UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DESIGN AND IMPLEMENTATION OF A PRECISION THREE-DIMENSIONAL BINOCULAR IMAGE TRACKER FOR DEPARTING AIRCRAFT

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

DAVID D. SANDMANN
Norman, Oklahoma
2013

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/7897

DESIGN AND IMPLEMENTATION OF A PRECISION THREE-DIMENSIONAL BINOCULAR IMAGE TRACKER FOR DEPARTING AIRCRAFT

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

_____
Dr. John Fagan, Chair

_____
Dr. Chris Ramseyer

_____ _____
Dr. Joseph Havlicek

_____
Dr. James Sluss

_____
Dr. Monte Tull

© Copyright by DAVID D. SANDMANN 2013
All Rights Reserved.

Dedication

I have many people to thank for their assistance in earning this degree, without them this would not have been possible. First and far most are my parents, Charles and Phyllis Sandmann. Without their continual support and love none of this would have been possible. They have never stopped believing in me and have always supported me in all my efforts.

Next is my mentor and best friend, Dr. John Fagan. Through his never ending efforts, teaching, and guidance he has gotten me past tough times. He has shown me what it takes to be a good engineer and taught me that ‘Life is Good’.

Last but not least are all the Faculty, Staff, and Graduate students from the University that have helped me. They have provided support and insight to this difficult problem.

Acknowledgements

Dr. John Fagan

Dr. John Dyer

Dr. Rick Pentagraph

Master Jacob Henderson

Master Jordan Kuehn

Master Vasilly Mayer

Dean Alexander

Jim Thrash

Table of Contents

Dedication 4

Acknowledgements iv

List of Tables vii

List of Figures viii

Abstract x

Chapter 1: Introduction 1

1.1 Obstacle Clearance Guidelines 2

1.2 Wake Turbulence 4

1.3 A/P Standards and TERPs 5

1.4 Visual Tracking 6

Chapter 2: Literature Review 9

Chapter 3: Techniques 12

3.1 Window Tracking 12

3.1.1 Median Filtering 12

3.2 Feature Tracking 13

3.2.1 Tracking Local Features 14

3.2.2 Optical Flow Methods 14

3.2.3 Tracking Extended Features 14

3.3 Rigid Shapes 15

3.3.1 Planar Rigid Shapes 15

3.3.2 Solid Rigid Objects 15

Chapter 4 Experimental Setup 17

4.1 Camera Systems 17
4.2 Binocular Cameras 19
4.3 Prototype Construction 24
4.4 Camera Setup 29
4.5 Software Model Development 38
Chapter 5 Data Analysis 43
5.1 Calibration Flights 44
5.2 Data Processing 46
5.3 Validation Flights 49
5.4 Results 50
Chapter 6 Conclusions 56
References 57
Appendix A: LabVIEW Code 61
Appendix B MatLab Code 65
Appendix B1 Datum Transformations 65
Appendix B2 Validation Flights 67
Appendix B3 Calibration Matrix Script 71
Appendix C: Sample Data 74
Appendix C1: Sample Calibration Data 74
Appendix C2: Sample Flight Data 79
Appendix D: Generated Tracks 80

List of Tables

Table 1: Surveyed Points 29

Table 2 East North Up Coordinates 34

Table 3 Standard Deviations for Calibration Flights 53

Table 4 23 October Flight 11:29:13 79

List of Figures

Figure 1 Departure Design Criteria [1] 3

Figure 2 Pinhole Camera Model [28] 18

Figure 3 Perspective Projection [29] 19

Figure 4 Parallel Optical Axes [30] 21

Figure 5 General Vergent Stereo Setup [31] 22

Figure 6 Master Box 26

Figure 7 Outside of Master Box 27

Figure 8 Camera Placement 31

Figure 9 Transformations between ENU and ECEF Coordinates [32] 33

Figure 10 Flow Chart 40

Figure 11 Raw Tracking Data 43

Figure 12 Calibration Flights 45

Figure 13 July 26th Flight 48

Figure 14 Generated Flight 48

Figure 15 Validation Flights Left 50

Figure 16 Vertical Plot Calibration Flight Right 51

Figure 17 Horizontal Plot Calibration Flight Right 51

Figure 18 Vertical Plot Calibration Flight Left 52

Figure 19 Horizontal Plot Calibration Flight Left 52

Figure 20 November 9th Flight 53

Figure 21 November 9th Generated Track 54

Figure 22 November 19th Flights 55

Figure A 23 Camera Setup 61
Figure A 24 Cross Hairs 61
Figure A 25 Tracking 62
Figure A 26 Zip and Send 63
Figure A 27 Zip Data 63
Figure A 28 FTP Download 63
Figure A 29 Front Panel 64
Figure D 30 November 8th Flight 80
Figure D 31 November 8th Generated Track 80
Figure D 32 November 9th Flight 81
Figure D 33 November 9th Generated Track 81
Figure D 34 November 15th Flight 82
Figure D 35 November 15th Generated Track 82
Figure D 36 November 15th Flight 83
Figure D 37 November 15th Generated Track 83
Figure D 38 November 9th Flight 84
Figure D 39 November 9th Generated Track 84

Abstract

This dissertation presents the result of the conceptualization, design and implementation of a new, novel and low cost Binocular Tracking System for departing Aircraft. This system is a unique design due to the commercial off-the-shelf (COTS) components used and the distinct modular algorithms developed for the implementation of tracking aircraft.

Recent economic pressures and changing Federal Aviation Administration (FAA) regulations have raised serious concern that obstacle clearance requirements are not being met on commercial aircraft departure. Moreover, local airport procedures do not always align with the requirements for Terminal Instrument Procedures (TERPs) established by the FAA. The flight track data collected by this system is being used by the FAA to assess the magnitude of the problem and determine steps to align airport and TERPs procedures, while also mitigating obstacle clearance violations and thus the risk of departing aircraft encountering an obstacle.

Each of the binocular tracking systems uses three cameras. One camera is directed towards the runway, initializes the tracking algorithms, and identifies the type of aircraft. The other two cameras form the binocular tracking system. These dual cameras are aligned in a vergent stereo configuration across the departure path to provide the maximum overlap in the field of view to produce a superior depth resolution.

The modular tracking algorithms allow a large volume of tracking data to be accumulated that provides the FAA information on departing aircraft. This dissertation discusses the details of the binocular tracking system's conceptualization, design, and

implementation, including hardware and software development of the tracking system. This dissertation also includes system setup, data collection, processing and error analysis of the system's performance in the field.

Chapter 1: Introduction

In recent years, and in the current economic environment, air carriers have incurred an increase in operational cost due to complying with new regulations and performance requirements. These new, more stringent regulations are designed to efficiently increase airspace use. The Federal Aviation Administration (FAA) has three fundamental regulatory goals:

1) Obstacle Clearance.

2) Wake Turbulence.

3) Airport (A/P) standards and Terminal Instrument Procedures (TERPs) moving toward same criteria (D. Alexander, personal communication, October 5, 2013).

The FAA is extremely concerned that air carriers are violating the obstacle departure clearance guidelines which increase the risk of collision (D. Alexander, personal communication, October 5, 2013). The increased economic pressures on the air carriers are causing them to increase payload and also decrease the power, to improve fuel economy and comply with new noise abatement restrictions, needed to climb properly to maintain the safety margin of departing aircraft from our nation's largest airports. The ability to closely fly centerlines has added new concerns for wake turbulence. Spacing used for wake turbulence avoidance at many airports in many cases does not meet the FAA requirements, placing aircraft in jeopardy. Furthermore, airport operations may produce procedures that vary at times with the FAA's TERPs criteria, contributing to potential obstacle clearance violations (D. Alexander, personal communication, October 5, 2013). The FAA needs an inexpensive way to accurately

track the aircraft during the departure phase at a wide variety of airports to determine the performance effects of the next generation systems and flights procedures and develop the next collision risk model. The system needs to accurately track the three-dimensional position relative to the runway centerline of departing aircraft up to a range of five miles. In addition, the monitoring system needs to be able to identify the aircraft's tail numbers so the FAA will be able to keep track of the type of aircraft and precisely determine departure profiles in the critical first moments of flight after takeoff. Ninety to ninety-five percent of the potential risk of any flight is in these first few moments of the departure flight. A secondary purpose of precise position tracking is the desired ability to merge airport clearance standards with terminal instrument procedures standards.

1.1 Obstacle Clearance Guidelines

The FAA has specific regulations regarding obstacle clearance for both departure and approach procedures in the terminal area [1], [2]. Obstacle clearance guidelines for departures are incorporated in FAA preplanned procedures which ensure obstruction clearance from the departure end of the runway to the appropriate start of an en-route path. En-route structures are defined as any obstacle that penetrates the obstacle clearance surface [1]. The primary purpose of departure procedures is to provide obstacle clearance. The second purpose is to increase efficiency at busier airports by reducing the number of communications and departure delays through the use of preplanned departure procedures. Terminal Instrument Procedures (TERPs) are developed by the FAA based on the ability to meet the required obstacle clearance

guidelines. Precision navigation has significantly improved the ability to keep aircraft in a narrowly defined corridor, which does encourage operators to push the obstacle clearance envelope for established TERPs. Two standards currently exist for designing departure procedures: Obstacle Departure Procedures (ODP) and Standard Instrument Departures (SID) see Figure 1 below [1], [2].

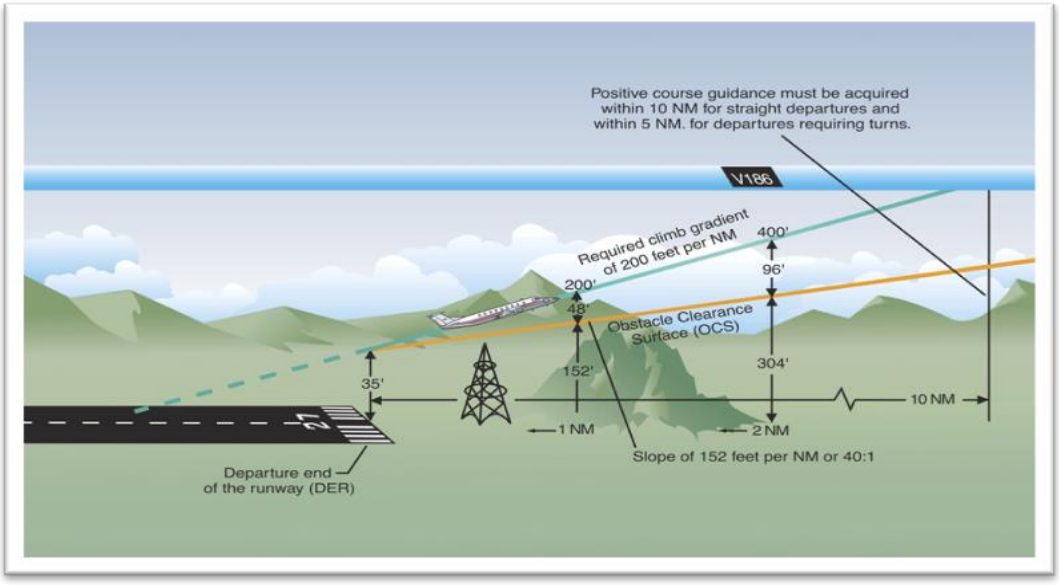


Figure 1 Departure Design Criteria [1]

In the design criteria of these guidelines, obstacle clearance is based on the assumption that the aircraft will cross the end of the runway at 35 feet, make no turns until 400 feet above the departure end of the runway, and climb at a minimum rate of 200 feet per nautical mile [2]. When approved obstacles penetrate the obstacle clearance surface, a higher climb gradient is required. Obstacles that are located within 1 nautical mile (NM) of the departure end of the runway and penetrate the 40:1 obstacle clearance standard are referred to as low, close-in obstacles. Here the standard required obstacle clearance of 48 feet per NM to clear these obstacles requires a climb gradient

greater than 200 feet per NM for a very short distance. The low, close-in obstacles are what the FAA is primarily concerned with (D. Alexander, personal communication, October 5, 2013).

1.2 Wake Turbulence

All aircraft produce wake turbulence. Wake vortices are formed anytime an airfoil is producing lift. Lift is generated by the creation of a pressure differential over the wing surfaces. The lower pressure occurs over the upper surface and the higher pressure under the wing. This pressure differential triggers the rollup of the airflow aft of the wing resulting in swirling air masses trailing downstream of the wingtips. Viewed from behind the generating aircraft, the left vortex rotates clockwise and the right vortex rotates counterclockwise. The strength of the vortex is governed by the weight, speed, and shape of the wing of the generating aircraft. The vortex characteristics of any given aircraft can also be changed by extension of flaps or other wing configuring devices as well as by change in speed. However, the basic factor is weight and the vortex strength increases proportionately with weight [2].

During takeoff, jet engine blast (thrust stream turbulence) can cause damage and upsets if encountered at close range. Exhaust velocity versus distance studies at various thrust levels have shown a need for light aircraft to maintain an adequate separation behind large turbojet aircraft. Vortices are generated from the moment aircraft leave the ground, since trailing vortices are a by−product of wing lift. The greatest vortex strength occurs when the generating aircraft is HEAVY, CLEAN, and SLOW [2]. It is

these effects of wake turbulence that the FAA is concerned with and needs to document (D. Alexander, personal communication, October 5, 2013).

1.3 A/P Standards and TERPs

The Purpose of the United States Standard for Terminal Instrument Procedures is to prescribe the criteria for the formulation, review, approval and the publishing of procedures for IFR (Instrument Flight Rules) operations to and from civil and military airports [24].

TERPS criteria have been established for the following Instrument Procedures: Precision Approach (PAR, ILS, MLS), Non-Precision Approach (VOR, TACAN, LNAV, NDB, ASR), Approach with Vertical Guidance (LDA, LPV, VNAV), Required Navigation Performance (RNP) and for Departure Procedures (DP).

The key considerations for developing Terminal Instrument Approach and Departure Procedures include but are not limited to existing obstructions, ground/satellite based equipment, lighting and aircraft category.

TERPS criteria specify the minimum measure of obstacle clearance that is considered by the Federal Aviation Administration (FAA) to supply a satisfactory level of vertical protection from obstructions and are predicated on normal aircraft operations [2], [24].

Understanding the complexities of Terminal Instrument Procedures is difficult since it is a highly specialized field requiring directed study and/or experience in its applications. This is another reason for the development of the precision vision system

which precisely determines departure profiles in the critical first moments of flight, an area where no collision risk model exists.

1.4 Visual Tracking

Tracking targets using video has become very cost-effective recently due to the increased speed of low-cost computers. Video images contain a large amount of data which used to require dedicated special purpose hardware and offline processing. However, with modern computing techniques, this can now be accomplished in near real-time. Video tracking is the process of locating a moving object (or multiple objects) over time using one or more cameras. It has a variety of uses: security and surveillance [12], aircraft tracking [13], guided landings and takeoffs [20], traffic control, medical imaging and targeting in military applications [5]. In military applications, video tracking is very attractive because vision sensors are passive and robust to jamming compared to the current use of radar [3]. Binocular (Stereoscopic) video tracking uses two cameras to track an object in a video sequence. Binocular tracking has an advantage over single camera video tracking because, with binocular tracking, the distance to the object in question can be determined without prior knowledge of the object being tracked.

The objective of binocular video tracking is to associate target objects in consecutive video frames and then calculate their distance, and height, from a certain point, usually from the cameras. The association can be especially difficult when the objects are moving fast relative to the frame rate. Another difficultly that increases the complexity of the problem is when the tracked object changes orientation over time, as

aircraft do. For these situations video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object. Mian [3] basically describes how tracking algorithms can be divided into two categories:

1) Optical flow - exploit the fact that there is minimal change in images that are taken at small intervals of time.
2) Local feature - extract local features (regions) from the first frame and search for the corresponding features in the subsequent frames.

In a similar paper, Trucco et al [4] describe tracking systems that must address the two basic problems:

1) Motion problem: predict the location of an image element being tracked in the next frame; that is, identify a limited search region in which the element is expected to be found with a high degree of probability.
2) Matching problem: (also known as detection or location) identify the image element in the next frame, if it exists, within the designated search region.

Both of these explanations are basically the same. There are a variety of algorithms, each having strengths and weaknesses, that address each of these problems.

This dissertation develops a unique, precision, low-cost system that is capable of tracking any type of aircraft over a wide range of departure trajectories. Using an innovative ensemble of off-the-shelf components, and developing a set of modular software algorithms, the system can identify each aircraft and record its three-dimensional departure during day and low-light levels. Chapter 2 discusses the current

literature that addresses the two basic problems with video tracking and the algorithms used. Chapter 3 looks at all the current techniques used in visual tracking. Chapter 4 describes the experimental setup that was used to develop the unique binocular 3-D tracking system. Chapter 5 discusses the data collected from the new system implementation and how it was validated to prove the final goal of this dissertation. Finally, Chapter 6 discusses the results and conclusions. This dissertation accomplishes two goals: 1) the development of a unique, precision, low-cost system that can be implemented with off the shelf components; 2) Determination of the clearances the air-carriers are maintaining for proper safety limits in departures.

Chapter 2: Literature Review

There has been a tremendous amount of data published on visual tracking [4]. Anytime a new and novel project is undertaken, a review needs to be conducted to make sure others have not attempted, or completed this project or any aspect of this project before. While there are limited publications on visual tracking of aircraft for use in the military [3], [5], the bulk of the published research on visual tracking relates to human facial recognition or tracking of objects for avoidance in machine vision [6], [7], and [8]. To the best of my knowledge, there are no published references related to visually tracking aircraft during the low-altitude phase of departure prior to radar acquisition.

The goal is to develop a binocular video tracking system that can first identify the aircraft as it rolls down the runway and then track the departing aircraft for the required distance or length of time. In order to accomplish this task numerous areas in current literature were considered. As mentioned briefly in the first section, any tracking system must consider two basic problems:

1) Optical flow/Motion problem: this concept exploits the fact that there are minimal changes in images that are taken at small intervals of time even in the presence of relative motion between the camera and objects in the same scene.
2) Matching/feature based algorithms: this concept identifies the image element in the next frame that is within the designated search region [10], [11].

The simplest approach to the motion/optical flow problem is to define the search area in the next frame as a fixed-size region surrounding the target position in the

previous frame [12]. The Kalman filter is a well-known optimal recursive estimator that can be used for this [7], [13], and [19]. The Kalman filter can work well in this application since there is only one object to track. The Kalman filter is based on Gaussian distribution and it can be used to track multiple objects as long as the tracking algorithm initiates separate recursive filters for each target [7]. In non-Gaussian or nonlinear dynamics are involved, then the Kalman filter is generally not an optimal solution to the problem. If it is determined that other aircraft will be within the search area, then particle filtering may be preferable to allow multimodal distributions, which may be a direct consequence of tracking of multiple targets [19].

The matching, or feature-based, tracking algorithms basically identify features in the previous frames and match them to the current frame. Features can be image locations with special properties such as edges, lines, and corners. For feature detection the object of interest must be detected by an object detection algorithm like a Haar-based detector [6]. As with multiple targets, Condensation (particle filtering solution) [10] can be incorporated to track numerous targets. In all cases the tracker that is developed must include the following [6]:

1) Robustness to clutter: the tracker does not get confused by image elements that resemble the target being tracked (i.e. at short distances birds may resemble planes).
2) Robustness to occlusion: the target is not be lost by temporary occlusion (i.e. plane being partially or completely hidden due to cloud(s), must be able to coast

for some number of frames before the algorithm can re-acquire or drop the track).

3) False positives/negatives: only valid targets should be classified as valid and any other image element ignored.
4) Agility: tracker should follow the target moving with significant speed and acceleration (i.e. plane turning out of reference).
5) Stability: accuracy should be maintained over time.

