded history [28]. Precision navigation is a force enabler which combines the classic tenets of mass and surprise in ways which were previously impossible, leading to the development of the doctrine of precision combat [70].

The precision combat revolution has led to new tactics and development of a highly-focused combat force which exploits a minimum of weapons to produce the maximum effect on the enemy. In addition, precision combat creates minimal collateral damage and fratricide [26]. The Air Force's reliance on precision combat tactics, combined with a lack of alternative precision navigation technologies, has led to a dependence on GPS [26, 57]. Unfortunately, GPS requires a direct line of sight to the satellite constellation and is vulnerable to attack or disruption from internal or external sources [26]. Thus, precision combat is currently impossible in areas not serviced by GPS (e.g., under trees, indoors, underwater, underground or in jamming environments).

The motivation of this work is to address the limitations of the current precision navigation methods by fusing optical and inertial systems. This concept is inspired by observing the precision navigation capabilities of animals. Research has indicated that animals utilize visual and inertial observations to navigate with precision in the air, land, and sea [29]. This powerful natural demonstration of navigation principles guides this work, which builds on previous research to make precision navigation using visual and inertial measurements possible for military applications. Not only does the fusion of optical and inertial sensors reduce or eliminate the reliance on GPS, it introduces the overwhelming advantages of precision combat into mission areas which were previously impossible such as urban or underground combat.

A general overview of the available measurements and sensor fusion concepts is presented in the next section.

1.1 Problem Definition

Navigation is defined as the determination of relative position, velocity, and orientation from a predefined reference with respect to time. A trajectory is defined as the continuous time history of position, velocity, and orientation of a body.

In order to estimate a trajectory, sensors are employed to make measurements of physically observable properties of the environment which provide information regarding the navigation state. For this research, three different types of passive sensors are utilized: accelerometers, gyroscopes, and imaging sensors.

1.1.1 Accelerometers. Accelerometers are designed to measure *specific force*, which is the sum of linear acceleration and gravitation. The measurements from real sensors are corrupted by errors which are described in further detail in Chapter II, Section 2.5.1. In general terms, the accelerometer measurement is a nonlinear function of the linear acceleration, gravitation, and measurement error sources such as thermal noise, manufacturing errors, etc. A simple accelerometer measurement model is shown in Figure 1.1.

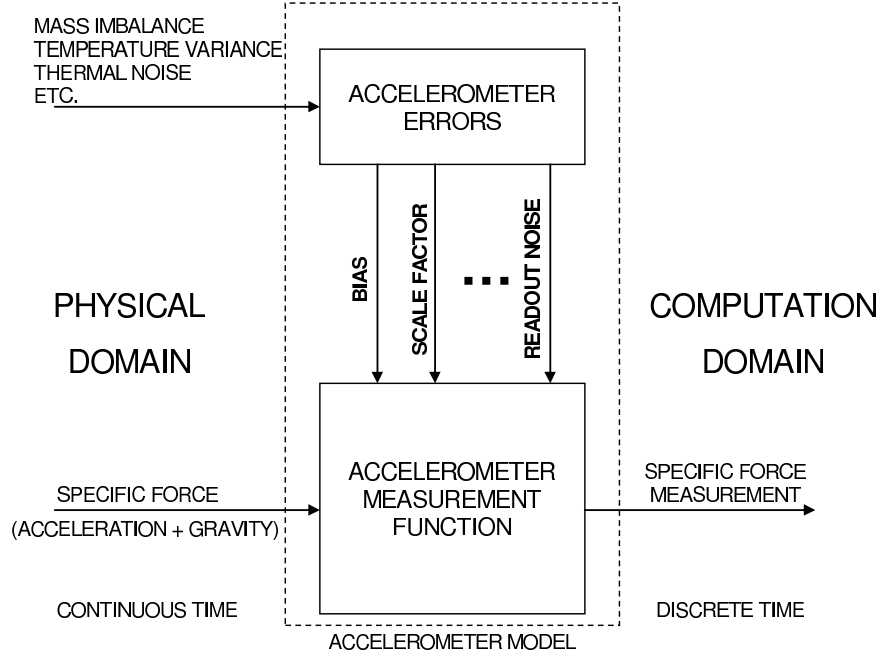


Figure 1.1: General accelerometer block diagram. The accelerometer transforms the specific force into a measurement in the computation domain. The measurement can be corrupted by a number of physical sources.

1.1.2 Gyroscopes. Gyroscopes measure angular rotation rate with respect to an inertial reference. The measurements are corrupted by errors which are discussed in further detail in Chapter II, Section 2.5.2. In the most general terms, the gyroscope measurement is a nonlinear function of the angular rotation rate, attitude, and measurement errors. A simple measurement model for a rate gyroscope is shown in Figure 1.2.

1.1.3 Imaging Sensors. Optical imagers measure the intensity pattern of light focused on an array of light sensors. This two-dimensional intensity pattern is defined as an image. The scene is the set of characteristics of the environment which emit or reflect photons toward the imaging sensor. Imaging sensors are corrupted by errors which are also discussed in further detail in Chapter II, Section 2.7. Thus, in general terms, the image is a nonlinear function of the position, attitude, scene, and measurement errors as shown in Figure 1.3.

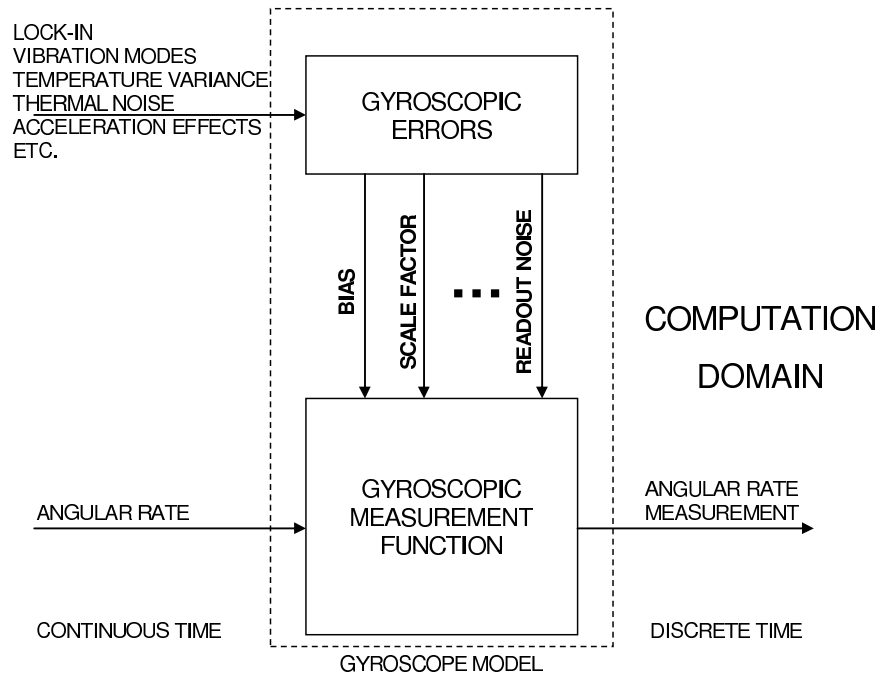


Figure 1.2: General rate gyroscope block diagram. The rate gyroscope transforms the physical angular rotation rate into a measurement in the computation domain. The measurement can be corrupted by a number of physical sources.

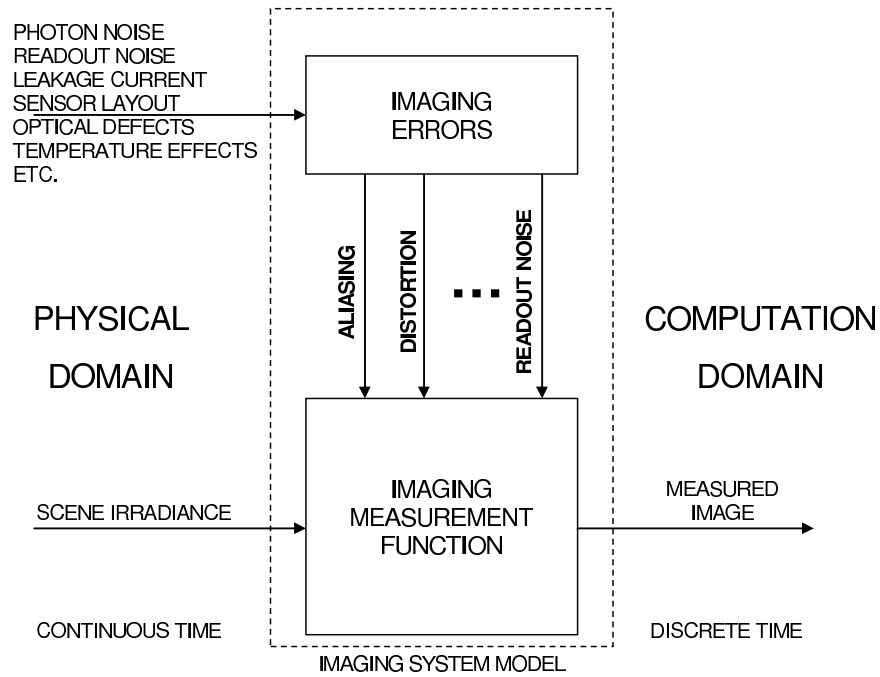


Figure 1.3: General imaging system block diagram. The imaging system transforms the scene irradiance pattern into a measurement in the computation domain or image. The measurement can be corrupted by a number of physical sources.

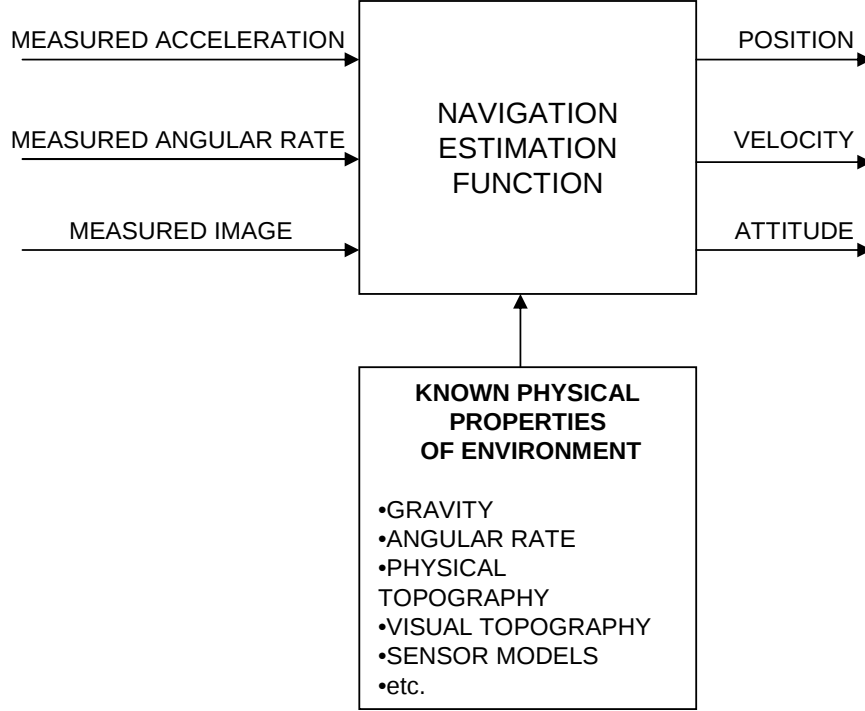


Figure 1.4: Trajectory estimation block diagram. The estimator is a nonlinear function which combines sensor measurements with known properties of the physical environment to determine the navigation state.

1.1.4 Trajectory Estimation. The overall objective of this research is to derive a statistically rigorous, online method to estimate the trajectory, based upon accelerometer, gyroscope, and imaging sensor measurements. In general terms, the trajectory estimator is a nonlinear function which combines measurements from inertial and optical sensors with the known physical properties of the environment to produce an estimate of position, velocity, and attitude. This model is shown in Figure 1.4.

More specifically, the deep integration (i.e., fusion) of optical and inertial sensors are explored and natural synergies between the sensors are exploited [43]. This includes topics such as stochastic projections, optical and inertial calibration, predictive feature tracking algorithms, and long-term navigation. Ultimately, a mathematically rigorous solution is sought which provides a self-contained precision navigation capa-

bility to the Air Force and reduces the reliance on GPS. The specific contributions of this research are presented in the next section.

1.2 Research Contributions

As mentioned in the previous section, the objective of this research is to exploit the natural synergy between imaging and inertial sensors through deep integration techniques, that is, combining image and inertial sensor data at a more fundamental level in order to exploit mutual synergy between the two devices [43]. Specifically, the following contributions to the science of navigation are claimed as a result of this work.

The primary contribution is the definition of the feature space and development of the mathematics required to perform statistically rigorous transformations in this feature space in order to predict the features at a future time. This stochastic projection leverages the collection of inertial measurements between images in order to improve the accuracy and speed of the attendant feature correspondence algorithm. The feature space and the stochastic projection algorithm are developed in a general and specific sense in Chapter IV.

This method of stochastic projections is a key enabler of the second contribution, which is the development of a deeply-integrated, online, image and inertial navigation algorithm [68, 69]. The deep level of integration is a unique feature of this work and results in the integration of the inertial and imaging sensors at a more fundamental level than has been done up to this point, which results in a new level of synergy. This synergy can be summarized in the following manner. Given a series of raw measurements from inertial and imaging sensors, these measurements are filtered at the raw measurement level, using an optimal state estimate derived from the collection of previous measurements. In other words, not only does the imaging sensor help estimate (and correct) errors in the inertial sensor, the inertial sensor helps estimate (and correct) errors in the imaging sensor, or, more specifically, the feature space transformation and of the image. This technique is described in Chapter IV.

In addition to the two primary contributions, two contributions are made in the calibration of imaging and inertial sensors. The first contribution is the addition of self-calibration states to the Kalman filter which allows the imaging and inertial sensor to estimate calibration parameters using known optical references (e.g., celestial bodies and surveyed ground points) and independent measurements of the trajectory, such as GPS. This technique is evaluated using simulation, ground, and flight data. Secondly, the achievable limits of calibration accuracy are examined as a relation to the fundamental limit of sensor accuracy.

The final important contribution of this work is the implementation of the deep-integration theory in an experimental imaging and inertial navigation sensor. The sensor, combined with the algorithms developed in Chapter IV, demonstrated the autonomous operation of an image and inertial sensor in both the ground and flight environment. In addition, the performance improvement under varying navigation scenarios is demonstrated using this novel navigation system.

This dissertation is organized as follows. Chapter II provides a detailed mathematical background for the optical and inertial integration problem, including a description of the notation, primary coordinate frames and transformations, inertial and optical sensor models, kinematic models, and an example of an online estimation algorithm. Chapter III presents methods used to solve similar problems of interest and discusses contributions and potential improvements. Chapter IV derives and presents the candidate image and inertial sensor fusion algorithm. Chapter V provides methodology for calibration and boresight of optical and inertial sensors. Chapter VI describes the experimental data collection system and presents the results. Finally, the conclusions regarding the overall theory and concepts of image and inertial navigation as well as potential areas for future exploration are presented in Chapter VII.

II. Mathematical Background

This chapter reviews the mathematical background required to adequately pose the problem of optical and inertial sensor fusion. The chapter begins by defining a standard mathematical notation which is used throughout the document. Next, the navigation reference frames are defined and a mathematical technique for transforming coordinates between reference frames is defined. The concept of a mathematical error model is presented to provide background for the presentation of a rigorous mathematical model of the components in inertial and optical sensors, including models describing their respective nonlinear properties. Next, a short discussion of the Kalman filter is presented as a commonly implemented online recursive estimation algorithm. In addition, the method of optimal batch estimation for both linear and nonlinear models is presented in order to complement the Kalman filter discussion. Finally, a method to rigorously define perturbations of direction cosine matrices is presented, which is of fundamental importance for the remainder of the document.

2.1 Mathematical Notation

The following mathematical notation is used:

- **Scalars:** Scalars are represented by upper or lower case letters in italic type, (e.g. x or G)
- **Vectors:** Vectors are denoted by lower case letters with bold font, (e.g. $\mathbf{x}$ or ϕ .) The vector $\mathbf{x}$ is composed of a column of scalar elements x_i , where i is the element number.
- **Unit Vectors:** Vectors with unit length, as defined by the two norm, are denoted by the *check* character, as in $\|\check{\mathbf{x}}\| \equiv 1$.
- **Matrices:** Matrices are denoted by upper case letters in bold font. For example, the matrix $\mathbf{X}$ is composed of elements X_{ij} where i is the row index and j is the column index.

- **Vector Transpose:** The vector (or matrix) transpose is identified by a superscript Roman T , as in $\mathbf{x}^T$.
- **Estimated variables:** Variables which are estimates of random variables are denoted by the *hat* character, as in $\hat{\mathbf{x}}$.
- **Computed Variables:** Variables which are corrupted by errors are denoted by the *tilde* character, as in $\tilde{\mathbf{x}}$.
- **Homogeneous Coordinates:** Vectors in homogeneous coordinate form are denoted by an *underline*, as in $\underline{\mathbf{x}}$. Homogeneous coordinate vectors are defined to have a value of 1 in the last element.
- **Direction Cosine Matrices:** Direction cosine matrices from frame a to frame b are denoted by $\mathbf{C}_a^b$.
- **Reference Frame:** If a vector is expressed in a specific reference frame, a superscript letter is used to designate the reference frame (e.g., $\mathbf{p}^a$ is a vector in the a frame.)
- **Relative Position or Motion:** In cases where it is important to specify relative motion, combined subscript letters are used to designate the frames (e.g., $\boldsymbol{\omega}_{ab}$ represents the rotation rate vector *from* frame a *to* frame b .)

2.2 Reference Frames

The concept of *navigation reference frames* is an important fundamental for expressing the position, velocity, and orientation of a body. For this research, the following reference frames are defined based on those presented in [7] and [64]:

- True inertial frame (*I-frame*)
- Earth-fixed inertial frame (*i-frame*)
- Earth-centered Earth-fixed frame (*e-frame*)
- Navigation frame (*n'-frame*)

- Earth-fixed navigation frame (*n-frame*)
- Body frame (*b-frame*)
- Binocular disparity frame (*c₀-frame*)
- Camera frame (*c-frame*)

Each of these are described in the paragraphs that follow.

The true inertial frame (*I-frame*) is a theoretical reference frame where Newton's laws of motion apply. The frame is defined by an non-accelerating, non-rotating orthonormal basis in $\mathbb{R}^3$. Because of the relative nature of the universe, the true inertial frame has no predefined origin or orientation.

The Earth-centered inertial frame (*i-frame*) is an orthonormal basis in $\mathbb{R}^3$, with origin at the center of mass of the Earth. The *x* and *y* axes are located on the equatorial plane, aligned with so-called *fixed* stars. The *i-frame* is a non-rotating frame but does accelerate with respect to the true inertial frame, because the Earth revolves about the sun, the sun revolves about our galaxy, etc. For terrestrial navigation purposes, however, it can be considered an inertial reference.

The Earth-centered Earth-fixed frame (*e-frame*) is an orthonormal basis in $\mathbb{R}^3$, with origin also at the Earth's center of mass. The *e-frame* is rigidly attached to the Earth, with the *x* axis on the equatorial plane pointing toward the Greenwich meridian, the *z* axis aligned with the north pole, and the *y* axis on the equatorial plane pointing toward 90 *degrees* east longitude. Because the *e-frame* is a true Cartesian reference frame, some navigation computations are simplified.

The vehicle-fixed navigation frame (*n'-frame*) is an orthonormal basis in $\mathbb{R}^3$, with origin located at a predefined point on a vehicle (e.g., the vehicle's center of gravity or the center of a triad of inertial sensors, etc.) The vehicle-fixed navigation frame's *x*, *y*, and *z* axes point in the north, east and down (NED) directions, respectively. Although the concept of *down* is open to interpretation, for the purposes of this research down is defined as the direction a plumb line would point due to gravity. The *n'-frame*

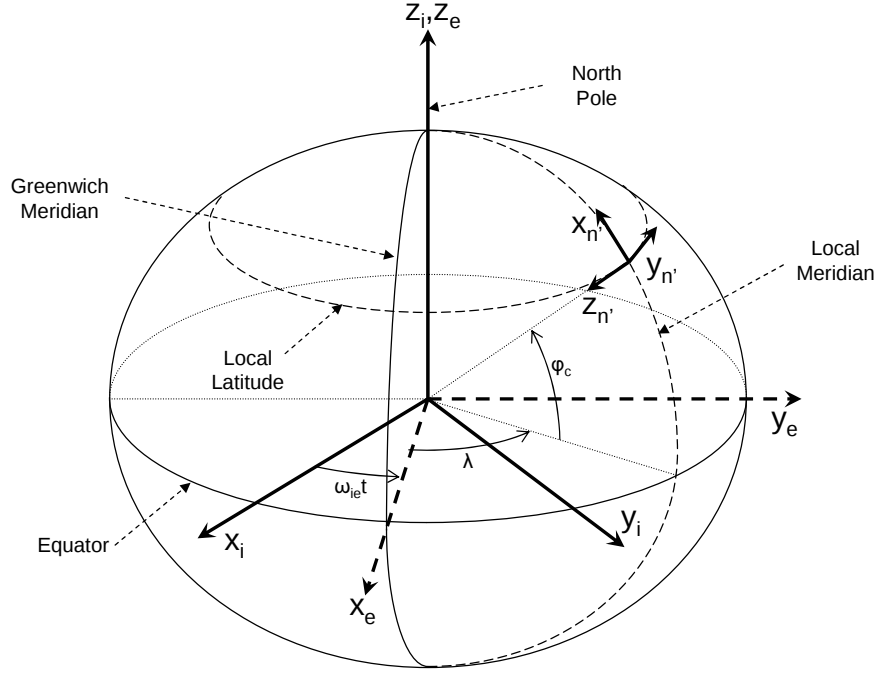


Figure 2.1: Inertial, Earth and vehicle-fixed navigation frame. The inertial and Earth frames originate at the Earth's center of mass while the vehicle-fixed navigation frame's origin is located at a fixed location on a vehicle.

rotates with respect to the e -frame due to translational motion of the vehicle. The i , e and n' -frames are illustrated in Figure 2.1. The n -frame is illustrated in Figure 2.2.

The Earth-fixed navigation frame (n -frame) is an orthonormal basis in $\mathbb{R}^3$, with origin located at a predefined location on the Earth, typically on the surface. The Earth-fixed navigation frame's x , y , and z axes point in the north, east, and down (NED) directions relative to the origin, respectively. As in the previous case, down is defined as the direction of the gravity vector. In contrast to the navigation frame, the Earth-fixed navigation frame remains fixed to the surface of the Earth. While this frame is not useful for very-long distance navigation, it can simplify the navigation kinematic equations for local navigation routes.

The body frame (b -frame) is an orthonormal basis in $\mathbb{R}^3$, rigidly attached to the vehicle with origin co-located with the navigation frame. The x , y , and z axes point out the nose, right wing, and bottom of an aircraft, respectively. Strapdown inertial

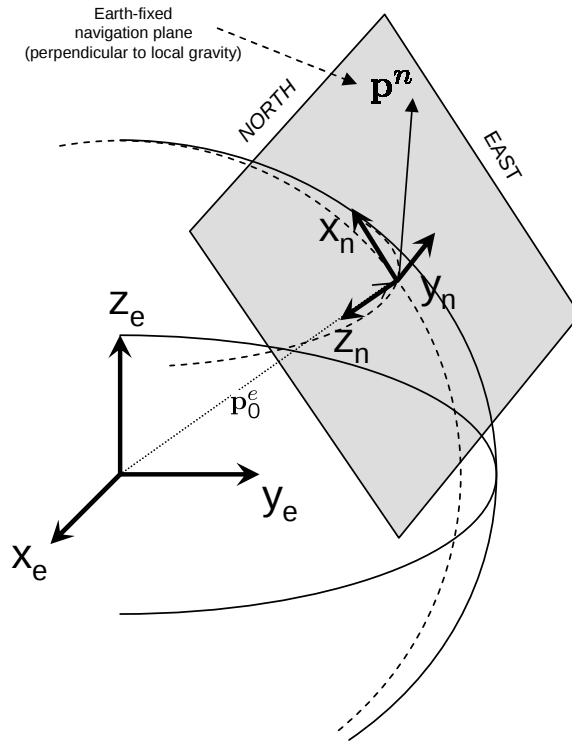


Figure 2.2: Earth-fixed navigation frame. The Earth-fixed navigation frame is a Cartesian reference frame which is perpendicular to the gravity vector at the origin and fixed to the Earth.

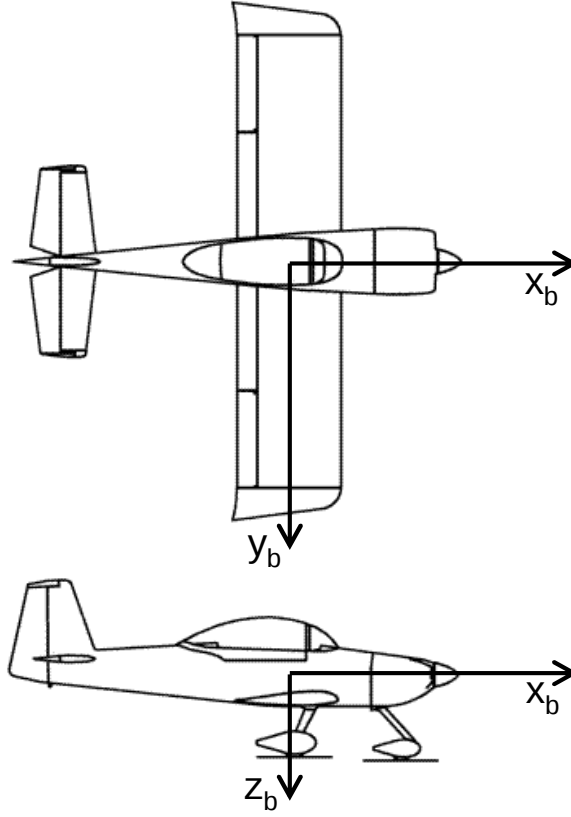


Figure 2.3: Aircraft body frame illustration. The aircraft body frame originates at the aircraft center of gravity.

sensors are fixed to the b -frame, although they may not be located at the origin or aligned with the axes. The b -frame is shown in Figure 2.3.

The camera frame (c -frame) is an orthonormal basis in $\mathbb{R}^3$, rigidly attached to a camera, with origin at the camera's optical center. The x and y axes point up and to the right, respectively, and are parallel to the image plane of the camera. The z axis points out of the camera perpendicular to the image plane. The c -frame is shown in Figure 2.4.

The binocular disparity frame (c_0 -frame) is an orthonormal basis in $\mathbb{R}^3$, which is rigidly attached to the lever arm located between cameras in a binocular configuration, with origin at a specified point on the lever arm. The x , y , and z axes point forward, right, and down, respectively. The c_0 -frame is shown in Figure 2.5.

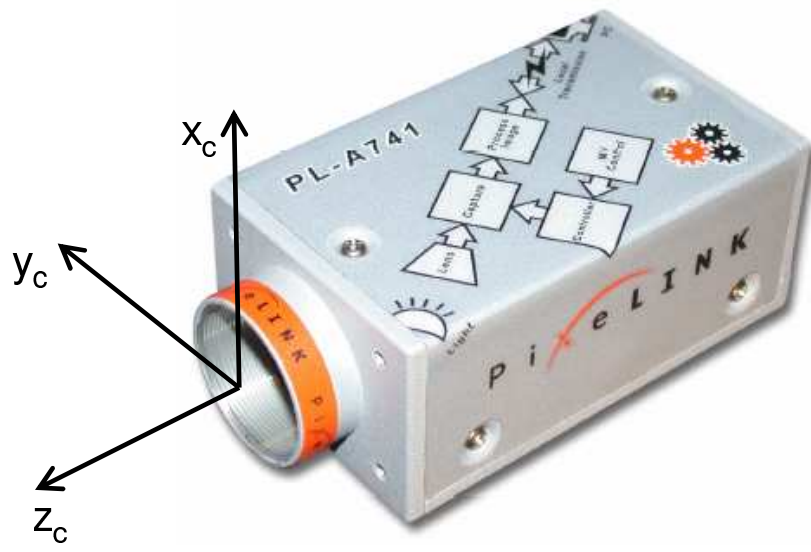


Figure 2.4: Camera frame illustration. The camera reference frame originates at the optical center of the lens.

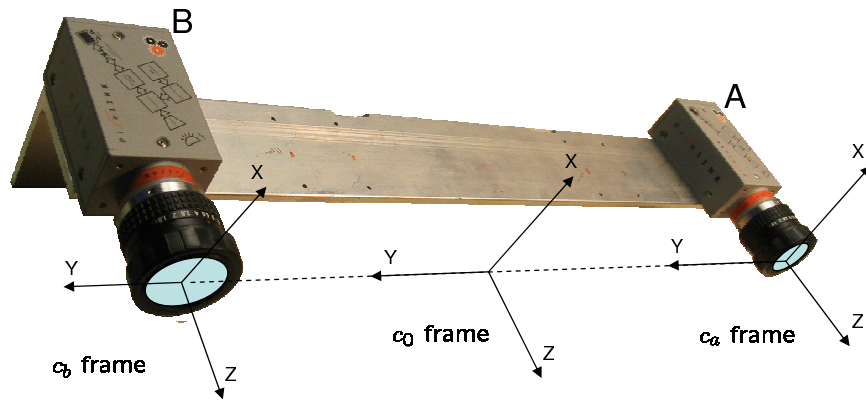


Figure 2.5: Binocular disparity frame illustration. The binocular disparity frame originates at the midpoint between the optical center of the two camera frames, c_a and c_b .

2.3 *Coordinate Transformations*

Coordinate transformations describe the relationships between vectors expressed in various reference frames. As discussed in [64], the available transformations are classified as having either three or four-parameters. Three parameter transformations suffer from singularities at certain orientations, whereas four parameter transformations do not. For the purposes of this research, the relevant transformations are Euler angles and direction cosine matrices.

Euler angles are elements of a three-component vector corresponding to a specific sequence of single-axis rotations to rotate from one reference frame to another. A common application of Euler angles is expressing the transformation from the body to navigation frame of an aircraft as a roll, pitch and yaw angle. In this example, the singularities can be seen at pitch angles of $\pm 90 \text{ deg}$.

Direction cosine matrices (DCM) are four-parameter transformations expressed as a 3×3 matrix. The matrix consists of the inner product (or *Cosines*) of each unit basis vector in one frame with each unit basis vector in another frame. The matrix form of the direction cosine is convenient for transforming vectors, as in:

$$\mathbf{p}^b = \mathbf{C}_a^b \mathbf{p}^a \quad (2.1)$$

where $\mathbf{C}_a^b$ is defined as the matrix of the inner products of the unit basis vectors. Each element ($C_{a(ij)}^b$) corresponds to the scalar projection (or *cosine*) of the i -th axis of the a frame onto the j -th axis of the b frame.

When used to transform right-hand Cartesian coordinates, the DCM has the following properties:

1. $\text{Det}(\mathbf{C}_a^b) \equiv |\mathbf{C}_a^b| = 1$
2. $\mathbf{C}_b^a = (\mathbf{C}_a^b)^{-1} = (\mathbf{C}_a^b)^T$
3. $\mathbf{C}_a^c = \mathbf{C}_b^c \mathbf{C}_a^b$

The time derivative of the direction cosine matrix is defined as [64]

$$\dot{\mathbf{C}}_a^b = \mathbf{C}_a^b \boldsymbol{\Omega}_{ba}^a \quad (2.2)$$

where $\boldsymbol{\Omega}_{ba}^a$ is the skew-symmetric form of the rotation vector from frame b to frame a , expressed in the a frame ($\boldsymbol{\omega}_{ab}^a$). The skew-symmetric operator of a vector $\boldsymbol{\omega}$ is defined as

$$(\boldsymbol{\omega} \times) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (2.3)$$

where ω_x , ω_y , and ω_z are the scalar elements of $\boldsymbol{\omega}$.

The above development allows expression of the navigation state (position, velocity and orientation) in a convenient vector notation. In addition, vectors can be easily transformed between reference frames, even when the frame is rotating. In the next section, the mathematics of perturbation analysis and error models are presented.

2.4 Error Models and Perturbation Analysis

In many cases, properties of physical systems are well-modeled using systems of nonlinear differential equations [37]. Consider the homogeneous nonlinear differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t]; \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (2.4)$$

For a given input function, $\mathbf{u}_0(t)$, and initial condition, $\mathbf{x}_0$, there exists a nominal solution trajectory:

$$\bar{\mathbf{x}}(t); \quad t \in [t_0, \infty) \quad (2.5)$$

Given this nominal solution trajectory, the effects of small changes (i.e., *perturbations*) in the initial conditions or input function can be expressed using a Taylor series expansion about the nominal solution trajectory. These perturbations are denoted

using the following notation:

$$\mathbf{x}(t) = \bar{\mathbf{x}}(t) + \delta\mathbf{x}(t); \quad t \in [t_0, \infty) \quad (2.6)$$

$$\mathbf{u}(t) = \bar{\mathbf{u}}(t) + \delta\mathbf{u}(t); \quad t \in [t_0, \infty) \quad (2.7)$$

The Taylor series approximation about the nominal is then

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t] + \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{B}(t)\delta\mathbf{u}(t) + \boldsymbol{\varepsilon}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t) \quad (2.8)$$

where $\boldsymbol{\varepsilon}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t) \rightarrow 0$ as $\delta\mathbf{x} \rightarrow 0$ and $\delta\mathbf{u} \rightarrow 0$ and

$$\mathbf{F}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{array} \right] \bigg|_{\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t} \quad (2.9)$$

$$\mathbf{B}(t) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_n} \end{array} \right] \bigg|_{\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t), t} \quad (2.10)$$

By neglecting the $\boldsymbol{\varepsilon}$ term and subtracting the nominal solution, a time-varying linear differential equation remains. This equation is an approximation of the difference from the nominal trajectory and is called the *linear perturbation equation* [37]:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{B}(t)\delta\mathbf{u}(t) \quad (2.11)$$

A time-varying linear differential equation has a unique solution, which becomes explicit when $\mathbf{F}(t)$ and $\mathbf{B}(t)\delta\mathbf{u}(t)$ are piecewise continuous [37]. Under these conditions, the solution to the time-varying linear differential perturbation equation (2.11) is given by

$$\delta\mathbf{x}(t) = \boldsymbol{\Phi}(t, t_0)\delta\mathbf{x}_0 + \int_{t_0}^t \boldsymbol{\Phi}(t, \tau)\mathbf{B}(\tau)\delta\mathbf{u}(\tau)d\tau \quad (2.12)$$

where $\boldsymbol{\Phi}(t, t_0)$ is the state transition matrix from time t_0 to time t .

The perturbation solution's linear form allows the use of linear estimation methods (e.g., Kalman filtering) and analysis tools which is discussed in more detail in Section 2.8.

2.5 *Inertial Navigation*

In this section, basic inertial navigation concepts are presented, including a discussion of the strapdown navigation sensors, the measurements available, a derivation of the system dynamics, and the resulting differential error equations.

2.5.1 Accelerometer Sensor Model. Strapdown inertial navigation systems consist of accelerometers and gyroscopes, usually mounted in orthogonal triads. Contrary to their name, accelerometers do not directly measure acceleration. According to Einstein's Theory of Special Relativity [9], acceleration due to motion relative to inertial space and acceleration due to gravitation (or mass attraction) are indistinguishable. Thus, accelerometers produce a measurement of *specific force*, which is the difference between the acceleration relative to inertial space and the gravitation vector, defined by

$$\mathbf{f} = \ddot{\mathbf{p}} - \mathbf{g} \quad (2.13)$$

where $\mathbf{f}$ is the specific force vector, $\mathbf{p}$ is the position vector, and $\mathbf{g}$ is the gravitation vector.

The accelerometer output is corrupted by a number of error sources, which are a function of the design of the sensor [64]. In general terms, the dominant accelerometer errors are as follows:

- **Bias:** A bias is a constant or slowly-varying additive error. Some bias components can be measured and corrected through factory calibration techniques, although others will inevitably remain. Biases can change after power cycles or after temperature fluctuations.

- **Scale Factor:** A scale factor error is a constant or slowly-varying multiplicative error. As with bias errors, some scale factor effects can be corrected through calibration.
- **Sensor Misalignment:** Sensor misalignment errors are a result of mechanical fabrication and installation errors. These errors result in a difference between the accelerometers' sensitive axes and the platform reference.
- **Vibration:** These errors cause a measurement bias as a function of certain vibrational modes.
- **Measurement Noise:** Measurement noise is observed as an additive error component with high-bandwidth power spectral density. This noise component is the theoretical result of many high-bandwidth sources such as electrical noise, thermal noise, etc.
- **Gravity Model Errors:** Because the accelerometer measures specific force, the acceleration due to local gravity must be added to the accelerometer output to produce an estimate of acceleration in the inertial frame. Errors in this local gravity model produce additive errors in acceleration. Note the gravity model errors are not errors in the accelerometer measurement itself, however they arise when converting from specific force to acceleration.

The above error sources can be modeled as a quadratic function. Thus, the measured accelerometer output in the body reference frame, $\mathbf{f}_m^b$, is

$$\mathbf{f}_m^b = \mathbf{f}^b + \mathbf{A}_1 \mathbf{f}^b + \mathbf{A}_2^T (\mathbf{f}^b)^T \mathbf{f}^b \mathbf{A}_2 + \mathbf{a}^b + \boldsymbol{\eta}_a^b \quad (2.14)$$

where the $\mathbf{f}^b$ vector is the true specific force in the body frame, $\mathbf{A}_1$ and $\mathbf{A}_2$ are the combined accelerometer first and second-order error component matrices, $\mathbf{a}^b$ is the bias vector, and $\boldsymbol{\eta}_a^b$ is the additive measurement noise vector.

2.5.2 Gyroscope Sensor Model. The gyroscopes used in strapdown navigation systems measure angular rate relative to inertial space, $\boldsymbol{\omega}_{ib}^b$. By rearranging

Eqn. (2.2), the measurement can be expressed using the inertial to body direction cosine matrix as

$$\boldsymbol{\Omega}_{ib}^b = \mathbf{C}_i^b \dot{\mathbf{C}}_b^i \quad (2.15)$$

where $\boldsymbol{\Omega}_{ib}^b$ is the skew-symmetric form of the angular rate vector, $\boldsymbol{\omega}_{ib}^b$. The skew symmetric form is defined in Eqn. 2.3.

Gyroscopic sensors are subject to errors which corrupt the angular rate measurement [64]. The dominant gyroscopic errors are as follows:

- **Fixed Bias (g -independent):** This bias is a constant or slowly-varying additive error which is independent of acceleration.
- **Acceleration-dependent Bias:** This is a bias which is a function of the current acceleration applied to the sensor. This bias has components which act both in the gyroscope's sensitive axis as well as orthogonal to the sensitive axis.
- **Anisoelastic Bias:** The anisoelastic bias component is proportional to the product of acceleration along orthogonal axes.
- **Scale Factor:** A scale factor error is a constant or slowly-varying multiplicative error.
- **Sensor Misalignment:** Sensor misalignment errors are a result of mechanical fabrication and installation errors. These errors result in a difference between the gyroscopes' sensitive axes and the platform reference.
- **Measurement Noise:** Measurement noise is observed as an additive error component with high-bandwidth power spectral density. This noise component is the theoretical result of many high-bandwidth sources such as electrical noise, thermal noise, etc.

Similarly to the accelerometer errors, the gyroscopic errors can also be modeled as a quadratic function. Thus, the measured angular rate output in the body reference

frame, $\boldsymbol{\omega}_{ib_m}^b$, is

$$\begin{aligned}\boldsymbol{\omega}_{ib_m}^b &= \boldsymbol{\omega}_{ib}^b + \mathbf{B}_1 \boldsymbol{\omega}_{ib_m}^b + \mathbf{B}_2^T (\boldsymbol{\omega}_{ib_m}^b)^T \boldsymbol{\omega}_{ib_m}^b \mathbf{B}_2 \\ &\quad + \mathbf{B}_3 \mathbf{f}^b + \mathbf{b}^b + \boldsymbol{\eta}_b^b\end{aligned}\tag{2.16}$$

where $\mathbf{B}_1$ and $\mathbf{B}_2$ are the combined gyroscopic first and second order error components, $\mathbf{B}_3$ is the acceleration-dependent bias term, $\mathbf{b}^b$ is the bias, and $\boldsymbol{\eta}_b^b$ is the additive measurement noise.

2.5.3 Inertial Navigation Dynamics. In this section, the nonlinear dynamic equations governing a strapdown inertial navigation system are developed from the basic navigation equations. These dynamic equations are used in the next section to derive a linear perturbation error model.

The error equations are developed in the Earth-fixed navigation frame defined in Section 2.2. The Earth-fixed navigation frame is preferred for the following reasons:

- The Earth-fixed navigation frame is a Cartesian space in $\mathfrak{R}^3$ which makes the units consistent between axes and is conducive to good scaling.
- The Earth-fixed navigation frame results in a simplified dynamic model with minimal loss of accuracy over short distances.
- Earth-fixed navigation frame cartesian navigation state parameters can be easily transformed to geodetic coordinates (i.e., latitude, longitude, and altitude) for interpretation and analysis.

The navigation states of interest are measures of position, velocity, and orientation with respect to the navigation frame, defined as:

- The $\mathbf{p}^n$ vector is the three-dimensional position of the vehicle relative to the origin of the Earth-fixed navigation frame, expressed in the Earth-fixed navigation frame.

- The $\mathbf{v}^n$ vector is the three-dimensional velocity of the vehicle relative to the Earth-fixed navigation frame, expressed in the Earth-fixed navigation frame.
- The $\mathbf{C}_b^n$ matrix is the DCM which transforms vectors from the body frame to the Earth-fixed navigation frame.

The origin location of the Earth-fixed navigation frame is defined relative to the ECEF frame as $\mathbf{p}_0^e$. The orientation of the navigation frame is defined by the direction cosine matrix, $\mathbf{C}_n^e$. Because the Earth-fixed navigation frame is, by definition, rigidly attached to the Earth, the position and orientation are static, e.g. $\dot{\mathbf{p}}_0^e = \mathbf{0}$ and $\dot{\mathbf{C}}_n^e = \mathbf{0}$.

2.5.4 Attitude Dynamics. Applying the definition of the derivative of a DCM provided in Eqn. (2.2) to $\mathbf{C}_b^n$ yields

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \quad (2.17)$$

where $\boldsymbol{\Omega}_{nb}^b$ is the Earth-fixed navigation frame-to-body frame angular rate, expressed in the b frame and in skew-symmetric form. In vector form, this vector is the difference between the body-to-inertial angular rate ($\boldsymbol{\omega}_{ib}^b$), the Earth's sidereal angular rate ($\boldsymbol{\omega}_{ie}^e$), and the angular rate of the navigation frame ($\boldsymbol{\omega}_{en}^n$), all expressed in the b frame:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b \boldsymbol{\omega}_{en}^n - \mathbf{C}_n^b \mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e \quad (2.18)$$

By definition, the angular rate of the navigation frame relative to the Earth, ($\boldsymbol{\omega}_{en}^n$), is zero, which simplifies the above equation to

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b \mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e \quad (2.19)$$

Converting Eqn. (2.19) to skew-symmetric form, substituting into (2.17), and simplifying yields the time-varying attitude dynamic equation

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{ib}^b - \mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \mathbf{C}_b^n \quad (2.20)$$

2.5.5 *Position and Velocity Dynamics.* The position and velocity dynamics are calculated by first expressing the position in the i frame:

$$\mathbf{p}^i = \mathbf{C}_e^i [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] \quad (2.21)$$

where $\mathbf{p}_0^e$ is the location of the origin of the navigation frame, with respect to the e -frame, and $\mathbf{C}_n^e$ is the DCM from the navigation frame to the e -frame. Taking the derivative of 2.21 using the chain rule yields

$$\dot{\mathbf{p}}^i = \mathbf{C}_e^i \mathbf{C}_n^e \dot{\mathbf{p}}^n + \mathbf{C}_e^i \boldsymbol{\Omega}_{ie}^e [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] \quad (2.22)$$

An additional derivative yields the acceleration with respect to the inertial frame:

$$\ddot{\mathbf{p}}^i = \mathbf{C}_e^i \mathbf{C}_n^e \ddot{\mathbf{p}}^n + 2\mathbf{C}_e^i \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \dot{\mathbf{p}}^n + \mathbf{C}_e^i (\boldsymbol{\Omega}_{ie}^e)^2 [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] \quad (2.23)$$

Recalling the definition of specific force from Eqn. (2.13) and substituting into (2.23) yields

$$\mathbf{f}^i + \mathbf{g}^i = \mathbf{C}_e^i \mathbf{C}_n^e \ddot{\mathbf{p}}^n + 2\mathbf{C}_e^i \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \dot{\mathbf{p}}^n + \mathbf{C}_e^i (\boldsymbol{\Omega}_{ie}^e)^2 [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] \quad (2.24)$$

Solving for the acceleration with respect to the Earth-fixed navigation frame ($\ddot{\mathbf{p}}^n$) results in

$$\ddot{\mathbf{p}}^n = \mathbf{f}^n - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \dot{\mathbf{p}}^n - \mathbf{C}_e^n (\boldsymbol{\Omega}_{ie}^e)^2 [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] + \mathbf{g}^n \quad (2.25)$$

The velocity with respect to the Earth-fixed navigation frame is defined as

$$\dot{\mathbf{p}}^n = \mathbf{v}^n \quad (2.26)$$

Substituting (2.26) into (2.24) and expressing the specific force in the b frame yields the time-varying position and velocity dynamic equation:

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \dot{\mathbf{p}}^n - \mathbf{C}_e^n (\boldsymbol{\Omega}_{ie}^e)^2 [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] + \mathbf{g}^n \quad (2.27)$$

In the next section, these dynamic equations are used to derive error dynamics using perturbation methods.

2.5.6 Development of Observation Equations. The development of the strap-down INS dynamics is completed with respect to the Earth-fixed navigation frame. It is necessary in some cases to determine the relationship between coordinates expressed in the Earth-fixed navigation frame and a global reference system, which motivates the development of an appropriate coordinate system transformation. In this case, the well-known World Geodetic System of 1984 (WGS-84) is used. The parameters of interest are defined as

- L is the *geodetic latitude* in radians.
- λ is the *longitude* in radians.
- h is the *height above the WGS-84 ellipsoid* in meters.
- $\mathbf{v}^{n'}$ is the *velocity vector* with respect to the Earth, expressed in the navigation frame.
- ϕ, θ and ψ are the *roll, pitch* and *yaw* angles (in radians) to rotate from the navigation frame to the body frame.

Because the WGS-84 system origin is located at the center of the Earth [7], the Earth-fixed navigation frame parameters are transformed to the Earth-centered

Earth-fixed reference frame using the following transformations:

$$\mathbf{p}^e = \mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n \quad (2.28)$$

$$\mathbf{v}^e = \mathbf{C}_n^e \mathbf{v}^n \quad (2.29)$$

$$\mathbf{C}_b^e = \mathbf{C}_n^e \mathbf{C}_b^n \quad (2.30)$$

The observed parameters are calculated in two steps. Longitude is calculated directly using the four-quadrant arctangent function and elements of the ECEF position vector, $\mathbf{p}^e$, as

$$\lambda = \arctan\left(\frac{p_y^e}{p_x^e}\right) \quad (2.31)$$

The geodetic latitude and height above ellipsoid are calculated by solving the following nonlinear inverse relationships, assuming a WGS-84 ellipsoidal Earth model [7]:

$$(p_x^e)^2 + (p_y^e)^2 = [N + h]^2 \cos^2 L \quad (2.32)$$

$$p_z^e = [N(1 - e^2) + h] \sin L \quad (2.33)$$

where N is the *geodetic radius of meridian* in meters, defined by

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 L}} \quad (2.34)$$

In the above equation, a is the WGS-84 semi-major axis, and e is the WGS-84 ellipsoid eccentricity [7]. The navigation frame velocities are computed using the earth frame to navigation frame (North-East-Down) transformation ($\mathbf{C}_e^{n'}$) [64]:

$$\mathbf{C}_e^{n'} = \begin{bmatrix} -\sin L \cos \lambda & -\sin L \sin \lambda & \cos L \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos L \cos \lambda & -\cos L \sin \lambda & -\sin L \end{bmatrix} \quad (2.35)$$

The navigation frame velocity ($\mathbf{v}^{n'}$) is simply the matrix multiplication of the earth frame to navigation frame DCM and the earth frame velocity vector:

$$\mathbf{v}^{n'} = \mathbf{C}_e^{n'} \mathbf{v}^e \quad (2.36)$$

To obtain the Euler angles, the direction cosine matrix from the navigation frame to the body frame ($\mathbf{C}_{n'}^b$) is calculated as

$$\mathbf{C}_{n'}^b = \mathbf{C}_e^b (\mathbf{C}_e^{n'})^T \quad (2.37)$$

The $\mathbf{C}_{n'}^b$ matrix can be expressed equivalently composed of Euler angles [64]:

$$\mathbf{C}_{n'}^b = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \cos \theta \sin \phi \\ \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.38)$$

where ϕ , θ , and ψ are the roll, pitch, and yaw Euler angles, respectively. Hence, the Euler angles can be computed using terms from $\mathbf{C}_{n'}^b$:

$$\phi = \arctan \left(\frac{\mathbf{C}_{n'23}^b}{\mathbf{C}_{n'33}^b} \right) \quad (2.39)$$

$$\theta = -\arcsin \left(\mathbf{C}_{n'13}^b \right) \quad (2.40)$$

$$\psi = \arctan \left(\frac{\mathbf{C}_{n'12}^b}{\mathbf{C}_{n'11}^b} \right) \quad (2.41)$$

Note that the use of Euler angles causes a singularity when $\theta = \pm \frac{\pi}{2}$, as expected.

2.6 Inertial Navigation Error Model

As shown in the previous section, the inertial navigation dynamics are expressed as nonlinear differential equations. Analyzing the differences between the output of an inertial navigation system and the true trajectory motivates the development of

an inertial navigation error model. This perturbation model can be used to estimate the vehicle's true trajectory by fusing optical and inertial sensors.

2.6.1 Inertial Sensor Error Model. For this derivation, the accelerometer and gyroscopic errors are modeled as a bias plus a random noise. The biases are modeled as first-order Gauss-Markov processes [37]. The random measurement noise is modeled as an additive white Gaussian noise process. The resulting accelerometer measurement model becomes

$$\mathbf{f}_m^b = \mathbf{f}^b + \mathbf{a}^b + \mathbf{w}_a^b \quad (2.42)$$

where $\mathbf{a}^b$ is the accelerometer bias and $\mathbf{w}_a^b$ is an additive white Gaussian noise process.

