Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations | Student Graduate Works

3-26-2015

Urban Environment Navigation with Real-Time Data Utilizing Computer Vision, Inertial, and GPS Sensors

Johnathan L. Rohde

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Electrical and Electronics Commons

Recommended Citation

Rohde, Johnathan L., "Urban Environment Navigation with Real-Time Data Utilizing Computer Vision, Inertial, and GPS Sensors" (2015). *Theses and Dissertations*. 56.
https://scholar.afit.edu/etd/56

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.

URBAN ENVIRONMENT NAVIGATION WITH REAL TIME DATA UTILIZING VISION, INERTIAL, AND GPS

THESIS

Johnathan L. Rohde, 1st Lieutenant, USAF

AFIT-ENG-MS-15-M-037

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-MS-15-M-037

REAL-TIME IMPLEMENTATION OF VISION, INERTIAL, AND GPS SENSORS TO NAVIGATE IN AN URBAN ENVIRONMENT

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

Johnathan L. Rohde, B.S.E.E.

1st Lieutenant, USAF

March 2015

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-MS-15-M-037

REAL-TIME IMPLEMENTATION OF VISION, INERTIAL, AND GPS SENSORS TO NAVIGATE IN AN URBAN ENVIRONMENT

THESIS

Johnathan L. Rohde, B.S.E.E.
1st Lieutenant, USAF

Committee Membership:

John F. Raquet, PhD
Chair

Kyle J. Kauffman, PhD
Member

David R. Jacques, PhD
Member

Abstract

The purpose of this research was to obtain a navigation solution that used real data, in a degraded or denied global positioning system (GPS) environment, from low cost commercial off the shelf sensors. The sensors that were integrated together were a commercial inertial measurement unit (IMU), monocular camera computer vision algorithm, and GPS. Furthermore, the monocular camera computer vision algorithm had to be robust enough to handle any camera orientation that was presented to it.

This research develops a visual odometry 2-D zero velocity measurement that is derived by both the feature points that are extracted from a monocular camera and the rotation values given by an IMU. By presenting measurements as 2-D zero velocity measurements, errors associated with with scale, which is unobservable by a monocular camera, are properly handled. The 2-D zero velocity measurements are represented as two normalized velocity vectors that are orthogonal to the vehicle's direction of travel, and are used to determine the error in the INS's measured velocity vector. This error is produced by knowing which directions the vehicle is not moving in, given by the 2-D zero velocity measurements, and comparing it to the IMU-reported direction of travel.

The performance was evaluated by comparing results that were obtained when different sensor pairings of a commercial IMU, GPS, and monocular computer vision algorithm were used to obtain the vehicle's trajectory. Three separate monocular cameras, that each pointed in a different directions, were tested independently. Finally, the solutions provided by the GPS were degraded (i.e., the number of satellites available from the GPS were limited) to determine the effectiveness of adding a monocular computer vision algorithm to a system operating with a degraded GPS solution.

Table of Contents

Page

List of Figures

List of Tables

REAL-TIME IMPLEMENTATION OF VISION, INERTIAL, AND GPS SENSORS TO NAVIGATE IN AN URBAN ENVIRONMENT

I. Introduction

This thesis outlines the research efforts that were used to pair low cost, commercial-off-the-shelf sensors to provide a navigation solution for a ground vehicle in a GPS-degraded environment. Sensors that were used in this research were a commercial grade inertial measurement unit (IMU), global positioning system (GPS), and a monocular camera computer vision algorithm. Motivation behind this thesis was to provide a navigation platform that could operate off of low cost sensors when a degradation of the GPS was encountered. While the implementation of a paired monocular camera computer vision algorithm with an IMU and GPS is not a new concept, the development of the monocular camera computer vision algorithm being used as a 2-D zero velocity update with real data in this research is. Furthermore, the implementation of the monocular camera computer vision algorithm in a GPS degraded environment shows merit of the sensor as a viable aid when operating in an urban terrain.

Since the launch of the first satellite in 1978 [21], GPS has provided more accurate worldwide navigation capability than any other navigation tool that exists. The use of GPS as a precision navigation and targeting tool was demonstrated in the first Gulf War, where the importance and utility of GPS was proven [12]. Because of the great success using GPS, a strong reliance on GPS has been developed. However, GPS is susceptible to periods of unavailability, which can be caused by the terrain, environment, or interference from an outside source. In these situations, a viable

alternative to GPS navigation has to be implemented.

This research will use real data to validate the work found in [15] that utilized simulated data. To do so, a monocular camera computer vision algorithm (that measures 2-D normalized velocity vectors that are orthogonal to the direction of travel) will be integrated with an IMU, and a range of GPS availabilities, from no satellite vehicles available to a full GPS navigation solution. The results will show how the monocular camera computer vision algorithm presented in this work, capable of operating with a monocular camera pointing in any direction, can help mitigate errors from a GPS outage, and provide a better navigation solution than which would be available if the system operated on GPS and IMU solutions alone.

1.1 Research Objective

The main objective for this research was to implement a monocular camera computer vision algorithm that uses real data to provide velocity error feedback to a commercial grade IMU. The monocular camera computer vision algorithm presented in [15], which use features detected by a monocular camera and rotation given by the IMU, will be implemented in this thesis. In conjunction with the monocular camera computer vision algorithm and the IMU, a range of GPS solutions will be employed to test the effectiveness of aiding the system with a monocular camera. By pairing the three sensors, monocular camera computer vision algorithm, IMU, and GPS, the usability of a monocular camera as a supplement to GPS and how the overall performance of the navigation system can be increased will be shown.

1.2 Scope

Because all of the measurements used in this thesis came from the All Source Positioning Navigation (ASPN) test set, test scenarios were limited to the data that had been previously recorded. This meant that additional runs for further evaluation could not be conducted. It also meant the lack of some measurements in certain scenarios had to be accounted for and removed when providing the results of the system.

1.3 Thesis Overview

Chapter II of this thesis develops the mathematical background required to effectively complete this work along with the math notation. Included in this is the camera model, feature detection, epipolar geometry, coordinate frames and transformations, the Kalman Filter, and the extended Kalman filter (EKF). An overview of related research is also presented.

Chapter III presents the methodology behind each sensors measurements. The IMU simulation algorithm and the error state model are also presented here. Finally, all the sensors are tied together through the EKF, which is described in this chapter.

Chapter IV highlights the results that were obtained by the culmination of this research. Each scenario will be presented with an IMU only solution, GPS aided, computer vision aided, and both GPS and computer vision aided. Final results will be compared to each other to show overall system performance increase with the addition of each sensor.

Chapter V will provide a summary of everything discussed in previous chapters, along with the future work to be considered.

II. Background

The information that is outlined in this chapter is background material, which is required to understand this thesis.

2.1 Mathematical Notation

The mathematical notation that will be implemented for this thesis is laid out as follows.

- **Vectors:** Vector's will be formed in column elements and annotated with bold font lower case letters, (e.g. $\mathbf{y}$ or $\boldsymbol{\rho}$). A vectors specific scalar elements will be represented by $\mathbf{x}_i$, where i defines the column element number.
- **Matrix**: Matrices will be annotated with bold font upper case letters, (e.g. $\mathbf{X}$ or $\boldsymbol{\Psi}$). Matrix row and column elements will be annotatd as $\mathbf{X}_{ij}$, where i is the row index and j is the column index.
- **Scalar**: Scalars will be annotated with a non-bold lower or uppercase letters, (e.g. f or Z)
- **Reference Frames**: Specific reference frames will be annotated with an upper case letter followed by a lower case superscript, (e.g. F^n or G^b)
- **Direction Cosine Matrix**: DCMs will dictate the state in which the frame is being transformed from (subscript) to the frame that it is being transformed to (superscript) (e.g. C_s^b or C_b^n).

2.2 Camera Properties

In this section, information pertaining to the camera model, feature detection, and epipolar constraints between images will be conceptualized.

Thin Lens.

The imagery information that is given by the monocular camera is a feature's re-projection from a real world location through a lens onto a camera image. The scene's projection on to the camera's image (in this instance) is pictorially shown by the thin lens camera model (Figure 1) [7].

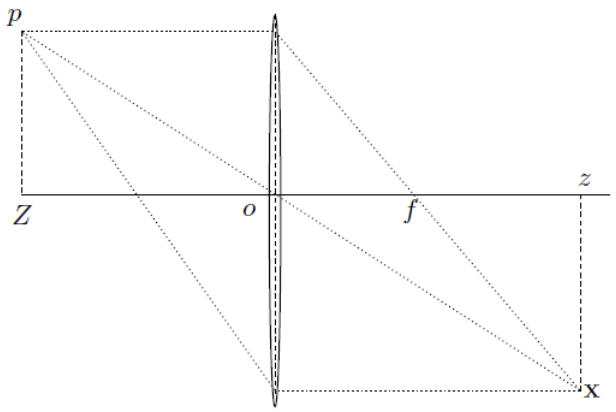

Figure 1. Thin Lens Camera Model [7]

The projection of the feature p onto the camera image, in the thin lens camera model (Figure 1), is determined by the feature's relative position to the camera and the optical lens of the camera. The function of the thin lens is characterized by two properties[7]. The first property is that all rays entering the aperture parallel to the optical axis intersect on the optical axis at a distance f from the optical center. The second property is that all rays through the center of the axis are undeflected. The values that are obtained from the thin lens camera model are derived by using the *fundamental equation of the thin lens* [7]:

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

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

to zero, then the rays are forced to enter the optical lens at the center of the lens. From the second property of the thin lens, it is established that those rays which pass through the center of the optical lens are undeflected. Similar geometric triangles that were seen in the thin lens model earlier can now be seen in the pin hole camera model (Figure 2).

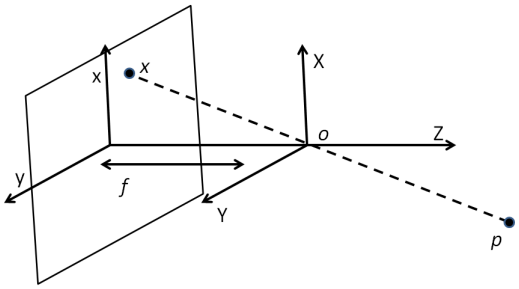

Figure 2. Pinhole Camera Model

From the pinhole camera model, the coordinates of the feature $\mathbf{p} = [X, Y, Z]^T$ can be related to its $\mathbf{x}$ image coordinates by the *prospective projection* [7]:

$$x = -f\frac{X}{Z}, \quad y = -f\frac{Y}{Z} \tag{2}$$

The location of $\mathbf{x}$ on the image plane is inverted from it original position, which creates a negative sign in the *prospective projection* equation. To solve the problem of the reverse sign, the frontal pinhole model will be used (Figure 3).

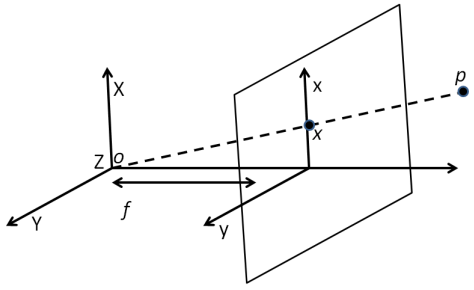

Figure 3. Frontal Pinhole Camera Model

From the frontal pinhole camera model, which will be used in this research, the coordinates of feature p can now be related to its x image coordinates by

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \tag{3}$$

From the image coordinates (x, y), the actual pixel coordinates of the feature can be determined. The translation from the origin to the pixel coordinates is shown in Figure 4.

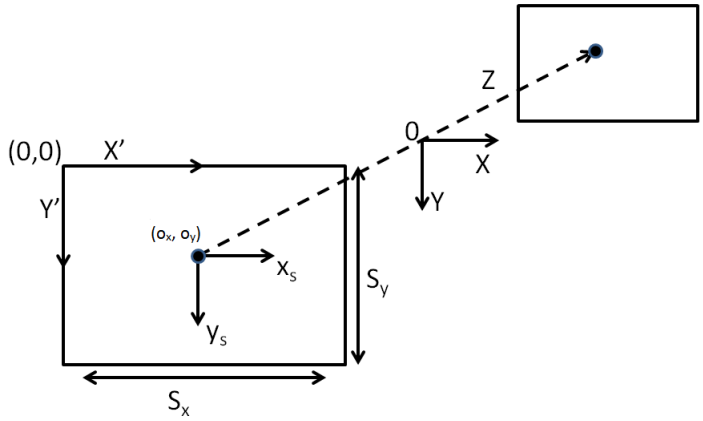

Figure 4. Transformation from normalized coordinates to coordinates in pixels.

The ratio of the pixels to the frames width and height are given by S_x and S_y. The origin of the pixel coordinates, for this thesis, will be defined as the upper left corner of the image plane. The translation from the original reference frame to the top left corner is given by:

$$\begin{aligned} x' &= x_s + o_x \\ y' &= y_s + o_y \end{aligned} \tag{4}$$

where o_x and o_y are the coordinates (in pixels) of the principle point to the image reference frame, and x_s and y_s are the number of pixels away from the origin in horizontal and vertical axes [7]. The homogeneous solution for the pixel coordinates

can be achieved by:

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

Feature Detection.

One of the key aspects of using a computer vision algorithm is the ability to perform feature detection of objects in an image. Feature detection that was invariant to scale and rotation was performed using the method detailed in [6]. This section will present the Scale Invariant Feature Transform (SIFT) algorithm that will be used in this research.

The SIFT algorithm defined in [6] works off of four key elements.

1. **Scale-space peak selection**: Features are selected based off of their invariance to scale by by using the difference-of-Gaussian function.

2. **Keypoint localization**: A fit of nearby data for location, edge response, and peak magnitude is performed.

3. **Orientation assignment**: Orientations are assigned to each keypoint location based on the local image properties. This enables the descriptors to be rotation invariant.

4. **Keypoint descriptor**: A weight is assigned to local image gradients at the selected scale in the region around the detected keypoints, and the set of weighted gradients are given as a descriptor associated with the keypoint.

From the keypoints that have been determined by the previous elements, the features are then matched to features in a corresponding image. These features are matched by a Best-Bin-First (BBF) algorithm. This defines a feature point that is matched

based off of its first closest neighbor distance to its second closest neighbor. The value assigned in this distance ratio is determined by the user of the algorithm, and can be set very high or low depending on the validity of the matches required. Setting a higher value can return features that are not matches, while setting a too low of a number can provide for an over determined system that rejects actual matches. A more defined explanation of the SIFT algorithm can be found in [6].

Epipolar Geometry.

Given a feature that appears in two separate images, from the same monocular camera, the rotation and translation of the camera can be determined. This rotation and translation can then be used to solve for rotation and translation of a vehicle, if the camera was rigidly mounted to it. Solving for these two factors can be done by using the epipolar constraints of the system [4]. If it is known that the feature itself has not moved, it can be implied that the only thing that has moved is the camera. The shared feature between two images can be seen in Figure 5.

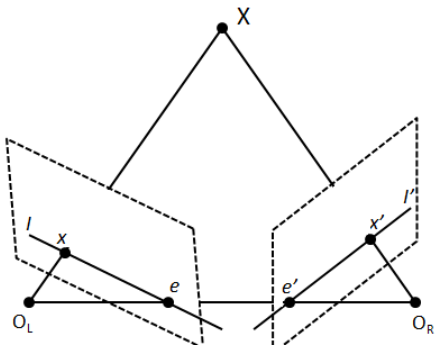

Figure 5. X is the feature in the real world, with no known location. x and x′ is the representative feature on each of the camera images. O_L and O_R are the origins of the two separate images in the navigation frame. e and e' are the epipoles of the images.

The epipolar constraints are in turn used to determine the essential matrix. The essential matrix, is a function of the rotation and translation, which satisfies the

constraints

$$\mathbf{x}'\mathbf{E}\mathbf{x} = 0 \tag{6}$$

where $\mathbf{E}$ is the essential matrix of the system. To determine the essential matrix from images, where the origin and the feature position are not known, the eight point algorithm can be used [7]. Using at least eight different features in two separate camera images, eight different linearly independent *Kronecker product* [7] vectors can be determined:

$$\begin{aligned} &\mathbf{a} \doteq \mathbf{x} \otimes \mathbf{x}' \\ &\mathbf{a} = [x_1x_2', x_1y_2', x_1z_2', y_1x_2', y_1y_2', y_1z_2', z_1x_2', z_1y_2', z_1z_2'] \\ &\boldsymbol{\chi} \doteq [\mathbf{a}^1, \mathbf{a}^2, ..., \mathbf{a}^n]^T \end{aligned} \tag{7}$$

where $\boldsymbol{\chi}$ is the vector of *Kronecker products* given by each matched features between the two images.

When no outside noise sources are affecting the system it can be shown that

$$\boldsymbol{\chi}\mathbf{E}^s = 0 \tag{8}$$

where $\mathbf{E^s}$ is a stacked vector that is equal to the eigenvector that corresponds to the smallest eigenvalue associated with $\boldsymbol{\chi}^T\boldsymbol{\chi}$ [7] and is given as

$$\mathbf{E}^s = [\mathbf{e}_{11}, \mathbf{e}_{12}, \mathbf{e}_{13}, \mathbf{e}_{21}, \mathbf{e}_{22}, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_{32}, \mathbf{e}_{33}]^T \tag{9}$$

The values found for $\mathbf{E}^s$ can then be unstacked into a 3x3 matrix that gives an

estimate of the essential matrix $\mathbf{E}_{est}$ as

$$\mathbf{E}_{est} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \tag{10}$$

By preforming a singular value decomposition on the essential matrix estimate, the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ can be obtained along with the diagonal matrix $\mathbf{S}$. For the essential matrix to be considered valid, its determinant must be zero and its two non-zero singular values must be equal [16]. To conform to the requirements of the valid essential matrix, the diagonal matrix is set manually to the following values

$$\mathbf{S}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{11}$$

the projected essential matrix can be determined [16]:

$$\mathbf{E} = \mathbf{U}\mathbf{S}'\mathbf{V}^T \tag{12}$$

2.3 Attitude Representation

This section depicts three different mathematical representations that are used to define the attitude of a body. The three representations are the direction cosines, Euler angles, and quaternions [20], and they will be discussed in further detail in this section.

Direction Cosine.

The direction cosine matrix (DCM) is a 3x3 matrix, where each row and column represents a unit vector, and the entire DCM is used to rotate one frame of reference into another frame of reference [20].

DCM Vector Transformation.

The DCM, in this thesis, will be denoted by the symbol convention of $\mathbf{C}_s^b$. Where the components of $\mathbf{C}_s^b$ are given as:

$$\mathbf{C}_s^b = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{13}$$

The elements in the respective row and column represents the cosine angle between the reference frame, in this case the sensor frame and the body frame. Given a vector quantity in the sensor frame $\mathbf{n}^s$, the vector can be transformed into the body frame by multiplying it by the DCM as shown by:

$$\mathbf{n}^b = \mathbf{C}_s^b \mathbf{n}^s \tag{14}$$

One of the properties of the DCM, that will be used in this research, is the transposing or inverting of the original DCM, allowing for transformation from the body frame to the sensor frame:

$$\mathbf{C}_b^s = (\mathbf{C}_s^b)^T \tag{15}$$

$$\mathbf{n}^s = \mathbf{C}_b^s \mathbf{n}^b \tag{16}$$

Euler Angles.

By performing a set of three successive rotations about the three axes, a transformation from one reference frame to the next can be obtained. Euler angles used for the transformation are represented by the rotation about their respective axis roll(ϕ) about the x-axis, pitch(θ) about the y-axis, and yaw(ψ) about the z-axis.

Euler Angle Vector Transformation.

By using the rotation about the perspective axis three separate DCMs are defined as seen below:

$$\mathbf{C}_1 = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$\mathbf{C}_2 = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{18}$$

$$\mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{19}$$

where $\mathbf{C}_1$ is the rotation around the z axis, $\mathbf{C}_2$ is the rotation about the y axis, and $\mathbf{C}_3$ is the rotation about the x axis. By combining the three separate DCMs a transformation matrix, capable of transforming from the sensor to the body axes, can be obtained:

$$\mathbf{C}_s^b = \mathbf{C}_3\mathbf{C}_2\mathbf{C}_1 \tag{20}$$

Quaternions.