This research develops a tracking algorithm that uniquely incorporates many of the above mentioned features which are required for the tracking of departing aircraft. For instance, since there will only be one plane to track at a time the use of Kalman filtering is preferable. In addition, if a variable frame rate is used to compensate for different aircraft speeds better resolution can be obtained on the pixel level. In the papers examined to date a fixed rate is incorporated. But due to changing speed and distance of the aircraft expected to be tracked in this project, it may be desirable to have a variable frame rate in order to produce a better tracking algorithm.

There are many different areas in the tracking algorithm that can use feature detection. While traveling down the runway, the cameras will have a side shot of the aircraft. During liftoff the cameras will be able to detect the aircraft's rotation, the turning up of the nose in order to climb. And finally the plane might bank left or right and a full top view of the plane may be seen. Each one of these situations allows identification of special properties that can be used in feature based matching. All these areas must be considered during the development of the tracking algorithm.

Chapter 3: Techniques

There have been many different techniques developed for visual tracking. This chapter discusses the major techniques starting from the simplest and advancing to the more complex with emphasis on how they can be used to solve the current problem in the low cost system.

3.1 Window Tracking

The window tracking technique is one of the simplest. Basically, a small rectangular region is defined on the image. This rectangular region reduces the size of the video image to be processed. Once the window is defined it is tracked from frame to frame by correlation-like correspondence methods, (i.e. cross correlation, sum of squared differences and other metrics) [4].

The search region in the next frame can be determined by a Kalman Filter (KF) or simply taken as a fixed-size window centered on the previous target. Window matching techniques can be applied at any pixel. This type of tracking is often used in stereo vision systems.

3.1.1 Median Filtering

The median filter is a type of sliding-window spatial filter. It replaces the center value in the next frame for the sliding-window with the median pixel values from the previous sliding-window.

3.1.2 Centroiding

The centroid of an object is the point of balance or the center of mass. Centroiding is useful because the centroid of an object is invariant to any geometry distortion regardless of scaling, rotation or translation. The centroid is defined as:

$$X_c = \frac{x_1 m_1 + x_2 m_2 + \cdots + x_n m_n}{M_T}, \quad (1)$$

where

$$m_1 = mass\ for\ point\ x_1,$$

$$m_2 = mass\ for\ point\ x_2,$$

$$M_T = total\ mass.$$

Centroiding is efficient when used for binary images or grayscale images since these consist of only 2 levels which are either black (pixel value = 0) or white (pixel value = 255).

3.2 Feature Tracking

Features are defined as detectable parts of an image which can be used to support a vision task [4]. These basically can be any corner, line contour or specially defined region.

In order to use feature tracking it is necessary to define two sequential frames, I_t and I_{t+1}, and then associate features in I_{t+1} with corresponding features in I_t (if they exist). This technique is very similar to window tracking, but the main difference is that features are now particular image elements and not just a defined window.

3.2.1 Tracking Local Features

Local features are image locations with special properties such as edges, lines and corners [27]. Local features offer some invariance to image changes caused by changes in illumination thereby improving detectability over time. Edges remain detectable over a wide range of viewpoints. Also extracting features reduces substantially the amount of data to process which is desirable in a low cost system.

Features that belong to the target are usually selected by performing an exhaustive search over the whole initial image. After that the search region can be limited by a window or other KF type method.

3.2.2 Optical Flow Methods

These methods do not use features; they are based on image brightness constancy constraint [4], [9]. This method assumes that local intensities do not vary significantly from frame to frame. Only an approximation of target velocity can be computed from an image sequence because the only data measurable are changes of intensities which are not only caused by motion but also by reflections, illumination changes. The intrinsic difference between feature tracking and optical flow method is that the former are based on feature comparisons within arbitrary search areas and can match distance features [11].

3.2.3 Tracking Extended Features

Extended features cover a larger part of the image than local features; they can be contours of basic shapes (e.g. ellipses, rectangles), or free-form contours, or image

regions. Extended features are more robust to clutter than local features because they rely on a larger image area. But the larger image area requires longer processing time [4], [12].

Both regions and contours are popular extended features [4]. Regions are defined as connected image parts with distinguishing intensity or color properties. Contours are defined as intuitively meaningful features, often taken as the boundary of a whole object.

3.3 Rigid Shapes

3.3.1 Planar Rigid Shapes

The shape of the image of a planar, rigid object depends on its position and orientation with respect to the camera, as well as the type of imaging projection [6], [12]. Simple shapes like rectangles or circles can be captured by closed-form models to predict their appearance in the image. This method assumes knowledge of which image features belong to the target and which are background. This is sometimes difficult and is usually tackled with a priori knowledge about target properties.

3.3.2 Solid Rigid Objects

Using a model of the solid rigid object being imaged, tracking can yield 3-D position and orientation in time and therefore 3-D velocities [20]. The basic component is an instantaneous estimation of rotation and translation in space. First, contours detected in the image and contours in the model are put in correspondence. Then the

pairs of matching contours are used to estimate the rotation and translation of the model contours generating the best approximation of the observed image by minimizing an integral function of the distances between corresponding contours. This step is repeated for each new frame of a video sequence using the previous estimate as an initial guess.

After experimenting with numerous techniques it was decided to use a window tracking technique. A basic window is defined as the region of interest (ROI) and centroiding is computed in this window. Once motion is detected and the centroid is found a simple two-state Kalman Filter is applied to define where the next window will be located.

Chapter 4 Experimental Setup

Before any design and fabrication is started, background information is needed on how cameras operate especially stereoscopic cameras. All images are multi-dimensional signals (two space dimensions). In order to track aircraft it is necessary to understand how to take the two-dimensions (2-D) given in an image and relate this to three-dimensions (3-D) necessary for tracking.

4.1 Camera Systems

The basic camera consists of a lens and detector array. Ambient light is reflected by the objects in the camera field of view, passes through the lens and is absorbed by the detector. This type of image is referred to as Reflection Imaging; image information is surface information describing how the object reflects/absorbs incident radiation [26]. To model the ideal camera, the pinhole model is used. This model can mathematically describe how the geometry of an image projects to the geometry of the intensities in the image. Anytime an image is taken, a reduction of dimensionality (from 3-D to 2-D) occurs and information is lost. The information that is lost is the spatial 3-D relationship between the objects in the scene being imaged. But there is still a precise geometric relationship that relates the spatial 3-D coordinates to image 2-D coordinates under this model. Using real-world coordinates a point in 3-D space can be defined as (X, Y, Z). The origin (X, Y, Z) = (0, 0, 0) is taken to be the lens center. For the image coordinates, let any point in the 2-D image be represented by (x, y). The x – y plane is chosen parallel to the X – Y plane and the optical axis passes through both origins. In the idealized camera model the lens is treated as a simple

pinhole through which all light hitting the image plane passes. The image plane is one focal length, f, from the lens. This can be illustrated in Figure 2 below.

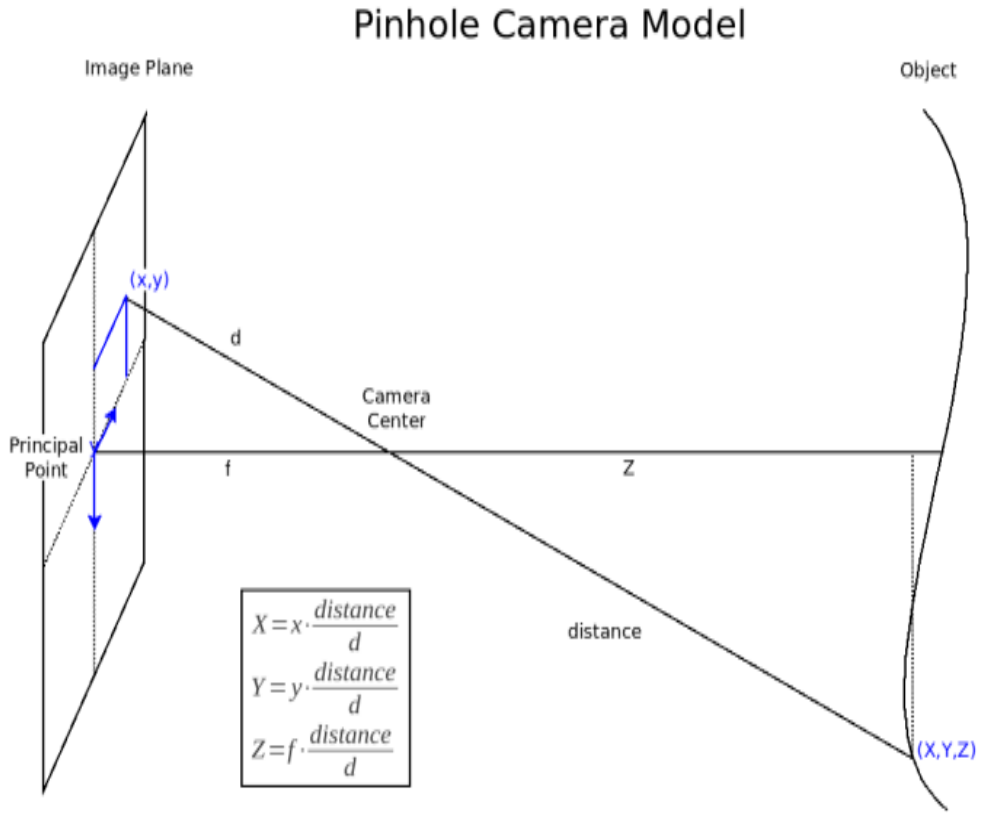


Figure 2 Pinhole Camera Model [28]

This model shows that the image is reversed and upside down, as in reality. It can also be seen that the image is smaller than the actual scene, since it is a projection. Before mathematical models can be incorporated one more change is required. To change the model, the Upright Projection Model is used. It is convenient to think of a front image plane rather than the actual image plane so that the image is not inverted in direction relative to the world. This model inverts the image plane and brings it back into the positive coordinates where the focal length, f, is now towards the original image. This representation can be seen in Figure 3, below.

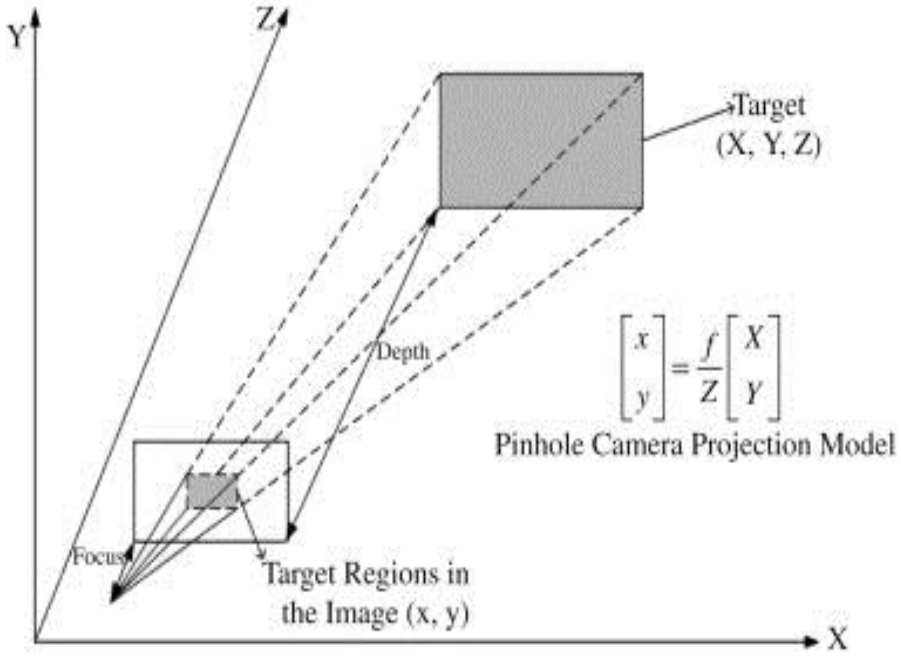


Figure 3 Perspective Projection [29]

Mathematical equations can now be used to equate our points to distances and sizes. Using similar triangles the object and image side lengths are in proportion. The following relationship between 3-D space coordinates (X, Y, Z) and 2-D image coordinates (x, y) can be related as

$$(x, y) = \frac{f}{Z} * (X, Y). \quad (2)$$

4.2 Binocular Cameras

The region that a camera can see is known as the view frustum which can be described as the 3-D shape similar to a pyramid. In order for the binocular system to operate the object of interest must be visible in the view frustum of each of the two cameras. Longer focal lengths increase the accuracy of measurements while decreasing the volume of the frustum. Smaller focal lengths increase the volume of the frustum

while decreasing accuracy of measurement. Variable focal lengths, i.e. zoom, complicate the acquisition system and make it more expensive.

Two pinhole cameras together will help collect more information than a single camera. Using two cameras together will allow the measurement of size and position of an object with no prior knowledge of the object. The 3-D structure of the object is computed from two flat projected images. The basic stereo vision system utilizes parallel optical axes as shown in the following Figure 4. No rotations are required and simple geometric relationships apply. In theory the camera x-axes are collinear and their y-axes and z-axes are parallel. To compute the subject position, a triangle is defined by the two cameras and the target. The stereoscopic basis or baseline is defined as the distance between the cameras. Each point on the object (X, Y, Z) is projected onto differing points within each camera's image plane (x's, y's). The difference in measurements x_1 and x_2 is known as the disparity, or parallax. Disparity (d) refers to the difference in the image location of the same 3-D point when projected under perspective to two different cameras [14]. Using similar triangles and the disparity between the cameras gives

$$Z = f * \frac{B}{x_1 - x_2} = f * \frac{B}{d}, \tag{3}$$

$$X = x_1 * \frac{Z}{f} = B + x_2 * \frac{Z}{f}, \tag{4}$$

$$Y = y_1 * \frac{Z}{f} = y_2 * \frac{Z}{f}. \tag{5}$$

Computing the subject distance z is reduced to the intersection of two lines. Knowledge of the camera geometry (f, and B) and measured qualities (x1 and x2) allows computation or coordinates (y, z) or two equations with two unknowns.

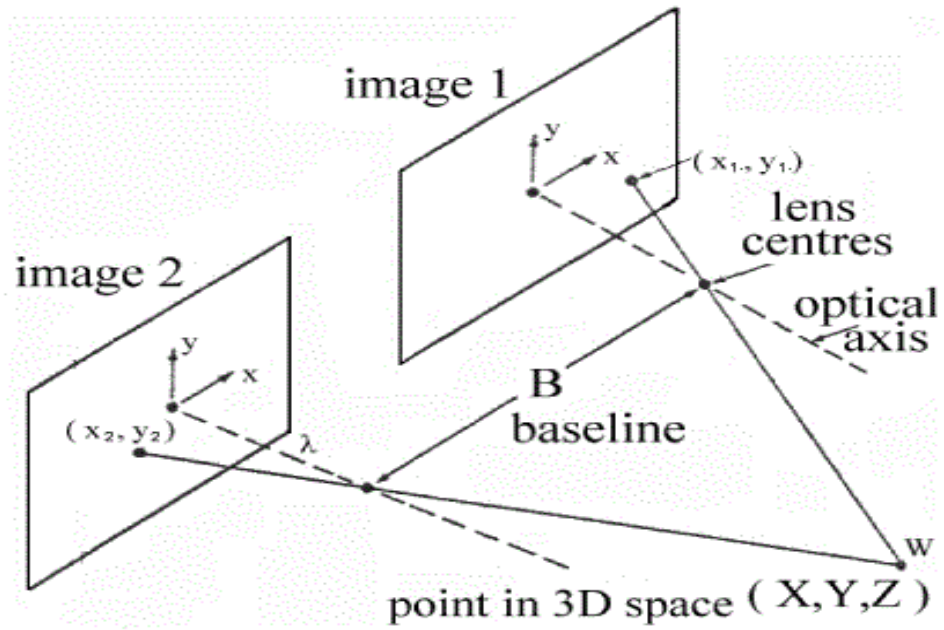


Figure 4 Parallel Optical Axes [30]

The foundation of stereoscopic systems is the intersection of these two epipolar lines. Stereo matching is simplified if the relative orientations of the cameras are known. In the simplest case shown in Figure 4 the two image planes are identical and are parallel to the baseline. This setup is called a normal image pair. This parallel baseline geometry offers a simple correspondence between points but the achievable depth resolution is low in comparison to a vergent imaging geometry [17], [18].

To cover the area needed to track the aircraft, this normal setup cannot be used. A more general setup, vergent stereo, orients the cameras toward a common point some specified distance from the center of the camera baseline. Vergent stereo provides the maximum overlap in the fields of view to the two cameras on a given object, providing a natural mechanism for concentrating visual attention [18]. Now the optical axes are not perpendicular to the baseline and are not parallel to each other. This complicates

the mathematics and requires knowledge of the camera's internals (i.e. focal length, optical axis). A generalized camera setup is shown in Figure 5 below.

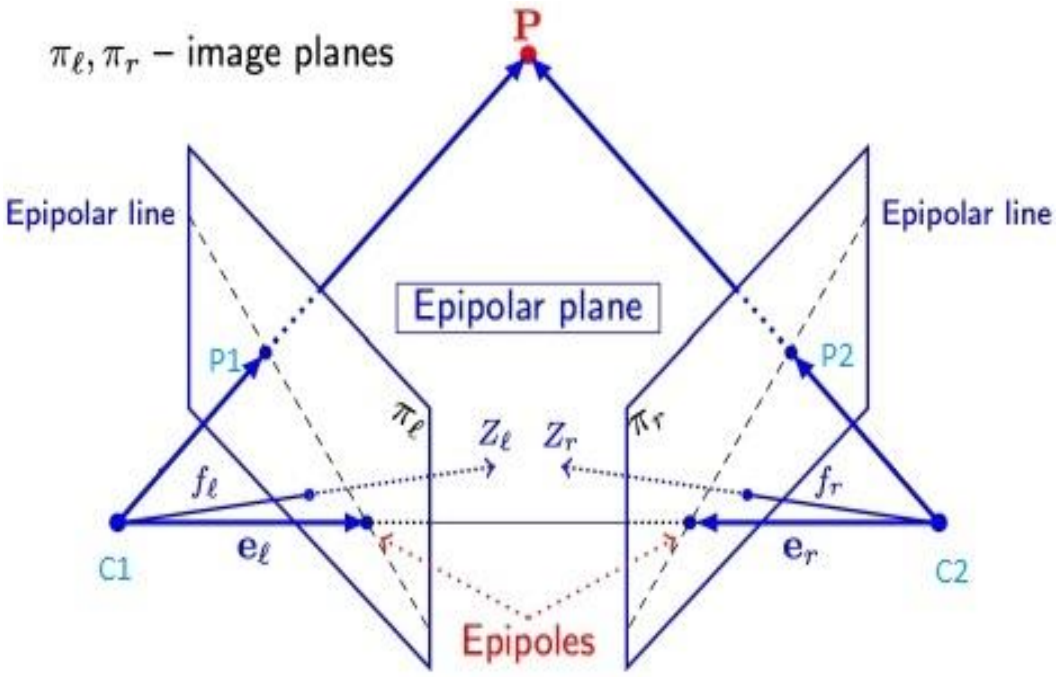


Figure 5 General Vergent Stereo Setup [31]

The epipolar plane shown in Figure 5 is defined as the plane that contains the 3-D point P, the two optical centers C_1 and C_2 and the two image points P_1 and P_2 to which P projects. The epipolar lines are defined as the intersection of the epipolar plane with the two image planes π_l and π_r. This projection relationship is significantly more complex than the parallel stereo geometry. The search space for a point on one frame is defined by an epipolar line in the second frame, dependent on the baseline separation of the cameras and the vergence angle. Using the above information the camera model can be expressed as

$$\begin{bmatrix} u \\ v \\ s \end{bmatrix} = P_f R_\theta T_B \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{6}$$

with

$$x_{im} = \frac{u}{s} \; and \; y_{im} = \frac{v}{s},$$

where P_f is the camera parameter matrix, $R_{\boldsymbol{\theta}}$ is the rotational matrix, and T_B is the translational matrix. The image plane coordinates are given in x_{im} and y_{im}. Expanding these out to show all the parameters, we have

$$P_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-1}{f} & 1 \end{bmatrix}, \tag{7}$$

$$R_{\theta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{\theta_z} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

$$T_B = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

The focal length is given as f and can vary between cameras, the angle from parallel from the runway is given by $\boldsymbol{\theta}$, and given in degrees, and finally the baseline is given by x_0 (in meters). Since absolute center is defined as the center point on the runway between boxes the matrix will have x_0 and $-x_0$ for the master box and slave box respectively.