Similarly, the resulting gyroscope measurement model is

$$\boldsymbol{\omega}_{ib_m}^b = \boldsymbol{\omega}_{ib}^b + \mathbf{b}^b + \mathbf{w}_b^b \quad (2.43)$$

where $\mathbf{b}^b$ is the gyroscope measurement bias and $\mathbf{w}_b^b$ is an additive white Gaussian noise process.

The bias is a first-order Gauss-Markov process which is expressed by the following differential equations

$$\dot{\mathbf{a}}^b = -\frac{1}{\tau_a} \mathbf{a}^b + \mathbf{w}_{a_{bias}}^b \quad (2.44)$$

$$\dot{\mathbf{b}}^b = -\frac{1}{\tau_b} \mathbf{b}^b + \mathbf{w}_{b_{bias}}^b \quad (2.45)$$

The time constants are represented by τ_a and τ_b for the accelerometer and gyroscopic biases, respectively. The processes are driven by the white noise terms, $\mathbf{w}_{a_{bias}}^b$ and $\mathbf{w}_{b_{bias}}^b$.

2.6.2 Attitude Error Development. The development of the attitude error dynamics begins with a definition of the error states. The attitude error vector ($\boldsymbol{\psi}$)

is modeled as a vector of small angles about the north, east and down axes of the navigation frame (ψ_n, ψ_e , and ψ_d , respectively) [71]. The attitude error vector is defined as

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_n \\ \psi_e \\ \psi_d \end{bmatrix} \quad (2.46)$$

Because the angular errors are assumed to be small, the computed body-to-Earth-fixed navigation frame DCM can be expressed as [64]

$$\tilde{\mathbf{C}}_b^n \approx [\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n \quad (2.47)$$

where $(\boldsymbol{\psi} \times)$ is the skew-symmetric form of $\boldsymbol{\psi}$. Taking the derivative of (2.47) with respect to time yields

$$\dot{\tilde{\mathbf{C}}}_b^n = -(\dot{\boldsymbol{\psi}} \times) \mathbf{C}_b^n + [\mathbf{I} - (\boldsymbol{\psi} \times)] \dot{\mathbf{C}}_b^n \quad (2.48)$$

Substituting (2.17) into (2.48) yields

$$\tilde{\mathbf{C}}_b^n \tilde{\boldsymbol{\Omega}}_{nb}^b = -(\dot{\boldsymbol{\psi}} \times) \mathbf{C}_b^n + [\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \quad (2.49)$$

Solving for $(\dot{\boldsymbol{\psi}} \times)$ results in

$$(\dot{\boldsymbol{\psi}} \times) = [\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \mathbf{C}_n^b - \tilde{\mathbf{C}}_b^n \tilde{\boldsymbol{\Omega}}_{nb}^b \mathbf{C}_n^b \quad (2.50)$$

The calculated body-to-earth-fixed navigation frame rotation rate vector, $\tilde{\boldsymbol{\omega}}_{nb}^b$, is defined as

$$\tilde{\boldsymbol{\omega}}_{nb}^b = \boldsymbol{\omega}_{ib_m}^b - \tilde{\mathbf{C}}_n^b \mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e \quad (2.51)$$

where $\boldsymbol{\omega}_{ib_m}^b$ is the body-to-inertial angular rate vector measured by the gyroscopes.

Substituting (2.47), (2.51), and (2.43) into (2.50) results in

$$\begin{aligned}
\left(\dot{\boldsymbol{\psi}} \times\right) &= [\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \mathbf{C}_n^b \\
&\quad - [\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n [\boldsymbol{\Omega}_{nb}^b + (\mathbf{b}^b \times) - [(\mathbf{C}_n^b [\boldsymbol{\psi} \times] \mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e) \times] \\
&\quad + (\mathbf{w}_b^b \times)] \mathbf{C}_n^b
\end{aligned} \tag{2.52}$$

Simplifying by eliminating common terms yields

$$\begin{aligned}
\left(\dot{\boldsymbol{\psi}} \times\right) &= -[\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n [(\mathbf{b}^b \times) - [(\mathbf{C}_n^b [\boldsymbol{\psi} \times] \mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e) \times] \\
&\quad + (\mathbf{w}_b^b \times)] \mathbf{C}_n^b
\end{aligned} \tag{2.53}$$

Eliminating second-order terms and collapsing the skew-symmetric form yields the linearized angular error differential equation

$$\dot{\boldsymbol{\psi}} = -[(\mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e) \times] \boldsymbol{\psi} - \mathbf{C}_b^n \mathbf{b}^b - \mathbf{C}_b^n \mathbf{w}_b^b \tag{2.54}$$

2.6.3 Velocity Error Development. The development of the velocity error dynamics proceeds similarly to the attitude errors. The position and velocity errors ($\delta \mathbf{p}^n$ and $\delta \mathbf{v}^n$, respectively) are defined as

$$\delta \mathbf{p}^n = \tilde{\mathbf{p}}^n - \mathbf{p}^n \tag{2.55}$$

$$\delta \mathbf{v}^n = \tilde{\mathbf{v}}^n - \mathbf{v}^n \tag{2.56}$$

These definitions are substituted into the dynamics model to derive the dynamics of the position and velocity errors.

Recall the acceleration dynamics equation (2.27) derived in Section 2.5.5:

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \dot{\mathbf{p}}^n - \mathbf{C}_e^n (\boldsymbol{\Omega}_{ie}^e)^2 [\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n] + \mathbf{g}^n \tag{2.57}$$

The true acceleration with respect to the Earth-fixed navigation frame, $\dot{\mathbf{v}}^n$, consists of a specific force, Coriolis effects, centripetal acceleration, and a gravitation term. The centripetal acceleration and gravitation terms are typically combined into a single *gravity* vector, $\mathbf{g}^e(\mathbf{p}^e)$. The gravity vector is formally defined as the gradient of the gravity potential, $W(\mathbf{p}^e)$, shown in [7] as the sum of the gravitation and centripetal acceleration as a function of Cartesian position relative to the center of the Earth:

$$W(\mathbf{p}^e) = \frac{GM}{\|\mathbf{p}^e\|} + \frac{1}{2}\mathbf{p}^{eT}\boldsymbol{\Omega}_{ie}^{eT}\boldsymbol{\Omega}_{ie}^e\mathbf{p}^e + H.O.T. \quad (2.58)$$

where GM is the Earth's gravitational constant. Substituting the gravity function into Equation (2.57) yields the acceleration dynamics equation:

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \mathbf{v}^n + \mathbf{C}_e^n \mathbf{g}^e(\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n) \quad (2.59)$$

The calculated velocity vector differential equation is corrupted by both the accelerometer measurement errors as well as the attitude errors developed in Section 2.5.5:

$$\dot{\tilde{\mathbf{v}}}^n = \tilde{\mathbf{C}}_b^n \mathbf{f}_m^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \tilde{\mathbf{v}}^n + \mathbf{C}_e^n \mathbf{g}^e(\mathbf{p}_0^e + \mathbf{C}_n^e \tilde{\mathbf{p}}^n) \quad (2.60)$$

Substituting the attitude, velocity, position, and accelerometer measurement error equations into (2.60) yields

$$\begin{aligned} \dot{\tilde{\mathbf{v}}}^n = & [\mathbf{I} - (\boldsymbol{\psi} \times)] \mathbf{C}_b^n (\mathbf{f}^b + \mathbf{a}^b + \mathbf{w}_a^b) - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e (\mathbf{v}^n + \delta \mathbf{v}^n) \\ & + \mathbf{C}_e^n \mathbf{g}^e(\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n + \mathbf{C}_n^e \delta \mathbf{p}^n) \end{aligned} \quad (2.61)$$

The acceleration error vector, $\delta \dot{\mathbf{v}}^n$, is

$$\delta \dot{\mathbf{v}}^n = \dot{\tilde{\mathbf{v}}}^n - \dot{\mathbf{v}}^n \quad (2.62)$$

Substituting (2.57) and (2.61) into (2.62) and eliminating second-order terms yields

$$\delta \dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{a}^b - (\boldsymbol{\psi} \times) \mathbf{C}_b^n \mathbf{f}^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \delta \mathbf{v}^n + \mathbf{C}_e^n \mathbf{G} \mathbf{C}_n^e \delta \mathbf{p}^n + \mathbf{C}_b^n \mathbf{w}_a^b \quad (2.63)$$

where $\mathbf{G}$ is the gradient of the gravity vector, $\mathbf{g}^e(\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n)$. This gradient is calculated as [7]

$$\mathbf{G} = \nabla g|_{\mathbf{p}^n} \quad (2.64)$$

$$\mathbf{G} = \frac{GM}{\|\mathbf{p}^e\|^3} [3\check{\mathbf{p}}^e (\check{\mathbf{p}}^e)^T - \mathbf{I}] - (\boldsymbol{\Omega}_{ie}^e)^2 \quad (2.65)$$

where $\check{\mathbf{p}}^e$ is the ECEF position unit vector. The second term of (2.63) can be rearranged to facilitate state-space form as follows

$$(\boldsymbol{\psi} \times) \mathbf{C}_b^n \mathbf{f}^b = \boldsymbol{\psi} \times \mathbf{f}^n \quad (2.66)$$

$$= -(\mathbf{f}^n \times) \boldsymbol{\psi} \quad (2.67)$$

Substituting into (2.63) yields the linear stochastic velocity error model:

$$\delta \dot{\mathbf{v}}^n = \mathbf{C}_e^n \mathbf{G} \mathbf{C}_n^e \delta \mathbf{p}^n - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \delta \mathbf{v}^n + (\mathbf{f}^n \times) \boldsymbol{\psi} + \mathbf{C}_b^n \mathbf{a}^b + \mathbf{C}_b^n \mathbf{w}_a^b \quad (2.68)$$

Finally, the position error is simply the kinematic relationship between position and velocity:

$$\delta \dot{\mathbf{p}}^n = \delta \mathbf{v}^n \quad (2.69)$$

2.6.4 State-space Model. The error dynamics can now be expressed using the standard linear, stochastic, state-space model driven by white noise [37]:

$$\delta \dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \quad (2.70)$$

Note these error dynamics correspond to a specific accelerometer and gyroscope error model. In this example, the errors are modeled by a time-varying bias plus ran-

dom noise. Adding or reducing the complexity of the error model would change the resulting error dynamics.

The navigation error state vector, $\delta\mathbf{x}$, consists of position ($\delta\mathbf{p}^n$), velocity ($\delta\mathbf{v}^n$), attitude (ψ), accelerometer bias ($\delta\mathbf{a}^b$), and gyroscope bias errors ($\delta\mathbf{b}^b$) and is expressed as a vector of fifteen elements:

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p}^n \\ - - - \\ \delta\mathbf{v}^n \\ - - - \\ \psi \\ - - - \\ \delta\mathbf{a}^b \\ - - - \\ \delta\mathbf{b}^b \end{bmatrix}_{15 \times 1} \quad (2.71)$$

The driving noise vector, $\mathbf{w}$, consists of the noise terms associated with the accelerometer measurement ($\mathbf{w}_a^b$), gyroscope measurement ($\mathbf{w}_b^b$), accelerometer bias ($\mathbf{w}_{a_{bias}}^b$), and gyroscope bias ($\mathbf{w}_{b_{bias}}^b$), and is expressed as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_a^b \\ - - - \\ \mathbf{w}_b^b \\ - - - \\ \mathbf{w}_{a_{bias}}^b \\ - - - \\ \mathbf{w}_{b_{bias}}^b \end{bmatrix}_{12 \times 1} \quad (2.72)$$

The overall error dynamics in augmented state-space form are

$$\begin{aligned}
\delta \dot{\mathbf{x}} = & \begin{bmatrix} \mathbf{0}_3 & | & \mathbf{I}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 \\ \hline & & & & & & & & \\ \mathbf{C}_e^n \mathbf{G} \mathbf{C}_n^e & | & -2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e & | & (\mathbf{f}^n \times) & | & \mathbf{C}_b^n & | & \mathbf{0}_3 \\ \hline & & & & & & & & \\ \mathbf{0}_3 & | & \mathbf{0}_3 & | & -(\mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e) \times & | & \mathbf{0}_3 & | & -\mathbf{C}_b^n \\ \hline & & & & & & & & \\ \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & -\frac{1}{\tau_a} \mathbf{I}_3 & | & \mathbf{0}_3 \\ \hline & & & & & & & & \\ \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & -\frac{1}{\tau_b} \mathbf{I}_3 \end{bmatrix}_{15 \times 15} \delta \mathbf{x} \\
& + \begin{bmatrix} \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 \\ \hline - & & - & & - & & - \\ \mathbf{C}_b^n & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 \\ \hline - & & - & & - & & - \\ \mathbf{0}_3 & | & -\mathbf{C}_b^n & | & \mathbf{0}_3 & | & \mathbf{0}_3 \\ \hline - & & - & & - & & - \\ \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{I}_3 & | & \mathbf{0}_3 \\ \hline - & & - & & - & & - \\ \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{0}_3 & | & \mathbf{I}_3 \end{bmatrix}_{15 \times 12} \mathbf{w} \quad (2.73)
\end{aligned}$$

Note that the error dynamics are a time-varying function of the trajectory flown.

2.7 Digital Imaging

In this section, concepts relative to digital imaging and object tracking are presented, including a discussion of the optical sensor model, projection theory, and fixed-object tracking.

2.7.1 Optical Sensor Model. An optical sensor is a device designed to measure the intensity of optical energy (light) entering the sensor through an aperture.

Imaging sensors consist of an array of light-sensitive detectors which create a multidimensional light intensity measurement. In this section, the basic physical properties of an optical sensor are presented, and a model representing an optical sensor is given.

For the purposes of this discussion, the *world* is defined as a collection of all real objects. Some objects are sources of radiometric illumination or *radiance*. These light sources illuminate the world and interact with the other physical objects through various types of reflection. The amount of light propagating along a certain direction is defined as the *irradiance* [36]. The physical irradiance pattern entering the aperture of the optical sensor is defined as the *scene* and is represented by a time-varying array of nonnegative rational numbers, $\mathbf{O}(x, y, t)$, corresponding to the photon arrival rate.

A digital optical imaging sensor consists of an aperture, lens, detector array, and sampling array. A simple imaging system model is shown in Figure 2.6. The lens focuses the scene on the detector array. The light pattern focused on the detector array is defined as the *image* and represented by, $\mathbf{I}(x, y, t)$. In statistical terms, the image is the mean photon arrival rate. The detector array converts the light energy into a voltage or a charge which is converted to a digital value by the sampling array. The sampling array is assumed to be a square grid, although other patterns can be designed (e.g., honeycomb) [24].

The lens is a natural low-pass filter in the spatial domain, with a cutoff frequency (f_c) defined by the aperture diameter (D), wavelength of light source (λ), and focal length (f) [15]:

$$f_c = \frac{D}{\lambda f} \quad (2.74)$$

Thus, a point source of light would appear slightly blurred on the image plane. Assuming spatial invariance, this blurring due to the lens is represented by the *point spread function*, $\mathbf{H}(x, y)$, where x and y are the spatial differences in the x and y directions, respectively. The image can now be expressed mathematically as the convolution of

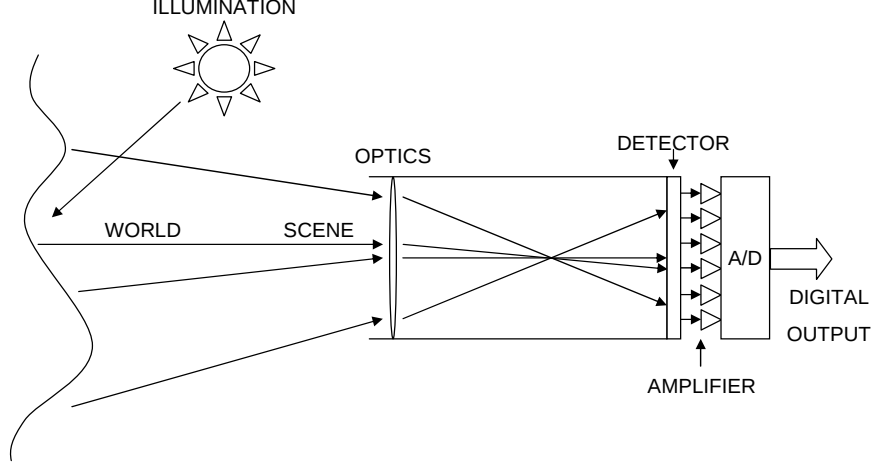


Figure 2.6: Imaging system model. The imaging system transforms the scene into a digital image. The major components of the camera are the optics, light detector, amplifier, and analog to digital converter.

the scene and point spread function [20]:

$$\mathbf{I}(x, y, t) = \int_{\xi \in \mathbf{X}} \int_{\rho \in \mathbf{Y}} \mathbf{O}(\xi, \rho, t) \mathbf{H}(x - \xi, y - \rho) d\rho d\xi \quad (2.75)$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the dimensions of the image array in the x and y directions, respectively.

The light energy in the image is integrated over a period defined as the *dwell time* (Δt). The integration output is defined as

$$\mathbf{I}_d(x, y, t_i) = \int_{t_i - \Delta t}^{t_i} \mathbf{I}(x, y, t) dt \quad (2.76)$$

At the end of an integration period, the accumulated charge of the sensor is proportional to the number of photons received, which is referred to as the *quantum efficiency* (p). The resulting accumulated charge is represented by

$$\mathbf{E}(x, y, t_i) = p \mathbf{I}_d(x, y, t_i) \quad (2.77)$$

Due to the quantum nature of light, this accumulated charge is corrupted by a random photon counting noise, sometimes referred to as *shot noise*. The accumulated charge is converted to a voltage by an amplifier with gain g , bias b , and additive readout noise parameter $\mathbf{w}_a$:

$$\mathbf{D}(x, y, t_i) = g \mathbf{E}(x, y, t_i) + b + \mathbf{w}_a \quad (2.78)$$

This voltage is then converted to a digital number in the quantizer:

$$\mathbf{D}_s(x, y, t_i) = q [\mathbf{D}(x, y, t_i)] \quad (2.79)$$

which can be stored in a computer's memory.

As mentioned above, the image model includes three sources of noise: photon counting noise, readout noise, and quantization noise. Photon counting noise is modeled as a Poisson process [48]. The probability distribution function is

$$P [\mathbf{E}(x, y, t_i) = k] = e^{-\mathbf{I}_d(x, y, t_i)} \frac{\mathbf{I}_d(x, y, t_i)^k}{k!}, \quad k \in \mathbb{Z}^* \quad (2.80)$$

Readout noise includes the effects of amplifier thermal noise on the readout voltages and is modeled as independent, identically distributed zero-mean Gaussian distributions with covariance matrix $\mathbf{P}_{aa}$. Quantization noise results from the errors induced by converting a nonnegative real voltage from the amplifier to a digital number with fixed precision. Assuming a linear quantizer with a precision of n bits and range of V volts, the quantization noise can be modeled as a zero-mean uniform distribution [48] with variance

$$\sigma_q^2 = \frac{1}{12} \left(\frac{V^2}{2^{2n}} \right) \quad (2.81)$$

In the next section, a mathematical model describing the relationship between point locations in the world and image is derived.

2.7.2 Projection Theory. The camera optical properties define the relationship between the scene and the projected image. Recalling the simple camera

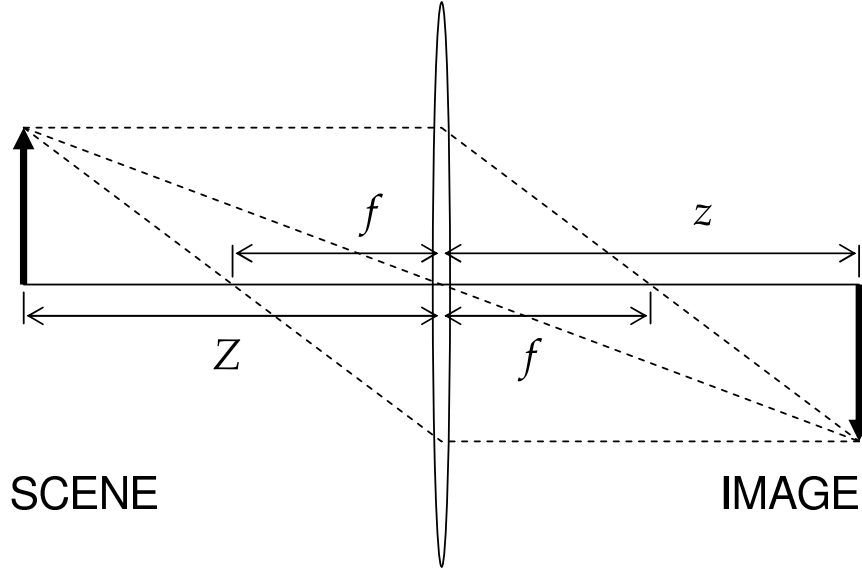


Figure 2.7: Thin lens camera model. The thin lens model directs parallel light rays toward the focus, resulting in an image. Figure is not to scale.

model (Figure 2.6), the lens focuses the incoming irradiance pattern (i.e., scene) onto the image plane. For a theoretical thin lens, the projection is a function of the focal length of the lens and the distance from the lens, as shown in Figure 2.7. This relationship is expressed by the *fundamental equation of the thin lens* [36]:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \quad (2.82)$$

where Z is the distance from the object to the lens, z is the distance from the lens to the image plane, and f is the focal length.

As the aperture of the thin lens decreases to zero, the system can be modeled as a pinhole camera (see Figure 2.8). In this model, all incoming light must pass through the optical center and is projected on an image plane located at a distance f from the lens. The resulting image is an inverted projection of the scene.

This model can be further simplified by placing a virtual image plane in front of the optical center, as shown in Figure 2.9. Given a point source at location $\mathbf{s}^c$ the resulting location of the point source on the image plane, relative to the optical center

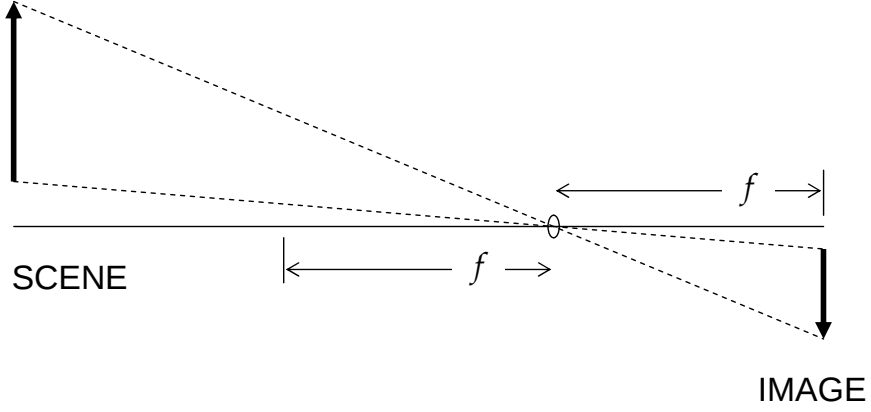


Figure 2.8: Pinhole camera model. The pinhole camera is a theoretical camera model where a thin lens aperture approaches zero. The projected image is inverted on the image plane.

of the camera, is given by

$$\mathbf{s}^{proj} = \left(\frac{f}{s_z^c} \right) \mathbf{s}^c \quad (2.83)$$

where s_z^c is the distance of the point source from the optical center of the camera in the z_c direction.

In order to interpret the calculated projection in a digital image, the physical image plane coordinates must be converted to a coordinate system based on pixel location. The following development defines the pixel coordinate system and derives the transformation from the physical image plane to pixel location. The image plane consists of an $(M \times N)$ grid of rectangular pixels with height H and width W , shown in Figure 2.10. The origin of the projection frame is located at the physical center of the array. The origin of the pixel coordinate system is located beyond the upper left corner of the array, such that the center of the upper left pixel corresponds to the (1,1) pixel coordinate. This definition of pixel coordinates corresponds to the elemental matrix locations when the image is stored in a computer.

The transformation from the projection coordinates to pixel coordinates is given by

$$\mathbf{s}^{pix} = \begin{bmatrix} -\frac{M}{H} & 0 & 0 \\ 0 & \frac{N}{W} & 0 \end{bmatrix} \mathbf{s}^{proj} + \begin{bmatrix} \frac{M+1}{2} \\ \frac{N+1}{2} \end{bmatrix} \quad (2.84)$$

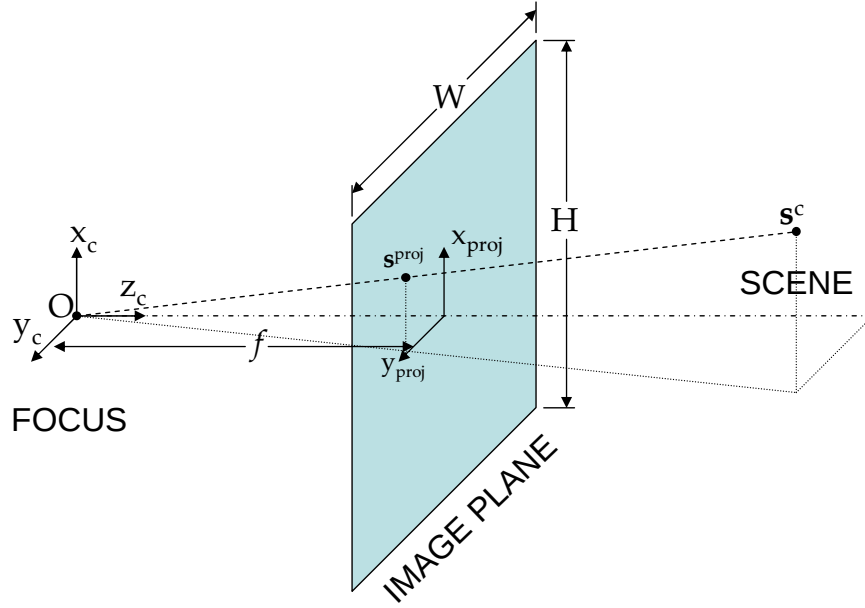


Figure 2.9: Camera projection model. The pinhole camera model is modified by placing a virtual image plane one focal length in front of the optical center. As a result, this model eliminates the image inversion present in the standard pinhole camera model.

Combining Equations (2.83) and (2.84) yields the following transformation from camera frame to homogeneous pixel coordinates

$$\underline{\mathbf{s}}^{pix} = \frac{1}{s_z^c} \begin{bmatrix} -f \frac{M}{H} & 0 & \frac{M+1}{2} \\ 0 & f \frac{N}{W} & \frac{N+1}{2} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{s}^c \quad (2.85)$$

$$= \frac{1}{s_z^c} \mathbf{T}_c^{pix} \mathbf{s}^c \quad (2.86)$$

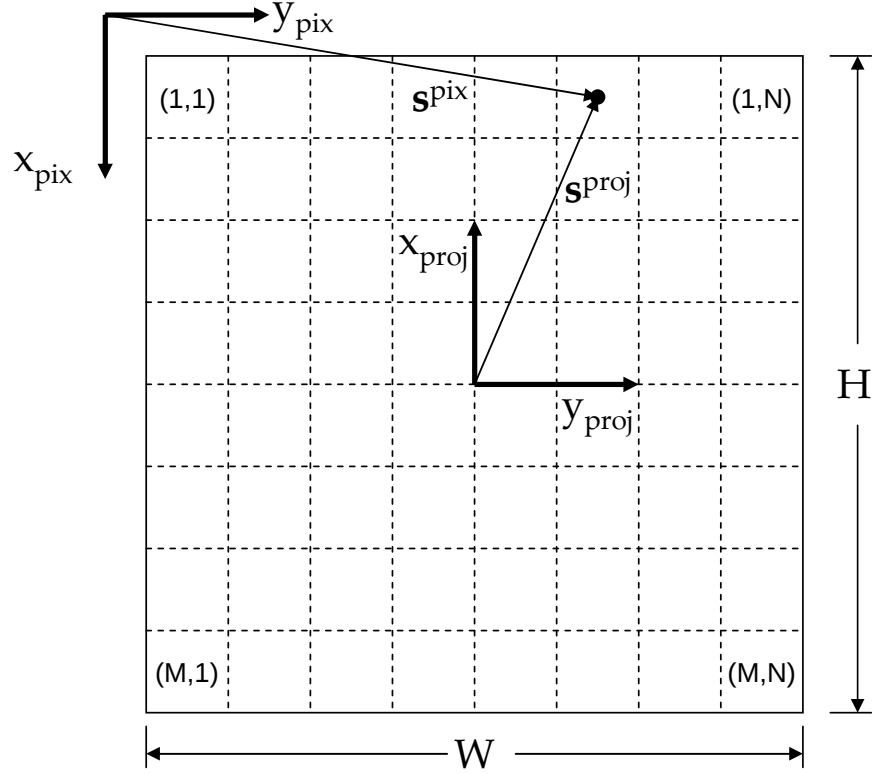


Figure 2.10: Camera image array. The camera imager consists of an $(M \times N)$ array of pixels. The physical height and width of the array are represented by H and W , respectively.

where $\mathbf{T}_c^{pix}$ is the homogeneous transformation matrix from camera frame to pixel frame. The inverse transformation is calculated as

$$\mathbf{T}_{pix}^c = (\mathbf{T}_c^{pix})^{-1} \quad (2.87)$$

$$= \begin{bmatrix} -\frac{H}{fM} & 0 & \frac{H(M+1)}{2fM} \\ 0 & \frac{W}{fN} & -\frac{W(N+1)}{2fN} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.88)$$

A transformation from a target location in Earth-fixed navigation frame coordinates to pixel coordinates can now be derived based on the navigation state. The geometry is shown in Figure 2.11. The line of sight vector, $\mathbf{s}$, is the vector difference between the target location $\mathbf{t}$ and the camera position, which are both available in Earth-fixed navigation frame coordinates:

$$\mathbf{s}^n = \mathbf{t}^n - \mathbf{p}^n \quad (2.89)$$

The resultant vector can be transformed to the camera reference frame using the Earth-fixed navigation-to-body and body-to-camera frame direction cosine matrices:

$$\mathbf{s}^c = \mathbf{C}_b^c \mathbf{C}_n^b \mathbf{s}^n \quad (2.90)$$

Finally, the pixel location is calculated using Eqn. (2.86).

2.7.3 Nonlinear Optical Distortion. As discussed in [36], the optical properties of a lens can be significantly different than the pinhole model. This distortion is primarily radial in nature and can be identified when straight lines appear curved in the image. An optical distortion model is desired which transforms locations in a distorted image to an equivalent pixel location which agrees with the pinhole camera model.

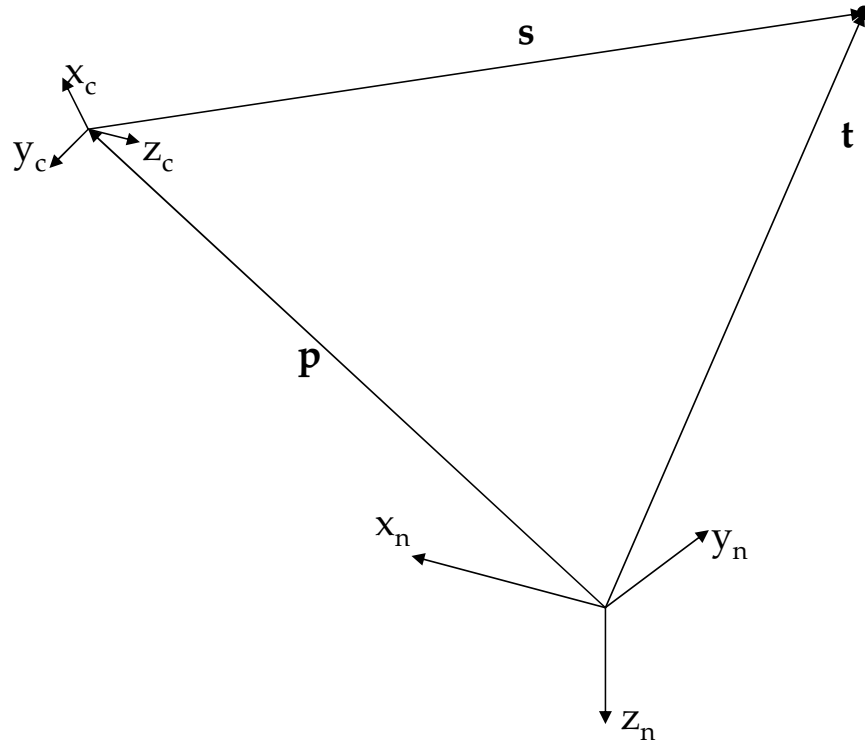


Figure 2.11: Target to image transformation geometry. The relationship between the camera position, ($\mathbf{p}$), and target location, ($\mathbf{t}$), can be expressed in pixel coordinates using transformations based on the navigation state and camera parameters.

In [36] and [11], the following radial distortion correction is proposed:

$$\mathbf{z}^u = \mathcal{P}_a(r)(\mathbf{z}^d - \mathbf{c}_0) + \mathbf{c}_0 \quad (2.91)$$

where $\mathbf{z}^d$ are the distorted pixel locations, $\mathbf{z}^u$ are the undistorted pixel locations, $\mathbf{c}_0$ is the center of optical distortion (in pixels), and $\mathcal{P}_a(r)$ is a polynomial expansion in r . This is defined as

$$\mathcal{P}_a(r) = 1 + \sum_{i=1}^2 a_{2i} r^{2i} \quad (2.92)$$

where r is the radial distance from the center of optical distortion

$$r = \|\mathbf{z}^d - \mathbf{c}_0\| \quad (2.93)$$

and a_n are the coefficients of the polynomial.

Concepts related to tracking objects in the scene are presented in the next section.

2.7.4 Fixed Object Tracking. As shown in the previous section, an image is a transformation based on the illumination of objects in the field of view. It is well known that a sequence of images of a particular scene can provide information regarding the camera location, camera orientation, and the location of objects in the field of view [36,47,62,63]. This is accomplished by identifying and matching common objects in multiple images, which is known as feature tracking.

Feature tracking methods typically begin by analyzing an image to determine the location of *features*—areas which are identifiable by some measure and which have the highest potential for locating the matching feature in subsequent images. The classic feature detection algorithms are the Lucas-Kanade feature tracker [35] and the similar Harris corner detector [18]. These algorithms attempt to find a maximum amount of detail information by calculating the eigenvalues of the following matrix

over a window, $\mathbf{W}$:

$$\mathbf{G} = \begin{bmatrix} \sum_{x,y \in \mathbf{W}} \mathbf{I}_x^2(x, y) & \sum_{x,y \in \mathbf{W}} \mathbf{I}_x(x, y) \mathbf{I}_y(x, y) \\ \sum_{x,y \in \mathbf{W}} \mathbf{I}_x(x, y) \mathbf{I}_y(x, y) & \sum_{x,y \in \mathbf{W}} \mathbf{I}_y^2(x, y) \end{bmatrix} \quad (2.94)$$

where $\mathbf{I}_x$ and $\mathbf{I}_y$ are the gradients of an image $\mathbf{I}$ in the x and y directions, respectively. A zero eigenvalue corresponds to zero detail in the direction of the eigenvector, which results from a flat image. The eigenvalues increase with increasing detail, indicating a potentially distinct feature. Features are declared by thresholding the eigenvalues and choosing the strongest candidates. Harris refines this method by defining the following metric:

$$C(\mathbf{G}) = \det(\mathbf{G}) + k [\text{trace}^2(\mathbf{G})] \quad (2.95)$$

which is equivalent to

$$C(\mathbf{G}) = (1 + 2k)\sigma_1\sigma_2 + k(\sigma_1^2 + \sigma_2^2) \quad (2.96)$$

where σ_1 and σ_2 are the eigenvalues of $\mathbf{G}$, and k is a tuning parameter. Smaller values of k favor detail in both directions, while larger values of k are more accepting of detail in only one direction [18].

More robust feature selection algorithms have been proposed. An example is the scale-invariant feature tracker (SIFT) method developed by Lowe [33]. These SIFT features are invariant over scale and rotational changes, thus providing a more robust solution for arbitrary motion. The SIFT algorithm is described in further detail in Chapter IV. In addition, Bhanu proposes the use of genetic algorithms for feature detection [3]. In any case, identifying robust features is a rich problem and remains an area of active research.

After a feature corresponding to a physical object has been identified in one frame, the same feature must be located in subsequent frames. The most common approach defines an error cost function designed to represent the “difference” between

the features. A minimization then determines the pixel location which has the lowest error cost, resulting in the best-match. For example, the classic Lucas-Kanade tracker seeks to minimize the sum of the squared differences of the image intensity over a specified region [35]. Given a sequence of images $\{\mathbf{I}(t_1), \mathbf{I}(t_1), \dots, \mathbf{I}(t_n)\}$, assume a feature is defined as a region $\mathbf{W}_1$ on $\mathbf{I}(t_1)$ as

$$\mathbf{F}_1 = \mathbf{I}(x, y, t_1), \quad x, y \in \mathbf{W}_1 \quad (2.97)$$

The sum of squared differences for the feature $\mathbf{F}_1$ and image $\mathbf{I}(t_n)$ is

$$\mathbf{C}_{1n}(\alpha, \beta) = \sum_{x \in \mathbf{W}_1} \sum_{y \in \mathbf{W}_1} [\mathbf{I}(x - \alpha, y - \beta, t_n) - \mathbf{I}(x, y, t_1)]^2 \quad (2.98)$$

The cost function, $\mathbf{C}$, is minimized over the shift vector, (α, β) , to determine the best feature match.

In this section, a simple camera and image model is presented which incorporates measurement errors. The model is then used to define feature selection and correspondence methods for the most common feature tracking methods.

2.8 *Kalman Filtering*

In this section, Kalman filtering concepts are presented, including a development of the linear Kalman filter equations, the extension of the linear equations to nonlinear problems, and the benefits of tightly coupling sensors using a feedback Kalman filter.

2.8.1 Linear Kalman Filter. The Kalman filter is a recursive, optimal data processing algorithm [37]. The goal of the algorithm is to estimate the solution to a linear stochastic differential equation. In this development, the general form of the stochastic differential equation is explained, and the Kalman filter propagation and update equations are presented.

The general form of the linear stochastic differential equation is

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \quad (2.99)$$

where $\mathbf{x}$ is the state vector, $\mathbf{F}$ is the homogeneous system dynamics matrix, $\mathbf{B}$ is the input matrix, $\mathbf{u}$ is the input vector, $\mathbf{G}$ is the noise transformation matrix, and $\mathbf{w}$ is a vector of white noise processes. The white noise vector is a zero mean Gaussian process with covariance

$$\mathbb{E}\{\mathbf{w}(t)\mathbf{w}^T(t + \tau)\} = \mathbf{Q}(t)\delta(\tau) \quad (2.100)$$

where $\delta(t)$ is the Dirac delta function.

Because the state vector is a vector of random elements, the solution to the stochastic differential equation is a time-varying probability density function. The Kalman filter derivation assumes Gaussian distributions for all random variables, thus a complete characterization of the probability density function requires calculating only the mean and covariance of the state vector [66].

The mean of the state vector is defined as

$$\mathbf{m}_x = \mathbb{E}\{\mathbf{x}(t)\} = \mathbf{\Phi}(t, t_0)\mathbb{E}\{\mathbf{x}(t_0)\} + \int_{t_0}^t \mathbf{\Phi}(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau \quad (2.101)$$

where $\mathbf{\Phi}(t, t_0)$ is the state transition matrix from time t_0 to time t , and $\mathbb{E}\{\cdot\}$ is the expectation operator [37]. Note that this is equivalent to the solution of the deterministic linear differential equation.

The covariance of the state vector is defined as the difference between the mean square value and the mean:

$$\mathbf{P}_{xx}(t) = \mathbb{E}\{\mathbf{x}(t)\mathbf{x}^T(t)\} - \mathbf{m}_x(t)\mathbf{m}_x^T(t) \quad (2.102)$$

The state covariance matrix expressed as a function of the system dynamic parameters is

$$\mathbf{P}_{xx}(t) = \mathbf{\Phi}(t, t_0)\mathbf{P}_{xx}(t_0)\mathbf{\Phi}^T(t, t_0) + \int_{t_0}^t \mathbf{\Phi}(t, \tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^T(\tau)\mathbf{\Phi}^T(t, \tau)d\tau \quad (2.103)$$

Because the state vector is defined as a Gaussian process, the maximum *a posteriori* (MAP) estimate of the state is simply the mean [66]. The MAP state estimate is denoted with the *hat* notation:

$$\hat{\mathbf{x}}(t) = \mathbf{m}_x(t) \quad (2.104)$$

The covariance is an estimate of the uncertainty in the state estimate and is defined using the *hat* notation:

$$\hat{\mathbf{P}}(t) = \mathbf{P}_{xx}(t) \quad (2.105)$$

In addition to the state dynamics, the Kalman filter normally incorporates discrete measurements. The measurements are random vectors modeled as linear functions of the state vector with additive noise:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \quad (2.106)$$

where $\mathbf{H}(t_i)$ is the observation matrix and $\mathbf{v}(t_i)$ is a zero-mean white Gaussian noise vector with covariance defined by

$$\mathbf{E}\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\} = \mathbf{R}(t_i)\delta_{ij} \quad (2.107)$$

where δ_{ij} is the Kroeneker delta function.

Incorporating the system dynamics and update models into an algorithm results in two distinct operations: measurement updates and propagation. Assume measurements are available at discrete times, $(t_i, t_{i+1}, \dots, t_{i+n})$, and estimates of the mean

and covariance are available immediately prior to time t_i (denoted by t_i^- .) The mean and covariance after incorporating the update (denoted by t_i^+), is defined as [37]

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i) [\mathbf{z}(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)] \quad (2.108)$$

$$\hat{\mathbf{P}}(t_i^+) = \hat{\mathbf{P}}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\hat{\mathbf{P}}(t_i^-) \quad (2.109)$$

where $\mathbf{K}(t_i)$ is the Kalman gain matrix, defined by

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T(t_i) [\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)]^{-1} \quad (2.110)$$

After the update at time t_i has been completed, the state estimate and covariance are propagated to the next update time, t_{i+1} , using the solution to the stochastic differential equation previously derived:

$$\hat{\mathbf{x}}(t_{i+1}^-) = \Phi(t_{i+1}, t_i)\hat{\mathbf{x}}(t_i^+) + \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau \quad (2.111)$$

$$\begin{aligned} \hat{\mathbf{P}}(t_{i+1}^-) &= \Phi(t_{i+1}, t_i)\hat{\mathbf{P}}(t_i^+)\Phi^T(t_{i+1}, t_i) \\ &\quad + \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^T(\tau)\Phi(t_{i+1}, \tau)^T d\tau \end{aligned} \quad (2.112)$$

The combination of the update and propagation equations define a recursive algorithm which maintains an estimate of the state and the state uncertainty based on a system dynamics model and incorporating measurements.

2.8.2 Extended Kalman Filter. The linear Kalman filter developed in the previous section is based on a linear system model driven by white Gaussian noise. While this is an appropriate description of many real-world systems, it is not adequate for some problems [38]. These nonlinear problems motivate the development of the extended Kalman filter (EKF). The development begins with a definition of a nonlinear system model.

The proposed nonlinear system model is defined as the following stochastic differential equation driven by white Gaussian noise:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}\mathbf{w}(t) \quad (2.113)$$

Noise-corrupted, discrete measurements are available in the following form:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \quad (2.114)$$

Perturbation techniques are used to approximate the system about a nominal trajectory. This results in linear stochastic equations which can be solved using the traditional linear Kalman filter propagation and update equations [38].

The derivation proceeds by defining a nominal trajectory and estimating the *errors* about the nominal trajectory. Thus, the perturbation model for the state vector is defined as the sum of a nominal trajectory and an error state:

$$\mathbf{x}(t) = \bar{\mathbf{x}}(t) + \delta\mathbf{x}(t) \quad (2.115)$$

The nominal trajectory is defined as

$$\dot{\bar{\mathbf{x}}}(t) = \mathbf{f}[\bar{\mathbf{x}}(t), \mathbf{u}(t), t] \quad (2.116)$$

Applying the error modeling procedure presented in Section 2.4 yields the following approximation to the nonlinear stochastic dynamics model:

$$\dot{\bar{\mathbf{x}}}(t) + \delta\dot{\mathbf{x}}(t) = \mathbf{f}[\bar{\mathbf{x}}(t), \mathbf{u}(t), t] + \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \quad (2.117)$$

which is the sum of the deterministic nominal trajectory (2.116) and a stochastic linear differential equation

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \quad (2.118)$$

The state estimate is defined as the sum of the nominal trajectory and the estimated perturbation:

$$\hat{\mathbf{x}}(t) = \bar{\mathbf{x}}(t) + \delta\hat{\mathbf{x}}(t) \quad (2.119)$$

The state error covariance is denoted by $\mathbf{P}(t)$.

To propagate from time t_i^+ to t_{i+1}^- , the nominal trajectory is calculated using a nonlinear differential equation solver [50], with the state estimate at t_i^+ as the initial condition. The covariance is propagated using the time propagation equations presented in the previous section.

To incorporate the measurement update at t_{i+1} , the nonlinear update Equation (2.114) is linearized about the nominal trajectory, $\hat{\mathbf{x}}(t_{i+1}^-)$:

$$\mathbf{z}(t_{i+1}) = \mathbf{h}[\bar{\mathbf{x}}(t_{i+1}^-), t_{i+1}] + \mathbf{H}(t_{i+1})\delta\mathbf{x}(t_{i+1}) + \mathbf{v}(t_{i+1}) \quad (2.120)$$

where $\mathbf{H}(t)$ is

$$\mathbf{H}(t) = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}(t), t} = \left[\begin{array}{ccc} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_n} \end{array} \right] \bigg|_{\bar{\mathbf{x}}(t), t} \quad (2.121)$$

Defining the measurement as the sum of a nominal and an error component

$$\mathbf{z}(t_{i+1}) = \bar{\mathbf{z}}(t_{i+1}) + \delta\mathbf{z}(t_{i+1}) \quad (2.122)$$

and subtracting the nominal component from (2.120) yields the perturbation update equation:

$$\delta\mathbf{z}(t_{i+1}) = \mathbf{H}(t_{i+1})\delta\mathbf{x}(t_{i+1}) + \mathbf{v}(t_{i+1}) \quad (2.123)$$

The post-measurement error state vector, $\delta\hat{\mathbf{x}}(t_{i+1}^+)$, and covariance, $\mathbf{P}(t_{i+1}^+)$, are calculated using the Kalman update equations shown in the previous section.

Finally, the current error state estimate is incorporated into the nominal trajectory:

$$\bar{\mathbf{x}}(t_{i+1}^+) = \bar{\mathbf{x}}(t_{i+1}^-) + \delta\hat{\mathbf{x}}(t_{i+1}^+) \quad (2.124)$$

the error state estimate is reset to zero and the process is repeated.

2.9 Optimal Batch Estimation

In this section, the method of optimal estimation using batch measurements is presented, focusing on the problem of estimating parameters given nonlinear mathematical models.

2.9.1 Linear Mathematical Models. The concept of optimal batch estimation is introduced for the case where the observation function is well-modeled using a set of linear stochastic functions. The linear problem is defined using the following mathematical model [72]:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (2.125)$$

where $\mathbf{H}$ is the design (or influence) matrix, $\mathbf{x}$ is the solution vector, $\mathbf{z}$ is the observation vector, and $\mathbf{v}$ is the residual vector.

The goal of the regression is to estimate the values of $\hat{\mathbf{x}}$ which, given $\mathbf{H}$ and $\mathbf{z}$, minimize the sum-of-squares of the estimated residual vector, weighted by the uncertainty of the measurement error:

$$\min (\hat{\mathbf{v}}\mathbf{R}^{-1}\hat{\mathbf{v}}^T) \quad (2.126)$$

where $\mathbf{R}$ is the measurement error covariance kernel

$$\mathbf{R} = \mathbf{E} [\hat{\mathbf{v}}\hat{\mathbf{v}}^T] \quad (2.127)$$

where $\mathbf{E}[\cdot]$ is the expected value operator. The resulting expression which minimizes the sum-of-squares of the residual vector is shown to be [72]

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{z} \quad (2.128)$$

where the matrix $(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})$ is defined as the matrix of the normal equations. A unique solution exists if and only if the matrix of normal equations is nonsingular.

In the next section, the optimal batch estimation concept is expanded to treat nonlinear mathematical models.

2.9.2 Nonlinear Mathematical Models. In many cases, linear models are inadequate for system modeling, and a nonlinear system model is required. This is the situation for many systems, including image and inertial navigation solutions. A general nonlinear model which is useful for working with imaging systems is defined in [72] as

$$\mathbf{h}(\mathbf{x}, \mathbf{z}) = 0 \quad (2.129)$$

where $\mathbf{h}$ is a nonlinear system function, $\mathbf{x}$ is a vector of system parameters, and $\mathbf{z}$ is a vector of (perfect) observations.

In order to estimate the unknown system parameter vectors, perturbation techniques are employed. The model is linearized about nominal system parameter vectors which are defined as

$$\mathbf{x} = \bar{\mathbf{x}} + \delta \mathbf{x} \quad (2.130)$$

$$\mathbf{z} = \bar{\mathbf{z}} + \delta \mathbf{z} \quad (2.131)$$

where $\bar{\mathbf{x}}$ represents an initial approximation of the system parameters, and $\bar{\mathbf{z}}$ is the vector of (noise-corrupted) observations. Substituting equations (2.130) and (2.131) into (2.129) yields

$$\mathbf{h}(\bar{\mathbf{x}} + \delta \mathbf{x}, \bar{\mathbf{z}} + \delta \mathbf{z}) = 0 \quad (2.132)$$

Using perturbation techniques (see Section 2.4) Equation (2.132) is linearized about the nominal state, resulting in the following first-order approximation (i.e., assuming $\mathbf{h}$ is continuously differentiable in $\mathbf{x}$ and $\mathbf{z}$):

$$\mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{z}}) + \mathbf{H}_x \delta \mathbf{x} + \mathbf{H}_z \delta \mathbf{z} \approx 0 \quad (2.133)$$

where

$$\mathbf{H}_x = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}, \bar{\mathbf{z}}} \quad (2.134)$$

$$\mathbf{H}_z = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \right|_{\bar{\mathbf{x}}, \bar{\mathbf{z}}} \quad (2.135)$$

Given a vector of noise-corrupted observations and approximated system parameters, the misclosure vector ($\mathbf{w}$) is defined as

$$\mathbf{w} = \mathbf{h}(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \quad (2.136)$$

Substituting Equation (2.136) into (2.133) yields the linearized form of the regression:

$$\mathbf{w} + \mathbf{H}_x \delta \mathbf{x} + \mathbf{H}_z \delta \mathbf{z} \approx 0 \quad (2.137)$$

Assuming there is no prior information regarding the system parameters, the estimated parameter error vector which minimizes the weighted sum-of-squared errors is given by [72]

$$\hat{\delta \mathbf{x}} = - \left[\mathbf{H}_x^T (\mathbf{H}_z \mathbf{R} \mathbf{H}_z^T)^{-1} \mathbf{H}_x \right]^{-1} \mathbf{H}_x^T (\mathbf{H}_z \mathbf{R} \mathbf{H}_z^T)^{-1} \mathbf{w} \quad (2.138)$$

where $\mathbf{R}$ is the measurement error covariance kernel, defined in the previous section (2.9.1). The estimated measurement residual errors are given by

$$\hat{\delta \mathbf{z}} = -\mathbf{R} \mathbf{H}_z^T (\mathbf{H}_z \mathbf{R} \mathbf{H}_z^T)^{-1} (\mathbf{H}_x \hat{\delta \mathbf{x}} + \mathbf{w}) \quad (2.139)$$

Recalling equations (2.130) and (2.131), the estimated system parameters and measurement residual errors are then added to their respective nominal parameter vectors to produce the whole-valued state estimates:

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \delta\hat{\mathbf{x}} \quad (2.140)$$

$$\hat{\mathbf{z}} = \bar{\mathbf{z}} + \delta\hat{\mathbf{z}} \quad (2.141)$$

In case where there is prior knowledge regarding the values of the system parameters, and this uncertainty is represented by the covariance kernel $\mathbf{P}_{xx}$, defined as

$$\mathbf{P}_{xx} = \mathbf{E} [\delta\mathbf{x}\delta\mathbf{x}^T] \quad (2.142)$$

the regression equation (2.138) is modified such that the prior knowledge is properly weighted

$$\delta\hat{\mathbf{x}} = - \left[\mathbf{P}_{xx}^{-1} + \mathbf{H}_x^T (\mathbf{H}_z \mathbf{R} \mathbf{H}_z^T)^{-1} \mathbf{H}_x \right]^{-1} \mathbf{H}_x^T (\mathbf{H}_z \mathbf{R} \mathbf{H}_z^T)^{-1} \mathbf{w} \quad (2.143)$$

For cases where the solution ($\hat{\mathbf{x}}$) converges slowly, the process can be repeated using the whole-valued state estimate to calculate the partial derivatives. This iterative technique can improve the accuracy of future estimates, however local convergence is not guaranteed unless Lyapunov stability can be shown [45]. For a complete discussion of least-squares techniques, see [72], [30], and [67].