The final method that will be discussed in this thesis, used to transformed one frame into another, is the quaternion. By using a single rotation about a vector, defined in the reference frame, the quaternion attitude representation can be used to transform one co-ordinate frame to another [20]. This method is preferred when dealing with a single rotation about one axis and will be used directly when simulating the IMU measurements later on.

Quaternion Vector Transformation.

The quaternion is a four element vector that is a translation of an angular movement from one co-ordinate frame to another frame. To establish the quaternions needed to rotate from one frame to another, the angle rotation values about each axis must be used. The angle vector $\boldsymbol{\mu}$ is expressed as three separate components of the angel vector by $\boldsymbol{\mu}_x$, $\boldsymbol{\mu}_y$, and $\boldsymbol{\mu}_z$ and has a magnitude μ. The quaternion values can then be obtained by [20]:

$$\mathbf{q} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} cos(\mu/2) \\ \boldsymbol{\mu}_x/\mu sin(\mu/2) \\ \boldsymbol{\mu}_y/\mu sin(\mu/2) \\ \boldsymbol{\mu}_z/\mu sin(\mu/2) \end{bmatrix} \tag{21}$$

The quaternions values obtained can then be used to form a direction cosine matrix that can be used to rotate from one frame to another:

$$\mathbf{C} = \begin{bmatrix} (a^2+b^2-c^2-d^2) & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & (a^2-b^2+c^2-d^2) & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & (a^2-b^2-c^2+d^2) \end{bmatrix} \tag{22}$$

Just like the DCM seen earlier in this chapter, this **C** value can be used to rotate from one frame to another:

$$\mathbf{n}^s = \mathbf{C}\mathbf{n}^b \tag{23}$$

where **C** is used to rotate n from the body frame to the sensor frame.

2.4 Reference System

This section defines the reference system that will be used through out this thesis. It is known that the earth is not shaped as an exact sphere. Instead, it is shaped more like an ellipsoid with a slight flattening occurring at the poles. To correct for the flattening of the earth at the poles, the World Geodetic System 1984 (WGS 84) coordinate system serves as a geometric shape, represented as an ellipsoid, where the geometric mass has a fixed point at the center of the mass with a x, y and z axes [2]. The location of a point p on an ellipsoid is defined by its geodetic latitude, longitude, and height. Geodetic latitude at a point on the surface of the earth is the angle between the equatorial plane and a line normal to the reference ellipsoid, which passes through the point [20]. Meridian radius of curvature and transverse radius of curvature for point p, that has geodetic latitude and longitude, can be found using [20]

$$R_N = \frac{R(1-e^2)}{(1-e^2 sin^2 L)^{3/2}} \tag{24}$$

$$R_E = \frac{R}{(1-e^2 sin^2 L)^{1/2}} \tag{25}$$

where R_N is the meridian radius of curvature, R_E is the transverse radius of the curvature, e is the major eccentricity of the ellipsoid, R is the length of the semi-major axis, and L is the geodetic latitude [20]. By using the values determined by the meridian radius of curvature and transverse radius of curvature, point p's coordinates can be converted to a local level navigational frame.

2.5 Reference Frames

In order to avoid errors while navigating, a system needs to be defined in the correct reference frame. The reference frames that will be used in this thesis, as described in [20], will be discussed in this section.

Inertial Frame.

The inertial frame (i-frame) has an origin that is centered at the Earth's center and the axes are fixed with respect to the stars. The axes are defined by x_i, y_i, and z_i, as seen in Figure 6. In this frame of reference, the z_i axis coincides with the Earth's polar axis that is assumed to be invariant in direction.

Earth Frame.

The Earth frame (e-frame) has an origin that is centered at the center of the Earth, and the axes are fixed to the Earth. The axes are defined by x_e, y_e, and z_e in Figure 6. Just like the inertial frame, the z_e axis coincides with the the Earth's polar axis. The x_e axis lies along the intersection of the Greenwhich meridian plane and the Earth's equatorial plane. The Earth frame rotates about the z_i axis, with respect to the inertial frame, at a rate of Ω.

Navigation Frame.

The navigation frame (n-frame) is a local geographical frame that has its origin at the location of the navigational system. The position of the navigation frame p has axes that are aligned in the directions of north, east and local vertical (down) as shown in Figure 6. Rotation rate and velocity in the north, east and down direction of the navigation frame with respect to the earth frame, $\boldsymbol{\omega}_{en}^{n}$, is dictated by the location of the position p on the Earth. Rotation rate in the navigation frame with respect to

the earth's frame is measured as:

$$\boldsymbol{\omega}_{en}^{n} = \begin{bmatrix} \frac{V_E}{R_E+h} & \frac{-V_N}{R_N+h} & \frac{-V_E tanL}{R_N+h} \end{bmatrix}^T \tag{26}$$

where R_N and R_E are the meridian radius of curvature and transverse radius of curvature that were defined in (2.4). The measured velocity of the system in the navigation frame are detailed as V_E and V_N. The skew symmetric form, $\boldsymbol{\Omega}_{en}^{n}$, of the rotation rate is given as

$$\boldsymbol{\Omega}_{en}^{n} = \begin{bmatrix} 0 & \frac{V_E tanL}{R_N+h} & \frac{-V_N}{R_N+h} \\ \frac{-V_E tanL}{R_N+h} & 0 & \frac{-V_E}{R_E+h} \\ \frac{V_N}{R_N+h} & \frac{V_E}{R_E+h} & 0 \end{bmatrix} \tag{27}$$

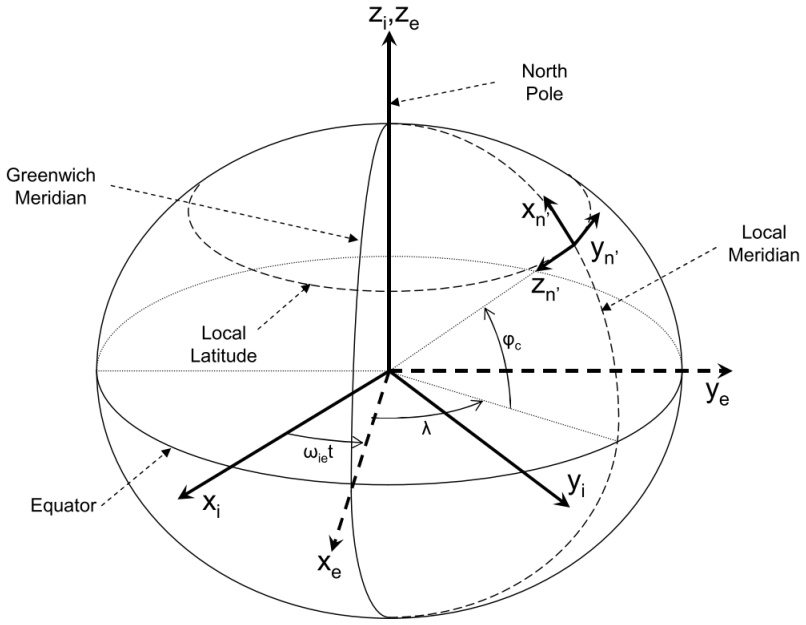

Figure 6. Earth , Inertial, and Navigation reference frames [23]

Body Frame.

The body frame (b-frame) is used to describe the orientation of a vehicle with an origin centered at the navigation frame. Figure 7 shows the axes in which the body rotates around to give the roll, pitch, and yaw of the body.

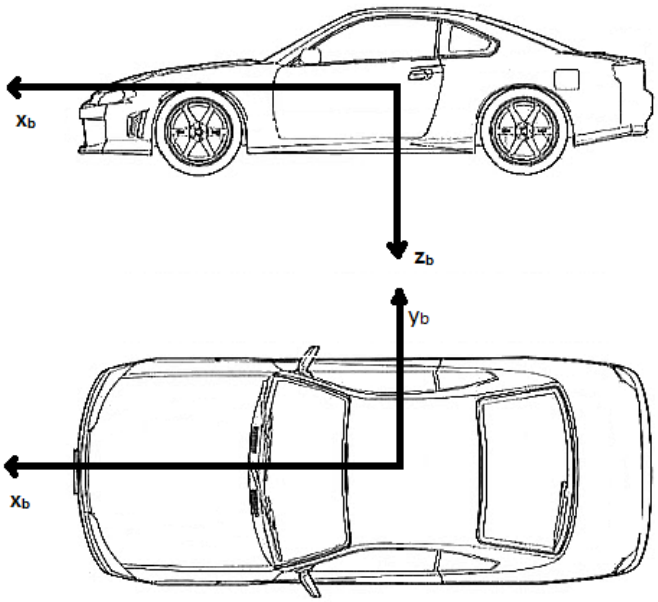

Figure 7. Body Frame of Reference

2.6 Kalman Filtering

This section will provide the back ground of the Kalman filter and the extended Kalman filter. These filters will be used through out this thesis to estimate states of a given system.

System Dynamics.

The Kalman filter is an optimal recursive data processing algorithm[9]. Background information pertaining to the Kalman filter has been defined many times before, and instead of establishing a new layout for the formulas, this thesis will draw from formulas presented in [23]. By knowing the systems dynamics, initial conditions,

and all statistical information describing the system, the Kalman filter can give the optimal estimate of a linear system. Before introducing the Kalman filter, a stochastic linear system model needs to be defined.

The stochastic linear system model is defined as a differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{28}$$

where $\mathbf{F}$ is the homogeneous system dynamics, $\mathbf{x}(t)$ is the state vector, $\mathbf{B}$ is the input matrix, $\mathbf{u}$ is the input vector, $\mathbf{G}$ is the noise matrix, and $\mathbf{w}(t)$ is the vector of Gaussian white noise. The noise vector has process noise strength $\mathbf{Q}(t)$ where:

$$E\{\mathbf{w}(t)\mathbf{w}^T(t+\tau)\} = \mathbf{Q}(t)\delta(\tau) \tag{29}$$

where $\delta(t)$ is the Dirac delta function. As stated earlier in this section, one of the key components needed by the Kalman filter to characterize the system, is the system's statistics. The system's mean ($\mathbf{m}_x(t)$) and covariance ($\mathbf{P}_{xx}(t)$) can be determined by:

$$\mathbf{m}_x(t) = E\{\mathbf{x}(t)\mathbf{x}^T(t)\} \tag{30}$$

$$\mathbf{P}_{xx}(t) = E\{\mathbf{x}(t)\mathbf{x}^T(t)\} - \mathbf{m}_x(t)\mathbf{m}_x^T(t) \tag{31}$$

where $E\{\}$ is the expectation operator.

The previous equations in this section characterize the system in continuous time. The Kalman filter deals with measurements as discrete inputs. The current time will be denoted by t_k, and the previous time step will be denoted by t_{k-1}. The discrete-time model will then have the form of:

$$\mathbf{x}(t_k) = \mathbf{\Phi}(t_k, t_{k-1})\mathbf{x}(t_{k-1}) + \mathbf{B}(t_k)\mathbf{u}(t_k) + \mathbf{G}(t_k)\mathbf{w}(t_k) \tag{32}$$

where $\mathbf{\Phi}$ is the state transition matrix, $\mathbf{x}$ is the vector of the discrete states, $\mathbf{B}$ is the discrete input matrix, $\mathbf{u}$ is the vector of discrete inputs, $\mathbf{G}$ is the discrete noise matrix, and $\mathbf{w}$ is the discrete noise strength. The state transition matrix $\mathbf{\Phi}$ can be calculated as:

$$\mathbf{\Phi}(t_k, t_{k-1}) = e^{\mathbf{F}(t_k)\Delta t} \tag{33}$$

where Δt is the change in time.

The discretized noise power matrix $\mathbf{Q}_d(t_k)$ will be determined by using the Van Loan method [22]. The three step Van Loan method is given as

$$\mathbf{A}^* = \begin{bmatrix} -\mathbf{F}(t_k) & \mathbf{G}(t_k)\mathbf{Q}(t_k)\mathbf{G}(t_k)^T \\ 0 & \mathbf{F}^T \end{bmatrix} \Delta t \tag{34}$$

$$\mathbf{B}^* = e^{\mathbf{A}^*} = \begin{bmatrix} ... & \mathbf{\Phi}(t_k)^{-1}\mathbf{Q}_d(t_k) \\ 0 & \mathbf{\Phi}_d^T \end{bmatrix} = \begin{bmatrix} \mathbf{B}11 & \mathbf{B}12 \\ \mathbf{B}21 & \mathbf{B}22 \end{bmatrix} \tag{35}$$

$$\mathbf{\Phi}(t_k) = (\mathbf{\Phi}(t_k)^T)^T \tag{36}$$

$$\mathbf{Q}_d(t_k) = \mathbf{B}_{22}^T\mathbf{B}_{12} = \mathbf{\Phi}(t_k)\mathbf{\Phi}(t_k)^{-1}\mathbf{Q}_d(t_k) \tag{37}$$

Discrete measurements for a linear system are modeled as:

$$\mathbf{z}(t_k) = \mathbf{H}(t_k)\mathbf{x}(t_k) + \mathbf{v}(t_k) \tag{38}$$

where $\mathbf{H}(t_k)$ is the observation matrix, $\mathbf{x}(t_k)$ is the state vector at that time, and $\mathbf{v}(t_k)$ is the discretized noise vector. Similarly, discrete measurements for a non linear system will be modeled as:

$$\mathbf{z}(t_k) = \mathbf{h}(\mathbf{x}(t_k), t_k) + \mathbf{v}(t_k) \tag{39}$$

where $\mathbf{h}$ is the non linear function of $\mathbf{x}(t_k)$ at time t_k.

Kalman Filter Equations.

This section will define the equations that are associated with the linear Kalman filter. The Kalman filter has two modes that it operates in; it is either propagating a measurement forward or updating the system with a new measurement. During the propagation state, the system is using the state transition matrix $\mathbf{\Phi}(t_k)$ to propagate the states estimate forward in time. Not only are the states propagated forward, but the systems error covariance matrix is propagated through time as well. The Kalman filter propagation model is shown here:

$$\hat{\mathbf{x}}(t_k^+) = \mathbf{\Phi}(t_k)\hat{\mathbf{x}}(t_k^-) \tag{40}$$

$$\hat{\mathbf{P}}(t_k^+) = \mathbf{\Phi}\hat{\mathbf{P}}(t_k^-)\mathbf{\Phi}^T + \mathbf{Q}_d(t_k) \tag{41}$$

where $\hat{\mathbf{x}}$ is an estimate of the state's mean, $\hat{\mathbf{P}}$ is an estimate of the covariance of the system, t_k^- denotes the estimate of the covariance and mean at the time immediately prior to t_k and t_k^+ denotes the estimate of the mean and covariance after the update. When estimates of the covariance and mean are not available:

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) \tag{42}$$

$$\hat{\mathbf{P}}(t_k^+) = \hat{\mathbf{P}}(t_k^-) \tag{43}$$

When a measurement, $\mathbf{z}(t)$, from the system becomes available, an update of the values are performed. The update equation for the Kalman filter is:

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)(\mathbf{z}(t_k) - \mathbf{H}(t_k)\hat{\mathbf{x}}(t_k^-)) \tag{44}$$

$$\hat{\mathbf{P}}(t_k^+) = (\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k))\hat{\mathbf{P}}(t_k^-) \tag{45}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{K}_t$ is the Kalman gain. The Kalman gain given by

$$\mathbf{K}(t_k) = \hat{\mathbf{P}}(t_k^-)\mathbf{H}(t_k)^T(\mathbf{H}(t_k)\hat{\mathbf{P}}(t_k^-)\mathbf{H}(t_k)^T + \mathbf{R}(t_k))^{-1} \tag{46}$$

optimally weights the measurements that are received. The $\mathbf{R}$ value in the Kalman gain equation is the covariance of the measurements and is defined by:

$$\mathbf{R}(t_k) = E[(\mathbf{v}(t_k))(\mathbf{v}(t_k)^T] \tag{47}$$

The Kalman filter, if modeled correctly (i.e., errors in the filter to do not create a non determinant solution) and provided with new state estimates, will continually cycle between propagate and update. If the system is not modeled correctly, the Kalman filter eventually collapses because the state transition matrix will become non-invertible. The best place to check if the filter is operating correctly would be the output of the residuals. Residuals ($\mathbf{r}$) of the system can be calculated by:

$$\mathbf{r} = \mathbf{z}(t_k) - \mathbf{H}(t_k)\hat{\mathbf{x}}(t_k^-) \tag{48}$$

Residual values will usually increase exponentially if the Kalman filter is modeled incorrectly.

Extended Kalman Filter.

In the case of a measurement or system that is non-linear, the normal Kalman filter can not be implemented and the use of the extended Kalman filter (EKF) is required. This section will discuss the background behind the EKF [8]. Just like the Kalman filter, the EKF has two modes, update and propagate. The EKF is described

by the dynamic model:

$$\dot{\mathbf{x}}(t_k) = \mathbf{f}[\mathbf{x}(t_k), \mathbf{u}(t_k), t_k] + \mathbf{G}(t_k)\mathbf{w}(t_k) \tag{49}$$

where the $\mathbf{x}(t_k)$ value is the state vector, $\mathbf{u}(t_k)$ is a vector of inputs , and $\mathbf{w}(t_k)$ is the zero-mean white Gaussian noise for the system. The measurement equation for the EKF is similar to that of the Kalman filter with a small change of it being able to handle the nonlinear models. The measurement equation for this filter is:

$$\mathbf{z}(t_k) = \mathbf{h}[\mathbf{x}(t_k), t_k] + \mathbf{v}(t_k) \tag{50}$$

A large part of the EKF is to be able to linearize around the measurements that are received. This action can be completed by taking the Jacobian of both the dynamics equation and the measurement equation. The Jacobian for the two equations are derived as:

$$\mathbf{F}[t_k; \mathbf{x}_n(t_k)] \triangleq \left.\frac{\delta \mathbf{f}[\mathbf{x}, \mathbf{u}(t_k), t_k]}{\delta \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t_k)} \tag{51}$$

$$\mathbf{H}[t_k; \mathbf{x}_n(t_k)] \triangleq \left.\frac{\delta \mathbf{h}[\mathbf{x}, t_k]}{\delta \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t_k)} \tag{52}$$

The EKF states and covariances are propagated in the same manner as found in Equations (41) and (40). The updates for the EKF utilize both the linear and non linear equations and can be seen here:

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}(t_k)\{\mathbf{z}(t_k) - \mathbf{h}[\hat{\mathbf{x}}(t_k^-), t_k]\} \tag{53}$$

$$\mathbf{P}(t_k^+) = \mathbf{P}(t_k^-) - \mathbf{K}(t_k)\mathbf{H}[t_k; \hat{\mathbf{x}}(t_k^-)]\mathbf{P}(t_k^-) \tag{54}$$

The Kalman gain equation for the EKF is similar to that of the Kalman filter and is calculated as:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T[t_i; \hat{\mathbf{x}}(t_i^-)]\{\mathbf{H}[t_i; \hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-)\mathbf{H}^T[t_i; \hat{\mathbf{x}}(t_i^-)] + \mathbf{R}(t_i)\}^{-1} \tag{55}$$

2.7 Previous Research

Implementation of a paired visual system and commercial IMU for a more robust navigation system in an urban environment is not a new concept. For this research, only systems that used monocular imagery along with IMU integration were reviewed. The four most related research areas that will be evaluated in this section are [15], [13],[17], and [10].

Simulation Platform for Vision Aided Inertial Navigation[15].

Png's research is the predecessor to the research that is developed in this document. Instead of implementing the vision aided navigation in a real world environment, his research simulated all of the sensors that were used. By simulating the data, normal anomalies that are inherent with IMU's and GPS could be reduced or even eliminated. The simulated system was able to apply errors in a specific area and then evaluate the results knowing that nothing else was affecting the system. A relationship between the measurements provided with different qualities of cameras was just one of the scenarios that was able to be tested. Another factor that was tested, which was difficult to achieve in this research, was a satellite vehicle (SV) that was consistently insight for the entire duration of the scenario and was not affected by outside interference (i.e., multi-path, building obstruction, or atmospheric effects).

Png's research further developed the value of having a visual odometry system to aid the system in a GPS degraded environment [15]. Conclusions drawn from his

research, that will now be verified in this research, is that the system sees a boost in overall performance when the system is aided with vision and GPS.