In order to perform general stereo computations the following items must be known: (refer to Figure 5)

1) The pose of the master camera C_1 in the world coordinate system W and some camera internals such as focal length. All this information will be represented by a camera matrix which mathematically defines a ray in 3-D space for every image point P_1.
2) The pose of the remote camera C_2 in the world coordinate system W and its internal parameters.
3) The correspondence of the 3-D point to the two 2-D image points (P, P_1, P_2).
4) A formula for computing P from the two imaging rays PC_1 and PC_2).

A world coordinate system, W, must be defined. The origin of our world coordinate system is defined at the intersection of the line formed between the two boxes and the line formed between the two surveyed points on each end of the runway. Since this dissertation deals with the tracking of aircraft, the coordinate system uses the East-North-Up (ENU) convention. Using this convention, the center of the runway between the two boxes is (0, 0, 0) and the positive x-axis is pointing east, the positive y-axis is pointing north and the positive z-axis is pointing up, directed normal to the local tangent plane. The exact ENU coordinates are calculated in Section 4.4.

4.3 Prototype Construction

In order to develop and test different algorithms it was necessary to design and build a test prototype. This project identifies an aircraft from its takeoff through its departure for about 5 miles. Based on discussions with the standards section, and the

technical center of the FAA, it was determined that the system would be comprised of two boxes and three cameras. The boxes sit approximately 250 feet to each side, from the center line of the runway, or 500 feet from each other. Two cameras are placed in the first box and one in the second box. The second box only contains a wireless network camera, battery and solar panel, and is called the remote box. This box communicates directly to the first box wirelessly. The first box has two cameras and is the master controller of the system. The first box also contains an air-card and computer that communicates with a remote server to download data each day.

The prototype master box is shown in Figure 6. Both cameras can be clearly seen. The Axis P1343 is mounted to look sideways in order to detect when a plane is ready for departure. It is also used to identify the type of aircraft and tail number designation in order to keep accurate records of departures. The Axis 211A looks down the runway and is used in conjunction with the Axis 211W camera in the remote box to track the aircraft. Both of these cameras have a zoom lens attached. The prototype boxes incorporated a 1-to-2 zoom lens. Testing was conducted to make sure the pixel resolution needed to track the aircraft could be maintained. These tests are discussed later in this section. The main computer can also be seen. A National Instruments PXI chassis, with a PXI-8106 embedded controller is used. This controller has an Intel Core 2 Duo processor T7400 (2.16 GHz dual core processor). A solid-state hard drive was installed and memory was upgraded to 4 gigabytes to help in processing speed.

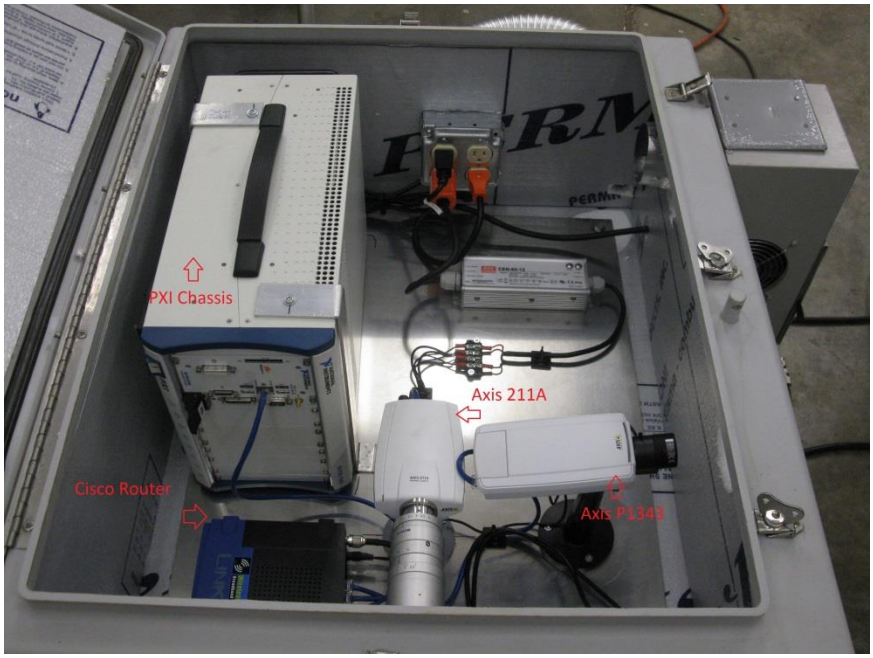


Figure 6 Master Box

The Cisco wireless-G router is seen in front of the PXI chassis. The router is connected to a 2.4 GHz 14 dBi dual polarized dual feed sector panel antenna. The complete setup can be seen in Figure 7. The remote box has a 2.4 GHz 14 dBi flat patch wireless LAN antenna mounted on the side of the box. The boxes are placed with line of sight directly across the runway from each other. The two boxes were originally tested at the University of Oklahoma Research Campus with a separation of at least 500 feet. No communication problems were observed during initial testing. Furthermore, performance of the modified prototype installation at Will Rogers World Airport has also demonstrated exceptional inter-box communication performance.

The master box requires airport power due to the power requirements of the PXI chassis and the air conditioner needed to keep all the components cool. The air conditioning unit can be seen in Figure 7.



Figure 7 Outside of Master Box

After construction of the prototype boxes, a series of tests, with a scale model, were run to test the initial setup code and to judge the lens usability. Numerous problems were found with the lens, router, and side camera.

The first test was conducted indoors to test the zoom function of the camera. A scale model of a 747-100 was used to simulate what a real plane would look like at a scaled distance. This model was chosen because the 747 will be one of the types of aircraft tracked and if the system has the resolution to track the 747 it will be able to track the remaining aircraft. The model was set 100 feet from the camera. The scale model has a 1 foot wing span which corresponds to a true 93 feet wing span. Using similar triangles:

$$\frac{1}{100} = \frac{93}{d}.$$

This gives a distance of 9300 feet, or the real distance a plane would be with the wingspan of 93 feet. Changing the distance to pixel area produces 110 pixels for this

plane at 9300 feet. At three miles, 15,840 feet, the plane should still be visible with 68 pixels. Further measurements allowed calculation of the horizontal frustum at 3.4367 degrees.

It wasn't until outside tests were performed that the lens was found to be unusable. Since the first lens was a manual adjustment iris, problems were found when the sun passed over the front of the box. A new lens with auto iris was introduced. This lens was a Fujifilm 1/3" 15 – 50 mm (3.3x) zoom lens. The angle of view increased to 5°35' x 4°12' at full zoom. The auto iris allows the camera to function throughout the day without image distortion or bleeding/saturation.

The next set of tests that were conducted used both boxes. The boxes were set 700' feet apart and the Linksys routers were tested to make sure the remote camera stayed linked and synced to the base computer. The initial tests were successful in that the boxes stayed linked. It wasn't until the air card was attached that problems were found. The air card would disconnect the Linksys router from the base computer or it would not connect to the internet if both were used. A new router, Cradlepoint MBR1400, was employed to solve the problem. This router was specifically designed to use air card(s) and be a wireless access point unlike the Linksys.

The last problem was not discovered until the boxes were placed at the airport here at KOUN. The Axis P1343 started exhibiting problems after it had been running a few days. The picture started getting darker and showed a red hue. Adjustments to hue and contrast didn't help the problem so the defective camera was replaced with the same camera that looks out the front, Axis 211W.

4.4 Camera Setup

In order to use the camera model derived in section 4.2, surveyed points and camera orientation needed to be known. Using an Ashtech Z-Extreme™ surveying system both boxes were surveyed as well as a point on the runway. These points, as well as runway information, are recorded in Table 1 below.

Point	Latitude	Longitude	Height (feet)
Remote	35.4027227	-97.606510917	1193.256
Master	35.402724275	-97.6049097167	1196.113
Waypoint	35.3909211139	-97.6057126333	1228.015
17R Base	35.40595	-97.6057	1282 (physical)
17R (35L)	35.379	-97.6057	1263 (physical)
Center	35.402778	-97.605833	1198.567

Table 1: Surveyed Points

The internal parameters of the cameras needed to be calculated. A program was written which overlaid crosshairs, or known pixel points, on the video image. This program was used in two steps. First, each camera was pointed to the surveyed point, the waypoint, on the runway and aligned such that a known pixel point corresponded to said surveyed point. Second, a series of images were captured with aircraft on the runway using the same crosshair overlay. Once these images were recorded the image file, *.avi, was processed using NI's Vision Assistant. The images were played until both cameras aligned with the same image. The plane's dimensions were known and the distance from the cameras is known, being at the surveyed point, and using the

measurement tool the number of pixels was counted. A small amount of math was used to transform pixels to sensor image size.

The Axis 211W has a ¼" progressive scan RGB CMOS VGA sensor. This ¼" image sensor actually has a 3.2 mm width and 2.4 mm height. This corresponds to a 640 x 480 pixel sensor. Using these numbers produces

$$\frac{3.2\ mm}{640\ pixels} = .005\ \frac{mm}{pixels} \text{ for horizontal,}$$

$$\frac{2.4\ mm}{480\ pixels} = .005\ \frac{mm}{pixels} \text{ for vertical.}$$

It was straight forward to calculate the focal length. Using similar triangles, derived in Section 4.1, we can find the focal length by

$$f = Z * \frac{x}{X}, \tag{10}$$

where Z = distance from camera to plane, X = aircraft's dimension, and x = image size. Doing the calculations gives:

$$f_{Master} = 41.67\ mm,$$

$$f_{Remote} = 43.66\ mm.$$

The differences in focal length are due to human measurement error. Since these cameras have a manual focal and zoom adjustment, during setup the cameras were adjusted to a surveyed point on the runway. This small difference is taken into account in the camera software model which allows discrepancies in camera orientation.

The box placements and general pose of the cameras are shown in Figure 8 below. This photo shows the actual runway at Will Roger World Airport in which the boxes are placed. The black lines show the orientation of the principle axis of the cameras. The yellow lines show the theoretical (planned) placement of the cameras.

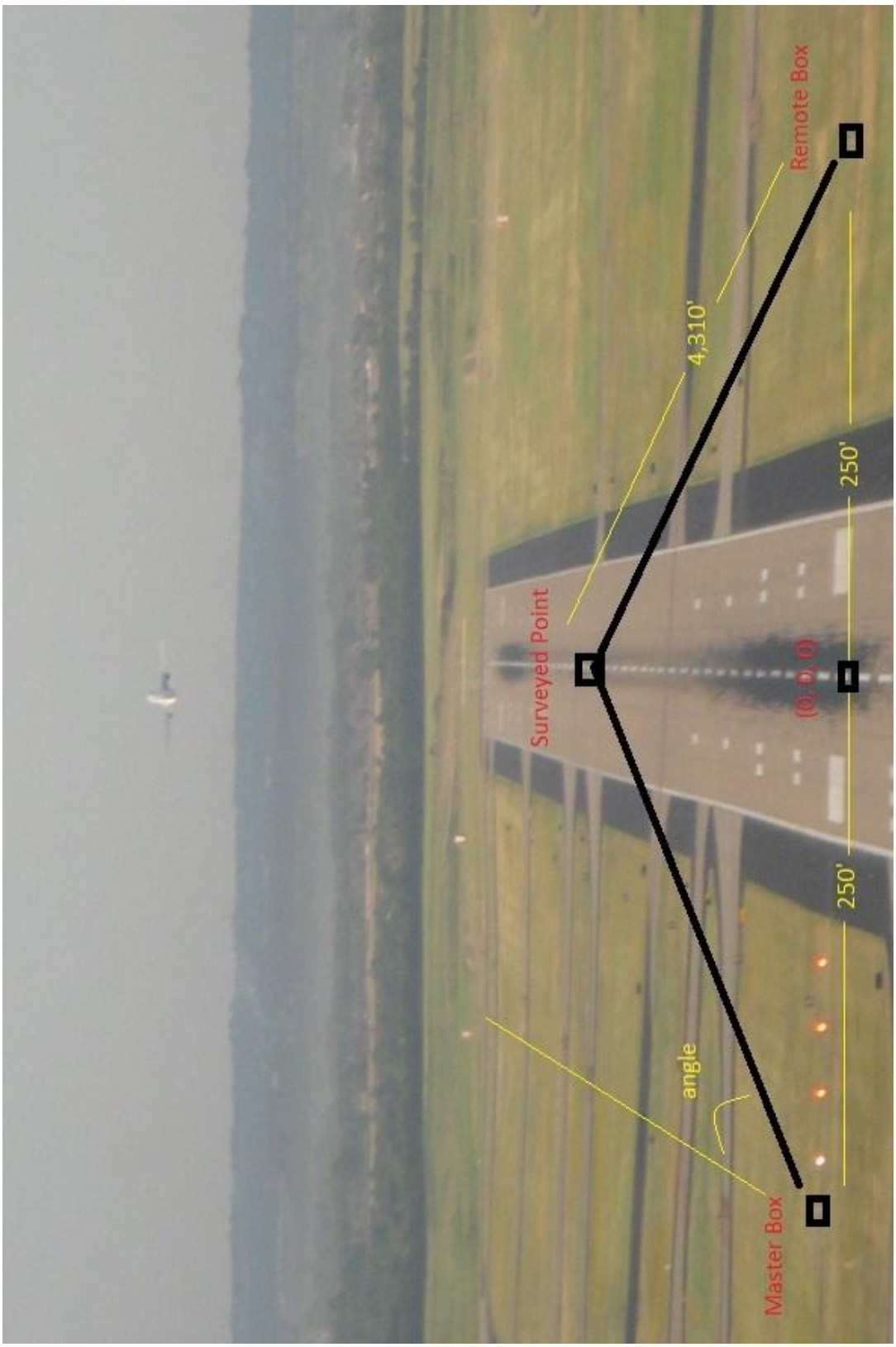


Figure 8 Camera Placement

Two datum transformations are necessary to convert from the GPS surveyed points into ENU coordinates. The first transformation is from GPS to earth centered earth fixed (ECEF) coordinates. The conversion from Latitude, Longitude, and Altitude (LLA) to ECEF (in meters) is given by

$$X = (N + h)\cos\varphi\cos\lambda, \tag{11}$$

$$Y = (N + h)\cos\varphi\sin\lambda, \tag{12}$$

$$Z = (\frac{b^2}{a^2}N + h)\sin\varphi, \tag{13}$$

where

$$\varphi = \text{latitude},$$

$$\lambda = \text{longitude},$$

$$h = \text{height above ellipsoid (meters)},$$

$$N = \text{Radius of Curvature (meters)}, \text{defined as}$$

$$= \frac{a}{\sqrt{1 - e^2 sin^2\varphi}},$$

$$a = 6378137,$$

$$b = a(1 - f),$$

$$f = \frac{1}{298.257223563},$$

$$e = \sqrt{\frac{a^2 - b^2}{b^2}}.$$

The second transformation is from ECEF to ENU coordinates. The relation between the local East, North, Up (ENU) coordinates and the (*x*,*y*,*z*) Earth-Centered-Earth-Fixed (ECEF) coordinates is illustrated in figure 9 on the next page.

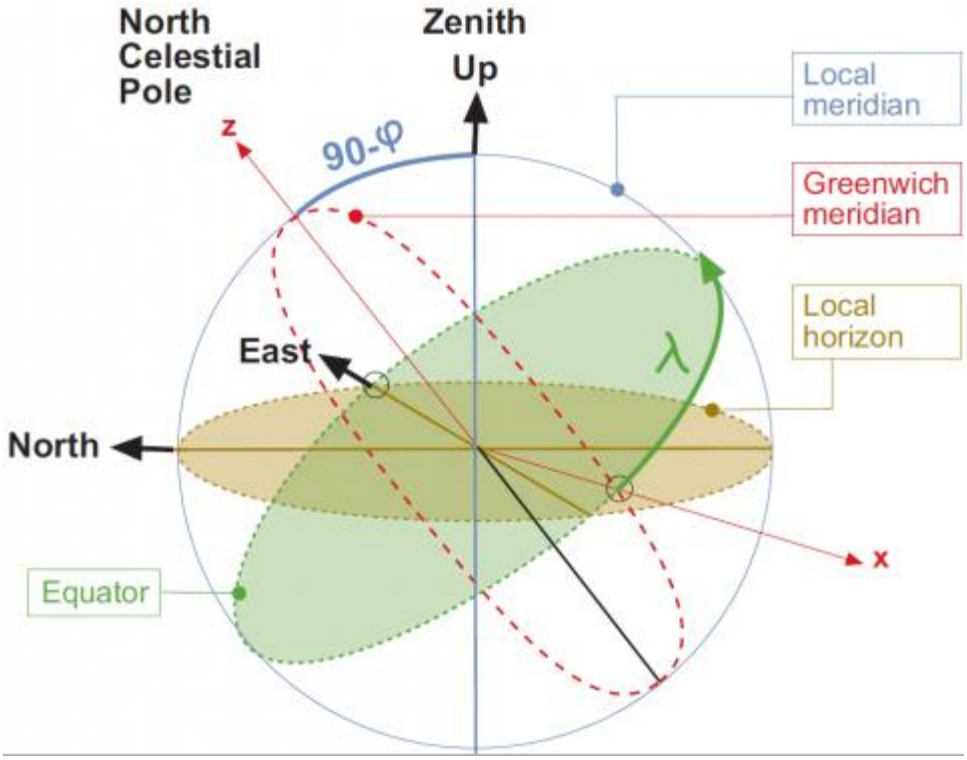


Figure 9 Transformations between ENU and ECEF Coordinates [32]

From Figure 9 it follows that the ENU coordinates can be transformed to the (*x*,*y*,*z*) ECEF by two rotations:

1. A clockwise rotation over east-axis by an angle 90 – φ to align the up-axis with the z-axis.
2. A clockwise rotation over the z-axis by an angle 9- + φ to align the east-axis with the x-axis.

This can expressed in mathematics as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{R}_3\left[-\left(\pi/2 + \lambda\right)\right]\boldsymbol{R}_1\left[-\left(\pi/2 - \varphi\right)\right]\begin{pmatrix} E \\ N \\ U \end{pmatrix}, \tag{14}$$

where

$$\boldsymbol{R}_1[\boldsymbol{\theta}] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix},$$

$$\boldsymbol{R}_2[\boldsymbol{\theta}] = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix},$$

$$\boldsymbol{R}_3[\boldsymbol{\theta}] = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which yields

$$E = (-\sin\lambda\,, \cos\lambda\,, 0)\begin{bmatrix} X_{ECEF} \\ Y_{ECEF} \\ Z_{ECEF} \end{bmatrix}, \tag{15}$$

$$N = (-\cos\lambda\sin\varphi\,, -\sin\lambda\sin\varphi\,, \cos\varphi)\begin{bmatrix} X_{ECEF} \\ Y_{ECEF} \\ Z_{ECEF} \end{bmatrix}, \tag{16}$$

$$U = (\cos\lambda\cos\varphi\,, \sin\lambda\cos\varphi\,, \sin\varphi)\begin{bmatrix} X_{ECEF} \\ Y_{ECEF} \\ Z_{ECEF} \end{bmatrix}. \tag{17}$$

Using the center point between the boxes and setting this point to (0, 0, 0) in ENU coordinates, a program was developed using the above equations to transform the surveyed points into ENU. This program is included in Appendix B1. The converted points are given in Table 2 below.