2.10 Direction Cosine Perturbation Methods

In order to use perturbation methods, the calculation of partial derivatives is required. While partial derivative calculations for vectors is straightforward, this is not necessarily the case for direction cosine matrices. This issue is addressed by representing small angles using a three dimensional vector. From [64], a direction

cosine matrix can be approximated by

$$\mathbf{C}_b^a = [\mathbf{I} + \boldsymbol{\theta} \times] \quad (2.144)$$

where $\boldsymbol{\theta}$ is a three-element vector of small angles.

Noting the multiplicative nature of direction cosine matrices allows the expression of any direction cosine matrix as the product of a direction cosine matrix and a small rotation. For example, the arbitrary direction cosine matrix, $\mathbf{C}_c^a$, can be expressed as

$$\mathbf{C}_c^a = [\mathbf{I} + \boldsymbol{\theta} \times] \mathbf{C}_c^b \quad (2.145)$$

This property is ideal for expressing small angular errors of the navigation system based on an estimated direction cosine matrix of interest which simplifies the calculation of partial derivatives and reduces the number of states in the filter. This concept is illustrated using an example.

A line of sight vector of interest, $\mathbf{s}^c$, is given by

$$\mathbf{s}^c = \mathbf{C}_b^c \mathbf{C}_n^b [\mathbf{y}^n - \mathbf{p}^n] \quad (2.146)$$

The true body to navigation frame DCM can be expressed equivalently as the product of a calculated body to navigation frame DCM and a small angular change due to errors:

$$\mathbf{C}_b^n = [\mathbf{I} - \boldsymbol{\psi} \times] \mathbf{C}_{\tilde{b}}^{\tilde{n}} \quad (2.147)$$

Substituting Eqn. (2.147) into (2.146) yields

$$\mathbf{s}^c = \mathbf{C}_b^c \mathbf{C}_{\tilde{n}}^b [\mathbf{I} + \boldsymbol{\psi} \times] [\mathbf{y}^n - \mathbf{p}^n] \quad (2.148)$$

Now calculation of the partial derivative of the attitude error vector, $\boldsymbol{\psi}$, is straightforward:

$$\frac{\partial \mathbf{s}^c}{\partial \boldsymbol{\psi}} = \frac{\partial}{\partial \boldsymbol{\psi}} (\mathbf{C}_b^c \mathbf{C}_{\tilde{n}}^b (\boldsymbol{\psi} \times) [\mathbf{y}^n - \mathbf{p}^n]) \quad (2.149)$$

Rearranging the skew-symmetric form yields

$$\frac{\partial \mathbf{s}^c}{\partial \boldsymbol{\psi}} = \frac{\partial}{\partial \boldsymbol{\psi}} (-\mathbf{C}_b^c \mathbf{C}_{\tilde{n}}^b ([\mathbf{y}^n - \mathbf{p}^n] \times) \boldsymbol{\psi}) \quad (2.150)$$

$$\frac{\partial \mathbf{s}^c}{\partial \boldsymbol{\psi}} = -\mathbf{C}_b^c \mathbf{C}_{\tilde{n}}^b ([\mathbf{y}^n - \mathbf{p}^n] \times) \quad (2.151)$$

This partial derivative can now be used in an estimation algorithm. Once an estimate of the small angle is available, $\hat{\boldsymbol{\psi}}$, the resulting estimate of the direction cosine matrix is

$$\hat{\mathbf{C}}_b^n = [\mathbf{I} - \hat{\boldsymbol{\psi}} \times] \mathbf{C}_b^{\tilde{n}} \quad (2.152)$$

In the next chapter, an analysis of the current state-of-the-art image-based navigation techniques is presented.

III. Navigation Using Imaging and Inertial Sensors - An Analysis of the State of the Art

Navigation using images and inertial measurements is an area of active research. This chapter describes the current state of the art and establishes a context for this work.

3.1 *Early Methods*

The use of various types of optical measurements for navigation is not new. In fact, some of the oldest navigation techniques are based on optical measurements of fixed objects. In this section, the classic techniques of celestial navigation and driftmeters are presented.

3.1.1 Celestial Navigation. The mathematical predictability and observability of the heavens provide an excellent source of navigation. Celestial navigation is one of the oldest known methods for long-distance navigation and was practiced by the world's earliest global travelers, the Phoenicians and the Polynesians [10]. Early navigators used the sky to determine direction by observing the stars rising in the east, setting in the west and rotating about the north. Later, development of equipment capable of measuring the angles between the horizon and stars (e.g., astrolabe, sextant), combined with precision timekeeping, enabled navigators to determine latitude and longitude directly [5]. In this section, the mathematics related to celestial observations are developed.

In simplest terms, the celestial observation geometry is shown in Figure 3.1. The celestial observation is a measurement of the angle from a star to the local vertical vector. This angle describes a circular line of potential positions centered on the projection of the star of the Earth's surface. The location of the star is described by the meridian angle and declination angle, λ_t and ϕ_t , respectively, from the inertial reference frame. The location can also be described by the unit vector $\check{\mathbf{t}}^i$.

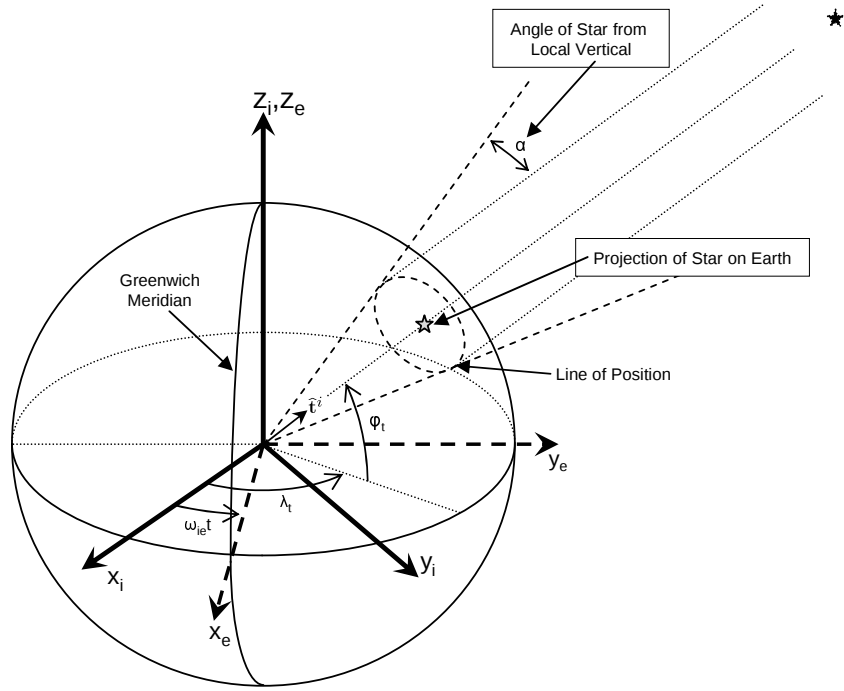


Figure 3.1: Celestial observation geometry. The celestial observation is a measurement of the angle from a star to the local vertical vector. This angle describes a circular line of potential positions centered on the projection of the star of the Earth's surface. The location of the star is described by the meridian angle and declination angle, λ_t and ϕ_t , respectively, from the inertial reference frame. The location can also be described by the unit vector $\hat{t}^i$.

The generalized location of a celestial body (e.g., star, planet, satellite, etc.) can be expressed in the Earth-centered inertial, or i , frame as the vector $\mathbf{t}^i$. The position vector can be expressed equivalently using a unit vector as

$$\mathbf{t}^i = D\check{\mathbf{t}}^i \quad (3.1)$$

where $D = \|\mathbf{t}^i\|$. The line of sight vector to the object in the ECEF frame, $\mathbf{s}^e$, is

$$\mathbf{s}^e = D\mathbf{C}_i^e\check{\mathbf{t}}^i - \mathbf{p}^e \quad (3.2)$$

where $\mathbf{p}^e$ is the location of the sensor in the e frame. And in the camera frame

$$\mathbf{s}^c = D\mathbf{C}_e^c\mathbf{C}_i^e\check{\mathbf{t}}^i - \mathbf{C}_e^c\mathbf{p}^e \quad (3.3)$$

The line of sight vector is normalized to homogeneous coordinates by dividing by the z component:

$$\underline{\mathbf{s}}^c = \frac{1}{s_z^c}\mathbf{s}^c \quad (3.4)$$

and then transformed to pixel coordinates:

$$\underline{\mathbf{s}}^{pix} = \mathbf{T}_c^{pix}\underline{\mathbf{s}}^c \quad (3.5)$$

where $\mathbf{T}_c^{pix}$ is the intrinsic camera projection matrix. Adding pixel measurement noise defines a measurement equation of the form

$$\mathbf{z}^{pix} = \begin{bmatrix} \underline{s}_x^{pix} \\ \underline{s}_y^{pix} \end{bmatrix} = \mathbf{h}(\mathbf{p}^e, \mathbf{C}_b^e, \mathbf{t}^e, \mathbf{T}_c^{pix}) + \mathbf{v} \quad (3.6)$$

The measurement can be simplified by noting the distance to a star is much greater than the radius of the Earth (e.g., $\mathbf{p}^e/D \approx 0$.) Returning to Equation (3.2)

and applying the infinite distance approximation, yields

$$\mathbf{s}^e \approx \mathbf{C}_i^e \check{\mathbf{t}}^i \quad (3.7)$$

which effectively removes $\mathbf{p}^e$ from the measurement. The resulting simplified measurement is

$$\mathbf{z}^{pix} = \begin{bmatrix} \underline{s}_x^{pix} \\ \underline{s}_y^{pix} \end{bmatrix} = \mathbf{h}(\mathbf{C}_b^e, \mathbf{t}^e, \mathbf{T}_c^{pix}) + \mathbf{v} \quad (3.8)$$

This shows celestial measurements are primarily useful for obtaining precise body orientation. This orientation measurement can be combined with inertial measurements to estimate position.

3.1.2 Driftmeter Navigation. Prior to the invention of the inertial navigation system, the *driftmeter* was used by bomber crews in World War II to determine groundspeed by tracking the apparent drift of objects on the ground [65]. Later, a method using optical measurements of the lunar surface was suggested to provide navigation information for the Apollo missions [39].

3.2 Non-inertial Optical Navigation Methods

Although this research is focused on the fusion of optical and inertial measurements, a discussion of non-inertial optical navigation methods illustrates relevant optical navigation techniques.

Hagen [17] proposes an extended Kalman filter for estimating an aircraft's position and orientation using optical measurements and a known topographic map of elevations. The algorithm uses feature detection techniques to select and track so-called "tokens" from a video sequence. These tokens are defined as a collection of circular integrals of pixel intensities:

$$e_i(x, y) = \int_0^{2\pi} \mathbf{I}(x + r_i \cos \alpha, y + r_i \sin \alpha) d\alpha \quad (3.9)$$

at various radii (r_i) which are invariant to linear shifts and rotations. The candidate tokens are then pruned to maximize a uniqueness measure which ensures the tokens are not too closely spaced and the terrain around the token is linear. Both of these constraints are designed to improve the stability of the extended Kalman filter.

The tokens are then tracked using a correlation algorithm which minimizes the norm of the difference between the original token and the current image. Search efficiency was improved by constraining the search area using the current navigation state estimate and covariance.

A different approach to estimating position using aerial images is proposed by Sim [55, 56] and generalized by Lerner [32]. The method begins by calculating a number of correspondence vectors, arranged in a grid pattern, between a pair of images. The resulting vector array is defined as the optical flow. The optical flow array is then compared to a digital terrain model corresponding to the scene. This results in two different, but complementary, updates. The average optical flow vector is treated as a velocity measurement and integrated to estimate position. Because the position is not measured directly, it is subject to drift over time. The second measurement type correlates the optical flow pattern with the predicted optical flow pattern calculated using the digital terrain model, similar to the Terrain Contour Matching (TERCOM) technique for cruise missile guidance [14]. This measurement provides a direct measurement of position relative to the Earth and serves to eliminate long-term drift. The resulting navigation solution uses an ad hoc approach to integrate the two measurements. The trajectory is then estimated using an iterative, nonlinear solver.

The effects of sensor geometry for navigation and obstacle avoidance were studied in [23]. In this research, an autonomous helicopter was equipped with two side-looking cameras and one omnidirectional video camera and flown through paths between tall buildings (i.e., urban canyons). The concept was motivated by the theorized use of optical flow by insects for relative navigation and obstacle avoidance [59]. The

vehicle was able to reliably determine the distance from each wall using the optical flow measurement.

This is related to a developing field which is attempting to use biological vision research to navigate using optical flow. Interestingly, the insect compound eye and tangential neuron structure has been shown to directly provide indications of optical flow patterns tuned to specific elementary motions [8, 13, 31]. This discovery has spawned a series of biomimetic or bio-inspired techniques for guidance and navigation. In addition, the parallel-friendly nature of optic-flow processing has led to the development of embedded very-large scale integration (VLSI) components with imaging sensors with the goal of providing elementary motion estimates [19, 21].

The advantages of incorporating a stereo imaging sensor for robot navigation over uncertain terrain was investigated in [41, 42]. The system modeled the locations of landmarks as a Gaussian-distributed random variable and estimation algorithms were developed using an optimal maximum-likelihood motion estimation model versus the less-optimal least-squares formulation. Initial simulation results demonstrated nonlinear error growth rate for optical-only navigation. Incorporating an attitude sensor reduced the error growth significantly.

This research also showed innovation in the method used to select and track features. The features were selected to maximize the Förstner interest operator [12] with detail in the direction parallel to both cameras. This is equivalent to tracking points which will provide the most accurate depth measurement. In addition, the predicted feature location is constrained using an ad hoc method based on assuming a relatively flat terrain model. Finally, the beneficial effects of tracking objects in multiple images for stabilizing and improving long-distance navigation is noted.

Applying Bayesian methods to solve the correspondence between features in multiple images was approached by Bedekar [2]. This maximum *a posteriori* (MAP) estimator solves for the best correspondence estimate while accounting for random camera errors and random feature location error. Let the vectors $\mathbf{z}_{1j}, \mathbf{z}_{2j}, \dots, \mathbf{z}_{nj}$

represent the observed feature locations for given feature, j , in image coordinates from a sequence of images $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n$, respectively. The true navigation state vector for each image $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n$ is given by $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, respectively. The estimated navigation state vector $\hat{\mathbf{x}}_n$ is modeled as a Gaussian random vector with mean equal to the true navigation state vector and covariance $\mathbf{P}_n$. The true location of each feature, j , in world coordinates is given for each epoch by $\mathbf{q}_{1j}, \mathbf{q}_{2j}, \dots, \mathbf{q}_{mj}$.

The observed feature locations can now be expressed as a function of the true location in world coordinates and the navigation state:

$$\mathbf{z}_{ij} = \mathbf{P}(\mathbf{x}_i, \mathbf{q}_j) + \mathbf{v}_{ij} \quad (3.10)$$

where $\mathbf{P}(\cdot, \cdot)$ is the function which projects features into the image plane based on the navigation state and $\mathbf{v}_{ij}$ is a random measurement noise. The triangulation problem is solved by choosing $\mathbf{q}$ and $\mathbf{x}$ which maximizes the *a posteriori* probabilities:

$$p(\mathbf{q}_j | \mathbf{z}_{1j}, \mathbf{z}_{2j}, \dots, \mathbf{z}_{nj}, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_n) \quad j = 1, 2, \dots, M_i \quad (3.11)$$

over the various correspondence combinations between true feature locations and the locations in the image.

This approach is important because it provides the basis for treating the navigation state and the feature location measurements as random variables, resulting in a navigation state estimate which is statistically rigorous. This concept could be extended into a stochastic process by incorporating a navigation state dynamics model and additional measurements.

3.3 *Inertially-aided Optical Navigation*

In this section, relevant approaches for inertially-aided optical navigation are described and analyzed.

3.3.1 Bhanu, Roberts, and Ming. Integrating inertial measurements with a digital image sequence to enhance object avoidance was proposed by Bhanu in [52] and [4]. This system leverages inertial measurements to calculate the translation and rotation between images to help solve the correspondence problem and estimate object distance. While the method is not directly related to solving the navigation problem, there are a number of similarities which motivate a discussion.

The algorithm consists of seven steps which are executed between pairs of images taken at times t_a and t_b .

1. Calculate the navigation state, $\tilde{\mathbf{x}}(t_a)$ and $\tilde{\mathbf{x}}(t_b)$ corresponding to images, $\mathbf{I}(t_a)$ and $\mathbf{I}(t_b)$, respectively.
2. Select interest points, $\mathbf{z}(t_a)$ and $\mathbf{z}(t_b)$, from each frame.
3. Locate the focus of expansion in each image.
4. Project the focus of expansion and interest points onto image plane parallel to image a .
5. Match the interest points in frame b to frame a .
6. Compute the range to each interest point.
7. Create a dense map of range to obstacles.

The interest points (i.e., features) are calculated using a derivative of the Harris feature detector with k parameter equal to zero (see Eqn. 2.95). The focus of expansion is the projection of the position difference between times t_b and t_a onto the image plane at time t_b . This focus of expansion vector is the common vanishing point for stationary objects due to translational camera motion.

The interest points in frame b are then “derotated” into an image plane parallel to frame a using the change in attitude detected by the inertial sensor and the projection operator. This establishes an equivalent theoretical image space which results from a purely translational motion. This, combined with the projection of the focus of expansion, allows the algorithm to more easily establish correspondence between

interest points in both images. The derotation operation is expressed by the following equation:

$$\mathbf{z}^{c_a}(t_b) = \mathbf{C}_n^{c_a} \mathbf{C}_{c_b}^n \mathbf{z}^{c_b}(t_b) \quad (3.12)$$

$$\mathbf{z}^{c_a}(t_b) = \mathbf{C}_{c_b}^{c_a} \mathbf{z}^{c_b}(t_b) \quad (3.13)$$

where $\mathbf{C}_{c_b}^{c_a}$ is the DCM from camera frame at time t_a to time t_b , $\mathbf{C}_n^{c_a}$ is the DCM from the navigation frame to the camera frame at time t_a , $\mathbf{C}_{c_b}^n$ is the DCM from the camera frame at time t_b to the navigation frame, $\mathbf{z}^{c_b}(t_b)$ and $\mathbf{z}^{c_a}(t_b)$ are the location of interest points at time t_b coordinatized in the c_b and c_a frames, respectively. The focus of expansion, $\mathbf{p}_{ab}^{c_b}$, is simply the rotation of the position difference between epochs t_a and t_b into the camera frame at time t_b

$$\mathbf{p}_{ab}^{c_b} = \mathbf{C}_{c_a}^{c_b} \mathbf{p}_{ab}^{c_a} \quad (3.14)$$

Once the image is derotated and the focus of expansion is established, the correspondence between interest points is calculated using goodness-of-fit metrics. One relevant metric is the correspondence search constraint placed on each point. This constraint ensures each interest point lies in a cone-shaped region, with apex at the focus of expansion, bisected by the line joining the focus of expansion and the interest point in the camera frame at t_a . While this constraint is not statistically rigorous, it does show the value of using inertial measurements to aid the correspondence problem, which is an important contribution. The second contribution of this work is the concept of applying transformations to the image based on inertial measurements to produce a motion model which is more tractable. In this example, a purely translational model was preferred; however, other transformations are possible.

3.3.2 Roumeliotis, Johnson, and Montgomery. Another interesting proposal for augmenting inertial navigation with image-based motion estimation is presented in [53]. In this paper, the problem of landing an interplanetary probe on Mars is

addressed. The approach combines measurements from an inertial sensor, image sensor, and laser altimeter using an extended Kalman filter. What makes this work unique is the direct exploitation of scene homography for use as updates to a filter. In other words, based on an assumed relatively flat scene (e.g., looking down at the ground during descent), the feature tracking algorithm can produce a measurement of pitch and roll.

A comparison of homographic techniques is presented in [1], where a homography is defined as a 3×3 matrix capable of describing the projective transformation of points viewed on a plane between images. In order to compute a homography, a minimum of four point correspondences from objects on a plane are required. As can be expected, homographic methods work the best when the scene is well-modeled as a plane; however, the method can be extended to arbitrary, but known, structures.

3.3.3 Pachter, Polat, and Porter. A theory for INS aiding by tracking a ground object at unknown location is presented in [44, 46, 47]. In this approach, an aircraft is equipped with a telescope with a gimbal mounting which enables the user to measure angles and angular rates to a fixed ground target, similar to a modern driftmeter [65].

For clarity, the initial derivation is performed using a two-dimensional profile. Assuming a straight flight path with constant angle of attack, the location of the unknown target and the aircraft velocity is estimated using measurements of the initial deflection angle and angular rate from the driftmeter. Once these parameters are estimated, the angle from the aircraft velocity vector to the initial line of sight (LOS) to the target is calculated. Assuming a non-accelerating or rotating flight path, the angle of attack can then be estimated directly from future angle measurements to the target. Finally, this angle of attack measurement is used to estimate the navigation state using inertial measurements and a weighted least squares formulation. This illustrates a mathematical technique for estimating angle of attack and sideslip using optical and inertial measurements for non-maneuvering flight.

3.3.4 *Raquet and Giebner.* Raquet proposes an extended Kalman filter which combines optical measurements of an object at unknown location with inertial and barometric altimeter measurements [51]. In the following paragraphs, an overview of the filter derivation and the flight test results are presented.

The filter is a tightly-coupled, feed-forward design, error state model, commonly used in inertial navigation systems [37]. A block diagram of the filter is shown in Figure 3.2. As mentioned previously, the filter estimates the *errors* in the navigation state produces by the INS. This perturbation model reduces the effects of unmodeled super-linear dynamics and improves performance [37,38]. Thus, the state vector, $\delta\hat{\mathbf{x}}$, which includes the inertial error states, can be defined as

$$\delta\hat{\mathbf{x}} = \begin{bmatrix} \delta\mathbf{p}^n \\ \delta\mathbf{v}^n \\ \boldsymbol{\psi} \\ \delta h_{baro} \\ \mathbf{t}_1^n \\ \vdots \\ \mathbf{t}_n^n \end{bmatrix} \quad (3.15)$$

where $\delta\mathbf{p}^n$, $\delta\mathbf{v}^n$, and $\boldsymbol{\psi}$ are the position, velocity, and attitude errors, respectively. The barometric altimeter error, δh_{baro} is modeled as a first-order Gauss-Markov process [37]. The three dimensional target location estimates for n ground targets are represented by $\mathbf{t}_1^n \dots \mathbf{t}_n^n$. The targets are modeled as stationary (zero dynamics) with a small additive random walk (process noise) to avoid filter lock-out caused by a covariance eigenvalue approaching zero [37].

The state dynamics are a function of the INS error model and are expressed as a stochastic linear differential equation:

$$\delta\dot{\mathbf{x}} = \mathbf{F} \left(\hat{\mathbf{p}}^n, \hat{\mathbf{C}}_b^n, \mathbf{F}_m^b, \boldsymbol{\omega}_{ib_m}^b \right) \delta\mathbf{x} + \mathbf{G} \left(\hat{\mathbf{C}}_b^n \right) \mathbf{w} \quad (3.16)$$

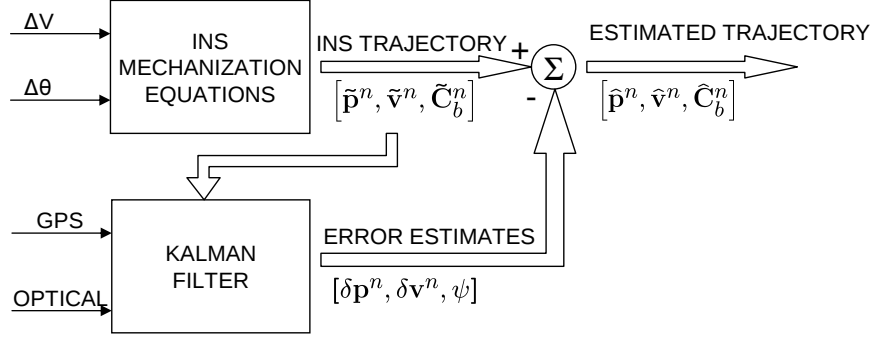


Figure 3.2: Block diagram of feed-forward navigation Kalman filter. The Kalman filter uses external measurements to estimate the *errors* in the current navigation state produced by an inertial navigation system (INS). The estimated errors are then removed from the INS state resulting in a nearly optimal estimate.

which is a function of the current estimated trajectory and inertial measurements (see Section 2.6.)

The optical measurement model describes the pointing vector from the aircraft to the each target as a function of the current state. The measurement equation for target n is expressed as

$$\mathbf{z}_n = \begin{bmatrix} \alpha_n \\ \beta_n \end{bmatrix} = \mathbf{h}_n(\mathbf{x}) + \mathbf{v} \quad (3.17)$$

where α_n and β_n are the horizontal and vertical angles to target n . The $\mathbf{x}$ vector is the true navigation state. Using the method of perturbations, the measurement is linearized about the estimated aircraft state:

$$\delta \mathbf{z}_n = \mathbf{H}_n \delta \mathbf{x} + \mathbf{v} \quad (3.18)$$

where $\mathbf{H}_n$ is the Jacobian of the nonlinear measurement function, $\mathbf{h}_n[\cdot]$, about the current state estimate. Note that the elements of the $\mathbf{H}$ matrix related to orientation error are not modeled in this implementation.

The linearized dynamics and measurement update equations were incorporated into an extended Kalman filter and tested using flight test data. The test trajectory was a circular pattern, approximately three kilometers above four fixed targets,

chosen to keep the targets in the camera field of view. The pixel location to each target was measured by hand and converted to angular measurements for processing. Incorporating optical measurements improved the navigation position error by 74 percent over the free inertial solution. This demonstration showed the benefits of incorporating optical measurements of objects at unknown locations with inertial sensors.

3.3.5 Strelow, Singh, and Adams. Strelow [60–62] approaches the problem of integrating optical and inertial sensors using optimal online and batch techniques. Optimal online processing estimates the current navigation state conditioned on all prior measurements, using a some statistical measure of optimality. Note that the definition of optimality is dependent on the actual measure chosen. As the name implies, online methods can theoretically be used to estimate the navigation state in real-time, onboard some vehicle. Conversely, batch processing involves estimating an optimal trajectory, conditioned on *all* measurements for the entire trajectory. Batch methods typically have improved performance over online methods, especially in highly non-linear situations such as correspondence solutions, and can serve to define the best achievable performance for a given trajectory and measurement realization [60].

The online algorithm described in [61] integrates inertial and optical measurements using an extended Kalman filter algorithm. Using notation consistent with the

development in Chapter II, the state vector for n tracked targets, $\mathbf{x}(t)$, is

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{p}^e(t) \\ \mathbf{v}^e(t) \\ \dot{\mathbf{v}}^e(t) \\ \boldsymbol{\phi}(t) \\ \boldsymbol{\omega}_{eb}^b(t) \\ \mathbf{a}^{b_0} \\ \mathbf{b}^{b_0} \\ \mathbf{g}^{b_0} \\ \mathbf{t}_1^e \\ \vdots \\ \mathbf{t}_n^e \end{bmatrix} \quad (3.19)$$

where $\mathbf{p}^e$, $\mathbf{v}^e$, and $\dot{\mathbf{v}}^e$ are the position, velocity, and acceleration vectors; $\boldsymbol{\phi}$ is a vector of Euler angles from the world frame to the body frame; $\boldsymbol{\omega}_{eb}^b(t)$ is the angular velocity vector; $\mathbf{a}^{b_0}$, $\mathbf{b}^{b_0}$, and $\mathbf{g}^{b_0}$ are the accelerometer bias, gyro bias and gravity vectors with respect to the body coordinate frame at $t = 0$, respectively. Finally, the target locations in the world frame are expressed by the vectors, $\mathbf{t}_1^e, \dots, \mathbf{t}_n^e$. The state estimate and covariance is propagated using the extended Kalman filter algorithm presented in Section 2.8.2.

The image measurements utilizes the pixel location measurement for each target in the image, $\mathbf{z}_n(t_i)$, which is expressed as the projection of the relative Euclidian target location into the image plane:

$$\mathbf{z}_n(t_i) = \begin{bmatrix} x_{pix}(t_i, n) \\ y_{pix}(t_i, n) \end{bmatrix} = \boldsymbol{\pi} [\mathbf{C}_e^c(t_i) \mathbf{s}_n^e(t_i)] \quad (3.20)$$

where $\boldsymbol{\pi}$ is the projection operator, x_{pix} and y_{pix} is the target location in pixels, and $\mathbf{C}_e^c(t_i)$ is the DCM from the ECEF to the camera frame at time t_i . The three

dimensional target line of sight vector, $\mathbf{s}_n^e(t_i)$ is defined as

$$\mathbf{s}_n^e(t_i) = \mathbf{t}_n^e - \mathbf{p}^e(t_i) \quad (3.21)$$

In this derivation, note that the camera frame is aligned with the body frame, (i.e., $\mathbf{C}_b^c = \mathbf{I}$.)

In addition, the accelerometer and gyroscope measurements, $\mathbf{z}_a(t_i)$ and $\mathbf{z}_b(t_i)$, respectively, are incorporated as independent measurements using the measurement equations:

$$\mathbf{z}_a = \mathbf{C}_e^b \dot{\mathbf{v}}^e - \mathbf{C}_{b_0}^b [\mathbf{g}^{b_0} + \mathbf{a}^{b_0}] + \mathbf{v}_a \quad (3.22)$$

$$\mathbf{z}_b = \boldsymbol{\omega}_{eb}^b - \mathbf{b}^b + \mathbf{v}_b \quad (3.23)$$

where $\mathbf{v}_a$ and $\mathbf{v}_b$ are additive random noise vectors.

The filter is initialized using a short initial measurement sequence and a batch algorithm to estimate visible features. Features which become visible after the filter is initialized are incorporated using a stochastic map approach, described in [58], if certain criteria are met.

The candidate target must first be tracked through a small, predetermined number of frames in order to provide an initial estimate for the three-dimensional location, $\hat{\mathbf{t}}_n$, and to indicate some level of stability in the track. Once a three dimensional solution is available for a candidate target, a quality measure is calculated based upon the following heuristic. Let l describe the length of the longest axis of the covariance matrix, $\mathbf{P}_{tt}$, associated with the candidate target. The ratio $r = l/b$ is calculated, where b is the longest baseline between camera centers used to calculate the initial target location. If r is less than a fixed threshold, the target state is added to the filter using the state augmentation procedure described in the next paragraph. Note that the r heuristic is a measure of the “goodness” of the geometry described by the trajectory and target location.

The new target is added to the current state vector. The state covariance matrix is augmented using

$$\mathbf{P}_{tt}^e = \mathbf{G}_x \mathbf{P}_{xx} \mathbf{G}_x^T + \mathbf{G}_t \mathbf{P}_{tt}^b \mathbf{G}_t^T \quad (3.24)$$

$$\mathbf{P}_{xt}^e = \mathbf{G}_x \mathbf{P}_{xx} \quad (3.25)$$

where $\mathbf{P}_{xx}$ is the current filter covariance. $\mathbf{G}_t$ and $\mathbf{G}_x$ are the Jacobians of the transformation from body (camera) to ECEF frame:

$$\mathbf{G}_t = \left. \frac{\partial \mathbf{C}_b^e}{\partial \mathbf{t}} \right|_{\mathbf{x}=\hat{\mathbf{x}}, \mathbf{t}=\hat{\mathbf{t}}_n} \quad (3.26)$$

$$\mathbf{G}_x = \left. \frac{\partial \mathbf{C}_b^e}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}, \mathbf{t}=\hat{\mathbf{t}}_n} \quad (3.27)$$

According to the authors, this method accounts for noise in the image observations and state estimate, but does not account for any relative error between the recent state estimates. Cases where this error is significant (e.g., point feature correspondence errors) are recommended for future research.

The batch algorithm uses the Levenberg-Marquardt method [50] to minimize a nonlinear cost function representing the total error of the current trajectory. The overall cost function is the sum of the visual and inertial errors:

$$E_{combined} = E_{visual} + E_{inertial} \quad (3.28)$$

The visual error term, E_{visual} , is an expression of the image reprojection error

$$E_{visual} = \sum_{i,j} D(\boldsymbol{\pi}_c^{pix} [\mathbf{C}_e^c(t_i) \mathbf{t}_j^e], \mathbf{s}_{ij}^{pix}) \quad (3.29)$$

where $\mathbf{s}_{ij}^{pix}$ is the observed pixel location of target j at time t_i , $\mathbf{t}_j^e$ is the location of target j in the ECEF coordinate frame, $\mathbf{C}_e^c(t_i)$ is the transformation from the world

to camera frame at time t_i , and $\pi_c^{pix}[\cdot]$ is the image projection function. $D(\cdot)$ is a Mahalanobis (weighted two-norm) distance function.

The inertial error term, $E_{inertial}$, is an expression of the errors in the position, velocity, and attitude vectors:

$$\begin{aligned}
E_{inertial} = & \sum_{i=1}^n D[\mathbf{p}^e(t_i), \mathbf{p}_{inertial}^e(t_i) | \mathbf{p}_{inertial}^e(t_{i-1})] \\
& + \sum_{i=1}^n D[\mathbf{v}^e(t_i), \mathbf{v}_{inertial}^e(t_i) | \mathbf{v}_{inertial}^e(t_{i-1})] \\
& + \sum_{i=1}^n D[\boldsymbol{\psi}(t_i), \boldsymbol{\psi}_{inertial}(t_i) | \boldsymbol{\psi}_{inertial}(t_{i-1})] \quad (3.30)
\end{aligned}$$

where $\mathbf{p}_{inertial}^e(t_i) | \mathbf{p}_{inertial}^e(t_{i-1})$ is the inertial position solution at t_i integrated from the inertial position solution at t_{i-1} , $\mathbf{v}_{inertial}^e(t_i) | \mathbf{v}_{inertial}^e(t_{i-1})$ is the inertial velocity solution at t_i integrated from the inertial velocity solution at t_{i-1} , and $\boldsymbol{\psi}_{inertial}(t_i) | \boldsymbol{\psi}_{inertial}(t_{i-1})$ is the inertial attitude solution (expressed in Euler angles) at t_i integrated from the inertial attitude solution at t_{i-1} . $\mathbf{p}^e(t_i)$, $\mathbf{v}^e(t_i)$, and $\boldsymbol{\psi}(t_i)$ are the camera position, velocity, and attitude at time t_i . The distance function, $D(\cdot)$, is also a Mahalanobis distance function.

The online and batch algorithms are then tested using a calibrated camera and inertial sensor mounted on a robotic arm. The sensor is subjected to a four degree-of-freedom trajectory, resembling a clover pattern. The tests are performed using a combination of methods:

- Batch processing using image and inertial measurements.
- Online processing using 15, 30, and 45 images for initialization
- Batch processing using images only.

The results show the value of adding inertial measurements to vision processing. The performance of the image-only batch algorithm is unable to converge on a realistic solution under the test conditions. Combining image and inertial measurements using

batch processing results in the best performance. The online algorithms combining inertial and visual measurements perform much better than the visual only, and show improved performance with increased batch initialization size.

In addition to the batch and online processing methods, Strelow also proposes incorporating inertial measurements to constrain the correspondence search between image frames [62]. This constraint on the image search space is a similar concept to the field of expansion method proposed by Bhanu [4]; however, Strelow generalizes the approach by exploiting epipolar geometry.

The projection of an arbitrary point in an image is described by an epipolar line in a second image (see Figure 3.3.) All epipolar lines in an image converge at the projection of the focus of the complimentary image. Combining knowledge of the translation and rotation between images and the pixel location of a candidate target in the first image, a correspondence search is then constrained to an area near the epipolar line. This approach is shown in Figure 3.4.

There are some limitations in Strelow’s work which warrant further research. First, while he develops an initial concept of using inertial measurements to constrain the correspondence search along an epipolar line, it is ad hoc in the sense that is not a stochastically rigorous development. In addition, the planar correspondence search constraints needs to be expanded to multiple dimensions when incorporating higher dimensional features. Another limitation of Strelow’s work is the limited applicability to an airborne environment, since his experiments are restricted to captive robot arms and land-based robots with very limited range. An airborne platform is expected to navigate over a relative large area, without visual overlap, for a period of hours. Finally, he does not exploit *a priori* world and scene information, which could be of significant value to an operationally useful navigation system.

In the next chapter, image and inertial navigation algorithms are proposed which address the shortcomings identified in the previous sections.

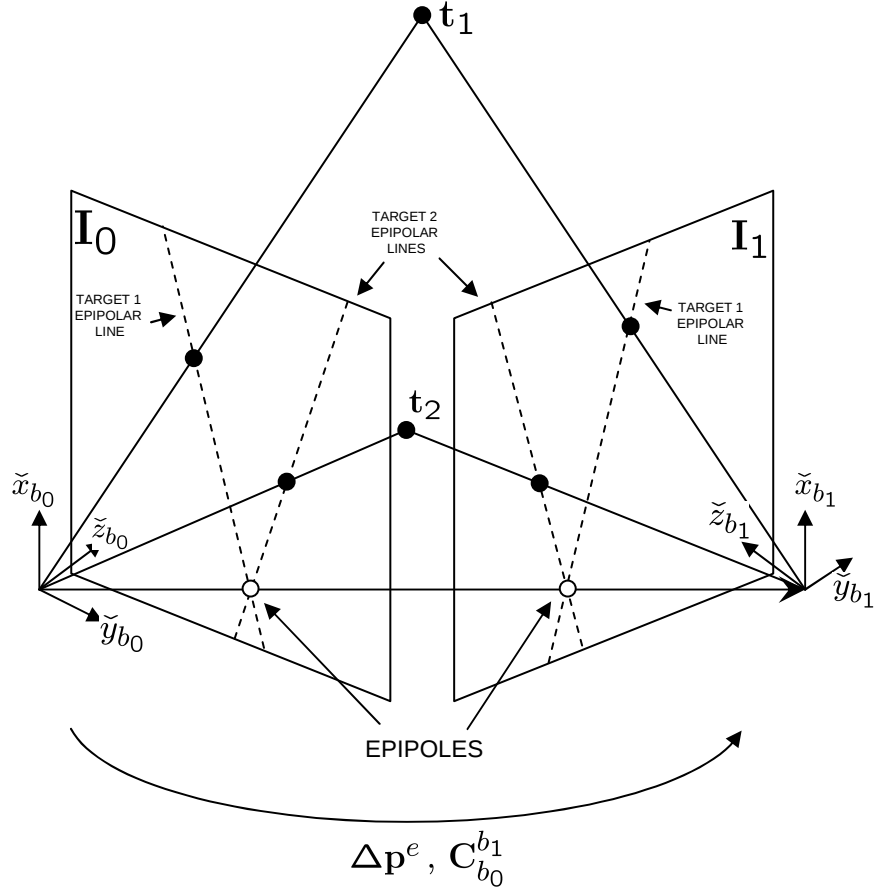


Figure 3.3: Epipolar geometry between two images. The projection of an arbitrary point in an image is described by an epipolar line in a second image. All epipolar lines in an image converge at the projection of the focus of the complimentary image.

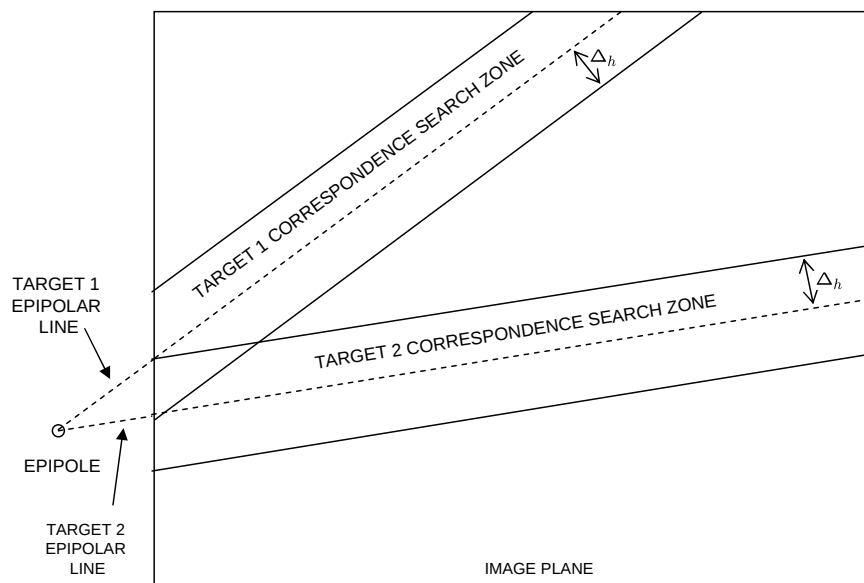


Figure 3.4: Correspondence search constraint using epipolar lines. Given a projection of an arbitrary point in an initial image, combined with knowledge of the translation and rotation to a second image, the correspondence search can be constrained to an area near the epipolar line. Note the epipole can be located outside of the image plane, as shown in this example.

IV. Image and Inertial Fusion Navigation Algorithm

This chapter presents an innovative approach for rigorously integrating imaging and inertial sensors at a deeper level than the current state-of-the-art. The chapter is organized as follows. First, the image and inertial fusion problem is presented and explained through a series of fundamental concepts. Next, these concepts are developed using rigorous mathematical techniques. These mathematical techniques are then incorporated into an extended Kalman filter algorithm used to implement the image and inertial fusion algorithm for online navigation. Finally, a discussion of tracking maintenance and various potential image measurements which can be exploited to improve the navigation solution are presented.

4.1 *Proposed Image and Inertial Fusion Algorithm Walkthrough*

In this section, the fundamental steps of the image and inertial fusion for navigation problem are presented in order to provide a conceptual reference for the forthcoming detailed development. The concept is based on the following assumptions:

- A strapdown inertial sensor is available, and rigidly mounted relative to one, or multiple digital cameras.
- The digital camera(s) capture images at some interval. The time of image capture is known relative to the inertial measurements.
- An initial navigation state with known accuracy statistics is available.
- An estimate of the distance to objects in the scene is available either through exploitation of knowledge of the world (e.g., terrain models) or through binocular stereopsis.

Based on the above assumptions, an example navigation trajectory is shown in Figure 4.1.

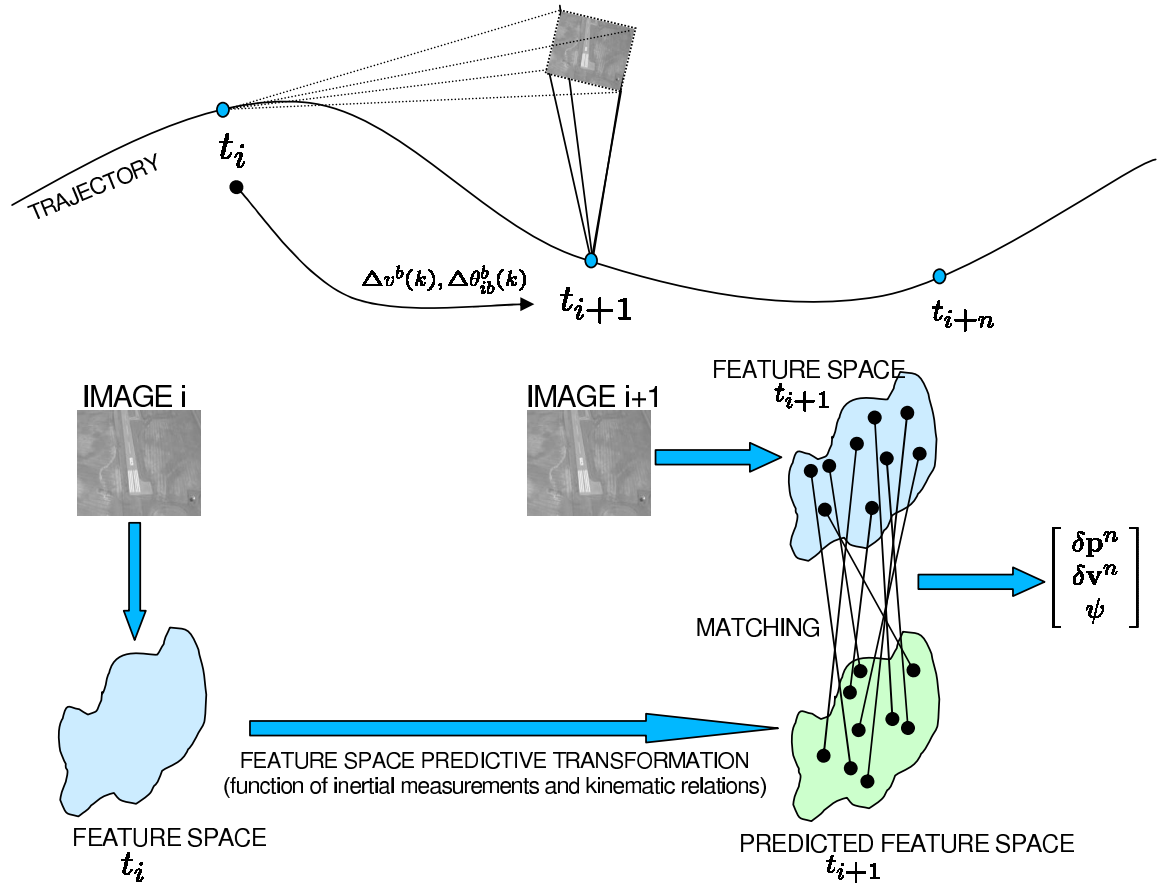


Figure 4.1: Overview of image-aided inertial algorithm.

The algorithm consists of the following fundamental steps:

- Image capture
- Transformation from image to feature space
- Propagation of navigation state to next image event
- Feature space propagation to next image event
- Statistical feature correspondence
- Trajectory error estimation

As presented in Chapter II, the digital imaging device measures the light intensity pattern projected through optics onto a sensor. The projection is a nonlinear function of the scene and lighting conditions, the camera optics, and the pose of the camera relative to the scene. The measured image can be thought of as a surface in three dimensions (two spatial and one intensity) which is corrupted by measurement noise, optical distortions, and possibly spatial aliasing.

While camera images are easily interpreted by humans, a transformation of the image to a space which more directly supports the automated navigation algorithm is desired. This space is termed the *feature space*. In general terms, the feature space transformation transforms the image array into a collection of *feature vectors*, located in the feature space. Many potential feature transformations have been proposed (see Section 2.7.4). The most desirable feature space transformation for exploiting image navigation information decomposes the feature space into separable *pose* and *object* dimensions. An ideal set of pose dimensions would be completely dependent on the relative pose of the object, but independent on the type of object or feature. Conversely, the ideal set of object dimensions would be completely independent of the relative pose of the object, and only depend on the type of object. This is similar to human interpretation of objects. For example, a pencil always looks like a pencil, no matter what orientation or size it appears to our eyes. Although no algorithm is currently known which can achieve true independence between the pose

and object bases, feature transformation algorithms have been proposed which display scale and rotation independence. The details of the feature transformation algorithm investigated during this work are discussed in the next section.

Once a collection of features is calculated for the current image, inertial measurements are integrated so that the relative change in pose can be estimated when the next image is captured. As described in Section 2.5.3, the calculated navigation state vector can be expressed as nonlinear differential equations which are functions of the accelerometer and gyroscope measurements, initial conditions, gravity vector, and Earth rotation rate vector. In addition to estimating the mean of the navigation state, the navigation state uncertainty covariance is propagated using the navigation error model and the Kalman filter propagation relations shown in Section 2.6.1 and 2.8, respectively.

The estimated changes in the navigation state are then used to predict the location of features through a stochastic projection transformation. This transformation operates on the pose dimensions of each feature vector from the previous image and results in an estimated pose and pose uncertainty at the time of the next image. The concept is illustrated in Figure 4.2. This predicted collection of feature vectors can then be compared to the features detected in the current image. This comparison consists of matching feature vectors from one epoch to the next by comparing the respective feature vectors using a rigorous, statistical weighting. This rigorous statistical feature matching based on inertial measurements integrates the image and inertial sensors at a deeper level than previous methods and is a significant contribution of this research. The statistical matching is described in detail in Section 4.2.

Once feature matches are determined, the errors between the predicted feature location and the actual feature location are used to correct errors in the navigation state. This is accomplished using the Kalman filter update equations described in Section 2.8 and discussed for this specific implementation in Section 4.2.

The cycle is then repeated for each image captured. In the next section, a detailed discussion of the components of the image and inertial fusion algorithm is presented.

4.2 Detailed Discussion of Image and Inertial Fusion Algorithm

In this section, the components of the image and inertial fusion algorithm are presented in more mathematical detail in order to illustrate the general feature transformation concepts presented in Section 4.1.

4.2.1 Stochastic Feature Transformation. As mentioned in the previous section, the transformation of a captured image into feature space is an important step toward extracting navigation information. For this work, this is accomplished using a variant of the scale-invariant feature tracking (SIFT) algorithm developed by Lowe [33]. While the SIFT algorithm is not a contribution of this research, it is presented in this section for clarity as it is the starting point for much of the research that follows. The SIFT feature transformation algorithm produces feature vectors with pose and object dimensions which are decoupled in translation, scale, and rotation and partially decoupled in affine (warping) motions. These characteristics make this algorithm attractive for navigation estimation, since the stochastic projection can be applied to the pose dimensions with only small effects on the object dimensions.

