MONOCULAR VISION LOCALIZATION USING A GIMBALED LASER RANGE SENSOR

THESIS

Don J. Yates, Second Lieutenant, USAF

AFIT/GE/ENG/10-31

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

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

AFIT/GE/ENG/10-31

MONOCULAR VISION LOCALIZATION USING A GIMBALED LASER RANGE SENSOR

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Don J. Yates, B.S.

Second Lieutenant, USAF

March, 2010

AFIT/GE/ENG/10-31

MONOCULAR VISION LOCALIZATION USING A GIMBALED LASER RANGE SENSOR

Don J. Yates, B.S.
Second Lieutenant, USAF

Approved:

//Signed//	
Lt Col M. J. Veth Ph.D. (Chairman)	Date
//Signed//	
Maj K. Fisher Ph.D. (Member)	Date
//Signed//	
Dr. R. Beard (Member)	Date

Abstract

There have been great advances in recent years in the area of indoor navigation. Many of these new navigation systems rely on digital images to aid an inertial navigation estimates. The Air Force Institute of Technology (AFIT) has been conducting research in this area for a number of years. The image-aiding techniques are centered around tracking stationary features in order to improve inertial navigation estimates. Previous research has used stereo vision systems or terrain constraints with monocular systems to estimate feature locations. While these methods have shown good results, they do have drawbacks. First, as unmanned exploration vehicles become smaller in size the distance available to create a baseline between two cameras decreases resulting in a decrease of distancing accuracy. Second, if using a monocular system, terrain data might not be known in an unexplored environment. This research explores the use of a small gimbaled laser range sensor and monocular camera to estimate feature locations. The gimbaled system consists of a commercial off-the-shelf range sensor, a pair of hobby-style servos, and a micro controller that accepts azimuth and elevation commands. The system is approximately 15x8x12 cm and weighs less than 120 grams.

This novel approach, called laser-aided image inertial navigation, provides precise depth measurements to key features. The location of these key features are then calculated based on the current state estimates of an Extended Kalman filter. This method of estimating feature locations is tested both by simulation and real world imagery. Navigation experiments are presented which compare this method with previous image-aided filters. While only a limited number of tests were conducted, simulated and real world flight tests show that the monocular laser-aided filter can accurately estimate the trajectory of a vehicle to within a few tenths of a meter. This is done without terrian contraints or any prior knowledge of the operational area.

Acknowledgements

First, I have to thank my advisor for his patience and support. It's annoying to have to repeat the same thing more than once, but you were always willing to take the time to teach me the obvious. You taught me how to slow down, think things through, and not run head-long into a problem.

Thank you to my committee members for being willing to support me in my research. Thank you to friends for listening to my ideas, and helping review my thesis.

Thank you to my parents for believeing in me when others didn't. Thank you for encouraging me, and providing me with the tools I needed to get me where I am today.

Most of all, I want to thank my wife for all her love, support, and help. You always know how to calm me down when things aren't going well, and are willing to listen to me, even when you don't have a clue what I'm saying. Thank you.

Don J. Yates

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures ix

List of Tables xi

List of Abbreviations xii

1. Introduction 1

1.1 Purpose 1

1.2 Contributions 2

1.3 Outline 2

2. Background Material 3

2.1 Mathematical Notation 3

2.2 Reference Frames 4

2.2.1 Earth-Fixed Inertial Frame 5

2.2.2 Earth-Centered-Earth-Fixed (ECEF) 5

2.2.3 Navigation Frame 5

2.2.4 Body Frame 6

2.2.5 Camera Frame 6

2.2.6 Gimbal Frame 8

2.3 Transformation Matrices 8

2.4 Projection Theory 10

2.5 Inertial Navigation Systems 14

2.6 Kalman Filter 16

Page

2.7 Extended Kalman Filter 19
2.8 Image Aiding Techniques 21
2.9 Scale Invariant Feature Transformation 24
2.9.1 Scale-Space Decomposition 24
2.9.2 Extrema Detection 25
2.9.3 Feature Descriptor Calculation 25
2.10 Laser Range Sensors 28
2.11 Stereo verses Laser 28
2.12 Related work 29

3. Methodology 31
3.1 Navigation Filter 31
3.2 System Model 31
3.3 Feature Location Estimation 34
3.4 Line of Sight Vector Calculations 35
3.4.1 Rotation Calculation 35
3.4.2 Offset Correction 36
3.4.3 Correcting for Translation and Rotation 36
3.5 Measurement Model 38
3.6 Error Analysis 46

4. Results 48
4.1 Hardware Overview 48
4.2 Simulation Experiment 53
4.3 Feature Location Test 59
4.4 Vicon Experiment 60
4.5 Hallway Experiment 69

Page

5. Conclusions 75

5.1 Future Work 75

5.2 Summary 76

Bibliography 77

List of Figures

Figure		Page
2.1	Earth and Vehicle-Fixed Navigation Frame	6
2.2	Navigation Frame	7
2.3	Vehicle Body Frame	7
2.4	Gimbal Frame	8
2.5	Projective Camera Model	11
2.6	Pinhole Camera Model	12
2.7	Camera Image Array	13
2.8	Kalman Filter Cycle	18
2.9	Image-Inertial Algorithm	21
2.10	Stochastic Feature Projection	23
2.11	Scale Decomposition	26
2.12	Octaves of the Difference of Gaussian Functions Over a Scale-Space	27
2.13	Possible Keypoint	27
3.14	Navigation Filter Block Diagram	32
3.15	Camera GLRS Disparity	37
3.16	Translation Illustration	39
3.17	Feature Position Geometry	40
4.18	Hardware Block Diagram	49
4.19	Experimental Hardware	50
4.20	Experimental Hardware Alignment	50
4.21	Experimental Platform Setup	52
4.22	Simulated Hallway	53
4.23	Simulated Hallway Position Errors	54
4.24	Simulated Hallway Attitude Errors	55
4.25	RSS Horizontal Error Plot	56

Figure Page

4.26 RSS Vertical Error Plot 57

4.27 RSS Attitude Error Plot 58

4.28 Position Estimates 59

4.29 Estimated Vicon Trajectory 61

4.30 Vicon Position Trajectory 62

4.31 Vicon Attitude Trajectory 63

4.32 Vicon Position Error 64

4.33 Vicon Attitude Error 65

4.34 RSS Horizontal Error 66

4.35 RSS Vertical Error 67

4.36 RSS Attitude Error 68

4.37 Experimental Hallway 69

4.38 Platform Hallway Trajectory Estimate 70

4.39 Platform Hallway Vertical Trajectory 71

4.40 Platform Hallway Attitude Estimate 72

4.41 Simulated Position Results of Near Ceiling Features 73

4.42 Simulated Attitude Results of Near Ceiling Features 74

List of Tables

Table		Page
3.1	Navigation Filter Uncertainty Table	47
3.2	Feature Location Uncertainties by Distance	47
4.3	IMU Specifications	51
4.4	Feature Location Estimates	60

List of Abbreviations

Abbreviation		Page
GPS	Global Positioning System	1
AFIT	Air Force Institute of Technology	1
GLRS	Gimbaled Laser Range Sensor	2
MAV	Micro Aerial Vehicle	2
ECEF	Earth-Centered Earth-Fixed	5
NED	North-East-Down	6
DCM	Direction Cosine Matrix	9
INS	Inertial Navigation System	14
IMU	Inertial Measurement Unit	14
EKF	Extended Kalman Filter	20
SIFT	Scale Invariant Feature Transformation	22
SLAM	Simultaneous Localization and Mapping	30
RSS	Root-Sum-Square	55

MONOCULAR VISION LOCALIZATION USING A GIMBALED LASER RANGE SENSOR

I. Introduction

The advent of modern warfare has brought warfighters off the battlefield and into cities and towns. No longer do massive forces move from battlefield to battlefield engaging the enemy, rather much smaller groups of soldiers move from city block to city block, and from house to house. The military has begun using small robotic vehicles to take warfighters out of harms way, but the complexities of fighting in these urban environments presents many challenges. One significant challenge is that the Global Positioning System (GPS), which has become the standard for navigation, may not be available, especially indoors or underground. The use of autonomous robotic vehicles depends on accurate navigation solutions, thus alternatives to GPS navigation need to be explored.

There have been great advances in recent years in the area of non-GPS navigation. Some of these advances have come by the fusing of data from different sensors to improve inertial navigation estimates. The Air Force Institute of Technology (AFIT) has been researching techniques that combine digital images and inertial sensors. This is the technique that will be presented in this research.

1.1 Purpose

Image inertial navigation systems, in unstructured indoor environments, have traditionally used binocular camera systems. By creating a large baseline between cameras, depth estimates to objects in the cameras' field of view can be calculated. Unfortunately, as technology continues to develop and as unmanned vehicles continue to get smaller and smaller, the available distance to separate binocular systems also continues to decrease. As the binocular disparity decreases, so do the accuracies of the depth estimates. Binocular systems also have problems with observability, which arises when the distance to the object being measured is large when compared to the distance between the cameras.

This research focuses on the development and testing of a novel approach to solve the monocular depth estimation problem. The system to be presented uses a gimbaled laser range sensor (GLRS) and a single camera to estimate locations of interesting objects. Once a object has been localized, observations of this object can then be used to help localize the vehicle to which the camera is attached. The application of this approach is to provide accurate imaged-aided inertial navigation estimates, with an emphasis in application on micro aerial vehicles (MAV).

1.2 Contributions

Important contributions of this work include development and testing of a novel gimbaled laser range sensor and development of a measurement model with corresponding influence matrixes for use in an extended Kalman filter. Contributions also include data from simulated and real world experiments, which help to quantify the performance of a navigation filter which uses this sensor, a camera, and commercial grade IMU.

1.3 Outline

The research presented in this thesis is organized as follows. First, background material will be presented in Chapter II. This material will not cover every topic in depth, but will lay the foundation for the research presented in Chapter III. Chapter III covers the methodology of the proposed laser-aided image inertial algorithm, including development of the measurement models and uncertainties. Chapter IV goes over the experiment setup and results of the navigation filter. Finally, Chapter V will give conclusions drawn from the results in Chapter IV and discusses possible future work.

II. Background Material

This chapter presents background material needed to understand the basics concepts of this thesis. Every topic will not be covered in depth, however basic concepts are presented. The topics presented include reference frames, transformation matrices, projection theory, inertial navigation systems, Kalman filters, and image aiding techniques.

2.1 Mathematical Notation

Before beginning the background material for this thesis, it is important to understand the mathematical notation that will be used. This will provide insight regarding the equations at a quick visual inspection. The notation for this thesis is as follows:

Scalars: Scalars are denoted by lower or upper case italic font, for example x or G.

Vectors: Vectors are represented by lower case letters with bold font, for example $\mathbf{x}$, or $\boldsymbol{\psi}$. The vector $\mathbf{x}$ is composed of a column of scalar elements x_i, where i is the element number.

Matrices: Matrices are denoted by upper case letters in bold font, for example $\mathbf{A}$. The matrix $\mathbf{A}$ is composed of elements A_{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 T, as in $\mathbf{X}^T$.

Estimated Variables: Estimated variables are given a *hat* character directly above the variable, for example $\hat{x}$.

Computed Variables: Variables that are corrupted by noise are given a *tilde* character directly above the variable, for example $\tilde{x}$.

Direction Cosine Matrices: Direction cosine matrices from frame a to frame b are denoted by $\mathbf{C}_a^b$.

Homogeneous Coordinates: Vectors in homogeneous coordinates are represented by an *underline*, for example $\underline{\mathbf{x}}$. Homogeneous vectors are defined to have a 1 in the last column element.

Reference Frames: A superscript is used to denote which reference frame a vector is in, for example the vector $\mathbf{x}^c$ is represented in the c frame.

Relative Position or Motion: In cases were it is important to specify relative motion, combined subscript letters are used to designate the frames, for example ω_{ab} denotes the rotation rate vector *from* frame a *to* frame b.

Functions: Functions are denoted by boldface font with their parameters depicted in parentheses, for example $\mathbf{F}(\mathbf{u}, t)$.

All other notation is straightforward and should be understood from context.

2.2 Reference Frames

In order to describe the position and orientation (i.e., attitude) of any object, a frame of reference or coordinate system must be established. Coordinate systems are used to express position, velocity, acceleration, attitude, and angular rates of objects. By convention all coordinate systems in this work are orthogonal and right-handed. There are several reference frames used in this research, including:

- Earth-Fixed inertial frame (i-frame)
- Earth-Centered Earth-Fixed frame (e-frame)
- Navigation frame (n-frame)
- Body frame (b-frame)
- Camera frame (c-frame)
- Gimbal frame (g-frame)

These frames are necessary for the following reasons:

- On-board sensors like accelerometers and rate gyros measure information with respect to the inertial frame.
- Targets of interest as well as vehicle paths are specified in the navigation frame.
- Forces and torques are applied in the body frame.
- Camera data is relative in the camera frame.
- Range data is relative in the gimbal frame.

2.2.1 Earth-Fixed Inertial Frame. An inertial reference frame is defined as a frame of reference which is non-rotating and non-accelerating [1]. This is the frame in which Newton's laws hold. Although the Earth-Fixed inertial frame is not a true inertial frame, because the Earth rotates around the sun, it is a valid inertial frame for terrestrial navigation. The Earth-Fixed inertial frame has its z_i axis aligned with the rotation axis of the Earth. This is roughly aligned with the Earth's poles. The x_i and y_i axis are lie in the equatorial plane. In particular, the x_i axis points toward fixed stars, and the y_i axis completes the right-handed reference frame.

2.2.2 Earth-Centered-Earth-Fixed (ECEF). The Earth-Centered Earth-Fixed (ECEF) frame, sometimes called the Earth frame, has its z_e axis aligned with the z_i axis, and the x_e and y_e axis lie in the equatorial plane. The x_e axis is fixed at the intersection of the Greenwich meridian and the equatorial plane, and the y_e axis is again defined using the right-handed rule.

2.2.3 Navigation Frame. It is entirely possible to define positions of objects using only the ECEF frame, however it is easier for humans to understand local-level navigation frames than ECEF. For example, if one were to describe the position of a chair in front of himself or herself, it is more meaningful to say it's one meter to the right and two meters forward than to give the ECEF coordinates. This is the reason for the navigation frame. The navigation frame is a rotating frame that is tangential to the surface of the Earth in which x_n is defined as pointing toward the North Pole, y_n is defined as pointing East, and

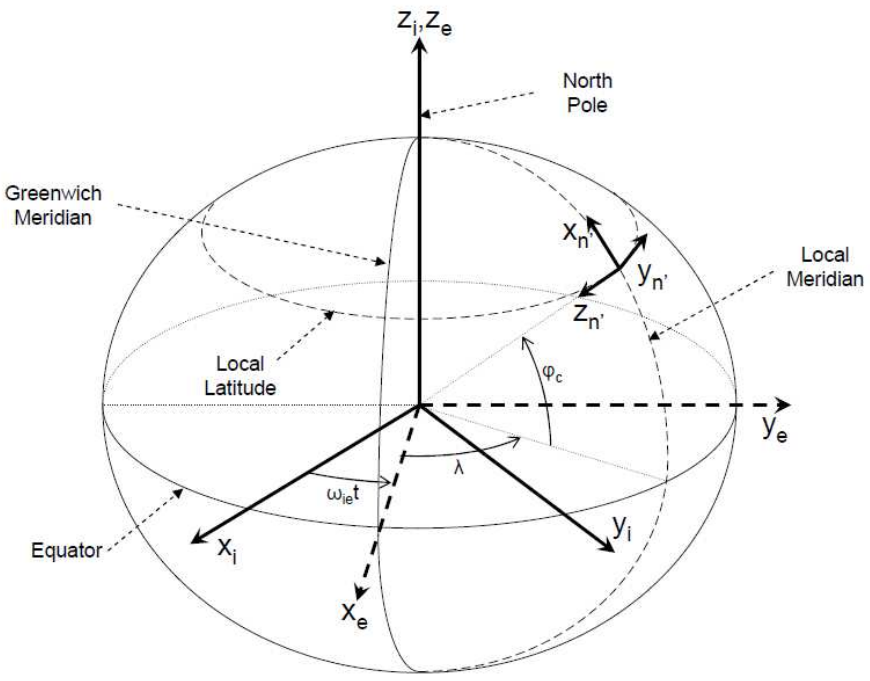

Figure 2.1. The inertial and Earth frames originate at the Earth's center of mass while the vehicle-fixed navigation frame's origin is located at a fixed location on a vehicle [2].

z_n is defined as pointing toward the center of the Earth, as shown in Figure 2.1. This is a north-east-down (NED) convention. The origin of the navigation frame is centered at the center of the navigation system, as shown in Figure 2.2.

2.2.4 Body Frame. The body frame has its origin at the center of mass of the vehicle. In this frame, x_b points out the front of the vehicle, y_b points out the right side of the vehicle, and z_b points out the belly or bottom of the vehicle, as shown in Figure 2.3. The body frame is defined with respect to the roll, pitch, and yaw angles about the navigation frame.

2.2.5 Camera Frame. The camera frame originates at the optical center and is defined so that the x_c axis points up in the image plane, the y_c axis points to the right of the image plane, and the z_c axis points out the optical axis of the camera, in other words, out the front of the camera.

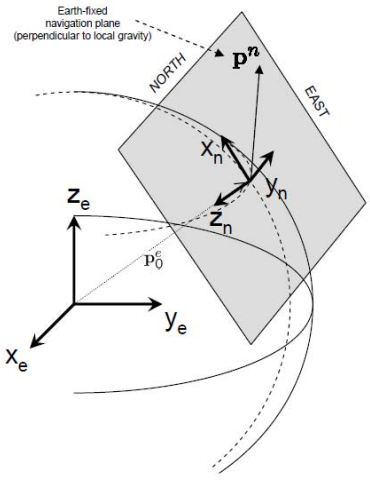

Figure 2.2. The navigation frame is a Cartesian reference frame that is tangential to the surface of the Earth [2].

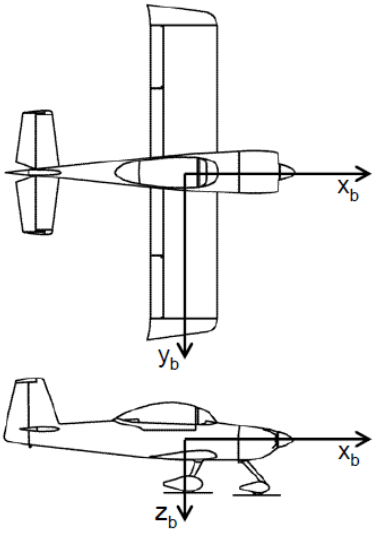

Figure 2.3. A rotating reference frame with the origin at the body's center of gravity [2].

2.2.6 Gimbal Frame. The gimbal frame originates at the gimbal rotation center and is oriented so that x_g points along the beam axis of the laser sensor, y_g points to the right of the laser axis, and z_g points down from the laser axis.

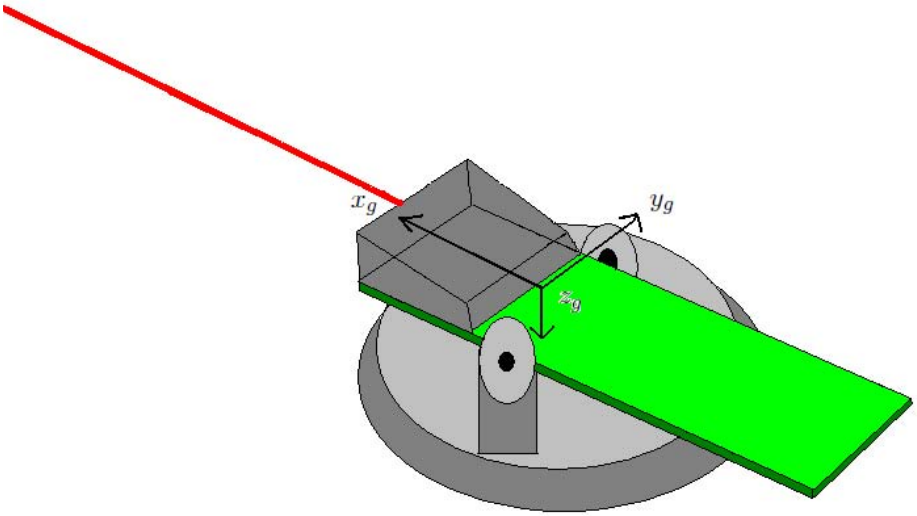

Figure 2.4. The gimbal frame is aligned with the beam axis of the laser, and the rotation center of the gimbal

2.3 Transformation Matrices

In order to move from one reference frame to another a set of transformation matrices are need. Transformation matrices define relationships between vectors in different reference frames. The notation

$$\mathbf{v}^i = \mathbf{C}_n^i \mathbf{v}^n \tag{2.1}$$

denotes that the rotation matrix $\mathbf{C}$ transforms the vector $\mathbf{v}^n$ from the n frame to the i frame.

Rotation matrices, like the one above, have specific properties. Rotation matrices are orthogonal, $\mathbf{C}^T = \mathbf{C}^{-1}$, and their determinants must be equal to one, $det(\mathbf{C}) = 1$. A direction cosine matrix (DCM) is an example of a rotation matrix.