Point	East (meters)	North (meters)	Up (meters)
Center	0	0	0
Master	74.0564	1.17266	-0.56964
Remote	-71.4028	0.99789	-1.0626
Waypoint	1.11654	-1,308.435	-1.34586

Table 2 East North Up Coordinates

Using the transformed data from Table 2 the camera orientations can now be calculated. The total distance between the cameras is 145.46 meters or 477.2 feet. The baseline for the master box is 74.0564 meters whereas; the baseline for the remote box is -71.4028 meters. This is the first translation used in the translation matrix. The second translation is in the Up direction. From Table 2 it can be seen that the master box is 0.56964 meters below zero point while the remote box is 1.0626 meters below zero point. The final translation is in the North direction. From the table it can be seen that the master box is 1.17266 meters north of (0, 0, 0) and the remote box is 0.99789 meters north of (0, 0, 0). Using all these values the translational matrices for the master and remote boxes become

$$T_M = \begin{bmatrix} 1 & 0 & 0 & 74.0564 \\ 0 & 1 & 0 & 1.17266 \\ 0 & 0 & 1 & -0.56964 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_R = \begin{bmatrix} 1 & 0 & 0 & -71.4028 \\ 0 & 1 & 0 & 0.99789 \\ 0 & 0 & 1 & -1.0626 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

There are two different rotations that need to be taken into consideration. First is the horizontal rotation of the cameras. This rotation is viewed in the East – North plane and the rotations can be calculated from the surveyed waypoint and boxes. These rotations can be simply calculated using basic trigonometry according to

$$\tan\theta = \frac{opposite}{adjacent}.$$

The rotation away from parallel for the master box is given by

$$\theta_{M_H} = \tan^{-1}\frac{74.0564}{1308.435} = 3.23944 \text{ degrees}.$$

Whereas the rotation away from parallel for the remote box is

$$\theta_{R_H} = \tan^{-1}\frac{71.4028}{1308.435} = -3.123598 \text{ degrees.}$$

The next rotation is in the East – Up plane. This rotation deals with how the principle axis of each camera is aligned. Each camera was aligned with the surveyed point in both the horizontal and vertical direction with the pixel point (320, 440) in the vertical direction. The principle point for the master box is rotated by

$$\theta_{M_P} = -1.5575 \text{ degrees}$$

and the principle point for the remote box is rotated by

$$\theta_{R_P} = -1.5225 \text{ degrees.}$$

Using these values, the rotational matrices for the master and remote boxes become

$$R_{\theta_{Mx}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.998402 & -0.056509 & 0 \\ 0 & 0.056509 & 0.998402 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{\theta_{Mz}} = \begin{bmatrix} 0.999631 & 0.027180 & 0 & 0 \\ -0.02718 & 0.999631 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{\theta_{Rx}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.998514 & 0.054490 & 0 \\ 0 & -0.054490 & 0.998514 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{\theta_{Rz}} = \begin{bmatrix} 0.999647 & 0.02657 & 0 & 0 \\ -0.02657 & 0.999647 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Combining the rotational matrices to give only one rotation matrix for each box gives

$$R_{\theta_M} = \begin{bmatrix} 0.999631 & 0.02718 & 0 & 0 \\ -0.0273657 & 0.99803359 & -0.056509 & 0 \\ -0.00153591 & 0.05648815 & 0.998402 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_{\theta_R} = \begin{bmatrix} 0.999647 & 0.02657 & 0 & 0 \\ -0.02653052 & 0.99816152 & 0.05449 & 0 \\ 0.0014478 & -0.05447077 & 0.998515 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The complete camera matrix for each box can now be expressed in the complete matrix form. Combining all the matrices for the master side gives

$$\begin{bmatrix} u \\ v \\ s \end{bmatrix} = \begin{bmatrix} 0.999631 & 0.02718 & 0 & 74.060946 \\ -0.0273657 & 0.99893359 & -0.05609 & -0.8240613 \\ -0.00153591 & 0.05648815 & 0.998402 & -0.6162322 \\ -0.036858898 & 1.3556072 & 23.959731 & -14.78839 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

with

$$x_{im} = \frac{u}{s} \text{ and } y_{im} = \frac{v}{s}.$$

Combining all the matrices for the remote side gives

$$\begin{bmatrix} u \\ v \\ s \end{bmatrix} = \begin{bmatrix} 0.999647 & 0.02657 & 0 & -71.351081 \\ -0.02653052 & 0.99816152 & 0.05449 & 2.8325077 \\ 0.0014478 & -0.05447077 & 0.998515 & -1.2187548 \\ 0.03316079 & -1.2476127 & 22.870247 & -27.914678 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

with

$$x_{im} = \frac{u}{s} \text{ and } y_{im} = \frac{v}{s}.$$

These final results for the master box and remote box give a measured camera matrix that transforms 3-D points in the real world to pixels in the image. In order to generate 3-D tracks from pixels in an image, the two stereo frames must be rectified. This can be accomplished by the stereo frames being re-projected using a virtual parallel baseline camera system and then resampled by these virtual cameras [25]. The correspondence process can then proceed as if the stereo frames were taken from this virtual parallel baseline camera system.

Since it is difficult to execute the above procedure for calculation of the camera matrices with enough precision using distance and angle measurements to obtain an

accurate model, it is necessary to perform a camera calibration procedure discussed in Section 5.1.

4.5 Software Model Development

The language used to develop this project is NI's LabVIEW Professional Development System. This platform was chosen for numerous reasons. First, it includes a vision toolbox which allows different communication protocols to different camera manufacturers. This was essential to allow commercial off-the-shelf (COTS) cameras to be used with the system. LabVIEW features an extensive library of tools for signal processing, analysis and visualization, so scientists and engineers can use one common set of software tools that support a wide variety of COTS-based hardware to develop scientific computing applications in less time and with less effort [16].

Next, it is a graphical programming language that incorporates virtual instruments. Virtual instrumentation is the combination of user-defined software and modular hardware that implements custom systems ("virtual instruments") with components for acquisition, processing/analysis and presentation. Virtual instruments commonly leverage mainstream technologies, platforms, and standards such as the PC, Ethernet, GPIB, USB, IEEE 1394, PCI, PCI Express and others. Virtual instrumentation combines such technology with an application-specific selection of modular hardware for signal I/O, instrument control, connectivity and other tasks [16].

Virtual instrumentation software is user defined and focused on the needs of the application. For instance, researchers can build custom virtual instruments that can apply real-time mathematics for processing, analysis, and control involving online (live)

and/or offline (from a file / database) signal I/O. Using the virtual instrumentation approach, applied mathematics is combined with real-time measurements, which helps researchers reduce the time to discovery and, potentially, the time to market and/or time to commercialization of potential products and services that result from research and development (R&D) [16].

Numerous variations of the model have been developed. The initial implementation designed was the camera setup. This virtual instrument VI was created to access the three cameras and view each one separately. A sub-VI which overlays a cross hair pattern on the video image was created which allowed accurate positioning. The main VI, Camera Setup, block diagram can be seen in Appendix A, Figure A1 and the sub-VI, cross-hairs, can be seen in Figure A2.

The main algorithm design is a producer-consumer structure. The basic flowchart is shown below in Figure 10.

This design has multi-while loops which allow continuous camera acquisition with parallel processing. In the producer loop, camera data is continuously obtained from all three cameras. If for any reason a camera loses the link with the main computer, a sub-routine, Cam Re-initialization, is run to re-establish the link. This sub-VI uses a state machine model that if any error occurs the state re-initializes the camera sessions. The producer loop is also responsible for identifying when an airplane is taking off and to send collected data to the FTP server.

Once an aircraft is identified, the producer loop queues the data from the remote and master cameras and sends it to the consumers' loop which process the data. The

consumers' loop is responsible for tracking the airplane and recording the relevant data. The main Tracking VI block diagram can be seen in Appendix A, Figure A3.

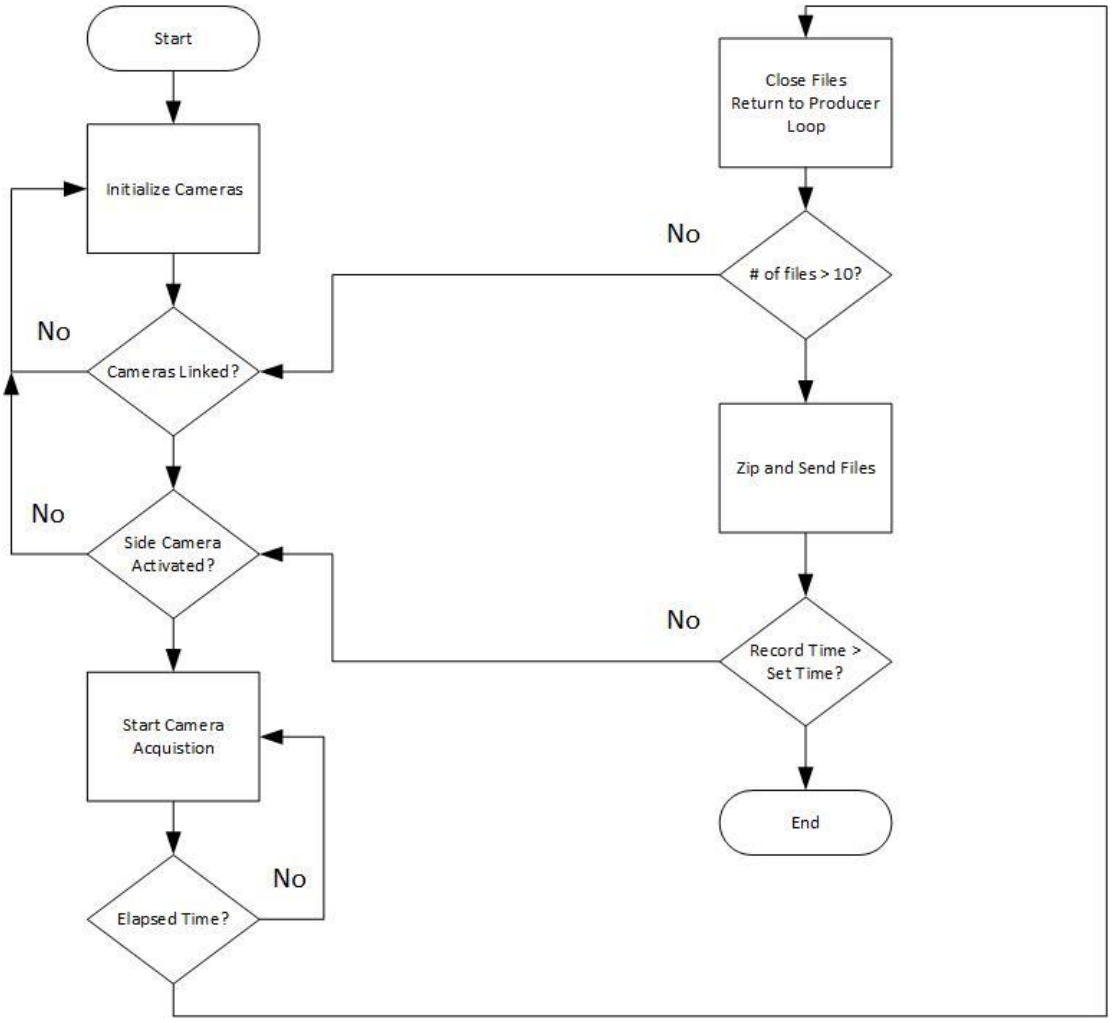


Figure 10 Flow Chart

The tracking algorithm can be broken down into individual steps. The first step in the producer loop is a color plane extraction. Extraction of a single plane helps with the speed of processing. The program is written so that any color plane desired can be extracted. Each color plane has been tested and at the current time the green color plane is used for the OKC implementation. This color plane provides the best pixel resolution

for tracking aircraft. Once the color plane is extracted both the master and remote images are queued and sent to the respective processing loop.

Once the images are in their respective loop they are first put into a shift register. This shift register allows processing of consecutive images, or if desired any variation of captured images. The camera's frame rate can be adjusted to produce any frame rate from 1 to 30 images per second. The shift register allows further processing of the frame rate. For instance, if the cameras are originally set to acquire 30 frames per second the program can be adjusted to use frames I_t and I_{t+1}, consecutive captured frames. Or by using the shift register the program can use I_t and I_{t+2} thereby reducing the effective frame rate to 15 frames per second. Being able to choose which frames to process helps with the next step.

Motion is detected by taking two different frames and subtracting them from each other. The result of this subtraction produces the difference between images, or motion. Using a variable frame rate helps with obtaining a clear image during subtraction due to the fact that during takeoff aircraft are slower and at long range there is less of a difference between frames. Using a fast frame rate would not always produce a subtraction image between consecutive frames assuming a straight-line between processed frames.

The next step is to perform a threshold on the differenced image. The pixel values are replaced if they fall within the upper value range and lower value range. This range is adjustable and the pixel values are replaced with 255. This basically turns the image into a binary image. This greatly reduces the amount of information that has to be processed.

Two basic morphological operations are performed next. These operations are performed to clear up noise and unwanted objects, i.e. birds. The first operation is called ‘open’ and can be enabled or disabled by the program. The purpose of the open operation is to remove small objects such as birds or grass movement. The second operation is called ‘close.’ The close operation is useful to remove holes and gaps that might occur in identifying the outline of the plane. The last four operations are performed in the main window tracking algorithm.

The main tracking algorithm uses a window tracking routine. Since it is known where the object to be tracked will originate, the window is initially defined along the runway on the side of the image respective to the camera orientation; master box image is along the right side. This window is the region of interest (ROI) and is where all processing is done. Once the unit is activated, only the ROI is processed to search for detection of motion and tracking of the aircraft. Once motion is detected a centroiding operation is applied finding the centroid of the object in motion. This centroid is used along with a two-state Kalman filter that predicts the ROI window for the next frame. The search window in the next frame is taken to be the predicted ROI with some overhead to allow for vertical and horizontal deviation during departure. The centroid (x and y pixels), along with the size of the object in number of pixels and precise UTC time stamp, is stored in a comma-separated-variable (CSV) file. The target is tracked until either the pixel count of the object falls below the threshold for tracking, or the target leaves the boundaries of either camera image. This is generally well beyond the typical visual segment and overlaps with the radar tracks as the target gains altitude.

Chapter 5 Data Analysis

Data is collected daily, zipped and sent to an FTP server. The main producer loop first checks the time. If the time is within the limits set by the program (adjustable), and an airplane is detected, it records 30 frames with the side camera and queues the master and remote camera data for the consumers' loop. Once the program has recorded data for 10 departures it calls a sub-VI called 'Zip and Send'. This sub-VI is composed of two other sub-VIs. The first VI is 'Zip Data' and the second is 'FTP download'. The block diagrams can be seen in Appendix A, Figure A4, Figure A5, and Figure A6, respectively.

The main data collected is recorded in a CSV file. This file records the centroid of the windowing function and the size of the tracked image. Both the master and remote CSV file are used together to generate the 3-D tracks. The raw data can be seen tracking an airplane in Figure 11 below.

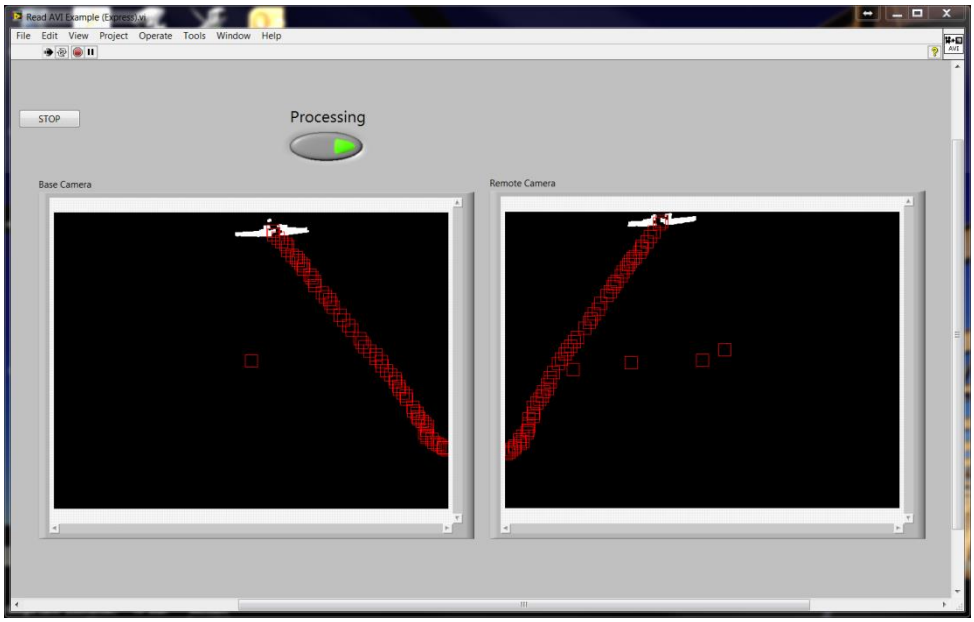


Figure 11 Raw Tracking Data

The aircraft can be seen as the white region in the images whereas; the window tracking can be viewed as the red rectangles moving as the airplane moves.

5.1 Calibration Flights

Before any real data analysis could be accomplished, calibration flights were conducted with a truth system. The truth data was harvested using a pair of Ashtech Z-Extreme™ receivers configured for kinematic differential post-processing. The Z-Extreme is a survey-grade, dual-frequency GPS receiver designed to achieve centimeter-accuracy positioning. It features 12 channels and built-in multipath mitigation. Its rechargeable battery allows it to run continuously for over 12 hours on one charge, while data is recorded to a removable PCMCIA memory card [15]. This truth system precisely records GPS data for an aircraft during departures. At the same time the binocular tracking system was recording the same flights. The truth system was set to record at a rate of 5 Hz. The plane used was a Seneca from OU. Even though this plane has a smaller cross-sectional area compared to the aircraft to be tracked, this plane was used to validate the system. Three different departures were recorded.

After raw (binary) GPS data is extracted from both receivers, it must be consolidated using a differential processing algorithm. Fortunately, there are many software packages that do this already. The current work made use of Waypoint Consulting's Grafnav® suite. It applies known base station coordinates, forward and reverse processing in time, and dual-frequency ionospheric correction to generate optimal position solutions. The processing options are highly configurable, but Grafnav

also includes presets for many applications. Flight data in this study is processed using the "airborne" settings. The three different calibration flights were flown and are shown in Figure 12 below.