The feature transformation consists of three components: scale-space decomposition, feature detection, and calculation of the feature description vector. The scale-space decomposition is calculated using Gaussian and Gaussian difference spatial filters [34]. A Gaussian spatial filter or “Gaussian blur” is defined as the following function:

$$\mathbf{g}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.1)$$

where x and y correspond to the spatial dimensions of the image in pixels and σ is the standard deviation of the blurring function. Thus, given an image, $\mathbf{i}(x, y)$, the

filtered image can be represented by a convolution in x and y :

$$\mathbf{l}(x, y, \sigma) = \mathbf{i}(x, y) \otimes \mathbf{g}(x, y, \sigma) \quad (4.2)$$

The above function can also be written equivalently as a multiplication in the spatial frequency domain:

$$\mathbf{L}(f_x, f_y, \sigma) = \mathbf{I}(f_x, f_y) \mathbf{G}(f_x, f_y, \sigma) \quad (4.3)$$

where $\mathbf{I}(f_x, f_y)$ is the two-dimensional Fourier transform of the image and $\mathbf{G}(f_x, f_y, \sigma)$ is the two-dimensional Fourier transform of the Gaussian blur function. For illustrative purposes, the spatial frequency response of the one-dimensional Gaussian blur function is shown in Figure 4.3. Note the cutoff frequency of the Gaussian function decreases as the standard deviation, σ , increases.

The difference of Gaussian filter is defined as:

$$\mathbf{f}(x, y, k, \sigma) = \mathbf{g}(x, y, k\sigma) - \mathbf{g}(x, y, \sigma) \quad (4.4)$$

where x and y correspond to the spatial dimensions of the image in pixels, $k > 1$ is the scaling frequency step constant, and σ is the base standard deviation. Thus, given an image, $\mathbf{i}(x, y)$, the filtered image can be represented by a convolution in x and y :

$$\mathbf{d}(x, y, k, \sigma) = \mathbf{i}(x, y) \otimes \mathbf{f}(x, y, k, \sigma) \quad (4.5)$$

Sample impulse responses of the difference of Gaussian filter are shown in Figure 4.4. The difference of Gaussian function can also be written equivalently as a multiplication in the spatial frequency domain:

$$\mathbf{D}(f_x, f_y, k, \sigma) = \mathbf{I}(f_x, f_y) \mathbf{F}(f_x, f_y, k, \sigma) \quad (4.6)$$

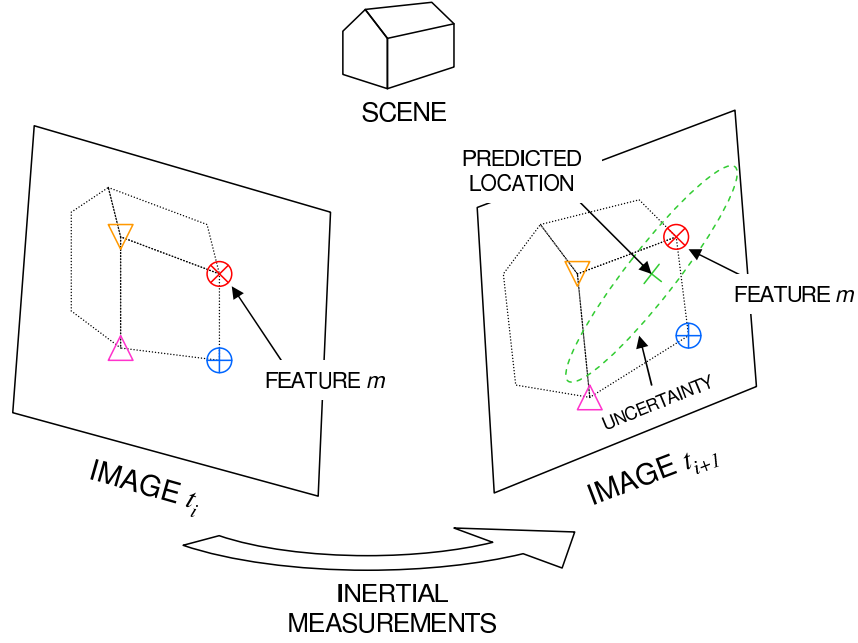


Figure 4.2: Stochastic feature projection. Optical features of interest are mapped into future images using inertial measurements and stochastic projections.

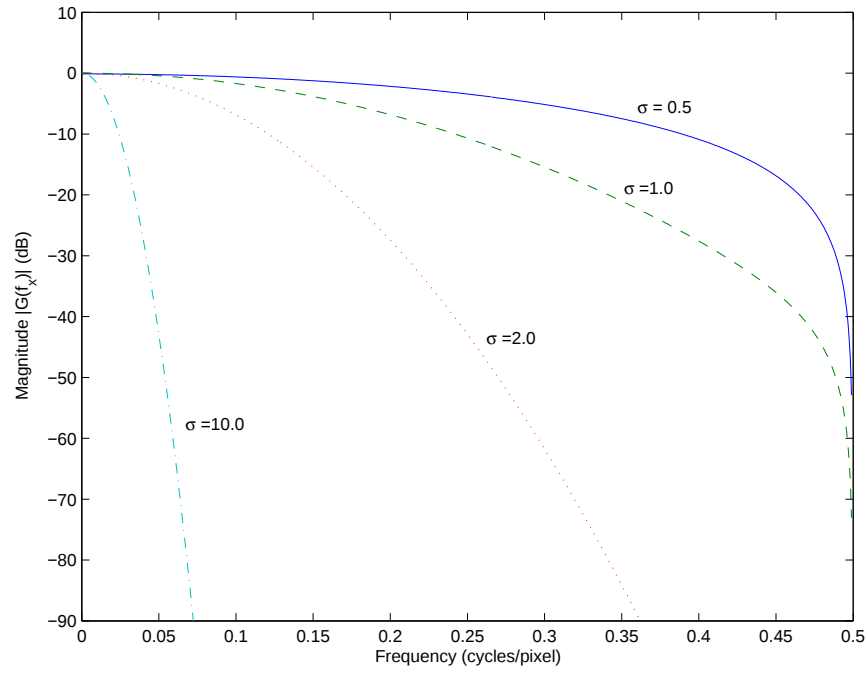


Figure 4.3: Frequency response of the Gaussian blur filter for varying blurring function standard deviations (σ). The cutoff frequency of the Gaussian function decreases as the standard deviation, σ , increases. Only one dimension of the function is shown for clarity.

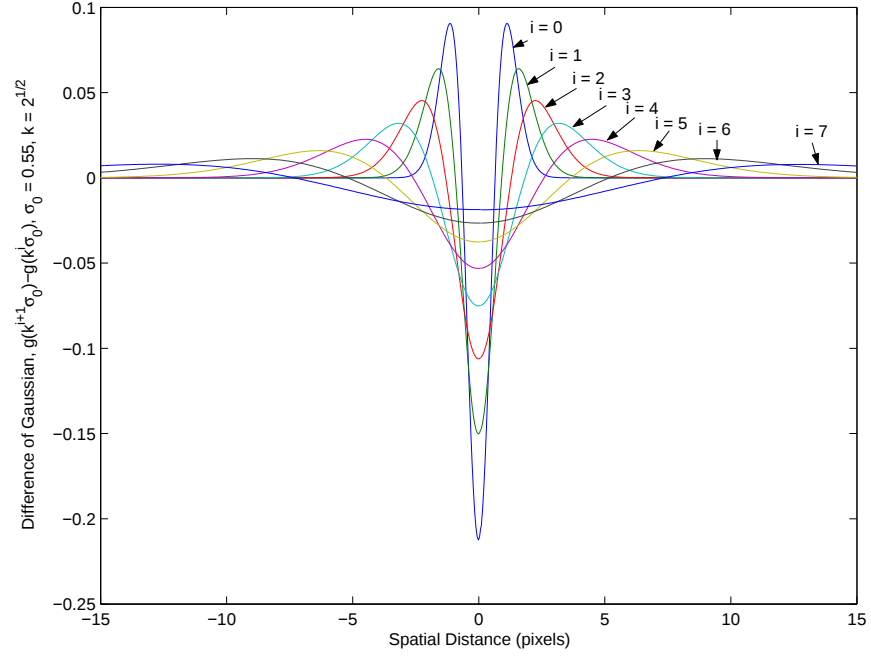


Figure 4.4: Impulse response of the difference of Gaussian filter.

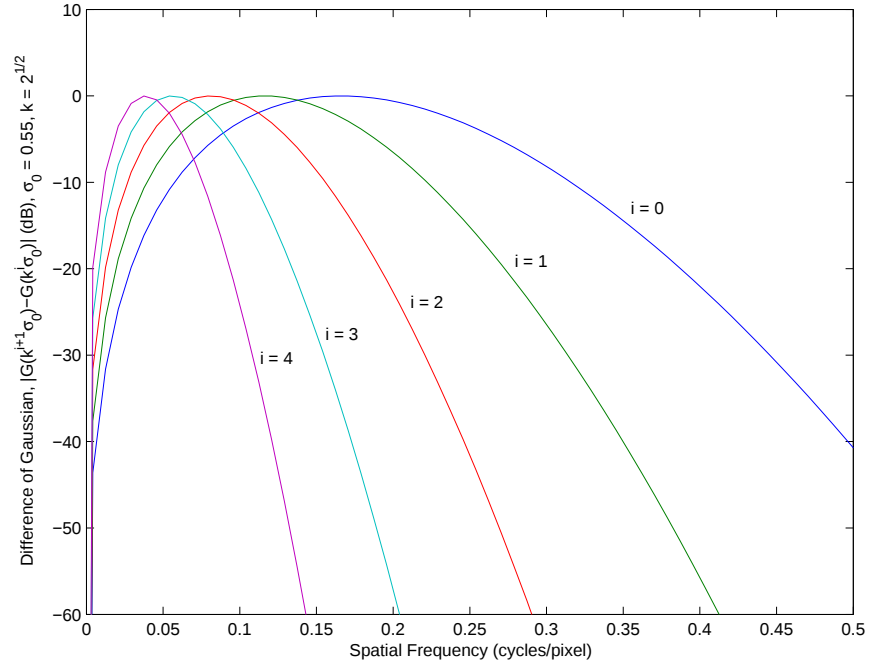


Figure 4.5: Frequency response of the difference of Gaussian filter. The center frequency of the bandpass response is a function of the standard deviation values of each component Gaussian function.



Figure 4.6: Sample image of airfield.

The spatial frequency response of the one-dimensional difference of Gaussian filter is shown in Figure 4.5. The spatial bandpass nature of the difference of Gaussian filter passes intensity variations in the image of a specific frequency, or scale.

By varying the scale parameter, $k > 1$, the image can be decomposed into multiple scale spaces, each of which is centered on a specific spatial frequency. In [34], the scale parameter is varied geometrically to maintain equal frequency spacing per octave of spatial frequency. Given an initial Gaussian filter standard deviation, σ_0 , and spacing parameter, k , the i th difference of Gaussian filter is defined as

$$\mathbf{f}(x, y, i) = \mathbf{g}(x, y, k^{i+1}\sigma_0) - \mathbf{g}(x, y, k^i\sigma_0) \quad (4.7)$$

A scale decomposition of a reference image (Figure 4.6) is shown in Figure 4.7. Note as the center frequency of the difference of Gaussian filter increases (with decreasing i), the filtered images show an increased sensitivity to higher spatial frequency detail.

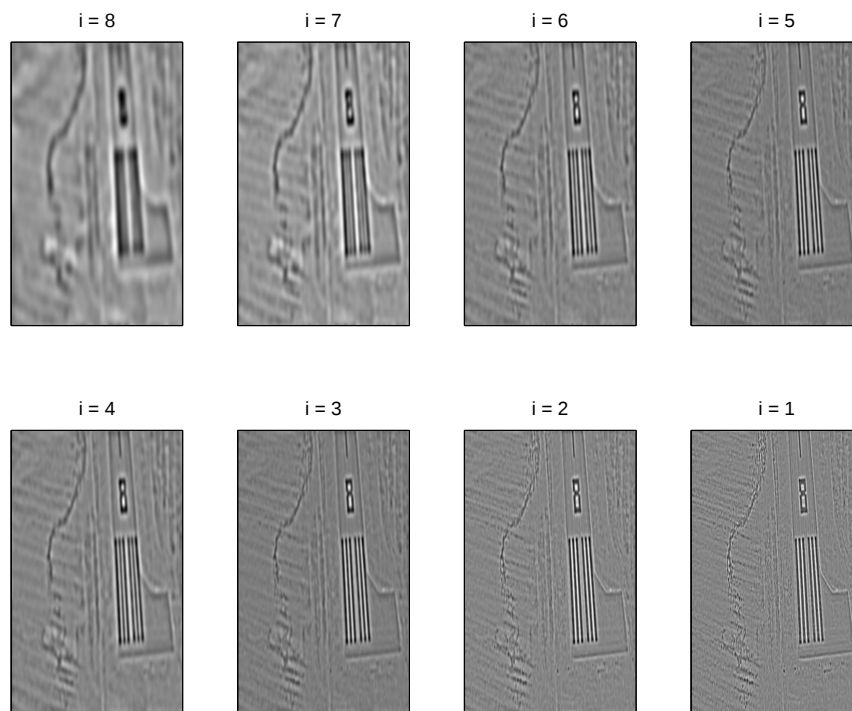


Figure 4.7: Sample image scale decomposition. As the filter center frequency increases (with decreasing i), the filtered images show increased sensitivity to higher spatial frequency detail.

After the image is decomposed into scale space, features are then detected which exhibit some local distinctiveness. This is accomplished by locating a voxel (i.e., volume element) which is the local maxima or minima of the 26 adjacent voxels in x , y , and scale (i). Once a candidate feature is located in a particular scale space, the amount of spatial detail is found by calculating the eigenvalues of the following matrix over a window, $\mathbf{W}$, which is centered on the candidate feature:

$$\mathbf{G} = \begin{bmatrix} \sum_{x,y \in \mathbf{W}} (\nabla \mathbf{f}_x)^2 & \sum_{x,y \in \mathbf{W}} \nabla \mathbf{f}_x \nabla \mathbf{f}_y \\ \sum_{x,y \in \mathbf{W}} \nabla \mathbf{f}_x \nabla \mathbf{f}_y & \sum_{x,y \in \mathbf{W}} (\nabla \mathbf{f}_y)^2 \end{bmatrix} \quad (4.8)$$

where $\nabla \mathbf{f}_x$ and $\nabla \mathbf{f}_y$ are the gradients of the scale space image $\mathbf{f}(x, y, i)$ in the x and y directions, respectively. As mentioned previously in Section 2.7.4, a zero eigenvalue corresponds to zero detail in the direction of the eigenvector, which results from a constant intensity image. The eigenvalues increase with increasing detail, indicating a potentially distinct feature. Features are declared by thresholding the eigenvalues and choosing the strongest candidates, which is refined using the Harris metric [18], which was introduced in Equation (2.95), and repeated here for clarity:

$$C(\mathbf{G}) = \det(\mathbf{G}) + k_t [\text{trace}^2(\mathbf{G})] \quad (4.9)$$

which is equivalent to

$$C(\mathbf{G}) = (1 + 2k_t)\sigma_1\sigma_2 + k_t(\sigma_1^2 + \sigma_2^2) \quad (4.10)$$

where σ_1 and σ_2 are the eigenvalues of $\mathbf{G}$, and k_t is a tuning parameter. Smaller values of k_t favor detail in both directions, while larger values of k_t are more accepting of detail in only one direction [18].

Once features are selected which have sufficient detail in multiple directions, the object dimensions of the feature vector are calculated as a function of the intensity gradient of the filtered scale space near the feature. The method presented by Lowe

builds a histogram of gradient orientations around the feature, then selects a primary orientation of gradient vector which corresponds to the maximum histogram bin. The object dimensions of the feature vector are then composed of the normalized histograms of gradient around the feature. For more information of the SIFT feature transformation algorithm, see [33], [34] and [27].

Thus, an image at time t_i is transformed to a collection of M vectors in feature space:

$$i(x, y, t_i) \longrightarrow \mathbf{z}_n^*(t_i) \quad \forall n \in \{1, 2, \dots, M\} \quad (4.11)$$

where, for the SIFT feature transformation algorithm, $\mathbf{z}_n^*(t_i)$ is partitioned into pose and object subspaces:

$$\mathbf{z}_n^*(t_i) = \begin{bmatrix} \mathbf{z}_n^{pose}(t_i)_{4 \times 1} \\ - \\ \mathbf{z}_n^{object}(t_i)_{128 \times 1} \end{bmatrix}_{132 \times 1} \quad (4.12)$$

where

$$\mathbf{z}_n^{pose}(t_i) = \begin{bmatrix} z_{n_x}(t_i) \\ z_{n_y}(t_i) \\ \sigma_n(t_i) \\ \theta_n(t_i) \end{bmatrix}_{4 \times 1} \quad (4.13)$$

where z_{n_x} and z_{n_y} are the pixel location in the image, σ_n is the scale of the feature, and θ_n is the primary orientation of the feature.

In the next section, the statistical predictive transformation of the feature vectors to a future time is presented.

4.2.2 Statistical Predictive Transformation of Feature Space. The theory is divided into three sections: estimating the initial landmark position and covariance based on the feature vectors calculated from the first image, using inertial measurements to propagate this augmented state to the time of the second image, and projecting this landmark position state onto the second image as a probability density function in feature space. In simpler terms, these equations allow the filter to predict where a stationary feature should appear in subsequent images, thus providing a statistical measure to constrain the search space within the image. This projection theory is novel and provides the framework for efficient and robust image and inertial integration.

4.2.2.1 Landmark Error Statistics. As described in Section 2.7, the image produced by an imaging sensor is a projection of the scene onto the imaging array. Thus each location on the imaging array corresponds to a three dimensional position in the scene. Each scene location corresponding to the pixel location of a feature is defined as a landmark. The landmark position corresponding to a pixel location is a nonlinear function of the navigation state, pixel location measurements from the feature vector, $\mathbf{z}(t_i)$, slant distance from the camera to the landmark, $d(t_i)$, camera to body direction cosine matrix, $\mathbf{C}_c^b$, and homogeneous camera projection matrix, $\mathbf{T}_c^{pix}$ (see Section 2.7.1 for a description):

$$\mathbf{y}^n = \mathbf{g}(\mathbf{p}^n(t_i), \mathbf{C}_b^n(t_i), \mathbf{z}(t_i), d(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}) \quad (4.14)$$

The pixel location measurement at time t_i is a nonlinear function of the navigation state, landmark position, and camera parameters (defined in Section 2.7):

$$\tilde{\mathbf{z}}(t_i) = \mathbf{h}(\mathbf{p}^n(t_i), \mathbf{C}_b^n(t_i), \mathbf{y}^n(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}) + \mathbf{v}(t_i) \quad (4.15)$$

where $\mathbf{v}(t_i)$ is a zero-mean, additive white Gaussian noise process with

$$E[\mathbf{v}(t_i)\mathbf{v}(t_j)] = \begin{cases} \mathbf{R}(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases} \quad (4.16)$$

The slant distance from the camera optical center to the terrain, $d(t_i)$, is not directly available from a single feature measurement and must be determined using additional information. The methods include exploiting a model of the terrain, estimating distance using binocular vision, and leveraging egomotion between multiple monocular images. The influence matrices must be carefully calculated in order to properly represent the contribution of the distance uncertainty, which may be highly dependent on the navigation state and measurement errors themselves. A detailed treatment of the potential cases and their effects on the distance uncertainty is presented in Section 4.3.3. The most general case, where $d(t_i)$ is a function of the navigation state, the pixel location, and some terrain model (h):

$$d(t_i) = \mathbf{d}(\mathbf{p}^n(t_i), \mathbf{C}_b^n(t_i), \mathbf{z}(t_i), h) \quad (4.17)$$

is assumed for the current development of the stochastic error projection equations. Simplifications of the general solution can then be used to describe situations where $d(t_i)$ is not a function of the navigation state (e.g., binocular range estimation).

Similarly to the navigation state, the calculated landmark position, $\tilde{\mathbf{y}}^n$, is also modeled as a perturbation about the true position:

$$\tilde{\mathbf{y}}^n = \mathbf{y}^n + \delta\mathbf{y}^n \quad (4.18)$$

and is a function of the calculated trajectory:

$$\tilde{\mathbf{y}}^n = \mathbf{g}\left(\tilde{\mathbf{p}}^n(t_i), \tilde{\mathbf{C}}_b^n(t_i), \tilde{\mathbf{z}}(t_i), \tilde{d}(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}\right) \quad (4.19)$$

Applying perturbation techniques to the landmark position function, the landmark error, $\delta \mathbf{y}^n$, can be expressed as a linear function of the errors of the navigation state ($\delta \mathbf{x}$), pixel measurement model ($\mathbf{v}(t_i)$), and terrain model uncertainty (δh):

$$\delta \mathbf{y}^n = \mathbf{G}_{yx} \delta \mathbf{x} + \mathbf{G}_{yh} \delta h + \mathbf{G}_{yz} \mathbf{v}(t_i) \quad (4.20)$$

where the influence coefficients are given by

$$\mathbf{G}_{yx} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\tilde{\mathbf{x}}, \tilde{\mathbf{z}}(t_i), \tilde{d}(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}} \quad (4.21)$$

$$\mathbf{G}_{yh} = \left. \frac{\partial \mathbf{g}}{\partial h} \right|_{\tilde{\mathbf{x}}, \tilde{\mathbf{z}}(t_i), \tilde{d}(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}} \quad (4.22)$$

$$\mathbf{G}_{yz} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{z}} \right|_{\tilde{\mathbf{x}}, \tilde{\mathbf{z}}(t_i), \tilde{d}(t_i), \mathbf{C}_c^b, \mathbf{T}_c^{pix}} \quad (4.23)$$

and

$$\delta d = \tilde{d} - d \quad (4.24)$$

Using the linearized position measurement, the landmark error is a zero-mean, Gaussian random vector. The landmark error covariance, $\mathbf{P}_{yy}(t_i)$, and cross-correlation matrix, $\mathbf{P}_{yx}(t_i)$ are defined as

$$\mathbf{P}_{yy}(t_i) = E[\delta \mathbf{y} \delta \mathbf{y}^T] \quad (4.25)$$

$$\mathbf{P}_{yx}(t_i) = E[\delta \mathbf{y} \delta \mathbf{x}^T] \quad (4.26)$$

Substituting (4.20) into (4.25), and noting the independence between navigation state and pixel measurement errors yields

$$\begin{aligned} \mathbf{P}_{yy}(t_i) &= \mathbf{G}_{yx} E[\delta \mathbf{x} \delta \mathbf{x}^T] \mathbf{G}_{yx}^T \\ &\quad + \mathbf{G}_{yh} E[\delta h^2] \mathbf{G}_{yh}^T \\ &\quad + \mathbf{G}_{yz} E[\mathbf{v}(t_i) \mathbf{v}^T(t_i)] \mathbf{G}_{yz}^T \end{aligned} \quad (4.27)$$

Substituting the previously defined covariance matrices for the navigation errors, terrain, and pixel measurement yields the final form of the landmark position error covariance:

$$\begin{aligned}\mathbf{P}_{yy}(t_i) &= \mathbf{G}_{yx}\mathbf{P}_{xx}(t_i)\mathbf{G}_{yx}^T \\ &\quad + \mathbf{G}_{yh}\sigma_h^2\mathbf{G}_{yh}^T + \mathbf{G}_{yz}\mathbf{R}\mathbf{G}_{yz}^T\end{aligned}\tag{4.28}$$

The cross-correlation matrices are calculated in a similar manner:

$$\mathbf{P}_{xy}(t_i) = \mathbf{P}_{xx}(t_i)\mathbf{G}_{yx}^T\tag{4.29}$$

$$\mathbf{P}_{yx}(t_i) = \mathbf{G}_{yx}\mathbf{P}_{xx}(t_i)\tag{4.30}$$

Note that $\mathbf{P}_{xx}(t_i)$ is the covariance matrix of the navigation state error, and is provided by the Kalman filter.

4.2.2.2 State Propagation. In this section, the nominal navigation state, navigation error state, and landmark error states are propagated from time t_i to t_{i+1} .

The nominal aircraft navigation state is propagated forward based on the nonlinear dynamics model given in Section 2.6, typically using a nonlinear differential equation solver (e.g., Euler, Trapezoidal, or Runge-Kutta) [50].

The landmark error dynamics are defined as a random walk:

$$\delta\dot{\mathbf{y}}^n = \mathbf{G}_y\mathbf{w}_y(t)\tag{4.31}$$

where $\mathbf{G}_y$ is the dynamics influence matrix and $\mathbf{w}_y(t)$ is a zero-mean, white Gaussian noise process with covariance kernel

$$E[\mathbf{w}_y(t)\mathbf{w}_y^T(t+\tau)] = \mathbf{Q}_y\delta(\tau)\tag{4.32}$$

As mentioned in Section 3.3.4, while the landmarks are assumed to be stationary (zero-dynamics), a small additive random walk is added to compensate for the effects of bias initialization errors and to prevent covariance eigenvalues from approaching zero [37].

The general navigation error stochastic differential equation is defined in Section 2.6.4 as

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}_x[\tilde{\mathbf{x}}^n(t), t]\mathbf{w}_x(t) \quad (4.33)$$

The navigation and landmark error covariance propagation dynamics are derived using the linearized dynamics models given in Section 2.6.4:

$$\begin{aligned} \dot{\mathbf{P}}_{xx}(t) &= \mathbf{F}(t)\mathbf{P}_{xx}(t) + \mathbf{P}_{xx}(t)\mathbf{F}^T(t) \\ &\quad + \mathbf{G}_x(t)\mathbf{Q}_x(t)\mathbf{G}_x^T(t) \end{aligned} \quad (4.34)$$

$$\dot{\mathbf{P}}_{xy}(t) = \mathbf{F}(t)\mathbf{P}_{xy}(t) \quad (4.35)$$

$$\dot{\mathbf{P}}_{yy}(t) = \mathbf{G}_y\mathbf{Q}_y\mathbf{G}_y^T \quad (4.36)$$

An equivalent expression for the time propagation is represented by the state transition matrix, $\Phi(t_{i+1}, t_i)$, which projects the navigation and landmark error covariance from time t_i to t_{i+1} [37]. The resulting expression for the navigation and landmark error covariance is:

$$\begin{aligned} \mathbf{P}_{xx}(t_{i+1}) &= \Phi(t_{i+1}, t_i)\mathbf{P}_{xx}(t_i)\Phi^T(t_{i+1}, t_i) \\ &\quad + \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, \tau)\mathbf{G}_x\mathbf{Q}_x\mathbf{G}_x^T\Phi^T(t_{i+1}, \tau)d\tau \end{aligned} \quad (4.37)$$

$$\mathbf{P}_{xy}(t_{i+1}) = \Phi(t_{i+1}, t_i)\mathbf{P}_{xy}(t_i) \quad (4.38)$$

$$\mathbf{P}_{yy}(t_{i+1}) = \mathbf{P}_{yy}(t_i) + (t_{i+1} - t_i)\mathbf{G}_y\mathbf{Q}_y\mathbf{G}_y^T \quad (4.39)$$

4.2.2.3 Projection of Uncertainty Statistics onto Image. The pixel projection function is used to project the navigation state and landmark location into

the feature space at time t_{i+1} . The pixel projection is

$$\mathbf{z}(t_{i+1}) = \mathbf{h}(\mathbf{p}^n(t_{i+1}), \mathbf{C}_b^n(t_{i+1}), \mathbf{y}^n(t_{i+1}), \mathbf{C}_c^b, \mathbf{T}_c^{pix}) \quad (4.40)$$

The estimated pixel location error, $\delta\mathbf{z}(t_{i+1})$, is modeled as a perturbation about the nominal pixel location:

$$\delta\mathbf{z}(t_{i+1}) = \hat{\mathbf{z}}(t_{i+1}) - \mathbf{z}(t_{i+1}) \quad (4.41)$$

where the estimated pixel location, $\hat{\mathbf{z}}(t_{i+1})$, is calculated using the nominal navigation state and landmark position:

$$\hat{\mathbf{z}}(t_{i+1}) = \mathbf{h}(\tilde{\mathbf{p}}^n(t_{i+1}), \tilde{\mathbf{C}}_b^n(t_{i+1}), \tilde{\mathbf{y}}^n(t_{i+1}), \mathbf{C}_c^b, \mathbf{T}_c^{pix}). \quad (4.42)$$

Perturbing the pixel projection function, the pixel location error can be expressed as a linear function of the errors of the navigation state and landmark position:

$$\delta\mathbf{z}(t_{i+1}) = \mathbf{H}_{zx}\delta\mathbf{x}(t_{i+1}) + \mathbf{H}_{zy}\delta\mathbf{y}(t_{i+1}) \quad (4.43)$$

where

$$\mathbf{H}_{zx} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \mathbf{C}_c^b, \mathbf{T}_c^{pix}} \quad (4.44)$$

$$\mathbf{H}_{zy} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{t}} \right|_{\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \mathbf{C}_c^b, \mathbf{T}_c^{pix}} \quad (4.45)$$

The pixel error covariance, $\mathbf{P}_{zz}(t_{i+1})$, is defined as

$$\mathbf{P}_{zz}(t_{i+1}) = E[\delta\mathbf{z}(t_{i+1})\delta\mathbf{z}^T(t_{i+1})] \quad (4.46)$$

Substituting (4.43) into (4.46), and eliminating independent error sources yields the pixel location covariance:

$$\begin{aligned}
\mathbf{P}_{zz}(t_{i+1}) = & \mathbf{H}_{zx}\mathbf{P}_{xx}(t_{i+1})\mathbf{H}_{zx}^T \\
& + \mathbf{H}_{zx}\mathbf{P}_{xy}(t_{i+1})\mathbf{H}_{zy}^T \\
& + \mathbf{H}_{zy}\mathbf{P}_{xy}^T(t_{i+1})\mathbf{H}_{zx}^T \\
& + \mathbf{H}_{zy}\mathbf{P}_{yy}(t_{i+1})\mathbf{H}_{zy}^T
\end{aligned} \tag{4.47}$$

Finally, the covariance of the pixel location errors can be summarized by combining the equations presented in the previous sections:

$$\begin{aligned}
\mathbf{P}_{zz}(t_{i+1}) = & \mathbf{H}_{zx}\mathbf{\Phi}(t_{i+1}, t_i)\mathbf{P}_{xx}(t_i)\mathbf{\Phi}^T(t_{i+1}, t_i)\mathbf{H}_{zx}^T \\
& + \mathbf{H}_{zx}\int_{t_i}^{t_{i+1}}\mathbf{\Phi}(t_{i+1}, \tau)\mathbf{G}_x\mathbf{Q}_x\mathbf{G}_x^T\mathbf{\Phi}^T(t_{i+1}, \tau)d\tau\mathbf{H}_{zx}^T \\
& + \mathbf{H}_{zx}\mathbf{\Phi}(t_{i+1}, t_i)\mathbf{P}_{xx}(t_i)\mathbf{G}_{yx}^T\mathbf{H}_{zy}^T \\
& + \mathbf{H}_{zy}\mathbf{G}_{yx}\mathbf{P}_{xx}(t_i)\mathbf{\Phi}^T(t_{i+1}, t_i)\mathbf{H}_{zx}^T \\
& + \mathbf{H}_{zy}\mathbf{G}_{yx}\mathbf{P}_{xx}(t_i)\mathbf{G}_{yx}^T\mathbf{H}_{zy}^T \\
& + \mathbf{H}_{zy}\mathbf{G}_{yh}\sigma_h^2\mathbf{G}_{yh}^T\mathbf{H}_{zy}^T \\
& + \mathbf{H}_{zy}\mathbf{G}_{yz}\mathbf{R}\mathbf{G}_{yz}^T\mathbf{H}_{zy}^T \\
& + (t_{i+1} - t_i)\mathbf{H}_{zy}\mathbf{G}_y\mathbf{Q}_y\mathbf{G}_y^T\mathbf{H}_{zy}^T
\end{aligned} \tag{4.48}$$

This equation shows how an initial navigation state error covariance, $\mathbf{P}_{xx}(t_i)$, height uncertainty variance, σ_h^2 , measurement noise (characterized by $\mathbf{R}$), and process noise (characterized by $\mathbf{Q}_x$ and $\mathbf{Q}_y$) can be rigorously projected to the feature space at a later time, t_{i+1} , as expressed by $\mathbf{P}_{zz}(t_{i+1})$.

In summary, given the pixel coordinates of a stationary ground landmark at time t_i , the predicted pixel coordinates of the same landmark at time t_{i+1} can be described by the bivariate Gaussian probability density function given in Equation (4.48). Thus, the correspondence search for the landmark can be constrained using a rigorous sta-

tistical confidence threshold (as opposed to approaches proposed in other research which are ad hoc). In the following section, the stochastic projection method is used to predict the location (and uncertainty) of a stationary landmark in an image.

4.2.3 Statistical Feature Matching. In the previous section, feature vectors from the image captured at time t_i are propagated stochastically to the next image capture time, t_{i+1} . After the image is captured at t_{i+1} , and the image is transformed into a collection of features, the next step is to determine the correspondence between the predicted feature set and the measured feature set at t_{i+1} . A statistically rigorous solution to this correspondence problem is presented in this section. This generalized solution, based on rigorous statistics of the navigation state, is an extension of the statistical projection theory presented in the previous section and is an additional contribution which results in a novel correspondence technique which is robust and efficient.

In general, determining the potential correspondence between a given feature vector predicted from an earlier image and the collection of feature vectors implies defining a metric which measures the quality of match. In this dissertation, the quality metric uses a statistical weighting of the difference between the predicted and candidate feature vectors, which is also known as the Mahalanobis distance. The Mahalanobis distance from the predicted feature vector, $\hat{\mathbf{z}}^*(t_{i+1})$, to the measured feature vector, $\mathbf{z}_n^*(t_{i+1})$, is defined as the weighted inner product:

$$D_n(t_{i+1}) = [\mathbf{z}_n^*(t_{i+1}) - \hat{\mathbf{z}}^*(t_{i+1})]^T \mathbf{P}_{z^*z^*}(t_{i+1}) [\mathbf{z}_n^*(t_{i+1}) - \hat{\mathbf{z}}^*(t_{i+1})] \quad (4.49)$$

where $\mathbf{P}_{z^*z^*}$ is the covariance of the predicted feature vector. As mentioned previously in Section 4.2.1, the feature vector covariance consists of strongly independent pose and object descriptor dimensions. Assuming the independence of scale and rotation

on the pixel location, the covariance matrix can be written in a block-diagonal form:

$$\mathbf{P}_{z^*z^*}(t_{i+1}) = \begin{bmatrix} \mathbf{P}_{zz}(t_{i+1}) & 0 & 0 & 0 \\ 0 & \mathbf{P}_{\sigma\sigma} & 0 & 0 \\ 0 & 0 & \mathbf{P}_{\theta\theta} & 0 \\ 0 & 0 & 0 & \mathbf{P}_{z_dz_d} \end{bmatrix} \quad (4.50)$$

where $\mathbf{P}_{zz}$ represented the pixel location uncertainty covariance, $\mathbf{P}_{\sigma\sigma}$ represents the scale uncertainty covariance, $\mathbf{P}_{\theta\theta}$ is the rotation uncertainty covariance, and $\mathbf{P}_{z_dz_d}$ is the uncertainty covariance of the descriptor. Thus, the associated covariances of the features serve to weight dimensions according to their statistical certainty. In the case of the SIFT algorithm and associated pose prediction using inertial measurements, there is no specific information provided by the algorithm regarding the statistical knowledge of $\mathbf{P}_{\sigma\sigma}$, $\mathbf{P}_{\theta\theta}$, or $\mathbf{P}_{z_dz_d}$. Reasonable assumptions regarding their values can be estimated by observing the statistics of the feature locator for a number of images. For the experiments presented in this dissertation, the scale and orientation parameters were given zero weight, (i.e., $\mathbf{P}_{\sigma\sigma}, \mathbf{P}_{\theta\theta} \rightarrow \infty \mathbf{I}$), and the distance metric was decomposed into a pose distance and an object descriptor distance:

$$D_{p_n}(t_{i+1}) = [\mathbf{z}_{p_n}(t_{i+1}) - \hat{\mathbf{z}}_p(t_{i+1})]^T \mathbf{P}_{z_pz_p}(t_{i+1}) [\mathbf{z}_{p_n}(t_{i+1}) - \hat{\mathbf{z}}_p(t_{i+1})] \quad (4.51)$$

$$D_{d_n}(t_{i+1}) = [\mathbf{z}_{d_n}(t_{i+1}) - \hat{\mathbf{z}}_d(t_{i+1})]^T \mathbf{P}_{z_dz_d}(t_{i+1}) [\mathbf{z}_{d_n}(t_{i+1}) - \hat{\mathbf{z}}_d(t_{i+1})] \quad (4.52)$$

where

$$\mathbf{P}_{z_pz_p}(t_{i+1}) = \begin{bmatrix} \mathbf{P}_{zz}(t_{i+1}) & 0 & 0 \\ 0 & \infty & 0 \\ 0 & 0 & \infty \end{bmatrix} \quad (4.53)$$

$$\mathbf{P}_{z_dz_d}(t_{i+1}) = \mathbf{I} \quad (4.54)$$

which has the result of:

- Including only the effects of the predicted pixel location in the pose distance metric;
- Weighting all components of the object descriptor vector equally.

In order to maintain independence of the pixel error measurement, and to improve search speed, the two distance metrics are used in succession. First, the pose metric is used to determine the subset of feature vectors existing within a statistical confidence bound (e.g., 2σ). Next, the object descriptor distance metric is calculated for the subset of features within the confidence bound. A successful match is declared if a feature is found with an object distance below a pre-defined threshold. In addition, to reduce the potential for false matches between objects with similar features within the search area, a uniqueness filter can also be applied which rejects potential feature matches which are not sufficiently distinct in feature space. This approach attempts to maintain a balance between using the statistical pose prediction for successful correspondence while not “coloring” the pixel error measurements, which are assumed to be independent of the filter states.

4.2.4 Navigation State Error Estimation. Once correspondence is determined between predicted and measured features, the pixel location of the feature in the current image is used to estimate and correct the errors in the navigation filter. In this section, the measurement equation is mathematically defined with respect to the augmented Kalman filter states.

Recalling Equation (4.15), the pixel location measurement is a function of the true navigation state, landmark location, camera parameters, and an additive noise term. Using the extended Kalman filter method presented in Section 2.8.2 requires linearization of the measurement equation about the current state estimate. The perturbation measurement equation for feature n at time t_{i+1} is given by

$$\delta \mathbf{z}_n(t_{i+1}) = \mathbf{H}_{zx} \delta \mathbf{x}(t_{i+1}) + \mathbf{H}_{zyn} \delta \mathbf{y}_n(t_{i+1}) + \mathbf{v}(t_{i+1}) \quad (4.55)$$

where the influence matrices, $\mathbf{H}_{zx}$ and $\mathbf{H}_{zy_n}$, are defined in Eqns. (4.44) and (4.45). The pixel measurement error, $\delta \mathbf{z}_n(t_{i+1})$, is defined as

$$\delta \mathbf{z}_n(t_{i+1}) = \hat{\mathbf{z}}_{p_n}(t_{i+1}) - \mathbf{z}_{p_n}(t_{i+1}) \quad (4.56)$$

The specific calculation of the influence matrices for this and other update types is presented in Section 4.3.3.

4.3 Image and Inertial Fusion Algorithm Extended Kalman Filter Implementation

In the previous section, the image-aided inertial navigation concept was presented in general mathematical terms. In this section, the mechanics involved in implementing the extended Kalman filter are discussed. Because of the complex nature of the filter, specific attention is given to selecting, adding, and deleting track states from the filter. In addition, the method for calculating the measurement influence matrices are presented for landmarks of opportunity, landmarks at known locations (reference landmarks), and celestial measurements.

4.3.1 Kalman Filter Description. An Extended Kalman Filter is constructed to estimate the errors in the calculated system parameters. In order to minimize the effects of linearization errors, the system parameters are periodically corrected by removing the current error estimate [37]. A block diagram of the system is shown in Figure 4.8.

As defined in Section 2.6.4, the navigation error state vector, $\delta \mathbf{x}$, consists of position, velocity, attitude, accelerometer bias, and gyroscope bias errors and is expressed

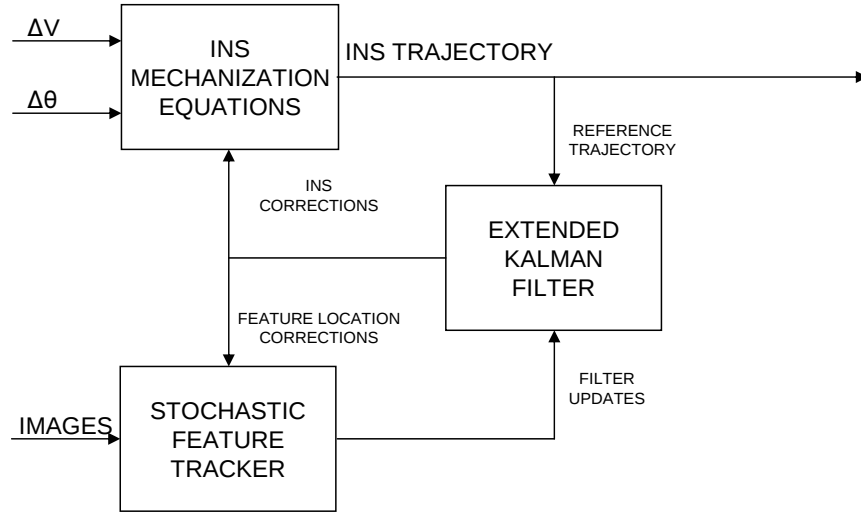


Figure 4.8: Image-aided inertial navigation filter block diagram. In this filter, the location of stationary objects are tracked and used to estimate and update the errors in an inertial navigation system. The inertial navigation system is, in turn, used to support the feature tracking loop.

as a vector of fifteen elements:

$$\delta \mathbf{x} = \begin{bmatrix} \delta \mathbf{p}^n \\ - - - \\ \delta \mathbf{v}^n \\ - - - \\ \boldsymbol{\psi} \\ - - - \\ \mathbf{a}^b \\ - - - \\ \mathbf{b}^b \end{bmatrix}_{15 \times 1} \quad (4.57)$$

Landmark tracks of interest are augmented to the above state vector using the procedure described in Section 4.3.2.2.

4.3.2 Feature Tracking Algorithm. In this section, the tracking algorithm is presented which describes the process of selecting, adding, and deleting states corresponding to specific landmarks from the filter.

4.3.2.1 Track Selection. As mentioned in the previous section, each landmark track requires three filter states to estimate the location of the landmark in the navigation frame. Practical limitations on computer resources effectively limit the number of landmarks that can be tracked concurrently. This limitation necessitates the implementation of a track management algorithm to decide which features to track.

The general concept for the track maintenance algorithm is to add and prune landmark tracks in order to provide the “best” navigation information to the filter. Although the optimal landmark choices are highly dependent on the trajectory and scene, some general “rules-of-thumb” were used in the track maintenance algorithm.

In general, features which can be easily and accurately tracked for long periods of time provide the best navigation information. This implies choosing features which are: strong and easily identified (to help maintain track), locally distinct (to eliminate false correspondence), and well-separated in image space (to maximize filter observability). Thus, when Kalman Filter landmark track states are available, the feature space of the current image is searched and new landmarks are added based on the above criteria. The filter states are augmented in accordance with the stochastic projection algorithm defined in Section 4.2.2.

In order to maintain only the best tracks, stale landmark tracks (i.e., no successful correspondence available for a given period of time) are pruned by replacing the associated filter states with new candidate tracks. Other track maintenance approaches are possible which could theoretically improve the track performance (e.g., semi-causal, multiple model, or velocity prediction); however, these approaches will not be pursued in this document.

4.3.2.2 Track Addition. Once a new track is identified, the navigation filter must maintain an estimate of the landmark location and uncertainty. This is accomplished by replacing the current filter states corresponding to an inactive track. For simplicity of implementation, the filter maintains a static number of landmark track states. When no landmark is currently being tracked in a particular state, there are, by definition, no available measurements which affect those states. Because the landmark track states are not dynamically coupled with the navigation states, inactive states have no effect on the navigation solution, which allows a simple state replacement strategy versus a more complicated dynamical approach. An example illustrating the concept is presented below.

In this example, the states corresponding to the n -th landmark are updated with a new track. In order to rigorously augment the filter states, the landmark location uncertainty covariance and cross-correlation matrices (calculated in Section 4.2.2) are required. For this illustration, the new uncertainty covariance and cross-correlation matrices are given by $\mathbf{P}_{xy}^+$ and $\mathbf{P}_{yy}^+$, respectively. As shown in Figure 4.9, the n -th landmark error states are replaced with the $\delta\mathbf{y}_n^+$. Because the landmark position errors are defined about the mean, the initial error estimate is the $\mathbf{0}_{3 \times 1}$ vector. Additionally, the existing covariance matrices are replaced with either the newly calculated matrices, or zeros as indicated by the figure.

4.3.2.3 Track Deletion. The feature tracking algorithm maintains a list of the latest successful feature update for each track. Tracks which have not been successfully updated in a given time or number of image frames are identified for replacement using the method described in the previous section.

4.3.3 Measurement Models. In this section, the influence matrices relating to various measurement types are derived and compared.

4.3.3.1 Landmark of Opportunity. The first measurement of interest is the landmark of opportunity. As described in Section 4.2.2.1, incorporating land-

marks of opportunity into the navigation filter requires two steps: 1) determining the initial landmark location and associated influence matrices, and 2) calculating the measurement errors and associated influence matrices.

The method to determine the initial landmark location and associated influence matrices varies primarily on the method used to estimate the distance to the landmark. Three methods are addressed in this section: monocular imaging with statistical terrain model, binocular stereopsis with no terrain model, and time-iterated.

4.3.3.2 Landmark Location Estimation Using Statistical Terrain Model.

In this section, the landmark initial location estimate, $\mathbf{y}^n$, is calculated given the undistorted pixel location of a feature of interest, $\mathbf{z}$, and a statistical estimate of the terrain height above the navigation reference plane, h_t . An illustration of the measurement geometry is shown in Figure 4.10.

The homogeneous pointing vector in the camera frame is

$$\underline{\mathbf{s}}^c = \mathbf{T}_{pix}^c \mathbf{z} \quad (4.58)$$

The resulting landmark location is given by

$$\mathbf{y}^n = \mathbf{p}^n + \mathbf{C}_b^n [\mathbf{p}_{cam}^b + d\mathbf{C}_c^b \underline{\mathbf{s}}^c] \quad (4.59)$$

where d is the distance to the landmark, projected in the camera z direction. This distance is a function of the navigation state and the mean of the terrain height and is derived by expanding and rearranging Equation (4.59):

$$d\mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c = \mathbf{y}^n - [\mathbf{p}^n + \mathbf{C}_b^n \mathbf{p}_{cam}^b] \quad (4.60)$$

Extracting the z component and dividing by the resulting scalar yields the landmark distance:

$$d = \frac{[\mathbf{y}^n]_z - [\mathbf{p}^n + \mathbf{C}_b^n \mathbf{p}_{cam}^b]_z}{[\mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c]_z} \quad (4.61)$$

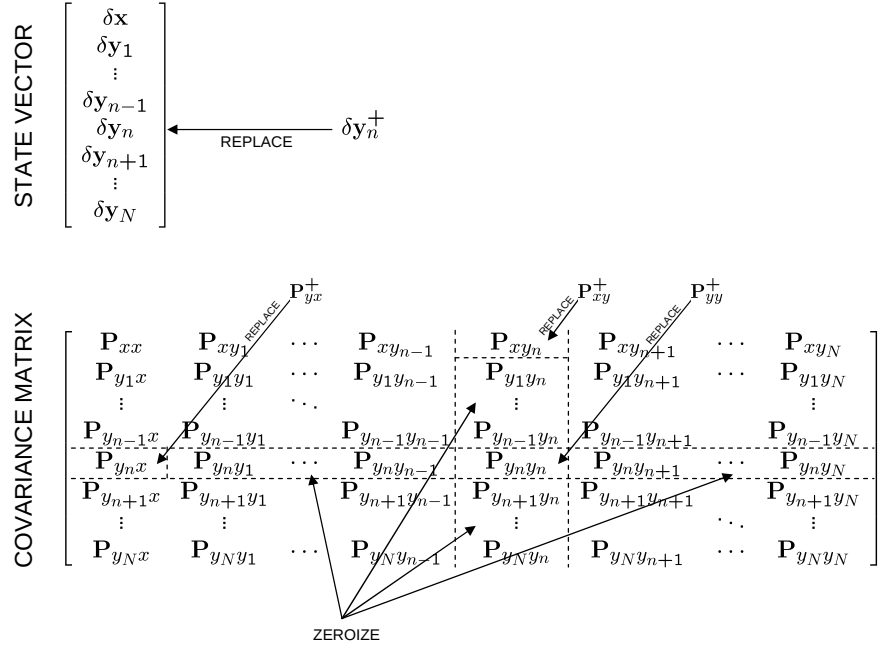


Figure 4.9: Landmark track state replacement. When replacing a stale landmark track with a new landmark track, both the covariance and cross-correlation matrices must be updated to reflect the new geometry.

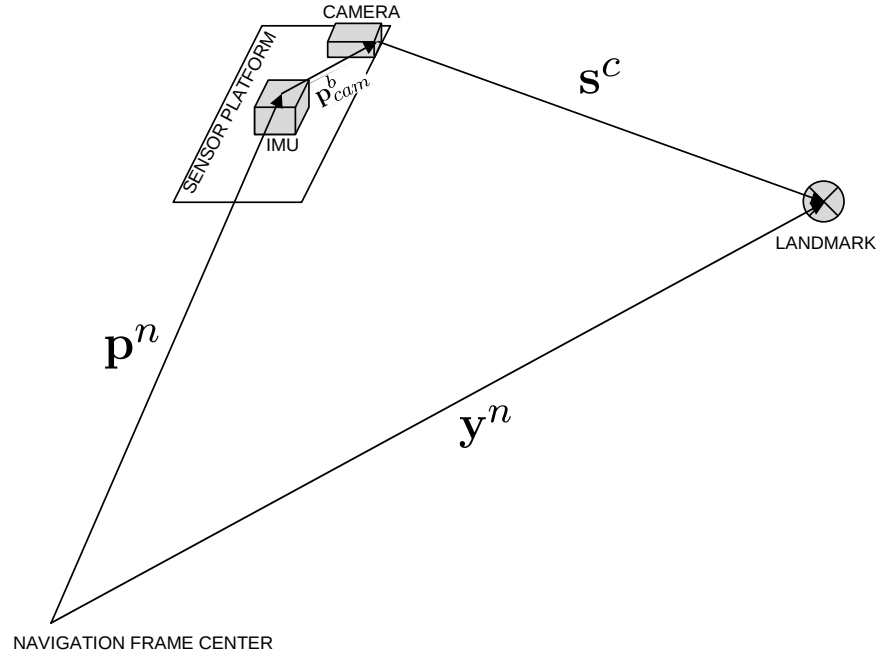


Figure 4.10: Monocular imaging geometry. The line of sight vector from the camera to the landmark, $\mathbf{s}^c$, is a function of the landmark location, $\mathbf{y}^n$, sensor platform location, $\mathbf{p}^n$, and the lever arm from the inertial sensor to the camera, $\mathbf{p}_{cam}^b$.

where the $[\cdot]_z$ transformation is the linear functional producing the z vector component, which is defined as

$$[\mathbf{p}]_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{p} = p_z \quad (4.62)$$

Note the operation reduces vectors to scalars and matrices to vectors.