A DCM is a 4-parameter rotation expressed as a 3×3 matrix [1]. The advantage of a DCM is that, because it is a 4-parameter rotation, it does not have singularities. DCMs are commonly used to rotate between the body frame and the navigation frame. Roll, pitch, and yaw are expressed between these frames. Roll is defined as a rotation of the body about the x_b-axis, pitch is a rotation about the y_b-axis, and yaw is a rotation about the z_b-axis. These angles, known as Euler angles, will be denoted by ϕ, θ, ψ for roll, pitch, and yaw respectively. The DCM to rotate from the body frame to the navigation frame is given by Equation (2.2) [1].

$$\mathbf{C}_b^n = \begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & cos\psi sin\theta cos\phi + sin\psi sin\phi \\ sin\psi cos\theta & sin\psi sin\theta sin\phi + cos\psi cos\phi & sin\psi sin\theta cos\phi - cos\psi sin\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix} \tag{2.2}$$

The time derivative of $\mathbf{C}_b^n$ is given by [1]

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \Omega_{nb}^b \tag{2.3}$$

where Ω_{nb}^b is the skew-symmetric form of the rotation vector from the navigation frame to the body frame, expressed in the body frame, $\boldsymbol{\omega}_{nb}^b$. The skew-symmetric operator of a vector $\boldsymbol{\omega}$ is defined as [1]

$$\boldsymbol{\omega}\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.4}$$

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

A homogeneous transformation matrix combines both a rotation matrix and a translation vector in a single matrix. The homogeneous transformation matrix from frame n to frame i is given by [3]

$$\mathbf{T}_n^i = \begin{bmatrix} \mathbf{C}_n^i & -\mathbf{v}_n^i \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.5}$$

where $\mathbf{0} \in \Re^3$ is a row vector of zeros. The vector $\mathbf{v}_n^i$ is a translation vector resolved in the i^{th} coordinate frame.

2.4 Projection Theory

This research uses a simple projective camera model as shown in Figure 2.5. A projective camera model is a modified form of the pinhole camera model shown in Figure 2.6. The pinhole camera model has an infinitely small lens. This camera model assumes all rays pass through the same point, the optical center. The difference between the two models is that in a projective model the image plane is moved one focal length in front of the optical center, instead of behind the optical center. This removes the issue of inverting the image. The focal length, f, is defined by

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \tag{2.6}$$

where Z is the distance from an object to the camera lens and z is the distance from the lens to the image plane.

The image plane consists of an $(M \times N)$ array of pixels, where M is the height of the pixel array and N is the width of the array. This camera model assumes a ray of light is reflected off an object in the image scene, passes through the image plane, and arrives at the optical center. The path of the light ray is called the line of sight vector, and can be calculated based on where the ray intersects the image plane.

It is important to note that the image plane is defined so that x points down, along the M direction, and y points to the right, the N direction as shown in Figure 2.7. This is opposite the standard computer vision notation.

In [4], Trucco shows that by using this camera model the transformation from meters in the camera frame ($\mathbf{s}^c$) to pixels in the image plane ($\mathbf{z}_u$) is

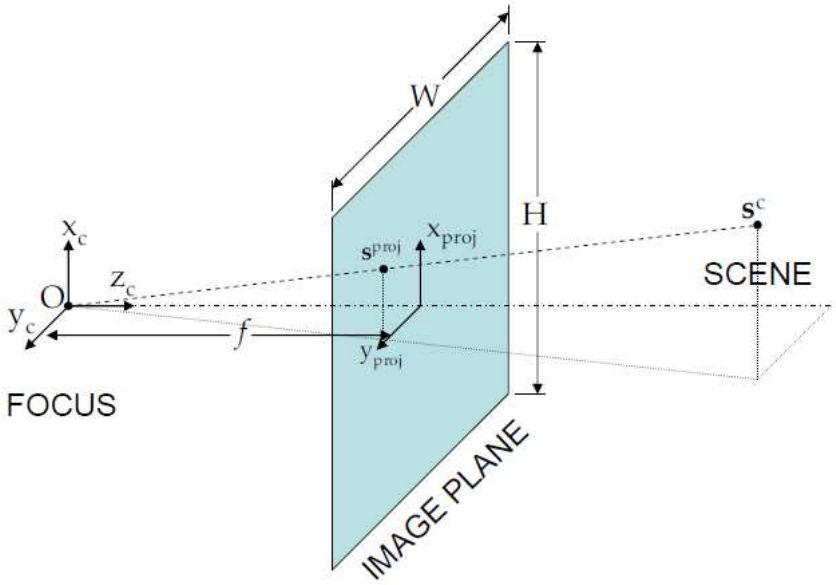

Figure 2.5. A simple projective camera model based on a pinhole camera model. This model has the image plane moved one focal length in front of the optical center [2].

$$\mathbf{z}_u = \frac{1}{s_z^c} \begin{bmatrix} -f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{s}^c \tag{2.7}$$

where f_x and f_y are the focal lengths of the camera, and c_x, c_y are pixel locations at the origin of the image plane, known as the principal points. s_z^c is the z component of the line of sight vector, s^c.

Equation (2.7) can be re-written as

$$\mathbf{z}_u = \frac{1}{s_z^c} \mathbf{T}_c^{pix} \mathbf{s}^c \tag{2.8}$$

The matrix $\mathbf{T}_c^{pix}$ is sometimes referred to as the calibration or intrinsic matrix and can be found using a number of calibration techniques [5, 6]. The calibration matrix, $\mathbf{T}_c^{pix}$, projects a 3-dimensional scene onto a 2-dimensional image plane; because of this, scale is not maintained.

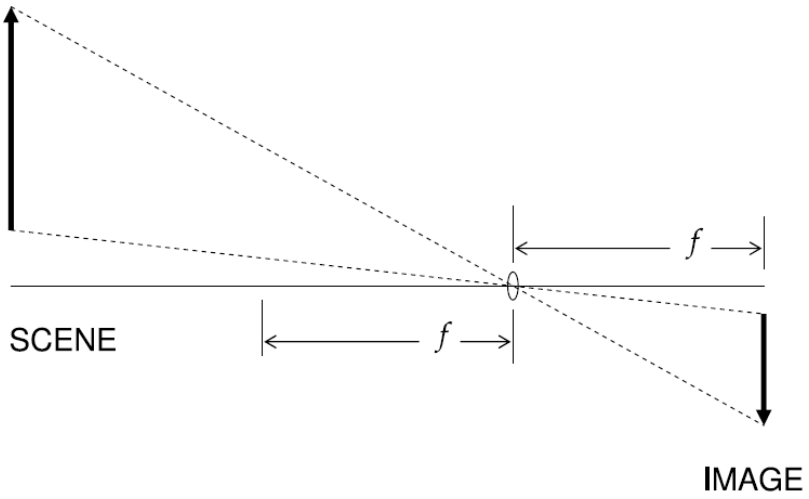

Figure 2.6. The pinhole camera model has an infinitely small lens. This camera model assumes all rays pass through the same point, the optical center. Note the projected image is inverted [2].

Because the projection does not maintain scale, it is common practice to set the z component of $\mathbf{s}^c$ equal to one. This is done by normalizing $\mathbf{s}^c$ by its z component, i.e.,

$$\underline{\mathbf{s}}^c = \frac{\mathbf{s}^c}{s_z^c} \tag{2.9}$$

where $\underline{\mathbf{s}}^c$, called the homogeneous line of sight vector, is the normalized version of $\mathbf{s}^c$.

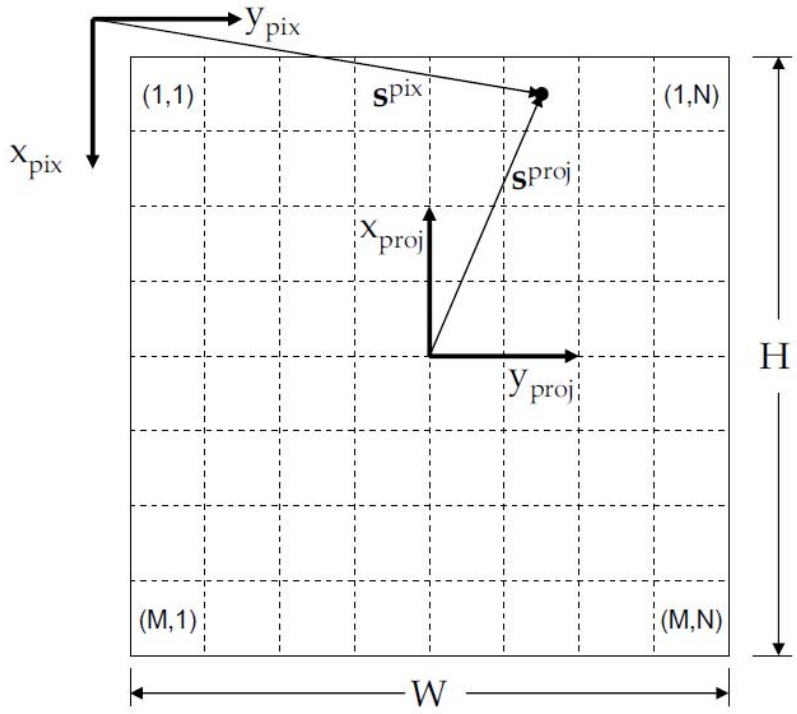

Figure 2.7. A camera image consists of an $(M \times N)$ array of pixels. The physical height and width of the array is represented by H and W, respectively [2].

2.5 Inertial Navigation Systems

An inertial navigation system (INS) calculates velocities, positions, and attitude angles by integrating measurements from an inertial measurement unit (IMU). The IMU measures specific force and rotation rates of the body with respect to inertial space. These systems typically consist of 2 triads of orthogonal mounted gyros and accelerometers. The accelerometers are normally aligned with the body's x_b, y_b, and z_b axis. An INS can be rigidly attached to a platform body, commonly called a strapdown system, or mounted on an inertially stabilized gimbal. Strapdown INS comprise 90% of systems in use today [7]. A strapdown INS is used in this research.

At the heart of any INS are the accelerometers and gyros. Accelerometers measure specific force. Specific force is the difference between the gravitation vector and an acceleration vector.

$$\mathbf{f}^b = \mathbf{a}^b - \mathbf{g} \tag{2.10}$$

where $\mathbf{a}^b$ is acceleration caused by body movement, $\mathbf{g}$ is acceleration due to gravity, and $\mathbf{f}_b$ is the body force measurement that the accelerometer reads. This means that in order to integrate the accelerometer measurements to calculate body movement, one must know the starting position and the local gravity field vector. Equation (2.11) is commonly called the navigation equation [1], where $\mathbf{a}^n$ is integrated to obtain position.

$$\mathbf{a}^n = \mathbf{C}_b^n \mathbf{f}^b + \mathbf{g}^n \tag{2.11}$$

here $\mathbf{C}_b^n$ is the DCM that transforms force in the body frame to force in the navigation frame. This DCM is calculated by integrating the rate gyros.

Gyros measure angular rates of the body with respect to the inertial frame $\boldsymbol{\omega}_{ib}^b$. In order to calculate the rotation rate of the body with respect to the navigation frame, $\boldsymbol{\omega}_{nb}^b$, the rotation rate of the Earth and the rotation rate of the navigation frame with respect to the Earth-Fixed Inertial frame must be known, i.e.,

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b[\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n] \tag{2.12}$$

where $\boldsymbol{\omega}_{ie}^{n}$ is a vector describing the rotation of the earth, and $\boldsymbol{\omega}_{en}^{n}$ is the rotation rate of the navigation frame about the Earth-Fixed Inertial frame due to the vehicle's velocity. Because a low quality IMU (which cannot sense Earth rate) will be used for this research, and because the distance traveled by the vehicle is short (less than 50 meters), $\boldsymbol{\omega}_{ie}^{n}$ and $\boldsymbol{\omega}_{en}^{n}$ can be neglected.

Another force that can be neglected, but is included here for completeness, is accelerations due to the Coriolis effect. The Coriolis effect is seen whenever a body is moving with respect to a rotating frame. A derivation of the equations of Coriolis can be found in [8]. The Coriolis acceleration can be expressed as

$$\frac{d^2\mathbf{p}^e}{dt^2} = -2\boldsymbol{\omega}_{ie}^{e} \times \mathbf{v}^e \tag{2.13}$$

where $\mathbf{p}^e$ is the position of the vehicle in the Earth frame, $\mathbf{v}^e$ is the velocity of the vehicle in the Earth frame, and $\boldsymbol{\omega}_{ie}^{e} = [0 \;\; 0 \;\; \Omega]^T$. The constant Ω is the turn rate of the Earth frame with respect to the i-frame.

The Coriolis correction can be calculated by knowing the rotation rate of the Earth and the groundspeed of the vehicle in the navigation frame [1], i.e.,

$$\mathbf{c}_{corr} = (2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n}) \times \mathbf{v}_{e}^{n} \tag{2.14}$$

where ω_{ie}^{n} is the rate of change of the Earth frame with respect to the inertial frame expressed in the navigation frame, ω_{en}^{n} is the rate of change of the navigation frame with respect to the Earth frame expressed in the navigation frame, and $\mathbf{v}_{e}^{n}$ is vehicle velocity in the Earth frame expressed in the navigation frame.

Although Coriolis acceleration, the rotation rate of the Earth, and rotations of the navigation frame are ignored, there are some sources of errors that should not be ignored. These sources of errors include bias errors, scale factors errors, misalignment errors, and measurement noise. Some of the errors, for example biases, can be estimated and subtracted out of the measurement. However, not all the errors can be corrected, and these

errors cause the gyros to drift over time. In low quality gyros, the drift rate is fast enough that only a few seconds can pass before attitude estimates are far from the truth.

Even though an INS's measurements drift over time, they do have benefits. A major benefit of an INS is its large bandwidth. A typical INS can be sampled around 100Hz, which means it can capture high dynamics. While this is true, most INS systems are coupled with other sensors that have lower bandwidth, but also have slower, if any, drift. This is called INS aiding. Aiding is accomplished by updating the INS estimates with measurements from other sensors and, in the case of this research, consists of a camera and laser range sensor. The fusion of the INS state estimates and other sensors is done with a Kalman filter.

2.6 Kalman Filter

A Kalman filter estimates system states by fusing a system model, measurement models, and sensor data in the presence of noise. A state is a characteristic of a system that is needed in order to capture accurately the system's dynamics, for example; positions, velocities, and attitudes.

In the most basic case, a linear system and measurement model and zero-mean, white Gaussian noise sources, a Kalman filter is an optimal estimator of the system states. The model can be represented by a differential equation in terms of the system states ($\mathbf{x}$), inputs ($\mathbf{u}$), and noise sources ($\mathbf{w}$). The following differential equation expresses how the states change overtime:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.15}$$

where $\mathbf{F}(t)$ is the state dynamic matrix, $\mathbf{B}(t)$ is the matrix that defines how inputs affect each state , and $\mathbf{G}(t)$ is the matrix that defines how the noise sources affect each state. The noise sources are assumed to be zero-mean, white Gauissan processes. The noise source covariance is defined as

$$\mathbf{E}[\mathbf{w}(t)\mathbf{w}^T(t+\tau)] = \mathbf{Q}(t)\delta(\tau) \tag{2.16}$$

where $\mathbf{E}[\ ]$ is the expectation operator, $\mathbf{Q}(t)$ is process noise, and δ is the Dirac delta function.

In order to use the Kalman filter in a digital computer, the differential equation shown in Equation (2.15) must be transformed to a difference equation of the form

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k \tag{2.17}$$

where the notation $\mathbf{x}_{k+1}$ denotes the state estimate at time $k+1$ and $\mathbf{x}_k$ is the state estimate at time k. The state transition matrix, Φ_k, is defined as

$$\Phi_k = e^{\mathbf{F}_k \Delta t} \tag{2.18}$$

where Δt is the time step of the discrete system. $\mathbf{B}_k$ and $\mathbf{G}_k$ are calculated as [9]

$$\mathbf{B}_k = \int_{t_k}^{t_k+1} \Phi(t_{k+1}, \tau)\mathbf{B}(\tau)d\tau \tag{2.19}$$

$$\mathbf{G}_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^T(\tau)\Phi^T(t_{k+1}, \tau)d\tau \tag{2.20}$$

where $\mathbf{Q}$ is the process noise covariance matrix. The difference equation shown in Equation (2.17) can be implemented in a computer; however, it is also necessary to have a model of the sensor measurements.

The discrete time measurement model is defined as

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.21}$$

where $\mathbf{v}_k$ is the measurement noise and $\mathbf{H}_k$ is the observation matrix, which relates the measurements to the states.

There are two steps to a Kalman filter that are repeated recursively: the propagation and the update phase, as shown in Figure 2.8. In the propagation phase the system model uncertainty is allowed to grow. The filter uncertainty grows because of noise and

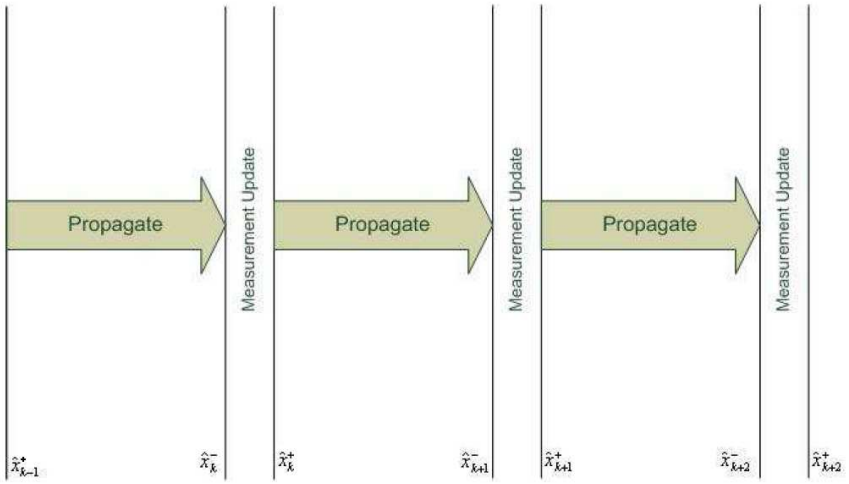

Figure 2.8. Kalman filter update and propagation cycle from time $k-1$ to time $k+2$ [9].

disturbances to the system. The rate of growth depends upon $\mathbf{Q}$ and Φ_k. The state uncertainties are propagated by

$$\hat{\mathbf{x}}_{k+1}^{-} = \Phi_k \hat{\mathbf{x}}_k^{+} + \mathbf{B}_k \mathbf{u}_k \tag{2.22}$$

where the plus or minus sign, $\hat{\mathbf{x}}_{k+1}^{-}$ $\hat{\mathbf{x}}_k^{+}$, denotes whether or not a measurement has been incorporated in to the state estimates. The state estimate uncertainty, $\mathbf{P}$, is propagated as

$$\mathbf{P}_{k+1}^{-} = \Phi_k \mathbf{P}_k^{+} \Phi_k^{T} + \mathbf{Q}_k \tag{2.23}$$

Once the state estimates have been propagated, a prediction of what the sensor measurements should be can be calculated. This is done by transforming the state estimates from state-space to the measurement space through the observation matrix.

$$\mathbf{z}_{est} = \mathbf{H}\hat{\mathbf{x}}^{-} \tag{2.24}$$

By subtracting the measurement prediction $\mathbf{z}_{est}$ from the realized measurement, a residual is calculated. The residual is then used to update the filter. Its impact on the state estimate depends on the Kalman gain $\mathbf{K}$.

The Kalman gain is a weighting for the state estimates and the sensor measurements. If the Kalman gain is low, the filter will depend more on the state estimates, which come from the system model, than on the sensor measurements. If the Kalman gain is high; the filter will depend more on the sensor measurements than on the state estimates. The Kalman gain is calculated by taking into account both the state estimate uncertainty and the sensor measurement uncertainty. This is seen in the following equation:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{2.25}$$

where $\mathbf{R}_k$ is the sensor noise covariance matrix.

When a sensor measurement is available, the new information is included in the filter by an update. The Kalman filter update equations are shown below [9].

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_{meas} - \mathbf{H}_k \hat{\mathbf{x}}_k^-] \tag{2.26}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- \tag{2.27}$$

The performance of a Kalman filter can be "tuned" by adjusting the process noise covariance matrix, $\mathbf{Q}$, and the sensor noise covariance matrix, $\mathbf{R}$. These matrices are set based upon knowledge of the system model and the quality of sensors used by the filter.

2.7 Extended Kalman Filter

As stated earlier, when the system and measurement models are linear and the disturbances to the system and sensors are zero-mean, white Gaussian noise sources, a Kalman filter is an optimal estimator of the system states. In many instances the use of

a linear system model is not valid. The extended Kalman filter (EKF) was developed to estimate states of nonlinear systems. A nonlinear system has the form [10]

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) + \mathbf{G}\mathbf{w}(t) \tag{2.28}$$

An EKF takes the nonlinear state and/or measurement dynamics and linearizes them around a nominal trajectory or operating condition. This linearization is a Taylor series expansion where all terms higher than second order are disregarded. A sample linearization of the function $\boldsymbol{f}$ is given by:

$$\mathbf{A}[t_i, \mathbf{x}(t_i^-)] = \frac{\partial \boldsymbol{f}[\mathbf{x}, \mathbf{u}, t]}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}(t_i^-), \overline{\mathbf{u}}} + H.O.T. \tag{2.29}$$

The linearized models are only valid about the current operation condition; therefore, a linearization is performed at each time step.