Vision-Aided Inertial Navigation using Planar Terrain Features [13].

Panahandeh/Jansson discuss the implementation of a ground facing monocular camera attached to an IMU that detects ground plane features. The strength behind this system is its ability to give observability to the scale of a feature while only using a single camera.

By locking the camera in a downward facing pose, scale estimation of the 3-D camera translation can be solved. Features that are detected can all be thought of as having the same distance from the camera. However, this is not the case when a camera is pointed forward or sideways. A dynamic background like this can lead to measurement errors because of the nonlinearity between features.

Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC [17].

Scaramuzza/et al discuss the use of nonholonomic constraints of a wheeled vehicle, that has an imagery system with a very fast frame rate. They also note two separate outlier rejection algorithms that can be used for feature rejection. The whole concept behind this paper is the amount of information that can be used by constraining the dynamics of the vehicle to the Ackerman steering principle [18]. A structure from motion (SFM) technique is also used for the outline of the vehicle estimated motion. This can be done because the frame rate of the camera is so quick, and there are no gaps in feature detection between frames.

Also important to note is the use of the two outlier rejection methods of the system. One of them is the single point random sample consensus (RANSAC) algorithm that

cuts down on the computational time for outlier rejection. The other is a histogram voting technique, that uses the angle calculated to each feature to provide a histogram of angles that coincide with each other. Points are then voted on and then chosen by the their angle which is related to most of the other features.

Realtime Implementation of Visual-Aided Inertial Navigation using Epipolar Constraints [10].

Norrison/et al discuss the real time implementation of a visual aided inertial navigation system that is based off of epiploar constraints. While measurements equations that are presented in their work and this thesis are going to differ, the way that they use the IMU to perceive a pure translation of the imagery system is similar. They refer to their method as rotation unwrapping of feature points (FPs) based on the IMU sensed rotation over the period between images.

IMU rotation measurements are considered to have very low error (1 deg) over time periods up to a minute [10]. By combining the rotation, given by the IMU, and the pure translation of the FPs that have been rotation unwrapped, the fundamental matrix can be well defined. Having a well defined Fundamental matrix plays a large part in their outlier rejection, which works off the principle of:

$$\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0 \tag{56}$$

where $\mathbf{F}$ is the fundamental matrix. Instead of defining the value of rejection to be zero, their research changes the value to a rejection threshold value that is defined by the angle between points and distance of movement. They then incorporate their returned measurements as errors in position. These measurements are calculated by a bundle adjustment of the current position error detected by FPs along with previously determined camera measurements that hold the most statistical relevance.

Summary.

This chapter developed the mathematical notation and specific equations that will be utilized throughout this thesis. Camera properties such as: the thin lens, pinhole camera model, and projective equations were illustrated. The relationship of consistent camera images was utilized by epipolar geometry between features in an image, to determine the essential matrix.

Attitude representation was also described as a way to move from one frame of reference to another frame of reference. The reference frames discussed in this chapter included the Earth Center Earth Fixed (ECEF), Inertial, Navigation, and the Body frame. The specific attitude of the system will be represented by DCMs, Euler angles, or quaternions.

Development of the state space model along with the techniques behind state and covariance estimation from the Kalman filter and EKF were also developed. Measurements that will be supplied to the filter, will be further developed in the next chapter.

A summary of previously accomplished work in the same area was also conducted. These research points show some ideas of additional models that could be added to the work presented in this thesis or future research. Some of the ideas that would be pertinent to include in this work could not be accomplished because of the data sets that were provided.

III. Methodology

This chapter outlines the methods that were used to complete this thesis. Illustrations of the information obtained from monocular camera computer vision algorithm, IMU, and GPS will also be shown. Additionally, measurement equations that link individual monocular computer vision algorithm, IMU, and GPS to the system along with system dynamics will be described. A block diagram describing the loosely coupled system is shown in Figure 8.

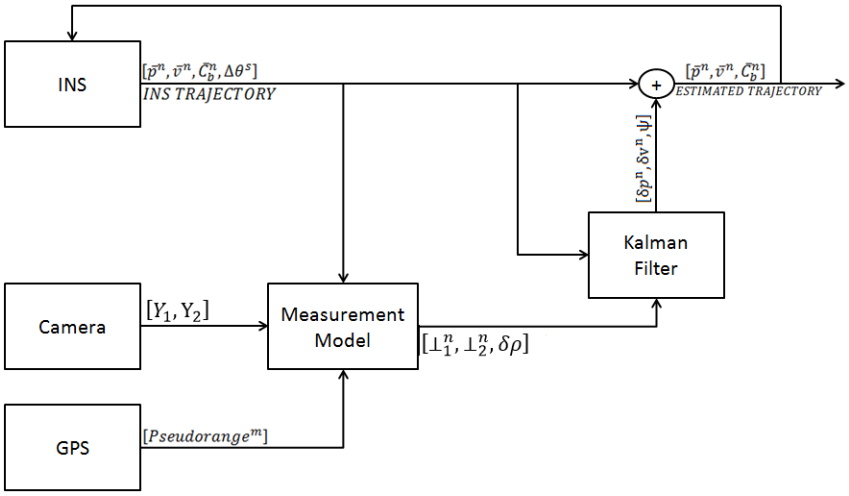

Figure 8. Loosely coupled feedback approach utilized in this work. The monocular camera system provides $\mathbf{Y}_1$ and $\mathbf{Y}_2$ which are the matched features detected in two sequential images. GPS provides pseudoranges for each satellite that is measured from the SV to the receiver on the vehicle. The INS provides estimated position ($\bar{\mathbf{p}}^n$) in the navigation frame, estimated velocity ($\bar{\mathbf{v}}^n$) in the navigation frame, estimated orientation ($\bar{\mathbf{C}}_b^n$) of the vehicle, and gyro measurement ($\Delta\boldsymbol{\theta}^s$) in the IMU's frame of reference. The Extended Kalman filter provides errors for the position ($\delta\mathbf{p}^n$) in the navigation frame, velocity ($\delta\mathbf{v}^n$) in the navigation frame, and tilt errors ($\boldsymbol{\psi}$).

3.1 Imagery Information

This section will define the characteristics of feature detection implemented in this thesis, along with development of the monocular camera computer vision algorithm 2-D zero velocity measurements that can be supplemented into the EKF for measurement updates.

Feature Detection.

As pairs of images are processed throughout this research, a feature detection and matching algorithm is being run to ensure that features between two images match. For this particular research, the Scale Invariant Feature Transform (SIFT) algorithm, defined in Section 2.2, was used as a tool for feature detection. Features detected by a monocular camera at two separated times are compared and then accepted or rejected based on the Mahalanobis distance between the descriptors. Users are allowed to define the distance ratio, which defines the minimum Euclidean distance for the descriptor from its nearest neighbor to its second closest neighbor. Depending on the value set for the distance ratio one of three things can happen: mismatches can be accepted as matches, matches can be rejected because the system is overdetermined, or correct matches can be returned. By setting the distance ratio of the system to be .35, the features that are returned as matches seem to have very few mismatches. Images are undistorted and then run through the SIFT algorithm.

After using SIFT, a pixel location of matching features between the two images is returned. By using the intrinsic camera calibration matrix, feature pixel locations are then converted to projected x and y locations as seen:

$$\mathbf{A} = \begin{bmatrix} F_x & 0 & O_x \\ 0 & F_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \tag{57}$$

$$\mathbf{s}_{proj} = \mathbf{A}^{-1}\mathbf{s}_{pix} \tag{58}$$

where $\mathbf{A}$ is the camera calibration matrix, F_x and F_y are determined by the focal length and pixel count for both the height and width of the image, O_x and O_y are the intersection points of the optical axis with the image plane, and $\mathbf{s}_{proj}$ and $\mathbf{s}_{pix}$ are

the vectors of the camera frame and the projected frame. By way of demonstration, Figure 9 shows a pair of images with a line connecting the features that have been matched by the SIFT algorithm.

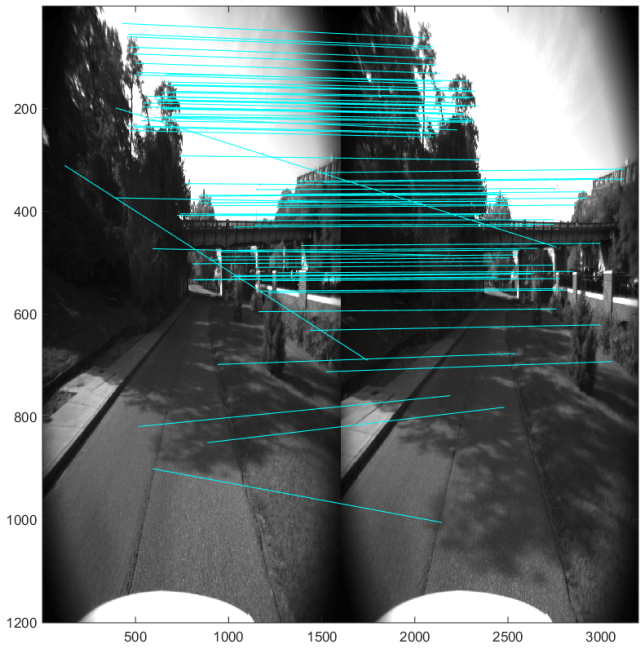

Figure 9. Sequence of two images side-by-side with matching features connected by a green line.

Outlier Rejection.

Due to the possibility of incorrect features being matched between subsequent images, a way of rejecting points had to be determined. There are several different ways that outliers can be rejected from the system. One way of removing outliers is to use a histogram of closely related angular values that relate the features in the first image to the features in the second image, and to only keep those points that fall into the most heavily populated bins [17]. This method is used to detect the amount of movement by its corresponding angular movement in the image. Another way of rejecting outliers is to use the epipolar constraints of the system and use the calculated

fundamental matrix to reject points [10]. The fundamental matrix could be calculated one of two ways: with rotation and translation measurements given by the IMU or by using the matched features in the image. For this thesis, a best fit fundamental matrix from image features was used for outlier rejection. A RANSAC [3] algorithm, which calculated the best fit fundamental matrix, was implemented and used during the course of this entire work. In addition to setting a low distance ratio when performing SIFT on images, the use of RANSAC was utilized to reject ten percent of the points that were run through the algorithm. A ten percent point rejection by the RANSAC algorithm was accomplished by setting the rejection threshold very high, and then through an iterative process, the threshold was decreased until no more than ninety percent of the matched features remained. Using the same two images that were seen in Figure 9, the features that have now been rejected are connected by a red line and shown in Figure 10.

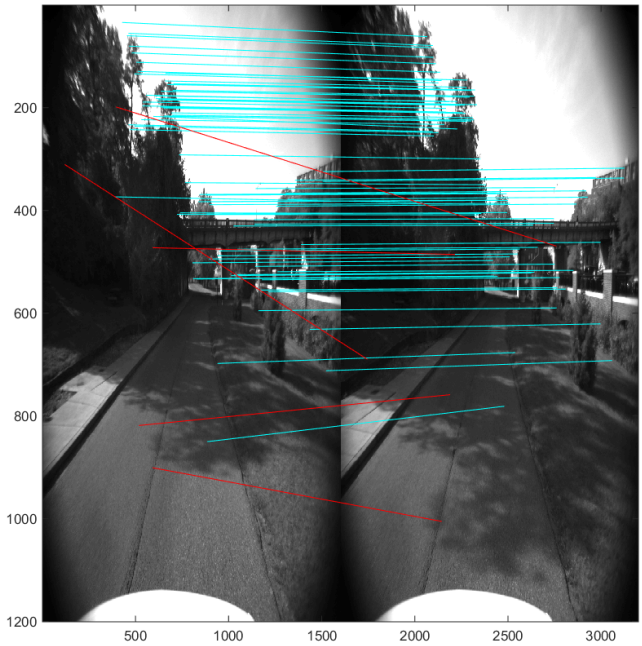

Figure 10. Sequence of two images side-by-side with matching features connected by a green line and RANSAC rejected features connected by a red line.

Camera Measurements.

Based off of the feature's projected location between images, the values of the translation in the camera frame could be obtained. A common usage of feature points is in an eight point algorithm [4] to determine the essential or fundamental matrix, as described in chapter two. Once the essential matrix, defined in [4], has been determined, a solution for the translation $\mathbf{t}$ and rotation $\mathbf{C}_{image_1}^{image_2}$ can be solved for, so that their values satisfy the equation:

$$\mathbf{E} = \mathbf{C}_{image_1}^{image_2}[\mathbf{t}]_\times \tag{59}$$

However, multiple solutions can be derived from this equation and even when a solution is thought to be correct, the slightest deviation in the system's rotation can be costly to the translation vector returned. Instead, this research looks at using the rotation obtained from the IMU, and only solving for the translation of the system ($\mathbf{t}$). IMU's, even at the commercial grade level, provide very reliable rotation measurements for short periods of time. Common methods of IMU integration with images is done by using the vehicle's orientation at the time of images [24]. For this research, the rotation, sensed by the IMU, of the system over the time period between the images is tracked and stored. Once the second image is obtained, the rotations are then integrated up to the time of the image. Next, the final rotation is translated from the IMU's frame of reference into the camera's frame of reference. Features from the first image are then rotated into the second images frame of reference using the rotation matrix obtained from the IMU's rotation measurements. By rotating features from one frame into another, the only disparity left from the feature would be contributed to the translation of the vehicle. Finally, a matrix of translation values is obtained that relates all of the features to a specific translation sensed by them. Due to lack

of depth knowledge, or distance of features in the direction of movement, there can be an unobservability in the magnitude of the translation vector. To mitigate the problem and correct to a best fit translation measurement, the use of singular value decomposition (SVD) was used for the set of homogeneous solutions that are found. An outline of the equations used to compute the translation of the vehicle, defined in [19], can be seen here:

$$\mathbf{Y}_1^{image_2} = \mathbf{C}_{image_1}^{image_2} * \mathbf{Y}_1^{image_1} \tag{60}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{Y}_2^{image_2}(2) * \mathbf{Y}_1^{image_2}(3) - \mathbf{Y}_1^{image_2}(2) \\ \mathbf{Y}_2^{image_2}(1) * \mathbf{Y}_1^{image_2}(1) - \mathbf{Y}_1^{image_2}(3) \\ \mathbf{Y}_2^{image_2}(1) * \mathbf{Y}_1^{image_2}(2) - \mathbf{Y}_2^{image_2}(2) * \mathbf{Y}_1^{image_2}(3) \end{bmatrix} \tag{61}$$

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = SVD(\mathbf{T}) \tag{62}$$

where $\mathbf{Y}_1^{image_1}$ are the set of features located image one in image one's reference frame, $\mathbf{Y}_2^{image_2}$ are the set of features located in image two in image two's reference frame, $\mathbf{Y}_1^{image_2}$ are the features from image one rotated into the second image's reference frame, $\mathbf{C}_{image_1}^{image_2}$ is the DCM used to rotate features from image one to image two, $\mathbf{T}$ is a matrix of the sensed translation values for all features between the two images, $\mathbf{U}$ is a matrix of the left singular vectors of $\mathbf{T}$, and $\mathbf{S}$ is a diagonal matrix that represents the singular values of $\mathbf{T}$. The left singular vector of $\mathbf{U}$ associated with the smallest singular value of $\mathbf{S}$ is then used as the measured translation vector $\mathbf{t}^b$ as sensed by the monocular camera computer vision algorithm in the body's frame of reference.

Zero Velocity Vectors.

Instead of using the exact translation vector $\mathbf{t}^b$ computed from the monocular camera computer vision algorithm, 2-D zero velocity vectors orthogonal to the direction of travel will be calculated. Because the velocity vector $\mathbf{t}^b$ has no magnitudes

associated with its values, $\mathbf{t}^b$ can only really give a sense of direction and not the overall magnitude of movement. By calculating the 2-D zero velocity vectors and applying them to the velocity measurements given by the INS, errors in the direction of movement can be evaluated instead of the magnitude of movement. Before converting the translation vector $\mathbf{t}^b$ into the 2-D zero velocity vector, the translation vector has to be rotated into the correct sensor orientation. In this case, the translation vector, predicted by the images, would be rotated from the second image camera's orientation into the navigation frame. However, the translation vector obtained by the computer vision algorithm would not be rotated into the navigation frame defined at the time of the final image. The reason for not rotating the translation into the navigation frame at the time of the second image is that the translation sensed by the cameras is not for a specific point in time, but is better depicted as translation sensed over a period of time between two separate poses. As a result, the values of the sensed translation vector are instead rotated into the navigation frame of the vehicle at a time halfway between the two images. This can be done because the time between measurements is small, so there is not much rotation. Once the translation vector $\mathbf{t}^b$ has been rotated into the correct frame, the 2-D zero velocity vectors can be obtained. An example of how these orthogonal vectors are obtained is represented here:

$$\mathbf{t}^n = \mathbf{C}_b^n * \mathbf{t}^b \tag{63}$$

$$\begin{bmatrix} \perp_1^n \\ \perp_2^n \end{bmatrix} = N(\mathbf{t}^n) \tag{64}$$

where $\mathbf{C}_b^n$ is the DCM that rotates the translation from the body's frame of reference to the navigation's frame of reference at a time in between the two images, N is used to denote the operation of determining the null space of the translation vector $\mathbf{t}^n$, and $\perp_1^n$ and $\perp_2^n$ are the two orthogonal vectors in the navigation frame.

3.2 IMU Measurements

Due to lack of data within some of the ASPN scenarios tested in this research, simulated commercial grade IMUs had to be generated in order to complete the work outlined in this thesis. This section will develop the process in which these IMU measurements were generated.

Truth Data.

To create the simulated commercial grade IMUs, a truth data set was utilized. Truth data sets were created with differential GPS and a navigation grade IMU during the time of the scenario runs. Information that was contained in these data sets were latitude, longitude, altitude, velocity in the north, east and down directions, time of measurements, roll, pitch, and yaw. All of the measurements for the vehicle were taken in the navigation frame. Measurements for the truth system were generated at 10 Hz.

Data Interpolation.

To turn the truth data sets into measurements that resembled an IMU, and were usable by the camera, the time between measurements had to be drastically reduced. For this research, IMU measurements were generated at 100 Hz; therefore, this meant that the truth data first had to be interpolated at 100 Hz.

Interpolation of the data was very straight forward for the time, latitude, longitude, and altitude. When interpolating the roll, pitch and heading, the values had to be transformed into quaternions. By doing so, a rotation about a single axis could be implemented.

Simulated Measurements.

From the interpolated data, IMU measurements that represent the true integrated specific force and angular rate were calculated. Next, the measurements were corrupted according to an error model. Commercial grade IMU error values that were used can be seen here:

Table 1. Commercial Grade IMU Parameters

σ_{gyro}	8.7E-3
GyroTimeConstant	3600
AngularRandomWalk (ARW)	6.5E-4
σ_{accel}	1.96E-1
AccelTimeConstant	3600
VelocityRandomWalk (VRW)	4.3E-3

From the error values given, the commercial grade IMU could be generated using the integrated measurement equations derived in [14]. These equations will be used to simulate the error induced IMU measurements from the true system's measurements.

Integrated Angular Rate Gyro Measurement.

Integrated angular rate measurements were obtained from:

$$\Delta\boldsymbol{\theta}_{meas} = \Delta\boldsymbol{\theta}_{true} + \mathbf{b}_{\Delta\theta} + \mathbf{w}_{\Delta\theta} \tag{65}$$

where $\Delta\boldsymbol{\theta}_{true}$ is the true change in angular rate for the time period since the last measurement, $\mathbf{b}_{\Delta\theta}$ is the time correlated bias (TCB) error, and $\mathbf{w}_{\Delta\Theta}$ is the measurement's white Gaussian noise error. $\mathbf{b}$ has an initialization value that is determined by the system's gyro time correlated bias sigma. After initialization, the system bias,

modeled as a first-order Gauss Markov (FOGM) process, in which:

$$\dot{\mathbf{b}} = -\frac{1}{T}\mathbf{b} + \mathbf{w}_b \tag{66}$$

where T is the time constant, $\mathbf{b}$ is the time correlated gyro bias, and $\mathbf{w}_b$ is a white Gaussian noise. To be useful, the system now needs to be modeled in discrete time. From [9] the state transition matrix for the system is given as:

$$\mathbf{\Phi}(t_k, t_{k-1}) = e^{-(t_k - t_{k-1})/T} \tag{67}$$

By letting the transient values die out, where t_o approaches negative infinity, the discrete noise value can be calculated from the variance as [9]:

$$\begin{aligned} E[\mathbf{b}(t)^2] &= \sigma_b^2 = \frac{1}{2}\mathbf{Q}_d(t_k)T \\ \mathbf{Q}_d(t_k) &= \frac{2\sigma_b^2}{T} \end{aligned} \tag{68}$$

where σ_b^2 is the variance of the system. White Gaussian noise that is added to the system in the discrete time is expressed as:

$$\sigma_{w_{\Delta\theta}}^2 = ARW^2 \Delta t \tag{69}$$

where the ARW is a value associated with the grade of IMU being used.