By definition, the z component of the landmark location is given by the terrain height, h_t :

$$[\mathbf{y}^n]_z = h_t \quad (4.63)$$

Substituting Eqn. (4.63) into (4.61) yields

$$d = \frac{h_t - [\mathbf{p}^n + \mathbf{C}_b^n \mathbf{p}_{cam}^b]_z}{[\mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c]_z} \quad (4.64)$$

Once the location of the landmark is calculated, the influence matrices from the various uncertainties can be calculated. For this example, the following error contributors are considered:

- Position uncertainty of the platform, $\delta \mathbf{p}^n$
- Velocity uncertainty of the platform, $\delta \mathbf{v}^n$
- Attitude uncertainty of the platform, ψ
- Attitude uncertainty of the camera, α
- Pixel measurement uncertainty, $\delta \mathbf{z}$
- Terrain elevation uncertainty, δh_t

The influence matrix corresponding to the position uncertainty is defined as

$$\mathbf{G}_{yp} = \frac{\partial \mathbf{y}^n}{\partial \mathbf{p}^n} \quad (4.65)$$

Evaluating the partial derivative yields

$$\mathbf{G}_{yp} = \mathbf{I}_{3 \times 3} + \mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c \frac{\partial d}{\partial \mathbf{p}^n} \quad (4.66)$$

where

$$\frac{\partial d}{\partial \mathbf{p}^n} = -\frac{1}{[\mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b]_z} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad (4.67)$$

The influence matrix corresponding to the platform attitude uncertainty is defined as

$$\mathbf{G}_{y\psi} = \frac{\partial \mathbf{y}^n}{\partial \boldsymbol{\psi}} \quad (4.68)$$

Evaluating the partial derivative using the method described in Section 2.10 yields

$$\mathbf{G}_{y\psi} = \mathbf{C}_b^n \left([\mathbf{p}_{cam}^b + d\mathbf{C}_{c\bar{\mathbf{s}}}^b] \times \right) + \mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b \frac{\partial d}{\partial \boldsymbol{\psi}} \quad (4.69)$$

where

$$\frac{\partial d}{\partial \boldsymbol{\psi}} = d \frac{[(\mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b) \times]_z}{[\mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b]_z} \quad (4.70)$$

The influence matrix corresponding to the camera attitude uncertainty is defined as

$$\mathbf{G}_{y\alpha} = \frac{\partial \mathbf{y}^n}{\partial \boldsymbol{\alpha}} \quad (4.71)$$

Evaluating the partial derivative yields

$$\mathbf{G}_{y\alpha} = -d\mathbf{C}_b^n \mathbf{C}_c^b [\bar{\mathbf{s}}^c \times] + \mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b \frac{\partial d}{\partial \boldsymbol{\alpha}} \quad (4.72)$$

where

$$\frac{\partial d}{\partial \boldsymbol{\alpha}} = -d \frac{[\mathbf{C}_b^n \mathbf{C}_c^b (\bar{\mathbf{s}}^c \times)]_z}{[\mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b]_z} \quad (4.73)$$

The influence matrix corresponding to the pixel location measurement uncertainty is defined as

$$\mathbf{G}_{yz} = \frac{\partial \mathbf{y}^n}{\partial \mathbf{z}} \quad (4.74)$$

Evaluating the partial derivative yields

$$\mathbf{G}_{yz} = d\mathbf{C}_b^n \mathbf{C}_c^b \mathbf{T}_{pix}^c \begin{bmatrix} \mathbf{I}_{2 \times 2} \\ 0 & 0 \end{bmatrix} + \mathbf{C}_b^n \mathbf{C}_{c\bar{\mathbf{s}}}^b \frac{\partial d}{\partial \mathbf{z}} \quad (4.75)$$

where

$$\frac{\partial d}{\partial \mathbf{z}} = d \frac{\left[\mathbf{C}_b^n \mathbf{C}_c^b \mathbf{T}_{pix}^c \begin{bmatrix} \mathbf{I}_{2 \times 2} \\ 0 \quad 0 \end{bmatrix} \right]_z}{[\mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c]_z} \quad (4.76)$$

Finally, the influence matrix corresponding to the terrain elevation uncertainty is defined as

$$\mathbf{G}_{yh} = \frac{\partial \mathbf{y}^n}{\partial h_t} \quad (4.77)$$

Evaluating the partial derivative yields

$$\mathbf{G}_{yh} = \frac{1}{[\mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c]_z} \mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c \quad (4.78)$$

The remaining influence matrices are all zeros.

Once a landmark state has been added to the filter, additional measurements of the pixel location of the target are processed using the measurement equation:

$$\mathbf{z} = \mathbf{T}_c^{pix} \underline{\mathbf{s}}^c \quad (4.79)$$

where $\underline{\mathbf{s}}^c$ is the homogeneous form of the line-of-sight vector, $\mathbf{s}^c$, defined as

$$\mathbf{s}^c = \mathbf{C}_b^c [\mathbf{C}_n^b (\mathbf{y}^n - \mathbf{p}^n) - \mathbf{p}_{cam}^b] \quad (4.80)$$

The calculation of the measurement influence matrices proceeds in a similar fashion. The influence matrix corresponding to the position uncertainty is defined as

$$\mathbf{H}_{zp} = \frac{\partial \mathbf{z}}{\partial \mathbf{p}^n}. \quad (4.81)$$

Evaluating the partial derivative yields

$$\mathbf{H}_{zp} = \mathbf{T}_c^{pix} \frac{s_z^c \frac{\partial \mathbf{s}^c}{\partial \mathbf{p}^n} - \mathbf{s}^c \left[\frac{\partial \mathbf{s}^c}{\partial \mathbf{p}^n} \right]_z}{(s_z^c)^2} \quad (4.82)$$

where

$$\frac{\partial \mathbf{s}^c}{\partial \mathbf{p}^n} = -\mathbf{C}_b^c \mathbf{C}_n^b \quad (4.83)$$

The influence matrix corresponding to the camera misalignment is defined as

$$\mathbf{H}_{z\alpha} = \frac{\partial \mathbf{z}}{\partial \alpha} \quad (4.84)$$

Evaluating the partial derivative yields

$$\mathbf{H}_{z\alpha} = \mathbf{T}_c^{pix} \frac{s_z^c \frac{\partial \mathbf{s}^c}{\partial \alpha} - \mathbf{s}^c \left[\frac{\partial \mathbf{s}^c}{\partial \alpha} \right]_z}{(s_z^c)^2} \quad (4.85)$$

where

$$\frac{\partial \mathbf{s}^c}{\partial \alpha} = (\mathbf{s}^c \times) \quad (4.86)$$

The influence matrix corresponding to the landmark location uncertainty is defined as

$$\mathbf{H}_{zy} = \frac{\partial \mathbf{z}}{\partial \mathbf{y}^n} \quad (4.87)$$

Evaluating the partial derivative yields

$$\mathbf{H}_{zy} = \mathbf{T}_c^{pix} \frac{s_z^c \frac{\partial \mathbf{s}^c}{\partial \mathbf{y}^n} - \mathbf{s}^c \left[\frac{\partial \mathbf{s}^c}{\partial \mathbf{y}^n} \right]_z}{(s_z^c)^2} \quad (4.88)$$

where

$$\frac{\partial \mathbf{s}^c}{\partial \mathbf{y}^n} = \mathbf{C}_b^c \mathbf{C}_n^b \quad (4.89)$$

All other influence matrices are zero.

4.3.3.3 Landmark Location Estimation Using Binocular Stereopsis.

Another method for determining the distance of an object within an image is to utilize binocular measurements. Binocular measurements do not require a terrain model, which makes this method appropriate for navigation in unknown environments, or when the sensor is on or near the ground. In order to properly interpret binocular

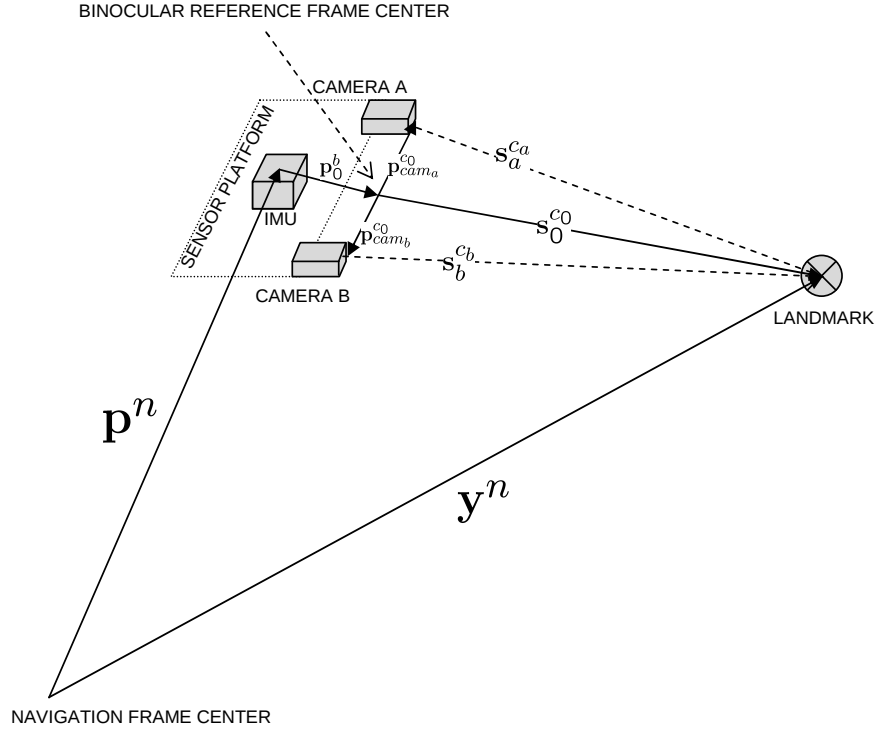


Figure 4.11: Binocular imaging geometry. The line of sight vector from the binocular reference point to the landmark, $\mathbf{s}_0^{c_0}$, is a function of the landmark location, $\mathbf{y}^n$, sensor platform location, $\mathbf{p}^n$, the lever arm from the inertial sensor to the binocular reference point, $\mathbf{p}_0^b$, and the respective lever arms from the binocular reference point to cameras a and b , $\mathbf{p}_{cam_a}^{c_0}$ and $\mathbf{p}_{cam_b}^{c_0}$, respectively.

measurements, it is convenient to establish a binocular disparity reference frame, which is located between the optical centers of two cameras. The resulting binocular imaging geometry is shown in Figure 4.11.

Solving for a landmark location using binocular measurements requires three steps: 1) feature selection and matching between binocular image pairs, 2) calculation of the relative line of sight vector from the binocular origin to the landmark, and 3) estimating the landmark location and augmenting the Kalman filter.

The feature selection and matching process begins by selecting a candidate feature from one of the images in the binocular pair, using the same location quality metric presented previously for the monocular imaging calculation. This establishes a candidate binocular feature. Once a base feature is chosen, the feature location of the

binocular match is statistically projected into the feature space of the corresponding image. For example, given the location of a feature from image a , $\mathbf{z}_a(t_i)$, a statistical binocular projection is desired to predict the corresponding binocular match mean and uncertainty in the feature space of image b . The pixel location in the b frame can be described by

$$\mathbf{z}^b = \mathbf{T}_{c_b}^{pix} \underline{\mathbf{s}}^{c_b} \quad (4.90)$$

and

$$\underline{\mathbf{s}}^{c_b} = \mathbf{C}_{c_0}^{c_b} [k \mathbf{C}_{c_a}^{c_0} \mathbf{T}_{pix}^{c_a} \mathbf{z}^a + \mathbf{p}_{cam_a}^{c_0} - \mathbf{p}_{cam_b}^{c_0}] \quad (4.91)$$

where $\mathbf{T}_{pix}^{c_a}$ and $\mathbf{T}_{c_b}^{pix}$ are the camera projection matrices for camera a and b , respectively, $\mathbf{C}_{c_a}^{c_0}$ and $\mathbf{C}_{c_b}^{c_0}$ are the orientation direction cosine matrices for camera a and b , respectively. In addition, $\mathbf{p}_{cam_a}^{c_0}$ and $\mathbf{p}_{cam_b}^{c_0}$ are the location vectors from the c_0 frame for camera a and b , respectively (see Figure 4.11). Finally, k is the (unknown) distance parameter.

Variations in k describe a portion of the epipolar line in the secondary image. Depending on the system conditions, some or all of the parameters can be treated as random variables. In this example, the relative camera location and orientations, distance, and pixel location errors are modeled as random variables with Gaussian distributions. While the distribution of the distance parameter, k , is most accurately modeled as a uniform random variable, in order to simplify the calculation of the search boundary, a Gaussian distribution is desired which is representative of the uniform random variable when projected into the corresponding image. An acceptable Gaussian distribution for k which results in accurate predictions of the binocular pixel search space is empirically determined as a function of the binocular disparity. The binocular disparity, d_{binoc} , is defined as

$$d_{binoc} = \|\mathbf{p}_{cam_a}^{c_0} - \mathbf{p}_{cam_b}^{c_0}\| \quad (4.92)$$

The distance parameter, k , is thus modeled as a Gaussian distribution with mean and variance given by

$$\bar{k} = \mathbf{E}[k] = 4.5d_{binoc} \quad (4.93)$$

$$\sigma_k^2 = \mathbf{E}[(k - \bar{k})^2] = 9d_{binoc}^2 \quad (4.94)$$

which is determined empirically to result in a reasonable approximation of the binocular search space when projected along the epipolar line (see Figure 12(b)).

The predicted pixel location mean is calculated using the mean values of the parameters (see Equations (4.90) and (4.91)):

$$\hat{\mathbf{z}}^b = \mathbf{f} [\bar{\mathbf{z}}^a, \mathbf{T}_{pix}^{c_a}, \bar{\mathbf{C}}_{c_a}^{c_0}, \bar{k}, \bar{\mathbf{p}}_{cam_a}^{c_0}, \bar{\mathbf{p}}_{cam_b}^{c_0}, \bar{\mathbf{C}}_{c_0}^{c_b}, \mathbf{T}_{c_b}^{pix}] \quad (4.95)$$

Assuming independent error sources, the pixel location error covariance matrix is given by

$$\begin{aligned} \mathbf{P}_{z_b z_b} = & \mathbf{H}_{z_b z_a} \mathbf{R}_{z_a z_a} \mathbf{H}_{z_b z_a}^T + \mathbf{H}_{z_b p_a} \mathbf{P}_{p_a p_a} \mathbf{H}_{z_b p_a}^T + \mathbf{H}_{z_b \alpha_a} \mathbf{P}_{\alpha_a \alpha_a} \mathbf{H}_{z_b \alpha_a}^T + \mathbf{H}_{z_b p_b} \mathbf{P}_{p_b p_b} \mathbf{H}_{z_b p_b}^T \\ & + \mathbf{H}_{z_b \alpha_b} \mathbf{P}_{\alpha_b \alpha_b} \mathbf{H}_{z_b \alpha_b}^T + \mathbf{H}_{z_b k} \sigma_k^2 \mathbf{H}_{z_b k}^T + \mathbf{R}_{z_b z_b} \end{aligned} \quad (4.96)$$

where $\mathbf{R}_{z_a z_a}$ and $\mathbf{R}_{z_b z_b}$ are the (independent) pixel location error covariance matrices for images a and b , respectively. The binocular lever arm location and orientation uncertainty covariances are given by $\mathbf{P}_{p_a p_a}$ and $\mathbf{P}_{\alpha_a \alpha_a}$ for camera a and $\mathbf{P}_{p_b p_b}$ and $\mathbf{P}_{\alpha_b \alpha_b}$ for camera b . The influence matrices are defined as the partial derivatives calculated

about the estimated states ($\tilde{\mathbf{x}}$):

$$\mathbf{H}_{z_b z_a} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{z}_{z_a}} \right|_{\tilde{\mathbf{x}}} \quad (4.97)$$

$$\mathbf{H}_{z_b p_a} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{p}_{cam_a}^{c_0}} \right|_{\tilde{\mathbf{x}}} \quad (4.98)$$

$$\mathbf{H}_{z_b \alpha_a} = \left. \frac{\partial \mathbf{h}}{\partial \alpha_a} \right|_{\tilde{\mathbf{x}}} \quad (4.99)$$

$$\mathbf{H}_{z_b p_b} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{p}_{cam_b}^{c_0}} \right|_{\tilde{\mathbf{x}}} \quad (4.100)$$

$$\mathbf{H}_{z_b \alpha_b} = \left. \frac{\partial \mathbf{h}}{\partial \alpha_b} \right|_{\tilde{\mathbf{x}}} \quad (4.101)$$

$$\mathbf{H}_{z_b k} = \left. \frac{\partial \mathbf{h}}{\partial k} \right|_{\tilde{\mathbf{x}}} \quad (4.102)$$

Once the feature location has been projected into the corresponding binocular feature space, the matching process consists of determining the features which are within a given statistical distance of the prediction, then choosing the feature which minimizes the comparative feature description distance. This is equivalent to the procedure discussed in Section 4.2.3. An example of the binocular matching process is shown in Figures 12(a) and 12(b).

Once a binocular pair of pixel locations corresponding to a landmark have been determined, the next step is to estimate the relative line of sight vector from the origin of the binocular frame to the landmark. From Figure 4.11, the respective pixel locations can be expressed as a function of the relative line of sight vector:

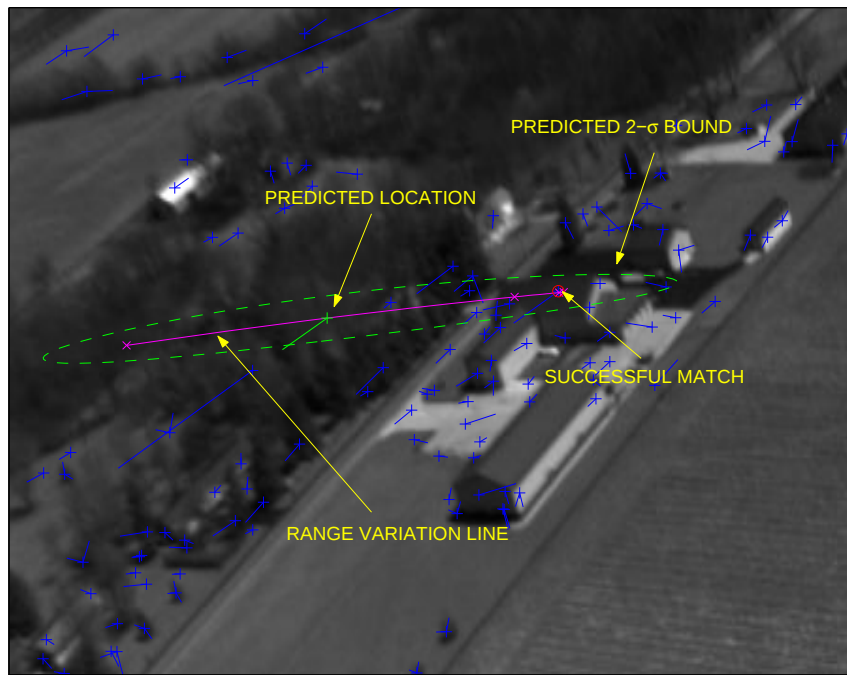
$$\mathbf{z}^a = \mathbf{h}_a[\mathbf{s}_0^{c_0}, \mathbf{C}_{c_0}^{c_a}, \mathbf{p}_{cam_a}^{c_0}, \mathbf{T}_{c_a}^{pix}] \quad (4.103)$$

$$\mathbf{z}^b = \mathbf{h}_b[\mathbf{s}_0^{c_0}, \mathbf{C}_{c_0}^{c_b}, \mathbf{p}_{cam_b}^{c_0}, \mathbf{T}_{c_b}^{pix}] \quad (4.104)$$

where $\mathbf{s}_0^{c_0}$ is the relative line of sight vector, which can be estimated using nonlinear regression techniques such as iterative least squares (see Section 2.9).



(a) Left image.



(b) Right image.

Figure 4.12: Binocular feature prediction. The location of a selected feature (blue + with red circle) is predicted from the left image to the right image using a statistical model of the relationship between the two cameras. Note the 2σ bound agrees well with the range variation line. The range variation line indicates where the predicted landmark will appear at a distance from 1 meter to 1000 meters (from left to right).

Expanding equations (4.103) and (4.104) yields

$$\mathbf{z}^a = \mathbf{T}_{c_a}^{pix} \underline{\mathbf{s}}_a^{c_a} \quad (4.105)$$

$$\mathbf{z}^b = \mathbf{T}_{c_b}^{pix} \underline{\mathbf{s}}_b^{c_b} \quad (4.106)$$

where

$$\mathbf{s}_a^{c_a} = \mathbf{C}_{c_0}^{c_a} [\mathbf{s}_0^{c_0} - \mathbf{p}_{cam_a}^{c_0}] \quad (4.107)$$

$$\mathbf{s}_b^{c_b} = \mathbf{C}_{c_0}^{c_b} [\mathbf{s}_0^{c_0} - \mathbf{p}_{cam_b}^{c_0}] \quad (4.108)$$

The estimation procedure results in an estimated line of sight vector, $\hat{\mathbf{s}}_0^{c_0}$, and an associated covariance matrix, $\mathbf{P}_{s_0 s_0}$. The landmark location in the navigation frame can now be estimated by substituting the line of sight vector estimate and the navigation state estimate into the landmark location equation:

$$\mathbf{y}^n = \mathbf{p}^n + \mathbf{C}_b^n [\mathbf{p}_0^b + \mathbf{C}_{c_0}^b \mathbf{s}_0^{c_0}] \quad (4.109)$$

Once the landmark location and the associated covariance and cross-correlation matrices have been calculated, the Kalman filter states can be augmented using the procedure described in Section 4.3.2.2.

This procedure has assumed the binocular regression algorithm converges to the correct minimum. When the landmark is a large distance away, relative to the binocular disparity, pixel measurement noise can create a situation where the two line of sight vectors are skewed outward and result in negative (i.e., impossible) distance estimates. This condition can be easily detected prior to the regression and another binocular feature can be selected.

4.3.3.4 Landmark Location Estimation Using Egomotion and Monocular Measurements. In the absence of terrain information or binocular images, inertial measurements can be exploited to estimate the distance to landmarks using egomo-

tion. In this scheme, the pixel location of a selected landmark is measured over two or more images until the distance becomes observable. Once the distance is observable, the feature is tracked (and used for navigation) by augmenting the Kalman filter state vector in an identical fashion to that described in Section 4.3.2.2. The concept of utilizing a series of monocular measurements with unknown terrain information is not new (see Strelow [62]), however, this research improves upon Strelow's method by leveraging inertial measurements to statistically constrain the correspondence search.

The proposed procedure consists of two steps: 1) given a feature location in an image at time t_i , predicting the feature location in the image at time t_{i+1} , and 2) estimating the landmark location from a series of pixel measurements. The statistical feature prediction method is presented first.

Predicting the location of a feature from one image to another using inertial measurements is similar to the binocular prediction technique presented in Section 4.3.3.3, except that the binocular disparity is estimated using the change in navigation state. The feature selection and matching process begins by selecting a candidate feature from the image at time t_i , using the same location quality metric presented previously for the monocular imaging calculation. Once a candidate feature is chosen, the feature location is statistically projected into the feature space at time t_{i+1} . The monocular egomotion imaging geometry is shown in Figure 4.13. For example, given the location of a feature from the image at time t_i , $\mathbf{z}(t_i)$, a statistical projection is desired to predict the corresponding feature location in the feature space at time t_{i+1} . The pixel location at time t_{i+1} can be described by

$$\mathbf{z}(t_{i+1}) = \mathbf{T}_c^{pix} \underline{\mathbf{s}}^c(t_{i+1}) \quad (4.110)$$

and

$$\mathbf{s}^c(t_{i+1}) = \mathbf{C}_b^c \{ \mathbf{C}_b^{n^T}(t_{i+1}) [\mathbf{C}_b^n(t_i) (k \mathbf{C}_s^b \underline{\mathbf{s}}^c(t_i) + \mathbf{p}_{cam}^b) + \mathbf{p}^n(t_i) - \mathbf{p}^n(t_{i+1})] - \mathbf{p}_{cam}^b \} \quad (4.111)$$

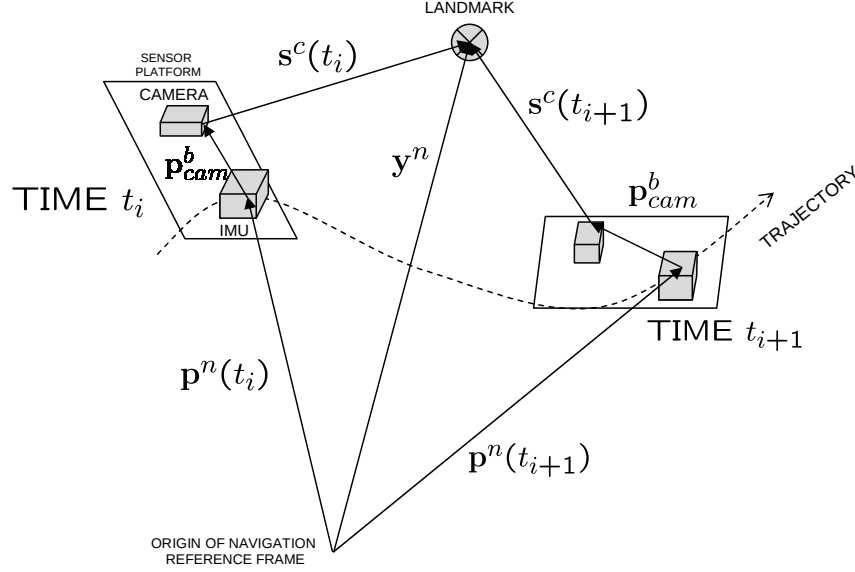


Figure 4.13: Monocular egomotion geometry. The line of sight vector from the camera location at time t_{i+1} is a function of the landmark location and the camera egomotion from time t_i to t_{i+1} .

where $\mathbf{p}^n(t_i)$ and $\mathbf{p}^n(t_{i+1})$ are the positions of the vehicle at times t_i and t_{i+1} , respectively. In addition, the direction cosine matrices $\mathbf{C}_b^n(t_i)$ and $\mathbf{C}_b^n(t_{i+1})$ are the body to navigation frame DCMs at times t_i and t_{i+1} . The unknown distance parameter is given by k . Finally, the homogeneous landmark location vector at time t_i is a function of the pixel measurement, $\mathbf{z}(t_i)$:

$$\underline{\mathbf{s}}^c(t_i) = \mathbf{T}_{pix}^c \mathbf{z}(t_i) \quad (4.112)$$

Variations in k describe a portion of the epipolar line in the second image. While k is most accurately described as a uniform random variable, in order to simplify the calculation of a search boundary, a representative Gaussian distribution is desired. From Equation (4.111), the distance between the camera location at time t_i and t_{i+1} is defined as

$$d = \|\mathbf{C}_b^c \left[\mathbf{C}_b^{n^T}(t_{i+1}) (\mathbf{C}_b^n(t_i) \mathbf{p}_{cam}^b + \mathbf{p}^n(t_i) - \mathbf{p}^n(t_{i+1})) - \mathbf{p}_{cam}^b \right] \|\quad (4.113)$$

In a similar fashion to that presented in Section 4.3.3.3, the distance parameter, k , is modeled as a Gaussian distribution with mean and variance given by

$$\bar{k} = \mathbf{E}[k] = 4.5d \quad (4.114)$$

$$\sigma_k^2 = \mathbf{E}[(k - \bar{k})^2] = 9d^2 \quad (4.115)$$

The predicted pixel location mean and covariance can now be calculated. Generalizing Equations (4.110)-(4.112) results in the nonlinear measurement equation:

$$\mathbf{z}(t_{i+1}) = \mathbf{h}[k, \mathbf{p}^n(t_i), \mathbf{p}^n(t_{i+1}), \mathbf{C}_b^n(t_i), \mathbf{C}_b^n(t_{i+1}), \mathbf{z}(t_i)] + \mathbf{v}(t_{i+1}) \quad (4.116)$$

where $\mathbf{v}(t_{i+1})$ is additive, white, Gaussian noise. The pixel location error at time t_{i+1} is given by

$$\delta \mathbf{z}(t_{i+1}) = \mathbf{H}_{zx_i} \delta \mathbf{x}(t_i) + \mathbf{H}_{zx_{i+1}} \delta \mathbf{x}(t_{i+1}) + \mathbf{H}_{zk} \delta k + \mathbf{H}_{zz_i} \mathbf{v}(t_i) + \mathbf{H}_{zz_{i+1}} \mathbf{v}(t_{i+1}) \quad (4.117)$$

where $\delta \mathbf{x}(t_i)$ is the navigation state error at time t_i , $\delta \mathbf{x}(t_{i+1})$ is the navigation state error at time t_{i+1} , $\mathbf{v}(t_i)$ is the measurement noise at time t_i , and $\mathbf{v}(t_{i+1})$ is the measurement noise at time t_{i+1} . The influence matrices are given by $\mathbf{H}_z$ and are the respective partial derivatives of the measurement function.

Calculation of the pixel location error covariance requires knowledge of the state transition matrix from time t_i to time t_{i+1} , $\Phi_{t_i}^{t_{i+1}}$, and the additive process noise matrix, $\mathbf{Q}_d$, which are available from the Kalman filter propagation equations (see Section 2.8). The navigation error covariance can now be expressed as a function of the state transition matrix and process noise:

$$\mathbf{P}_{xx}(t_{i+1}) = \Phi_{t_i}^{t_{i+1}} \mathbf{P}_{xx}(t_i) \Phi_{t_{i+1}}^{t_i} + \mathbf{Q}_d \quad (4.118)$$

The pixel location uncertainty at time t_{i+1} is defined as

$$\mathbf{P}_{zz}(t_{i+1}) = \mathbf{E} [\delta \mathbf{z}(t_{i+1}) \delta \mathbf{z}(t_{i+1})^T] \quad (4.119)$$

Substituting Equations (4.117)-(4.118) into (4.119), and noting the independence of the pixel measurement noise and initial distance estimate yields the pixel location uncertainty:

$$\begin{aligned} \mathbf{P}_{zz}(t_{i+1}) = & \mathbf{H}_{zx_i} \mathbf{P}_{xx}(t_i) \mathbf{H}_{zx_i}^T + \mathbf{H}_{zx_{i+1}} \Phi_{t_i}^{t_{i+1}} \mathbf{P}_{xx}(t_i) \mathbf{H}_{zx_i}^T + \mathbf{H}_{zx_i} \mathbf{P}_{xx}(t_i) \Phi_{t_{i+1}}^{t_i} \mathbf{H}_{zx_{i+1}}^T \\ & + \mathbf{H}_{zx_{i+1}} \Phi_{t_i}^{t_{i+1}} \mathbf{P}_{xx}(t_i) \Phi_{t_{i+1}}^{t_i} \mathbf{H}_{zx_{i+1}}^T + \mathbf{H}_{zx_{i+1}} \Phi_{t_i}^{t_{i+1}} \mathbf{Q}_d \Phi_{t_{i+1}}^{t_i} \mathbf{H}_{zx_{i+1}}^T \\ & + \mathbf{H}_{zk} \sigma_k^2 \mathbf{H}_{zk}^T + \mathbf{H}_{zz_i} \mathbf{R}_{zz}(t_i) \mathbf{H}_{zz_i}^T + \mathbf{R}_{zz}(t_{i+1}) \end{aligned} \quad (4.120)$$

Now that the predicted pixel location mean and covariance are determined, the correspondence search for the feature of interest now proceeds in an identical fashion to the binocular method. This search procedure can be accomplished for each candidate feature in an image, and results in a list of correspondences between features in multiple images.

Given a successful correspondence between a single feature or a collection of features, the measurements can be used to estimate the relative location of each landmark. Once a landmark has an initial position estimate, the navigation state vector is augmented and subsequent measurements are used to refine the landmark location and estimate the errors in the navigation state. A complete method to accomplish this task is presented by Strelow [62].

This approach has some limitations which warrant discussion. The first issue is the potential lack of distance observability when the camera translation distance is small relative to the landmark distance. This is equivalent to the issues encountered when attempting to estimate the location of distant landmarks in a binocular scenario (see Section 4.3.3.3). In addition, even when sufficient egomotion exists in order to observe the landmark position, unconstrained optical methods suffer from

the so-called scale ambiguity [36]. This scale ambiguity limits the observability of an image correspondence measurement to relative angles and a direction of translation vector. The scale of the translation is not observable. While this is not a catastrophic condition, any errors in the estimated velocity state result in a distance bias, which causes an un-modeled accumulation of errors in the extended Kalman filter. In order to compensate for this effect, the process noise must be increased as discussed in [38]. An example of this effect is demonstrated via simulation in Chapter VI.

4.4 *Absolute Measurements*

In the previous section, a theory for exploiting the apparent location of landmarks in multiple images was presented, where incomplete information regarding the landmark location was available. This is a form of relative navigation, because the estimated location of the landmark (and navigation state) are all relative to previous estimates of the navigation state, and, by definition, multiple measurements in time are required to extract navigation information.

An absolute measurement can be exploited when complete position information is available for a given landmark. In contrast to relative measurements, absolute measurements do not suffer from long-term drift due to integration of errors. In addition, since absolute measurements do not require estimation of the landmark location, no additional filter states are required. In this section, three types of absolute landmark measurements are presented and analyzed.

4.4.1 Reference Landmark. A reference landmark requires both the prior knowledge of the physical location relative to the navigation frame and the feature descriptor vector. Performing a reference landmark measurement requires three steps: prediction of the landmark location in feature pose space, performing a statistically-constrained correspondence search for the feature of interest in the current measured feature vectors, and finally incorporating the measured location of the reference landmark into the filter.

Given a known reference landmark with position, $\mathbf{y}^n$, and uncertainty, $\mathbf{P}_{yy}$, the pixel location in the current image is a nonlinear function of the navigation state and the landmark location:

$$\mathbf{z} = \mathbf{T}_c^{pix} \underline{\mathbf{s}}^c \quad (4.121)$$

where

$$\mathbf{s}^c = \mathbf{C}_b^c [\mathbf{C}_n^b (\mathbf{y}^n - \mathbf{p}^n) - \mathbf{p}_{cam}^b] \quad (4.122)$$

The pixel location uncertainty, $\mathbf{P}_{zz}$, is defined as

$$\mathbf{P}_{zz} = \mathbf{E} [\delta \mathbf{z} \delta \mathbf{z}^T] \quad (4.123)$$

where

$$\delta \mathbf{z} = \mathbf{H}_{zx} \delta \mathbf{x} + \mathbf{H}_{zy} \delta \mathbf{y} + \mathbf{v} \quad (4.124)$$

The influence matrices, $\mathbf{H}_{zx}$ and $\mathbf{H}_{zy}$ are the partial derivatives of the measurement equation with respect to the navigation state and landmark errors, respectively. The pixel measurement errors are represented by $\mathbf{v}$. Substituting Equation (4.124) into (4.123), and noting the independence between the navigation, landmark, and pixel measurement errors results in the following expression for the pixel location uncertainty covariance

$$\mathbf{P}_{zz} = \mathbf{H}_{zx} \mathbf{P}_{xx} \mathbf{H}_{zx}^T + \mathbf{H}_{zy} \mathbf{P}_{yy} \mathbf{H}_{zy}^T + \mathbf{R}_{zz} \quad (4.125)$$

The main difference between the statistics of relative and absolute landmarks is the independence of the landmark position and navigation state errors.

Once the pixel location and associated covariance matrix have been calculated, the next step is to determine the potential correspondence between the features measured from the current image. This procedure is identical to that presented in Section 4.2.3, where the feature distance is calculated for all features within a search region defined by statistics of the predicted pixel location.

Finally, if there is a successful match, the measured pixel location errors are used to update the navigation states. The rightmost two terms in Equation (4.124) are combined to produce the measurement equation:

$$\delta \mathbf{z} = \mathbf{H}_{zx} \delta \mathbf{x} + \mathbf{v}^* \quad (4.126)$$

where

$$\mathbf{R}_{zz}^* = \mathbf{E} \left[\mathbf{v}^* \mathbf{v}^{*T} \right] \quad (4.127)$$

$$\mathbf{R}_{zz}^* = \mathbf{H}_{zy} \mathbf{P}_{yy} \mathbf{H}_{zy}^T + \mathbf{R}_{zz} \quad (4.128)$$

4.4.2 Revisitation of Previously Tracked Landmark. When navigating using landmarks of opportunity (i.e., no or incomplete prior position information), the methods presented in this chapter require the Kalman filter to maintain an estimate of the three-dimensional location of the landmark. By maintaining a database of these landmarks, their feature descriptors, and position estimates, this information can be leveraged when revisiting these landmarks and resulting in absolute position measurements. In other words, these methods allow a properly equipped sensor to “learn” its environment and exploit this information to eliminate a major source of error. This concept is demonstrated using both simulation and experimental methods in Chapter VI.

The development of the measurement equations for a previously estimated landmark is identical to the previous section, with one consideration. The pixel location uncertainty covariance shown in Equation (4.125), assumes that the navigation state errors and the landmark location errors are independent. This is not always true when revisiting a previously tracked landmark as the cross-correlation matrix is non-zero. Fortunately, the errors decorrelate over time due to the integration of noise sources within the filter. Thus, care must be taken to ensure revisited landmarks are sufficiently decorrelated before using them for reference landmark updates.

4.4.3 *Celestial Measurements.* As mentioned in Section 3.1.1, celestial bodies represent a very high-quality source of optical navigation measurements. The key difference between celestial measurements and landmark measurements is the great distance to celestial bodies relative to the size of the Earth.

The celestial measurement is simply the current pixel location of an identified celestial object. Given a celestial object n , located at a known direction relative to the Earth (based on astronomical almanac data) [25], is represented by the unit vector, $\check{\mathbf{s}}_n^e(t_i)$, at time t_i [16]. The object direction unit vector in the camera frame, $\check{\mathbf{s}}_n^c(t_i)$, is a function of the navigation frame orientation, vehicle orientation, and camera-to-body orientation:

$$\check{\mathbf{s}}_n^c(t_i) = \mathbf{C}_b^c \mathbf{C}_n^b \mathbf{C}_e^n \check{\mathbf{s}}_n^e(t_i) \quad (4.129)$$

where $\mathbf{C}_e^n$ is the Earth frame to navigation frame direction cosine matrix. Using the camera model, the undistorted pixel location, $\mathbf{z}(t_i)$, is a function of the camera projection matrix, $\mathbf{T}_c^{pix}$, and the homogeneous form of the object direction unit vector:

$$\mathbf{z}(t_i) = \mathbf{T}_c^{pix} \check{\mathbf{s}}_n^c(t_i) + \mathbf{v}(t_i) \quad (4.130)$$

where $\mathbf{v}(t_i)$ is additive white Gaussian noise.

By employing the techniques presented in Section 4.3.3, the influence matrix is calculated in a straightforward manner. The influence matrix corresponding to the attitude uncertainty is defined as

$$\mathbf{H}_{z\psi} = \frac{\partial \mathbf{z}}{\partial \psi} \quad (4.131)$$

Evaluating the partial derivative yields

$$\mathbf{H}_{z\psi} = \mathbf{T}_c^{pix} \frac{\check{s}_z^c \frac{\partial \check{s}_z^c}{\partial \psi} - \check{\mathbf{s}}^c \left[\frac{\partial \check{\mathbf{s}}^c}{\partial \psi} \right]_z}{(\check{s}_z^c)^2} \quad (4.132)$$

where

$$\frac{\partial \check{\mathbf{s}}^c}{\partial \boldsymbol{\psi}} = -\mathbf{C}_b^c \mathbf{C}_n^b (\mathbf{C}_e^n \check{\mathbf{s}}^e \times) \quad (4.133)$$

The influence matrix corresponding to the camera misalignment is defined as

$$\mathbf{H}_{z\alpha} = \frac{\partial \mathbf{z}}{\partial \boldsymbol{\alpha}} \quad (4.134)$$

Evaluating the partial derivative yields

$$\mathbf{H}_{z\alpha} = \mathbf{T}_c^{pix} \frac{\check{s}_z^c \frac{\partial \check{\mathbf{s}}^c}{\partial \boldsymbol{\alpha}} - \check{\mathbf{s}}^c \left[\frac{\partial \check{\mathbf{s}}^c}{\partial \boldsymbol{\alpha}} \right]_z}{(\check{s}_z^c)^2} \quad (4.135)$$

where

$$\frac{\partial \check{\mathbf{s}}^c}{\partial \boldsymbol{\alpha}} = (\mathbf{C}_b^c \mathbf{C}_n^b \mathbf{C}_e^n \check{\mathbf{y}}^e) \times \quad (4.136)$$

All other influence matrices are zero.

In the next chapter, a practical image and inertial navigation sensor is designed and calibrated using the algorithms presented in the previous sections.

V. Experimental Navigation System Design and Calibration

Due to the complex nature of the image-aided inertial navigation problem, experimental data collections are critical in order to demonstrate the performance of the proposed navigation algorithm. In this chapter, the design and calibration of the experimental navigation system is presented. The chapter begins by outlining an overall system architecture, then describes the sensors and their respective data products. Finally, the calibration and tuning methods are presented.

5.1 *Experimental Image-Aided Navigation System Overview*

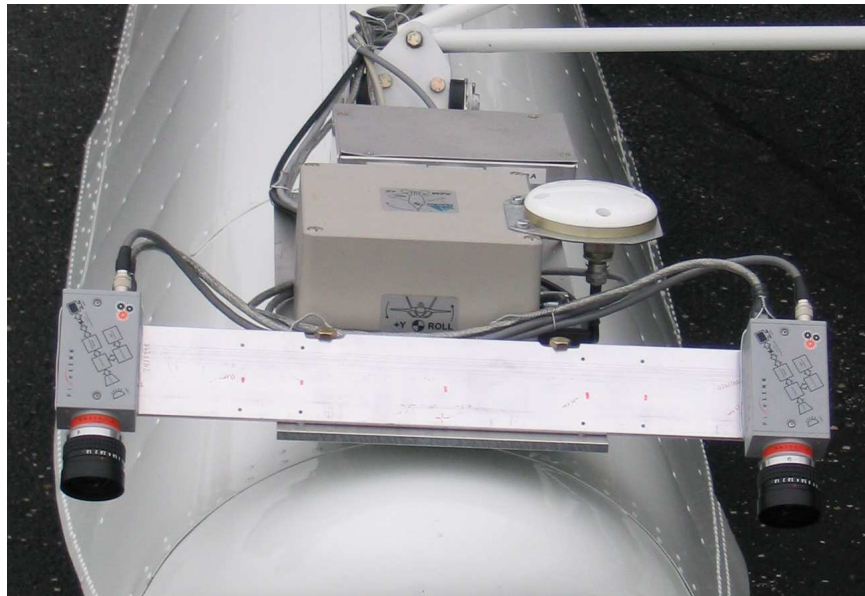
The image-aided navigation system used in the research consists of a navigation computer, inertial measurement unit, and a pair of digital cameras. As shown in Figure 5.1, the system is configured in two ways, depending on the conditions of the data collection. In the ground system, the cameras are mounted level to the sensor plate with approximately 22 cm of binocular disparity. In the flight system, the cameras are mounted on a rigid bracket which is pointed downward by approximately 45 degrees in order to maintain a view of the ground. The cameras are spaced approximately 44 cm apart. The larger binocular disparity increases the effective range for reliably determining the landmark distances using binocular stereopsis.

The sensors are connected to a navigation computer using various protocols. A discrete bus is used to synchronize the sensors with the master clock. A block diagram of the system is shown in Figure 5.2. The details of the system components are given in the following sections.

5.1.1 Sensors. As shown in the block diagram, the sensor test platform consists of digital cameras and inertial measurement units. The digital cameras are both PixeLINK PL-A741 machine vision cameras. The PL-A741 camera uses a monochrome complementary metal-oxide semiconductor (CMOS) imaging sensor with 1280x1024 pixel (1.3 megapixel) resolution. The pixels are 6.7 μm square which results in a sensor size of 8.576 mm $\times$ 6.921 mm. The PL-A741 camera accepts stan-



(a)



(b)

Figure 5.1: Image-aided inertial navigation system. The navigation system consists of a tactical-grade IMU, consumer-grade IMU, and two monochrome digital cameras mounted to a rigid plate. Panel (a) shows the system used for ground testing. Panel (b) shows the system as installed for flight testing.

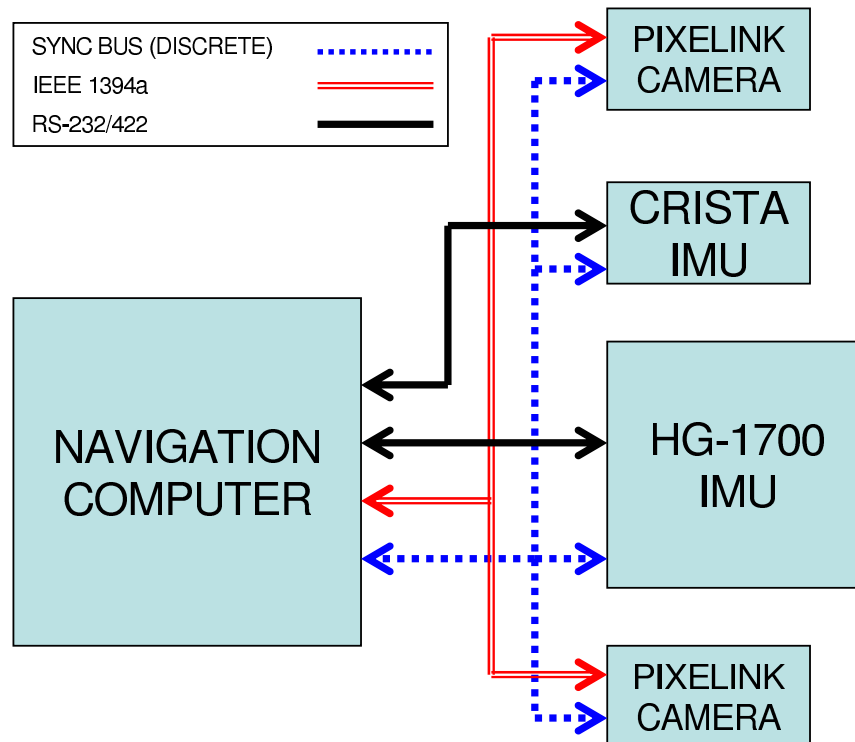


Figure 5.2: Navigation system block diagram. The image-aided inertial navigation system consists of tactical and consumer grade IMUs, two digital cameras, and a navigation computer. The sensors are synchronized using a discrete bus.

dard C-mount lenses. The camera communicates with the navigation computer using an IEEE-1394a (FireWire) interface, which is capable of transmitting approximately 3 frames per second at full resolution. The PL-A741 includes an external interface which can be used to trigger synchronized frame captures. Finally, the camera includes a global shutter which exposes all of the pixels simultaneously. This avoids one of the primary limitations of standard, line-scanned video sources. The complete PL-A741 specifications are listed in [49].

The cameras are paired with Pentax/Cosmicar C30405 C-mount lenses. These lenses feature a manual iris which ranges from full closed to $f/1.8$. The 4.8 mm focal length results in a horizontal field of view of approximately 45 degrees.

In order to test the performance of the system with variations in inertial sensors, both tactical and consumer-grade IMUs are mounted to the platform. The system is configured to record measurements from both sensors simultaneously so that the same navigation profile can be processed using each of the sensors separately.

The Honeywell HG1700 is a tactical-grade, strapdown inertial measurement unit. The unit consists of a GG1308 ring-laser gyroscope and triad of RBA-500 accelerometers which produce measurements at 100 Hz. The gyroscopes and accelerometers have a dynamic range of ± 1000 degrees per second and ± 50 g, respectively. Selected specifications for the HG1700 are shown in Table 5.1.

The Crista IMU is a consumer-grade, strapdown inertial measurement unit. The IMU consists of micro-electrical mechanical systems (MEMS) gyroscopes and accelerometers which produce measurements at 100 Hz. The gyroscopes and accelerometers have a dynamic range of ± 300 degrees per second and ± 10 g, respectively. Selected specifications for the Crista IMU are shown in Table 5.1.

5.2 Calibration and Alignment

The ultimate goal of the calibration and alignment process is to determine the relative orientation between an imaging and inertial sensor, or more specifically, the

Table 5.1: Inertial measurement sensor specifications for the Cloud Cap Technology Crista consumer-grade IMU [6] and Honeywell HG1700 tactical-grade IMU [22]. The parameters noted with an asterisk are not included in the specifications and are estimates.

Parameter (Units)	Crista IMU	HG1700
Sampling interval (ms)	5.0	10.0
Gyro bias sigma (deg/hr)	1800	1.0
Gyro bias time constant (hr)	2*	2*
Angular random walk ($deg/\sqrt{hr}$)	2.23	0.3
Gyro scalefactor sigma (PPM)	10000	150
Accel bias sigma (m/s^2)	0.196	0.0098
Accel bias time constant (hr)	2*	2*
Velocity random walk ($m/s/\sqrt{hr}$)	0.261	0.57*
Accel scalefactor sigma (PPM)	10000	300

sensitive axes of both sensors. The process can be divided into two main steps. The first step consists of calibrating the optical sensors with respect to unknown optical properties. The final step is to determine the relative location and orientation between the sensitive axes of the imaging and inertial sensors. Both steps are presented in detail in the following sections and address some limitations found in the current literature.