One key difference between a traditional Kalman filter and an EKF is that the state estimates are propagated through the nonlinear dynamics, not the state transition matrix, i.e.,

$$\hat{\mathbf{x}}(t_i^-) = \hat{\mathbf{x}}(t_{i-1}^+) + \int_{t_{i-1}}^{t_i} \boldsymbol{f}[\hat{\mathbf{x}}(t/t_i), \mathbf{u}(t), t] dt \tag{2.30}$$

The propagation of the covariance matrix $\mathbf{P}$ does not change and is still performed by Equation (2.23). The state transition matrix Φ at time t_i is calculated by evaluating the linearized state dynamics matrix with the state estimates at time t_i. Because Φ depends on the current state estimate, a covariance analysis cannot be performed without actual sensor measurements.

If the measurement model is also nonlinear, at each update the observation matrix $\mathbf{H}$ is also linearized about the current state estimates, i.e.,

$$\mathbf{H}[t_i, \mathbf{x}(t_i^-)] = \frac{\partial h[\mathbf{x}, t]}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}(t_i^-)} \tag{2.31}$$

Because of its proven performance, the EKF has become the industry standard and will be the filter used in this thesis. The implementation and methodology of the EKF in the navigation filter will be discussed in the following chapter.

2.8 Image Aiding Techniques

As mentioned in Section 2.5, an INS must be aided in order to bound the uncertainty caused by gyro drift and measurement noise. This research will focus on aiding the inertial sensors through vision. The image-aiding algorithm is based on the research presented in [2]. An overview of this algorithm is shown in Figure 2.9.

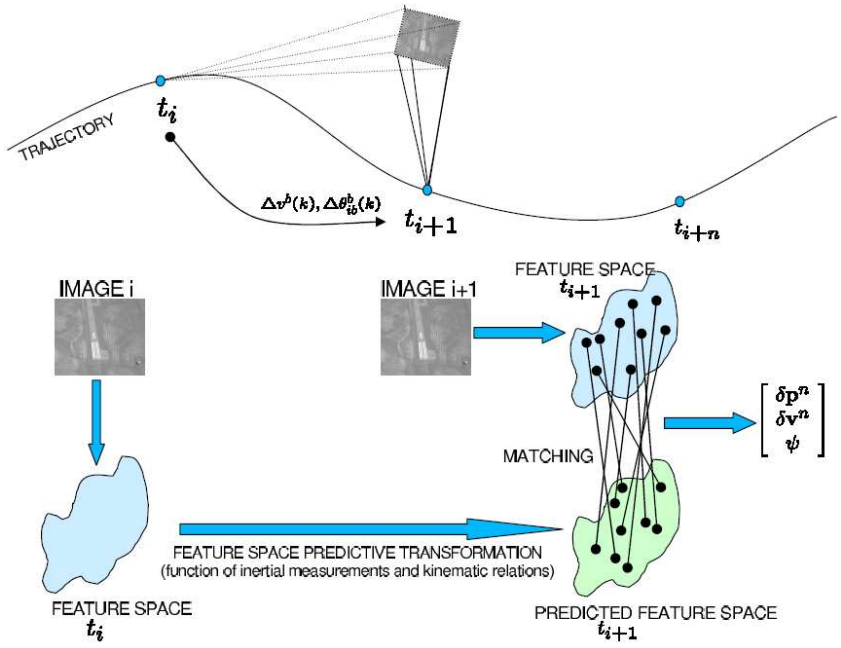

Figure 2.9. Overview of the image inertial algorithm [2]. Features are extracted at time t_i and propagated to the next time epoc with inertial data. The propagated features are then compared to the current features.

As seen in Figure 2.9, the image inertial algorithm consists of six steps:

1. Image capture
2. Transformation from the image space to the feature space
3. Propagation of the navigation state to the next image capture
4. Feature space propagation to the next image capture
5. Statistical feature correspondence
6. Navigation state error estimation and correction

1. In the first step, an image is captured. As discussed in Section 2.4, an image consists of an array of pixels, which is a projection of a 3-dimensional scene onto a 2-dimensional plane. Each image must be captured at a known time relative to the IMU data. Once the image is captured and time tagged, features can be extracted.

2. There are a number of different ways to extract features in an image ([11, 12, **?**, 13]); however, the ideal feature transformation would be one that can decompose a feature into an object and its pose, meaning position and attitude. This feature space parallels human perception. Humans see an object, and no matter where that object is or how it is oriented, it does not change the object. A pencil is a pencil, no matter what direction it's pointing or where it is. The scale invariant feature transformation (SIFT) [14], which will be discussed in the next section, cannot completely separate an object and its pose (there are no current transformations that can); however, it can account for scale and rotations of an object. Thus, the feature space consists of keypoints and descriptors. This is the method used in this research for feature extraction.

3. After the image has been transformed to the feature space, the navigation states are propagated using the methods outlined in Section 2.5 and 2.6, to the next image capture. This allows the relative changes in position and attitude to be estimated. As described in Section 2.6 the uncertainty covariance is also propagated to the next image capture.

4. The estimated changes in the navigation states are used to predict the locations of features in the next image. This is done through a stochastic projection transforma-

tion. This transformation takes a keypoint of a feature, which describes the position, and calculates a new position and position uncertainty, as shown in Figure 2.10. The position uncertainty is used to constrain the feature match.

5. The projected features are compared to the features extracted in the current frame using statistical weighting. The statistical weighting is done by calculating the Mahalanobis distance of the projected and measured feature vectors. For a derivation of the Mahalanobis distance, see [15].

6. Once the feature correspondence is done, measurements of the errors between the projected feature positions and the measured positions are used to update the navigation state vector. The specific implementation of the measurement corrections is discussed in Section 3.5.

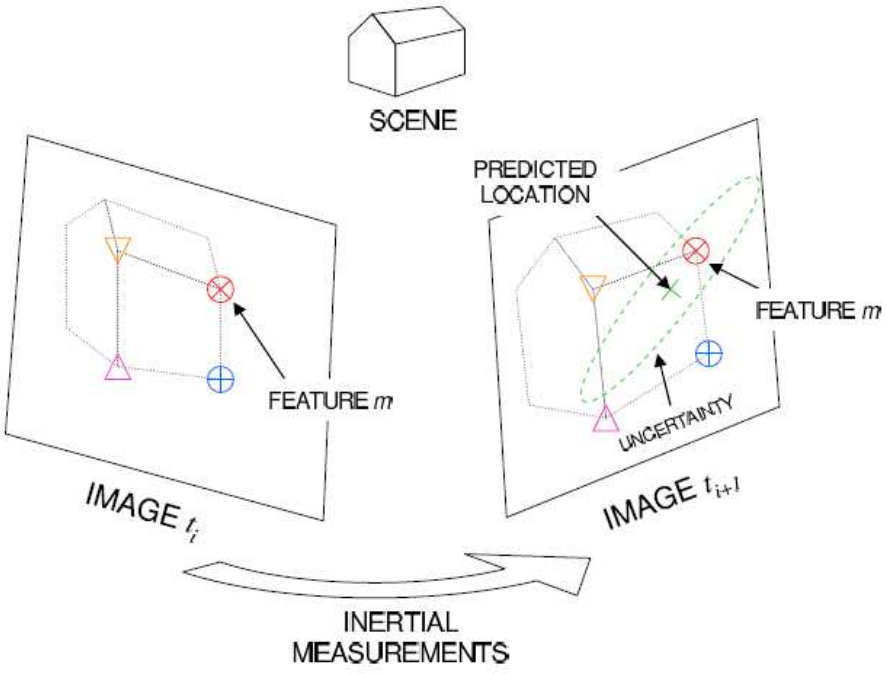

Figure 2.10. Feature are projected into the next image based on inertial data. The projected features have an associated uncertainty ellipse [2].

While it is important to understand the image-inertial algorithm as a whole, the major contribution of this research is in the feature localization and measurement update. It's important, therefore, to have a clear understanding of what a feature is and how the feature space is calculated.

2.9 Scale Invariant Feature Transformation

As mentioned in the previous section, the scale invariant feature transformation creates a feature space that consists of two main components, keypoints and descriptors. The keypoint contains the feature's location in the image plane and the feature's scale and orientation. The descriptor is a vector of 128 elements. These elements make up a set of 8 orientation histograms. Both keypoints and descriptors will be explained further in the following sections.

There are three steps in the SIFT algorithm: scale-space decomposition, extrema detection, and the feature descriptor calculation.

2.9.1 Scale-Space Decomposition. The first step in extracting features is to decompose the image into scale-spaces. This is done by convolving an image, $\mathbf{i}(x, y)$, with a set of Gaussian functions. This causes a blurring effect that can be seen in Figure 2.11. The Gaussian function, known as a Gaussian spatial filter, is defined as [14]

$$\mathbf{G}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2.32}$$

where σ is the standard deviation of the blurring function. A difference of Gaussians is expressed as

$$\mathbf{D}(x, y, k, \sigma) = \Big[\mathbf{G}(x, y, k\sigma) - \mathbf{G}(x, y, \sigma)\Big] \tag{2.33}$$

where k is the multiplier that determines the scale-space.

The resulting collection of images

$$\mathbf{I}(x, y, k\sigma) = \mathbf{D}(x, y, k, \sigma) * \mathbf{i}(x, y) \tag{2.34}$$

are placed together into octaves. An octave is a set of images such that the multiplier k is defined as

$$k = 2^{\frac{1}{s}} \tag{2.35}$$

where s is the number of images. Each octave contains twice the scale-space as the octave below it. Figure 2.12 shows how the difference of Gaussians are grouped in to octaves.

2.9.2 Extrema Detection. Once the image is decomposed into scale-spaces, local maximums and minimums are found by comparing each point with its neighboring points. See Figure 2.13. Note that each point is compared to the surrounding points, and the point directly above and below its scale-space.

Once candidate features are located, their spatial details are found by calculating the eigenvalues of the matrix A defined as [2]

$$\mathbf{A} = \begin{bmatrix} \sum_{x,y\in w}(\nabla f_x)^2 & \sum_{x,y\in w}\nabla f_x \nabla f_y \\ \sum_{x,y\in w}\nabla f_x \nabla f_y & \sum_{x,y\in w}(\nabla f_y)^2 \end{bmatrix} \tag{2.36}$$

where w is a window centered on the candidate feature. ∇f_x ,∇f_y are the gradients of the scale space image in the x and y directions. The eigenvalues are compared against a threshold. If the eigenvalue is greater than the threshold, its location (x, y, σ) is extracted and added to the keypoints.

2.9.3 Feature Descriptor Calculation. The last step is to assign the keypoint a direction. This direction is calculated by first creating a histogram of 36 bins of orientation gradients around the keypoint. The 3 highest value bins are used to interpolate the peak and that direction is then selected as the keypoint's orientation. The descriptor is calculated by finding the gradient magnitudes of the pixels surrounding the keypoint. The regions around the keypoint are subdivided into regions and orientation histograms are calculated for each region. In this way, the SIFT algorithm provides the location and orientation of a keypoint and the orientations of the regions immediately around the keypoint.

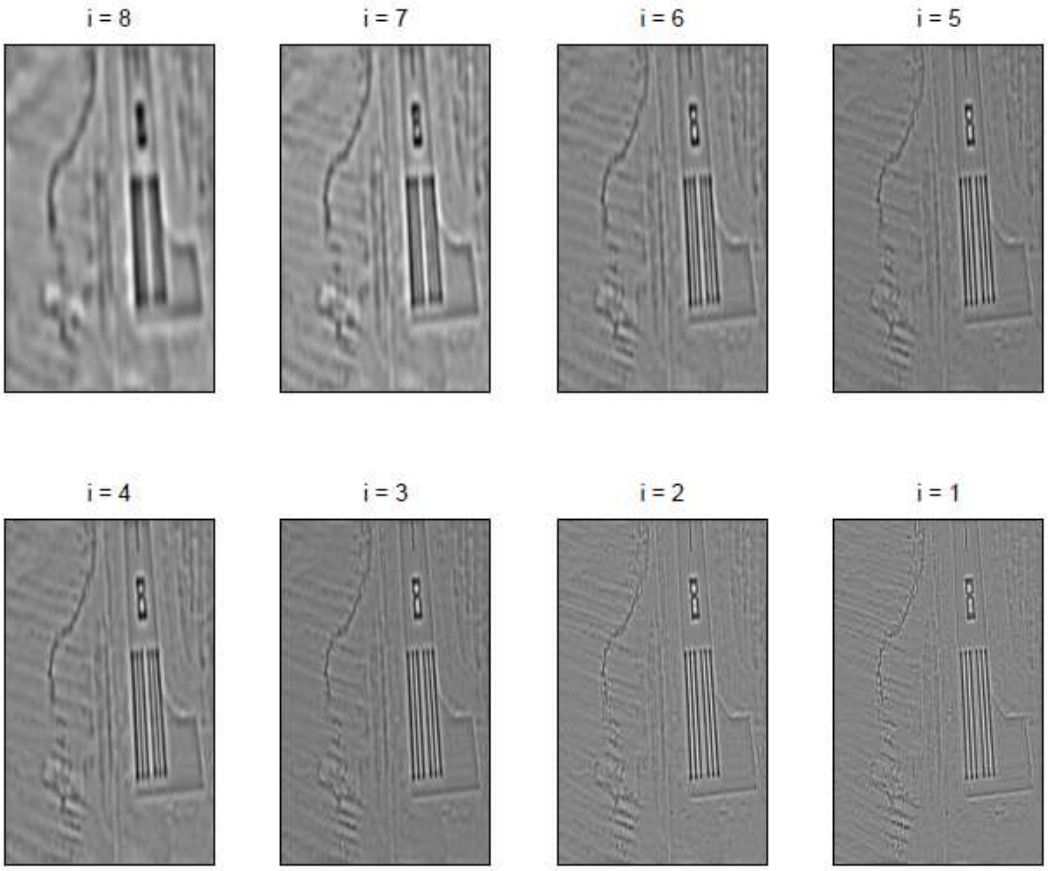

Figure 2.11. An image passed through a Gaussian spatial filter. As the center frequency of the filter increases (decreasing i), the images show increased sensitivity to higher spatial frequency detail [2].

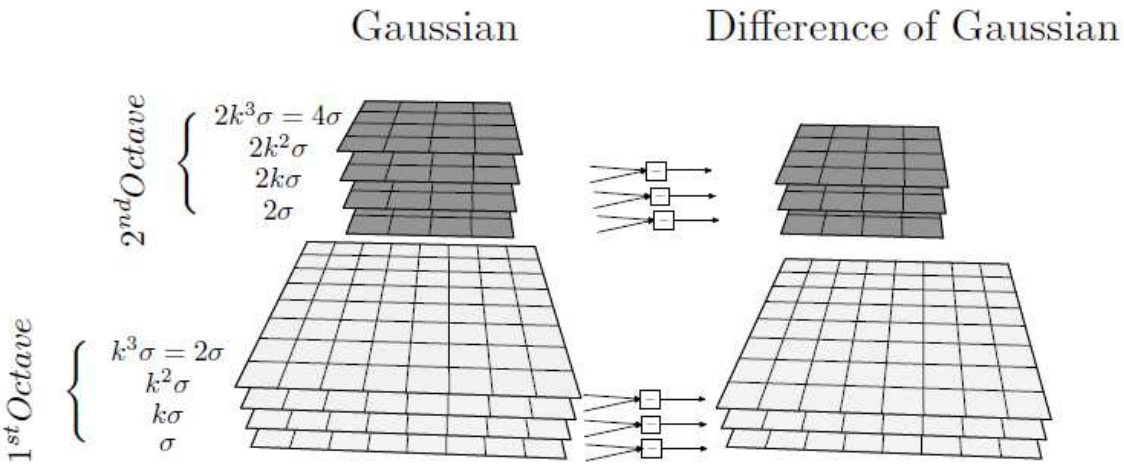

Figure 2.12. Octaves of the Difference of Gaussian Functions over a Scale-Space. Gaussian blurred images using the variable scale σ are stacked on the left. On the right are the image differences of the blurred images [16].

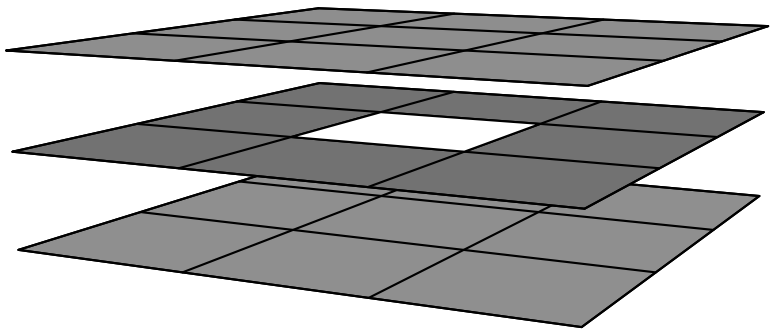

Figure 2.13. Possible Keypoint. This sample point is a local maximum among its immediate neighbors [16].

2.10 Laser Range Sensors

A SIFT feature will only provide two dimensions of a 3-dimentional object. In order to calculate the third dimension another sensor must be used. In this research, the extra sensor is a laser ranger.

Laser range sensors operate in similar fashion to sonar. Just as a sonar sensor emits sound waves and measures the time it takes the wave to return, laser range sensors emit a light wave and time how long it takes for the light wave to return. In this way, each sensor can determine distance, since objects that are far away will have a longer return time than close objects. Laser and sonar sensors can also determine distance based on the reflected wave's phase shift.

There are, however, two important differences. First, because light rays have much smaller wavelengths, laser range sensors have an extremely narrow field of view and are capable of much higher resolution. Second, because light travels much faster than sound waves, laser range sensors are capable of much faster sampling rates. Because of their high resolution and fast sampling rates, laser range sensors are becoming more and more widely used in robotic navigation research.

2.11 Stereo verses Laser

A common technique to estimate depth with a MAV is to use a stereo camera system. This section compares general stereo systems and a monocular system with a GLRS.

Stereo camera systems use two cameras to estimate depth. This approach is similar to human vision. While stereo camera systems are common, a major drawback with current technology is processing time. If the system is using feature extraction to locate and track features, images from both cameras need to be processed and compared with each other. This doubles the time required to process images, compared to a monocular system with laser depth. This means that faster, more powerful, and possibly larger computers maybe needed.

Another disadvantage with stereo systems is that depth estimate accuracy depends on the distance to the feature, and the baseline between the cameras. If the baseline is

small compared to the distance to a feature, the depth estimate is less accurate. With a GLRS the actual depth measurement is fixed, based on the accuracy of the laser. This assumes no errors in pointing the GLRS.

An advantage of stereo systems is weight. Depending on the application, small, lightweight cameras maybe used. A typical web camera weights only 10 grams. The weight of a GLRS would be around 40 grams, although the addition of supporting hardware, such as frame grabbers, for an additional camera may negate this benefit.

As technology improves, and computer speed increases, stereo camera systems may prove to be a better solution for most applications. However, the specific application will still determine which system offers the greatest benefits.

2.12 Related work

Much of the previous work regarding monocular vision localization involves using multiple images and knowledge of the vehicle's motion to estimate feature locations. This method, called egomotion, is presented in [17]. Although the author showed that this method can work, it does have some limitations. First, because this method is basically a binocular system over time, it suffers from the same observability issues that a stereo camera system does. Secondly, unconstrained optical methods suffer from scale ambiguity [18]. Scale ambiguity limits the observability of the vehicle motion. It is impossible to tell from two-dimensional images if a vehicle is rapidly approaching a large object, or if it is slowly approaching a small object. Both vehicle motions will look the same without outside information, such as a map.

In [19], a map of the work area is created prior to navigation. Once the map is known, the vehicle searches the camera images for straight-line correspondences to the map in order to localize its position. This is a model-based method. Model-based methods are well-suited for absolute localization. The drawbacks of this approach are that (1) it requires prior knowledge of the area and (2) the computational costs of storing a map and matching the corresponding features are high.

Another method of absolute localization that does not require prior information is simultaneous localization and mapping (SLAM). In [20], a monocular camera is used to measure feature locations and simultaneous build a map of the area. The feature localization is done by exploiting the certain geometry of the corners of a corridor. This is also a limitation of the algorithm, as it assumes flight in a corridor or similar structured room.

There is a large body of work that involves a monocular camera and scanning laser range sensors. Most of these systems involve using a one-dimensional laser; the laser is rotated to create a slit through the scene, as in [21]. Harati uses a mono camera to estimate distances to features and then cross validates these estimates with a scanning laser. This approach gives accurate depth estimates to features close to the laser slit, but the measurement uncertainty grows as the features move away from the laser measurements. In [22], a 360 degree scanning laser is combined with a monocular camera to accurately localize a vehicle. Although the system proved to be very precise, its large size and weight are not well-suited for use on small aerial vehicles.

A smaller sensor package is presented in [23]. In that article, a Swiss Ranger camera provides depth measurements by means of a 3D point cloud. The Swiss Ranger uses modulated IR light and measurements of the reflected light phase shift to estimate distances. The 3D point cloud of depth measurements are then compared to pixel locations in a monocular camera. This system is capable of creating a dense 3D map of the environment, which is then used for vehicle localization. Unfortunately, this system is still too large for use on small MAVs.

Many elaborate solutions have been developed to solve the monocular depth estimate problem, but the question that this research tries to answer is, does a more simple and straightforward solution exist? Can depth estimates be obtained with a simple, direct range measurement?