Integrated Specific Force Measurement.

Integrated specific force measurements are very similar to those found for the integrated angular rate gyro measurements, with the main difference coming from the values used to model the noise measurements. The integrated specific force measure-

ment is modeled as:

$$\Delta \mathbf{v}_{meas} = \Delta \mathbf{v}_{true} + \mathbf{b}_{\Delta v} + \mathbf{w}_{\Delta v} \tag{70}$$

where $\Delta \mathbf{v}_{true}$ is the change in velocity over the time period, $\mathbf{b}_{\Delta v}$ is the TCB error, and $\mathbf{w}_{\Delta v}$ is the measurement's white Gaussian noise error. The values for $\mathbf{\Phi}$ and $\mathbf{Q}_d(t_k)$ are expressed in the same manner as the angular rate gyro measurement. The Gaussian noise is even expressed in the same way, except the angular random walk (ARW) in now changed to velocity random walk (VRW) as seen here:

$$\sigma^2_{w_{\Delta v}} = VRW^2 \Delta t \tag{71}$$

where VRW is the velocity random walk associated with the specific IMU.

3.3 INS Error Model

This section develops navigation error state vector $\delta \mathbf{x}$ for the inertial navigation system (INS) error models used to relate position error ($\delta \mathbf{p}^n$), velocity error ($\delta \mathbf{v}^n$), tilt error ($\boldsymbol{\psi}$), accelerometer bias ($\mathbf{a}^b$), and gyro bias ($\mathbf{b}^b$), seen in Equation 72, to their respective states [23]. Accelerometer bias and gyro bias have already been discussed in Section 3.2, and will only be implemented in the error states in this section. These models will then be used in the system's state-space model to propagate errors when error measurements are not available.

$$\delta \mathbf{x} = \begin{bmatrix} \delta \mathbf{p}^n \\ \hdashline \delta \mathbf{v}^n \\ \hdashline \boldsymbol{\psi} \\ \hdashline \mathbf{a}^b \\ \hdashline \mathbf{b}^b \end{bmatrix}_{15x1} \tag{72}$$

Attitude Errors.

Attitude errors are modeled as the vector ($\boldsymbol{\psi}$) that takes into account the navigation frame angle errors in the north, east and down axes. This vector is a right hand system defined as:

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_n \\ \psi_e \\ \psi_d \end{bmatrix} \tag{73}$$

where ψ_n, ψ_e, and ψ_d are the small angle errors in the navigation frame. From [23], the following linearized angular error differential equation has been derived:

$$\dot{\boldsymbol{\psi}} = -[(\mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e)\times]\boldsymbol{\psi} - \mathbf{C}_b^n \mathbf{b}^b - \mathbf{C}_b^n \mathbf{w}_b^b \tag{74}$$

where $(\mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e)\times$ is the Earth's sidereal angular rate rotated in the navigation frame and displayed in the skew-symmetric form, $\mathbf{b}^b$ is the gyroscope measurement, $\mathbf{C}_b^n$ is the DCM from the body frame to the navigation frame, and $\mathbf{w}_b^b$ is the additive white Gaussian noise process of the gyro.

Velocity Error.

Velocity errors are modeled in the vector form of $\delta\mathbf{v}^n$ and provide velocity errors sensed in that navigation frame in the north, east and down directions. This is also a right hand coordinate system that looks like:

$$\delta\mathbf{v}^n = \begin{bmatrix} \delta\mathbf{v}_n \\ \delta\mathbf{v}_e \\ \delta\mathbf{v}^d \end{bmatrix} \tag{75}$$

where $\delta \mathbf{v}_n$ is the IMU's velocity error in the navigation's frame north direction, $\delta \mathbf{v}_e$ is the IMU's velocity error in the navigation's frame east direction, and $\delta \mathbf{v}^d$ is the IMU's velocity error in the navigation's frame down direction. The linear stochastic velocity error model derived in [23] is given as:

$$\delta \dot{\mathbf{v}}^n = \mathbf{C}_e^n \mathbf{G}_g \mathbf{C}_n^e \delta \mathbf{p}^n - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \delta \mathbf{v}^n + (\mathbf{f}^n \times) \boldsymbol{\psi} + \mathbf{C}_b^n \mathbf{a}^b + \mathbf{C}_b^n \mathbf{w}_a^b \tag{76}$$

where $\mathbf{G}_g$ is the gradient of the gravity vector as described in [23], $\mathbf{p}^n$ is the position of the vehicle on the navigation frame, $\boldsymbol{\Omega}_{ie}^e$ is the rotation of the earth with respect to the inertial frame in a skew symmetric form, $(\mathbf{f}^n \times)$ is the skew-symmetric specific force in the navigation frame, $\mathbf{a}^b$ is the accelerometer bias in the body frame, $\mathbf{C}_n^e$ is the DCM from the navigation frame to the ECEF frame, and $\mathbf{w}_a^b$ is an additive white Gaussian noise process of the accelerometer in the body frame.

From the velocity error, the position error can then be derived from the kinematic relationship between the position and velocity

$$\delta \dot{\mathbf{p}}^n = \delta \mathbf{v}^n \tag{77}$$

State Space Model.

This section defines the state-space model that was used for the error models provided in Section 3.3. The state-space model used for this research was

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta \mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{78}$$

where $\mathbf{B}(t)$ has been omitted because there are no inputs into the system.

The linear dynamics matrix F [23], that is a time-varying function, is given by

$$\mathbf{F} = \left[\begin{array}{c|c|c|c|c} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \hline \mathbf{C}_e^n \mathbf{G}_g \mathbf{C}_n^e & -2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e & (\mathbf{f}^n \times) & \mathbf{C}_b^n & \mathbf{0}_3 \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & -(\mathbf{C}_e^n \boldsymbol{\omega}_{ie}^e) \times & \mathbf{0}_3 & -\mathbf{C}_b^n \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\frac{1}{\tau_a}\mathbf{I}_3 & \mathbf{0}_3 \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\frac{1}{\tau_b}\mathbf{I}_3 \end{array}\right]_{15x15} \tag{79}$$

The noise distribution matrix $\mathbf{G}$ is given as

$$\mathbf{G} = \left[\begin{array}{c|c|c|c} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \hline \mathbf{C}_b^n & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \hline \mathbf{0}_3 & -\mathbf{C}_b^n & \mathbf{0}_3 & \mathbf{0}_3 \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \hline \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{array}\right]_{15x12} \tag{80}$$

The noise source, $\mathbf{w}$, is defined by variables that have already been derived in this thesis, and takes on the form of

$$\mathbf{w} = \left[\begin{array}{c} \mathbf{w}_a^b \\ \hline \mathbf{w}_b^b \\ \hline \mathbf{w}_{a_{bias}}^b \\ \hline \mathbf{w}_{b_{bias}}^b \end{array}\right]_{12x1} \tag{81}$$

3.4 Measurement Updates

This section defines the measurement updates that are provided to the EKF described in Section 2.6. More importantly, this section will define the measurements updates that are provided by the monocular camera computer vision algorithm and

GPS measurements.

Camera Measurements Updates.

The 2-D zero velocity measurements were defined in section 3.1, and will now be implemented into the EKF in this section. Values obtained by the 2-D zero velocity measurements represent the directions that a vehicle is known to be not moving in. The reason for taking the 2-D zero velocity measurements as the orthogonal vectors, of the camera's generated vector, is because the true magnitude of movement is an unknown element of a monocular camera. A monocular camera system does not provide distance measurements to a detected feature, thus leaving the system as a direction of movement observation only. Overall magnitude of movement for the system is given by the INS's velocity measurements. Error measurements are generated from the INS velocity measurements in the north, east, and down direction being projected onto the 2-D zero velocity vectors generated by the monocular camera computer vision algorithm. These error measurements, generated from the INS velocity measurements being projected onto the 2-D zero velocity vectors, are computed by the dot product as seen in:

$$\mathbf{z}(t_k) = H(t_k)x(t_k) + v(t_k) \tag{82}$$

$$z_1^n = \begin{bmatrix} \perp_{(1,n)}^n & \perp_{(1,e)}^n & \perp_{(1,d)}^n \end{bmatrix} \bullet \begin{bmatrix} \mathbf{v}_{(n)}^n & \mathbf{v}_{(e)}^n & \mathbf{v}_{(d)}^n \end{bmatrix} \tag{83}$$

$$z_2^n = \begin{bmatrix} \perp_{(2,n)}^n & \perp_{(2,e)}^n & \perp_{(2,d)}^n \end{bmatrix} \bullet \begin{bmatrix} \mathbf{v}_{(n)}^n & \mathbf{v}_{(e)}^n & \mathbf{v}_{(d)}^n \end{bmatrix} \tag{84}$$

$$\mathbf{z}_{camera}(t_k) = \begin{bmatrix} z_1^n \\ z_2^n \end{bmatrix} \tag{85}$$

where $\mathbf{v}^n$ is the velocity vector generated by the INS in the navigation frame of north, east and down direction. It is important to note the time in which these measurements

happen. Since the camera measurement is taken over a period of time, and can't be related directly to a specific time instance, a way to relate the measurement to a specific INS measurement had to be devised. To accomplish this, the camera's 2-D zero velocity measurement was rotated into the navigation frame of reference using a DCM that was generated by the INS at a time between the two images. In turn, the velocity measurement that was taken from the INS was also taken at the same time. This means that there was a slight difference in the time the error measurement was being generated for and the actual velocity it was being related to. Because the dynamics of a terrestrial vehicle are not very high, the effects of doing this could be considered negligible.

The zero velocity that was generated by the camera measurement was then used to populate the measurement observation matrix $\mathbf{H_{image}(t_k)}$ in the following manner:

$$\begin{bmatrix} \mathbf{H_{(1,image)}} \\ \mathbf{H_{(2,image)}} \end{bmatrix} = \begin{bmatrix} \perp^n_{(1,n)} & \perp^n_{(1,e)} & \perp^n_{(1,d)} \\ \perp^n_{(2,n)} & \perp^n_{(2,e)} & \perp^n_{(2,d)} \end{bmatrix} \tag{86}$$

$$\mathbf{H}_{camera} = \left[\begin{array}{c|c|c|c|c} \mathbf{0}_{(1x3)} & \mathbf{H_{(1,image)}}_{(1x3)} & \mathbf{0}_{(1x3)} & \mathbf{0}_{(1x3)} & \mathbf{0}_{(1x3)} \\ \hline \mathbf{0}_{(1x3)} & \mathbf{H_{(2,image)}}_{(1x3)} & \mathbf{0}_{(1x3)} & \mathbf{0}_{(1x3)} & \mathbf{0}_{(1x3)} \end{array}\right]_{2x15} \tag{87}$$

where the values $\mathbf{H_{(1,image)}}$ and $\mathbf{H_{(2,image)}}$ are are placed in the columns that correspond to the velocity states.

Because there was no known measurement noise for this type of measurement, a test had to be run to solve for the noise that was associated with this measurement. Noise for the system was computed by using the true velocity at the time of a computer vision measurement and reading the actual measurement value that was produced. It is known that:

$$\mathbf{z}(t_k) = \mathbf{H}(t_k)\mathbf{x}(t_k) \tilde{=} 0 \tag{88}$$

if the system had no noise associated with it and true velocity values were implemented, the measurement should be equal to zero. Knowing that there is some noise associated with the system, the equation can then be re-written to obtain the noise associated with the computer vision measurements

$$\mathbf{z}(t_k) = \mathbf{v}(t_k) \tag{89}$$

By taking realization over an entire run, 1300 measurements total, a statistical representation of the computer vision measurements noise could be obtained. The measurements noise was then associated with the velocity of the vehicle and pictorially represented in Figures 11 and 12.

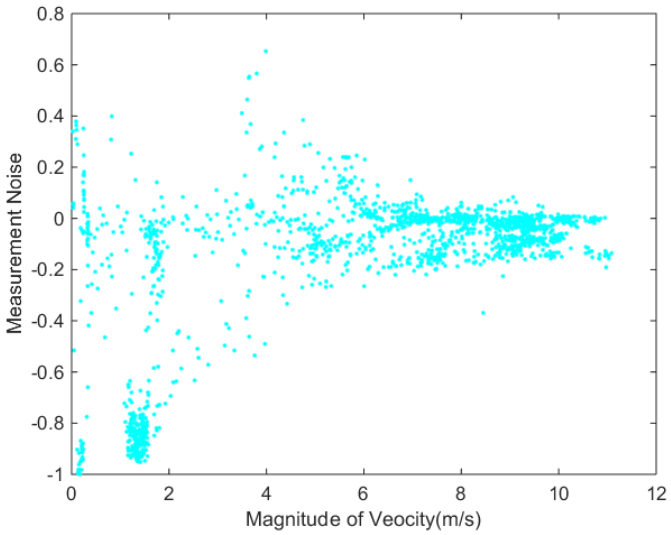

Figure 11. Unitless measurement noise obtained from the dot product of the 2-D zero velocity vector with the true velocity vector for the east direction of travel.

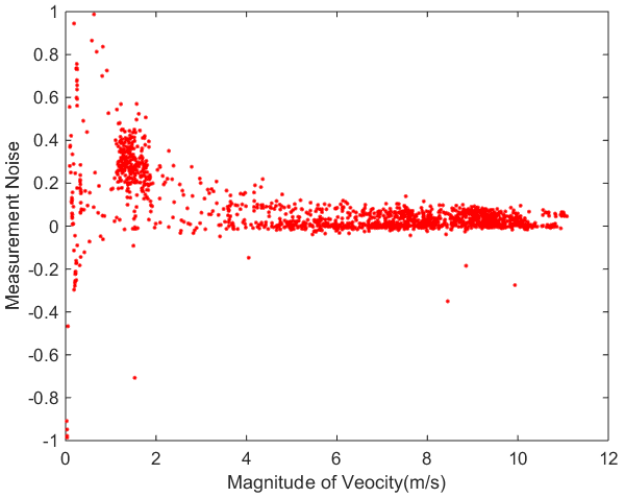

Figure 12. Unitless measurement noise obtained from the dot product of the 2-D zero velocity vector with the true velocity vector for the down direction of travel.

From the measurement noise values obtained, a decent approximation of the camera's measurement noise could be made. It was observed that the noise for the camera measurements were correlated to the actual speed of the vehicle. The faster the vehicle traveled, the less noise there was associated with the measurement. To take this into account, the varying sigma values, seen in Figure 13, were integrated in the system.

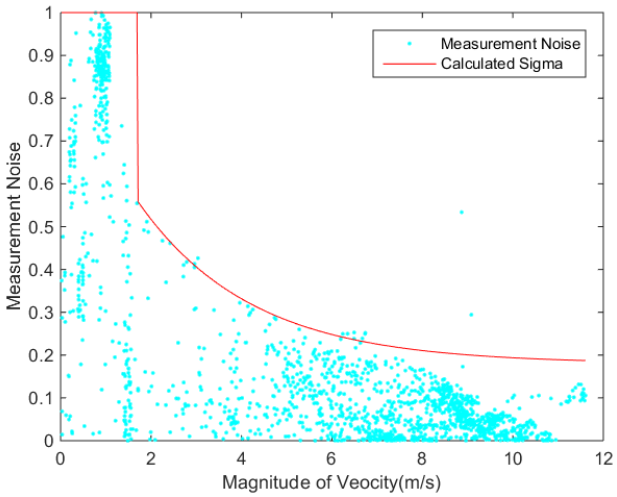

Figure 13. Calculated noise values based off the measured noise when taking vision sensor measurements with the true velocity measurements being supplied to the system.

Noise values were implemented as an exponential decay, with the higher values being seen at low velocities and lower values being seen at a higher velocities. Velocity of the system was determined by the magnitude of the INS's velocity in the north, east and down velocity. Finally, the covariance of the measurement was described as:

$$\mathbf{R}_{camera} = \begin{bmatrix} \sigma^2_{camera_{east}} & 0 \\ 0 & \sigma^2_{camera_{down}} \end{bmatrix} \tag{90}$$

where the $\sigma_{camera_{down}}$ and $\sigma_{camera_{east}}$ were determined by an exponential function that was driven by the velocity. The functions that were used for this were

$$\sigma_{camera_{east}} = 0.75e^{(-0.2\sqrt{(v_n^2+v_e^2+v_d^2)}+0.2)} \tag{91}$$

$$\sigma_{camera_{down}} = 0.9e^{(-0.6\sqrt{(v_n^2+v_e^2+v_d^2)}+0.1)} \tag{92}$$

When the magnitude of the velocity was below 1.8 (m/s) the σ_{camera} values for both the east and the down direction were set equal to 0.99.