5.2.1 Camera Calibration. The camera and lens system must be properly calibrated in order to accurately map a pixel location to a line of sight vector, which is critical to the performance of the system. As presented in Sections 2.7.3 and 2.7.2, the camera/lens model consists of both a projection model based on a pinhole camera with an associated nonlinear correction to remove the effects of radial optical distortion caused by the lens. The calibration parameters of interest are

- Horizontal pixel size (s_x)
- Vertical pixel size (s_y)
- Effective focal length (f)
- Nonlinear optical distortion parameters (a_2, a_4)

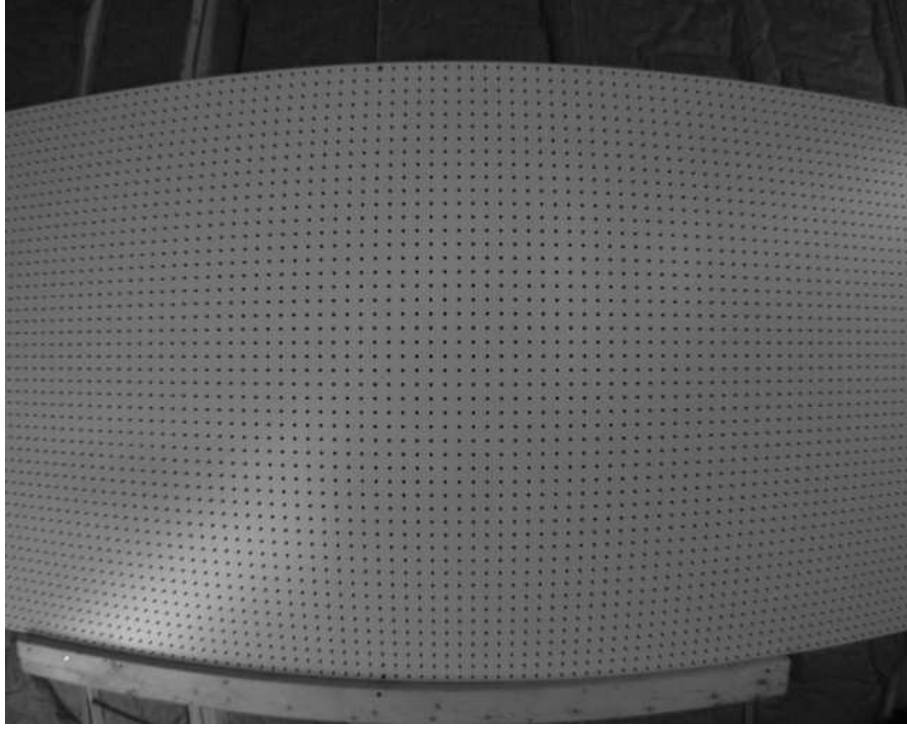


Figure 5.3: Sample image of optical calibration array. The calibration array consists of a 48×96 inch board with reference holes drilled at one inch intervals. The effects of the optical distortion which cause straight lines to appear curved are visible in the image.

- Center of optical distortion (c)

In this experiment, the specified values of the horizontal and vertical pixel size parameters are assumed to be accurate and are not explicitly estimated. This is justified due to the precise nature of the photolithographic process used to create the CMOS array.

The calibration is accomplished by imaging a series of surveyed reference points and measuring the resulting pixel location of each point. In order to maintain observability of all parameters, images must be taken at varying distances. The reference consists of a flat, 4×8 foot board with calibration holes drilled at one inch intervals. A sample image of the calibration array is shown in Figure 5.3. Note the effects of radial distortion in the image, which cause straight lines to appear curved in the image.

In order to predict the pixel location of each calibration point in the image, the relative pose of the camera is required. Unfortunately, the true pose of a camera

is difficult to independently measure to the relative precision required to properly observe the camera calibration parameters. Thus a method was used to estimate the pose and calibration parameters simultaneously. In estimation terminology, the pose is considered a nuisance parameter which is required in order to estimate the true parameters of interest.

Because the cameras are arranged for binocular imaging, estimation of the calibration parameters is accomplished concurrently. Since the relative pose between the cameras is constant and partially known, this imposes an additional constraint on the estimation process. Incorporation of the additional parameters required for binocular estimation results in the following parameter list.

- Effective focal length for camera a (f_a)
- Effective focal length for camera b (f_b)
- Nonlinear optical distortion parameters for camera a (a_{a_i})
- Nonlinear optical distortion parameters for camera b (a_{b_i})
- Center of optical distortion for camera a ($\mathbf{c}_a$)
- Center of optical distortion for camera b ($\mathbf{c}_b$)
- Location of camera a with respect to the binocular reference frame ($\mathbf{p}_{cam_a}^{c_0}$)
- Location of camera b with respect to the binocular reference frame ($\mathbf{p}_{cam_b}^{c_0}$)
- Binocular reference frame to camera a frame DCM ($\mathbf{C}_{c_0}^{c_a}$)
- Binocular reference frame to camera b frame DCM ($\mathbf{C}_{c_0}^{c_b}$)
- Location of binocular reference in navigation frame for image j ($\mathbf{p}_j^n$)
- Navigation to binocular reference frame DCM for image j ($\mathbf{C}_{n_j}^{c_0}$)

The binocular measurement geometry is shown in Figure 4.11.

The resulting nonlinear regression function for camera a , image j , is given by

$$\mathbf{h}_{1a}(f_a, \mathbf{c}_a, \mathbf{p}_j^n, \mathbf{C}_{n_j}^{c_0}, \mathbf{C}_{c_0}^{c_a}) - \mathbf{h}_{2a}(\mathbf{c}_a, a_{a_2}, a_{a_4}, \mathbf{z}_{a_j}^d) = 0 \quad (5.1)$$

and similarly for camera b , image j

$$\mathbf{h}_{1b}(f_b, \mathbf{c}_b, \mathbf{p}_j^n, \mathbf{C}_{n_j}^{c_0}, \mathbf{C}_{c_0}^{c_b}) - \mathbf{h}_{2b}(\mathbf{c}_b, a_{b2}, a_{b4}, \mathbf{z}_{b_j}^d) = 0 \quad (5.2)$$

where $\mathbf{z}^d$ represents the pixel location of the calibration point prior to removal of the optical distortion.

The $\mathbf{h}_1$ function represents the projection of the calibration point, $\mathbf{y}^n$, into the image plane using the pinhole camera model. This function is defined for camera a as

$$\mathbf{h}_{1a} = \begin{bmatrix} -f_a/s_x & 0 & \vdots & \mathbf{c}_{a2 \times 1} \\ 0 & f_a/s_y & \vdots & \vdots \end{bmatrix}_{2 \times 3} \underline{\mathbf{s}}^{c_a} \quad (5.3)$$

where

$$\mathbf{s}^{c_a} = \mathbf{C}_{c_0}^{c_a} [\mathbf{C}_n^{c_0} (\mathbf{y}^n - \mathbf{p}^n) - \mathbf{p}_{cam_a}^{c_0}] \quad (5.4)$$

The $\mathbf{h}_2$ function represents the effects of optical distortion, which is defined in Equation (2.91), and repeated here for clarity for camera a

$$\mathbf{h}_{2a} = (\mathbf{z}_a^d - \mathbf{c}_a) (1 + a_{a2}r^2 + a_{a4}r^4) + \mathbf{c}_a \quad (5.5)$$

where

$$r = \|\mathbf{z}_a^d - \mathbf{c}_a\| \quad (5.6)$$

Thus, a nonlinear regression can be performed by matching the true location of each calibration point with its projected location in each image.

When performing this calibration, care must be taken to promote observability in several areas. The first area of potential unobservability occurs when estimating the nonlinear optical distortion parameters, since both are functions of pixel radius (one in r^2 and one in r^4). This issue is mitigated by ensuring the calibration points are located at a wide variety of radial distances from the center of the image. The second area of weak observability occurs when estimating the camera orientation and location relative to the binocular reference frame. Assuming the unknown orientation of the

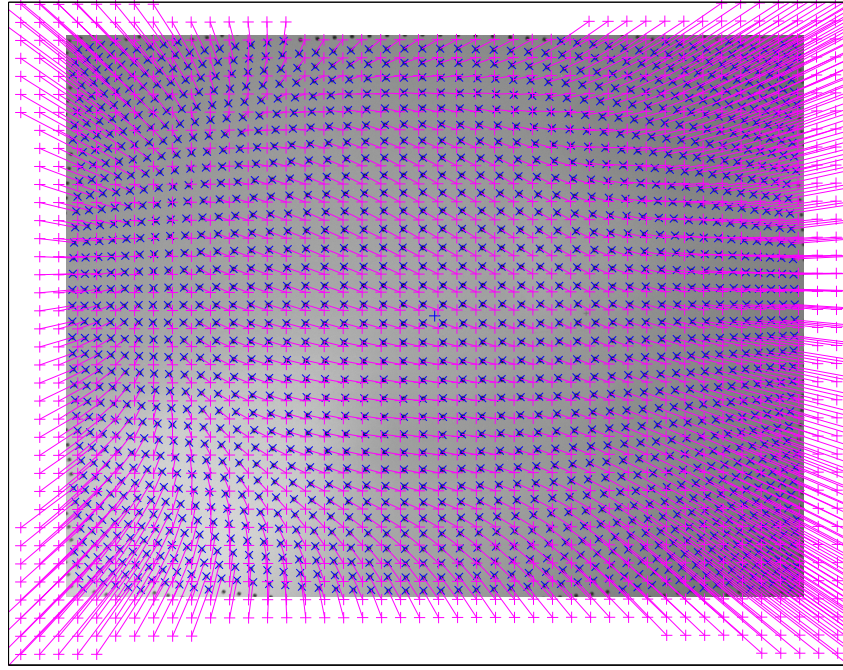
cameras relative to the binocular reference frame is small, the relative orientation between the cameras is estimated, instead of each orientation separately. In addition, since the relative location of the cameras is well known, an estimate of the value is not needed and the parameters are removed from the regression. Finally, there is unobservability between the distance component of the position vector and the estimated focal length. This is mitigated by accurately measuring the distance to the calibration array for each image and effectively removing this variable from the regression.

The binocular calibration is accomplished at three measured distances from the calibration array. A sample image which shows the prediction errors before and after the calibration process is shown in Figures 4(a) and 4(b). A total of 10397 measurements of calibrated locations are used to estimate the intrinsic binocular parameters. The resulting mean of the squared pixel matching errors was 0.84, which agreed with the expected pixel measurement uncertainty due to the correlation process.

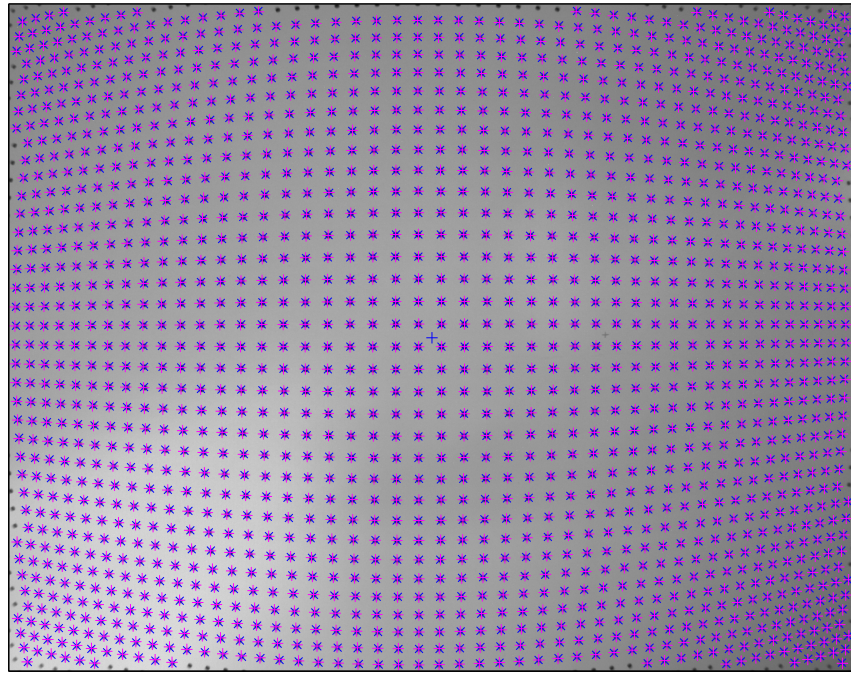
Once the binocular calibration has been completed, the next step is to estimate the relative location and orientation between the binocular rig and the inertial measurement unit. The procedure used to accomplish this is presented in the next section.

5.2.2 Sensor Alignment. In order to accurately combine the measurements from optical and inertial sensors, the relative position and orientation between the sensors must be known. More precisely, the relative position and orientation must be determined between the *sensitive axes* of the respective sensors. In this section, a method is proposed which estimates these parameters by combining independent truth information with imaging and inertial sensor measurements of features with known relative locations.

The method consists of two steps, which may be treated independently. The first task is to estimate the location and orientation of the IMU center by combining IMU measurements with a position and orientation source of appropriate accuracy (e.g.,



(a) Before calibration.



(b) After calibration.

Figure 5.4: Binocular camera calibration. This pair of sample images illustrates the improvements achieved during the camera calibration. The magenta plus symbols indicate the predicted locations of the calibration points and the blue crosses indicate the actual locations of the calibration points in the image.

GPS or high-quality stationary alignment [54]). In this experiment, differentially-corrected GPS measurements are optimally combined with the inertial measurements in order to produce an accurate position and orientation estimate (with associated uncertainty covariance) at times corresponding with image capture events.

The second task is to measure the pixel locations of features with known physical location relative to the navigation frame from a single or multiple images. This relationship is governed by the pixel measurement equation:

$$\mathbf{z} = \mathbf{h}(\mathbf{C}_{c_0}^{c_a}, \mathbf{C}_b^{c_0}, \mathbf{C}_n^b, \mathbf{y}^n, \mathbf{p}^n, \mathbf{p}_0^b, \mathbf{p}_{cam}^{c_0}) \quad (5.7)$$

where the body to binocular frame DCM, $\mathbf{C}_b^{c_0}$, and the relative location vector between the body and binocular frame, $\mathbf{p}_0^b$, are the object parameters of the nonlinear regression. More explicitly, the pixel measurement equation is given by

$$\mathbf{z} = \mathbf{T}_{c_a}^{pix} \underline{\mathbf{s}}^{c_a} \quad (5.8)$$

where

$$\underline{\mathbf{s}}^{c_a} = \mathbf{C}_{c_0}^{c_a} \{ \mathbf{C}_b^{c_0} [\mathbf{C}_n^b (\mathbf{y}^n - \mathbf{p}^n) - \mathbf{p}_0^b] - \mathbf{p}_{cam}^{c_0} \} \quad (5.9)$$

As discussed in the previous section, there are observability issues which must be understood in order to ensure an accurate estimate. The first issue is the limited visibility of the relative location vector. This issue is mitigated by measuring the relative location between the two sensors and leaving the relative location out of the estimation process. The next issue regarding observability is related to the fundamental measurement accuracy of the sensors, primarily the measurement accuracy of the gyroscope and the camera angular accuracy. This issue is investigated using simulations of inertial sensors with varying accuracy.

A qualitative estimate of achievable camera to body alignment accuracy for various inertial sensor models as a function of gyroscope random walk is shown in Figure 5.5. As expected, improving the quality of the inertial sensor yields improved

Table 5.2: Camera to inertial alignment accuracy accelerometer / gyroscope performance sensitivity. Theoretical alignment performance (in $1 - \sigma$ RMS arcseconds) is shown as a function of various combinations of sensors from the Crista, HG1700, and HG9900 IMUs after completing a multi-angle, stationary alignment profile.

	HG9900 Gyro	HG1700 Gyro	Crista Gyro
HG9900 Accel	11.1"	22.7"	41.3"
HG1700 Accel	11.6"	45.4"	71.8"
Crista Accel	12.2"	68.1"	291.2"

alignment, up to the accuracy of the image sensor to resolve the feature locations. In this case, the camera alignment estimate for the navigation grade IMU (HG9900) is limited by the image measurements. This could be improved by increasing the angular sampling frequency of the imaging sensor [15]. A sensitivity analysis is performed by creating theoretical combinations of accelerometer and gyroscopic sensors and estimating the boresight alignment accuracy using the standard alignment profile. The results, shown in Table (5.2), indicate the performance of the gyroscope has a stronger contribution to the achievable boresight accuracy, although both sensors influence the ultimate performance due to coupling between accelerometer and gyroscopic error states.

These results illustrate the relationship between the accuracy of the optical and inertial sensors and the ability to determine the relative orientation of the sensors. More fundamentally, even if a mechanical or other method existed to provide a “better” alignment (e.g., relative to the “case”), it would be impossible to prove as the sensor measurements limit the observability of the relative alignment.

In the next chapter, the mathematical model presented in Chapter IV is validated using simulation and experimental methods.

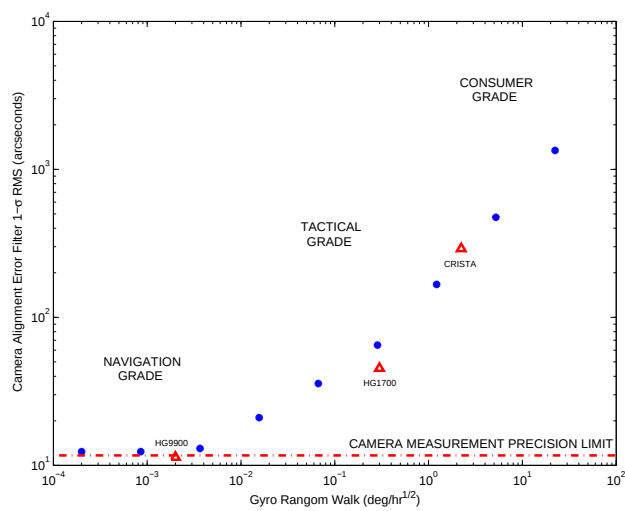


Figure 5.5: Typical camera-to-inertial alignment accuracy for various grades of IMU. Specific results are shown for the Crista, HG1700, and HG9900 IMUs after completing a multi-angle, stationary alignment profile.

VI. Simulation and Experimental Results

In order to validate the mathematical development presented in Chapter IV, the algorithms are evaluated using both simulation and experimental methods. An incremental testing philosophy is used to validate the accuracy of the stochastic feature projection algorithms prior to evaluating the system navigation performance. This chapter is arranged as follows. First, the stochastic projection algorithms are tested using a Monte Carlo simulation and demonstrated with flight test data. Next, the image and inertial fusion algorithm is tested using a ground simulation and verified with a ground test. Finally, the image and inertial fusion algorithm is tested in a simulated flight environment.

6.1 *Stochastic Projection Algorithm Tests*

This experiment validates the stochastic projection method using both simulated and real data collected from an airborne system. A Monte Carlo simulation of the test flight is performed to validate the stochastic projection model with respect to a statistically significant sampling of random error contributors. The flight test data are used to validate the performance of the algorithm in a real-world environment. In the next section, the simulation parameters and results are presented.

6.1.1 Monte Carlo Simulation. The performance of the stochastic projection method presented in Section 4.2.2 is verified using a statistically representative ensemble of sample functions (300 per run). The data collection system used on the experimental test flights is simulated in software, based on a reference trajectory chosen to generate an interesting observation geometry. This constant-altitude circular flight path is constructed such that a fixed terrain patch remained in the camera field of view throughout the flight. The simulated aircraft speed is 150 meters-per-second, altitude is 2296 meters, and bank angle is 27 degrees which describes a circular flight path with 4592 meter radius. The resulting slant range to the landmark is 5134 meters.

The Profgen trajectory generation software tool [40] is used to create the reference trajectory, and the resulting angular rate and specific force profiles. Simulated inertial measurements are generated by subjecting the true angular rate and specific force profiles to randomly generated measurement errors, with error statistics based on the respective inertial sensor model shown in Table 5.1.

The terrain elevation is simulated as a zero-mean, Gaussian random variable. Simulations are accomplished using a terrain elevation error standard deviation of 25 meters, representing a moderate accuracy terrain model. While sample images are not created, simulated landmarks are randomly dispersed over the terrain, with randomized elevations consistent with the terrain elevation error standard deviation. All simulations use a 10-second interval between two images, which is equivalent to 18.7 degrees of arc in the horizontal plane. The simulation geometry is shown in Fig. 6.1.

The results are shown in Fig. 6.2. In this figure, the predicted pixel location errors for each Monte Carlo sample function are represented by a “plus” symbol. The predicted $2\text{-}\sigma$ pixel location error bound is indicated by a line. Note the inclined elliptical nature of the $2\text{-}\sigma$ bound is a function of the trajectory and measurement geometry.

The same predicted pixel location errors are shown referenced to a 256×256 pixel image in Fig. 6.3. The stochastic constraint method shows a small correspondence search area which gives the highest probability of the landmark location. In this case, the stochastic constraint method is an improvement over the epipolar line search method [62] as it provides a smaller search area developed using a statistical model. By observing the resulting elliptical geometry of the representative search space, it can be seen that the statistical search space will not be larger than an equivalently width-constrained epipolar search space. In general practice, the search space will be smaller, especially when additional feature locator constraints are enforced (e.g., scale and rotation).

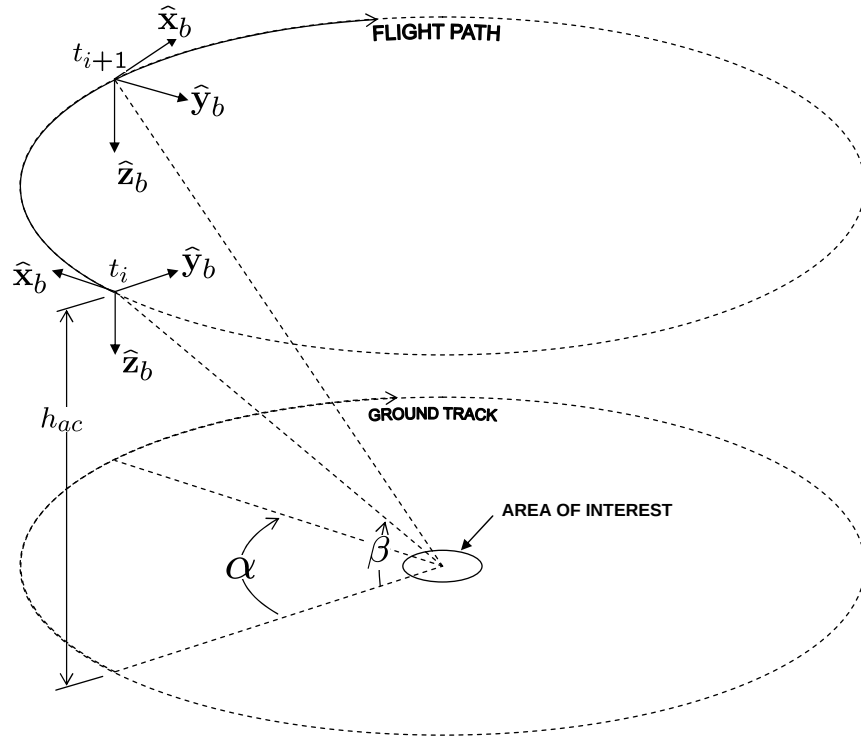


Figure 6.1: Simulated flight path. In order to generate a good observation geometry, the circular orbit is chosen such that a fixed terrain patch remains in the camera field of view throughout the flight.

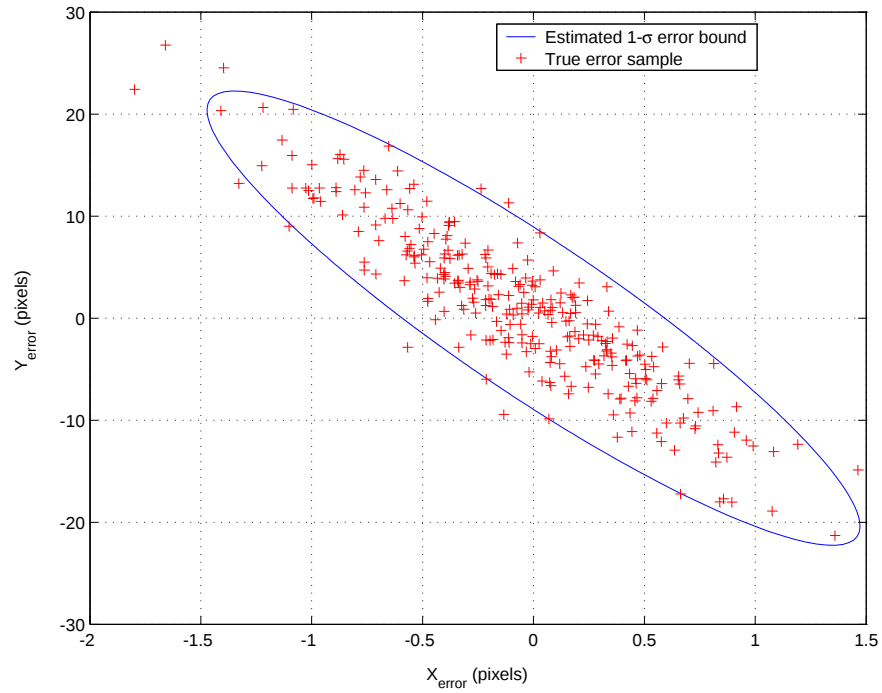


Figure 6.2: Landmark pixel location error and predicted $2\text{-}\sigma$ bound for 25 meter terrain elevation uncertainty. Note the actual pixel location errors are similar to the predicted error bound. *Note: X and Y axes have differing scales to show detail.*

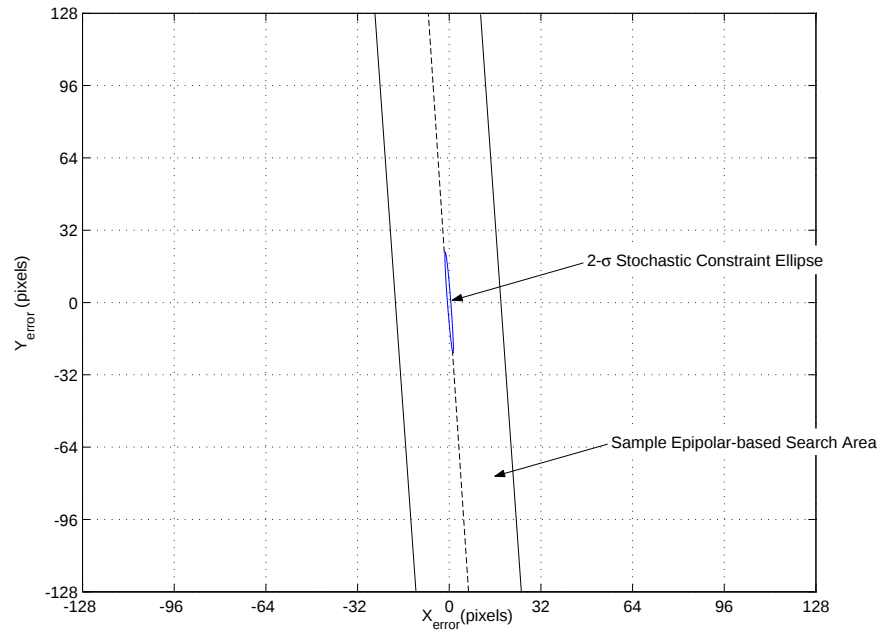


Figure 6.3: Landmark pixel location error and predicted $2\text{-}\sigma$ bound for 25 meter terrain elevation uncertainty referenced to a 256×256 pixel image. Note the stochastic constraint can limit the correspondence search area significantly compared to a search near the epipolar line.

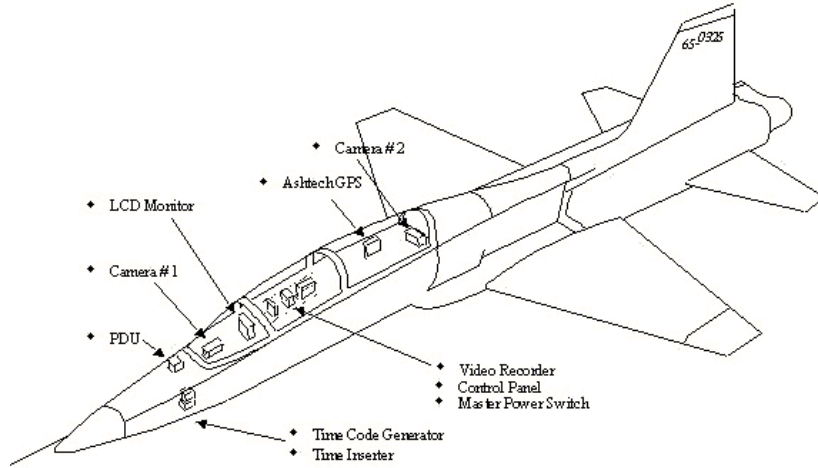


Figure 6.4: Northrop T-38 instrumented with synchronized digital video camera and inertial navigation system.

6.1.2 Flight Test Experiment. In this section, the stochastic projection method is implemented using image and inertial flight data collected on a Northrop T-38 “Talon” aircraft. The aircraft was equipped with a day-night monochrome digital video camera synchronized to a Honeywell H-764G Inertial Navigation System. The camera was mounted in the cockpit, pointing out the right wing. Flight data were collected in Fall of 2002 at Edwards Air Force Base, California.

The aircraft state dynamics are a function of the measurements from the strap-down inertial sensors. All states are estimated in the Earth-centered Earth-fixed reference frame previously defined. The error equations were developed based on [64, 71]. For this example, a three image sequence from a right turning profile is shown in Fig. 6.5. The results of the above method for predicting the future target location and uncertainty are shown in Fig. 6.6.

The target selected is the west corner of a building shown in Fig. 6.5. The estimated target location and $2\text{-}\sigma$ variance shown in Fig. 6.6 shows an predicted ellipsoidal uncertainty after one second and seven seconds of flight. Note the uncertainty ellipse increases with flight time, as expected. In each case, incorporating camera motion information can constrain the correspondence search space significantly. Note the

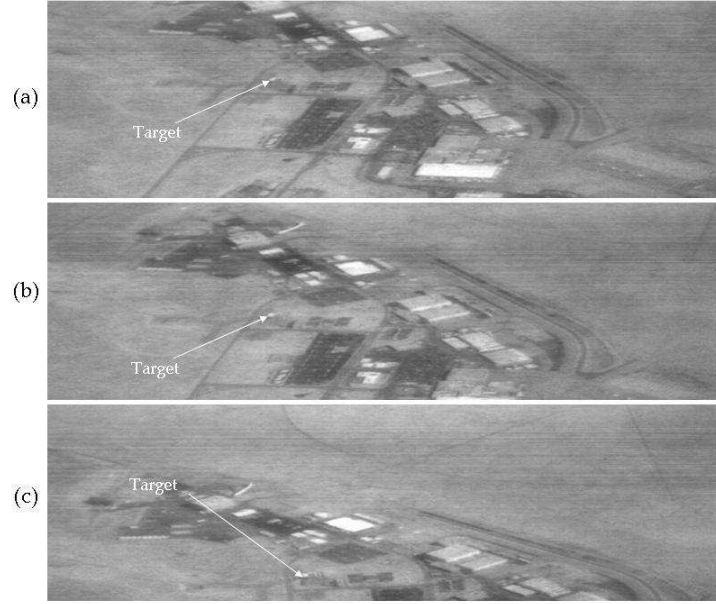


Figure 6.5: Three image sequence of an industrial area recorded during a T-38 flight, with a sample stationary ground landmark identified. Image (b) was taken 1 second after image (a). Image (c) was taken 7 seconds after image (a). The aircraft is in a right turn approximately 3.8 kilometers from the landmark.

true landmark location remains consistent with the predicted $2\text{-}\sigma$ uncertainty ellipse in the presence of real measurement noise and terrain model errors.

6.2 Image and Inertial Fusion Algorithm Ground Tests

After demonstrating the validity of the stochastic feature projection theory, the algorithm is incorporated into the extended Kalman filter navigation algorithm. The initial experiments are conducted using the sensor platform shown in Figure 5.1(a). This system is equipped with a tactical-grade IMU and a binocular pair of Pix-eLINK cameras. The cameras are mounted level with the platform with approximately 22 centimeters of separation.

The imaging and inertial fusion navigation algorithm is evaluated using both simulated and experimental ground profiles. The profiles are designed to provide a range of image types in order to exercise the feature tracking algorithm. The simulation and results are presented in the next section.

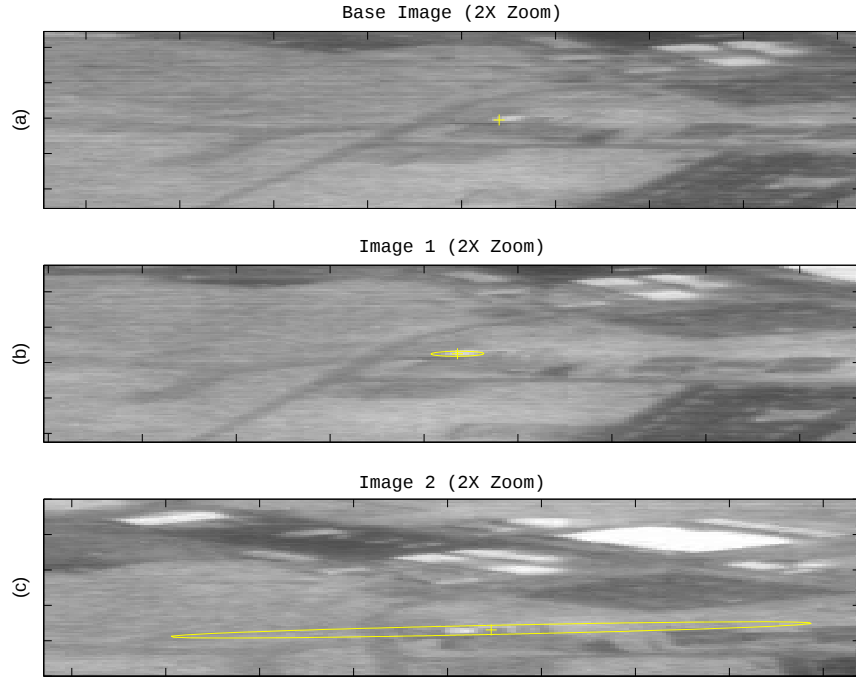


Figure 6.6: Predicted landmark location uncertainty using stochastic projection method. The landmark selected is the west corner of a building in the base image (a), represented by the crosshair. Using the stochastic projection method, the landmark mean and $2\text{-}\sigma$ variance is projected into two subsequent images to demonstrate the concept. The estimated landmark location and predicted $2\text{-}\sigma$ variance for image (b) shows an ellipsoidal uncertainty after one second of flight. Image (c) shows a further increase in the uncertainty after seven seconds of flight. In each subsequent image, constraining the correspondence search for the landmark to the ellipsoidal region reduces the required search area and would eliminate false matches with other features with a similar appearance (e.g., other building corners).

6.2.1 Simulation. The algorithm is tested using a Monte Carlo simulation of a standard indoor profile. The profile consists of a straight corridor, designed to be similar to the indoor experimental data collection presented in Section 6.2.2.

An accurate simulation of the navigation environment requires simulating the performance of the sensors in response to a true motion trajectory. The trajectory was generated using Profgen version 8.19 software package [40]. For each Monte Carlo navigation simulation run, the inertial sensor measurements are generated using the true trajectory and an inertial sensor error model.

Because of the inherent complexity of the optical environment, it is beyond the scope of this research to generate simulated images. Instead, a simulated feature set is created by randomly distributing features along a corridor surrounding the true trajectory. The features are each given random descriptor vectors in order to exercise the feature tracking algorithm. While this optical simulation method is appropriate for testing the image and inertial fusion algorithm, the results are not directly comparable to the real system performance, because imaging issues such as lighting conditions, motion blur, and affine changes in the feature descriptor due to pose changes are not modeled.

The simulated corridor is 3 meters wide, 3 meters high, and approximately 300 meters long. Features are randomly generated on the walls, floor and ceiling of the corridor with an average spacing of 0.25 features per square meter. Each feature is given a random primary length and orientation, which, combined with the true pose of the sensor, results in accurately simulated scale and orientation parameters in feature space. After a 60-second stationary alignment, the sensor platform accelerates to 0.5 meters per second, maintains this velocity until the end of the corridor, then accelerates to a stop at the end. The platform remains stationary for 60 seconds after coming to a stop. This results in a 660-second image and inertial navigation profile. Simulated images are collected at 2 Hz.

A Monte Carlo simulation is conducted using inertial sensor models representing the H1700 tactical-grade IMU and the Crista consumer-grade IMU. Each simulation consists of 60 runs, each with randomly generated inertial measurement errors due to random walks, sensor bias, and sensor scale-factor errors. In order to mitigate any potential effects due to the location of the features in the simulated environment, the feature locations and descriptors are randomly generated every 20 runs.

The position, velocity, and attitude errors using the tactical-grade IMU are shown in Figures 6.7-6.9. As expected, the inertial and imaging measurement errors accumulate, resulting in position and attitude drift.

The position, velocity, and attitude errors using the consumer-grade IMU are shown in Figures 6.10-6.12. In addition to proportionally larger errors in position, velocity, and attitude, significant excursions are noticed, especially in the attitude channels.

Root-sum-squared (RSS) errors are analyzed in order to provide a one-dimensional metric for a more direct comparison of the simulated system performance for different profiles. The RSS errors comparing the free-inertial and image-aided performance is shown in Figures 6.13-6.15. Over the 10-minute indoor profile, incorporating the image-aiding measurements improves the errors for both the consumer and tactical-grade sensors by many orders of magnitude. In addition, the image-aided consumer-grade sensor nearly equals the position error performance of the image-aided tactical-grade sensor, until the consumer-grade sensor begins to show some attitude divergences after approximately 400 seconds.

In this section, the proposed image and inertial navigation system is tested using a simulated indoor profile. The Monte Carlo simulation predicts dramatic improvements when fusing image and inertial measurements for both consumer and tactical-grade inertial sensors, which is an indicator of the significant navigation information provided by properly integrated image updates. In the next section, the experimental data collection profiles and results are presented.

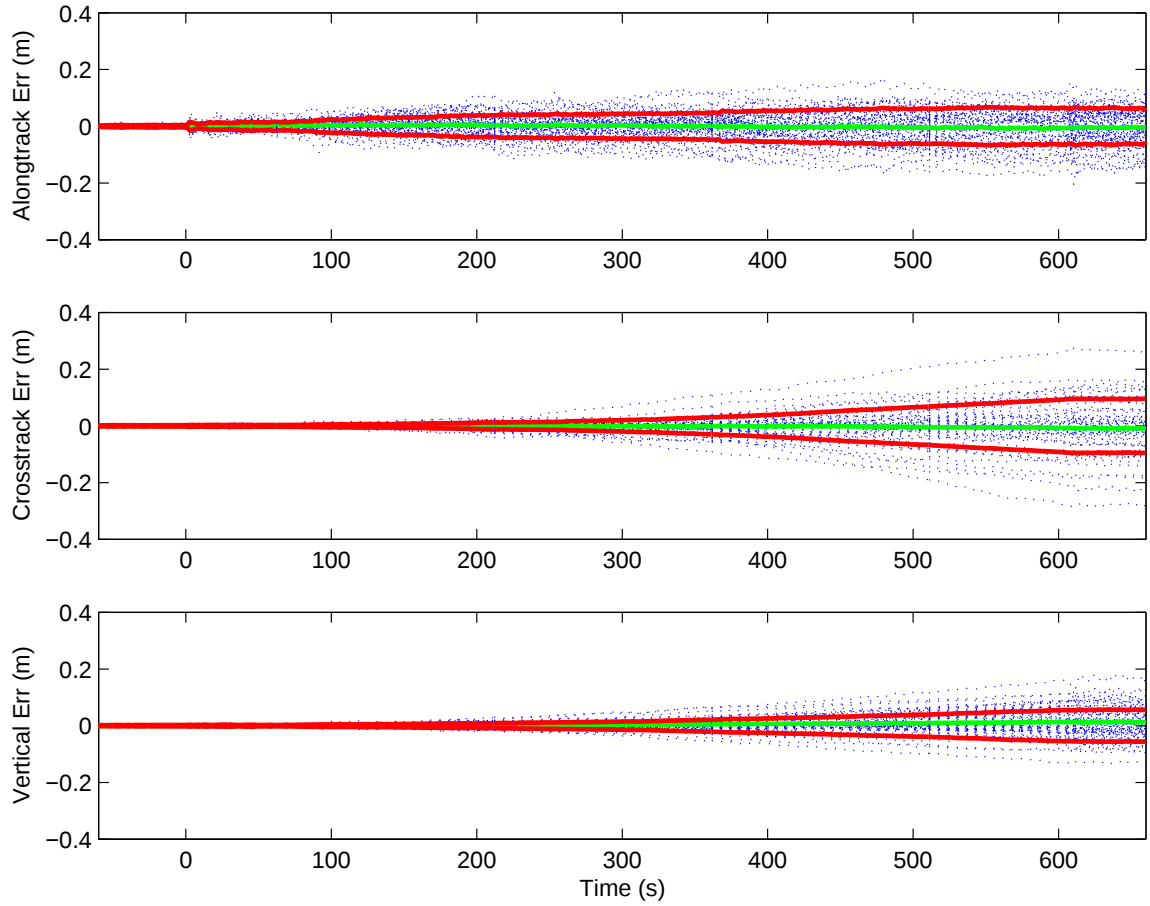


Figure 6.7: Simulated 60-run Monte Carlo position error results for indoor profile with a tactical-grade inertial sensor and image aiding using landmarks of opportunity. The position error sample functions are indicated by blue dotted lines. The ensemble mean and standard deviation are indicated by the green and red solid lines, respectively.

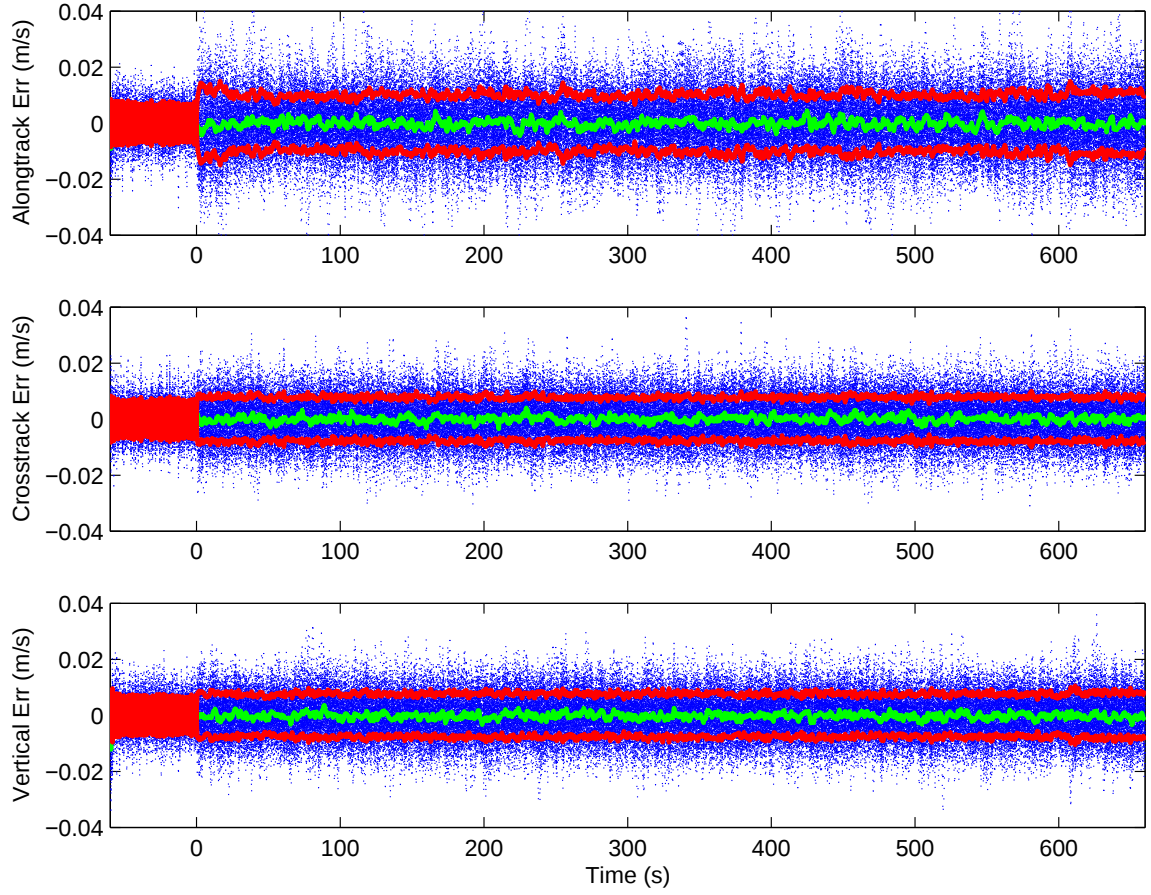


Figure 6.8: Simulated 60-run Monte Carlo velocity error results for indoor profile with a tactical-grade inertial sensor and image aiding using landmarks of opportunity. The velocity error sample functions are indicated by blue dotted lines. The ensemble mean and standard deviation are indicated by the green and red solid lines, respectively.

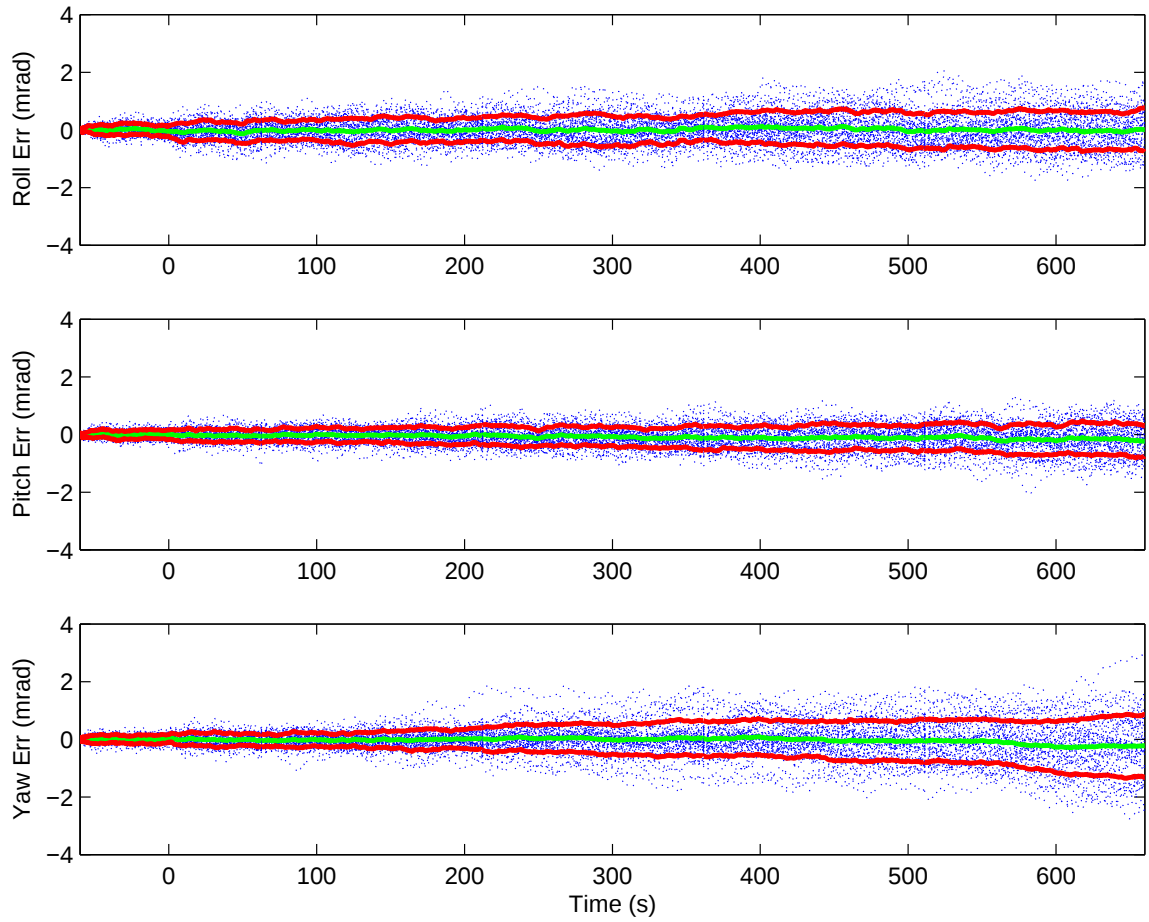


Figure 6.9: Simulated 60-run Monte Carlo attitude error results for indoor profile with a tactical-grade inertial sensor and image aiding using landmarks of opportunity. The attitude error sample functions are indicated by blue dotted lines. The ensemble mean and standard deviation are indicated by the green and red solid lines, respectively.

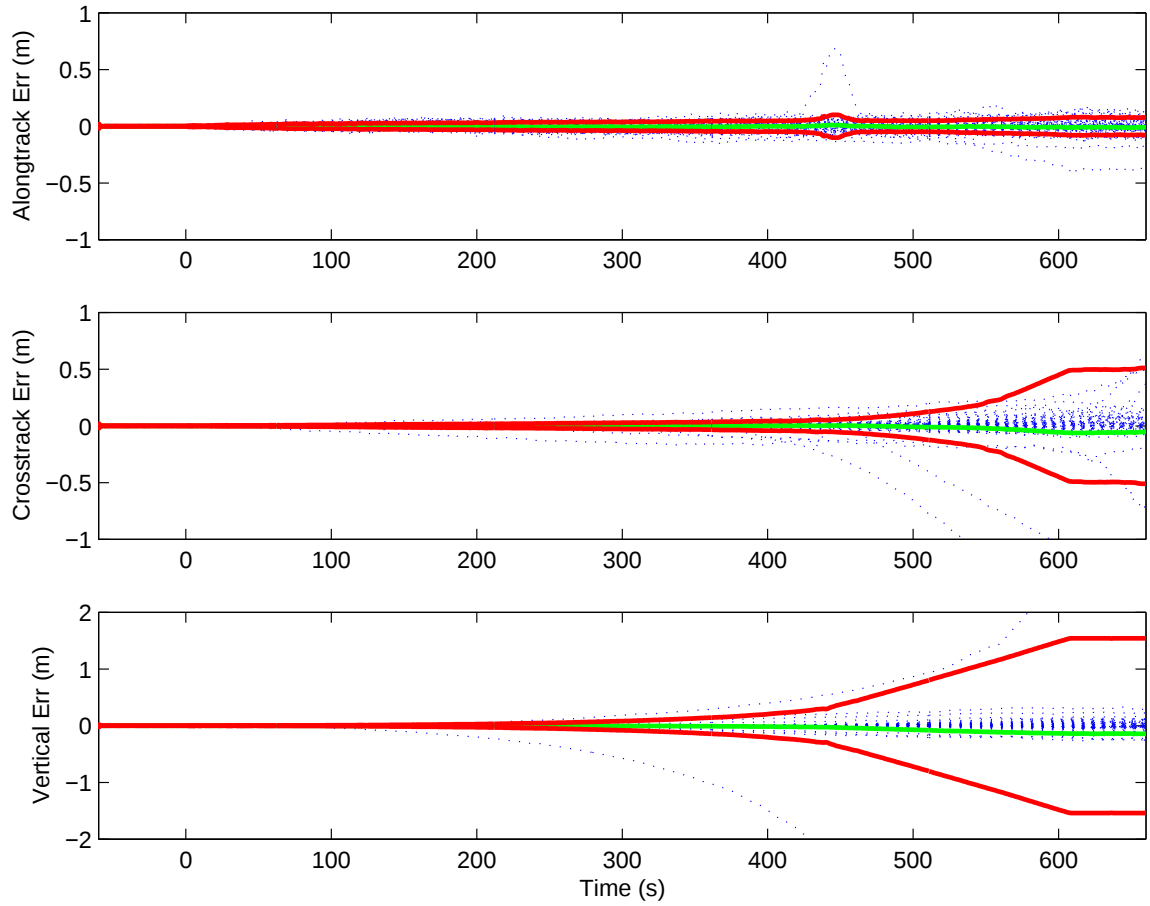


Figure 6.10: Simulated 60-run Monte Carlo position error results for indoor profile with a consumer-grade inertial sensor and image aiding using landmarks of opportunity. The position error sample functions are indicated by blue dotted lines. The ensemble mean and standard deviation are indicated by the green and red solid lines, respectively.