This research will use an understanding of the material presented thus far to develop a feature localization algorithm. A detailed development of the feature localization algorithm is presented in the next chapter.

III. Methodology

In this section, the navigation filter presented in [2] is shown and adaptations to the filter are discussed. This is followed by an overview of the feature location estimation algorithm and the assumptions made by the algorithm are stated. Next, the method used to calculate the gimbal line of sight vector is discussed, followed by a development of the measurement model. Finally, measurement model errors are analyzed.

3.1 Navigation Filter

The navigation filter used in this research is based upon the work presented in [2]. A block diagram of the navigation filter can be seen in Figure 3.14. The navigation filter tracks the location of stationary features to estimate and remove errors in the INS. The INS is then used to aid the feature tracker.

3.2 System Model

The system dynamics are captured using error models. For a full description of the error models, see [2]. It is important to note that error models were chosen in order to reduce errors caused by linearization of the measurement models. The error dynamics can be expressed as a linear, stochastic, state-space model driven by white Gaussian noise [9]:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{3.37}$$

The error dynamics $\delta\mathbf{x}(t)$ correspond to an accelerometer and gyroscope model that have biases and random noise. The biases are modeled as first-order Gauss-Markov processes [9]. The biases are estimated through the error state vector.

The error state vector $\delta\mathbf{x}(t)$ is comprised of fifteen base elements: position errors ($\delta\mathbf{p}^n$), velocity errors ($\delta\mathbf{v}^n$), attitude errors ($\boldsymbol{\psi}$), accelerometer bias errors ($\delta\mathbf{a}^b$), and gyroscope bias errors ($\delta\mathbf{b}^b$) with additional states for tracked features. The base error state

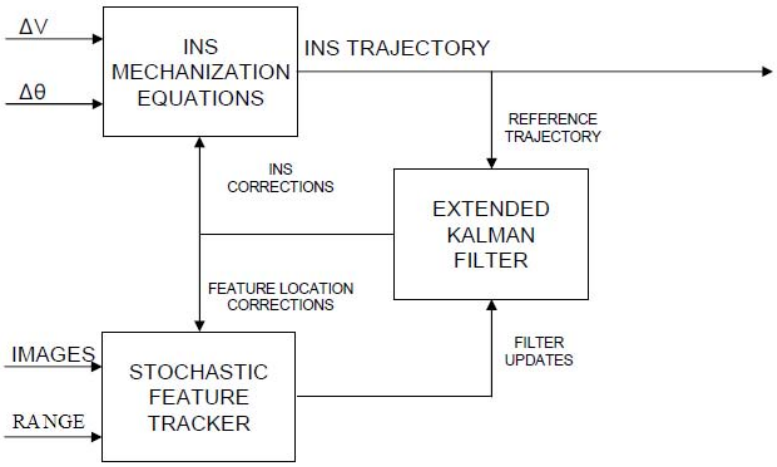

Figure 3.14. The navigation filter tracks the location of stationary features to estimate and remove errors in the INS. The INS is then used to aid the feature tracker [2].

vector is expressed as

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p}^n \\ - - - \\ \delta\mathbf{v}^n \\ - - - \\ \boldsymbol{\psi} \\ - - - \\ \delta\mathbf{a}^b \\ - - - \\ \delta\mathbf{b}^b \end{bmatrix}_{15\times 1} \tag{3.38}$$

The driving white noise vector, $\mathbf{w}$, consists of the noise associated with the accelerometer measurement ($\mathbf{w}_a^b$), gyro measurement ($\mathbf{w}_b^b$), accelerometer bias ($\mathbf{w}_{abias}^b$), and

gyro bias ($\mathbf{w}^b_{bbias}$). $\mathbf{w}$ is expressed as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^b_a \\ - - - \\ \mathbf{w}^b_b \\ - - - \\ \mathbf{w}^b_{a_{bias}} \\ - - - \\ \mathbf{w}^b_{b_{bias}} \end{bmatrix}_{12\times 1} \tag{3.39}$$

The overall error state dynamics in augmented form are [2]

$$\begin{aligned} \delta\dot{\mathbf{x}} = & \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - & - - - \\ \mathbf{C}^n_e\mathbf{G}\mathbf{C}^e_n & -2\mathbf{C}^n_e\boldsymbol{\Omega}^e_{ie}\mathbf{C}^e_n & (\mathbf{f}^n\times) & \mathbf{C}^n_b & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - & - - - \\ \mathbf{0}_3 & \mathbf{I}_3 & -(\mathbf{C}^n_e\boldsymbol{\omega}^e_{ie})\times & \mathbf{0}_3 & -\mathbf{C}^b_n \\ - - - & - - - & - - - & - - - & - - - \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & -\frac{1}{\tau_a}\mathbf{I}_3 & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - & - - - \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\frac{1}{\tau_b}\mathbf{I}_3 \end{bmatrix}_{15\times 15} \delta\mathbf{x} \\ + & \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - \\ \mathbf{C}^n_b & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - \\ \mathbf{0}_3 & -\mathbf{C}^n_b & \mathbf{0}_3 & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ - - - & - - - & - - - & - - - \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}_{15\times 12} \mathbf{w} \end{aligned} \tag{3.40}$$

Note that $(\times)$ denotes skew symmetric form (see Equation (2.4)). For a complete description of the filter development, see [2]. In this research, the only changes to the navigation filter are in the measurement model, which is presented in Section 3.4.

3.3 Feature Location Estimation

The feature localization is a five-step process.

1. Image capture and feature transformation
2. Feature selection
3. LOS vector calculation and correction
4. LOS scaling
5. Calculation of feature position

In the first step, an image is captured and features are extracted through SIFT. Once all the SIFT features are found, candidate features are selected based upon the feature's distinctiveness, spatial separation, and recognizability. If another feature needs to be added to the state vector, the LOS vector is calculated as discussed in Section 3.3. The LOS vector is used to steer the GLRS at the feature, and the range to the feature is returned. The range measurement is used to scale the homogeneous LOS vector. Once the LOS vector is scaled properly, it is transformed into the navigation frame. Then the position of the feature is calculated based upon the LOS vector and the current navigation state estimates. This approach is based on a few assumptions. These assumptions are listed below.

- An IMU is available and is rigidly mounted relative to a digital camera.
- The time images are taken is known relative to IMU measurements.
- The initial navigation states are known along with their statistics.
- There is sufficient alignment time for initial feature extraction.
- Tracked features are stationary.
- The IMU, camera, and GLRS are co-located.

3.4 Line of Sight Vector Calculations

In order calculate the position of a feature, an accurate depth measurement to the feature is needed. A GLRS can provide this depth measurement, provided it can be pointed at the feature.

3.4.1 Rotation Calculation. A camera provides a homogeneous line of sight vector $\underline{\mathbf{s}}^c$ to the feature. This line of sight vector can be calculated from Equation (2.8). It is repeated here for clarity.

$$\mathbf{z}_u = \frac{1}{s_z^c}\mathbf{T}_c^{pix}\mathbf{s}^c$$

Taking the inverse of $\mathbf{T}_c^{pix}$ yields

$$\mathbf{s}^c = s_z^c\mathbf{T}_{pix}^c\mathbf{z}_u \tag{3.41}$$

This transformation takes the undistorted (x, y) pixel location, $\mathbf{z}_u$, and creates a line of sight vector towards the feature in the camera frame. $\mathbf{s}^c$ is then normalized to create the homogeneous LOS vector, $\underline{\mathbf{s}}^c$.

$$\underline{\mathbf{s}}^c = \begin{bmatrix} \underline{s}_x^c \\ \underline{s}_y^c \\ 1 \end{bmatrix} \tag{3.42}$$

In order to point the GLRS at a feature, the homogeneous LOS vector must be rotated into the gimbal frame. If it is assumed that the GLRS, IMU, and camera are co-located, then the transformation between frames consists only of rotations. To rotate the homogeneous LOS vector into the gimbal frame, it must first be rotated from the camera frame to the body frame. This is done by

$$\underline{\mathbf{s}}^b = \mathbf{C}_c^b\underline{\mathbf{s}}^c \tag{3.43}$$

where

$$\mathbf{C}_c^b = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.44}$$

To move from the body frame to the gimbal frame the following transformation is used:

$$\underline{\mathbf{s}}^g = \mathbf{C}_b^g \underline{\mathbf{s}}^b \tag{3.45}$$

where

$$\mathbf{C}_b^g = \begin{bmatrix} cos(el) & 0 & sin(el) \\ 0 & 1 & 0 \\ -sin(el) & 0 & cos(el) \end{bmatrix} \begin{bmatrix} cos(az) & sin(az) & 0 \\ -sin(az) & cos(az) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.46}$$

The elevation, el, and azimuth, az, angels can be measured based on the commanded pulse width to the gimbal servos [3]. Equation (3.46) can be re-written as

$$\mathbf{C}_b^g = \begin{bmatrix} cos(el)cos(az) & cos(el)sin(az) & sin(el) \\ -sin(az) & cos(az) & 0 \\ -sin(el)cos(az) & -sin(el)sin(az) & cos(el) \end{bmatrix} \tag{3.47}$$

3.4.2 Offset Correction. The vector $\underline{\mathbf{s}}^g$ is the LOS vector pointing at the feature in the gimbal frame. It is important to note that any disparity that exists between the camera and the GLRS will cause a offset between the GLRS beam and the feature location, regardless of distance. An illustration of this is shown in Figure 3.15.

A method to remove this offset is to use the range reading (z_l) from the GLRS to adjust the azimuth and elevation commands to the gimbal.

$$c_{offset} = \mathbf{f}(z_l); \tag{3.48}$$

The correction term, c_{offset}, is a function of distance only, and can be found by calibration. Once the initial range to the feature is determined the correction can be made and the gimbal LOS vector can be recalculated. If both the vehicle and the feature are stationary, $\underline{\mathbf{s}}^g$ can now be used to calculate the feature's position by scaling it with the range measurement. The next section discusses accounting for vehicle movement.

3.4.3 Correcting for Translation and Rotation. The gimbal LOS vector $\mathbf{s}^g$ is only valid at the time the image is captured if the vehicle is moving. To correct for vehicle

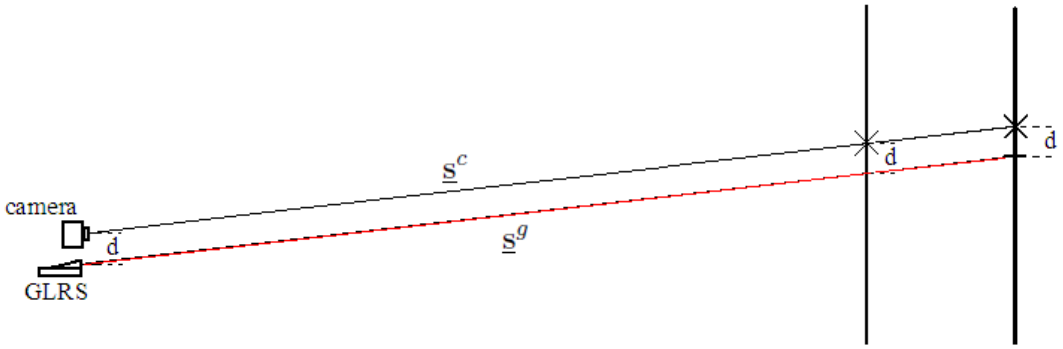

Figure 3.15. An offset between the camera and the GLRS results in an identical offset between the laser beam and the feature location.

movement, the navigation filter's estimates of rotation are used. If a two degree of freedom (azimuth and elevation) gimbal is used, then a correction for rotations can be calculated based upon the filter's current estimates of yaw,ψ_{curr}, and pitch, θ_{curr}, and the estimates of ψ_{pix} and θ_{pix} when the image was captured. The correction is applied in the body to gimbal rotation matrix, i.e.,

$$\mathbf{C}_b^g = \begin{bmatrix} cos(el_{corr})cos(az_{corr}) & cos(el_{corr})sin(az_{corr}) & sin(el_{corr}) \\ -sin(az_{corr}) & cos(az_{corr}) & 0 \\ -sin(el_{corr})cos(az_{corr}) & -sin(el_{corr})sin(az_{corr}) & cos(el_{corr}) \end{bmatrix} \tag{3.49}$$

where

$$el_{corr} = el + (\theta_{curr} - \theta_{pix})$$

$$az_{corr} = az + (\psi_{curr} - \psi_{pix})$$

Because the corrections only depend on the relative change in yaw and pitch angles, errors in the absolute measurement of ψ and θ do not affect the filters ability to correct for vehicle rotations.

Unfortunately, translations of the camera cannot be compensated for. In order to correct for translations one would need to know the location of the feature, which is being estimated. An illustration of this problem is shown in Figure 3.16. However, errors due to translation will have a larger effect on objects that are close than objects at a distance. If you're traveling in a car looking at a distant mountain, it appears stationary. On the other hand, a near traffic light appears to move much faster. Using this knowledge a possible solution to dealing with camera translations would be to test the ratio of velocity to feature distance measured by the GLRS. By dividing the vehicle velocity by the distance to the feature, an approximation of the measurement error can be found. If the velocity is large compared to the distance to the feature, then the resulting error will be large. If the velocity is small compared to the distance to the feature, then the error will be small. The resulting error calculation can then be compared to a threshold; if the error is too large a new feature can be selected. Because of implementation restrictions, this test will not be explored in this research. However, because it only takes roughly 5 ms/deg to steer the gimbal [24], and the dynamics of the test platform are comparatively slow, the errors due to translation are assumed to be small.

3.5 Measurement Model

Using the methods described in Sections 3.3 and 3.3.1, the distance from a feature to the center of the gimbal/body frame can be calculated. To calculate the position of the feature, the filter's estimate of the vehicle's position and the vector $\underline{\mathbf{s}}^b$, properly scaled and rotated into the navigation frame, are used. The GLRS provides the scalar z_l, which is the distance from the center of the gimbal/body frame to the feature, and it used to scale $\underline{\mathbf{s}}^b$.

The vector $\underline{\mathbf{s}}^g$ is need in order to point the GLRS at a feature. However, to calculate the feature's position in the navigation frame, it is necessary to find $\mathbf{s}^n$, which can be done by scaling and rotate the vector $\underline{\mathbf{s}}^b$ into the navigation frame.

The vector $\mathbf{s}^b$ is calculated by scaling $\underline{\mathbf{s}}^b$ by the distance to the feature, z_l.

$$\mathbf{s}^b = z_l \underline{\mathbf{s}}^b \tag{3.50}$$

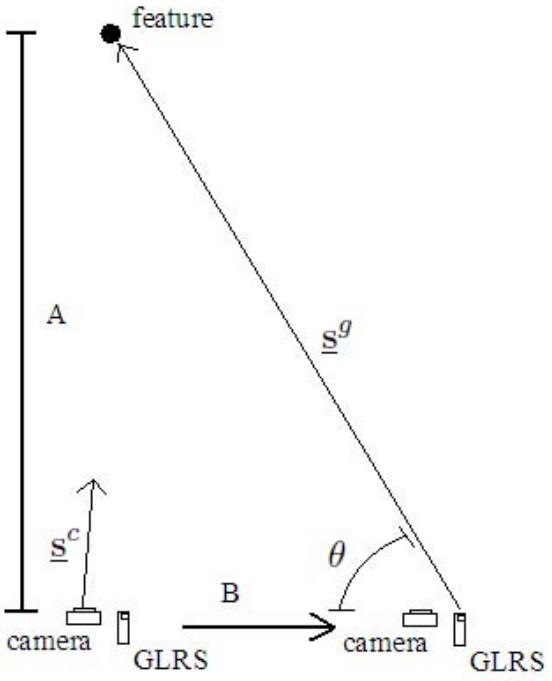

Figure 3.16. In order to find the angle θ to correct for the translation along **B**, the initial distance to the feature **A** would need to be known.

The rotation of $\mathbf{s}^b$ into the navigation is accomplished by

$$\mathbf{s}^n = \mathbf{C}_b^n \mathbf{s}^b \tag{3.51}$$

With the assumption that the IMU, camera, and GLRS are co-located, the feature's position $\mathbf{t}^n$ can be found by vector addition.

$$\mathbf{t}^n = \mathbf{p}^n + \mathbf{s}^n \tag{3.52}$$

Figure 3.17 shows an illustration of the geometry of the measurement.

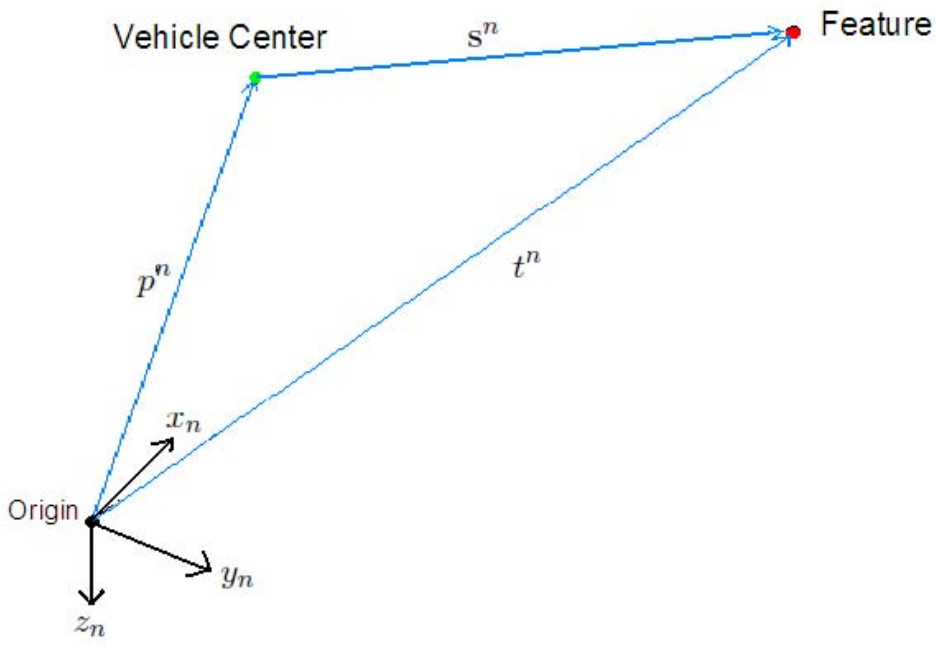

Figure 3.17. Feature and vehicle position geometry shown in the navigation frame. The feature location is found by vector addition.

Equation (3.52) can be written as

$$\mathbf{t}^n = \mathbf{p}^n + \mathbf{C}_b^n \mathbf{C}_c^b \mathbf{T}_{pix}^c z_l \underline{z}_u \tag{3.53}$$

It is important to realize that the measurement of the feature location comes from two different sensors. The camera provides the line of sight vector, and the GLRS provides the proper scaling. Both of these measurements have errors associated with them. To add the feature location to the EKF, the sensor needs to be modeled and the uncertainties of the measurements need to be quantified. The measurement from the camera can be modeled as

$$\underline{\mathbf{z}}_u = \mathbf{T}_c^{pix}\underline{\mathbf{s}}^c + \mathbf{v}_{cam} \tag{3.54}$$

where $\mathbf{v}_{cam}$ is a zero-mean, white Gaussian noise source with units in pixels. $\underline{\mathbf{s}}^c$ is the homogenous form of $\mathbf{s}^c$, where

$$\mathbf{s}^c = \mathbf{C}_b^c\mathbf{C}_n^b[\mathbf{t}^n - \mathbf{p}^n] \tag{3.55}$$

The laser range sensor can be modeled as

$$z_l = s_z^c + v_{zlaser} \tag{3.56}$$

where s_z^c is the z component of the line of sight vector in the camera frame, and v_{zlaser} is a zero-mean, white Gaussian noise source with units in meters.

Once the position of a feature has been initialized, its estimated location is added to the navigation state. By using the vehicle's position estimate and the estimated location of the feature, Equation (3.54) can be used to calculate an estimated LOS vector. This estimate, along with the actual measurement provided by the camera, is then used to update the filter states.

Both the camera model and the laser range sensor model are functions of the vehicle position, the attitude of the vehicle, and the feature location, i.e.,

$$\underline{\mathbf{z}}_u = \mathbf{f}(\mathbf{p}^n, \boldsymbol{\psi}, \mathbf{t}^n) \tag{3.57}$$

$$z_l = \mathbf{j}(\mathbf{p}^n, \boldsymbol{\psi}, \mathbf{t}^n) \tag{3.58}$$

In order to use the nonlinear measurement models above in an EKF, the observation matrix, $\mathbf{H}$, must be linearized about the current state estimate. This linearization is done by taking the partial derivatives of the measurement model with respect to the measurement parameters, $\mathbf{p}^n$, $\boldsymbol{\psi}$ and, $\mathbf{t}^n$,.