3.5 GPS Measurement Update

In this thesis, pseudorange measurements from different satellites were used to estimate position. Satellites were removed to simulate a GPS degraded environment. This allowed the system to observe different outages throughout any specific run. The end result was a system that could have any number of satellites in view, as to not give a full GPS solution, and rely more heavily on the use of the image aided measurements for system error state propagation. In order to correct for position errors without having a full GPS solution, calculated pseudoranges from the users position were compared to actual pseudoranges given by the satellite [15]. The measurement model

for a single GPS measurements found in [11] can be seen here

$$\delta z(t_k) = H(t_k)\delta x(t_k) + v(t_k) \tag{93}$$

$$\delta z_\rho(t_k) = \rho_{INS}(t_k) - \rho_{GPS}(t_k) \tag{94}$$

where $\delta z_\rho(t_k)$ is the error in pseudorange at the time of measurement, ρ_{INS} is the estimated pseudorange from the inertial navigation INS, and ρ_{GPS} is the measured pseudorange from the satellite vehicle (SV) to the vehicle GPS receiver. The measured pseudorange [11], that is non-linear, can be found by

$$\rho_{GPS}(t_k) = \sqrt{(x(t_k) - x_{GPS}(t_k))^2 + (y(t_k) - y_{GPS}(t_k))^2 + (z(t_k) - z_{GPS}(t_k))^2} + \delta b_r \tag{95}$$

where x, y, and z is the true position of the vehicle's receiver in the ECEF frame, x_{GPS}, y_{GPS}, and z_{GPS} is the SV's position in the ECEF frame, and $\delta b_r = c\delta t_r$ is the range error in meters due to the receiver's clock bias and is equal to the speed of light, c, multiplied by the receiver's clock offset,t_r. The corrected position of the receiver is then expressed as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_{INS} - \delta x \\ y_{INS} - \delta y \\ z_{INS} - \delta z \end{bmatrix} \tag{96}$$

where x_{INS}, y_{INS}, and z_{INS} are the mechanization of the outputs and δx, δy, and δz are the estimated position error. After applying the Taylor series expansion around the mechanized output and linearizing Equation 95, the linearized measurement [11] can be given as

$$\rho_{INS} - \rho_{GPS} = \frac{(x_{INS} - x_{GPS})(x_{INS} - x) + (y_{INS} - y_{GPS})(y_{INS} - y) + (z_{INS} - z_{GPS})(z_{INS} - z)}{\sqrt{(x_{INS} - x_{GPS})^2 + (y_{INS} - y_{GPS})^2 + (z_{INS} - z_{GPS})^2}} - \delta b_r \tag{97}$$

A line of sight vector from the satellite to the receiver's position can then be defined as

$$l_{INS} = \begin{bmatrix} l_{x,INS} \\ l_{y,INS} \\ l_{z,INS} \end{bmatrix} = \begin{bmatrix} \frac{(x_{INS}-x_{GPS})}{\sqrt{(x_{INS}-x_{GPS})^2+(y_{INS}-y_{GPS})^2+(z_{INS}-z_{GPS})^2}} \\ \frac{(y_{INS}-y_{GPS})}{\sqrt{(x_{INS}-x_{GPS})^2+(y_{INS}-y_{GPS})^2+(z_{INS}-z_{GPS})^2}} \\ \frac{(z_{INS}-z_{GPS})}{\sqrt{(x_{INS}-x_{GPS})^2+(y_{INS}-y_{GPS})^2+(z_{INS}-z_{GPS})^2}} \end{bmatrix} \tag{98}$$

where l_{INS} is the line of sight vector. By solving for the estimated position error instead of the corrected position seen in Equation 96, you get

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} = \begin{bmatrix} x_{INS} - x \\ y_{INS} - y \\ z_{INS} - z \end{bmatrix} \tag{99}$$

The error values in Equation 99 and line of sight values in Equation98 can then be substituted into Equation 97 to get

$$\delta z_\rho = \begin{bmatrix} l_{x,INS} & l_{y,INS} & l_{z,INS} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} - \delta b_r \tag{100}$$

From [11], the error values in Equation 99 can be converted from ECEF geodetic coordinates to linearized ECEF rectangular coordinates and expressed as

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} = \begin{bmatrix} -(R_n+h)\sin\varphi\cos\lambda & -(R_n+h)\cos\varphi\sin\lambda & \cos\varphi\cos\lambda \\ -(R_n+h)\sin\varphi\sin\lambda & (R_n+h)\cos\varphi\cos\lambda & \cos\varphi\sin\lambda \\ (R_n(1-e^2)+h)\cos\varphi & 0 & \sin\varphi \end{bmatrix} \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix} \tag{101}$$

where $\delta\varphi$ is the error in latitude, $\delta\lambda$ is the error in longitude, and δh is the error in altitude. From Equation 97 and Equation 101 the observability matrix $H(t_k)$ can be

given as

$$H_\rho(t_k) = \begin{bmatrix} l_{x,INS} & l_{y,INS} & l_{z,INS} \end{bmatrix} \begin{bmatrix} -(R_n+h)\sin\varphi\cos\lambda & -(R_n+h)\cos\varphi\sin\lambda & \cos\varphi\cos\lambda \\ -(R_n+h)\sin\varphi\sin\lambda & (R_n+h)\cos\varphi\cos\lambda & \cos\varphi\sin\lambda \\ (R_n(1-e^2)+h)\cos\varphi & 0 & \sin\varphi \end{bmatrix} \tag{102}$$

Finally, the measurement equation for the system is given as

$$\delta z_\rho(t_k) = H_\rho(t_k) \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix} - \delta b_r(t_k) \tag{103}$$

The covariance of the measurement was defined as

$$R = \begin{bmatrix} \sigma_{GPS}^2 \end{bmatrix} \tag{104}$$

where σ_{GPS} is the standard deviation of the GPS measurements and was set to a constant value of 6 meters.

IV. Results

Results that were obtained for the simulations utilizing a monocular computer vision algorithm, GPS aided navigation, and monocular computer vision algorithm and GPS aided navigation will be discussed in this chapter. Information will be presented in a logical order showing the improvement of the navigation solution due to different measurement sources being available.

4.1 Data Sets

Data sets that were used for this thesis are presented in this section. A collection of points generated by the AFIT Autonomous Navigation and Technology (ANT) center during the All Source Positioning Navigation (ASPN) program were used as test points throughout this thesis. The test points generated during the ASPN program and used as the real data source in this research, were defined as scenarios, and will be referenced as scenarios in this work. There were a total of four different scenarios that were used to test the results of a monocular camera computer vision algorithm, IMU sensor, and GPS sensor paired in different configurations. While not all angles of the monocular camera view were available during the test, enough diversity in the camera view was given to make comparisons between separate runs that show the benefits and drawbacks of different monocular camera views.

Scenario 2.

Scenario 2's data points were generated by a vehicle that was driving around the campus of Ohio State University (OSU). A rural area is depicted in this scenario, meaning there were sparse buildings, natural foliage, and low vehicle and pedestrian congestion. GPS was not adversely affected for most of the scenario, other than the

periods of time that the vehicle was crossing or sitting under a bridge. Camera views available for Scenario 2, were a left and forward facing camera. The horizontal and vertical trajectory of the vehicle can be seen in Figures 14 and 15.

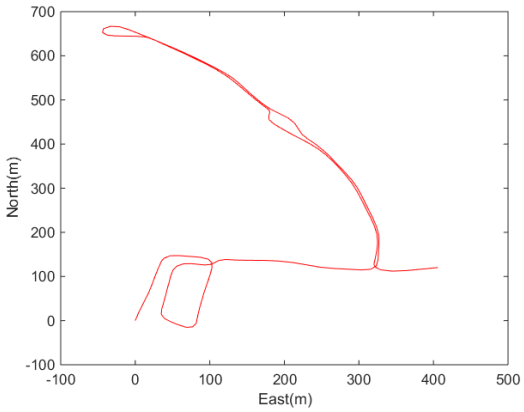

Figure 14. Scenario 2 true horizontal trajectory

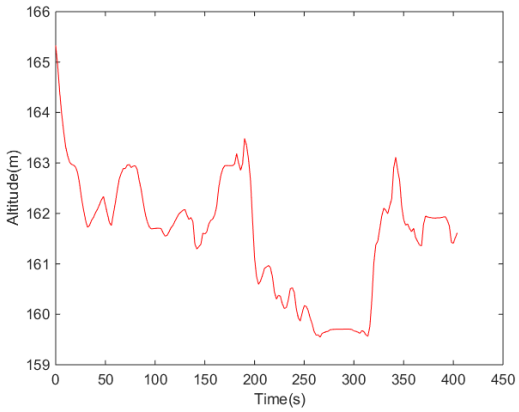

Figure 15. Scenario 2 true vertical trajectory

Scenario 6.

Scenario 6's data points were collected while a vehicle was driving circles in a parking lot. There were no buildings or natural foliage in the local vicinity of this scenario. Instead, the most obstructive points were generated from vehicles that were parked sparsely in the parking lot. GPS was available during the entire scenario, and no degradation of the system was exhibited. Camera views available for this scenario, were a front and right side facing camera. The horizontal and vertical trajectory of the vehicle can be seen in Figures 16 and 17.

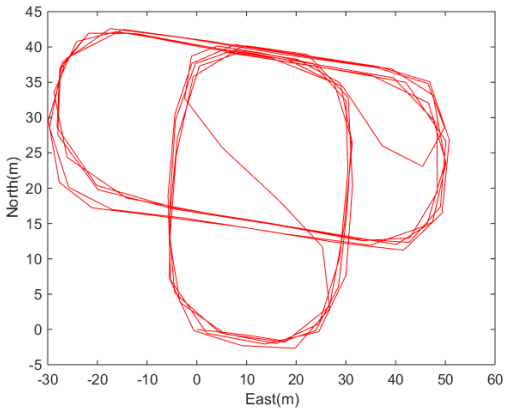

Figure 16. Scenario 6 true horizontal trajectory

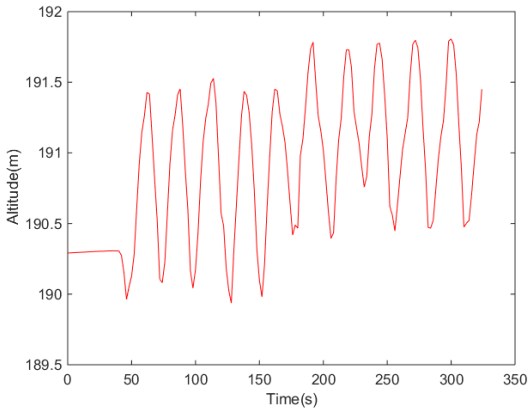

Figure 17. Scenario 6 true vertical trajectory

Scenario 7.

Scenario 7's data points were collected while driving around the downtown section of Columbus, OH. An urban environment is depicted in this scenario, meaning there were a considerable amount of large buildings, cross traffic, and pedestrians in close proximity of the vehicle. GPS was degraded for most of the run due to multi-pathing and signal outages caused by the buildings. Camera views that were available during this scenario, were a front, left side, and right side facing camera. The horizontal and vertical trajectory of the vehicle can be seen in Figures 18 and 19.

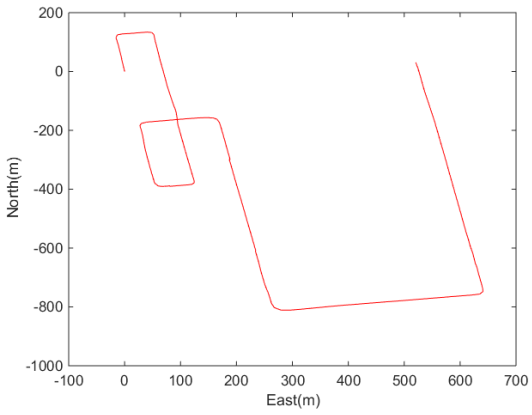

Figure 18. Scenario 7 true horizontal trajectory

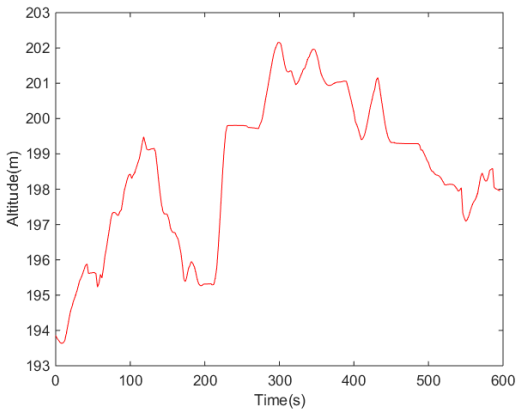

Figure 19. Scenario 7 true vertical trajectory

Scenario 17.

Scenario 17's environment was represented by the same type of environment depicted in Scenario 7. A left facing camera view was the only one available for this scenario. The horizontal and vertical trajectory of the vehicle can be seen in Figures 20 and 21.

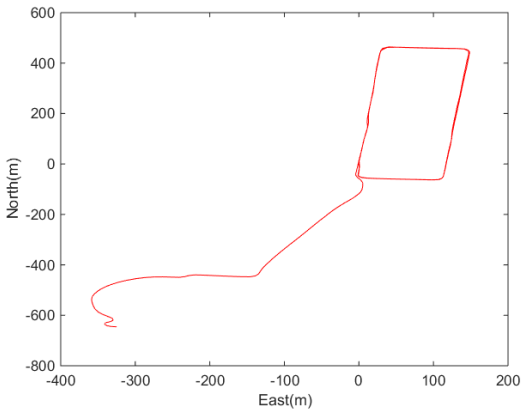

Figure 20. Scenario 17 true horizontal trajectory

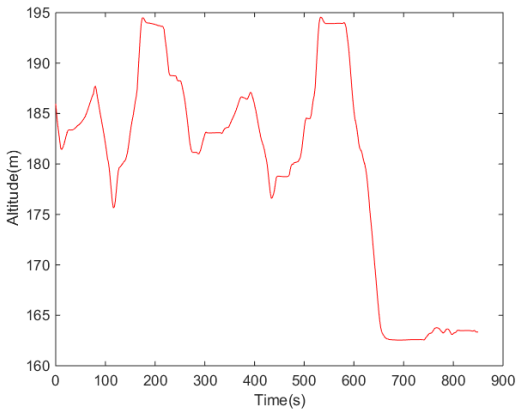

Figure 21. Scenario 17 true vertical trajectory

4.2 IMU Only Solution

Relevance of IMU aiding is developed in this section. While each scenario will not be depicted individually, an overall representation of how the system performs without aiding sensors will be developed here. For this section, Scenario 7's results of an unaided IMU will be visually represented. The actual trajectory of this scenario was annotated in Section 4.1. The horizontal and vertical trajectories estimated by the unaided IMU can be seen in Figures 22 and 23.

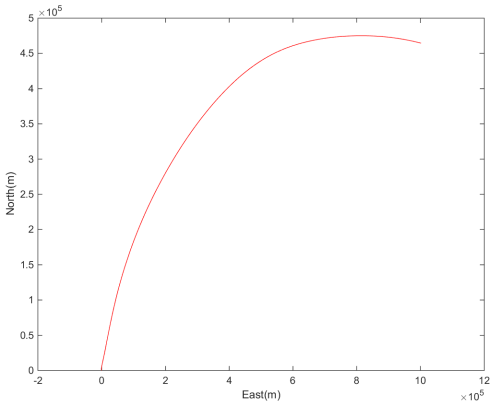

Figure 22. IMU's estimated horizontal trajectory for Scenario 7 with no aiding measurements.

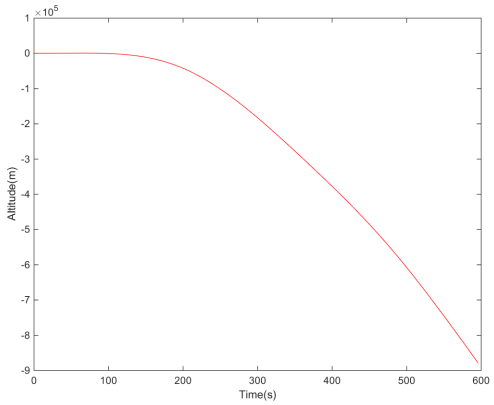

Figure 23. IMU's estimated vertical trajectory for Scenario 7 with no aiding measurements

As expected, the solution given by an unaided IMU is very poor. Measurements that are given by a commercial IMU tend to have a lot of bias added to them and can only perform well for very short periods of time if unaided. While the trajectories of the system do not provide much insight into what is happening, the position error and estimated standard deviation by the filter speaks volumes about the tuning of the system. This is important to note, because system tuning will play a role in every measurement that comes into the system. Values obtained for position error in Scenario 7 can be seen in Figure 24.

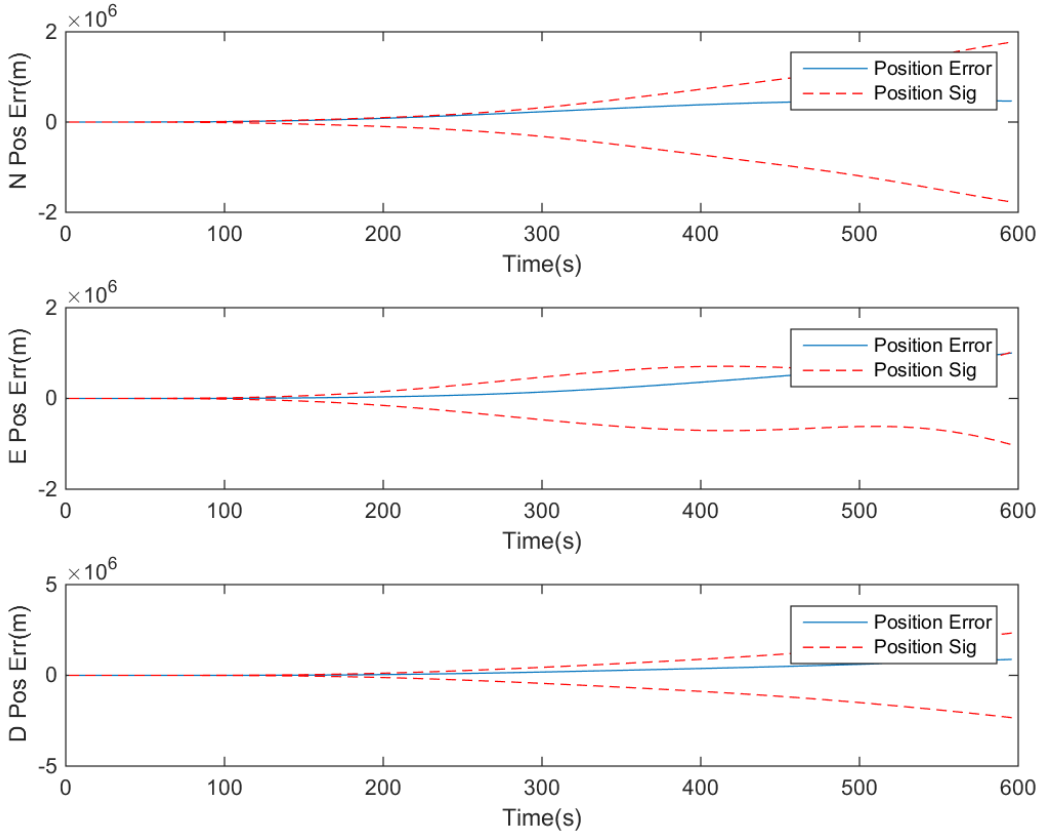

Figure 24. Scenario 7 IMU position error with filter estimated standard deviation with no aiding measurements.

While these figures are not enough to truly describe all the errors seen by the system, they do provide a visual representation of what is going on. Position errors seen in Figure 24 , for a simulated commercial grade IMU, match those that are expected from a commercial grade IMU highlighted in [5] and [1]. To give numerical values that can be compared between system configurations, the approach developed in [15] will be used, in which the distance root mean square (DRMS) error will be used to describe overall performance of the system. The mean DRMS error is defined by:

$$DRMS = \sqrt{\frac{\Sigma_{i=1}^{n}(x_i^2 + y_i^2)}{n}} \tag{105}$$

where n is the number of measurement in the epochs, x_i is the north error for the i^{th} epoch , and x_i is the east error for the i^{th} epoch. Because there were variances in the distances traveled, errors produced by each simulated IMU, and the lengths of each run, a separate DRMS error value had to be determined for each scenario. The DRMS values that were obtained for the unaided IMU scenarios can be seen in Table 2

Table 2. DRMS values for a single run with the an IMU that is not being aided

Scenario	DRMS(m)
Scenario 2 Commercial IMU	55165
Scenario 2 Simulated IMU	170934
Scenario 6 Simulated IMU	68066
Scanario 7 Simulate IMU	443094
Scenario 17 Simulated IMU	425145

Table 2 shows that the simulated and actual commercial grade IMU errors do not match exactly. This does not raise any concern though. Solutions provided by an IMU can perform better or worse than expected at any time, this particular simulation just happened to have a worse performance than that of the real IMU. Another thing to note is the parameters values, that were given by the vendor, for the real IMU used in Scenario 2. The parameters values for the real IMU were lower than those found in Table 1, meaning the IMU should perform a little better than what the simulated IMU did. The other scenarios show better and worse performance due to the duration of the run. The IMU is expected to get exponentially worse as the system operates without aiding. DRMS values will be given for each sensor's pairing and used to compare the system's performance for all the combination of sensors that are run in this thesis.

4.3 GPS Aided

This section will show the results of the system's performance when aided with different grades of GPS solution. Grades of GPS solution will be determined by the number of SVs in sight during the test run. The GPS will be restricted to a specific amount of satellites that is dictated by the grade of GPS required and the number of satellites that are actually available in the scenario. Sigma values given for the error in SV measurements have been set as 6 meters for open rural areas of travel, Scenarios 2 and 6, and have been increased to 7 meters for city environments, Scenarios 7 and 17, to account for multi path errors that can be induced into the system due to the vehicles surroundings.

Performance of the system operating with one SV will now be evaluated. To depict what is going on with the system, the horizontal trajectory of the scenario is given in Figure 25.

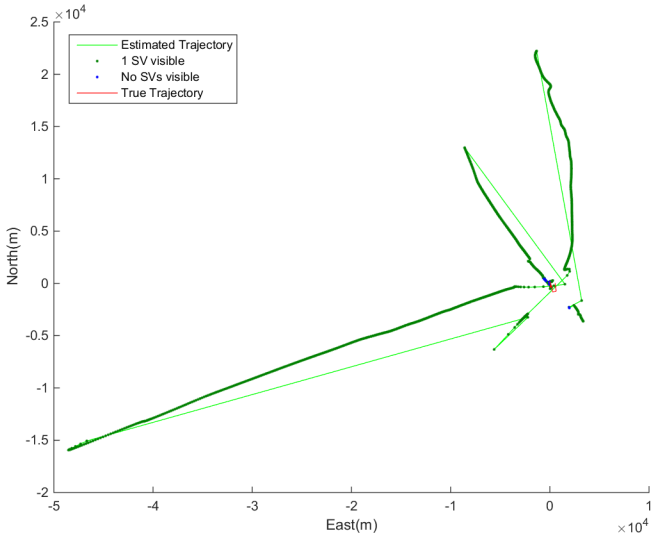

Figure 25. Scenario 7 horizontal trajectory while being aided by 1 SV.