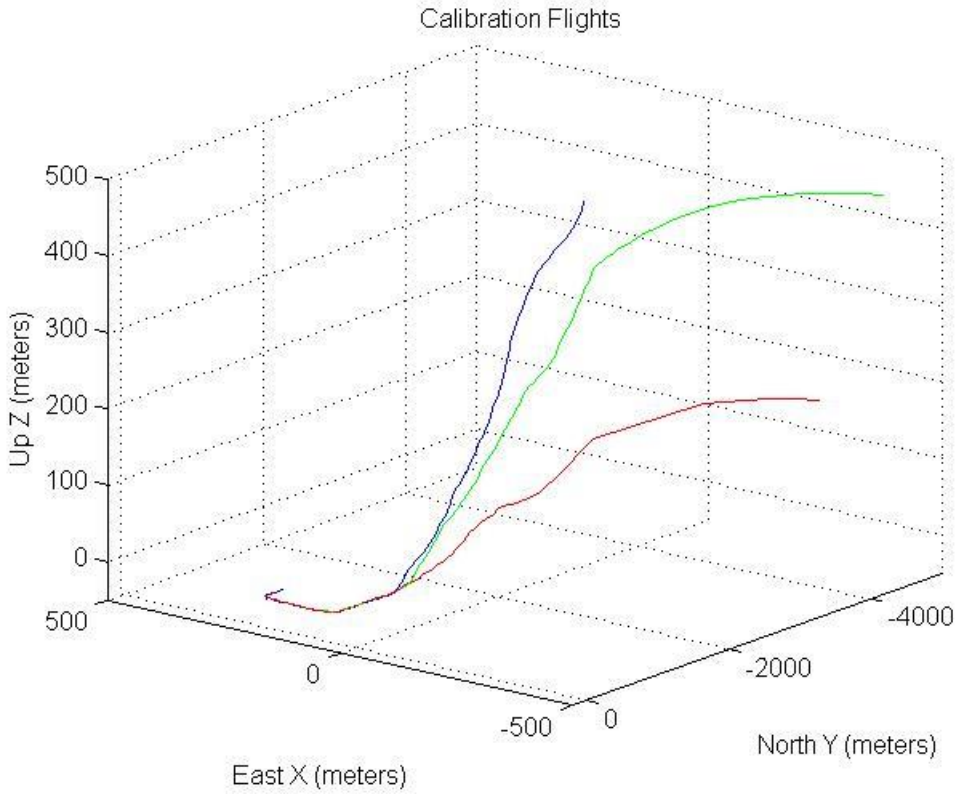


Figure 12 Calibration Flights

The model used to generate these tracks is shown in Appendix B2. The problem of camera calibration is to relate the locations of the pixels in the image array of a given camera to the real-valued points in the 3-D scene being imaged [14]. To calculate the camera matrices developed in section 4.4, a least squares solution using 3-D affine transformations was conducted.

5.2 Data Processing

The calibration flights were correlated to the flight data taken from the tracking system on the same day. One set of this data is shown in Appendix C1. This is necessary to relate the correspondence of the left image with the right image to recover the 3-D point. From this correspondence the parameters of the camera matrix can be recovered. Using the generalized camera model and eliminating the homogeneous scale factor s, two linear equations modeling the geometry of each imaging ray can be obtained according to

$$u_j = (c_{11} - c_{31}u_j)x_j + (c_{12} - c_{32}u_j)y_j + (c_{13} - c_{33}u_j)z_j + c_{14}, \quad (18)$$

$$v_j = (c_{21} - c_{31}v_j)x_j + (c_{22} - c_{32}v_j)y_j + (c_{23} - c_{33}v_j)z_j + c_{24}. \quad (19)$$

Separating the knowns from the unknowns into vectors gives

$$\begin{bmatrix} x_j & y_j & z_j & 1 & 0 & 0 & 0 & 0 & -x_j u_j & -y_j u_j & -z_j u_j \\ 0 & 0 & 0 & 0 & x_j & y_j & z_j & 1 & -x_j v_j & -y_j v_j & -z_j v_j \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{24} \\ c_{31} \\ c_{32} \\ c_{33} \end{bmatrix} = \begin{bmatrix} u_j \\ v_j \end{bmatrix}. \quad (20)$$

Each imaging ray gives two such equations and we obtain 2n linear equations from n calibration points. Representing these in classic matrix form gives

$$\boldsymbol{A}_{2nx11}\boldsymbol{x}_{11x1} \approx \boldsymbol{b}_{2nx1}. \quad (21)$$

Since there are 11 unknowns and 12 or more equations there is no vector of parameters **x** for which all the equations hold; a least-squares solution where the set of parameters such that the sum (over all equations) of the squared differences between the observed

coordinate and the coordinate predicted by the camera matrix is minimized [14], [25]. In order to do this the equation $\boldsymbol{x} = (\boldsymbol{A^t A})^{-1}\boldsymbol{A^t b}$ must be solved. Writing a MatLab script and importing the calibrated data a least-squares solution can be obtained. The MatLab script used is shown in Appendix B3. The calculated calibration matrix for the master box is given by

$$\begin{bmatrix} u \\ v \\ s \end{bmatrix} = \begin{bmatrix} -14.8128 & 0.0591 & 1{,}2353 & 185.0823 \\ -22.6011 & 0.0576 & 0.7885 & 267.0465 \\ -0.0730 & 0.0001 & -0.0136 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{22}$$

and the calculated calibration matrix for the remote box is:

$$\begin{bmatrix} u \\ v \\ s \end{bmatrix} = \begin{bmatrix} -31.7958 & 0.0051 & 5.6942 & -290.2434 \\ -34.4985 & 0.0348 & 8.3253 & 310.1422 \\ -0.1122 & 0.0001 & 0.0267 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{23}$$

Using the two calibrated camera models, the unknown 3-D point [x, y, z] can be computed from the two images [r_1, c_1] and [r_2, c_2]. Eliminating the homogeneous coordinates, the following imaging equations yield four linear equation in three unknowns [x, y, z]:

$$r_1 = (b_{11} - b_{31}r_1)x + (b_{12} - b_{32}r_1)y + (b_{13} - b_{33}r_1)z + b_{14}, \tag{24}$$

$$c_1 = (b_{21} - b_{31}c_1)x + (b_{22} - b_{32}c_1)y + (b_{23} - b_{33}c_1)z + b_{24}, \tag{25}$$

$$r_2 = (c_{11} - c_{31}r_2)x + (c_{12} - c_{32}r_2)y + (c_{13} - c_{33}r_2)z + c_{14}, \tag{26}$$

$$c_2 = (c_{21} - c_{31}c_2)x + (c_{22} - c_{32}c_2)y + (c_{23} - c_{33}c_2)z + c_{24}. \tag{27}$$

Any three of these four equations could be solved to obtain the point [x, y, z]; however, each subset of three equations would yield slightly different coordinates. To solve this problem the shortest distance between the two skew lines is measured along a

connecting line segment that is orthogonal to both lines. Each line is determined by the optical center of the camera and the image point.

Sample data from the tracking system is listed in Appendix C2. Using the generated camera matrix, 3-D flight paths can be generated from the CSV files. From a flight taken on 07/26/2013, a sample flight path is generated as shown in Figure 14 below.



Figure 13 July 26th Flight

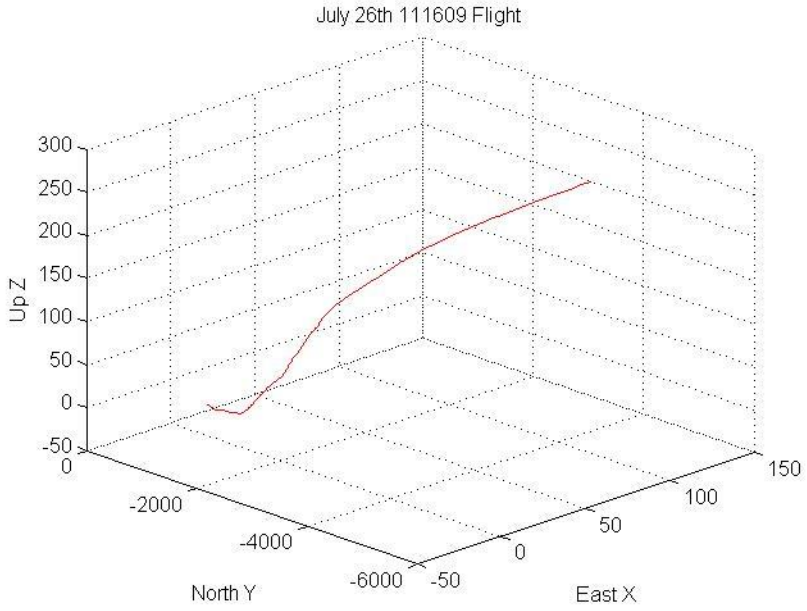


Figure 14 Generated Flight

Numerous generated flight paths are shown in Appendix D. Additional validation flights were again flown to verify the generated tracks.

5.3 Validation Flights

Validation flights were conducted in order to verify the correspondence of the master and remote cameras. These flights again used the same aircraft and truth system used to generate the calibration flights. Six flights were scheduled with different flight paths in order to completely cover the image planes. The first flight was to lift off, climb to 200 feet and then turn right. During the right turn the plane was continually instructed to climb an additional 200 feet. This was repeated two more times but for each different flight different attitudes were prepared. The second flight was to climb to 500 feet and the last was to climb to 1000 feet, all while executing right turns. The last three flights were to keep the same attitudes but to turn left after the initial attitude was obtained. The left turn validation flights are shown in Figure 15 on the following page.

These flights were correlated with the data taken from the binocular tracking system. Using the calibration matrix and the generated paths, the residual (error) was calculated. The residual error for the calibration matrices showed less than one pixel for the master box and under 3 pixels for the remote box. The results of these flights are given in the next section.

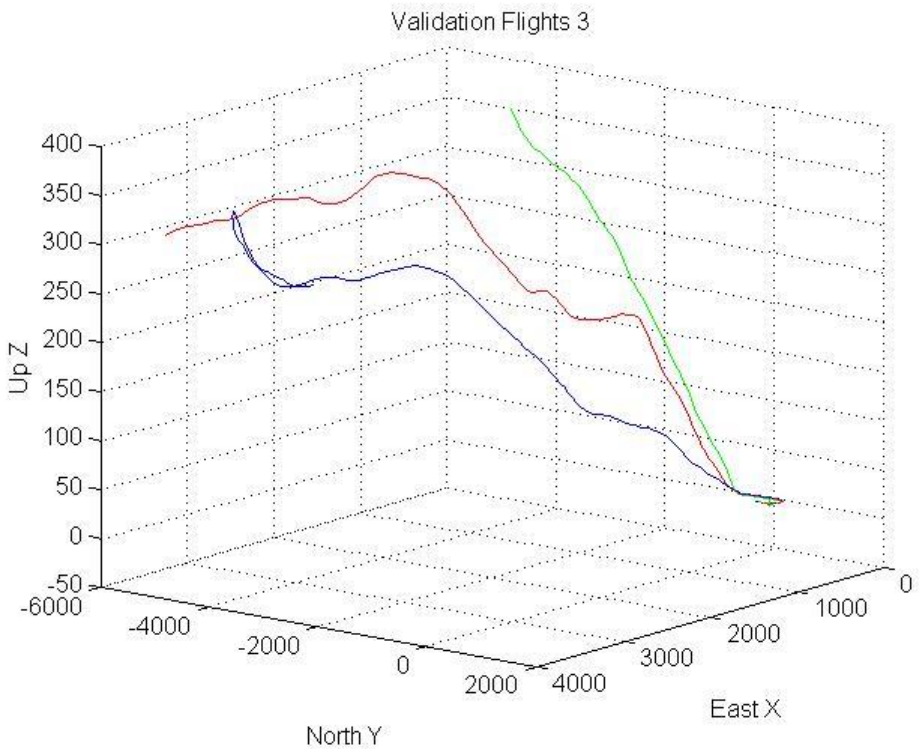


Figure 15 Validation Flights Left

5.4 Results

Each of the calibration flights (GPS data) was plotted against the generated flights from the calibration matrices. The GPS data is shown in blue whereas the generated flight is shown in red. Two sets of plots were generated for each flight in order to see differences. The first set is in the North-Up plane to identify any vertical difference. The second is in the East-North plane to identify tracking differences. Two sets of flights, one right turn and one left turn respectively, are shown in the Figures 16 - 19.

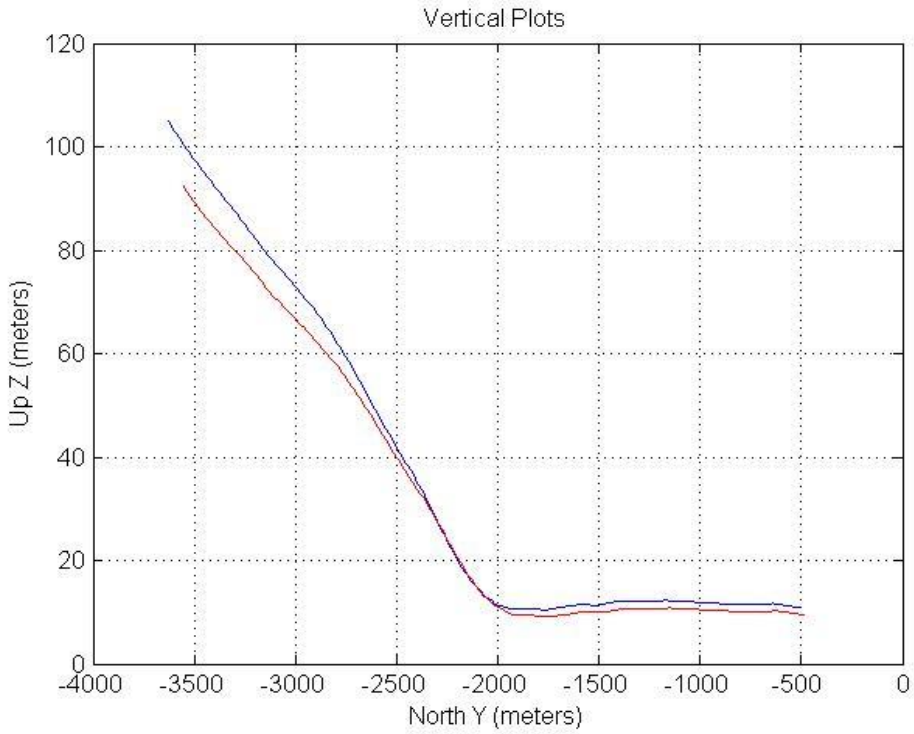


Figure 16 Vertical Plot Calibration Flight Right



Figure 17 Horizontal Plot Calibration Flight Right

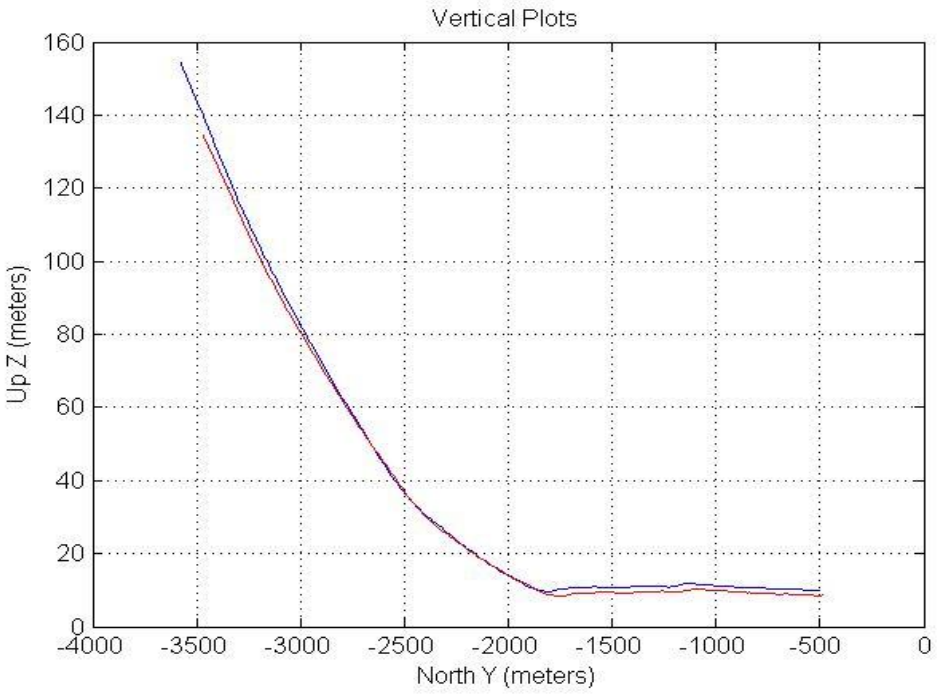


Figure 18 Vertical Plot Calibration Flight Left

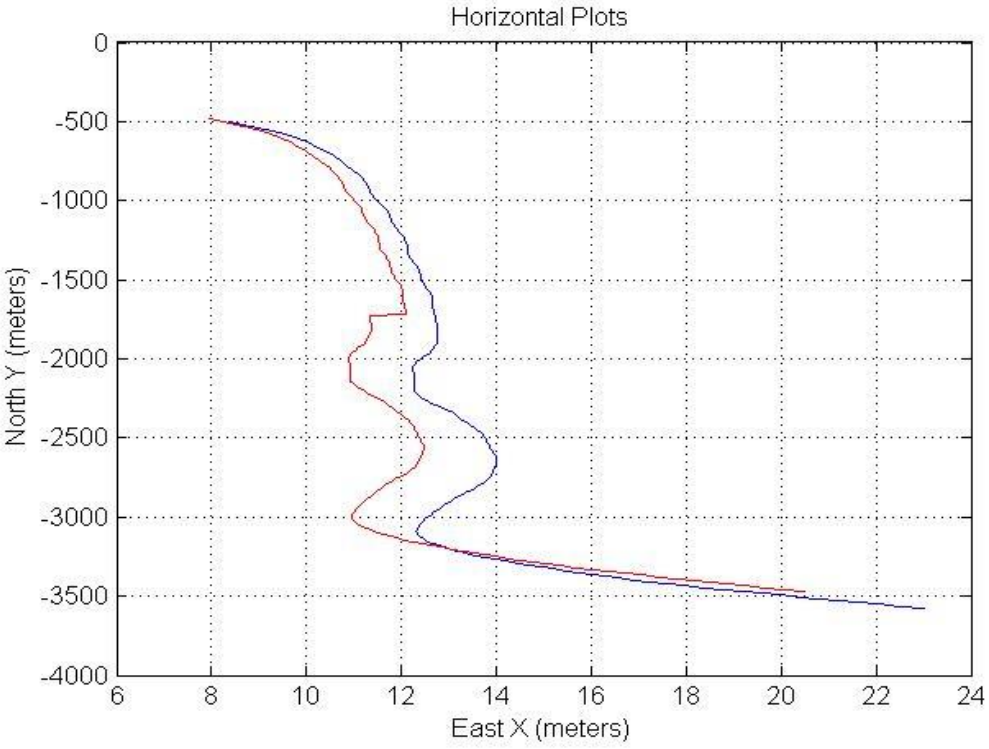


Figure 19 Horizontal Plot Calibration Flight Left

In order to get a better understanding of the above figures, both vertical error and horizontal error were calculated using all of the validation flights. From these the standard deviation (SD) was calculated for both values and listed in the Table 3below.

Flight	Horizontal SD (meters)	Vertical SD (meters)
Calibration Right	18.59	3.76
Calibration Left	27.61	5.52

Table 3 Standard Deviations for Calibration Flights

It can be seen that there is a greater SD in the 'Left Calibration' flights compared to the 'Right Calibration' flights. One cause for this difference could be that the calibration flights used two right turns and one left turn to 'map' the left and right pixels. Therefore when aircraft is in the left plane of the images, there is less rectification between pixel values and real world points causing a greater error. Focus and depth of field can also cause this type of error and needs to be considered.

An additional generated track, with side view, is shown in Figures 20 and 21.