The partial derivative of Equation (3.54) with respect to $\mathbf{p}^n$ is

$$\mathbf{H}_{\underline{\mathbf{z}}_u \mathbf{p}^n} = \frac{\partial \underline{\mathbf{z}}^u}{\partial \underline{\mathbf{s}}^c} \frac{\partial \underline{\mathbf{s}}^c}{\partial \mathbf{s}^c} \frac{\partial \mathbf{s}^c}{\partial \mathbf{p}^n} \tag{3.59}$$

where

$$\frac{\partial \underline{\mathbf{z}}_u}{\partial \underline{\mathbf{s}}^c} = \mathbf{T}_c^{pix} \tag{3.60}$$

and

$$\frac{\partial \underline{\mathbf{s}}^c}{\partial \mathbf{s}^c} = \frac{1}{(s_z^c)^2} \begin{bmatrix} -f_x & 0 & c_x \\ 0 & f_y & -c_y \\ 0 & 0 & 0 \end{bmatrix} \tag{3.61}$$

and

$$\frac{\partial \mathbf{s}^c}{\partial \mathbf{p}^n} = \mathbf{C}_b^c \mathbf{C}_n^b \tag{3.62}$$

The partial derivative of Equation (3.54) with respect to $\boldsymbol{\psi}$ is

$$\mathbf{H}_{\underline{\mathbf{z}}_u \boldsymbol{\psi}} = \frac{\partial \underline{\mathbf{z}}_u}{\partial \underline{\mathbf{s}}^c} \frac{\partial \underline{\mathbf{s}}^c}{\partial \mathbf{s}^c} \frac{\partial \mathbf{s}^c}{\partial \boldsymbol{\psi}} \tag{3.63}$$

where

$$\frac{\partial \mathbf{s}^c}{\partial \boldsymbol{\psi}} = \frac{\partial}{\partial \boldsymbol{\psi}} \mathbf{C}_b^c \mathbf{C}_n^b [I + \boldsymbol{\psi}(\times)](\mathbf{t}^n - \mathbf{p}^n) \tag{3.64}$$

$$= \mathbf{C}_b^c \mathbf{C}_n^b (\mathbf{s}^n \times) \quad , \quad \mathbf{s}^n = \mathbf{t}^n - \mathbf{p}^n \tag{3.65}$$

The partial derivative of Equation (3.54) with respect to $\mathbf{t}^n$ is

$$\mathbf{H}_{\underline{\mathbf{z}}_u \mathbf{t}^n} = \frac{\partial \mathbf{z}_u}{\partial \underline{\mathbf{s}}^c} \frac{\partial \underline{\mathbf{s}}^c}{\partial \mathbf{s}^c} \frac{\partial \mathbf{s}^c}{\partial \mathbf{t}^n} \tag{3.66}$$

where

$$\frac{\partial \mathbf{s}^c}{\partial \mathbf{t}^n} = \mathbf{C}_b^c \mathbf{C}_n^b \tag{3.67}$$

The partial derivative of Equation (3.56) with respect to $\mathbf{p}^n$ is

$$\mathbf{H}_{z_l \mathbf{p}^n} = \frac{\partial z_l}{\partial s_z^c} \frac{\partial s_z^c}{\partial \mathbf{p}^n} \tag{3.68}$$

where

$$\frac{\partial z_l}{\partial s_z^c} = 1 \tag{3.69}$$

and

$$\begin{aligned} \frac{\partial s_z^c}{\partial \mathbf{p}^n} &= [0 \;\; 0 \;\; 1] \frac{\partial \mathbf{s}^c}{\partial \mathbf{p}^n} && (3.70) \\ &= [0 \;\; 0 \;\; 1] \mathbf{C}_b^c \mathbf{C}_n^b && (3.71) \end{aligned}$$

The partial derivative of Equation (3.56) with respect to $\boldsymbol{\psi}$ is

$$\mathbf{H}_{z_l \boldsymbol{\psi}} = \frac{\partial z_l}{\partial s_z^c} \frac{\partial s_z^c}{\partial \boldsymbol{\psi}} \tag{3.72}$$

where

$$\begin{aligned} \frac{\partial s_z^c}{\partial \boldsymbol{\psi}} &= [0 \;\; 0 \;\; 1] \frac{\partial \mathbf{s}^c}{\partial \boldsymbol{\psi}} && (3.73) \\ &= [0 \;\; 0 \;\; 1] \mathbf{C}_b^c \mathbf{C}_n^b (\mathbf{s}^n \times) && (3.74) \end{aligned}$$

Finally, the partial derivative of Equation (3.56) with respect to $\mathbf{t}^n$ is

$$\mathbf{H}_{z_l \mathbf{t}^n} = \frac{\partial z_l}{\partial s_z^c} \frac{\partial s_z^c}{\partial \mathbf{t}^n} \tag{3.75}$$

where

$$\begin{aligned} \frac{\partial s_z^c}{\partial \mathbf{t}^n} &= [0 \;\; 0 \;\; 1] \frac{\partial \mathbf{s}^c}{\partial \mathbf{t}^n} && (3.76) \\ &= [0 \;\; 0 \;\; 1] \mathbf{C}_b^c \mathbf{C}_n^b && (3.77) \end{aligned}$$

The partials can now be evaluated at the current state estimates and the total state observation matrix $\mathbf{H}$ can be augmented with the evaluated partials.

In order to add the feature location estimate to the filter, the uncertainties of the measurements must be known. This is calculated by summing the uncertainty in the parameters in measurement Equation (3.53). This assumes parameters are independent. The uncertainty in the measurement parameters is found by calculating the measurement's influence matrices.

The calculation of the influence matrices ($\mathbf{G}$) is similar to the observation matrices. The influence matrices define how much the individual uncertainties effect the overall uncertainty of the feature location. The uncertainties that are considered in the calculation of the feature location are:

- Position uncertainty of the vehicle
- Attitude uncertainty of the vehicle
- Pixel measurement uncertainty
- Depth measurement uncertainty

All other influence matrices are considered to be zero.

The influence matrix corresponding to the position uncertainty is defined as

$$\mathbf{G}_{tp} = \frac{\partial \mathbf{t}^n}{\partial \mathbf{p}^n} \tag{3.78}$$

Evaluating the partial derivative yields

$$\mathbf{G}_{tp} = \mathbf{I}_{3\times 3} \tag{3.79}$$

This result is intuitive, as any uncertainty in the vehicle position will directly affect the uncertainty of the feature location.

The influence matrix corresponding to the attitude uncertainty is defined as

$$\mathbf{G}_{t\boldsymbol{\psi}} = \frac{\partial \mathbf{t}^n}{\partial \boldsymbol{\psi}} \tag{3.80}$$

Evaluating the partial derivative yields

$$\mathbf{G}_{t\psi} = [\mathbf{C}_b^n \mathbf{C}_c^b \mathbf{s}^c](\times) \tag{3.81}$$

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

$$\mathbf{G}_{t\mathbf{z}_u} = \frac{\partial \mathbf{t}^n}{\partial \mathbf{z}_u} \tag{3.82}$$

Evaluating the partial derivative yields

$$\mathbf{G}_{t\mathbf{z}_u} = z_l \mathbf{C}_b^n \mathbf{C}_c^b \mathbf{T}_{pix}^c \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{3.83}$$

The influence matrix corresponding to the range measurement uncertainty is defined as

$$\mathbf{G}_{tz_l} = \frac{\partial \mathbf{t}^n}{\partial \mathbf{z}_l} \tag{3.84}$$

Evaluating the partial derivative yields

$$\mathbf{G}_{tz_l} = \mathbf{C}_b^n \mathbf{C}_c^b \underline{\mathbf{s}}^c \tag{3.85}$$

The influence matrices needed by the EKF are now known. As mentioned earlier, assuming they are independent, the uncertainty in the parameters can be summed to find the total uncertainty in the feature location estimate, i.e.,

$$\mathbf{P}_{tt} = \mathbf{G}_{tp}\mathbf{P}_{pp}\mathbf{G}_{tp}^T + \mathbf{G}_{t\psi}\mathbf{P}_{\psi\psi}\mathbf{G}_{t\psi}^T + \mathbf{G}_{tz_u}\mathbf{P}_{z_u z_u}\mathbf{G}_{tz_u}^T + \mathbf{G}_{tz_l}\mathbf{P}_{z_l z_l}\mathbf{G}_{tz_l}^T + \mathbf{G}_{tp}\mathbf{P}_{p\psi}\mathbf{G}_{t\psi}^T + \mathbf{G}_{t\psi}\mathbf{P}_{\psi p}\mathbf{G}_{tp}^T \tag{3.86}$$

where $\mathbf{P}_{tt}$ is the over all uncertainty of the feature location, $\mathbf{P}_{pp}$ is the uncertainty of the vehicle position, $\mathbf{P}_{\psi\psi}$ is the uncertainty in attitude, $\mathbf{P}_{z_u z_u}$ is the uncertainty of the camera measurement, and $\mathbf{P}_{z_l z_l}$ is the uncertainty of the laser range sensor. The matrices $\mathbf{P}_{\psi p}$

and $\mathbf{P}_{p\psi}$ are cross correlation matrixes that define how the uncertainty in position effects uncertainty in attitude and vise-versa.

Using Equation (3.86) it is possible to determine the theoretical uncertainty of the feature location estimates given the uncertainties in the other measurements. This is discussed in the following section.

3.6 Error Analysis

This section uses Equation (3.86) to calculate the uncertainty in the feature location estimates. The sources of error that are considered are: vehicle position errors, attitude estimation errors, pixel measurements errors, and range measurement errors. Table 3.1 shows average uncertainties in position, attitude, camera and range measurements by a similar image-aided filter [25]. Using these values, the uncertainty in a feature's location given a nominal attitude and trajectory can be calculated.

For purpose of clarity, the body frame, navigation frame, and camera frame were aligned and collocated with each other during this analysis. The feature's location in the navigation frame was at one meter in the positive y-axis, one meter in the negative z-axis, and the depth (or position in the x-axis) was varied, as listed in Table 3.2.

Table 3.2 shows that, as the range to the feature increases, so does the uncertainty in its location. As expected, because of the accuracy of the laser measurement, the uncertainties in the y and z axis are greater than the uncertainty in the x axis. Table 3.2 also shows that at close range, less than five meters, feature locations should be accurate to within half a meter. In the following chapter, test results are presented which verify this error analysis.

Source	Uncertainty
$\mathbf{p}^n$	0.25 *meters*
$\boldsymbol{\psi}$	2 *degrees*
$\mathbf{z}_u$	1 *pixel*
z_l	0.01 *meters*

Table 3.1. Average uncertainty in position, attitude, camera and range measurements. The values were derived from previous research [25].

Feature Range	Feature Location Uncertainty
100	$\sigma_x = 3.18$, $\sigma_y = 18.68$, $\sigma_z = 18.68$
50	$\sigma_x = 1.61$, $\sigma_y = 9.34$, $\sigma_z = 9.34$
20	$\sigma_x = 0.24$, $\sigma_y = 3.74$, $\sigma_z = 3.74$
10	$\sigma_x = 0.08$, $\sigma_y = 1.88$, $\sigma_z = 1.89$
5	$\sigma_x = 0.05$, $\sigma_y = 0.46$, $\sigma_z = 0.46$

Table 3.2. Feature location uncertainties, units are in meters. Note that closer features have less uncertainty.

IV. Results

Simulated and real world data were collected and analyzed in order to determine if the methods introduced in the previous chapter are valid. This chapter details the simulation environment, real world hardware/software setup, and data collections experiments that will be used to draw conclusions in the following chapter.

First, the hardware and software that were developed and used are presented. This is followed by a simulated flight experiment. After simulation, an experiment to test the GLRS's ability to estimate feature locations is performed. Then a data collection run, using the Vicon motion capture system, is used to gather truth data to compare against the navigation filter's estimates. Finally, an eleven-minute data collection run through a hallway is performed and compared against similar stereo vision experiments.

4.1 Hardware Overview

Four main components are needed to test the monocular vision localization algorithm; a gimbaled laser range sensor, an IMU, a digital camera, and a computer. Figure 4.18 shows a block diagram of the key components and the sensor interfaces.

The key component for this research is a laser range sensor. There are many commercial laser range finders, however a sensor that has high precision (millimeter accuracy), long range (greater that 50 meters), and small size was desired. For this reason, the Fluke 416D Laser Distance Meter [26] was chosen. This sensor is accurate within approximately 1.5mm and has a range of sixty meters. Unfortunately, this sensor does not have an interface other than the LCD screen. Attempts were made to interface the unit with a micro-controller; however, these attempts were unsuccessful.

The laser range sensor was mounted to a Pandora pan and tilt gimbal [27]. This gimbal uses standard hobby-style analog servos (HS-81 and HS-65HB) to rotate the sensor. Figure 4.19 shows the laser range sensor mounted on the gimbal.

The GLRS is mounted so that its nominal x_c-axis is aligned with the camera's z_c-axis. Also, it is mounted as close as possible to the camera without impeding the gimbals' movement. The IMU is mounted behind the gimbal, again as close as possible to the camera

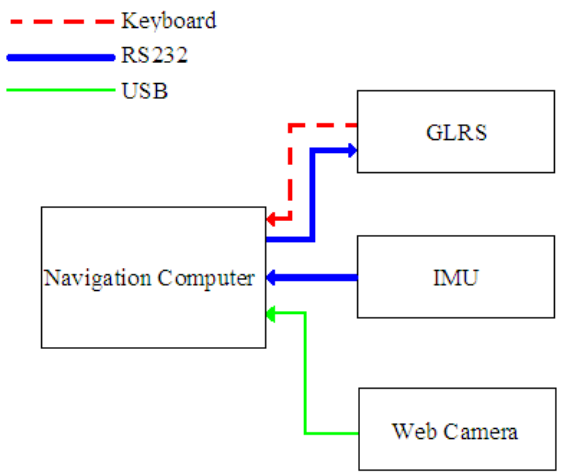

Figure 4.18. Hardware block diagram with sensor interfaces. The measurements made by the laser range sensor are recorded by the computer keyboard, due to issues with the sensor interface.

without impeding the GLRS. The IMU's x-axis is aligned with the camera's z_c-axis and the gimbals' nominal x_c-axis, as shown in Figure 4.20.

In order to reduce the weight of the system, a small commercial IMU and a lightweight web camera were chosen. While there are many commercial IMUs available the device used in this research is the Analog Devices ADIS16355 [28]. The ADIS16355 specifications are shown in Table 4.3. The web camera has a resolution of 320x240 pixels.

The total weight of the experimental setup is 195 grams, which is light enough to be carried by a quadrotor MAV such as the one in [29]. Unfortunately, because the distance measurements from the GLRS needed to be read from the LCD screen and typed into the computer, actual flight tests were not performed. An alternative approach taken was to mount the experimental setup as shown in Figure 4.19 to a cart, see Figure 4.21. An advantage of this approach is that the data from the sensors could be sent to a laptop computer without weight restrictions.

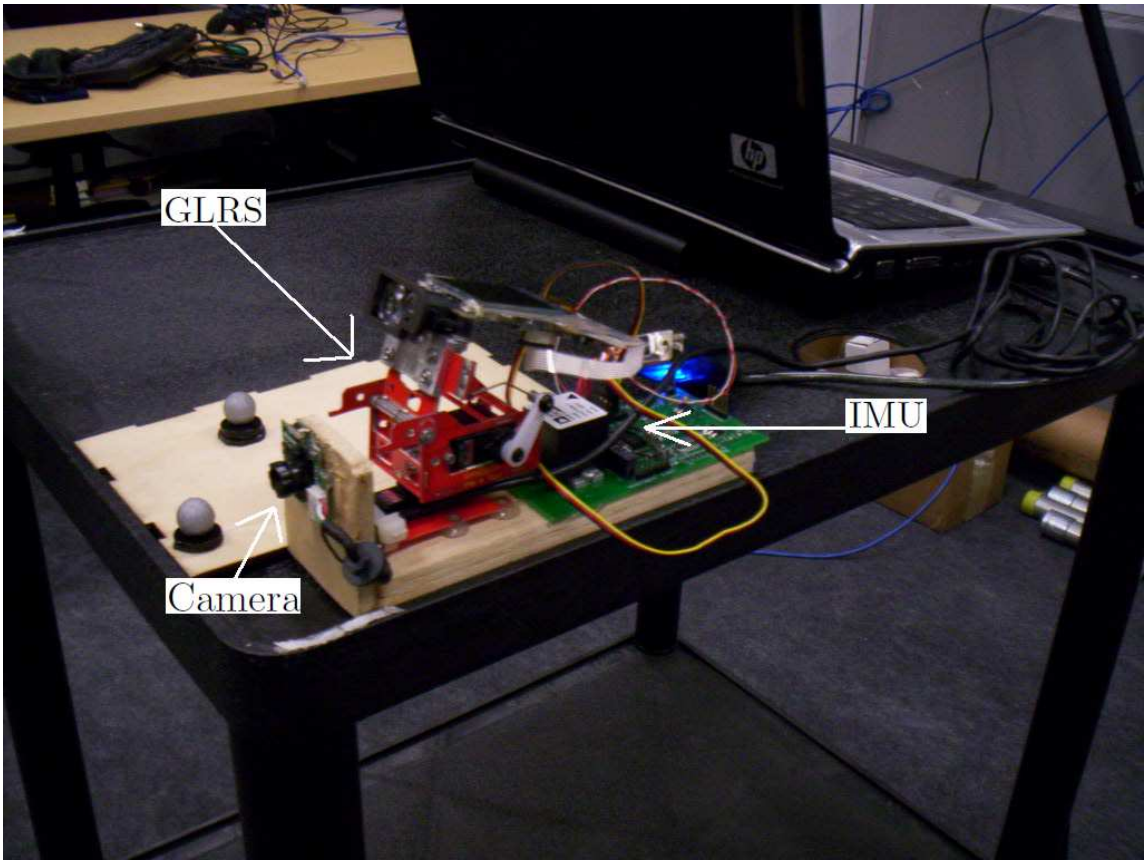

Figure 4.19. Experimental hardware. The laser range sensor is mounted to the gimbal directly above the camera and in front of the IMU.

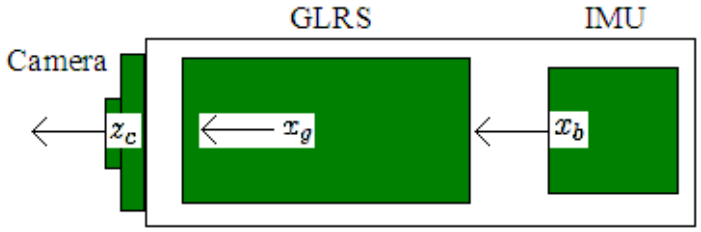

Figure 4.20. Experimental hardware alignment. The camera, GLRS, and IMU are aligned in a row.

Parameter	Value (Units)
Sampling interval	10 (ms)
Gyro bias sigma	0.015 (deg/s)
Gyro bias time constant	2^* (hr)
Angular random walk	4.2 (deg/$\sqrt{hr}$)
Accel bias sigma	0.7 (mg)
Accel bias time constant	2^* (hr)
Velocity random walk	2 (m/s/$\sqrt{hr}$)

Table 4.3. The ADIS16355 IMU specifications. An asterisk ($*$) denotes estimated values

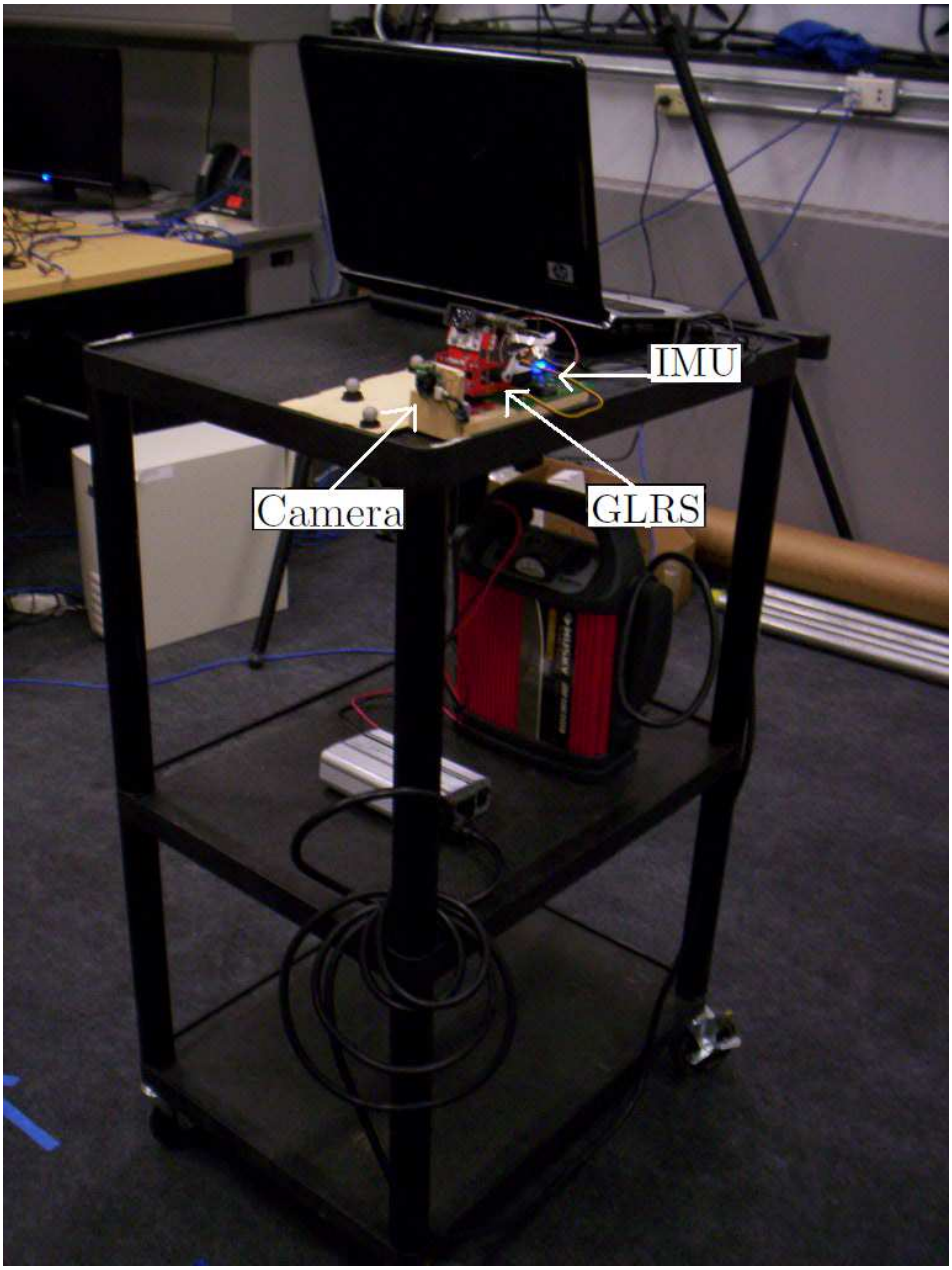

Figure 4.21. Experimental platform. Data from the camera, IMU, and GLRS are all sent to the laptop computer to be recorded.