The number of SVs seen during the duration of the run, and the particular SV used for each measurement will be shown in Figures 26 and 27.

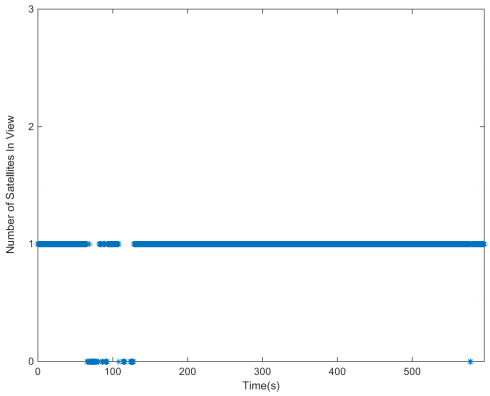

Figure 26. Scenario 7 number of SVs used for each measurement.

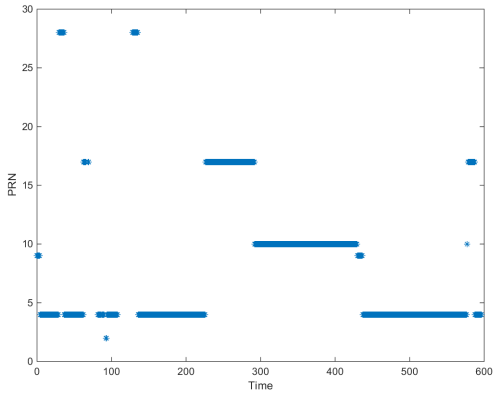

Figure 27. Scenario 7 SV being used to generate measurement.

As expected, the performance of the system is poor due to the low observability of just one SV. There is a little added benefit to the system during the transition from one SV to the next. This added benefit can be seen when the position is scoped back to the actual vehicle's trajectory seen in Figure 25, and it is directly correlated to the times that the system transition from one SV to another seen in Figure 27. Being that the data is real and satellites come and go, it was not feasible to produce a real system that received measurements from one specific SV for the entire scenario.

The actual DRMS values for all the scenario operating with one SV can be seen in Table 3.

Table 3. DRMS values for IMU being aided with 1 SV

Scenario	DRMS(m)
Scenario 2 Commercial IMU	1516
Scenario 2 Simulated IMU	3231
Scenario 6 Simulated IMU	13324
Scenario 7 Simulate IMU	14655
Scenario 17 Simulated IMU	1283619

The systems performed differently due to the number of SV transitions that were encountered and the duration of the run. Scenario 7 and 17 performed the worst because of their consistency of a specific SV being viewable for the duration of the scenario. When looking at Scenario 17 specifically, the system had a degradation in system performance when aided with a single SV, which is due to the consistency of a single SV being viewable for prolonged periods of time. From Figure 28, it is easy to see that Scenario 17 gets very little benefits from SVs coming in and out of view.

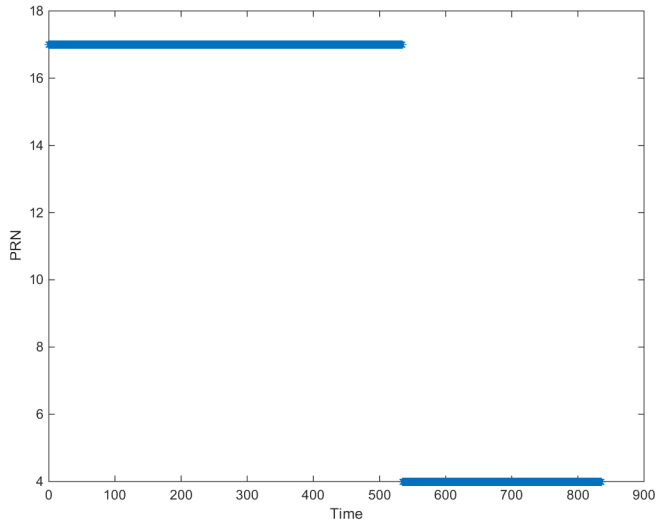

Figure 28. Scenario 17 SV being used to generate measurement vs time.

System performance for an IMU being aided with 2 SVs will now be evaluated. The vehicle's horizontal trajectory is shown in Figure 29, and the number of SVs in view at the time of measurement along with the specific SVs used for the measurements can be seen in Figures 30 and 31.

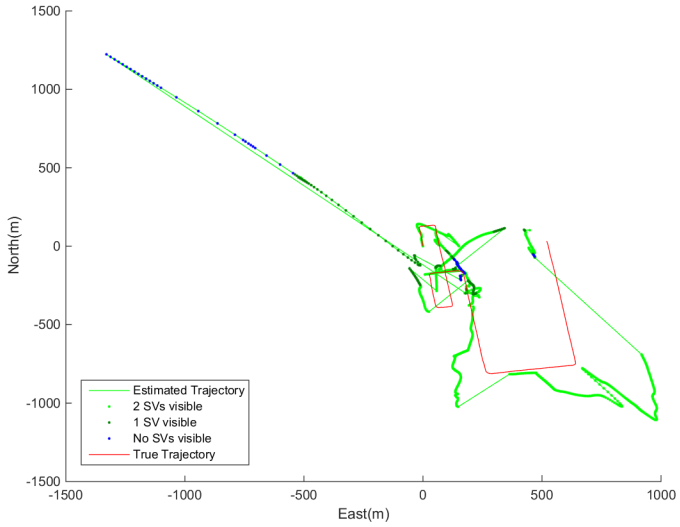

Figure 29. Scenario 7 horizontal trajectory while being aided by 2 SVs.

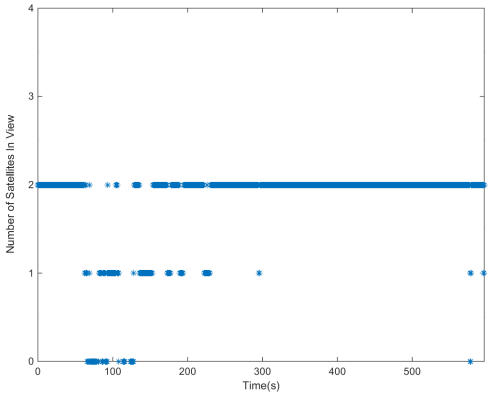

Figure 30. Scenario 7 number of SVs used for each measurement.

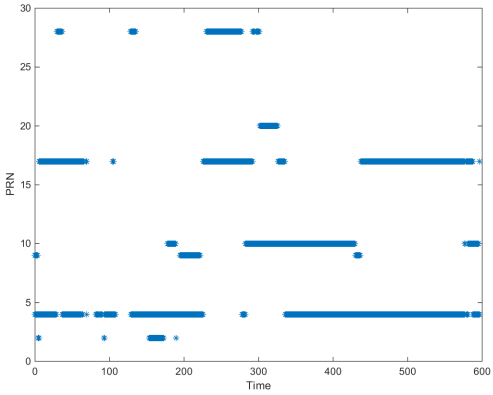

Figure 31. Scenario 7 SV being used to generate measurement vs time.

The same natural GPS outages seen in Figure 26 can be seen in Figure 30, which shows that the system is not arbitrarily negating specific measurements. The horizontal trajectory of the system also shows a drastic improvement compared to the trajectory seen when the system was operating with only one SV in view. This increase in performance is due to both the number of SVs being viewable at a specific time and the number of transitions between SVs due to natural outages.

How the system performed numerically is seen in Table 4.

Table 4. DRMS values for IMU being aided with 2 SVs

Scenario	DRMS(m)
Scenario 2 Commercial IMU	2376
Scenario 2 Simulated IMU	1124
Scenario 6 Simulated IMU	151
Scenario 7 Simulate IMU	225.8
Scenario 17 Simulated IMU	164010

All of the systems see an increase in performance when being aided with 2 SVs instead of 1, except for Scenario 2 operating with the real commercial grade IMU. It was noted, during the early stages of testing, that this IMU was actually performing worse than what the system was expected to do. This does not say anything about the IMU though, because the expected errors for the system are generated over numerous runs and the actual system performance can be better or worse than what is expected at any given time. For this case, the system is performing worse than what is expected, but is giving feedback to the filter as if its measurements were more accurate than what is being produced.

System performance will now be evaluated when 3 SVs are used to aid the IMU. The trajectory of the system along with the number of SVs in view during each measurement can be seen in Figures 32, 33, and 34.

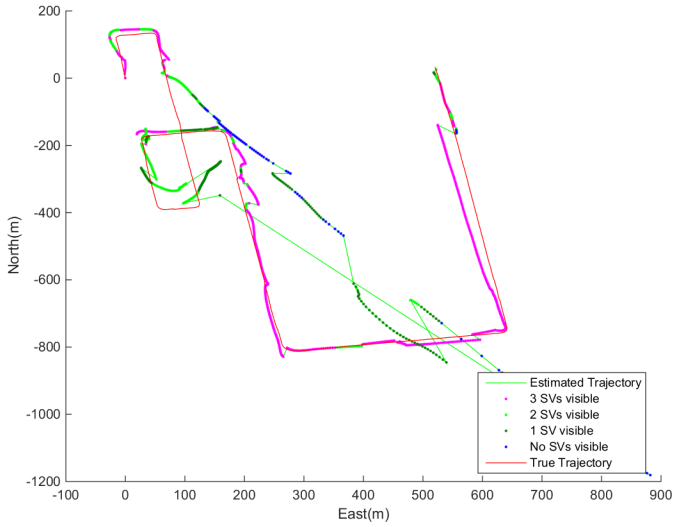

Figure 32. Scenario 7 horizontal trajectory of IMU being aided by 3 SVs.

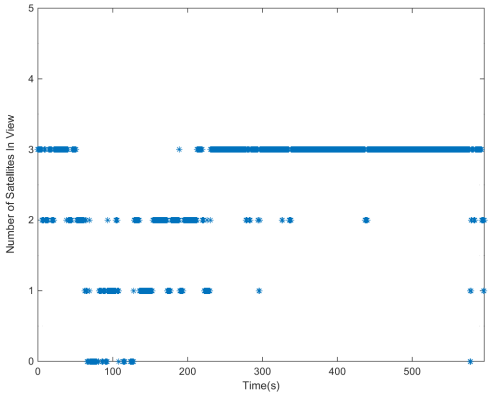

Figure 33. Scenario 7 number of SVs viewable vs time.

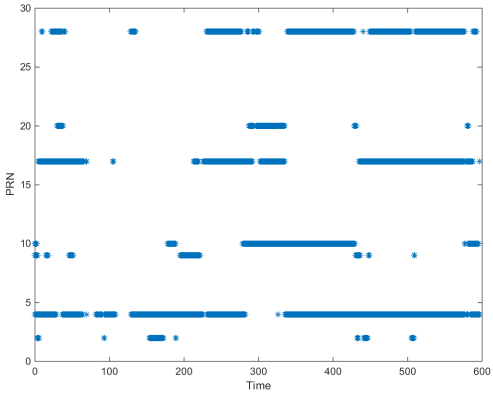

Figure 34. Scenario 7 SVs being used vs time.

The system performs very well when only 3 SVs are available. Normally, 3 satellites would not be expected to generate a good solution, due to unobservability of the four states that are being estimated. (For a simulation example, see [15].) In this case, however, the satellites came and went frequently due to blockages, so the actual satellites being tracked included more than 3 satellites, even though only 3 would be tracked in any one measurement epoch.

Numerical results for the system operating with the 3 SVs can be seen in Table 5.

Table 5. DRMS values for IMU being aided with 3 SVs

Scenario	DRMS(m)
Scenario 2 Commercial IMU	14.7
Scenario 2 Simulated IMU	10.4
Scenario 6 Simulated IMU	6.7
Scenario 7 Simulate IMU	14.7
Scenario 17 Simulated IMU	11.59

The total errors for the system are drastically reduced when a third satellite is available to the system. Even in the urban environment seen in Scenarios 7 and 17 large decrease in the errors are observed. This is due to a low number of outages caused by the surrounding buildings, which gave way for a pretty clear resolution of the position given by the GPS. Results for an IMU being aided with a full GPS solution will now be evaluated.The horizontal trajectory of the system is given in Figure 35, and the number of SVs available vs time along with the specific SVs being used are given in Figures 36 and 37.

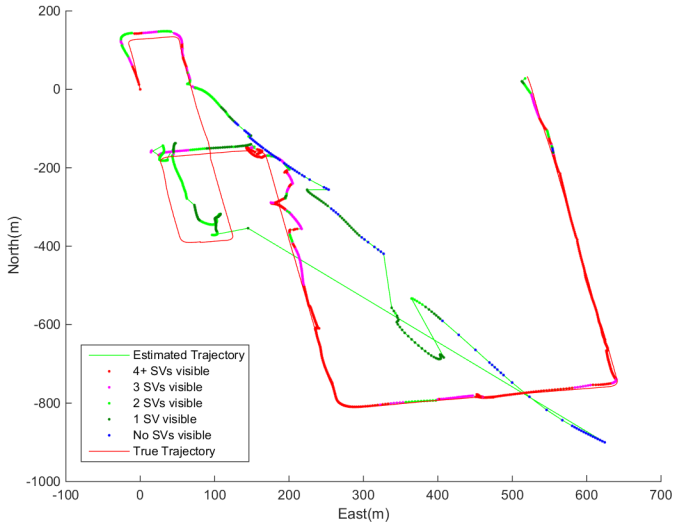

Figure 35. Scenario 7 horizontal trajectory of IMU being aided by all available SVs.

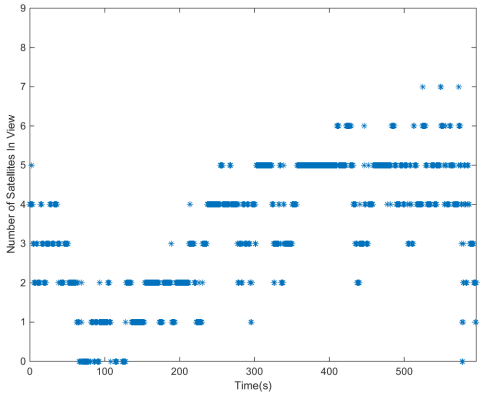

Figure 36. Scenario 7 number of SVs viewable vs time.

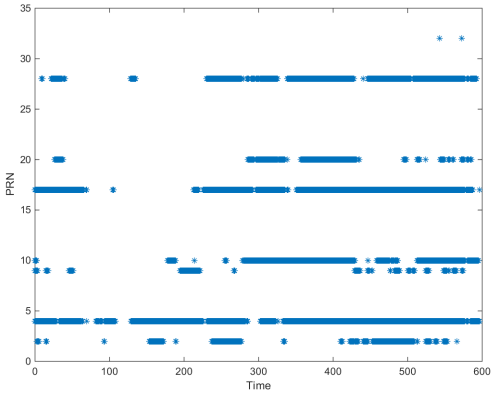

Figure 37. Scenario 7 SVs being used vs time.

As expected, the system performs very well when all the available satellites are used. During times of GPS degradation, the performance of the system decreases, but then corrects itself when more satellites become available. By evaluating the velocity and position errors with their respective standard deviation, a full understanding of how the system performed can be gained. These errors can be seen in Figures 38 and 39.

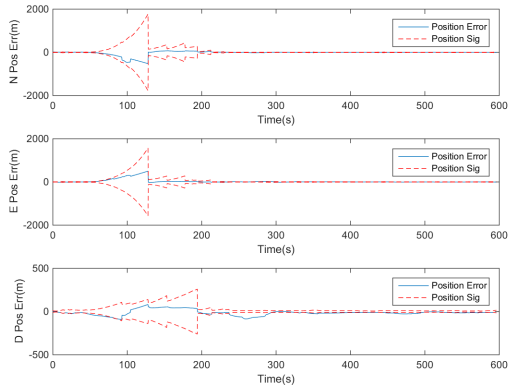

Figure 38. Scenario 7 IMU aided by all available SVs associated error in position with standard deviation.

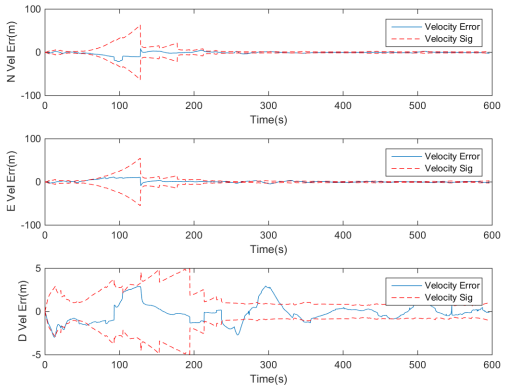

Figure 39. Scenario 7 IMU aided by all available SVs associated error in velocity with standard deviation

Error in the system position and velocity is minimal during the time of a full GPS solution, and the only time the system lacks in performance is during the GPS outages. As the number of viewable SVs decrease, the error in the system steadily grows. The growth in the position and velocity error correspond directly with the decrease of viewable SVs seen in Figure 36. Overall results for the system, when aided with different GPS solutions, can be seen in Table 6

Table 6. DRMS value comparison for the IMU being aided with different GPS solutions

Scenario	0 SV DRMS	1 SV DRMS	2 SV DRMS	3 SV DRMS	Available SV DRMS
Scenario 2 Com IMU	55165	1516	2376	14.7	9.9
Scenario 2 Sim IMU	170934	3231	1124	10.4	4
Scenario 6 Sim IMU	68066	13324	151	6.7	2.5
Scenario 7 Sim IMU	443094	14655	225.8	14.7	7.7
Scenario 17 Sim IMU	425145	1283619	164010	11.59	4.5

From the values given in Table 6, it is easy to see that each additional SV seen by the system reduces the overall DRMS error of the system. The two scenarios that showed a decrease in performance were affected by either the IMU or the presence of a single SV for a prolonged period of time.

4.4 Vision Aided

This section will explain the results obtained when aiding the IMU with the monocular camera computer vision algorithm. To stay consistent with what has already been presented, this section will display the figures of the results that were obtained during Scenario 7's test, but numerical DRMS results for all the scenarios will be given. Results in this section will compare the performance of the system as different monocular camera views are used to aid the IMU. Monocular camera views that will be presented in this section are going to be the forward facing, left side facing, and right side facing views.

Forward Facing Camera.

This system utilizes a camera that points in the vehicle's forward direction of movement. The horizontal trajectory seen in Figure 40 and the heading seen in Figure 41 illustrate the performance of the system when an IMU is aided with a forward facing vision sensor.

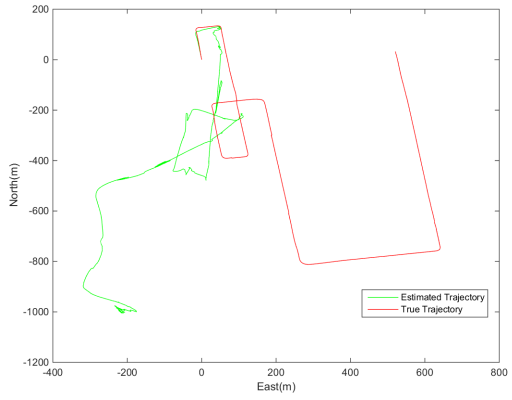

Figure 40. Scenario 7 estimated vs true horizontal trajectory with a forward facing monocular camera aiding an IMU.

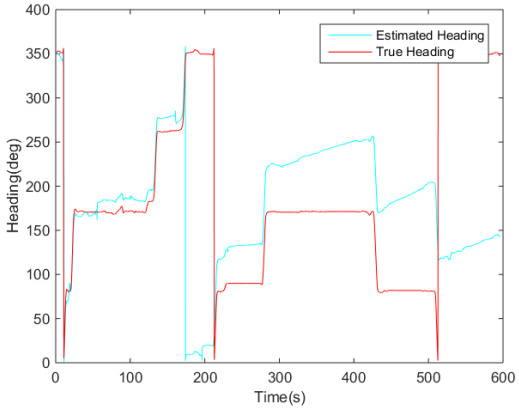

Figure 41. Scenario 7 estimated vs true heading with a forward facing monocular camera aiding an IMU.