Figure 20 November 9th Flight

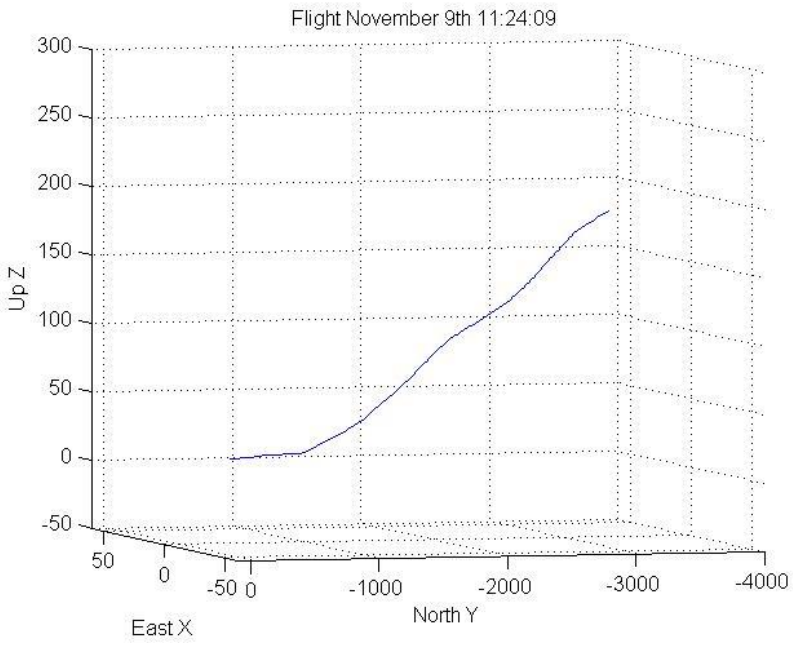


Figure 21 November 9th Generated Track

Using the generated matrices, numerous flights are plotted in the following Figure 22. Eleven flights are shown. Only a certain number of colors are available for generating plots.

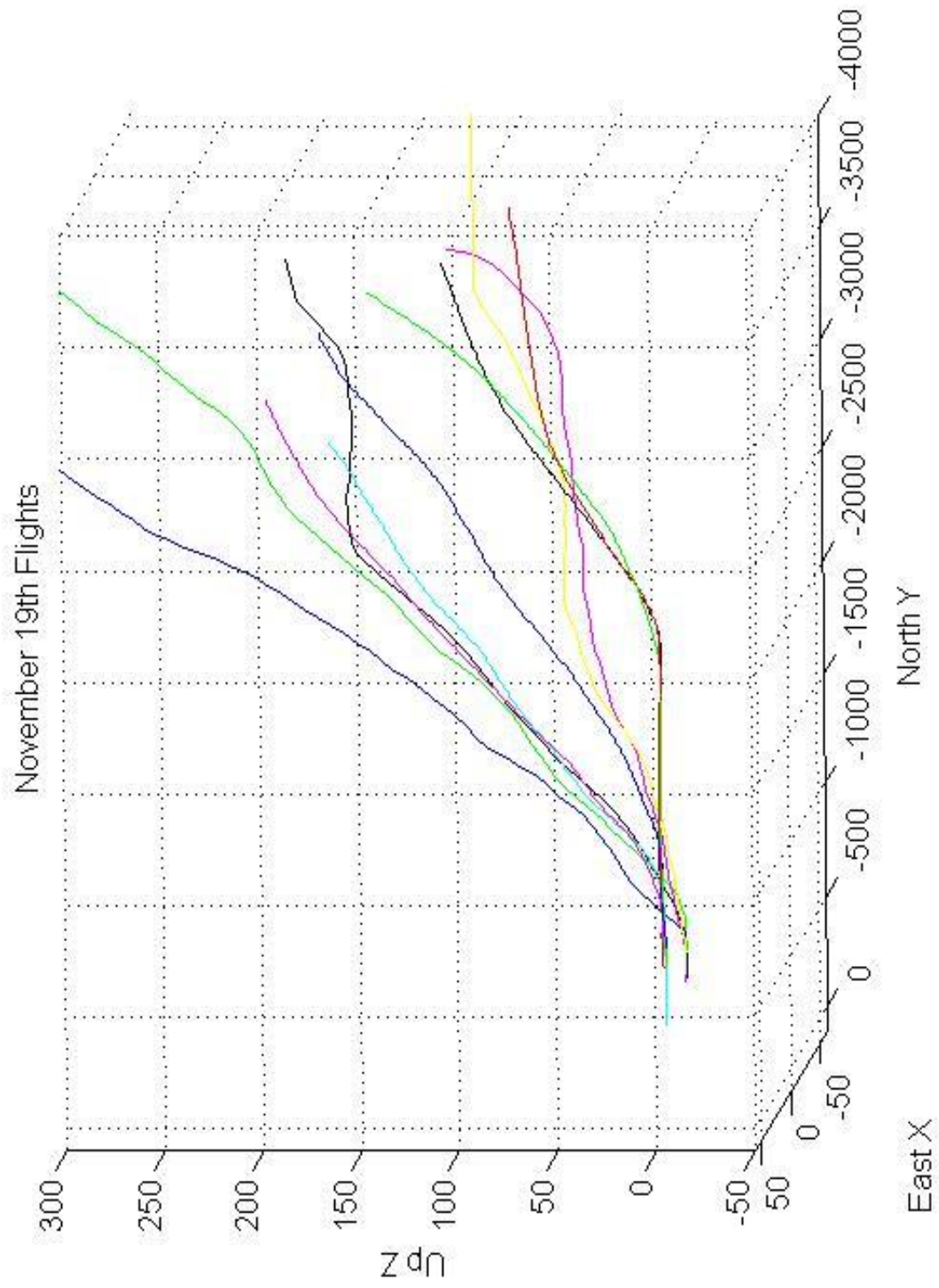


Figure 22 November 19th Flights

Chapter 6 Conclusions

The binocular tracking system developed for this dissertation is a unique and novel combination of software algorithms and COTS hardware. The system has proven its functional integrity and has successfully produced a large amount of departure data at the University of Oklahoma Westheimer Airpark (KOUN) and Oklahoma City Will Rogers World Airport (KOKC). The system has proven to be robust in different weather conditions since being installed in May 2013. The system survived the May 31st tornado that hit Will Rodgers World Airport and remains operational.

The system is new and novel and is capable of delivering a wide variety of departure data in order to help the FAA assess the magnitude of the problem and determine steps to align airport and TERPs procedures, while also mitigating obstacle clearance violations and thus the risk of departing aircraft to encounter an obstacle.

Future work in this field may include developing a system that will work on solar/battery power only. This would take away the dependence on airport power that the current system has and allow a more mobile system which could be deployed at different airports and runways. In addition a MatLab script can be added into the program which would allow all the track generation to be processed in the current program. As of the date of publication this is still accomplished via offline processing. Additional calibration flights covering more of the left and right images will allow for a better solution and reduce the system error. A single focal lens could be applied instead of the variable-focal lens used. This would ensure both cameras have the same focal length and reduce the cost.

References

1. C. Valeri, “Departure Procedure Climb Gradient and Calculating your Rate of Climb,” http://expertaviator.com/2011/02/03/departure-climb-gradient-and calculating-your-rate-of-climb.

2. FAA, “Aeronautical Information Manual Official Guide to Basic Flight Information and ATC Procedures,” Section 2. Departure Procedures, February 9, 2012.

3. A. Mian, “Realtime Visual Tracking of Aircrafts,” Computing: Techniques and Applications, 2008. DICTA '08.Digital Image, pp 351-356, 2008.

4. E. Turcco and K. Plakas, “Video tracking: A concise survey,” IEEE Journal of Oceanic Engineering, 31(2): 520-529, 2006.

5. A. Ali, H. Kausar, and M. Khan, “Automatic Visual Tracking and Firing System for Anti-Aircraft Maching Gun,” Proceedings of International Bhurban Conference on Applied Sciences and Technology, January 2009, pp 253-257.

6. P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” IEEE Computer Vision and Pattern Recognition, Vol.1, pp. 511-518, 2001.

7. L. Matthies, T, Kanade, and R. Szelisky, “Kalman filter based algorithms for estimating depth from image sequences,” International Journal of Computer Vision,” Vol. 3, pp. 209-236, 1989.

8. G. Manku, P. Jain, A. Aggarwal, and L. Kumar, “Object tracking using affine structure for point correspondence,” IEEE Computer Vision and Pattern Recognition, pp 704-709, 1997.

9. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” International Joint Conference on Artificial Intelligence, pp 674-679, 1981.

10. M. Isard and A. Blake, “Condensation – conditional density propagation for visual tracking,” International Journal of Computer Vision, 29(1): 5-28, 1998.

11. J. Barron, D. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” International Journal of Computer Vision, 12(1): 43-77, 1994.

12. R. Cipolla and M. Yamamoto, “Stereoscopic tracking of bodies in motion,” Image and Vision Computing, 8(1): 85-90, February 1990.

13. S. Moqqaddem, Y. Ruichek, R. Touahni and A. Sbihi, “A Spectral Clustering and Kalman Filtering Based Objects Detection and Tracking Using Stereo Vision with Linear Cameras,” 2011 IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, June 5-9, pp. 902-907, 2011.

14. L. Shapiro and G. Stockman, “Computer Vision,” ISBN 0-13-030796-3, Prentice-Hall, 2001.

15. Z-Xtreme Survey System. Santa Clara, CA : Thales Navigation, Inc.

16. National Instruments, “Advantages of Using LabVIEW in Academic Research,” February 01, 2012.

17. W. N. Klarquist and B. J. Super, “Patch-Based Stereo in a General Binocular Viewing Geometry,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 3, March 1997.

18. A. Bovik and W. N. Klarquist, “Adaptive Variable Baseline Stereo for Vergence Control,” Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, Vol. 3, pp. 1952-1959, April 1997.

19. A. Ali and S. M. Mirza, “Object Tracking using Correlation, Kalman Filter and Fast Means Shift Algorithms,” IEEE-ICET 2006 2nd International Conference on Emerging Technologies, Peshawar, Pakistan, pp. 174-178, November 2006.

20. L. Coutard and F. Chaumette, “Visual Detection and 3D Model-Based Tracking for Landing on an Aircraft Carrier,” 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 1746-1751, May 2011.

21. L. C. Wan, P. Sebastian, and Y. V. Voon, “Stereo Vision Tracking System,” 2009 Internatinal Conference on Future Computer and Communication, pp. 487-491, April 2009.

22. S. Lee, S. Yoon, H. J. Kim, and Y. Kim, “Wireless Stereo Vision System Development for Rotary-Wing UAV Guidance and Control,” 3rd International Conference on Sensing Technology, Tainan, Taiwan, pp. 168-173, November 2008.

23. M. Agrawal and L. S. Davis, “Camera Calibration using Spheres: A Semi-Definite Programming Approach,” Proceedings of the Ninth IEEE International Conference on Computer Vision, Vol. 2, pp. 782-789, 2003.

24. FAA, “8260.3B - United States Standard for Terminal Instrument Procedures (TERPS),” http://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.information/documentid/11698.

25. W. N., Klarquist, “FOVEA; a foveated vergent active stereo vision system for dynamic three-dimensional scene recovery,” Dissertation, University of Texas at Austin, May 1996.

26. Joseph P., Havlicek, "ECE 5273 Digital Image Processing," School of Electrical and Computer Engineering, University of Oklahoma, Spring 2005.

27. L. Zheng, Y. Chang, and Z. Li, "A Study of 3D Feature Tracking and Localization using a Stereo Vision System," Computer Symposium (ICS), 2010 International, pp. 402 – 407, 2010.

28. Pinhole Camera Model, http://docs.mitk.org/nightly-qt4/classmitk_1_1ToFDistanceImageToSurfaceFilter.html

29. Perspective Projection, http://photo.stackexchange.com/questions/41273/how-to-calculate-the-fov-in-degrees-from-focal-length-or-distance

30. Parallel Optical Axes, http://www.cs.ru.nl/~ths/rt2/col/h2/2fundENG.html

31. General Vergent Stereo Setup, http://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/773-GG/lectA-773.htm

32. Transformation between ECEF and ENU, http://www.navipedia.net/index.php/Transformations_between_ECEF_and_ENU_coordinates