Because flight tests were not conducted, data from the sensors was gathered by the computer and later used in the navigation filter. The images, IMU data, and range measurements were recorded using a custom-built graphical user interface (GUI). The GUI uses an open source computer vision library, OpenCV. Images were time tagged and used at roughly 2 frames per second. Using the Shi and Tomasi's method [30], features were extracted and a pointing vector was calculated to aim the GLRS at a feature. The Shi and Tomasi's method was used during data collections for increased speed, but SIFT was later used on the gathered data in the navigation filter. Once the distance to a feature was recorded, a new feature was selected. The pixel location, range data, and images were all time-tagged relative to the IMU's 100Hz clock.

4.2 Simulation Experiment

Simulated images and flight data were generated in MATLAB and the Profgen trajectory generation software [31]. In MATLAB a 40 meter hallway was created with simulated features. The Profgen software tool was then used to create a reference trajectory through the hallway. The trajectory created was for a level vehicle (zero roll, pitch, and yaw) that is stationary for 60 seconds then moves North through the hallway at 0.5 m/s until it reaches the end of the hallway. Figure 4.22 shows an illustration of the simulated hallway and trajectory.

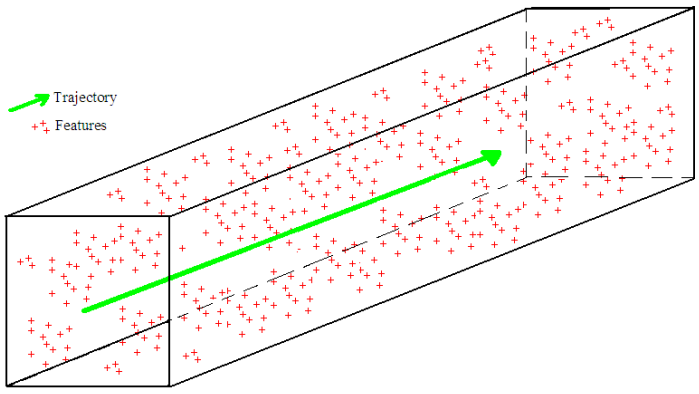

Figure 4.22. Simulated hallway. Simulated features are generated along the flight path of the vehicle.

Inertial data was generated by subjecting the true angular rates and specific force data to random measurement errors based on the IMU specifications in Table 4.3.

The simulations were based off the actual IMU and camera models that were used later for real data collections. Thus, the simulation results should be consistent with real world performance.

Once all the necessary data was generated, the simulated sensor data was processed by the navigation filter. During simulation only one feature could be added to the tracker per image, and a maximum of 12 features were tracked at any one time. Also, pixel noise was added to the simulated features to simulate optical distortions.

The performance of the navigation filter was tested using the ensemble statistics of 100 simulated hallway runs. Figures 4.23 and 4.24 show the ensemble position and attitude errors.

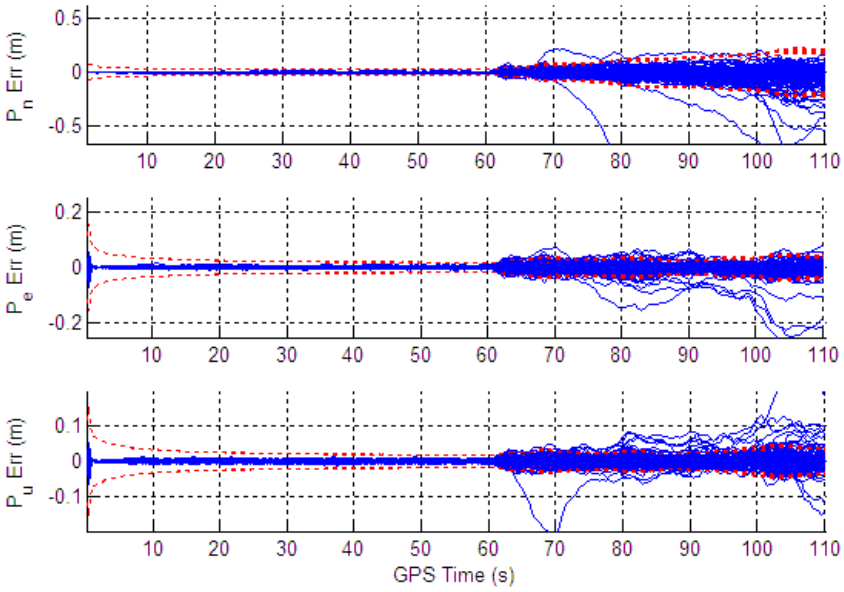

Figure 4.23. The ensemble statistics of error in position are shown, along with uncertainty bounds. The estimated one sigma uncertainty in the Northern trajectory is greater than the other two, but is still less than half a meter.

Notice that, after alignment, the filter estimates appear to be within the one-sigma bound more than 63% of the time with several outliers. This indicates that the filter may need to be tuned by adjusting the measurement and model uncertainties.

The ensemble data shows larger errors and uncertainty in the Northern position estimates than in the other two axes. This is attributed to lower observability in the direction of travel; recall that depth is only measured once during feature initialization. Also, the yaw axis has greater errors and uncertainty than either of the other two attitude axes; however, it is still accurate to within less than a degree on average.

A root-sum-square (RSS) calculation was performed to calculate the overall horizontal, vertical, and attitude errors. The results are shown in Figures 4.25 4.26 and 4.27.

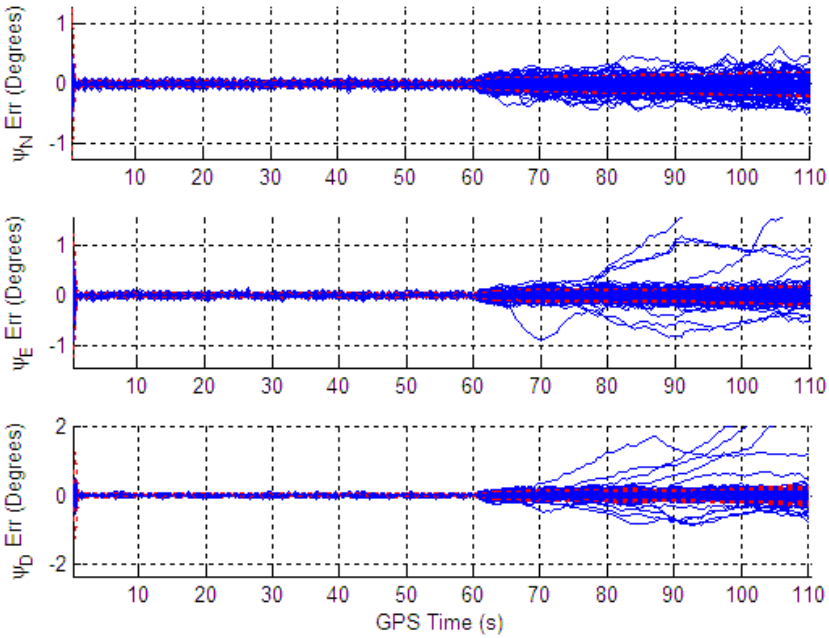

Figure 4.24. The ensemble statistics of error in attitude are shown. The filter estimates are constrained within a degree on average over a 110 second period.

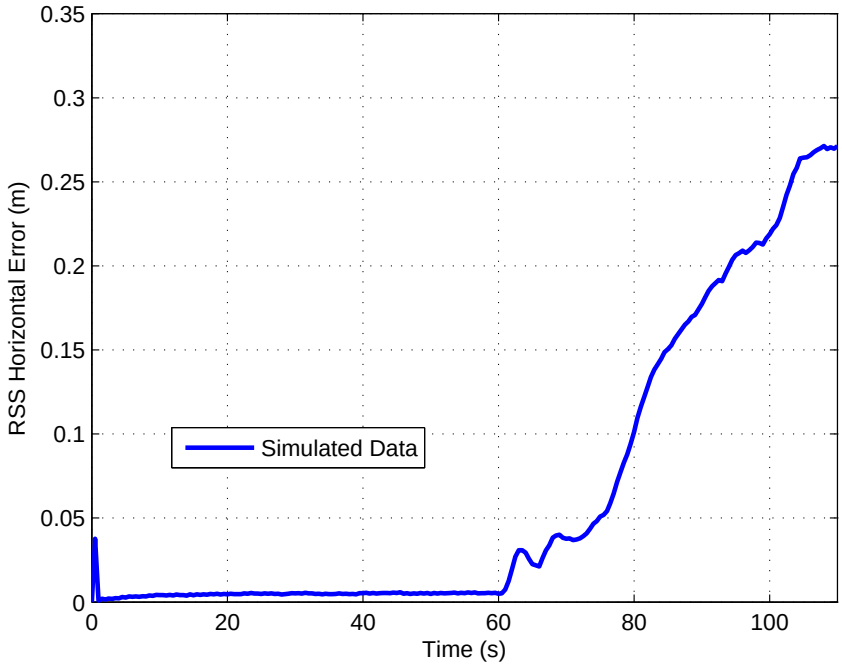

Figure 4.25. Simulated RSS horizontal error. The horizontal error is constrained to less than 30 centimeters. Note that the errors grow after 60 seconds, which is when the position and attitude alignments stop.

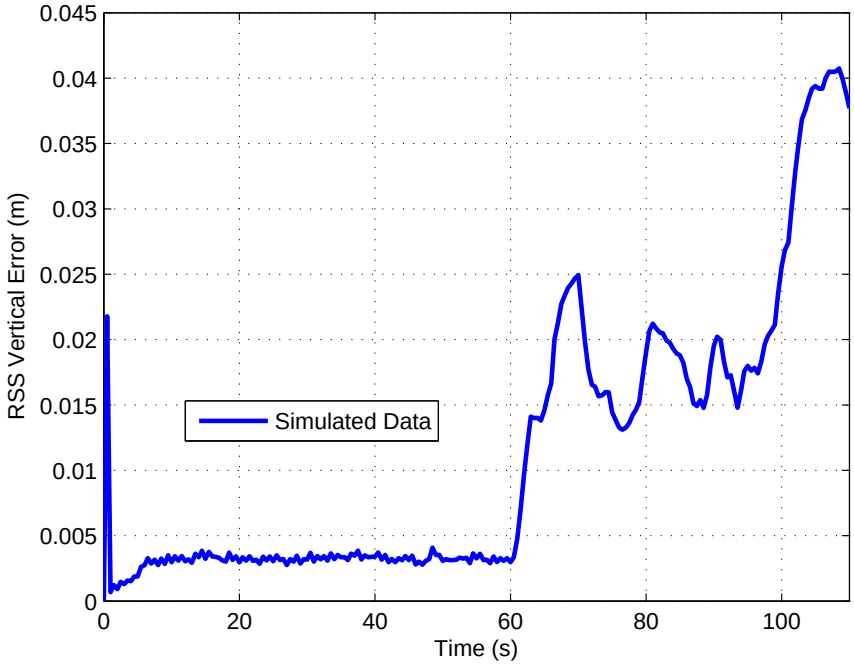

Figure 4.26. Simulated RSS vertical error. The vertical error is constrained to less than 5 centimeters.

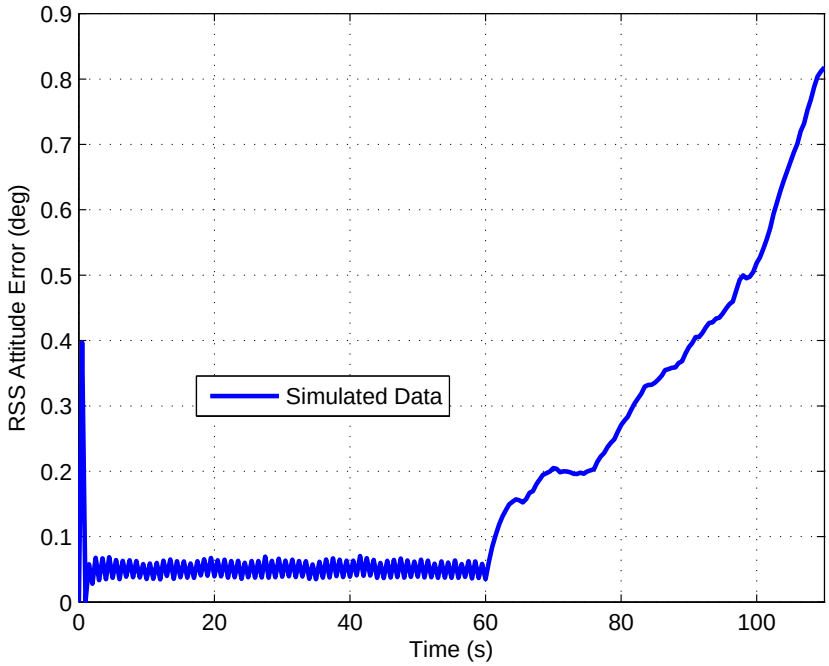

Figure 4.27. Simulated RSS attitude error. The attitude error is constrained to less than a degree.

Based on the ensemble statsics, the RSS horizontal error in position is constrained to within 30 centimeters. The vertical error is constrained to within 5 centimeters. The attitude error is less than a degree.

The simulated results show that an image-aided filter using a GLRS can accurately estimate a vehicle's trajectory and attitude. To test the validity of the simulation results, real data must be gathered and analyzed. Real world data is presented in the following sections.

4.3 Feature Location Test

In order to gain confidence in the GLRS's measurements, an experiment was conducted to test the accuracy of the calculated feature location. The experimental setup shown in Figure 4.21 was placed at a known location, then position and attitude alignments were performed. During the alignments, the navigation filter estimated four feature locations. Figure 4.28 shows the features that were used as well as a map with their estimated positions.

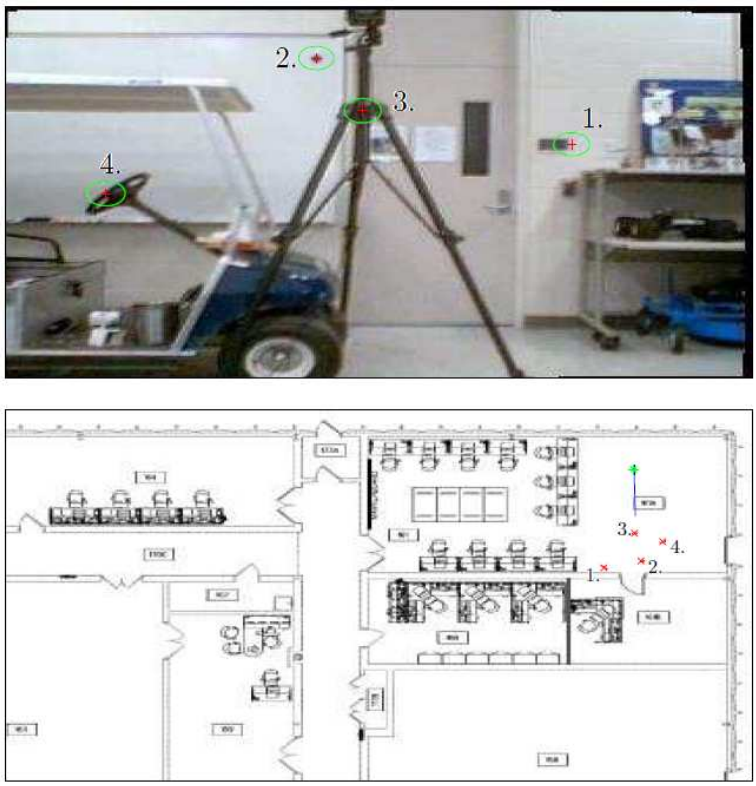

Figure 4.28. The features used are shown with a red cross. On the bottom is a map of the area used and the estimated position of each feature.

A tape measure was then used to measure the location of the features relative to the IMU, which was positioned at the center of the navigation frame. Table 4.4 shows the error between the estimated feature locations and the measured locations in the order in which features were added to the filter.

Order added	x_n error	y_n error	z_n error
1.	-0.03 meters	0.08 meters	-0.05 meters
2.	-0.12 meters	0.07 meters	0.05 meters
3.	-0.04 meters	0.04 meters	-0.9 meters
4.	-0.06 meters	-0.02 meters	0.03 meters

Table 4.4. The error between the estimated feature locations and the measured locations are shown. Errors are in the navigation frame.

If the position or attitude alignments were not performed, errors in the filters estimates would affect the feature location estimates. This emphasizes the need for both a position and attitude alignment while the first few features are initialized. As predicted in Section 3.5, Table 4.4 shows the errors in the feature locations are less than half a meter. In fact the errors are generally within a few centimeters. It is important to state that many more data samples are needed before statistical categorizing the accuracy of the system.

While the above results are not statistically meaningful, they do, however, give confidence in the GLRS's ability to calculate accurately a feature's location. Then next step will be to test the navigation filter's ability to produce accurate position and attitude estimates.

4.4 Vicon Experiment

To be able to determine the accuracy of the navigation filter's output, both in position and attitude, truth data must be available. This truth data was provided by the Vicon motion capture system. This system provides accurate, high-rate, three-dimensional position and attitude estimates. The estimates are produced using measurements from eight cameras and reflective markers that are placed on the object to be tracked. The camera measurements are then fed into the Vicon's IQ software [32] to produce a final trajectory.

In this experiment the test platform was allowed to run a position and attitude alignment for sixty seconds. While the alignment was performed, features were located

and added to the filter. A total of nine features were added before the platform was moved. After initializing the features, the platform was yawed in both directions then returned to center, after which, the platform was moved along the negative x_n-axis. Figure 4.29 shows a map with the platform's estimated trajectory and feature locations.

During alignment it was noted that some features that were initialized were dropped as soon as the laser beam was moved to another feature. This happened predominately when the laser beam was pointed directly at a lightly colored feature. A total of twelve features were initialized, yet three features had this problem and were dropped by the filter. The remainder of the experiment used only nine features. Figures 4.30 and 4.31 show the true and estimated trajectory and attitude during the experiment.

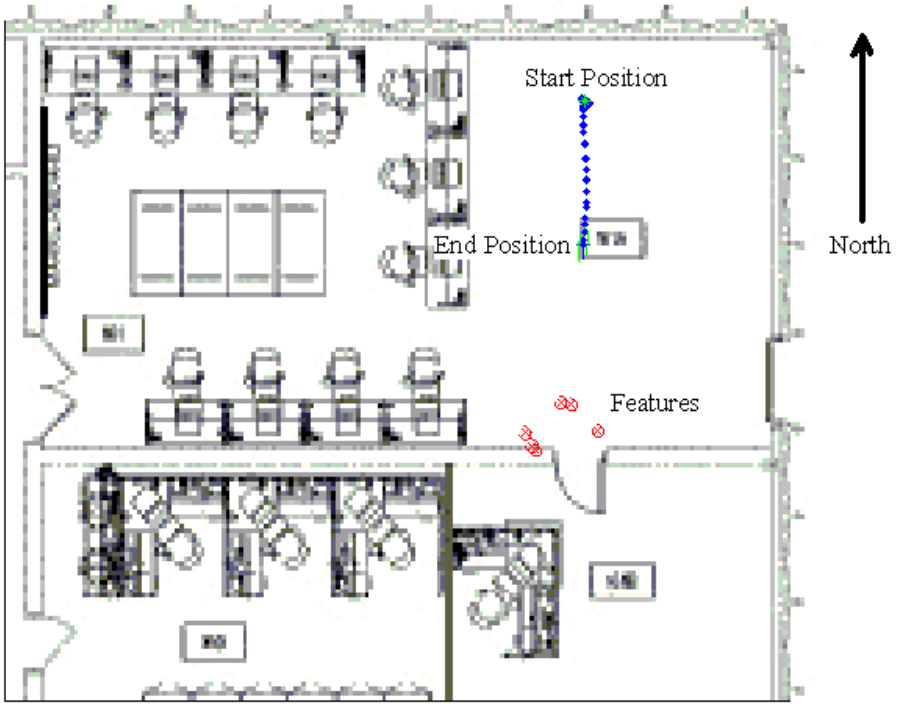

Figure 4.29. A map of the Vicon room and the estimated platform trajectory. Features are shown as red crosses with circles.