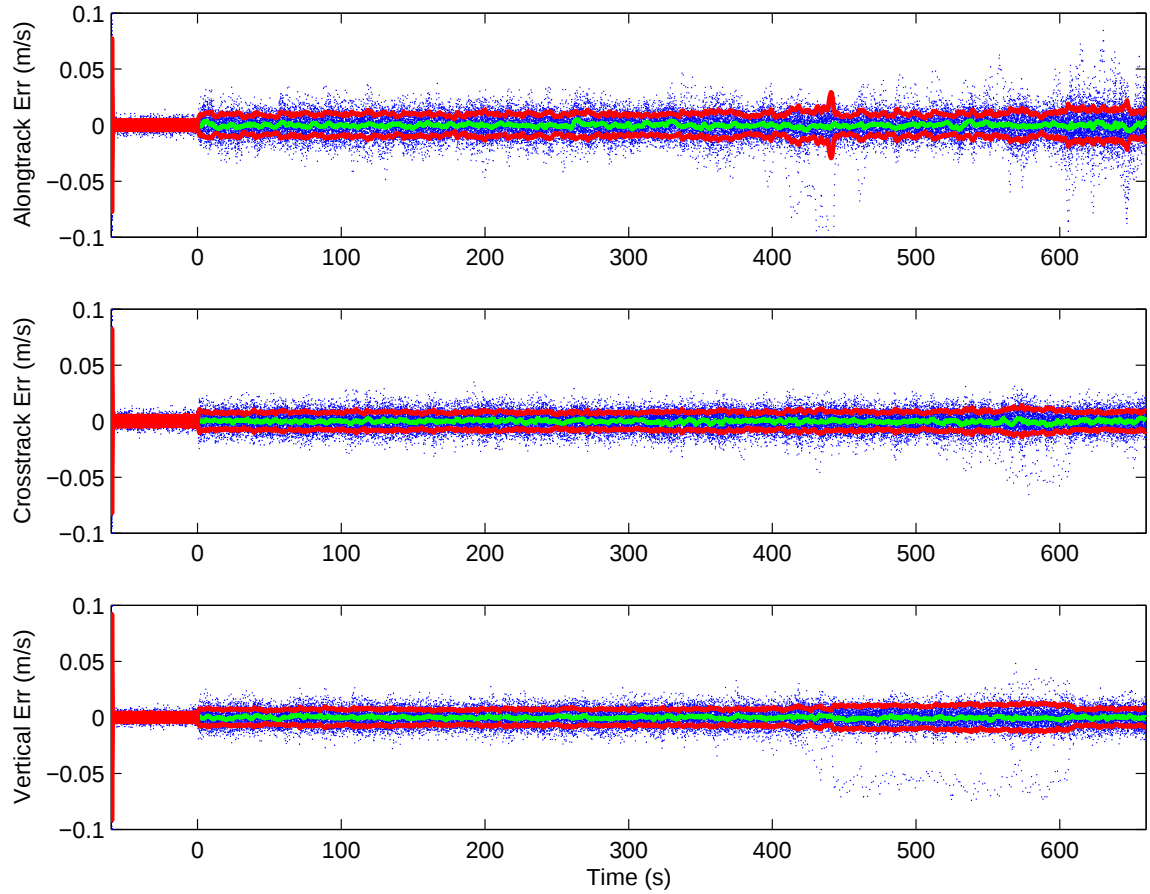


Figure 6.11: Simulated 60-run Monte Carlo velocity error results for indoor profile with a consumer-grade inertial sensor and image aiding using landmarks of opportunity. The velocity error sample functions are indicated by blue dotted lines. The ensemble mean and standard deviation are indicated by the green and red solid lines, respectively.

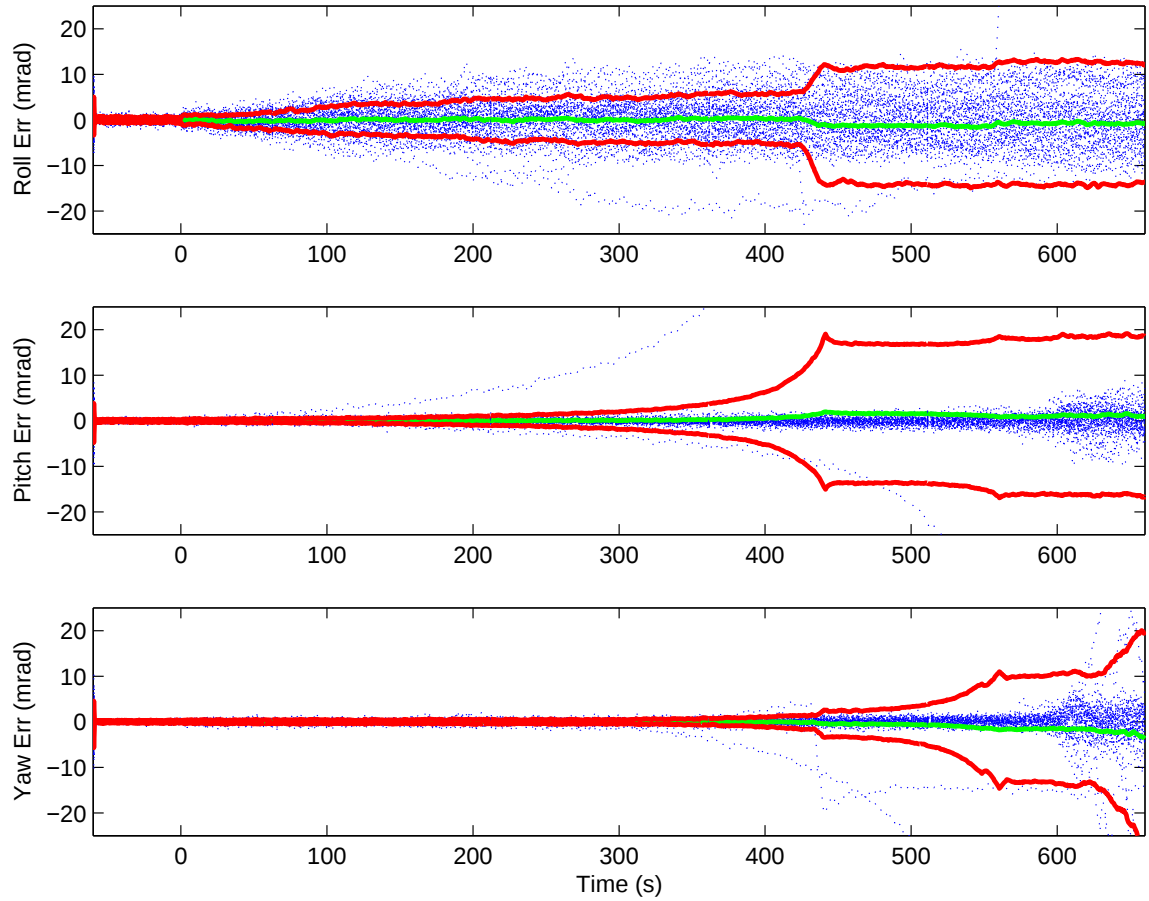


Figure 6.12: Simulated 60-run Monte Carlo attitude error results for indoor profile with a consumer-grade inertial sensor and image aiding using landmarks of opportunity. The attitude error sample functions are indicated by blue dotted lines. The ensemble mean and standard deviation are indicated by the green and red solid lines, respectively.

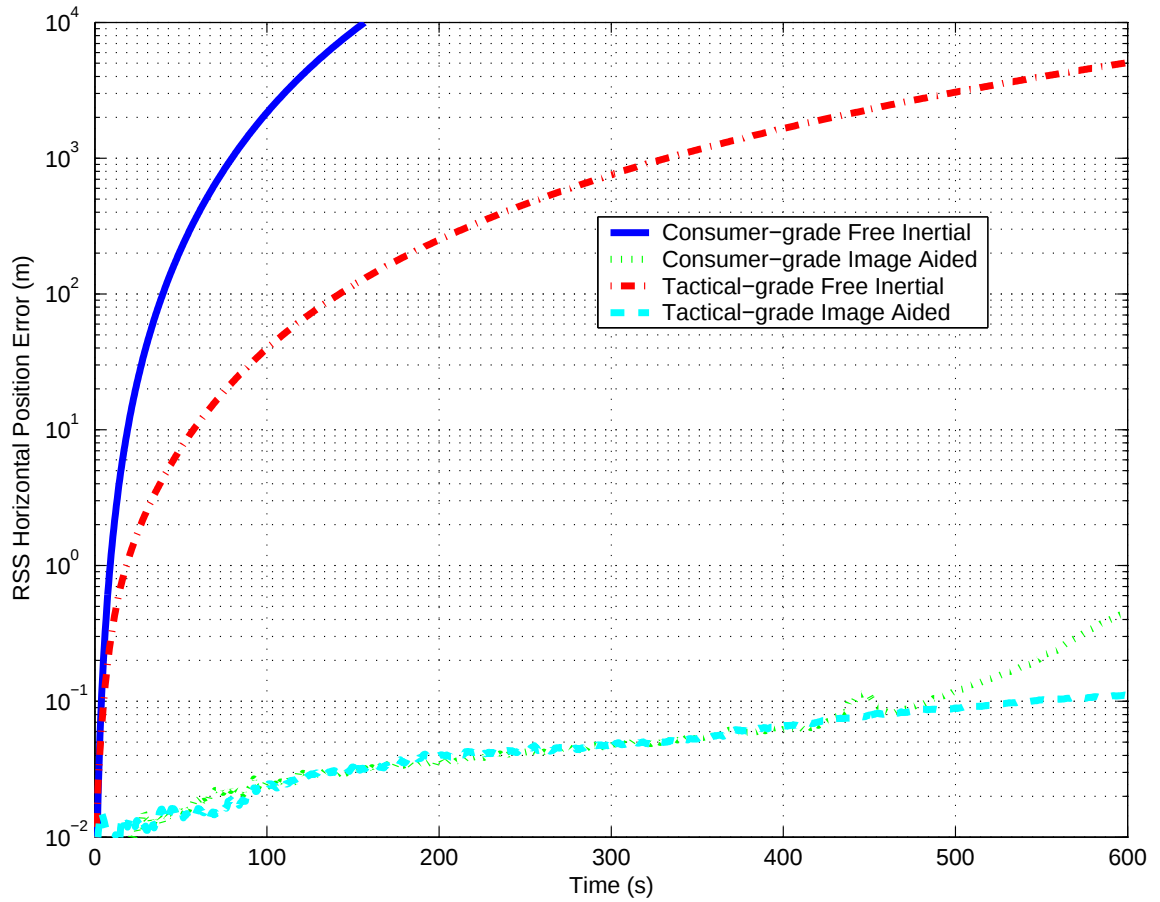


Figure 6.13: Simulated 60-run Monte Carlo root-sum-squared (RSS) horizontal position error for indoor profile using both consumer-grade and tactical-grade inertial sensors. The results are shown for four cases: 1) consumer-grade free inertial, 2) consumer-grade opportunity landmark tracking, 3) tactical-grade free inertial, and 4) tactical-grade opportunity landmark tracking.

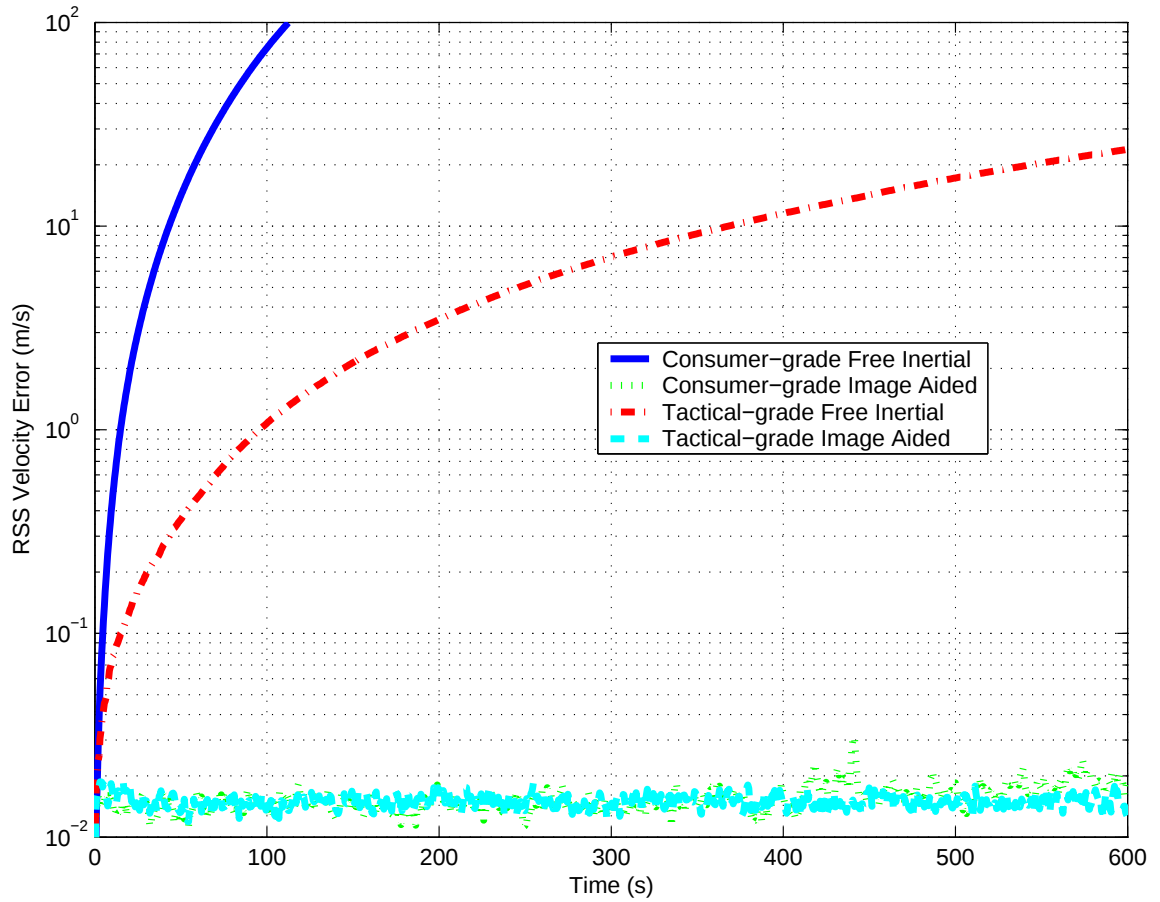


Figure 6.14: Simulated 60-run Monte Carlo root-sum-squared (RSS) velocity error for indoor profile using both consumer-grade and tactical-grade inertial sensors. The results are shown for four cases: 1) consumer-grade free inertial, 2) consumer-grade opportunity landmark tracking, 3) tactical-grade free inertial, and 4) tactical-grade opportunity landmark tracking.

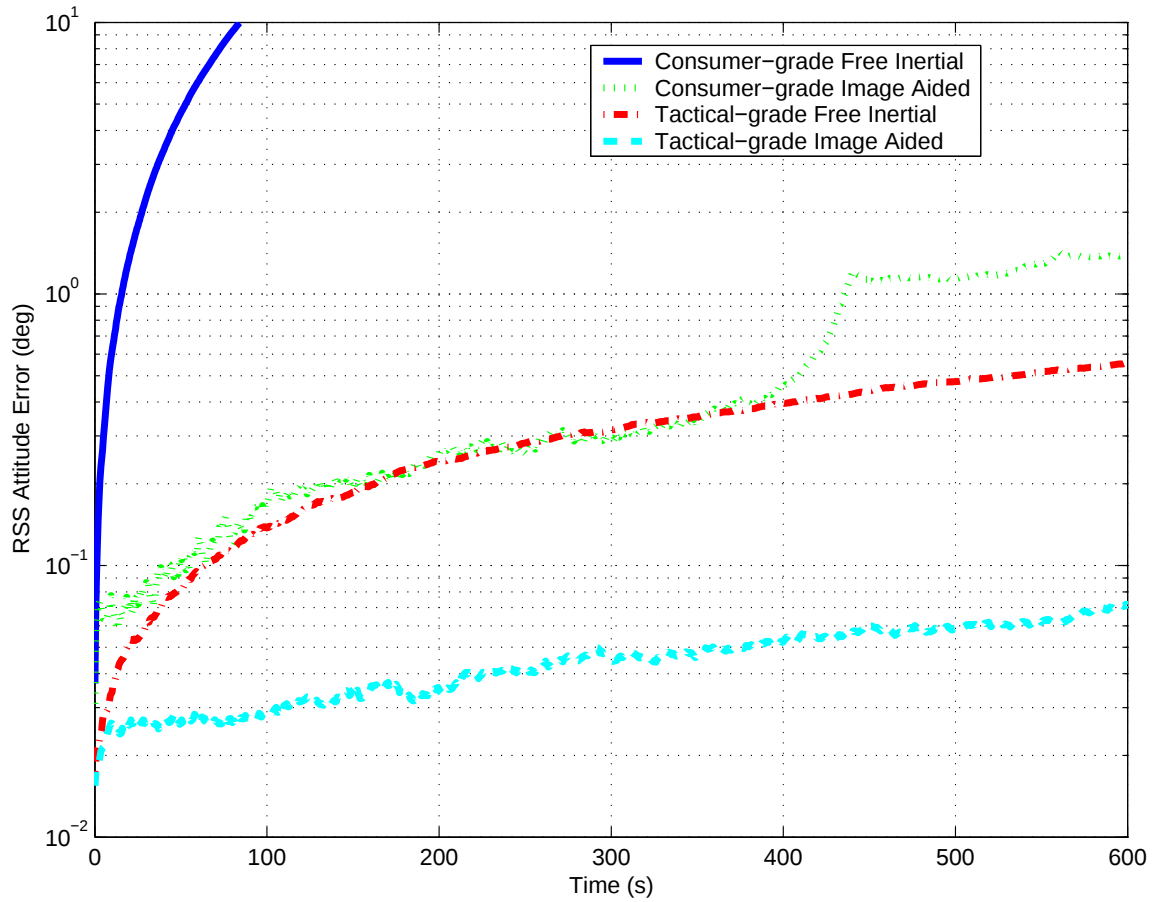


Figure 6.15: Simulated 60-run Monte Carlo root-sum-squared (RSS) attitude error for indoor profile using both consumer-grade and tactical-grade inertial sensors. The results are shown for four cases: 1) consumer-grade free inertial, 2) consumer-grade opportunity landmark tracking, 3) tactical-grade free inertial, and 4) tactical-grade opportunity landmark tracking.

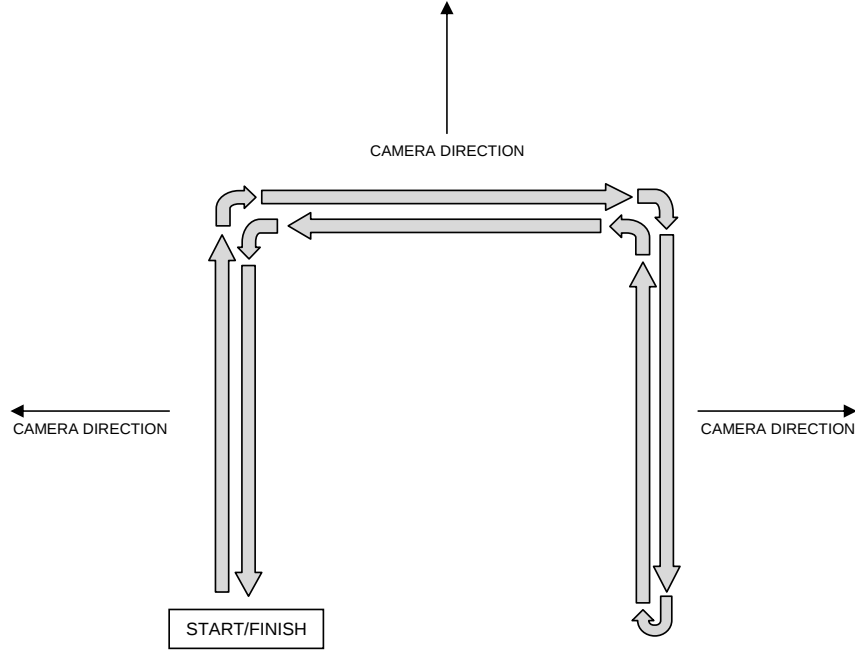


Figure 6.16: Outdoor ground data collection profile. The image-aided inertial navigation sensor is tested by traversing a U-shaped path and returning to the start position.

6.2.2 Experiment. The algorithm is tested experimentally using two ground navigation profiles designed to examine the operation of the feature tracking system in a real-world environment. The first profile consisted of a closed path over an outdoor parking area, shown in Figure 6.16. The outdoor path was traversed forward and backwards with the camera pointed toward the outside of the path. This trajectory resulted in seven segments that presented a scene change and forced the filter to search for new features. This outdoor scene consisted of a combination of man-made features (buildings, fences, roads, etc.) and natural features such as grass and trees. The profile began with a 10-minute stationary alignment period, followed by four minutes of navigation using only images and inertial measurements. No prior knowledge was used with any feature. The filter was limited to a maximum of ten features at any time. A sample image from the outdoor profile is shown in Figure 6.17.

This profile presents the algorithm with a challenging feature tracking environment due to the high-contrast lighting conditions, large variation in feature distance



Figure 6.17: Sample image from outdoor data collection. The outdoor data collection presents the filter with a combination of man-made and natural features. The crosses and ellipses indicate the locations and uncertainty of currently tracked landmarks.

(zero to infinite), and complicated images with semi-transparent objects overlapping at different ranges (e.g., multiple layers of tree limbs).

The filter successfully utilizes inertial measurements to predict and constrain the image correspondence search during the entire profile. In return, the feature correspondence updates and corrects the inertial measurement errors and significantly reduces the resulting drift in the navigation solution. Over the four-minute non-stationary profile, the navigation errors are estimated to be less than 1 meter in the horizontal plane and less than 3 meters in the vertical, when the image-aided inertial algorithm is enabled. Typical free-inertial performance for this inertial sensor is on the order of thousands of meters horizontal error. In addition, the unstable nature of the vertical channel during free-inertial navigation requires external aiding in order to maintain stability [64]. These initial results clearly show the benefits of the image-aided inertial navigation method.

The second profile consisted of a closed path in an indoor environment. The path began and ended at the same location and orientation in the Advanced Navigation Technology (ANT) Center laboratory, at the Air Force Institute of Technology. As in the previous profile, the data collection began with a 10-minute stationary alignment period. After the alignment period, the sensor was moved in a 10-minute loop around the hallways of the building. In contrast to the previous profile, the sensor was pointed primarily in the direction of travel. No prior knowledge was provided to the algorithm regarding the location of features or structure of the environment. A sample image from the indoor profile is shown in Figure 6.18.

The indoor profile presents the algorithm with different challenges from a feature tracking perspective. The indoor environment consists of repetitive, visually identical features (e.g., floor tiles, lights, etc.), which can easily cause confusion for the feature tracking algorithm. In addition, reflections from windows and other shiny surfaces might not be interpreted properly by the filter and could potentially result in navigation errors. Finally, the lower light intensity levels and large areas with poor



Figure 6.18: Sample image from indoor data collection. The indoor data collection presents the filter with man-made features in an office environment. The crosses and ellipses indicate the locations and uncertainty of currently tracked landmarks.

contrast (e.g., smooth, featureless walls) presents a relatively stark feature space. The indoor profile is performed twice for both the tactical and navigation-grade sensors.

As with the previous profile, the filter performs well. The filters' estimates of the trajectories are overlayed on a floor plan of the building in Figures 6.19 and 6.20 for the tactical and consumer-grade inertial sensors, respectively. In each figure, a comparison is made between the fused image-aided inertial trajectory estimate, the image-aided inertial trajectory with stochastic constraints disabled, and a free inertial trajectory. For both tactical and consumer-grade sensors, the estimated trajectory generally corresponds to the building's hallways, with excursions of less than 3 meters. In addition, the results of the free-inertial trajectories show the inherent lack of accuracy of the inertial sensor. With stochastic constraints purposely disabled, the trajectory estimates show relatively large trajectory errors due to false correspondence matches. This illustrates the catastrophic effects of incorporating false updates into an Extended Kalman Filter with inertial feedback and demonstrates the inher-

ent strength of applying robust correspondence methods, and in particular stochastic constraints.

The filter’s estimated trajectory for the image-aided consumer-grade sensor (run two) is examined in more detail in Figure 6.23, where the estimated location of landmarks used for tracking are highlighted. Note the landmarks correspond to the building walls, ceilings, and floors. More detail of the start/stop area is shown in Figure 6.22. A comparison of all image-aided inertial navigation results for both the tactical and consumer-grade sensors is shown in Figure 6.21. The difference in the estimated start and stop locations shows the accumulated errors in the filter over the path. Over the 10-minute profile, the path closure errors are less than 5 *m* in the horizontal plane and less than 5 *m* in the vertical for all sensors. Again, this is a significant improvement over the free-inertial performance. The results are particularly impressive as the solution is calculated using an online algorithm, with only raw inertial and image data. No *a-priori* knowledge of the environment is provided to the system.

6.3 Image and Inertial Fusion Algorithm Flight Tests

In addition to the ground tests, the image and inertial fusion algorithm is evaluated using a flight profile. The flight environment presents a different set of challenges to the system than the indoor environment. First, the dynamic range from the sensor to landmarks is large, which limits the usefulness of binocular ranging. While binocular measurements can provide reliable distance estimates for close landmarks, they are ineffective at longer distances. Fortunately, accurate terrain models are available to a high level of fidelity. Incorporating the terrain elevation into the landmark location estimation eliminates the need for binocular ranging at higher altitudes. Secondly, the flight environment would typically imply longer distance navigation than indoor conditions. Finally, due to the worldwide availability of high-quality satellite imagery, there is an existing set of reference features which could be exploited in the long-distance navigation problem.

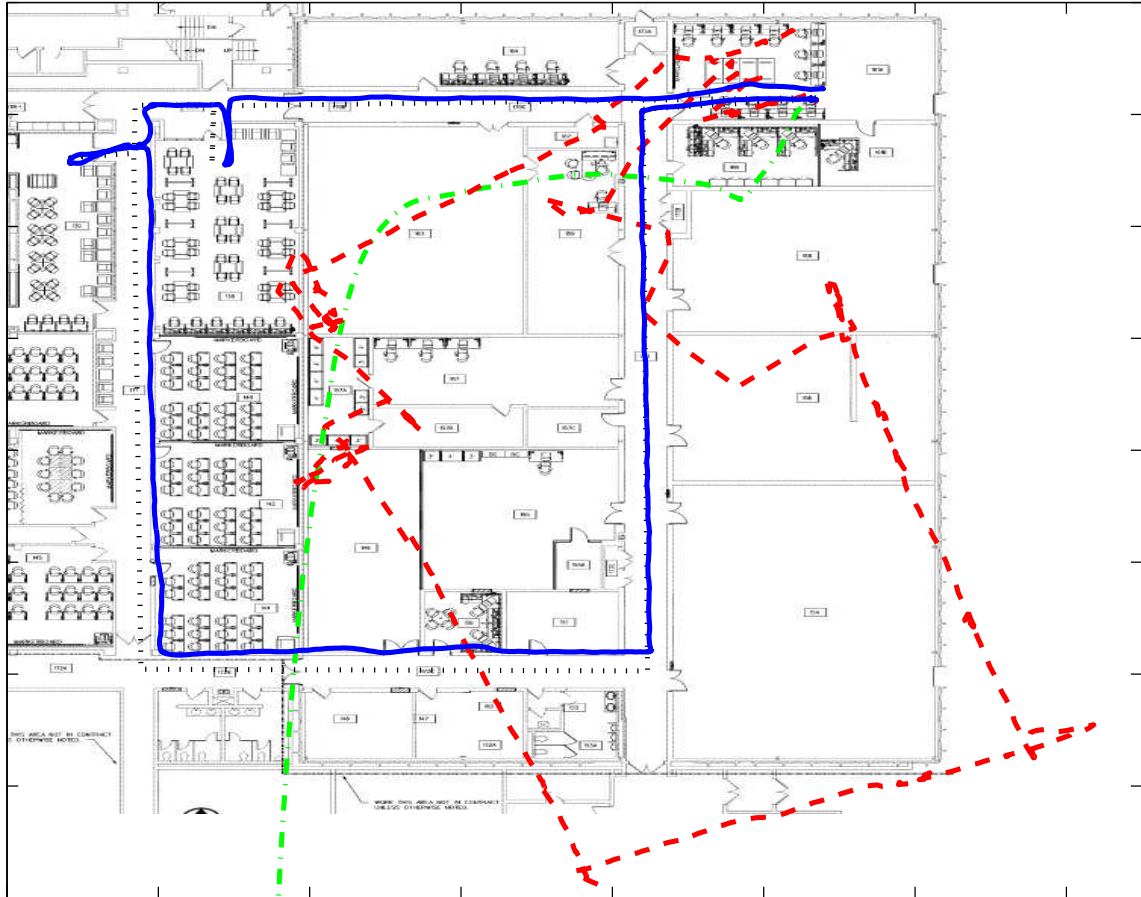


Figure 6.19: Estimated path from indoor data collection (run two) using tactical-grade inertial sensor. The filter's estimate of the path (indicated by the solid line) agrees well with the known path (indicated by the dotted line). The inertial-only best estimate of trajectory (indicated by the dash-dotted line) and image-aided inertial with stochastic constraints disabled (indicated by the dashed line) show large errors in position and heading. The inertial-only solution exceeds the scale of the image after 156 seconds.

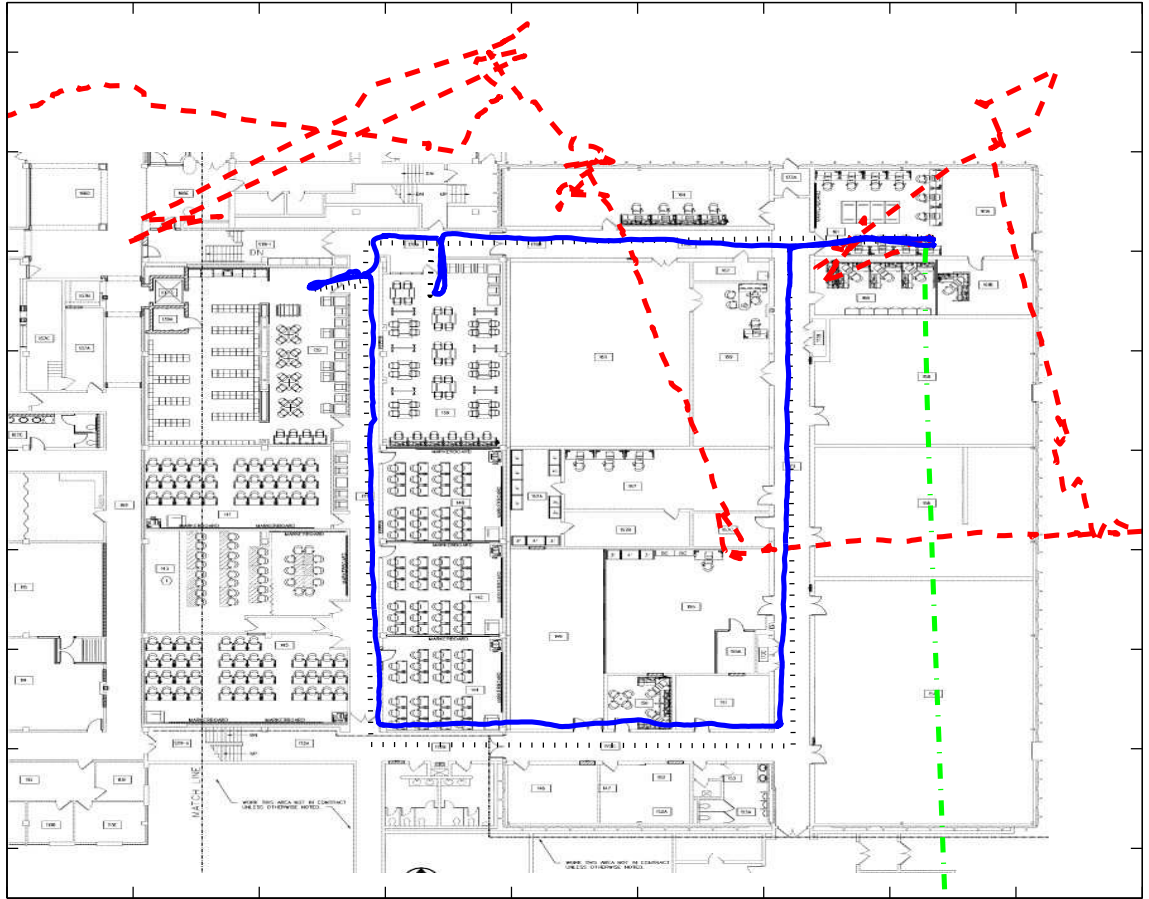


Figure 6.20: Estimated path from indoor data collection (run two) using consumer-grade inertial sensor. The filter’s estimate of the path (indicated by the solid line) agrees well with the known path (indicated by the dotted line). The inertial-only best estimate of trajectory (indicated by the dash-dotted line) and image-aided inertial with stochastic constraints disabled (indicated by the dashed line) show large errors in position and heading. The inertial-only solution exceeds the scale of the image after 11 seconds.

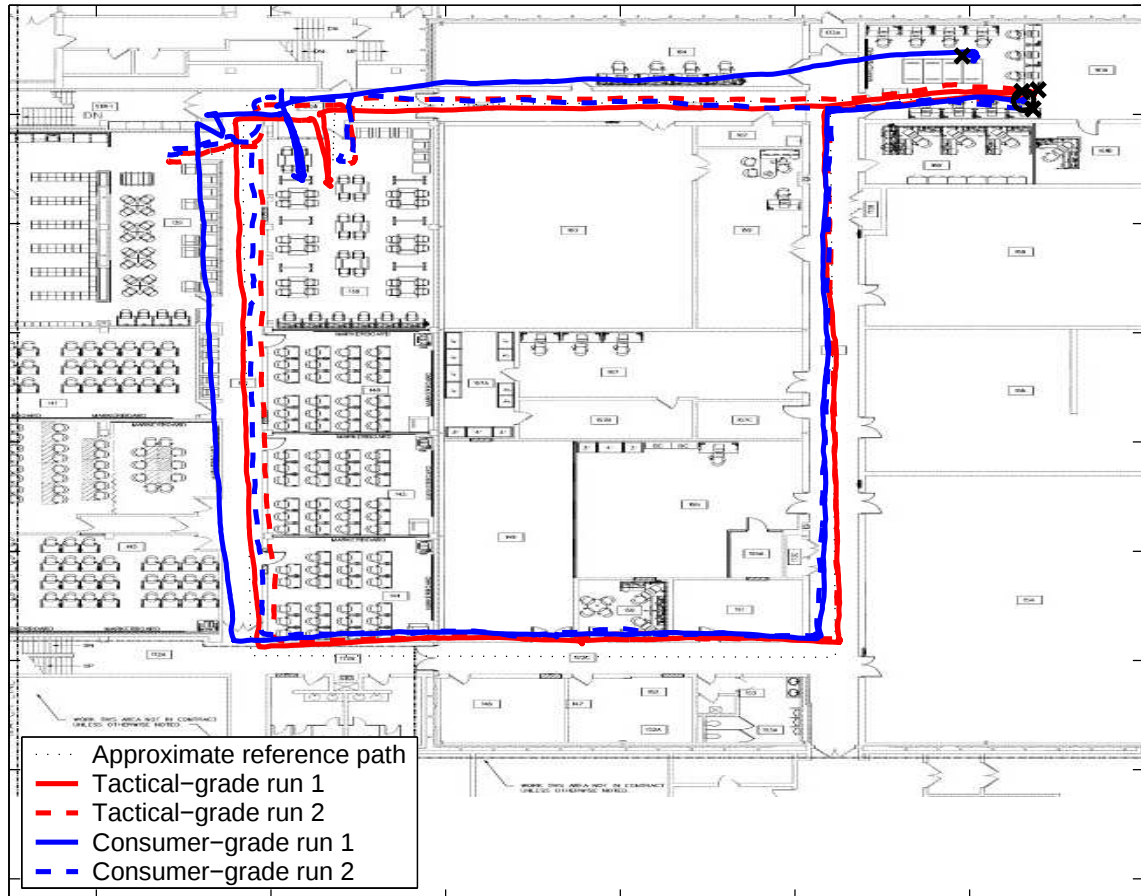


Figure 6.21: Performance comparison for image-aided tactical and consumer-grade inertial sensors for indoor profile runs one and two. The common start and stop location for both runs is indicated by the “o” symbol. The estimated stop location for each run is indicated by an “x” symbol.

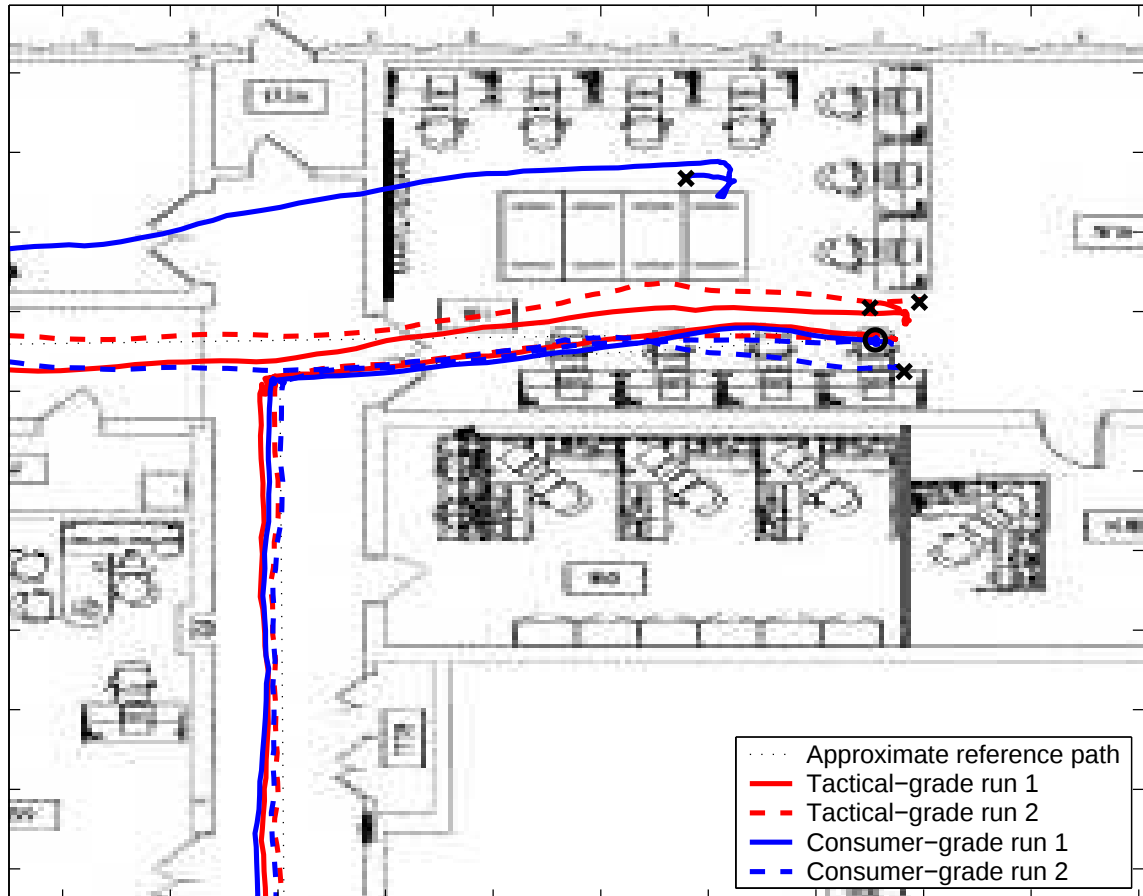
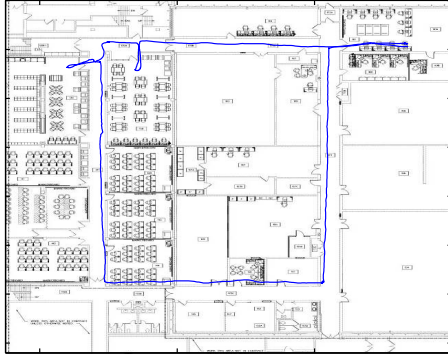
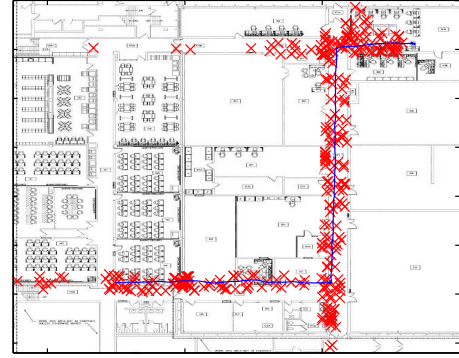


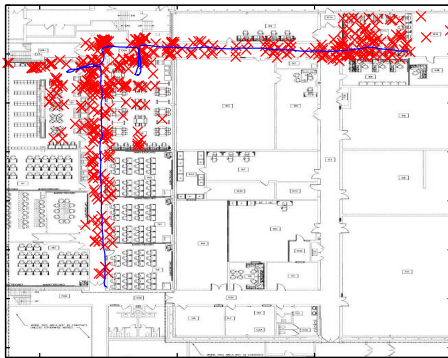
Figure 6.22: Enhanced detail of the start/stop area illustrating the estimated trajectory and feature locations for both the image-aided tactical and consumer-grade inertial sensors for indoor profile runs one and two. The difference between the estimated start and stop location illustrates the accumulated position error.



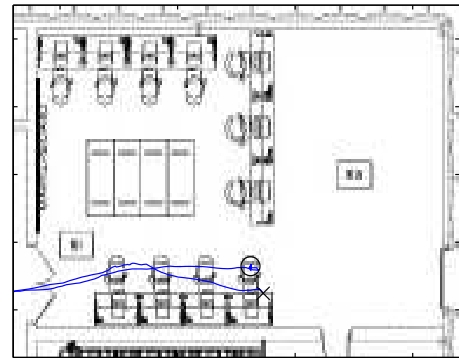
a) Entire path estimate



b) First half of path with tracked feature locations



c) Second half of path with tracked feature locations



d) Start/stop area detail

Figure 6.23: Estimated path and feature locations from indoor data collection (run two) for consumer-grade inertial sensor. Pane (a) shows the entire path estimate. Pane (b) shows the first half of the path along with the estimated location of the features (indicated by “x” symbols). Pane (c) shows the last half of the path and estimated feature locations. Pane (d) provides detail of the start/stop area.

Although experimental flight data was collected, poor-quality truth data were collected due to installation issues with the GPS antenna. This poor GPS observability resulted in a low-quality sensor calibration (see Section 5.2.2). As such, the resulting image-aided inertial navigation solution is difficult to evaluate. Additional flight data collections are required with reliable navigation and ground truth for an accurate system analysis.

As a result, the imaging and inertial fusion navigation algorithm is evaluated using a simulated flight profile. The profile is designed to evaluate the performance of the system in a flight environment using various levels of prior knowledge regarding the scene. The simulation setup and results are presented in the next section.

6.3.1 Simulation. Three simulation profiles are tested to evaluate the performance of the system with varying levels of prior scene knowledge. These knowledge levels represent three cases. Case one is that of navigation over terrain with unknown optical features, which is similar to the ground navigation problem presented in the previous section, with the exception that a statistical terrain model is used to estimate the feature distance versus binocular stereopsis. Case two evaluates the capability of the algorithm to recall previously estimated landmark locations and use them to update the navigation system. This would be representative of a loitering pattern over unknown territory. Finally, the algorithm is evaluated for the case where registered imagery (and the resulting registered landmark set) are available.

In order to match the experimental data collection, the simulation trajectory is a 14-*km*, oval holding pattern, flown at 1000 feet above ground level (AGL). A groundspeed of 50 *m/s* was flown, which results in a 10-minute pattern. The terrain is modeled as relatively flat with a 5-meter standard deviation. The camera is modeled after the flight system, with a 4.8-*mm* lens and angled at 45 degrees toward the ground. Both the H1700 tactical-grade and Crista consumer-grade IMUs are evaluated in the simulation. Each profile begins with a 60 second in-flight alignment, during a straight and level portion of the trajectory. The filter is limited to 12 concurrent landmark

tracks. A total of 30 simulated runs are conducted with randomized image and inertial measurements.

The tactical-grade IMU errors are shown in Figures 6.24-6.26. Four profiles are simulated. The first establishes a baseline of performance for the tactical-grade IMU with only barometric altimeter updates and no image aiding. As expected, the free inertial horizontal position error diverges quickly. Enabling the image aiding using landmarks of opportunity shows an improvement in position error. After 1200 seconds and two laps of the holding pattern, the image-aided horizontal root-sum-squared (RSS) error is less than 100 meters. The potential benefits of leveraging previously-learned landmarks can be seen at 620 seconds into the profile, where the second pass of the holding pattern begins. The navigation errors are reduced due to the relatively high-quality landmark location estimates available from the initial portion of the first pass. Finally, enabling the use of reference landmarks with known locations shows a consistent, meter-level, horizontal RSS error, which indicates the high-level of navigation potential available from an image-aided inertial navigation system combined with reference images.

The Monte Carlo simulation is repeated using the Crista inertial sensor model. The results are shown in Figures 6.27-6.29. Due to the relatively low-quality of this sensor, the free inertial (with barometric altimeter aiding) solution quickly diverges and is not useful for navigation for more than approximately 20 seconds. Enabling image-aiding using landmarks of opportunity shows significant improvement in the navigation performance while also revealing a significant source of error in the extended Kalman filter, which is amplified by the low-quality inertial measurements. As the filter errors increase, two issues occur which violate the assumptions of the extended Kalman filter: 1) the nonlinear dynamics were developed assuming small errors, and 2) the landmark measurement influence matrices are calculated assuming small errors. Both issues contribute to filter instability as the errors in the navigation state are not well represented by the state error covariance maintained by the

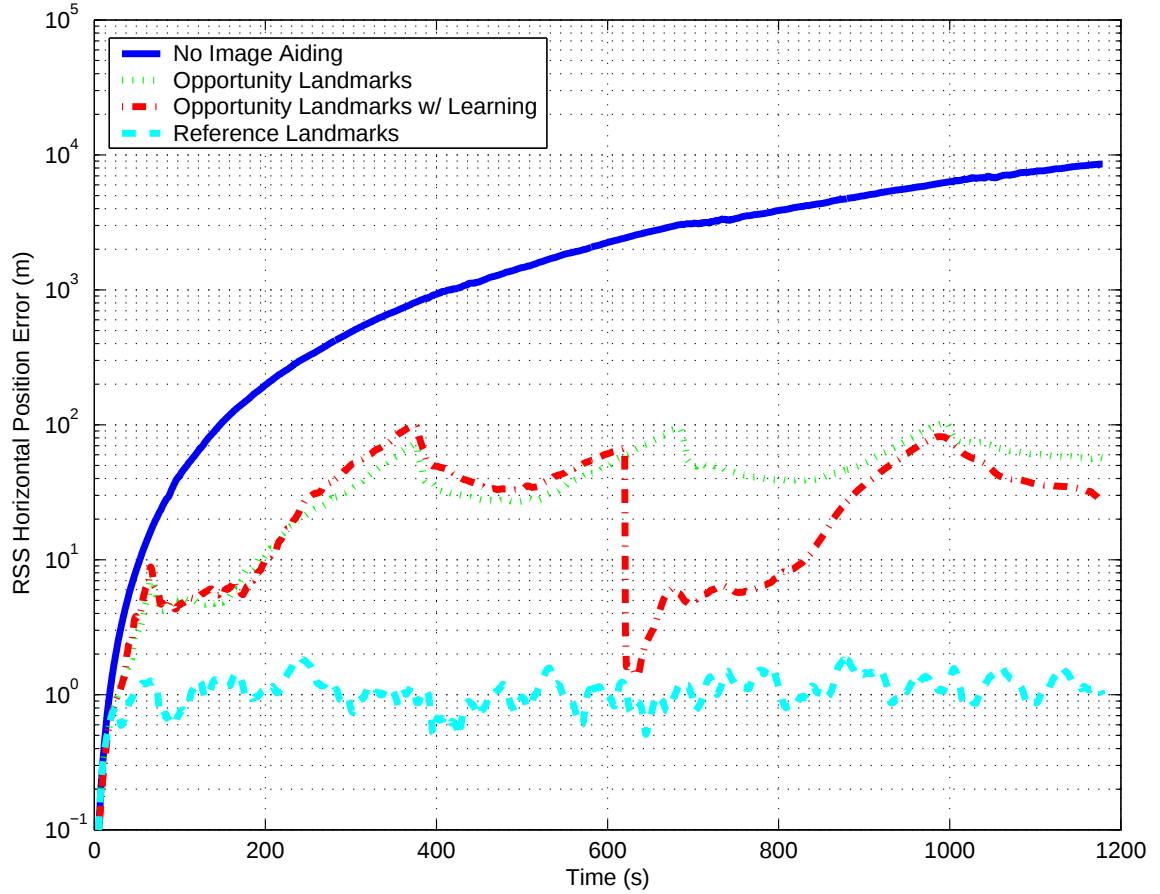


Figure 6.24: Simulated 30-run Monte Carlo root-sum-squared (RSS) horizontal position error for flight profile using a tactical-grade inertial sensor. The results are shown for four cases: 1) image updates disabled, 2) opportunity landmark tracking with terrain model, 3) opportunity landmark tracking with self-learning, and 4) landmark with known location tracking.