Appendix A: LabVIEW Code

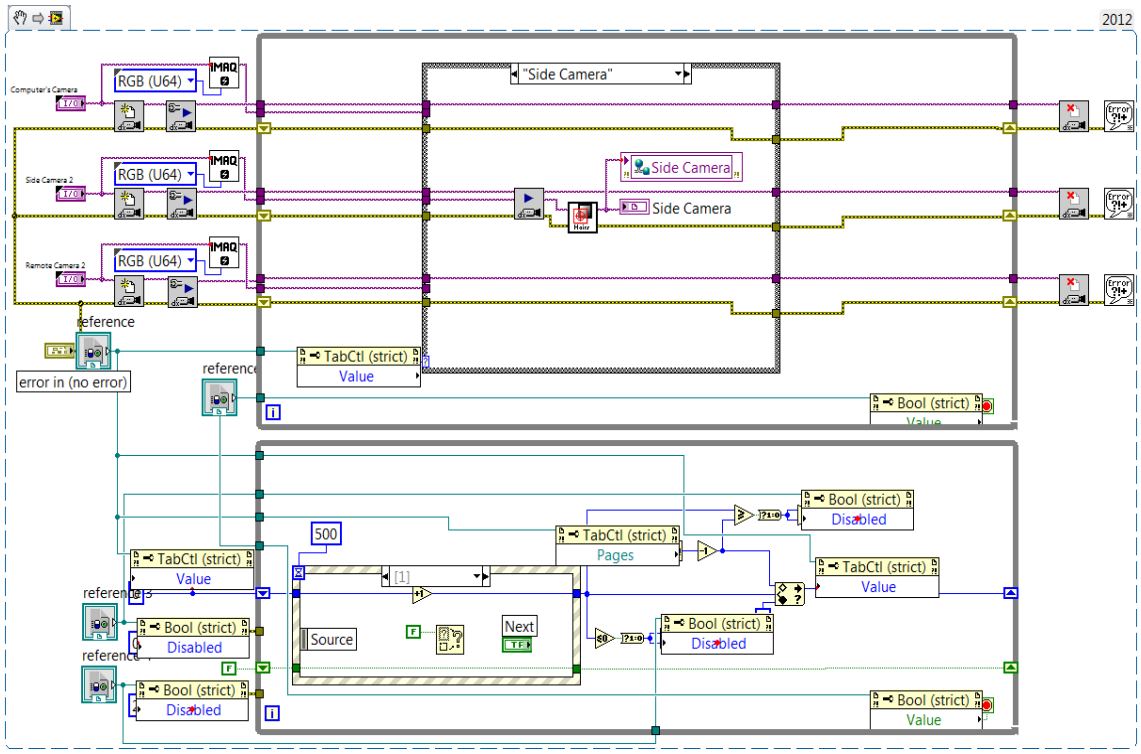


Figure A 23 Camera Setup

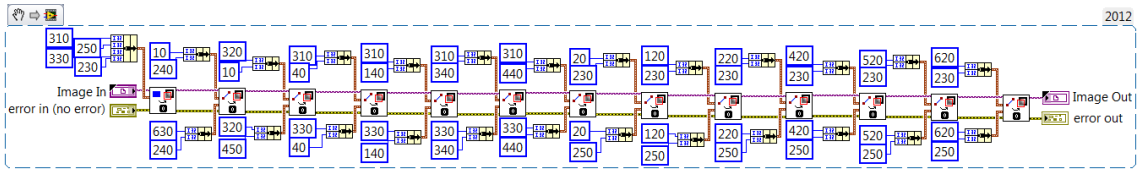


Figure A 24 Cross Hairs

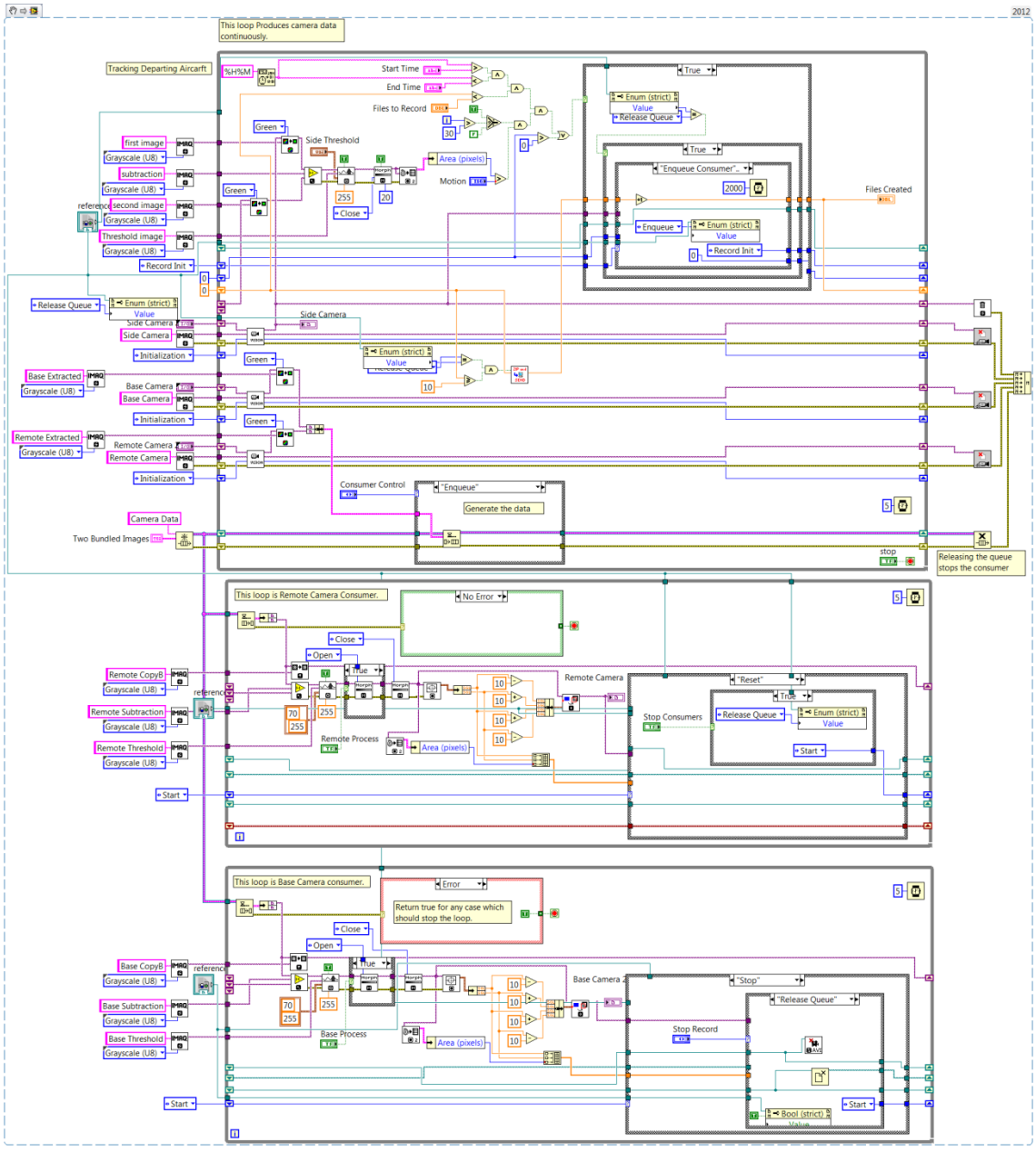


Figure A 25 Tracking

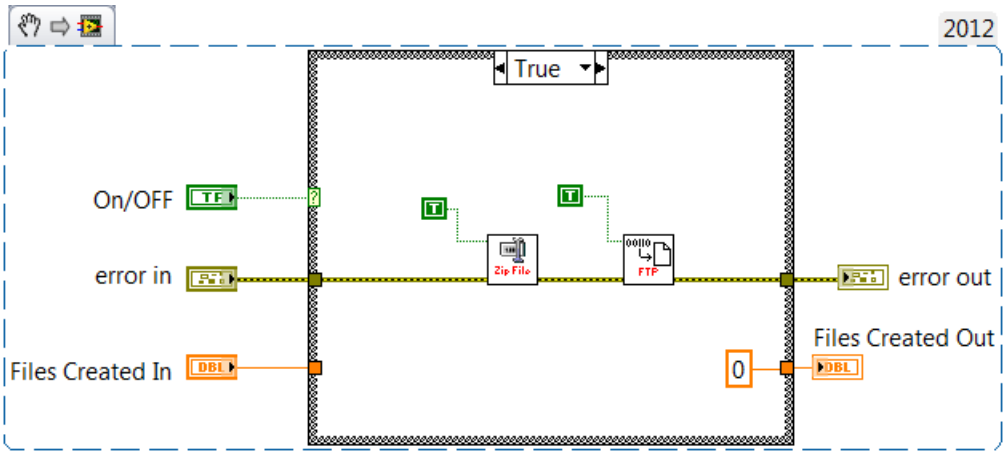


Figure A 26 Zip and Send

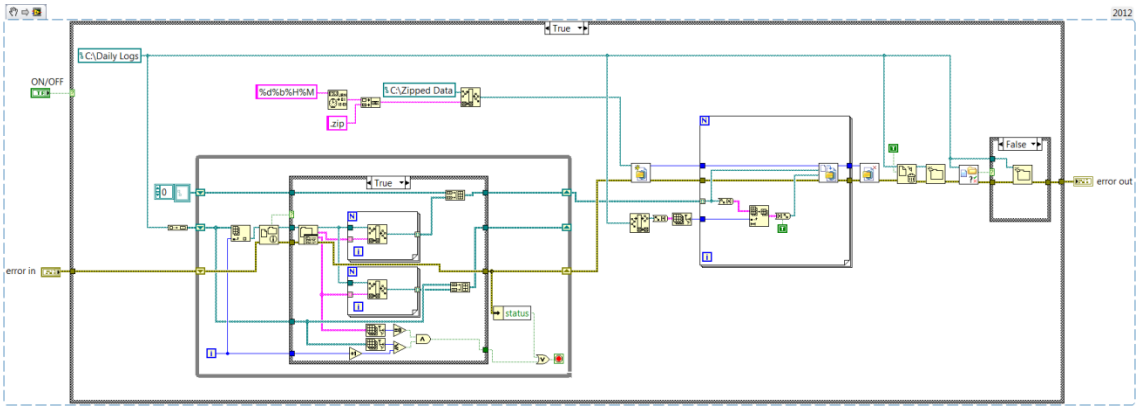


Figure A 27 Zip Data

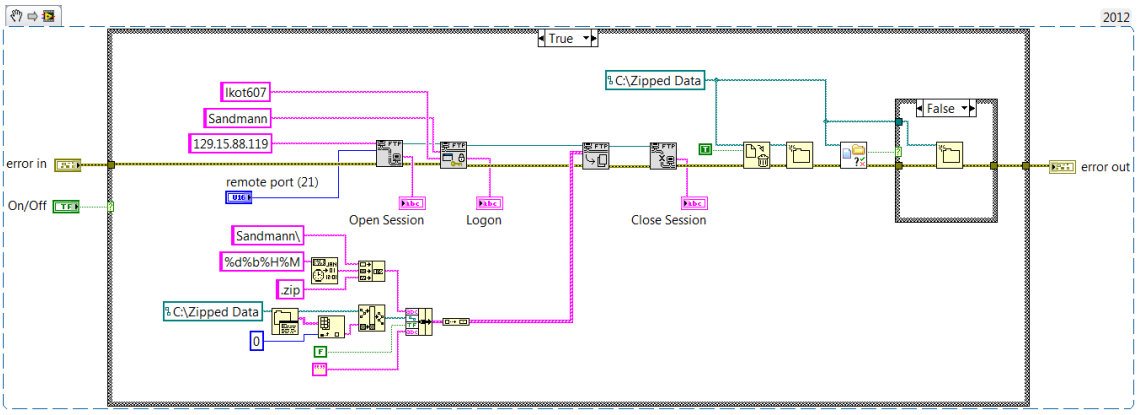


Figure A 28 FTP Download

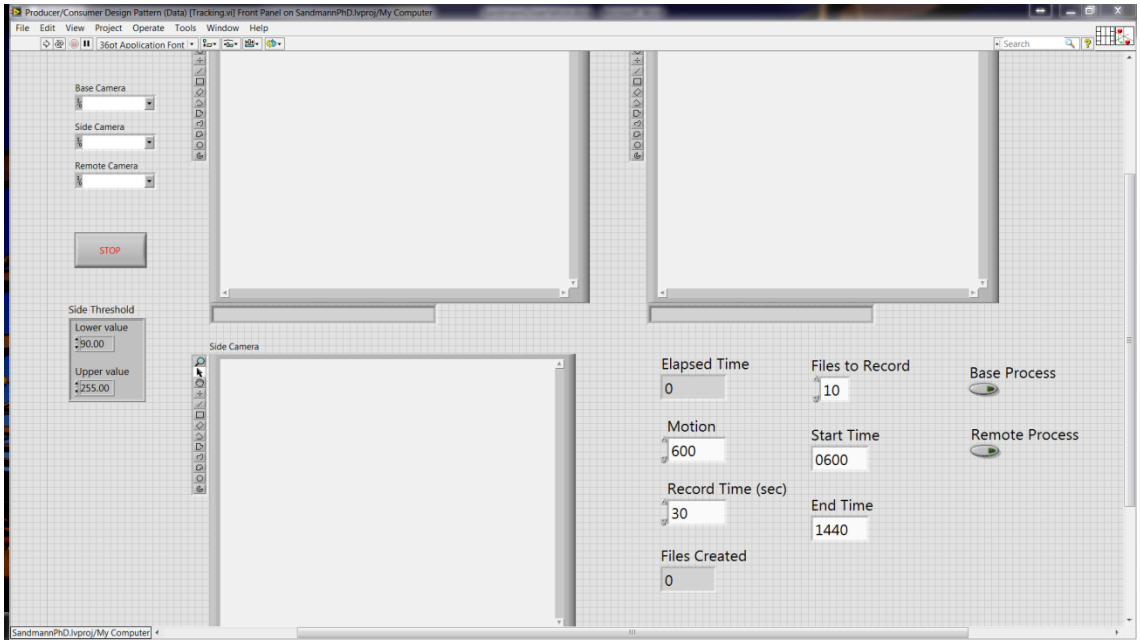


Figure A 29 Front Panel

Appendix B MatLab Code

Appendix B1 Datum Transformations

```
% This script converts single LLA point to ENU
% Reference point can be entered
LOC = input('Enter LLA coordinates to convert: ');
% Assume LOC is a 1x3 vector with:
%
% [Latitude Longitude Altitude]
%
% Latitude and Longitude are in decimal degrees
% Altitude is in meters
%
% Resulting ecef is in meters

LOC(2) = mod(LOC(2),360);
% make sure longitude is defined for a
%complete circle and not for +/- 180 degrees

a = 6378137;  % WGS84 Datum
f = 1/298.257223563; %WGS84 Datum
N = a/sqrt(1 - f*(2-f)*sind(LOC(1))^2);
ecef(1) = (N+LOC(3))*cosd(LOC(1))*cosd(LOC(2));
ecef(2) = (N+LOC(3))*cosd(LOC(1))*sind(LOC(2));
ecef(3) = ((1-f)^2*N+LOC(3))*sind(LOC(1));
%
xyz_dat = ecef;
%
ref_point = input('Enter ECEF coordinates of reference point: ');
% This point is 17R end of runway
%ref_point = [-0688881.017237612 -5158992.224638555 3674894.311855159];
% ref_point LLA = [35.405950 -97.605723 390.7536]
x = ref_point(1);
y = ref_point(2);
z = ref_point(3);
lambda = atan2(y,x);  %Find the longitude
ex2 = (2-f).*f./((1-f).^2);
c = a.*sqrt(1+ex2);
phi = atan(z./((sqrt(x.^2+y.^2).*(1-(2-f)).*f)));  %Find the latitude

h = 0.1; oldh = 0;  %Use the following method to find the optimum height
while abs(h-oldh) > 1.e-12
   oldh = h;
   N = c./sqrt(1+ex2.*cos(phi).^2);
   phi = atan(z./((sqrt(x.^2+y.^2).*(1-(2-f).*f.*N./(N+h)))));
```

```
  h = sqrt(x.^2+y.^2)./cos(phi)-N;
end

phi = phi.*180/pi;
%b = zeros(1,3);
%b(1) = fix(phi);
%b(2) = fix(rem(phi,b(1,1))*60);
%b(3) = (phi-b(1,1)-b(1,2)/60)*3600;
lambda = lambda*180/pi;
%l = zeros(1,3);
%l(1) = fix(lambda);
%l(2) = fix(rem(lambda,l(1,1))*60);
%l(3) = (lambda-l(1,1)-l(1,2)/60)*3600;

% This is the rotation matrix
F = [-sind(lambda) -sind(phi)*cosd(lambda) cosd(phi)*cosd(lambda);
cosd(lambda) -sind(phi)*sind(lambda) cosd(phi)*sind(lambda); 0 cosd(phi) sind(phi)];

[rows cols] = size(xyz_dat);
if rows > 1
   dp = xyz_dat - repmat(ref_point,[rows 1]);
else
   dp = xyz_dat - ref_point;
end;

east = dp*F(:,1);
north = dp*F(:,2);
up = dp*F(:,3);
ENU = [east north up];
```

Appendix B2 Validation Flights

```
% This script takes LLA and converts to ENU



fid = fopen('D:\Ph.D\Validation 2\Takeoff1 LLA.txt');
mydata1 = textscan(fid, '%f %f %f');
fclose(fid);
fid = fopen('D:\Ph.D\Validation 2\Takeoff2 LLA.txt');
mydata2 = textscan(fid, '%f %f %f');
fclose(fid);
fid = fopen('D:\Ph.D\Validation 2\Takeoff3 LLA.txt');
mydata3 = textscan(fid, '%f %f %f');
fclose(fid);
% Combine Matrices
X1 = mydata1{1}; % should be height in meters
Y1 = mydata1{2}; % should be latitude in decimal degrees
Z1 = mydata1{3}; % should be longitude in decimal degrees
X2 = mydata2{1}; % should be height in meters
Y2 = mydata2{2}; % should be latitude in decimal degrees
Z2 = mydata2{3}; % should be longitude in decimal degrees
X3 = mydata3{1}; % should be height in meters
Y3 = mydata3{2}; % should be latitude in decimal degrees
Z3 = mydata3{3}; % should be longitude in decimal degrees

LOC1 = horzcat(Y1,Z1,X1);
LOC2 = horzcat(Y2,Z2,X2);
LOC3 = horzcat(Y3,Z3,X3);

a = 6378137;  % % Define the ellipsoid based on WGS-84
f = 1/298.257223563; %WGS84 Datum
% Assume LOC is a nx3 matrix with:
%
% [Latitude Longitude Altitude]
%
% Latitude and Longitude are in decimal degrees
% Altitude is in meters
%
% Resulting ecef is in meters

% First validation flight
[rows,cols] = size(LOC1);

for i = 1:1:rows
% make sure longitude is defined for a complete circle and not for +/- 180 degrees
```

```
   LOC1(i,2) = mod(LOC1(i,2),360);
   %
   N = a/sqrt(1 - f*(2-f)*sind(LOC1(i,1))^2);
   ecef1(i,1) = (N+LOC1(i,3))*cosd(LOC1(i,1))*cosd(LOC1(i,2));
   ecef1(i,2) = (N+LOC1(i,3))*cosd(LOC1(i,1))*sind(LOC1(i,2));
   ecef1(i,3) = ((1-f)^2*N+LOC1(i,3))*sind(LOC1(i,1));
end

xyz_dat1 = ecef1;

% Second validation flight
[rows,cols] = size(LOC2);

for i = 1:1:rows
% make sure longitude is defined for a complete circle and not for +/- 180 degrees
   LOC2(i,2) = mod(LOC2(i,2),360);
   %
   N = a/sqrt(1 - f*(2-f)*sind(LOC2(i,1))^2);
   ecef2(i,1) = (N+LOC2(i,3))*cosd(LOC2(i,1))*cosd(LOC2(i,2));
   ecef2(i,2) = (N+LOC2(i,3))*cosd(LOC2(i,1))*sind(LOC2(i,2));
   ecef2(i,3) = ((1-f)^2*N+LOC2(i,3))*sind(LOC2(i,1));
end

xyz_dat2 = ecef2;

% Third validation flight
[rows,cols] = size(LOC3);

for i = 1:1:rows
% make sure longitude is defined for a complete circle and not for +/- 180 degrees
   LOC3(i,2) = mod(LOC3(i,2),360);
   %
   N = a/sqrt(1 - f*(2-f)*sind(LOC3(i,1))^2);
   ecef3(i,1) = (N+LOC3(i,3))*cosd(LOC3(i,1))*cosd(LOC3(i,2));
   ecef3(i,2) = (N+LOC3(i,3))*cosd(LOC3(i,1))*sind(LOC3(i,2));
   ecef3(i,3) = ((1-f)^2*N+LOC3(i,3))*sind(LOC3(i,1));
end

xyz_dat3 = ecef3;

%ref_point = input('Enter ECEF coordinates of reference point: ');
ref_point = [-0688881.017237612 -5158992.224638555 3674894.311855159];
% ref_point LLA = [35.405950 -97.605723 390.7536]
x = ref_point(1);
y = ref_point(2);
z = ref_point(3);
```

```
lambda = atan2(y,x);  %Find the longitude
ex2 = (2-f).*f./((1-f).^2);
c = a.*sqrt(1+ex2);
phi = atan(z./((sqrt(x.^2+y.^2).*(1-(2-f)).*f)));  %Find the latitude

h = 0.1; oldh = 0;  %Use the following method to find the optimum height
while abs(h-oldh) > 1.e-12
   oldh = h;
   N = c./sqrt(1+ex2.*cos(phi).^2);
   phi = atan(z./((sqrt(x.^2+y.^2).*(1-(2-f).*f.*N./(N+h)))));
   h = sqrt(x.^2+y.^2)./cos(phi)-N;
end

phi = phi.*180/pi;

lambda = lambda*180/pi;

% This is the rotation matrix
F = [-sind(lambda) -sind(phi)*cosd(lambda) cosd(phi)*cosd(lambda); cosd(lambda) -
sind(phi)*sind(lambda) cosd(phi)*sind(lambda); 0 cosd(phi) sind(phi)];
%
%
[rows cols] = size(xyz_dat1);
if rows > 1
    dp = xyz_dat1 - repmat(ref_point,[rows 1]);
else
    dp = xyz_dat1 - ref_point;
end;

east1 = dp*F(:,1);
north1 = dp*F(:,2);
up1 = dp*F(:,3);
ENU1 = [east1 north1 up1];
%
%
[rows cols] = size(xyz_dat2);
if rows > 1
    dp = xyz_dat2 - repmat(ref_point,[rows 1]);
else
    dp = xyz_dat2 - ref_point;
end;

east2 = dp*F(:,1);
north2 = dp*F(:,2);
up2 = dp*F(:,3);
```

```
ENU2 = [east2 north2 up2];
%
%
[rows cols] = size(xyz_dat3);
if rows > 1
   dp = xyz_dat3 - repmat(ref_point,[rows 1]);
else
   dp = xyz_dat3 - ref_point;
end;

east3 = dp*F(:,1);
north3 = dp*F(:,2);
up3 = dp*F(:,3);
ENU3 = [east3 north3 up3];
%
%
% Now plot 3D track

plot3(ENU1(1:1840,1),ENU1(1:1840,2),ENU1(1:1840,3),'b');
hold on
plot3(ENU2(1:1450,1),ENU2(1:1450,2),ENU2(1:1450,3),'r');
hold on
plot3(ENU3(1:1400,1),ENU3(1:1400,2),ENU3(1:1400,3),'g');
xlabel('East X')
ylabel('North Y')
zlabel('Up Z')
title('Validation Flights')
grid on
```

Appendix B3 Calibration Matrix Script

```
% This script converts calibration data and image data into
% camera matrix
%
mydata1 = csvread('D:\Ph.D\Validation Flights\Takeoff 1\17JulwGPS2.csv');
%mydata1 = csvread('D:\Ph.D\Validation Flights\Takeoff 1\Test.csv');
%
%fid = fopen('D:\Ph.D\Validation3\Flight3_1 LLA.txt');
%mydata3 = textscan(fid, '%f %f %f');
%fclose(fid);
%
% Construct Ax = b and solve for x = (A'A)^-1A'b
%
[r,c]=size(mydata1);
%
% First do Master box
%
bcounter = 1;
for i = 1:1:r

   if mydata1(i,3) == 0

   else
      b(bcounter,1)=mydata1(i,1);
      b(bcounter+1,1)=mydata1(i,2);

      A(bcounter,1)=mydata1(i,10);
      A(bcounter,2)=mydata1(i,11);
      A(bcounter,3)=mydata1(i,12);
      A(bcounter,4)= 1;
      A(bcounter,5)= 0;
      A(bcounter,6)= 0;
      A(bcounter,7)= 0;
      A(bcounter,8)= 0;
      A(bcounter,9)= -mydata1(i,10)*mydata1(i,1);
      A(bcounter,10)= -mydata1(i,11)*mydata1(i,1);
      A(bcounter,11)= -mydata1(i,12)*mydata1(i,1);

      A(bcounter+1,1)= 0;
      A(bcounter+1,2)= 0;
      A(bcounter+1,3)= 0;
      A(bcounter+1,4)= 0;
      A(bcounter+1,5)= mydata1(i,10);
```

```
        A(bcounter+1,6)= mydata1(i,11);
        A(bcounter+1,7)= mydata1(i,12);
        A(bcounter+1,8)= 1;
        A(bcounter+1,9)= -mydata1(i,10)*mydata1(i,2);
        A(bcounter+1,10)= -mydata1(i,11)*mydata1(i,2);
        A(bcounter+1,11)= -mydata1(i,12)*mydata1(i,2);

        bcounter = bcounter+2;
    end

end
%
% Now solve the equations
%
Xmaster = A\b;
%
% residuals of the computation
%
Rmaster = b-A*Xmaster;
Cmaster = [Xmaster(1:4)';Xmaster(5:8)';Xmaster(9:11)' 1];
%%
%Now do Remote box
%
loop = 1;
while loop ==1
    loop = input('Do you want to run remote box? (Y = 0) ');
end
%
bcounter = 1;
for i = 1:1:r

    if mydata1(i,6) == 0

    else
        b2(bcounter,1)=mydata1(i,4);
        b2(bcounter+1,1)=mydata1(i,5);

        A2(bcounter,1)=mydata1(i,10);
        A2(bcounter,2)=mydata1(i,11);
        A2(bcounter,3)=mydata1(i,12);
        A2(bcounter,4)= 1;
        A2(bcounter,5)= 0;
        A2(bcounter,6)= 0;
        A2(bcounter,7)= 0;
        A2(bcounter,8)= 0;
        A2(bcounter,9)= -mydata1(i,10)*mydata1(i,4);
```

```
        A2(bcounter,10)= -mydata1(i,11)*mydata1(i,4);
        A2(bcounter,11)= -mydata1(i,12)*mydata1(i,4);

        A2(bcounter+1,1)= 0;
        A2(bcounter+1,2)= 0;
        A2(bcounter+1,3)= 0;
        A2(bcounter+1,4)= 0;
        A2(bcounter+1,5)= mydata1(i,10);
        A2(bcounter+1,6)= mydata1(i,11);
        A2(bcounter+1,7)= mydata1(i,12);
        A2(bcounter+1,8)= 1;
        A2(bcounter+1,9)= -mydata1(i,10)*mydata1(i,5);
        A2(bcounter+1,10)= -mydata1(i,11)*mydata1(i,5);
        A2(bcounter+1,11)= -mydata1(i,12)*mydata1(i,5);

        bcounter = bcounter+2;
    end

end
%
% Now solve the equations
%
Xremote = A2\b2;
%
% residuals of the computation
%
Rremote = b2-A2*Xremote;
Cremote = [Xremote(1:4)';Xremote(5:8)';Xremote(9:11)' 1];
%%
% Check with original data
%
for i=1:1:r
    pixelmaster(i,:)=Cmaster*[mydata1(i,10);mydata1(i,11);mydata1(i,12);1];
    pixelremote(i,:)=Cremote*[mydata1(i,10);mydata1(i,11);mydata1(i,12);1];
end
%
plot3(mydata1(:,10),mydata1(:,11),mydata1(:,12),'g');
%plot3(ENU3(950:1400,1),ENU3(950:1400,2),ENU3(950:1400,3),'r');
%axis([-50 500 -5000 0 -50 1000]);
xlabel('East X')
ylabel('North Y')
zlabel('Up Z')
title('Flight')
grid on
```