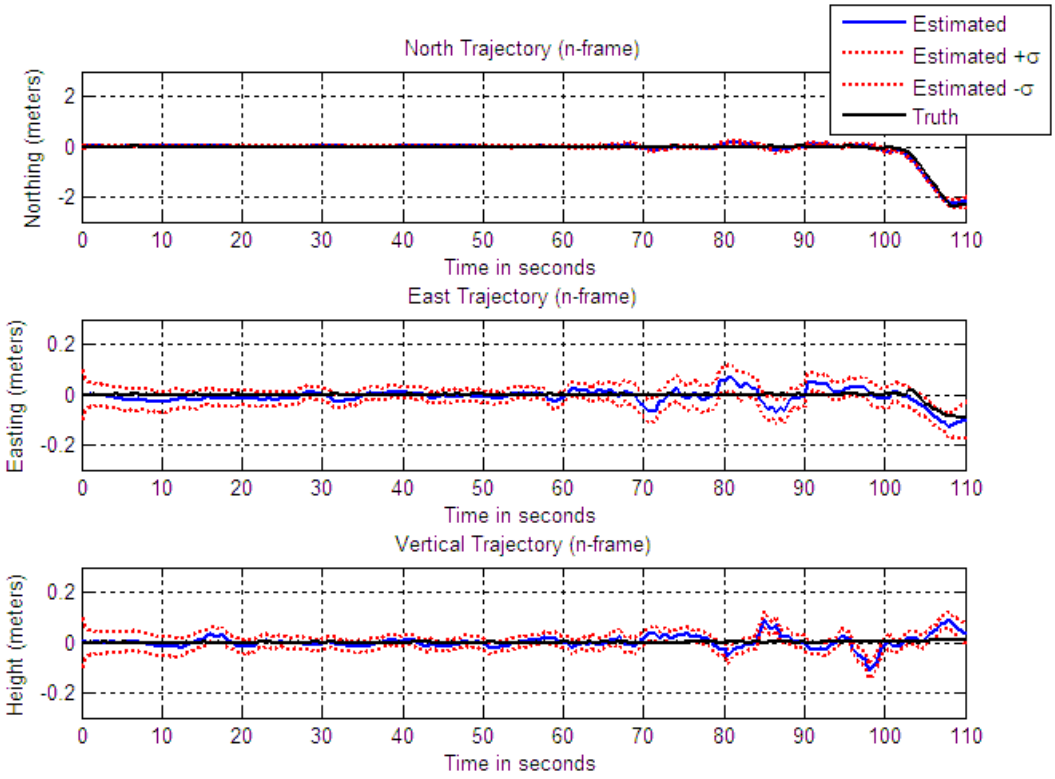

Figure 4.30. Position trajectory. The estimated trajectory is within a tenth of a meter to the true trajectory. The true trajectory is almost always within the one sigma standard deviation.

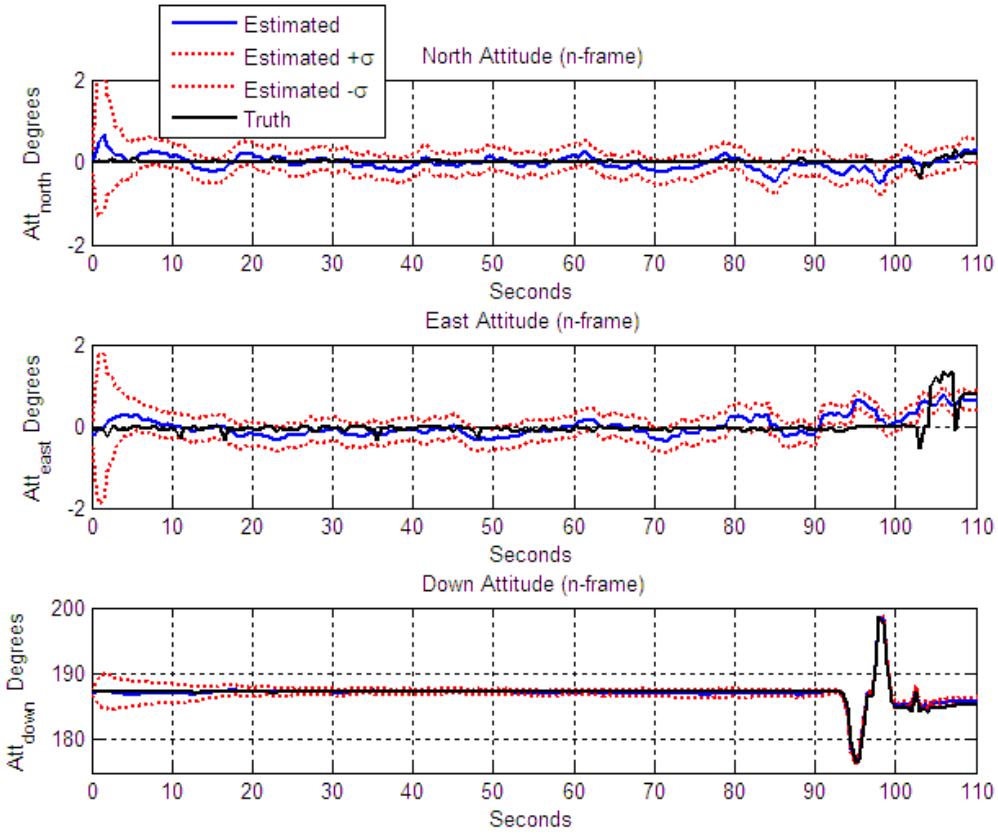

Figure 4.31. Attitude trajectory. Only the yaw axis was excited during this experiment, however there is a jump in pitch near the end of the run.

The filter's estimated trajectory closely follows the true trajectory. Figure 4.31 shows a jump in pitch near the end of the run, however the platform was kept level. This jump maybe explained by the Vicon camera's losing sight of a sufficient number of markers. Despite this, the estimated trajectory was within two-tenths of a meter to the true trajectory, and the attitude estimates were roughly within a degree, although only the yaw axis was excited during this experiment.

Figures 4.32 and 4.33 show the errors committed by the filter. The errors, as expected, are within the one sigma bound at least sixty-three percent of the time. This is an indication that the filter is properly tuned. Tuning was accomplished by adjusting the measurement model's uncertainties.

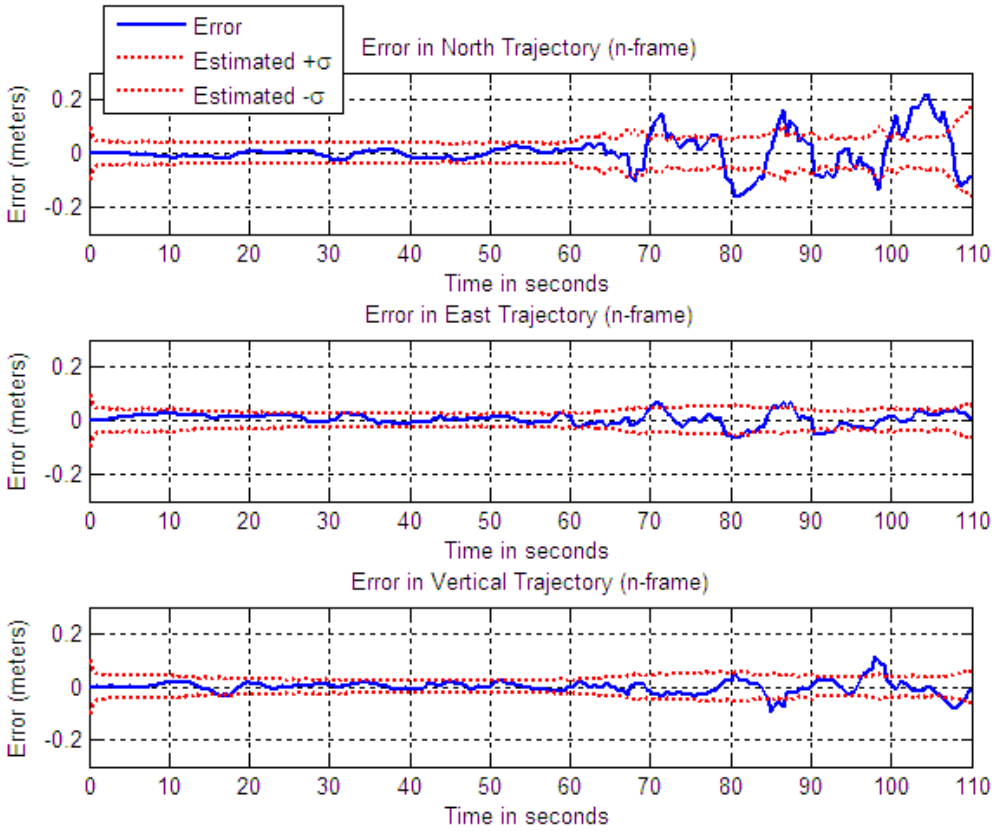

Figure 4.32. Position error. The one-sigma uncertainty bounds the errors sixty-three percent of the time. The filter's position uncertainty is less than a tenth of a meter.

Notice that the Northern trajectory has greater error and uncertainty than either the East or vertical axis. This agrees with the simulation and is again attributed to low observablity. Most of the features used by the filter in this experiment were near the center of the image plane. Picking features near the edges of the image plane and near the center would help with observability.

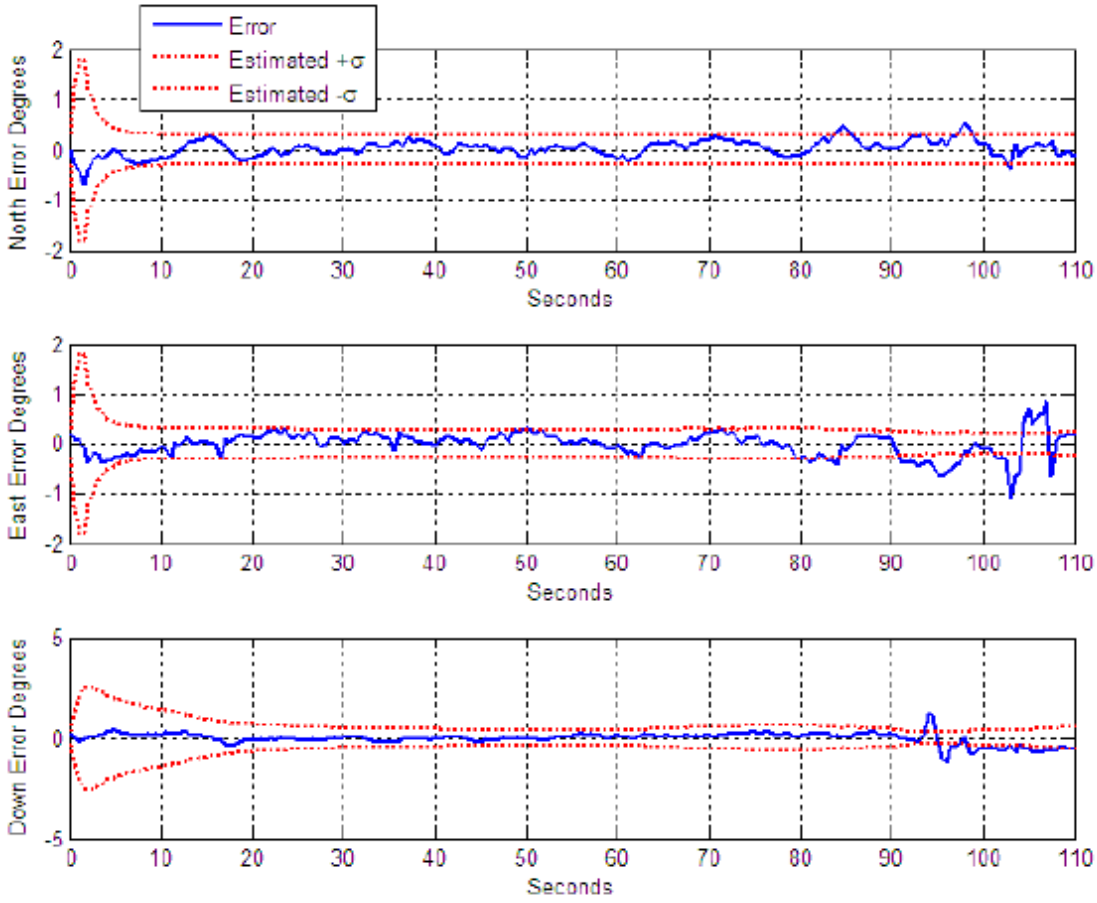

Figure 4.33. Attitude error. The error in attitude is generally constrained to less than half a degree. The errors in the z_n-axis happen when the platform yaws.

The error in attitude is generally constrained to less than half a degree. The errors in the z_n-axis happen when the platform yaws. The errors may be due to a misalignment between the image capture time and the IMU time. If the other two axis were excited, this misalignment error may have been observed. Figure 4.33 also shows a bias in the yaw estimate after it is excited. Notice that there is also a large error in pitch near the end of the data. This is attributed to the cameras losing sight of marks.

An RSS analysis was performed to gain a better understanding of the overall errors committed by the filter and to compare results to similar stereo aided filters. The horizontal RSS errors are shown in Figure 4.34. The errors are constrained to less than 0.25 meters, which is similar to the RSS horizontal errors of the stereo and monocular filters described in [25]. This result is also consistent with the results of the simulation. Figure 4.35 shows the RSS vertical errors.

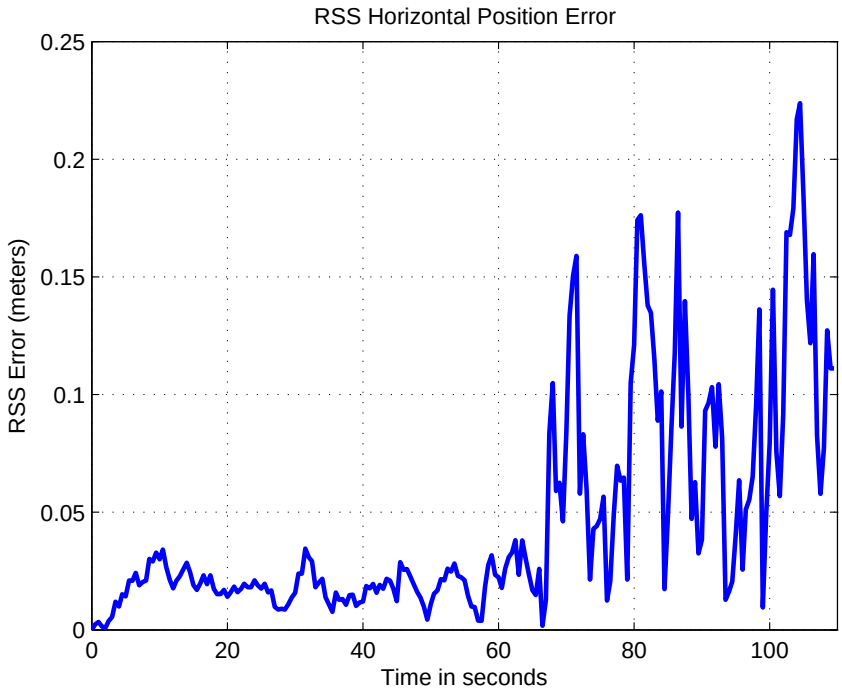

Figure 4.34. Root-sum-squared horizontal errors. The errors are constrained to less than 0.25 meters. Note that the errors grow after the alignments stop.

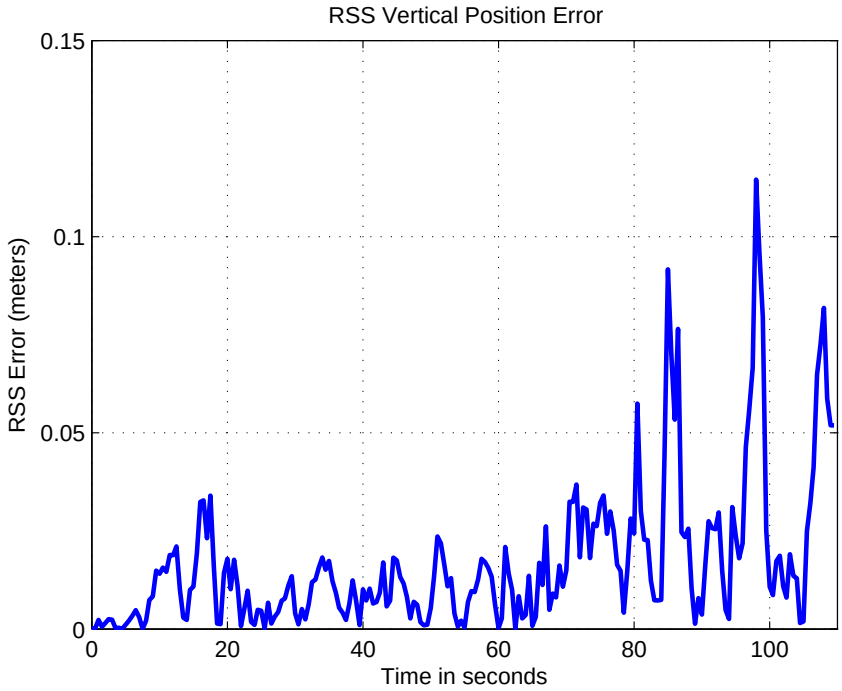

Figure 4.35. Root-sum-squared vertical errors. The errors are constrained to less than 0.15 meters.

The vertical error is smaller than the horizontal, which is consistent with the simulation, and is constrained to less than 0.15 meters. Again, this result is consistent with RSS vertical errors from similar stereo and monocular filters over the same flight time.

Figure 4.36 shows the RSS errors in attitude. The spike at the end of the plot is from when the platform was rotated. Even with this spike the errors are constrained to less than 1.5 degrees. Again, this is consistent with other vision-aided filters. One more experiment was conducted to better compare this research to previous vision-aided research at AFIT.

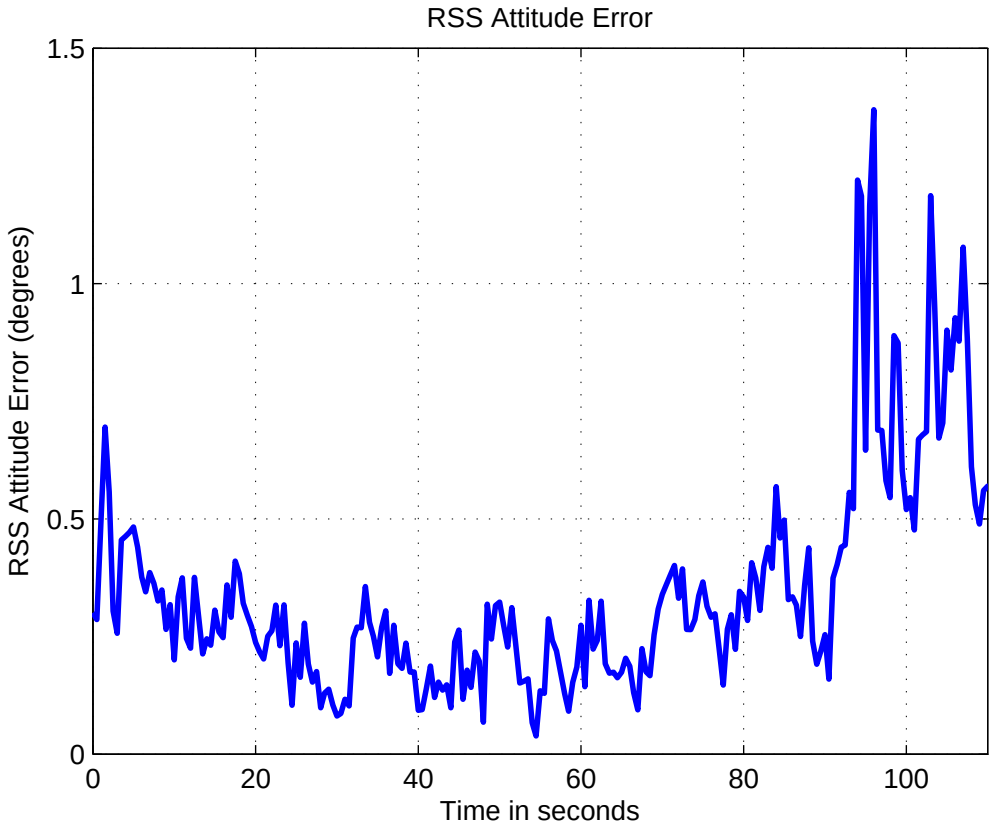

Figure 4.36. Root-sum-squared attitude errors. The yaw axis is the greatest contributor of error; it was also the only axis that was excited.

4.5 Hallway Experiment

A hallway experiment was conducted in order to further compare this research to other vision aided filters studied at AFIT. In this experiment the platform was positioned at a known location at the end of a hallway. Figure 4.37 shows a picture of the hallway used.

Figure 4.37. The hallway used for testing. This is the same hallway that was used in previous research [2] [25].

This hallway has points that were surveyed and marked. These surveyed points were used as starting and ending points. The platform was pushed as straight as possible down the hallway periodically stopping to add more features. When stopped, no alignments were performed. Only an initial sixty-second alignment was performed at the starting location. Figure 4.38 shows the estimated trajectory of the platform overlaid on a map of the hallway.