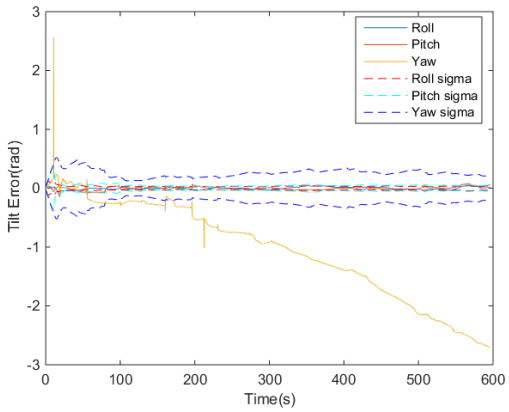

Figure 42. Scenario 7 tilt error vs time for a forward facing monocular camera aiding an IMU.

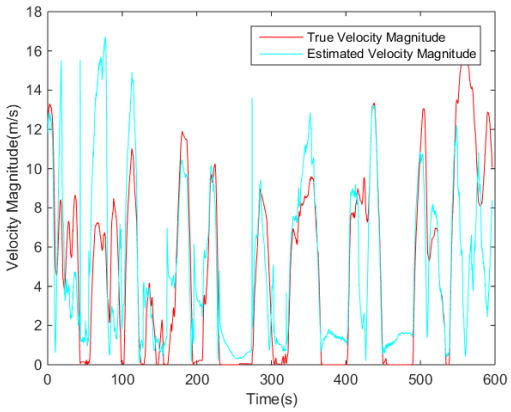

Figure 43. Scenario 7 true and estimated magnitude of velocity vs time for forward facing monocular camera aiding an IMU.

It is clearly seen that the system understands when the vehicle is observing a change in heading. From Figure 42, the tilt error that is seen in the system can be observed. As the vehicle continues to travel, the yaw tilt error of the system continues to grow, which contributes to the difference in the trajectory and heading observed. While the system seems to do really well observing changes in direction, the actual magnitude of movement, seen in Figure 43, is less observable. The inability of the system to sense the actual amount of movement is expected though. The monocular camera system is unable to detect the range to a feature that is detected, which leads to the lack of observability of the vehicle's actual velocity. The issue with the velocity was further exacerbated by the filter using previously recorded velocity values, from periods of time that occurred 1/8 of a second before the second image is actually being taken, as inputs into the monocular computer vision algorithm. This was due to the imagery sensor taking a measurement over a quarter of a second, and not being directly related to the velocity measurement that was given when the second image was taken. This will undoubtedly create errors that will steadily propagate and grow as the vehicle continues to move. Numerical results for all of the scenarios that had a forward facing camera can be seen in Table 7

Table 7. DRMS Values for a forward facing monocular camera aiding an IMU

Scenario	Forward DRMS(m)
Scenario 2 Commercial IMU	125.4
Scenario 2 Simulated IMU	289.2
Scenario 6 Simulated IMU	57.6
Scenario 7 Simulated IMU	509.4

Side Facing Camera.

This section will show the results for both the left and right side facing vision sensor aiding the IMU. These two side facing sensors were put together because their results were similar. The horizontal trajectories that were generated by both the right and left side facing cameras can be seen in Figures 44 and 45.

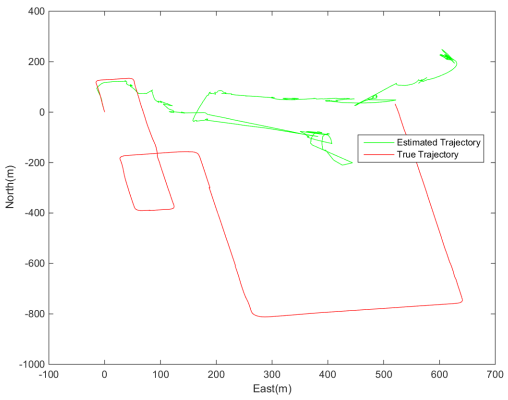

Figure 44. Scenario 7 horizontal trajectory with a left facing monocular camera aiding an IMU.

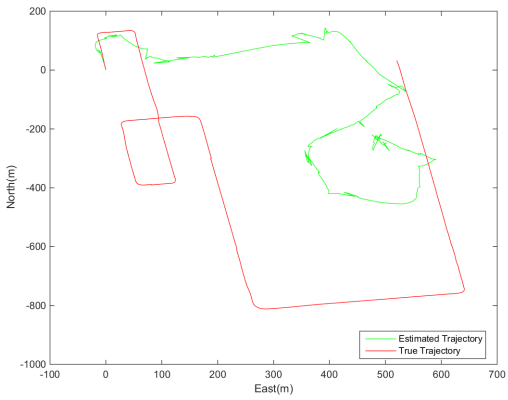

Figure 45. Scenario 7 horizontal trajectory with a right facing monocular camera aiding an IMU.

While the forward facing monocular camera does a good job of picking up variations in the vehicle's yaw, the side facing monocular camera can not give the same results. Side facing monocular cameras do not see features radiating outward from the center of the camera view like the forward facing monocular camera does. Instead,

side facing monocular cameras see the features moving in a horizontal trajectory across the image plain. This phenomenon can be seen in Figures 46 and 47, where the side facing monocular camera has horizontal lines connecting all the features, and the forward facing monocular camera has lines that start radiating vertically to their matched features. This is an issue since all horizontal movement leads to complications when the vehicle encounters a turn. While a forward facing monocular camera would start seeing a majority of the features moving horizontally across the plain, the side facing monocular camera sees the features acting in the same way that it has for the entire scenario. However, the features represented in the side facing monocular camera will give a better insight into the change in pitch that has been encountered by the vehicle.

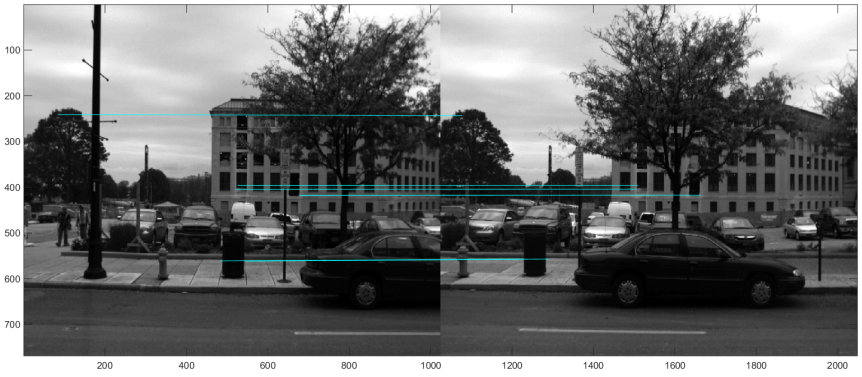

Figure 46. Scenario 7 side facing monocular camera feature observation in two sequential photos.

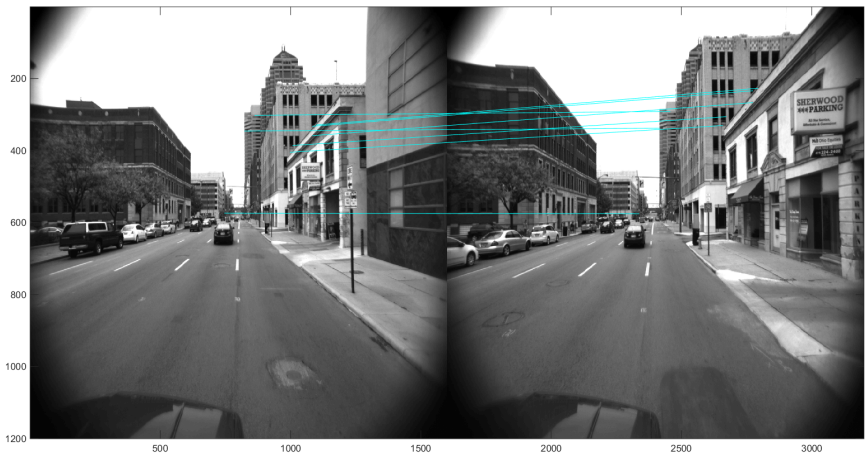

Figure 47. Scenario 7 forward facing monocular camera feature observation in two sequential photos.

The two side facing monocular cameras were able to detect a similar number of features in their respective field of vision. A comparison of the number of features detected by each camera at different velocities can be seen in Figures 48 and 49.

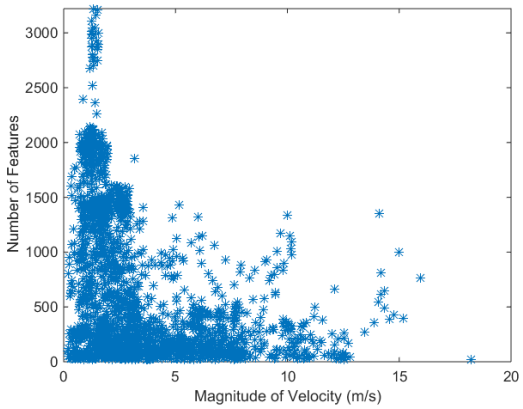

Figure 48. Scenario 7 features in view of left facing monocular camera vs velocity of the vehicle.

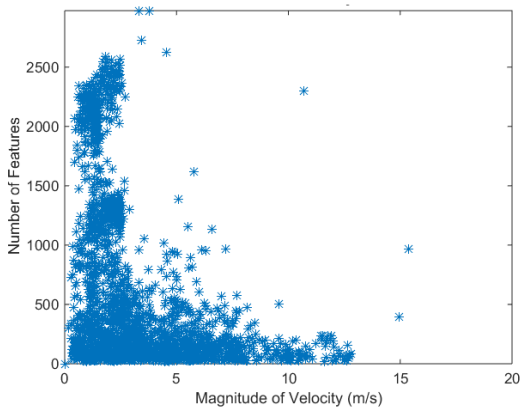

Figure 49. Scenario 7 features in view of right facing monocular camera vs velocity of the vehicle.

Similar feature detection between the two cameras may not be encountered if vehicle speeds were to increase during the scenario runs. Vehicle speeds hardly went above 15 m/s due to the vehicle operating in a city environment. If speeds were to

increase, and the proximity of the features in view (i.e., buildings) did not change, the system would probably not do as well because features in two sequential images would move in and out of image view too quickly for the camera to capture it. There were a few instances when the camera picked up a large amount of features at higher velocities, but this is only due to the fact that a clearing of the buildings normally in view was encountered (i.e., the vehicle drove through an intersection and the depth of the image drastically increased).

Errors in the position, velocity, and tilt encountered by the system can be seen in Figures 50, 51, and 52.

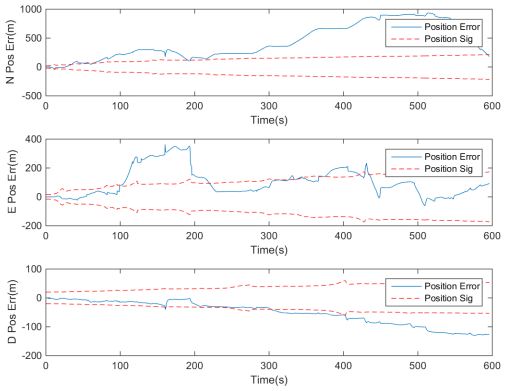

Figure 50. Scenario 7 error in position for the left facing monocular camera aiding the IMU.

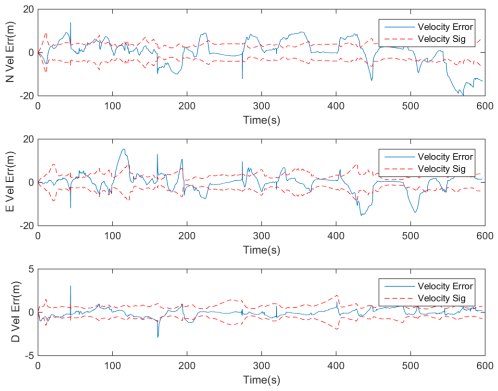

Figure 51. Scenario 7 error in velocity for the left facing monocular camera aiding the IMU.

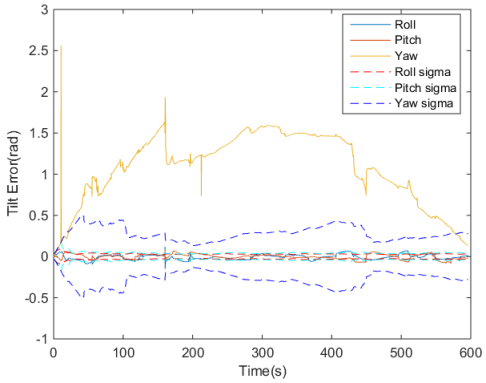

Figure 52. Scenario 7 tilt error for the left facing monocular camera aiding the IMU.

Error for position and the velocity of the system stayed within the thresholds of what the predicted error for the system was thought to be. The values that fall outside of the error in the north position are related to vehicle's direction of travel. As mentioned before, the vision sensor has no perception of total movement encountered in the forward direction, and can only really constrain the lateral movement of the vehicle. So, anytime the vehicle moves directly forward in a specific direction, north or east, the error in that direction is going to grow considerably more than what is normally expected. Tilt error of the system is considerable, as expected for a side facing camera. The ability of the side facing camera to detect a turn is minimal, due to the fact that all the features are moving in the same manner as they have always been seen moving.

Overall, the two side facing cameras did produce a noticeable decrease in position and velocity error over a commercial IMU operating on its own, and there was not noticeable difference from DRMS error values produced by the forward facing camera. Performance of the different facing vision sensors can be seen in Table 8.

Table 8. DRMS values for the different monocular cameras aiding the IMU.

Scenario	L Side DRMS	R Side DRMS	Forward DRMS
Scenario 2 Com IMU	184.0	N/A	125.4
Scenario 2 Sim IMU	173.9	N/A	289.2
Scenario 6 Sim IMU	N/A	115.7	57.6
Scenario 7 Sim IMU	540.4	362.7	509.4
Scenario 17 Sim IMU	567.9	N/A	N/A

4.5 IMU aided with Vision and GPS

This section will show the results that were obtained when aiding the IMU with a monocular computer vision algorithm and GPS simultaneously. While the poses of the monocular cameras for the scenario will not be changed, the number of SVs aiding the system will change. This section will further highlight the importance and usability of aiding an IMU with a monocular computer vision algorithm when GPS is degraded.

The trajectory of the system when a single SV paired with the monocular computer vision algorithm is used to aid the IMU can be seen in Figure 53.

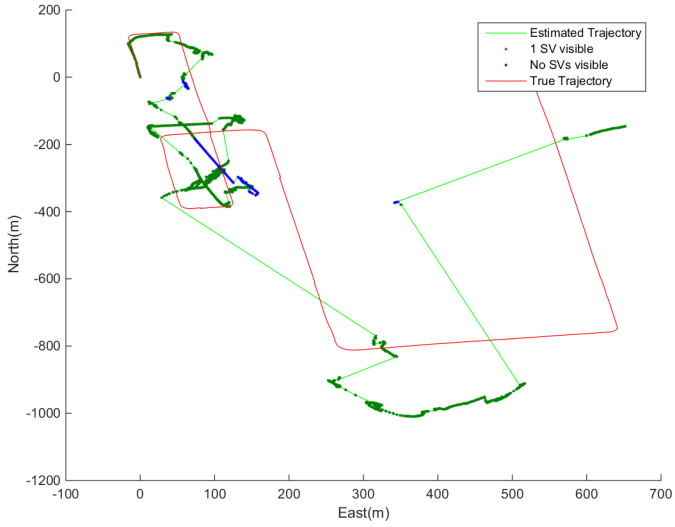

Figure 53. Scenario 7 single SV paired with the monocular camera computer vision algorithm to aid the IMU.

While a single SV has problems estimating the trajectory because of bad information provided by the IMU, the addition of the camera provides a relative position reference to the system. This makes updates from the SV more accurate and provides for a better solution. The position, velocity, and tilt errors given by the system can be seen in Figures 54, 55, and 56.

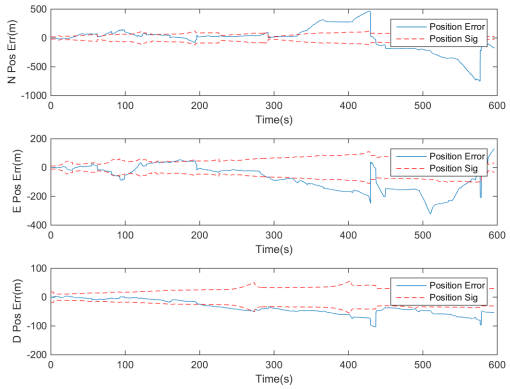

Figure 54. Scenario 7 error in position with 1SV and the monocular camera computer vision algorithm aiding the IMU.

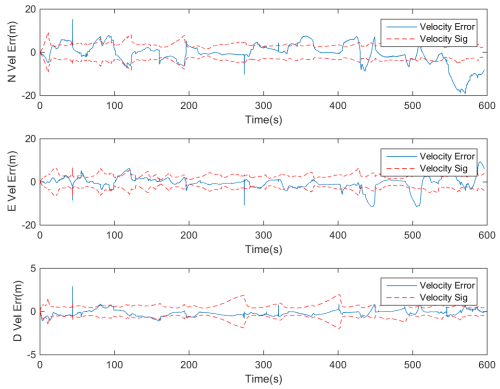

Figure 55. Scenario 7 error in velocity with 1SV and the monocular camera computer vision algorithm aiding the IMU.

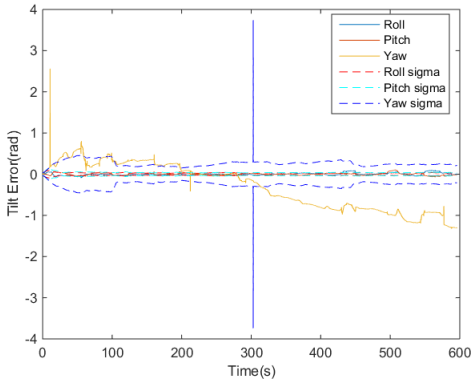

Figure 56. Scenario 7 tilt error with 1SV and the monocular camera computer vision algorithm aiding the IMU.

While the system is still not optimal, the decrease in position, velocity, and tilt errors can be observed compared to the two system operating independently. Heading tilt errors that were observed before with the vision sensor acting independently have been reduced due to the good position updates that are occasionally received by the GPS. Results for all the scenarios that have a single SV paired with the monocular computer vision algorithm to aid the IMU can be seen in Table 9.

Table 9. DRMS values for a single SV paired with the monocular camera computer vision algorithm to aid the IMU

Scenario	L Side DRMS(m)	R Side DRMS(m)	Forward DRMS(m)
Scenario 2 Com IMU	140.3	N/A	111.3
Scenario 2 Sim IMU	80	N/A	88.5
Scenario 6 Sim IMU	N/A	43.4	30.9
Scenario 7 Sim IMU	247.5	248.3	213.1
Scenario 17 Sim IMU	180.1	N/A	N/A

The number of SVs seen by the system will now be increased to two and will continue to be paired with themonocular computer vision algorithm to aid the IMU. The horizontal trajectory for the paired sensors aiding the IMU in this case can be seen in Figure 57.

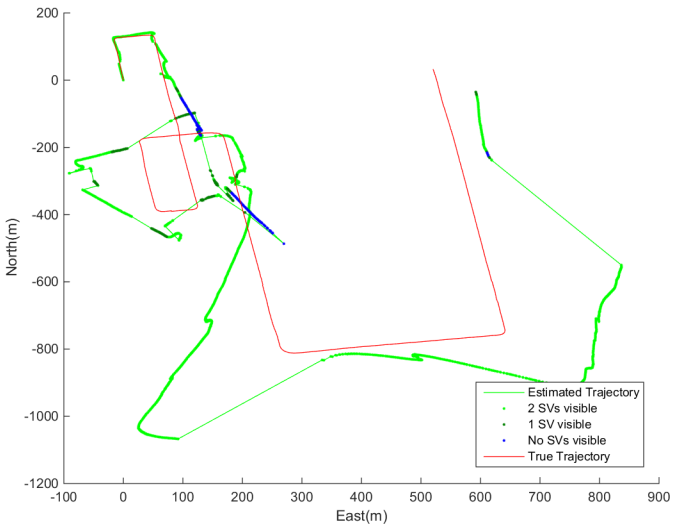

Figure 57. Scenario 7 two SVs paired with the monocular camera computer vision algorithm to aid the IMU.