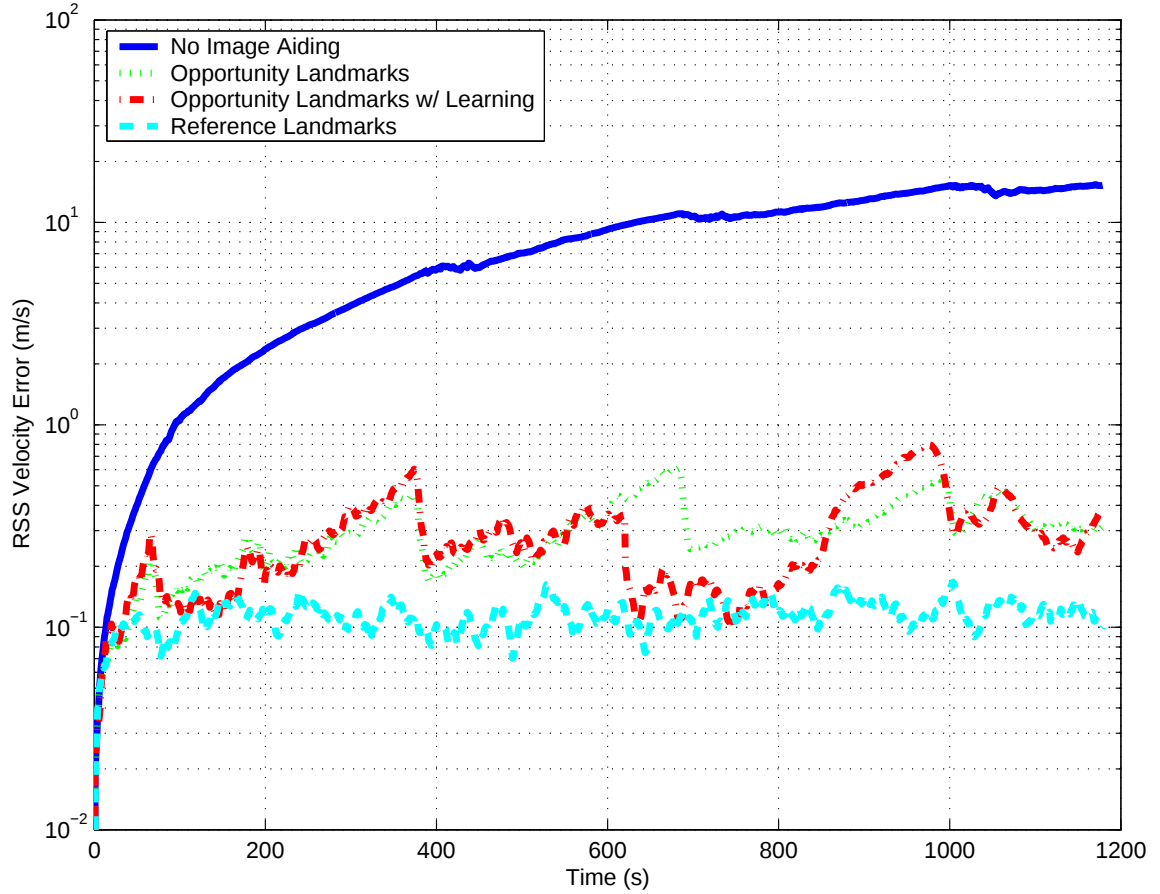


Figure 6.25: Simulated root-sum-squared (RSS) velocity error for flight profile using a tactical-grade inertial sensor. The results are shown for four cases: 1) image updates disabled, 2) opportunity landmark tracking with terrain model, 3) opportunity landmark tracking with self-learning, and 4) landmark with known location tracking. The errors were calculated using a 30-run Monte Carlo analysis.

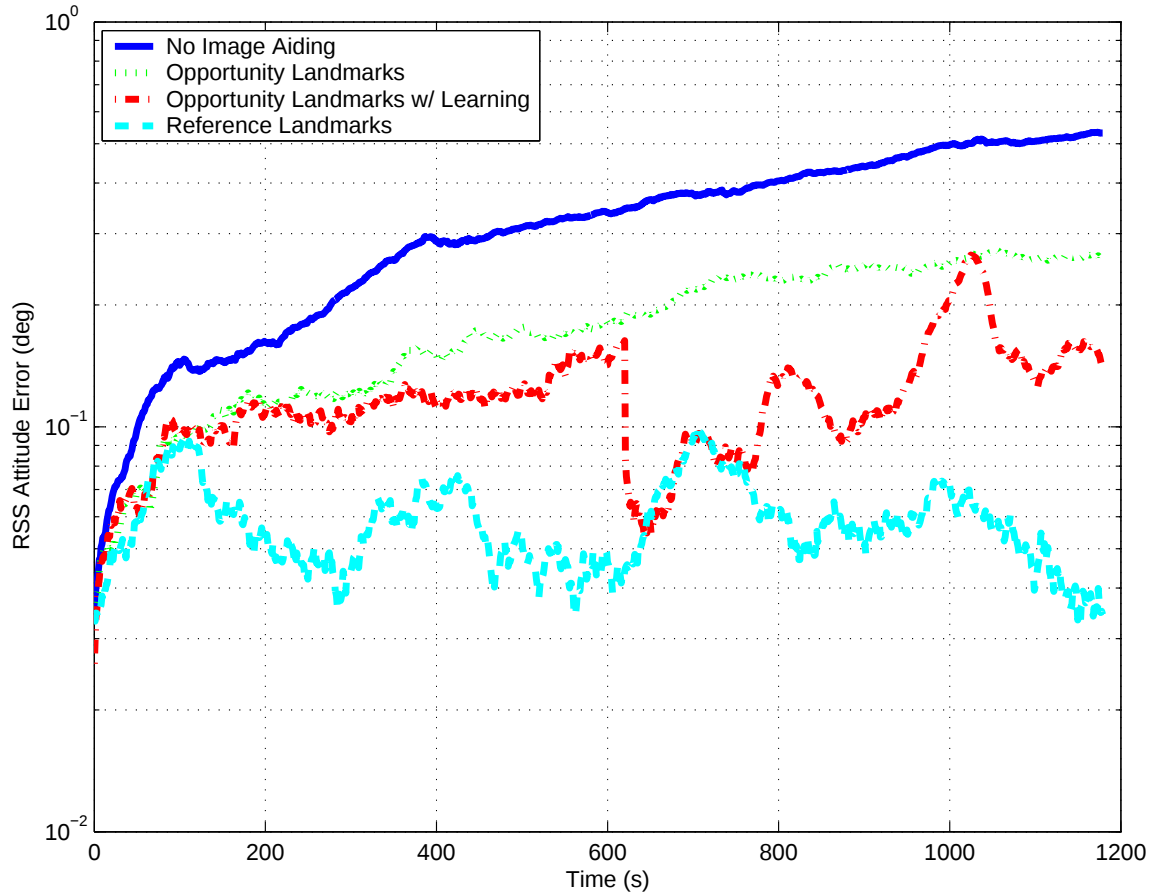


Figure 6.26: Simulated root-sum-squared (RSS) attitude error for flight profile using a tactical-grade inertial sensor. The results are shown for four cases: 1) image updates disabled, 2) opportunity landmark tracking with terrain model, 3) opportunity landmark tracking with self-learning, and 4) landmark with known location tracking. The errors were calculated using a 30-run Monte Carlo analysis.

filter [38]. In this simulation, the filter would quickly diverge after approximately 500 seconds.

In order to investigate this issue further, the Monte Carlo simulation is modified such that the linearization errors for image updates are artificially removed. All other error sources are enabled. Referencing Figures 6.27-6.29, shows significantly improved performance, which indicates that the image-update influence matrix calculation is the likely culprit. This being the case, particular attention must be paid to addressing these fundamental error sources, especially when large errors are expected.

The final Monte Carlo examines the performance when enabling reference landmarks with known locations. As with the tactical-grade simulation, the filter shows a consistent, meter-level, horizontal RSS error, which indicates the high-level of navigation potential available from an image-aided inertial navigation system combined with reference images, even with a low-quality inertial reference. In this scenario, the reference landmark updates dominate the performance of the inertial sensor, which, in essence, causes the filter to use the inertial sensor primarily to statistically constrain the correspondence search. Once a successful correspondence is determined, the filter weights the reference landmark update strongly, as expected, which is indicated by the comparable performance between the tactical and consumer grade sensors using reference landmark updates.

In the next chapter, conclusions are drawn regarding the performance of the system and contributions to the state of the science of navigation are highlighted.

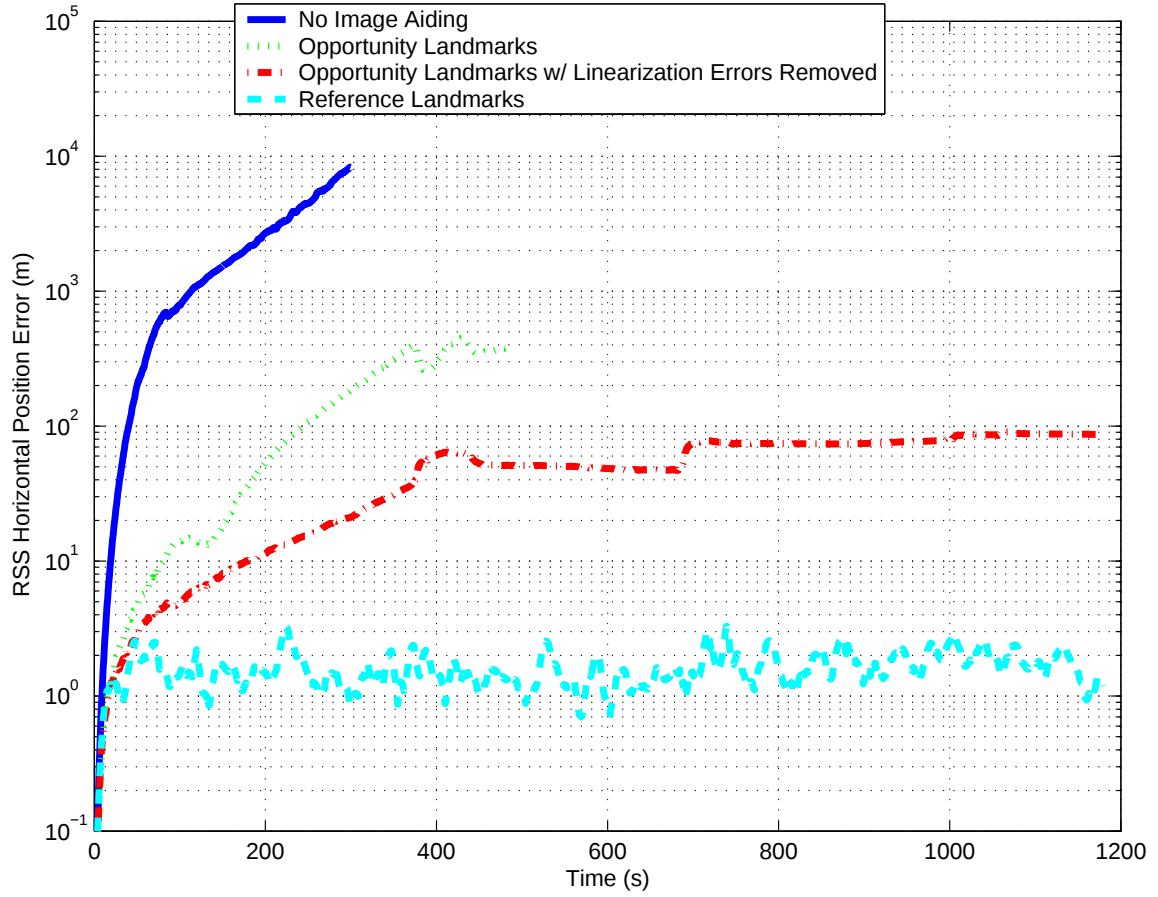


Figure 6.27: Simulated root-sum-squared (RSS) horizontal position error for flight profile using a consumer-grade inertial sensor. The results are shown for four cases: 1) image updates disabled, 2) opportunity landmark tracking with terrain model, 3) opportunity landmark tracking with linearization error removal, and 4) landmark with known location tracking. The errors were calculated using a 30-run Monte Carlo analysis. The landmark of opportunity simulated results were truncated prior to impending filter divergence due to violations of the filter assumptions due to large errors.

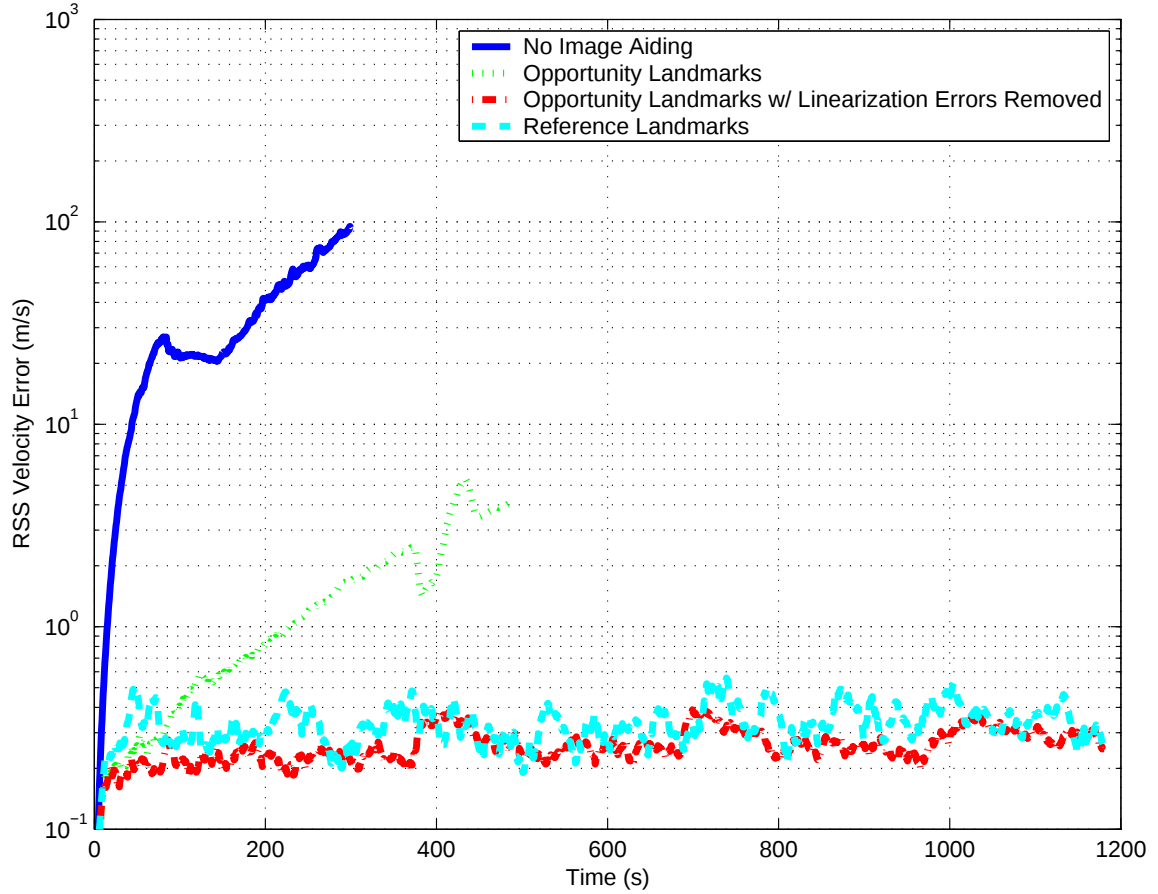


Figure 6.28: Simulated root-sum-squared (RSS) velocity error for flight profile using a consumer-grade inertial sensor. The results are shown for four cases: 1) image updates disabled, 2) opportunity landmark tracking with terrain model, 3) opportunity landmark tracking with linearization error removal, and 4) landmark with known location tracking. The errors were calculated using a 30-run Monte Carlo analysis. The landmark of opportunity simulated results were truncated prior to impending filter divergence due to violations of the filter assumptions due to large errors.

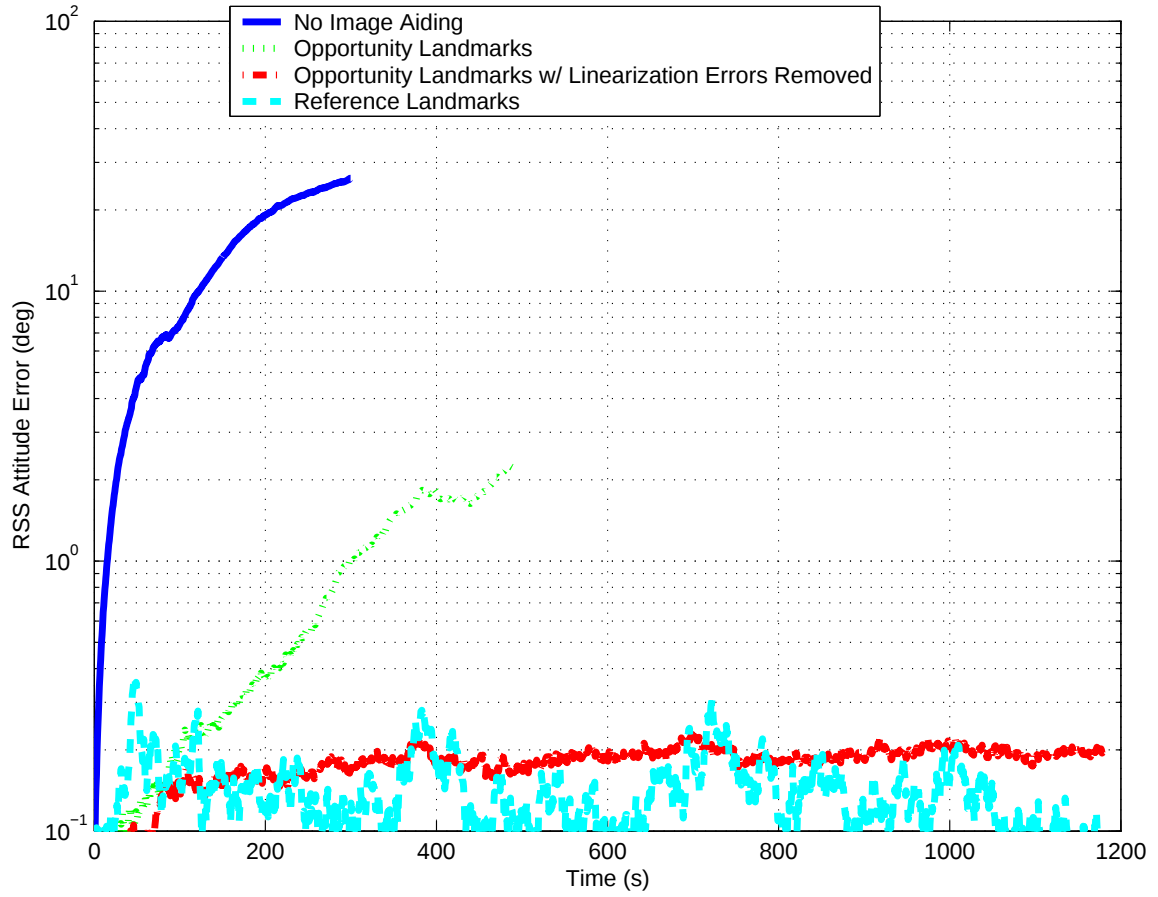


Figure 6.29: Simulated root-sum-squared (RSS) attitude error for flight profile using a consumer-grade inertial sensor. The results are shown for four cases: 1) image updates disabled, 2) opportunity landmark tracking with terrain model, 3) opportunity landmark tracking with linearization error removal, and 4) landmark with known location tracking. The errors were calculated using a 30-run Monte Carlo analysis. The landmark of opportunity simulated results were truncated prior to impending filter divergence due to violations of the filter assumptions due to large errors.

VII. Conclusions

This dissertation presents an approach for fusing optical and inertial sensors for robust, self-contained, passive, autonomous navigation. In this chapter, conclusions regarding the research effort are presented and discussed. In addition, potential areas for future research are addressed.

7.1 *Conclusions*

The most significant conclusion for this work is the demonstration of the viability of fusing imaging and inertial sensors for navigation. As mentioned in Chapter I, the goal for this research is to develop a method for integrating the sensors at a deeper level than the current state-of-the art. The technique is incorporated into an automated image-aided inertial navigation system and tested using simulation and experimental methods.

The novel sensor fusion technique is created by defining the feature space and developing the mathematics required to perform statistically rigorous transformations in this space to predict the features at a future time. The stochastic projection method successfully leverages the collection of inertial measurements between images to predict the location and uncertainty region of features which aids the attendant feature correspondence algorithm. The performance of this algorithm is demonstrated using both simulation and experiments in Chapter IV. In both cases, the experiments confirm the validity and accuracy of the stochastic projections.

The feature transformation algorithm is derived using fundamental stochastic models for various imaging conditions. Using these models, the statistical feature space transformation algorithm is applied to both monocular and multi-sensor designs for unknown, partially-known, and with fully-known information regarding the world. The ability to accurately predict the location of a feature statistically using all available information forms the basis of the image and inertial fusion technique.

In addition to the statistical feature transformation techniques, a new method of calibration of imaging and inertial sensors is presented and tested. The addition of

self-calibration states to the Kalman filter allows an integrated imaging and inertial sensor to self-estimate calibration parameters using known optical references (e.g., celestial bodies and surveyed ground points) and independent measurements of the trajectory, such as GPS. This technique is evaluated using simulation, ground, and flight data. In each experiment, the observability of the self-calibration states is verified and shown to be primarily related to the fundamental accuracy of the sensors. The ability to automatically correct for calibration errors using this technique enables the system to compensate for these errors outside of the laboratory environment.

A key contribution of this work is the development and implementation of the deep-integration theory in an experimental imaging and inertial navigation sensor. The image-aided inertial navigation system is evaluated using a combination of simulation, ground, and flight tests. In each experiment, the image-aided inertial navigation system consistently demonstrates several orders of magnitude improvement over the equivalent unaided inertial system (and even that of navigation-grade sensors). This development holds a great deal of promise to produce a low-cost, navigation-grade optical-inertial sensor for passive environments. In addition, the demonstrated capability to store self-surveyed features and recall them when needed enables this sensor to effectively “learn” an environment and eliminate navigation error drift while remaining in known areas. A number of operationally-realistic scenarios which leverage this revolutionary concept are discussed in the next section.

As expected, the extended Kalman filter navigation algorithm shows the deleterious effects of linearization errors. These errors are most prevalent in the presence of large attitude errors. When the linearization errors are artificially corrected during simulation, the navigation filter performance is consistent with the system model. These results show that the extended Kalman filter algorithm delivers sub-optimal performance and improvement could be realized using an estimator which does not suffer from the linearization errors of the extended Kalman filter. This conclusion shows that additional research into estimation algorithms has promise and should be investigated. Some potential approaches are presented in the next section.

In summary, this research develops a series of techniques to provide autonomous, passive navigation by incorporating image and inertial measurements. The method demonstrates navigation-quality performance with low-cost sensors and passive updates. While the approach demonstrates greatly improved performance over free inertial navigation with equivalent sensors, there are areas which merit additional research. These are addressed in the next section.

7.2 *Future Work*

This research presents a statistically-rigorous method to tightly integrate imaging and inertial sensors. The method is used to construct a navigation system which leverages optical signals of opportunity and does not require external navigation signals. In addition to the system presented in this research, there are many potential areas of research which exploit the natural synergy between optical and inertial sensors.

The first general area of additional research is the development and improvement of feature transformation algorithms. While the SIFT algorithm has demonstrated potential for image-aided navigation applications, there are other feature transformation algorithms which should be investigated. A revised SIFT algorithm has been proposed by Ke [27] which replaces the feature descriptor calculation of SIFT with a principal component analysis (PCA)-based decomposition. The PCA-SIFT algorithm has demonstrated improved feature matching performance over the traditional SIFT algorithm.

Conceptually, the most desirable feature transformation algorithm is one in which the location and object description dimensions are truly orthogonal. In simpler terms, the ideal transformation algorithm would detect and recognize *distinct objects*, in a similar manner to humans. Once an object was recognized, an additional level of navigation information could be extracted from the object, include a direct measurement of orientation and range. Inferring range information from the apparent size of an object with known dimensions is referred to as *stadiametric ranging* and

is commonly used for target ranging when external ranging sources are unavailable. The additional information provided by this type of measurement would theoretically improve the navigation performance of such a system, and could reduce the effects of linearization errors which were evident in the extended Kalman filter algorithm used in this research.

The next level of potential research is to explore a potentially deeper level of integration between inertial and imaging sensors. The concept would entail incorporating inertial measurements at the feature transformation level. In general terms, the approach can be clarified as follows. As discussed in Chapter III, the propagation of the navigation state using inertial measurements is subject to the effects of sensor errors, such that an uncertainty region exists when projecting the location of a feature from one image to the next. This uncertainty region is a function of three primary sources of error: 1) the initial uncertainty of the navigation state, 2) the accuracy of the inertial sensors, and 3) the integration time. Since the uncertainty is actually related to the *relative* navigation error between frames, most of the initial navigation state uncertainty is mitigated when the navigation errors are relatively constrained. Accordingly, if the algorithm could dynamically control the integration time (i.e., image sample rate), the uncertainty region for feature projection between images could be effectively managed. Thus, given a minimum feature size (in the feature space), the image sample rate could be dynamically varied such that the probability of false correspondence would be greatly reduced. Viewing the measured feature space as a sampling of the true feature space reveals parallel concepts between this method of false correspondence elimination and sampling theory. In fact, this proposed technique is a method of feature space anti-aliasing which uses a dynamic filter based on inertial measurements.

In the proposed image-aided navigation filter described in this research, the feature extraction calculations consumed the majority of computer processing time. In order to employ such a system in real-time required optimization of the attendant feature transformation algorithm. Two paths for improving the speed of these calcula-

tions are proposed. First, the feature transformation algorithm could be accelerated by exploiting parallel processing and pipelining. As shown in the SIFT algorithm, feature extraction can be accomplished using separable scale-space decompositions, which is ideal for parallel processing. The next improvement would constrain the feature extraction algorithm to the statistical search region predicted using the stochastic projection method developed in this research. By limiting the feature transformation calculation to a much smaller area, significant speed improvements could be realized.

Another area of promising research would investigate potential of other nonlinear filtering approaches. As mentioned in Chapter VI, the extended Kalman filter is sensitive to linearization errors, which are unmodeled. Although beyond the scope of this research, additional estimation schemes have been proposed which address these issues. Some examples are iterated extended Kalman filters, unscented particle filters, and multiple model extended Kalman filters. In addition, there is potential for truly nonlinear estimation sensors such as those found in neural networks.

As mentioned previously, there exists a natural synergy between imaging and inertial systems. It is evident based on this research that construction of a fused optical and inertial sensor would provide a standard, consistent, and affordable platform for image-aided inertial navigation. With the recent advances in MEMS and CMOS imaging technology, an image and inertial sensor integrated (and possible feature transformation circuits) on a single chip could open the doors for widespread usage. In addition to ensuring excellent time-synchronization between the image and inertial sensors, the single chip solution would virtually eliminate the need for field calibration, and could result in a very economical, and robust sensor.

In addition to the benefits in cost and performance, integrating the imaging and inertial sensor at the production level creates a “standard” image and inertial navigation sensor which can benefit enormously from multi-sensor designs. Scenarios which would greatly benefit from this scaling effect are classified as either single or multi-ship implementations. Mounting multiple integrated sensors on a single vehicle

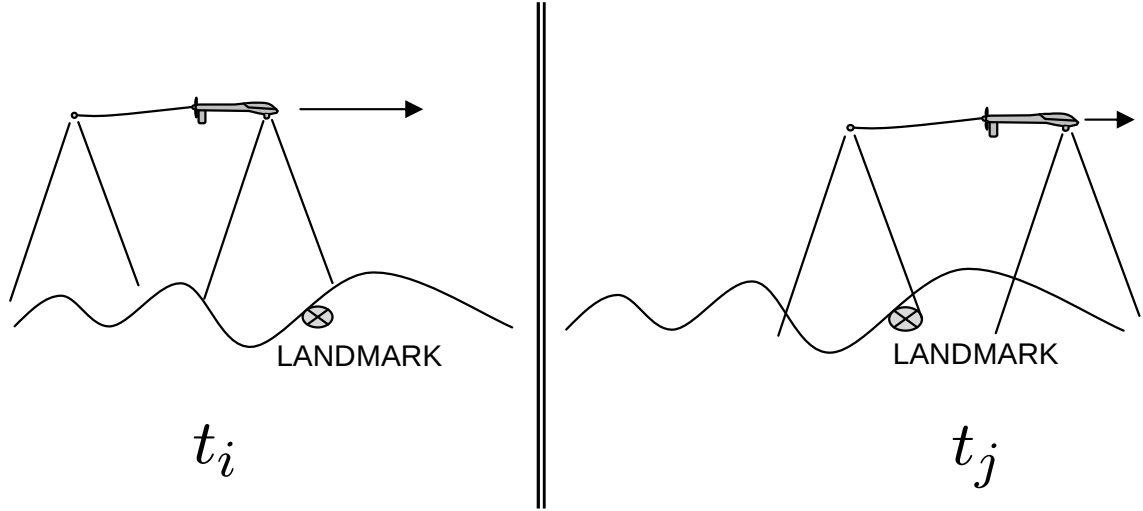


Figure 7.1: Multi-sensor imaging scenario from solo vehicle. Mounting separate integrated image and inertial sensors at the nose and from a towed sensor array would provide a long binocular disparity which could be used for accurate range estimation, even when targets are not simultaneously in the field of view.

would provide either a direct (single-epoch) or time-delayed stereoscopic measurement, which would help to reduce the range estimation error. As shown in Figure 7.1, mounting separate integrated sensors at the nose and tail (or towed sensor array) of an aircraft would benefit from a relatively long binocular disparity, resulting in accurate range estimation, even at higher altitudes.

Multi-ship implementations of the fused image and inertial navigation sensor offer even more exciting possibilities. The demonstrated ability of the sensor algorithm to “remember” features would allow each ship in the formation to effectively share observations of the same landmark(s). This would not only allow the formation to accurately navigate relative to the world, but would provide accurate location information for landmarks of interest in both the navigation reference frame and relative to each member of the group. In addition, because the navigation solution for each sensor is processed in parallel, scaling effects could be minimized. Finally, if ship-to-ship ranging information were available (e.g., based on measurements from an existing datalink), the performance of such a system could be extremely accurate and resistant to long-term navigation error growth, especially if the formation geom-

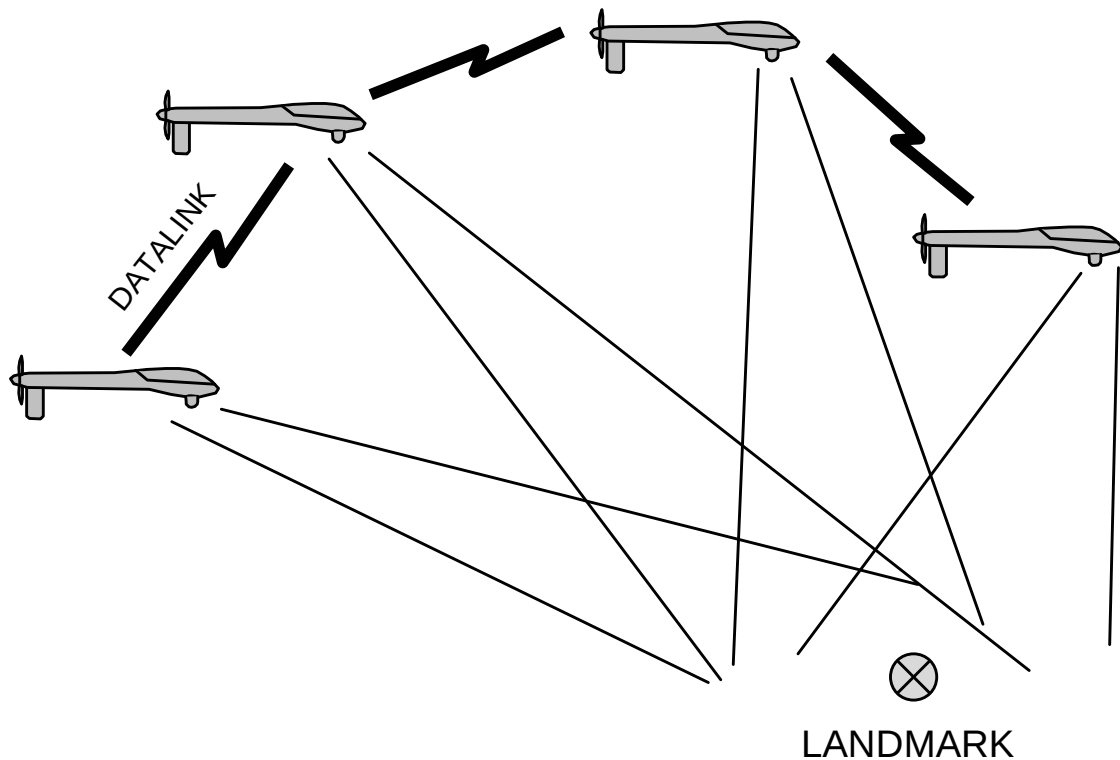


Figure 7.2: Multi-vehicle cooperative navigation scenario. Vehicles equipped with integrated image and inertial navigation systems could improve navigation and landmark location accuracy over non-cooperative systems, especially when the formation could be controlled to provided maximum observability.

etry could be controlled to maximize observability. A sample multi-ship, cooperative navigation scenario is shown in Figure 7.2.

As outlined in Chapter IV, exploiting previously recorded, georeferenced images for navigation can provide absolute navigation information, which is critical to counter the unbounded long-term drift associated with landmark of opportunity navigation. Based on simulation analysis, this technique has the accuracy potential to navigate for autonomous landing or for other situations where GPS may not be available. The most important aspect of all of these techniques is the inherent flexibility to use any available sources to provide the best navigation estimate possible. For example, an image-aided navigation system could leverage landmarks of opportunity when over unknown terrain, incorporate measurements from known landmarks when available,

and recall previously estimated landmarks when returning to base or loitering in a target area.

7.3 *Summary*

As evidenced by nature, there is a natural synergy between imaging and inertial sensors for navigation. The key to this synergy lies in coupling both sensors at the deepest level possible to generate robust navigation estimates under as many imaging conditions as possible. This research presents a step in that direction, however, as discussed in the previous section, there is much work to do. We have only begun to realize the potential of this technology.

Bibliography

1. Adams, Henele, Sanjiv Singh, and Dennis Strelow. “An Empirical Comparison of Methods for Image-based Motion Estimation”. *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, volume 1, 123–128. September 2002.
2. Bedekar, A.S. and R.M. Haralick. “Finding corresponding points based on Bayesian triangulation”. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 61–66. 1996.
3. Bhanu, Bir and Yingqiang Lin. “Genetic algorithm based feature selection for target detection in SAR images”. *Image and Vision Computing*, 2:591–608, 2003.
4. Bhanu, Bir, B. Roberts, and J. Ming. “Inertial navigation sensor integrated motion analysis for obstacle detection”. *Proceedings IEEE International Conference on Robotics and Automation*, 954–959. May 1990.
5. Bowditch, Nathaniel, LL.D. *The American Practical Navigator: 1995 Edition*. Defense Mapping Agency Hydrographic/Topographic Center, Bethesda, MD, 1995.
6. Cloud Cap Technology. “Crista Inertial Measurement Unit (IMU) Interface / Operation Document”. Specification, May 2004. URL: <http://www.coudcaptech.com/>.
7. DMA WGS-84 Development Committee. *Department of Defense World Geodetic System 1984 - Its Definition and Relationships with Local Geodetic Systems*. Technical Report 8350.2, Defense Mapping Agency, Washington DC, Washington, DC, September 1987.
8. Egelhaaf, Martin, Roland Kern, Holger G. Krapp, Jutta Kretzberg, Rafael Kurtz, and Anne-Kathrin Warzecha. “Neural Encoding of Behaviourally Relevant Visual-Motion Information in the Fly”. *Trends in Neurosciences*, 25(2):96–102, February 2002.
9. Einstein, Albert. *Relativity: The special and general theory*. Henry Holt, New York, NY, 1920.
10. Encyclopedia Britannica Premium Service. “Brittanica Student Encyclopedia”. <http://www.britannica.com/ebi/article?tocId=234470>, April 2005.
11. Faugeras, Oliver. *Three-dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
12. Förstner, W. and E. Gülch. “A fast operator for detection and precise locations of distinct points, corners, and centres of circular features.” *Proceedings of the Intercommission Conference on Fast Processing of Photogrammetric Data*, 281–305. 1987.

13. Franz, Matthias O. and Holger G. Krapp. "Wide-field, Motion-sensitive Neurons and Matched Filters for Optic Flow Fields". *Biological Cybernetics*, 83:185–197, 2000.
14. Golden, J.P. "Terrain Contour Matching (TERCOM): A Cruise Missile Guidance Aid". *Proceedings of the International Society for Optical Engineering (SPIE) Image Processing for Missile Guidance*, volume 238, 10–18. 1980.
15. Goodman, Joseph W. *Introduction to Fourier Optics*. McGraw Hill, Boston, Massachusetts, 1996.
16. Green, Robin M. *Spherical Astronomy*. Cambridge University Press, New York, NY, 1985.
17. Hagen, Espen and Eilert Heyerdahl. "Navigation by Optical Flow". *Proceeding of the 11th IAPR International Conference on Pattern Recognition*, volume 1, 700–703. 1992.
18. Harris, C. and M. Stephens. "A Combined Corner and Edge Detector". *Proceedings of the Alvey Conference*, 189–192. 1988.
19. Harrison, Reid R. and Christof Koch. "A Robust Analog VLSI Motion Sensor Based on the Visual System of the Fly". *Autonomous Robots*, 7:211–224, November 1999.
20. Hecht, Eugene. *Optics*. Addison Wesley, San Fransisco, California, 2002.
21. Higgins, Charles M. "Sensory Architectures for Biologically Inspired Autonomous Robots". *Biol. Bull.*, 200:235–242, April 2001.
22. Honeywell International Inc. "HG1700 Inertial Measurement Unit". Datasheet, August 2003.
23. Hrabar, Stefan and Gaurav S. Sukhatme. "A Comparison of Two Camera Configurations for Optic-Flow Based Navigation of a UAV Through Urban Canyons". *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, 2673–2680. September 2004.
24. Jain, Anil K. *Fundamentals of Digital Image Processing*. Prentice Hall, Inc., Upper Saddle River, New Jersey 07458, 1989.
25. Kaplan, G.H., J.A. Hughes, P.K. Siedelmann, C.A. Smith, and B.D. Yallop. "Mean and Apparent Place Computations in the New IAU System. III. Apparent, Topocentric, and Astrometric Places of Planets and Stars". *The Astronomical Journal*, 97:1197–1210, April 1989.
26. Kasper, West, Major. "GPS Vulnerability Testing". *GPS World Magazine*, May, 2004.
27. Ke, Yan and Rahul Sukthankar. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors". *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, volume 2, 506–513. 2004.

28. Keaney, Thomas A. and Eliot A. Cohen. *Gulf War Air Power Survey: Summary Report*. Technical report, Department of the Air Force, Washington, D.C., 1993.
29. Knierim, J.J., W.E. Skaggs, H.S. Kudrimoti, and B.L. McNaughton. "Vestibular and visual cues in navigation: a tale of two cities". *Ann NY Acad Sci*, June:399–406, 1996.
30. Krakiwsky, E.J. *A Synthesis of Recent Advances in the Method of Least Squares*. University of New Brunswick, Fredericton, New Brunswick, Canada, 1975.
31. Krapp, Holger G. and Roland Hengstenberg. "Estimation of Self-Motion by Optic Flow Processing in Single Visual Interneurons". *Nature*, 384:463–466, December 1996.
32. Lerner, R., E. Rivlin, and P. H. Rotstein. "Error Analysis for a Navigation Algorithm based on Optical-Flow and a Digital Terrain Map". *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR04)*, volume 1, I-604–I-610. 2004.
33. Lowe, David G. "Object Recognition from Local Scale-Invariant Features". *Proc. of the International Conference on Computer Vision*, volume 2, 1150–1157. September 1999. Corfu, Greece.
34. Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints". *International Journal of Computer Vision*, 60(2):91–110, 2004.
35. Lucas, B. D. and T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". *Proc. DARPA Image Understanding Workshop*, 121–130, 1981.
36. Ma, Yi, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision*. Springer-Verlag, Inc., New York, New York, 2004.
37. Maybeck, Peter S. *Stochastic Models Estimation and Control, Vol I*. Academic Press, Inc., Orlando, Florida 32887, 1979.
38. Maybeck, Peter S. *Stochastic Models Estimation and Control, Vol II*. Academic Press, Inc., Orlando, Florida 32887, 1979.
39. MIT Instrumentation Laboratory. *Space Navigation Guidance and Control Report*. Technical Report R-500, Massachusetts Institute of Technology, Boston, MA, 1965.
40. Musick, Stan. *Profgen: PC Software for Trajectory Generation*. Software Package v8.19, Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio, January 2004.
41. Olson, Clark F. "Maximum-likelihood Image Matching". *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 24(6):853–857, June 2002.

42. Olson, Clark F., Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. "Robust Stereo Ego-motion for Long Distance Navigation". *Proceedings of the IEEE Conference on Advanced Robotics*, volume 2, 453–458. June 2000.
43. Oxley, Mark E. and Steven N. Thorsen. "Fusion or Integration: What's the difference?" *7th International Conference on Information Fusion (FUSION 2004)*, 429–434. June 2004.
44. Pachter, Meir. "Vision Based Navigation Theory". *Proceedings of the 14th International Conference on Control Systems and Computer Science*. July 2-5 2003.
45. Pachter, Meir, J.J. D'Azzo, and Michael J. Veth. "Proportional and Integral Control of Non-linear Systems". *International Journal of Control*, 64(4):679–692, 1996.
46. Pachter, Meir and Alec Porter. "Bearings-only Measurements for INS Aiding: The Three-Dimensional Case". *Proceedings of the 2003 AIAA Guidance, Navigation and Control Conference*. 2003. AIAA paper number 2003-5354.
47. Pachter, Meir and Alec Porter. "INS Aiding by Tracking an Unknown Ground Object - Theory". *Proceedings of the American Control Conference*, volume 2, 1260–1265. 2003.
48. Papoulis, Athanasios and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, NY 10020, 2002.
49. PixeLINK. "PixeLINK PL-A741 Machine Vision Camera Datasheet". Specification, April 2004. URL: <http://www.pixelink.com/>.
50. Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++ Second Edition*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, United Kingdom, 2002.
51. Raquet, John F. and Michael Giebner. "Navigation Using Optical Measurements of Objects at Unknown Locations". *Proceedings of the 59th Annual Meeting of the Institute of Navigation*, 282–290. June 2003.
52. Roberts, B. and Bir Bhanu. "Inertial navigation sensor integrated motion analysis for autonomous vehicle navigation". *Journal of Robotic Systems*, 9:817–842, 1992.
53. Roumeliotis, S.I., A.E. Johnson, and J.F. Montgomery. "Augmenting inertial navigation with image-based motion estimation". *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, volume 4, 4326–4333. 2002.
54. Shin, Eun-Hwan. *Accuracy Improvement of Low Cost INS/GPS for Land Applications*. Technical Report 20156, University of Calgary, Calgary, Canada, 2001.
55. Sim, D.G., S.Y. Jeong, R.H. Park, R.C. Kim, S.U. Lee, and I.C. Kim. "Navigation parameter estimation from sequential aerial images". *Proceedings of the International Conference on Image Processing*, volume 2, 629–632. 1996.

56. Sim, D.G., R.H. Park, R.C. Kim, S.U. Lee, and I.C. Kim. "Integrated Position Estimation Using Aerial Image Sequences". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1–18, January 2002.
57. Simonsen, Kenneth S., Dean Nathans, Mark L. Suycott, John Wohlfiel, Arch Turner, and Bob Crumplar. "GPS: Our Strongest Asset or Weakest Link?" *CHIPS - the Department of the Navy Information Technology Magazine*, Spring, 2002.
58. Smith, R., M. Self, and P. Cheesman. *Estimating uncertain spatial relationships in robotics*. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, New York, 1990.
59. Srinivasan, M.V., M. Lehrer, W.H. Kirchner, and S.W. Zhang. "Range perception through apparent image speed in freely flying honeybees". *Vis. Neurosci.*, 6:519–535, May 1991.
60. Strelow, Dennis and Sanjiv Singh. "Optimal Motion Estimation from Visual and Inertial Measurements". *Proceedings of the Workshop on Applications of Computer Vision*. December 2002.
61. Strelow, Dennis and Sanjiv Singh. "Online Motion Estimation from Visual and Inertial Measurements". *Proceedings of the Workshop on Integration of Vision and Inertial Sensors (INERVIS 2003)*. June 2003.
62. Strelow, Dennis W. *Motion Estimation from Image and Inertial Measurements*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, November 2004.
63. Tekalp, A. Murat. *Digital Video Processing*. Prentice Hall PTR, Upper Saddle River, New Jersey 07458, 1995.
64. Titterton, D.H. and J.L. Weston. *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd., Lavenham, United Kingdom, 1997.
65. Van Sickle, S. *Modern Airmanship*. Van Nostrand Company, Inc., New Jersey, 1957.
66. Van Trees, Harry L. *Detection, Estimation, and Modulation Theory*. John Wiley and Sons, Inc., New York, New York, 2001.
67. Vanicek, P. and D.E. Wells. *The Least Squares Approximation*. University of New Brunswick, Fredericton, New Brunswick, Canada, 1972.
68. Veth, Michael J. and John F. Raquet. "Two-Dimensional Stochastic Projections for Tight Integration of Optical and Inertial Sensors for Navigation". *National Technical Meeting Proceedings of the Institute of Navigation*, 587–596. 2006.
69. Veth, Michael J., John F. Raquet, and Meir Pachter. "Stochastic Constraints for Robust Image Correspondence Search". *IEEE Transactions on Aerospace Electronic Systems*, Publication forthcoming.

70. Warden, John A., III. "Air Theory for the 21st Century". *Air and Space Expeditionary Operations*, v4.0:108–119, 1994.
71. Wei, M. and K. P. Schwarz. "A Strapdown Inertial Algorithm Using an Earth-Fixed Cartesian Frame". *Journal of the Institute of Navigation*, 37(2):153–167, Summer 1990.
72. Wells, D.E. and E.J. Krakiwsky. *The Method of Least Squares*. University of New Brunswick, Fredericton, New Brunswick, Canada, 1971.

Index

The index is conceptual and does not designate every occurrence of a keyword.

- acceleration, centripetal, 31
- accelerometer, 2, 19
- aperture, optical, 34
- astrolabe, 58

- batch estimation, optimal, 52
- binocular disparity, 110
- binocular disparity frame, 14
- binocular measurement, 109
- binocular stereopsis, 78
- body frame, 12

- camera frame, 14
- combat, precision, 1
- contour-matching, terrain, 62
- correspondence, feature, 97
- Crista IMU, 128

- difference of Gaussian, 83
- difference of Gaussian filter, 83
- direction cosine matrix, 16
- driftmeter, 61

- Earth-centered Earth-fixed frame, 11
- Earth-centered inertial frame, 11
- Earth-fixed navigation frame, 12
- ellipsoidal height, 25
- epipolar, 111
- Euler angles, 16, 25

- feature correspondence, 97
- feature space, 80
- feature tracker, Lucas-Kanade, 44
- feature vector, 80
- filter, difference of Gaussian, 83
- focus of expansion, 65

- Gauss-Markov process, first-order, 28
- geodetic latitude, 25
- geodetic radius of meridian, 26
- gradient, intensity, 88
- gravitation, 31
- gravity potential, 31
- gravity vector, 31
- gyroscope, 19
- gyroscopes, 3

- Harris corner detector, 44
- Harris detail metric, 45
- homogeneous pointing vector, 104
- Honeywell HG1700, 128

- image, 3, 35
- inertial reference frame, 11
- intensity gradient, 88
- irradiance, 35

- landmark, 90
- landmarks, opportunity, 100
- landmarks, reference, 100
- latitude, geodetic, 25
- lever arm, 14
- linear regression, 52
- longitude, 25
- Lucas-Kanade feature tracker, 44

measurement model, accelerometer, 28
 measurement model, gyroscopic, 28
 misclosure vector, 54

 navigation, 2
 navigation reference frame, 10
 navigation, relative, 120
 nonlinear regression, 53

 operator, skew-symmetric, 17
 optical flow, 62

 perturbation equation, linear, 18
 photon arrival rate, 35
 photon counting noise, 37
 pitch angle, 25
 point spread function, 35
 pose metric, 99
 precision combat, 1
 principal component analysis, 176

 quantization noise, 37

 radial distortion, 44
 radiance, 35
 reference frame, binocular disparity, 14
 reference frame, body, 12
 reference frame, camera, 14
 reference frame, Earth-centered Earth-fixed, 11
 reference frame, Earth-centered inertial, 11
 reference frame, Earth-fixed navigation, 12
 reference frame, inertial reference, 11
 reference frame, navigation, 10
 reference frame, vehicle-fixed navigation, 11
 relative navigation, 120
 roll angle, 25

 scene, 3, 35

 sextant, 58
 shot noise, 37
 SIFT, 82
 skew-symmetric operator, 17
 specific force, 2, 19, 31
 stadiametric ranging, 176
 state transition matrix, 94
 strapdown inertial sensors, 14

 TERCOM, 62
 thin lens equation, 38
 track management, 102

 vehicle-fixed navigation frame, 11
 velocity vector, 25
 VLSI, 63
 voxel, 88

 yaw angle, 25

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
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 this 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) 21-08-2006		2. REPORT TYPE Ph D Dissertation		3. DATES COVERED (From — To) Sept 2003 — Sep 2006		
4. TITLE AND SUBTITLE Fusion of Imaging and Inertial Sensors for Navigation				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Michael J. Veth, Major, 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 2950 Hobson Way WPAFB OH 45433-7765					8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DS/ENG/06-09	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/MN Dr. T.J. Klausutis 101 W. Eglin Blvd., Suite 210 Eglin AFB, FL 32542-6810 timothy.klausutis@eglin.af.mil (850) 883-0877					10. SPONSOR/MONITOR'S ACRONYM(S)	
					11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT The motivation of this research is to address the limitations of satellite-based navigation by fusing imaging and inertial systems. The research begins by rigorously describing the imaging and navigation problem and developing practical models of the sensors, then presenting a transformation technique to detect features within an image. Given a set of features, a statistical feature projection technique is developed which utilizes inertial measurements to predict vectors in the feature space between images. This coupling of the imaging and inertial sensors at a deep level is then used to aid the statistical feature matching function. The feature matches and inertial measurements are then used to estimate the navigation trajectory using an extended Kalman filter. After accomplishing a proper calibration, the image-aided inertial navigation algorithm is then tested using a combination of simulation and ground tests using both tactical and consumer-grade inertial sensors. While limitations of the Kalman filter are identified, the experimental results demonstrate a navigation performance improvement of at least two orders of magnitude over the respective inertial-only solutions.						
15. SUBJECT TERMS navigation, image processing, image registration, image-aided navigation, feature tracking, inertial navigation, passive navigation, autonomous navigation, simultaneous location and mapping (SLAM), computer vision, correspondence search, egomotion estimation, sensor fusion						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 207	19a. NAME OF RESPONSIBLE PERSON Dr. John F. Raquet	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) (937) 255-3636, ext 7264	