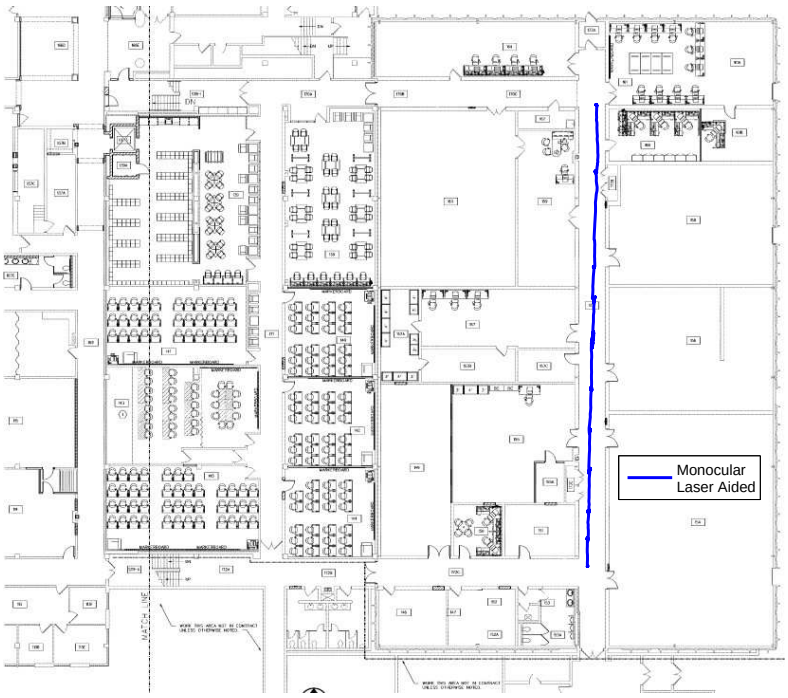

Figure 4.38. Platform Hallway Trajectory Estimate

There is no truth data for the experiment except for the starting and ending positions, and that care was taken to keep the platform level. At the end of the eleven-minute data collection, the horizontal error was less than half a meter. Possible sources of error could be from loss of features during movement. In fact, there were times during the flight when only two features were tracked.

Figure 4.39 shows the vertical estimate for the flight. Note that the altitude should have stayed near zero for the flight, but it is off by almost two meters. Also, there seems to be a stairway effect. The jumps occur at the same time the platform is moved. Because the large majority of features were tracked along the ceiling, there is low observability in the vertical direction. When the platform is moved forward the pixels move up in the image plane, and the filter cannot distinguish between forward movement and movement in the positive z_n-axis. This results in a bias in the positive z_n-axis.

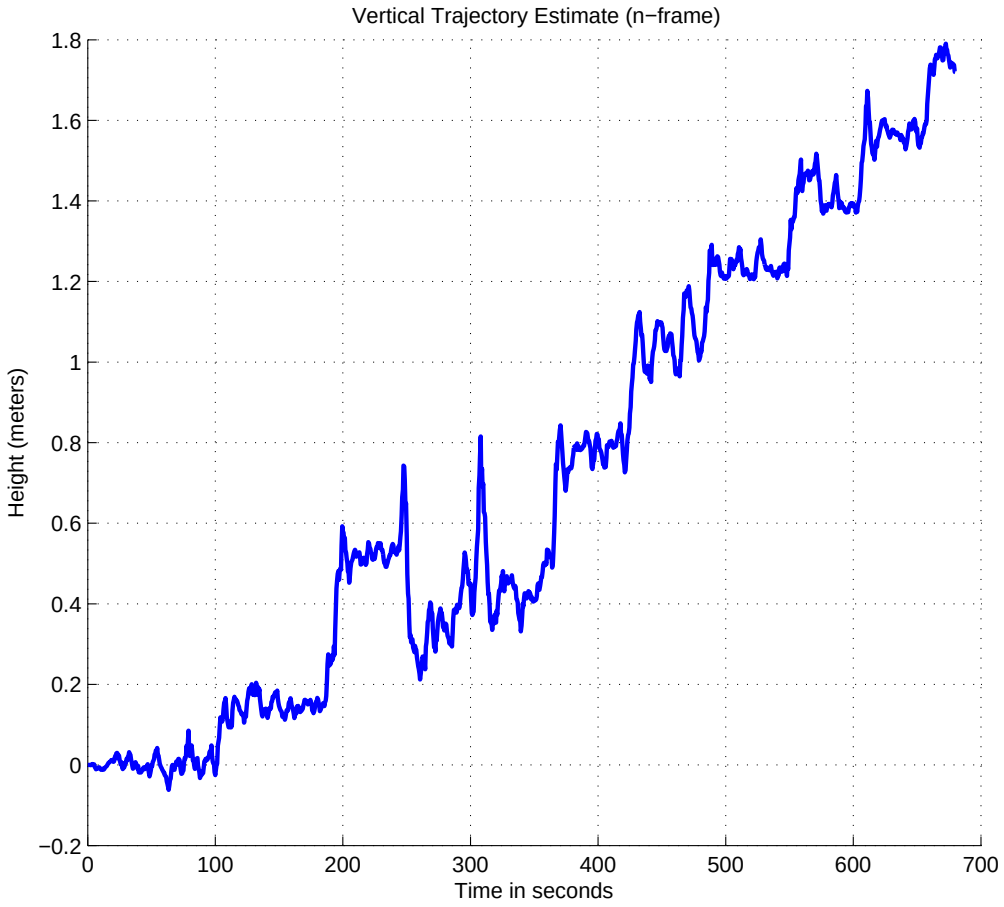

Figure 4.39. Platform hallway vertical trajectory. There is a stairway effect caused by periodic stops to add features to the navigation filter.

This bias can also be seen in the filter estimates of attitude. If the filter thinks it's moving in the negative z-axis, there should be a decrease in the estimate of pitch. Figure 4.40 shows this decrease. By the end of the run the filter's estimate of pitch should be close to zero, since the cart was kept level, however it's off by 2 degrees. Also, because the cart was kept level, roll should be close to zero, however, there are errors in roll as well. These errors could be due loss of features during movement, this could also be true for the pitch axis.

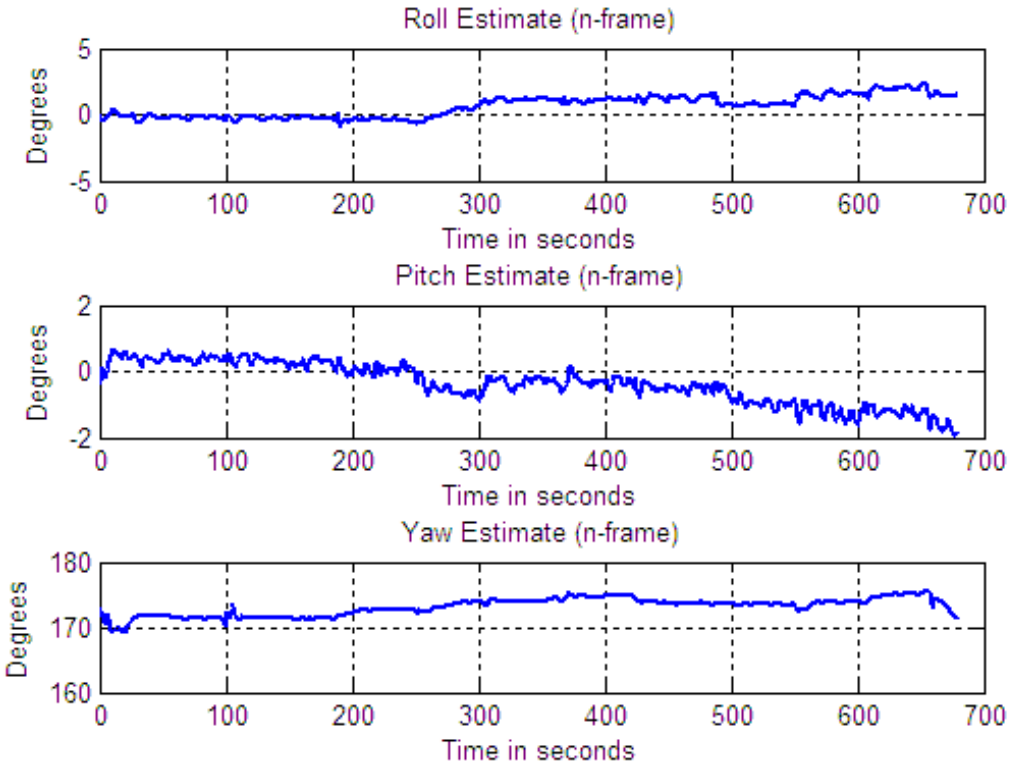

Figure 4.40. Filter's estimate of roll, pitch, and yaw. Note that pitch is consistantly decreasing.

To verify this theroy, a simulation was run in which only features near the top of the image plane were used. Figure 4.41 shows a 120 second simulated data run which only used features near the top of the images. A bias again, appears in the positive z_n-axis, which agrees with real world results presented above. There is also a clear bias in the East position estimate, which is due to an error in the yaw estimate. Figure 4.42 shows a small bias in the pitch axis as well, and a much larger bias in the yaw axis. While this is only one data set, and futher experiments are need, the results agree with real data results.

Comparing these results with previous research shows similar performance. In fact, the same bias in the positive z_n-axis is seen in [25] in both monocular and binocular filters during data collections in the same hallway.

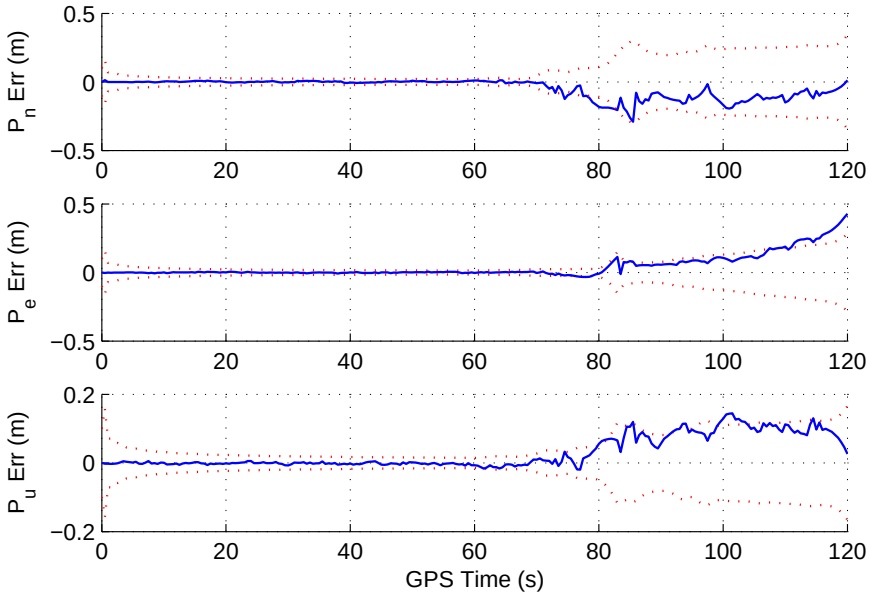

Figure 4.41. Position error of simulated data with near ceiling features. Notice that a bias appears in the positive z_n-axis and the East or y_n-axis.

Overall, the results in this chapter show that the monocular laser-aided filter successfully estimates the true navigation states, and agrees with simulated data. Furthermore, the monocular laser-aided filter appears to perform at the same level as other stereo and monocular aided filters. In the next chapter conclusions and future work are presented.

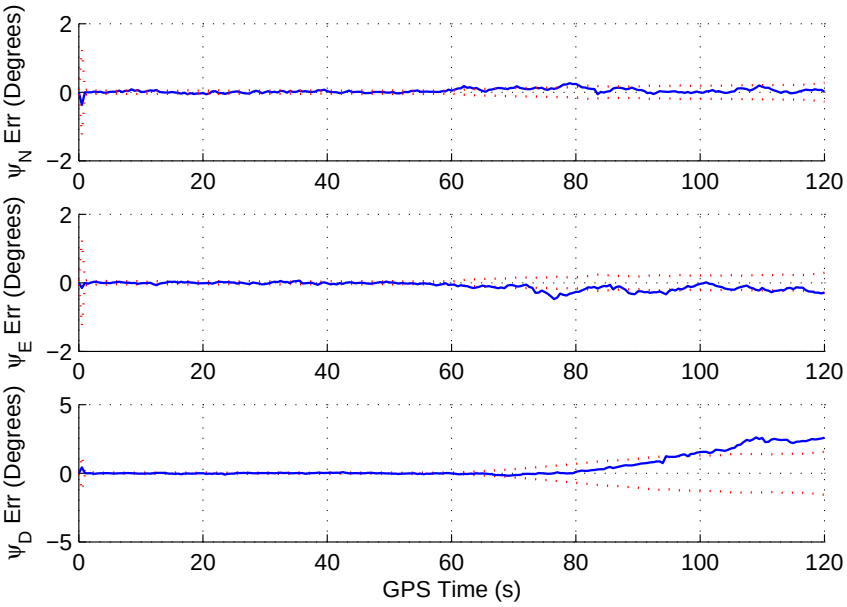

Figure 4.42. Attitude error of simulated data with near ceiling features. There is a bias in the pitch and yaw axis.

V. Conclusions

The goal of this research was to develop a straightforward way of estimating depth to features. This information is then used to estimate locations of features that are tracked in order to aid an IMU.

The straightforward method presented here was to use a gimbaled laser range sensor to estimate feature locations. Tests were performed to measure the accuracy of the GLRS's estimates. Using a tape measure, the estimated feature locations were shown to be accurate to within a few centimeters.

Simulations were then used to estimate the theoretical accuracy of an image-aided filter that had feature depth measurements provided by the GLRS. The simulations showed that the filter should be able to estimate trajectories to within three-tenths of a meter and attitude estimates to within a degree.

Following this, real world data runs were conducted to estimate the accuracy of the image-aided filter using the GLRS's estimates. The data runs showed that the filter could produce accurate trajectories, on the same level as previous image-aided filters. It was noticed that at times the laser beam interfered with the tracking of SIFT features. This happened more often with features that were close to the test platform and were surrounded by lightly colored objects, such as a white wall.

5.1 Future Work

The main obstacle faced during this research was interfacing a laser range sensor with a computer. Without this ability, real time experiments were not practical. In hindsight, a better choice for a laser sensor may have been the Aerius Photonics MLR100 miniature Laser Rangefinder [33]. The MLR100 has a range of 80 meters and is accurate to 19 centimeters. The MLR100's main advantage is that it has a USB interface. With the ability to interface directly a laser range sensor with the navigation filter, features could be added to the filter as needed. Further work needs to be done to test the monocular laser-aided filter's performance under a wider variety of flight profiles.

More work needs to be done in feature selection. When features are not well spread over the image plane, observability issues arise. Poor observability can cause the filter to commit estimation errors. Further work should be conducted to test alogithums that maximize observability and feature tracking time.

Also, statistical tests should be conducted in order to quantify the accuracy of the GLRS's pointing vector. If it is found that the GLRS can accurately point at features to within a small distance, candidate features could be selected that have a minimum scale, only large features are selected. This would help insure the range measurement is from the candidate feature and not an object in front of or behind the feature. A Monte-Carlo analysis of repeated data collections would provide these statistics.

It is possible that other techniques can be developed to help aid the navigation filter by using the GLRS. For example, if the filter has a sufficient number of features, or there are no good features to be found, the GLRS could still be used to measure distances to objects. The GLRS could be pointed directly in front, above, below, or to the sides of the vehicle to measure the distance between the platform and walls, ceiling, and the floor.

5.2 Summary

This research developed, built, and tested a GLRS for an image-aided filter. The performance of the navigation filter showed that position could be estimated to within tenths of meters; however, there are observability issues that arise when features are not well spread over the image plane. This could lead to problems in scenes where there are few distinct features. Overall, the result showed that accurate trajectories could be produced without using terrain constraints or multiple cameras. These findings will help designers and engineers build smaller and more accurate UAVs for commercial and military use.

Bibliography

1. D. Titterton and J. Weston., *Strapdown Inertial Naviagation Technology.* Institution of Electrical Engineers, Stevenage, UK, 2nd Edition, 2004.
2. M. J. Veth, *Fusion of Imaging and Inertial Sensors for Navigation.* Ph.D. Thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, 2006.
3. D. B. Barber, *Accurate Target Geolocation and Vision-Based Landing with Application to Search and Engage Missions for Miniature Air Vehicles.* MS. Thesis, Department of Mechanical Engineering, Brigham Young University, 2007.
4. E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision.* New Jersey, Prentice-Hall, 2002.
5. Z. Zhang, *A flexible new technique for camera calibration, Pattern Analysis and Machine Intelligence.* IEEE Transactions on, vol. 22, no. 11, pp. 13301334, 2000.
6. J. Bouguet, *Camera Calibration Toolbox for Matlab.* http://www.vision.caltech.edu/bouguetj/calib doc/index.html, 2004.
7. M. Pachter and A. Porter, *INS Aiding by Tracking An Unknown Ground Object- Theory.* Proccedings of the American Control Conference, vol 2 1260-1265, 2003.
8. B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation.* Hoboken, New Jersey: John Wiley and Sons, Inc., 2nd Edition., 2003.
9. P. S. Maybeck, *Stochastic Models, Estimation, and Control Volume 1.* Academic Press, Inc, 1979.
10. P. S. Maybeck, *Stochastic Models, Estimation, and Control Volume 2.* Academic Press, Inc, 1982.
11. C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of The Fourth Alvey Vision Conference,* Manchester, vol. 15, pp. 147–151, 1988.
12. I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," *In: Pattern Classification and Scene Analysis,* Duda, R. and Hart, P., John Wiley and Sons. , pp. 271–272, 1973.
13. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library.* Sebastopol, CA: O'Reilly Media, Inc., 2008.
14. D. G. Lowe, *Distinctive Image Features from Scale-Invariant Ketpoints.* International Journal of Computer Vision, 60(2):91-110, 2004.
15. P. C. Mahalanobis, "On the generalised distance in statistics," *Proceedings National Institue of Science. India*, 1936.
16. M. J. Smith, *Electronic Image Stabilization of Mobile Robotic Vision Systems.* MS Thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, 2009.

17. D. W. Strelow, *Motion Estimation from Image and Inertial Measurements*. Ph.D. Thesis, School of Computer Science, Carneige Mellon University, Pittsburgh, PA 15213, November, 2004.

18. Y. S. S. Ma, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision*. Springer-Verlag, Inc., New York, New York, 2004.

19. O. A. Aider, P. Hoppenot, and E. Colle, "A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences," *Robotics and Autonomous Systems*, vol. 52, pp. 229–246, 2005.

20. K. Celik, S.-J. Chung, and A. Somani, "Mono-vision corner slam for indoor navigation," *IEEE*, vol. 978, pp. 343–348, August, 2008.

21. A. Harati and R. Siegwart, "Orthogonal 3d-slam for indoor environments using right angle corners," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijing, China*, 2006.

22. K. O. Arras, N. Tomatis, and R. Siegwart, "Multisensor on-the-fly localization using laser and vision," *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

23. L.-P. Ellekilde, S. H. Jaime, V. Miro, and G. Dissanayake, "Dense 3d map construction for indoor search and rescue," *Journal of Field Robotics*, vol. 24, pp. 71–89, 2007.

24. J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target localization from a fixed-wing miniature air vehicle," *Proceedings of the 2006 American Control Conference, Minneapolis*, June, 2006.

25. J. R. Gray, *Deeply-Integrated Feature Tracking for Embedded Navigation*. MS Thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, 2009.

26. "Fluke 416d laser distance meter." Fluke Corporation, URL: http://us.fluke.com/, July. 2008.

27. "DPC AV, pandora pan and tilt kit." URL: http://www.dpcav.com/, June, 2008.

28. *High Precision Tri-Axis Inertial Sensor ADIS16350/ADIS16355*. Analog Devices, Inc. URL: http://www.analog.com/, Sept. 2009.

29. Stingu, Emanuel, and F. Lewis, "Quad-rotor specificaitons." URL: http://www.arri.uta.edu/acs/pestingu/UAV, 2008.

30. J. Shi and C. Tomasi, "Good features to track," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

31. S. Musick, "Profgen: Pc software for trajectory generation.." Software Package v8.19, Air Force Research Laboratory, Wright-Patterson AFB OH, January, 2004.

32. "Vicon MX." URL: http://www.vicon.com/products/viconmx.html, January, 2009.

33. "Aerius photonics MLR100 specification sheet." Aerius Photonics LLC., March, 2009.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
25–03–2010	Master's Thesis	Sept 2008 — Mar 2010

4. TITLE AND SUBTITLE

MONOCULAR VISION LOCALIZATION USING A GIMBALED LASER RANGE SENSOR

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Don Jared Yates, 2Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/10-31

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

INTENTIONALLY LEFT BLANK

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

There have been great advances in recent years in the area of indoor navigation. Many of these new navigation systems rely on digital images to aid an inertial navigation estimates. The Air Force Institute of Technology (AFIT) has been conducting research in this area for a number of years. The image-aiding techniques are centered around tracking stationary features in order to improve inertial navigation estimates. Previous research has used stereo vision systems or terrain constraints with monocular systems to estimate feature locations. While these methods have shown good results, they do have drawbacks. First, as unmanned exploration vehicles become smaller in size the distance available to create a baseline between two cameras decreases resulting in a decrease of distancing accuracy. Second, if using a monocular system, terrain data might not be known in an unexplored environment. This research explores the use of a small gimbaled laser range sensor and monocular camera to estimate feature locations.

15. SUBJECT TERMS

Feature Tracking, Extended Kalman Filter, Vision-aiding, Indoor Navigation, Laser-aiding, Gimbaled Laser Range Sensor

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			LtCol Michael J. Veth
U	U	U	U	92	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4541; michael.veth@afit.edu

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18