By increasing the number of SVs to two, the performance of the system, compared to monocular camera computer vision algorithm only and GPS only aiding, gradually

increases. While there are still periods of natural GPS outages encountered, the monocular camera computer vision algorithm now keeps the system's position and velocity errors from growing like it would if it had no additional aiding. Each of the other scenarios performance results, with two SVs and vision, can be seen in Table 10.

Table 10. DRMS values for two SVs paired with the monocular camera computer vision algorithm to aid the IMU

Scenario	L Side DRMS(m)	R Side DRMS(m)	Forward DRMS(m)
Scenario 2 Com IMU	52.2	N/A	29.6
Scenario 2 Sim IMU	54.4	N/A	71.3
Scenario 6 Sim IMU	N/A	25.6	16.8
Scenario 7 Sim IMU	91.8	152.4	151.5
Scenario 17 Sim IMU	193	N/A	N/A

The number of SVs viewable by the system will now be increased to 3, while still being aided with vision. The actual trajectory for the system can be seen in Figure 58.

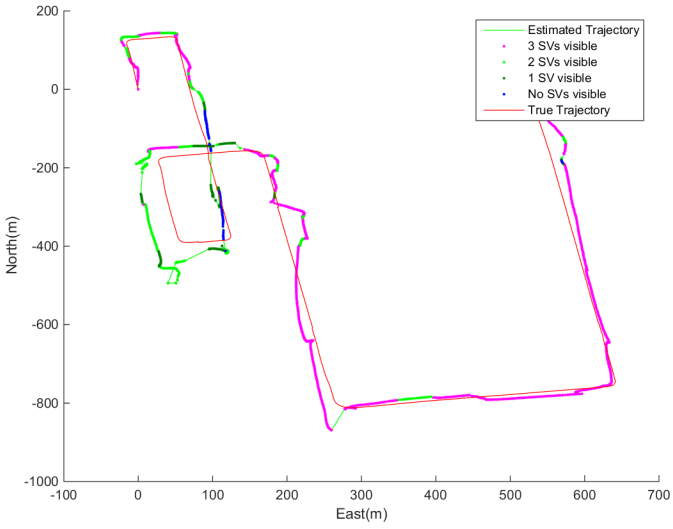

Figure 58. Scenario 7 3 SVs paired with the monocular camera computer vision algorithm to aid the IMU.

At this point, the system shows a trajectory that could almost match that which would be expected with a full GPS solution. The system does a better job of tracking the actual horizontal trajectory even when the GPS is degraded to less than 3 available SVs available. The monocular camera computer vision algorithm is able to provide a better solution during times of GPS degradation due to the better initialization points that have been created by the GPS. This drastically reduces the tilt errors that would slowly degrade the monocular camera computer vision algorithm over time. Each of the other scenarios performance results, with two SVs and the monocular computer vision algorithm, can be seen in Table 11.

Table 11. DRMS values for three SVs paired with the monocular camera computer vision algorithm to aid the IMU

Scenario	L Side DRMS(m)	R Side DRMS(m)	Forward DRMS(m)
Scenario 2 Com IMU	15.5	N/A	14.5
Scenario 2 Sim IMU	7.9	N/A	7.4
Scenario 6 Sim IMU	N/A	13.7	5.8
Scenario 7 Sim IMU	14.5	11.2	24.2
Scenario 17 Sim IMU	10.5	N/A	N/A

From the results, it seems as if the side facing monocular cameras give better results than the forward facing monocular camera in the urban environments, and forward facing monocular cameras give better results in rural open areas. This is due to the features that are seen by the camera during the scenarios. Side facing cameras may have a better recognition of features in a urban environment than they do in a rural environment because the object are man made and have a lot more distinct characteristics than what natural foliage may have. The traffic patterns seen in a city may also disrupt the forward facing camera by having features that are detected

being associated with a car that is turning. These vehicle would then tell the vision sensor that it is in a turn when it is actually going straight.

Finally, a full GPS solution is paired with the monocular camera computer vision algorithm to aid the IMU. The actual trajectory for the system can be seen in Figure 59

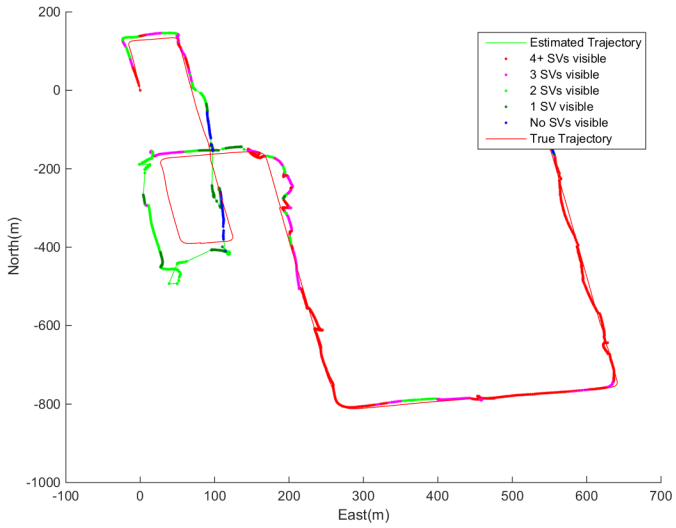

Figure 59. Scenario 7 four or more SVs paired with the monocular camera computer vision algorithm to aid the IMU.

Again, unavoidable GPS outages to the system are experienced, but the monocular computer vision algorithm helps drastically reduce the overall degradation of the system. The errors in the horizontal trajectory that were seen in Figure 35 are now reduced, and the vehicle's estimated trajectory is now closer to what the actual trajectory was. The errors for both the position and velocity can be seen in Figures 60 and Figure 61.

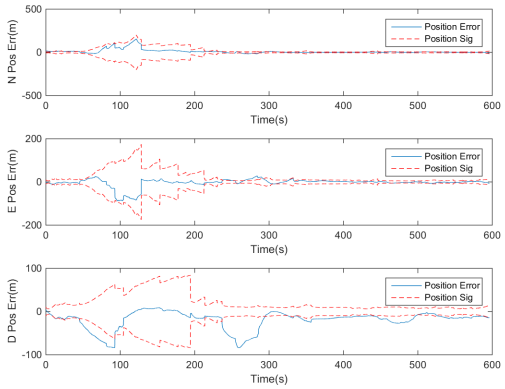

Figure 60. Scenario 7 with all available SVs paired with the monocular camera computer vision algorithm to aid the IMU position error with standard deviation.

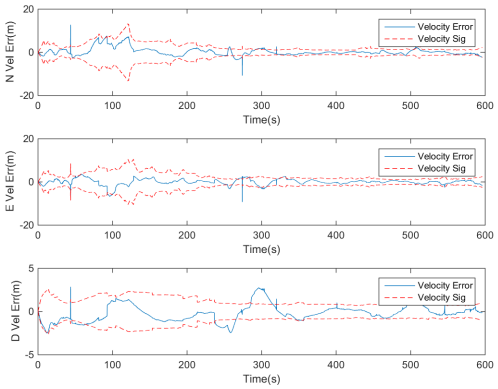

Figure 61. Scenario 7 with all available SVs paired with the monocular camera computer vision algorithm to aid the IMU velocity error with standard deviation.

When compared to the position and velocity error values of just GPS aiding the IMU, seen in Figures 38 and 39, the decrease in errors seen during GPS outages can be observed. Performance for all the systems being aided with GPS and vision can be seen in Table 12.

Table 12. DRMS value comparison for the IMU being aided with GPS and the monocular camera computer vision algorithm.

	0 SV DRMS	1 SV DRMS	2 SV DRMS	3 SV DRMS	Available SV DRMS
Scenario 2 Com IMU Forward Cam	125.4	111.3	29.6	14.5	10.8
Scenario 2 Com IMU L Cam	184.0	140.3	52.2	15.5	11.2
Scenario 2 Sim IMU Forward Cam	289.2	88.5	71.3	7.4	3.6
Scenario 2 Sim IMU L Cam	173.9	80	54.4	7.9	4.2
Scenario 6 Sim IMU Forward Cam	57.6	30.9	16.8	5.8	2.9
Scenario 6 Sim IMU R Cam	115.7	43.4	25.6	13.7	3.1
Scenario 7 Sim IMU Forward Cam	509.4	213.1	151.5	24.2	8.1
Scenario 7 Sim IMU L Cam	540.4	247.5	91.8	14.5	7.8
Scenario 7 Sim IMU R Cam	362.7	248.3	152.4	11.15	8.7
Scenario 17 Sim IMU L Cam	567.9	180.1	193	10.5	4.9

4.6 Summary

Performance of the system is greatly increased when using the monocular camera computer vision algorithm to aid the system of a GPS denied or limited environment. A comparison of all the results obtained in this thesis can be seen in Table 13.

Table 13. DRMS value comparison for all sensor combinations.

	0 SV DRMS	1 SV DRMS	2 SV DRMS	3 SV DRMS	Available SV DRMS
Scenario 2 Com IMU	55165	1516	2376	14.7	9.9
Scenario 2 Com IMU Forward Cam	125.4	111.3	29.6	14.5	10.8
Scenario 2 Com IMU L Cam	184.0	140.3	52.2	15.5	11.2
Scenario 2 Sim IMU	170934	3231	1124	10.4	4
Scenario 2 Sim IMU Forward Cam	289.2	88.5	71.3	7.4	3.6
Scenario 2 Sim IMU L Cam	173.9	80	54.4	7.8	4.2
Scenario 6 Sim IMU	68066	13324	151	6.7	2.5
Scenario 6 Sim IMU Forward Cam	57.6	30.9	16.8	5.8	2.9
Scenario 6 Sim IMU R Cam	115.7	43.4	25.6	13.7	3.1
Scenario 7 Sim IMU	443094	14655	225.8	14.7	7.7
Scenario 7 Sim IMU Forward Cam	509.4	213.1	151.5	24.3	8.1
Scenario 7 Sim IMU L Cam	540.4	247.5	91.8	14.5	7.8
Scenario 7 Sim IMU R Cam	362.7	248.3	152.4	11.2	8.7
Scenario 17 Sim IMU	425145	1283619	164010	11.6	4.5
Scenario 17 Sim IMU L Cam	567.9	180.1	193	10.5	4.9

As mentioned in [15], which was based off of all simulated measurements, the system acts like it has gained another SV by introducing the monocular camera computer vision algorithm into the system. While this thesis did not show that an IMU aided by a monocular camera computer vision algorithm alone can create a reliable solution, it did show that this solution is a great aid when operating in limited GPS areas.

This thesis also shows that there is a negligible difference between the navigation solution with a side facing camera and a forward facing camera. In general, the

number of features presented to the system was enough to give a trajectory solution. Even with putting a limit on the number of features required to obtain a solution, which was set at 20 features for this research, the test cases had very few instances that the monocular camera computer vision algorithm solution was not presented to the system.

Finally, the urban environments, depicted in Scenarios 7 and 17, had little degradation on the overall systems performance. While cross traffic, pedestrians, and proximity to buildings hindered the vision only navigation performance, the vision only system performed better than a single SV being used to aid the system. The performance of the system still saw a performance increase of one additional SV with the use of a the monocular camera computer vision algorithm aiding system and GPS.

V. Conclusion and Summary

The final conclusion and future works suggested by the work presented in this will be drawn in this chapter.

5.1 Conclusion

Use of a monocular camera computer vision algorithm, as an aiding sensor to the overall navigation solution, was presented in this thesis. Instead of using the normal eight point algorithm to determine the fundamental matrix and subsequently the rotation from that, the IMU's measured rotation values were used to constrain the features detected by a monocular camera to a translation only value. Regardless of the orientation of the monocular camera, similar final results were obtained for the left, right, and forward facing cameras. Furthermore, the introduction of each new satellite to the system only increased the overall accuracy of the system. Confirming what was already said in [15], the monocular computer vision algorithm, when aided with GPS, gave solutions that were equal to those found of a system with one additional SV. For example, a system with one SV and a monocular camera computer vision algorithm gave results similar to a system with two SVs and no monocular camera computer vision algorithm. It is important to note that the system did see different SVs during a run, which gave it a little better solution than if it tracked the same two SVs. During time periods of SVs coming in and out of view, the system gets a better solution because it now has more SVs in view. Scenario 17 shows how the system acts when the SV being viewed is held constant. The solution for GPS and IMU only solution in this case is worse than any of the other test runs.

5.2 Future Work

As mentioned earlier, there were some discrepancies that may have been introduced into the system because the monocular camera computer vision algorithm measurements were applied at time after the measurements. Instead of applying the measurements at the time of the second image to a value that the IMU measured half way between the two image time, a delayed state filter should be implemented. This will reduce the small errors that are constantly being added to and propagated forward. Currently, the error in the system will continually grow as long as the system is running.

Another thing that was not included in this work was batch updating. While the images of the system don't have to be stored to do a batch update, the translation obtained from the images can be easily stored without taxing the system's memory. Each new measurement that is given only adds a little more observability to the system.

Determining when the system is actually stopped is also very important. Because of the drift errors in the IMU and the inability of the monocular camera computer vision algorithm to detect when the vehicle is stopped, position errors will always be encountered when the vehicle is stationary. The reason behind the movement detected by the monocular camera computer vision algorithm is that the sensor interprets stationary features as movement in the forward direction only. By supplementing the system with a zero velocity measurement, the system can be constrained to no movement when the monocular camera computer vision algorithm system does not see features moving. This will keep the system from slowly drifting away when stopped.

Lastly, the orientation of a ground facing monocular camera should be tested with the measurement equations established in this thesis. By implementing this camera, the system will experience fewer outages due to sun light saturating the sensors. The

other advantages is the distance to features will be more observable. By knowing the height above the ground, less error of the system associated with observability in the camera's forward direction or z-direction can be experienced.

Bibliography

1. Alkhaldi, Humood. *Integration of a Star Tracker and Inertial Sensors Using an Attitude Update*. Master's thesis, AFIT, 2014.

2. Committee, DMA WGS-84 Development. *Department of Defense World Geodetic System 1984, Its definition and Relationship with Local Geodetic Systems*. In-house Technical Report 8350.2, Defense Mapping Agency, Washington, DC, 1987.

3. Fischler, Martin A and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*, 24(6):381–395, 1981.

4. Hartley, Richard and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

5. Kauffman, Kyle J. *Radar Based Navigation in Unknown Terrain*. Ph.D. thesis, AFIT, 2012.

6. Lowe, David G. "Distinctive image features from scale-invariant keypoints". *International journal of computer vision*, 60(2):91–110, 2004.

7. Ma, Yi. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer, 2004.

8. Maybeck, Peter S. *Stochastic models, estimation, and control*, volume 2. Academic press, 1982.

9. Maybeck, Peter S. *Stochastic models, estimation, and control*, volume 1. Academic press, 1994.

10. Nilsson, J-O, Dave Zachariah, Magnus Jansson, and Peter Handel. "Realtime implementation of visual-aided inertial navigation using epipolar constraints". *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, 711–718. IEEE, 2012.

11. Noureldin, Aboelmagd, Tashfeen B Karamat, and Jacques Georgy. *Fundamentals of inertial navigation, satellite-based positioning and their integration.* Springer Science & Business Media, 2012.

12. Pace, Scott, Gerald Frost, Irving Lachow, David Frelinger, and Donna Fossum. *The global positioning system: assessing national policies.* Technical report, DTIC Document, 1995.

13. Panahandeh, Ghazaleh and Magnus Jansson. "Vision-aided inertial navigation using planar terrain features". *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on*, 287–291. IEEE, 2011.

14. Penn, Timothy R. *All Source Sensor Integration Using an Extended Kalman Filter.* Master's thesis, AFIT, 2012.

15. Png, Jason. *Simulation Platform for Vision Aided Inertial Navigation.* Master's thesis, 2014.

16. Scaramuzza, Davide and Friedrich Fraundorfer. "Visual odometry [tutorial]". *Robotics & Automation Magazine, IEEE*, 18(4):80–92, 2011.

17. Scaramuzza, Davide, Friedrich Fraundorfer, and Roland Siegwart. "Real-time monocular visual odometry for on-road vehicles with 1-point ransac". *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 4293–4299. IEEE, 2009.

18. Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

19. Sun, Changming. "2-point linear algorithm for camera translation vector estimation with known rotation". *Robotica*, 18(05):557–561, 2000.

20. Titterton, David and John L Weston. *Strapdown inertial navigation technology*, volume 17. IET, 2004.

21. Tsui, James Bao-Yen. *Fundamentals of global positioning system receivers*. Wiley-Interscience, 2000.

22. Van Loan, Charles. *Computing integrals involving the matrix exponential*. Technical report, Cornell University, 1977.

23. Veth, Michael J. *Fusion of imaging and inertial sensors for navigation*. Ph.D. thesis, AFIT, 2006.

24. Zachariah, Dave and Magnus Jansson. "Camera-aided inertial navigation using epipolar points". *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, 303–309. IEEE, 2010.

List of Acronyms

GPS global positioning system

IMU inertial measurement unit

DCM direction cosine matrix

ASPN All Source Positioning Navigation

SFM structure from motion

RANSAC random sample consensus

FP feature point

EKF extended Kalman filter

ECEF Earth Center Earth Fixed

SIFT Scale Invariant Feature Transform

SVD singular value decomposition

TCB time correlated bias

FOGM first-order Gauss Markov

ARW angular random walk

VRW velocity random walk

INS inertial navigation system

SV satellite vehicle

ANT Autonomous Navigation and Technology

DRMS distance root mean square

WGS 84 World Geodetic System 1984

OSU Ohio State University

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

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

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26–03–2015	Master's Thesis	Sept 2013 — Mar 2015

4. TITLE AND SUBTITLE

Urban Environment Navigation with Real-Time Data Utilizing Computer Vision, Inertial, and GPS Sensors

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Rohde, Johnathan, L, 1 Lt, USAF

5d. PROJECT NUMBER

15G-103

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-ENG-15-M-037

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

Air Force Research Laboratory
101 West Eglin Blvd
Building 13AFRL
Eglin AFB FL 32542
DSN 875-2632, COMM 850-883-2632
Email: adam.rutkowski@eglin.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RY

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

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

14. ABSTRACT

The purpose of this research was to obtain a navigation solution that used real data, in a degraded or denied global positioning system (GPS) environment, from low cost commercial off the shelf sensors. The sensors that were integrated together were a commercial inertial measurement unit (IMU), monocular camera computer vision algorithm, and GPS. Furthermore, the monocular camera computer vision algorithm had to be robust enough to handle any camera orientation that was presented to it.

This research develops a visual odometry 2-D zero velocity measurement that is derived by both the features points that are extracted from a monocular camera and the rotation values given by an IMU. By presenting measurements as a 2-D zero velocity measurements, errors associated with with scale, which is unobservable by a monocular camera, can be removed from the measurements. The 2-D zero velocity measurements are represented as two normalized velocity vectors that are orthogonal to the vehicle's direction of travel, and are used to determine the error in the INS's measured velocity vector. This error is produced by knowing which directions the vehicle is not moving, given by the 2-D zero velocity measurements, in and comparing it to the direction of travel the vehicle is thought to be moving in.

The performance was evaluated by comparing results that were obtained when different sensor pairings of a commercial IMU, GPS, and monocular computer vision algorithm were used to obtain the vehicle's trajectory. Three separate monocular cameras, that each pointed in a different directions, were tested independently. Finally, the solutions provided by the GPS were degraded (i.e., the number of satellites available from the GPS were limited) to determine the effectiveness of adding a monocular computer vision algorithm to a system operating with a degraded GPS solution.

15. SUBJECT TERMS

LaTeX,Thesis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. John Raquet, AFIT/ENG
U	U	U	U	102	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4580; John.Raquet@afit.edu