Appendix C: Sample Data

Appendix C1: Sample Calibration Data

Base u	B v	B A	R u	R v	R A	GPS	ECEF X	ECEF Y	ECEF Z
320	240	0	320	240.000	0.000	12:02:45	-688911.364	-5159225.49	3674536.91
320	240	0	1.25	398.750	4.000	12:02:45	-688911.786	-5159228.63	3674532.46
633	389.5	6	8.667	397.667	3.000	12:02:45	-688912.202	-5159231.81	3674527.95
624.714	389.429	7	18.2	396.600	5.000	12:02:45	-688912.606	-5159235.02	3674523.37
615.2	389.4	5	25.667	395.667	3.000	12:02:45	-688913.008	-5159238.28	3674518.73
632.435	388.609	23	33.2	395.200	5.000	12:02:46	-688913.417	-5159241.6	3674514.03
629.203	389.156	64	43.5	394.000	2.000	12:02:46	-688913.839	-5159244.95	3674509.25
628.249	388.788	193	43.6	397.200	5.000	12:02:46	-688914.275	-5159248.35	3674504.42
624.975	389.955	244	59.5	394.000	2.000	12:02:46	-688914.733	-5159251.8	3674499.52
618.352	389.835	261	10.556	386.556	9.000	12:02:46	-688915.217	-5159255.28	3674494.55
612.387	390.454	269	10.103	401.724	29.000	12:02:47	-688915.711	-5159258.8	3674489.52
602.547	389.33	300	17.63	401.519	27.000	12:02:47	-688916.2	-5159262.38	3674484.42
589.943	391.868	228	25.067	398.833	30.000	12:02:47	-688916.689	-5159266	3674479.27
585.107	391.316	206	31.259	398.296	27.000	12:02:47	-688917.166	-5159269.67	3674474.05
579.043	392.22	186	34.448	398.517	29.000	12:02:47	-688917.644	-5159273.37	3674468.76
571.968	393.222	216	45.345	398.586	29.000	12:02:48	-688918.127	-5159277.13	3674463.42
564.291	392.143	230	51.259	398.815	27.000	12:02:48	-688918.612	-5159280.93	3674458.02
555.868	393.035	227	320	240.000	0.000	12:02:48	-688919.113	-5159284.78	3674452.55
546.776	393.881	219	69.423	398.731	26.000	12:02:48	-688919.627	-5159288.64	3674447.02
538.18	396.56	150	75.87	399.435	23.000	12:02:48	-688920.168	-5159292.57	3674441.44
533.82	396.752	161	76.958	398.958	24.000	12:02:49	-688920.727	-5159296.53	3674435.8
527.59	397.036	139	84.1	398.450	20.000	12:02:49	-688921.301	-5159300.53	3674430.09
522.553	397.789	123	96.222	399.222	9.000	12:02:49	-688921.873	-5159304.56	3674424.33
513.42	396.182	181	96.333	397.611	18.000	12:02:49	-688922.455	-5159308.63	3674418.52
511.302	395.948	172	115.6	398.400	10.000	12:02:49	-688923.037	-5159312.75	3674412.63
502.064	395.802	172	109.583	397.667	12.000	12:02:50	-688923.622	-5159316.91	3674406.7
494.408	396.337	98	123.111	396.667	9.000	12:02:50	-688924.207	-5159321.1	3674400.71
489.555	395.05	119	125.571	396.714	7.000	12:02:50	-688924.783	-5159325.34	3674394.67
480.218	395.481	133	133.071	393.357	14.000	12:02:50	-688925.367	-5159329.62	3674388.57
477.303	395.258	66	138.833	391.333	12.000	12:02:50	-688925.947	-5159333.94	3674382.42
472.403	396.21	62	145.727	392.909	11.000	12:02:51	-688926.53	-5159338.3	3674376.22

465.15	394.073	193	152.615	392.077	13.000	12:02:51	-688927.115	-5159342.68	3674369.96
459.343	397.014	70	320	240.000	0.000	12:02:51	-688927.704	-5159347.12	3674363.64
451.672	395.639	119	320	240.000	0.000	12:02:51	-688928.286	-5159351.58	3674357.27
453.608	395.6	125	174.864	390.636	22.000	12:02:51	-688928.879	-5159356.08	3674350.85
320	240	0	320	240.000	0.000	12:02:52	-688929.477	-5159360.61	3674344.38
443.851	395.248	161	192.125	389.750	16.000	12:02:52	-688930.088	-5159365.18	3674337.86
432	395.8	10	199.138	388.793	29.000	12:02:52	-688930.715	-5159369.8	3674331.3
428.114	395.529	70	203.944	387.444	18.000	12:02:52	-688931.362	-5159374.44	3674324.68
320	240	0	211.296	386.519	27.000	12:02:52	-688932.023	-5159379.12	3674318.01
426.005	391.903	207	214.538	385.538	13.000	12:02:53	-688932.707	-5159383.83	3674311.29
423.417	390.909	132	320	240.000	0.000	12:02:53	-688933.404	-5159388.55	3674304.52
415.562	389.659	217	221.444	381.889	9.000	12:02:53	-688934.116	-5159393.3	3674297.69
407.92	388.009	225	227.6	382.200	5.000	12:02:53	-688934.827	-5159398.11	3674290.83
406.452	385.817	115	231.4	377.200	5.000	12:02:53	-688935.541	-5159402.95	3674283.92
402.072	384.056	125	234.273	374.545	11.000	12:02:54	-688936.257	-5159407.83	3674276.95
320	240	0	239.833	372.833	6.000	12:02:54	-688936.947	-5159412.75	3674269.95
385.992	379.031	130	244.5	368.167	6.000	12:02:54	-688937.647	-5159417.68	3674262.9
377.757	376.445	247	249	364.000	3.000	12:02:54	-688938.33	-5159422.63	3674255.79
371.63	373.411	219	320	240.000	0.000	12:02:54	-688939.014	-5159427.66	3674248.65
368.206	370.538	160	257.167	356.500	6.000	12:02:55	-688939.711	-5159432.7	3674241.48
362.814	367.67	97	260.8	353.800	5.000	12:02:55	-688940.42	-5159437.77	3674234.25
359.5	365.66	100	262.5	351.500	2.000	12:02:55	-688941.149	-5159442.88	3674226.98
350.586	363.31	58	320	240.000	0.000	12:02:55	-688941.902	-5159448.02	3674219.67
349.348	360.896	115	320	240.000	0.000	12:02:55	-688942.664	-5159453.16	3674212.31
339.18	360.1	50	274.857	342.714	7.000	12:02:56	-688943.446	-5159458.33	3674204.9
330.485	358.242	33	320	240.000	0.000	12:02:56	-688944.228	-5159463.54	3674197.46
329.237	355.789	38	320	240.000	0.000	12:02:56	-688944.999	-5159468.75	3674189.97
327.986	354.859	71	284.333	336.667	3.000	12:02:56	-688945.743	-5159474.01	3674182.44
319.609	352.609	69	287	336.000	1.000	12:02:56	-688946.474	-5159479.31	3674174.89
312.75	351.594	32	290	333.500	2.000	12:02:57	-688947.198	-5159484.64	3674167.3
307.513	351.308	39	320	240.000	0.000	12:02:57	-688947.917	-5159490.03	3674159.7
305.881	348.716	67	296.2	330.200	5.000	12:02:57	-688948.637	-5159495.46	3674152.08
297.5	346.333	36	299.75	328.250	4.000	12:02:57	-688949.358	-5159500.97	3674144.45
320	240	0	303	324.000	1.000	12:02:57	-688950.091	-5159506.56	3674136.85
289.269	343.423	26	320	240.000	0.000	12:02:58	-688950.828	-5159512.19	3674129.21
286.275	342.275	51	320	240.000	0.000	12:02:58	-688951.552	-5159517.78	3674121.51
282.104	340.062	48	320	240.000	0.000	12:02:58	-688952.279	-5159523.35	3674113.73
278.372	337.698	43	313	319.000	4.000	12:02:58	-688952.995	-5159528.86	3674105.86
275.667	336.857	21	320	240.000	0.000	12:02:58	-688953.727	-5159534.4	3674097.96
270.682	333.955	22	321	312.000	1.000	12:02:59	-688954.459	-5159539.96	3674090.03
268.15	332.45	20	323	302.500	2.000	12:02:59	-688955.208	-5159545.58	3674082.09
266.604	331.146	48	326.5	309.000	4.000	12:02:59	-688955.962	-5159551.28	3674074.16

262.111	328.944	54	320	240.000	0.000	12:02:59	-688956.725	-5159557.02	3674066.22
262.182	327.091	11	333.5	304.500	2.000	12:02:59	-688957.484	-5159562.8	3674058.25
257.667	325.333	12	320	240.000	0.000	0.50208333	-688958.239	-5159568.58	3674050.25
259.311	322.644	45	320	240.000	0.000	0.50208333	-688958.987	-5159574.37	3674042.21
249.833	320.167	18	320	240.000	0.000	12:03:00	-688959.736	-5159580.19	3674034.15
252.333	318.267	15	345.25	294.750	4.000	12:03:00	-688960.475	-5159586.03	3674026.05
249.186	315.407	59	347.667	292.000	3.000	12:03:00	-688961.206	-5159591.89	3674017.94
247.053	313	19	320	240.000	0.000	12:03:01	-688961.936	-5159597.77	3674009.79
248.778	310.167	18	320	240.000	0.000	12:03:01	-688962.65	-5159603.67	3674001.62
246.533	308.2	15	320	240.000	0.000	12:03:01	-688963.36	-5159609.58	3673993.41
241.14	305.767	43	320	240.000	0.000	12:03:01	-688964.059	-5159615.51	3673985.17
234.333	304.889	9	320	240.000	0.000	12:03:01	-688964.748	-5159621.45	3673976.89
234.962	300	26	320	240.000	0.000	12:03:02	-688965.432	-5159627.4	3673968.59
226	299.25	8	320	240.000	0.000	12:03:02	-688966.109	-5159633.37	3673960.24
229.111	295.63	27	320	240.000	0.000	12:03:02	-688966.784	-5159639.35	3673951.87
221.9	294.4	10	320	240.000	0.000	12:03:02	-688967.462	-5159645.35	3673943.47
220.167	290.833	6	320	240.000	0.000	12:03:02	-688968.15	-5159651.37	3673935.04
219.242	287.758	33	320	240.000	0.000	12:03:03	-688968.852	-5159657.4	3673926.58
220.625	284.833	24	383	259.000	1.000	12:03:03	-688969.565	-5159663.42	3673918.07
217.615	283.385	13	320	240.000	0.000	12:03:03	-688970.28	-5159669.45	3673909.52
211	281.6	10	320	240.000	0.000	12:03:03	-688971.003	-5159675.48	3673900.93
206.1	278.4	10	320	240.000	0.000	12:03:03	-688971.727	-5159681.53	3673892.32
211.786	275.929	14	320	240.000	0.000	12:03:04	-688972.465	-5159687.6	3673883.67
204.385	274.077	13	320	240.000	0.000	12:03:04	-688973.216	-5159693.7	3673875.01
320	240	0	320	240.000	0.000	12:03:04	-688973.978	-5159699.84	3673866.34
208.364	269.273	11	320	240.000	0.000	12:03:04	-688974.76	-5159706.01	3673857.66
198.879	266.966	58	320	240.000	0.000	12:03:04	-688975.559	-5159712.23	3673848.97
199.846	264.462	13	320	240.000	0.000	12:03:05	-688976.376	-5159718.5	3673840.29
192.435	262.217	23	320	240.000	0.000	12:03:05	-688977.222	-5159724.78	3673831.58
320	240	0	320	240.000	0.000	12:03:05	-688978.093	-5159731.07	3673822.86
185.63	256.593	27	320	240.000	0.000	12:03:05	-688978.997	-5159737.36	3673814.1
182.7	254.9	10	320	240.000	0.000	12:03:05	-688979.934	-5159743.64	3673805.29
180.286	252.714	7	320	240.000	0.000	12:03:06	-688980.879	-5159749.92	3673796.46
180.667	250.833	6	320	240.000	0.000	12:03:06	-688981.838	-5159756.18	3673787.58
176.538	247.692	13	320	240.000	0.000	12:03:06	-688982.8	-5159762.45	3673778.68
176.25	247	4	320	240.000	0.000	12:03:06	-688983.75	-5159768.74	3673769.75
174	246	1	320	240.000	0.000	12:03:06	-688984.692	-5159775.07	3673760.82
172.4	243.8	5	320	240.000	0.000	12:03:07	-688985.603	-5159781.42	3673751.87
169.867	240.067	15	320	240.000	0.000	12:03:07	-688986.499	-5159787.81	3673742.92
168.643	233.214	14	320	240.000	0.000	12:03:07	-688987.358	-5159794.2	3673733.95
166.5	238.5	6	627.5	372.000	2.000	12:03:07	-688988.206	-5159800.71	3673725.01
164.4	236.8	5	320	240.000	0.000	12:03:07	-688989.034	-5159807.27	3673716.09

320	240	0	320	240.000	0.000	12:03:08	-688989.849	-5159813.87	3673707.19
160	233	3	320	240.000	0.000	12:03:08	-688990.656	-5159820.54	3673698.3
158	231.75	8	320	240.000	0.000	12:03:08	-688991.451	-5159827.24	3673689.45
155.75	230.25	4	320	240.000	0.000	12:03:08	-688992.246	-5159833.99	3673680.6
154.5	228.5	4	320	240.000	0.000	12:03:08	-688993.039	-5159840.77	3673671.77
150.556	226.222	9	320	240.000	0.000	12:03:09	-688993.838	-5159847.58	3673662.96
150.333	224.333	3	320	240.000	0.000	12:03:09	-688994.649	-5159854.44	3673654.18
148.4	222.8	5	320	240.000	0.000	12:03:09	-688995.467	-5159861.31	3673645.39
147	220	3	320	240.000	0.000	12:03:09	-688996.278	-5159868.21	3673636.61
144.5	218	4	320	240.000	0.000	12:03:09	-688997.097	-5159875.1	3673627.81
320	240	0	320	240.000	0.000	12:03:10	-688997.914	-5159881.95	3673618.98
141	212	3	320	240.000	0.000	12:03:10	-688998.724	-5159888.8	3673610.12
320	240	0	320	240.000	0.000	12:03:10	-688999.527	-5159895.61	3673601.24
320	240	0	320	240.000	0.000	12:03:10	-689000.32	-5159902.41	3673592.32
320	240	0	320	240.000	0.000	12:03:10	-689001.102	-5159909.21	3673583.39
135	202	1	320	240.000	0.000	12:03:11	-689001.867	-5159915.99	3673574.43
131	198.75	8	320	240.000	0.000	12:03:11	-689002.62	-5159922.77	3673565.46
132	197	1	320	240.000	0.000	12:03:11	-689003.371	-5159929.59	3673556.5
320	240	0	447.667	159.810	21.000	12:03:11	-689004.103	-5159936.42	3673547.54
127.714	192	7	320	240.000	0.000	12:03:11	-689004.84	-5159943.29	3673538.59
127	191	1	320	240.000	0.000	12:03:12	-689005.6	-5159950.2	3673529.65
320	240	0	320	240.000	0.000	12:03:12	-689006.383	-5159957.09	3673520.7
124	185.5	2	320	240.000	0.000	12:03:12	-689007.196	-5159963.98	3673511.75
122	183	4	320	240.000	0.000	12:03:12	-689008.051	-5159970.89	3673502.8
120.5	180.5	2	320	240.000	0.000	12:03:12	-689008.942	-5159977.81	3673493.85
116.5	177.833	6	320	240.000	0.000	12:03:13	-689009.868	-5159984.76	3673484.9
115.6	174.4	10	320	240.000	0.000	12:03:13	-689010.802	-5159991.66	3673475.94
114.5	172.5	6	320	240.000	0.000	12:03:13	-689011.745	-5159998.61	3673466.98
320	240	0	320	240.000	0.000	12:03:13	-689012.693	-5160005.56	3673458.03
111	168	1	320	240.000	0.000	12:03:13	-689013.633	-5160012.56	3673449.1
109.5	166	2	320	240.000	0.000	12:03:14	-689014.574	-5160019.56	3673440.19
108.333	164.333	3	320	240.000	0.000	12:03:14	-689015.52	-5160026.61	3673431.3
106	162	7	320	240.000	0.000	12:03:14	-689016.47	-5160033.7	3673422.45
105.333	159.667	3	320	240.000	0.000	12:03:14	-689017.428	-5160040.82	3673413.61
103.4	157.8	5	320	240.000	0.000	12:03:14	-689018.403	-5160047.92	3673404.78
102	156	1	320	240.000	0.000	12:03:15	-689019.395	-5160055	3673395.93
320	240	0	320	240.000	0.000	12:03:15	-689020.405	-5160062.08	3673387.08
98.5	152	2	320	240.000	0.000	12:03:15	-689021.427	-5160069.15	3673378.24
320	240	0	320	240.000	0.000	12:03:15	-689022.439	-5160076.25	3673369.41
96.667	148.667	3	320	240.000	0.000	12:03:15	-689023.444	-5160083.36	3673360.58
95	147	3	320	240.000	0.000	12:03:16	-689024.462	-5160090.4	3673351.72
91.333	142	6	320	240.000	0.000	12:03:16	-689025.459	-5160097.51	3673342.9

320	240	0	320	240.000	0.000	12:03:16	-689026.437	-5160104.6	3673334.08
320	240	0	320	240.000	0.000	12:03:16	-689027.392	-5160111.72	3673325.28
88	137	1	320	240.000	0.000	12:03:16	-689028.317	-5160118.86	3673316.51
320	240	0	320	240.000	0.000	12:03:17	-689029.221	-5160126.02	3673307.75
85.6	134.2	5	320	240.000	0.000	12:03:17	-689030.111	-5160133.2	3673299.02
320	240	0	320	240.000	0.000	12:03:17	-689030.957	-5160140.41	3673290.31
320	240	0	320	240.000	0.000	12:03:17	-689031.814	-5160147.56	3673281.59
320	240	0	320	240.000	0.000	12:03:17	-689032.65	-5160154.76	3673272.89
320	240	0	320	240.000	0.000	12:03:18	-689033.474	-5160161.91	3673264.17
79	127	1	320	240.000	0.000	12:03:18	-689034.307	-5160169.06	3673255.45

Appendix C2: Sample Flight Data

Remote x	Remote y	Remote A	Master x	Master y	Master A	UTC time
13.118	416.804	51.000	617.181	394.343	210.000	170002.580
10.85	417.267	60.000	612.686	393.461	204.000	170002.650
14.136	416.621	66.000	574.963	396.487	160.000	170002.680
15.045	416.788	66.000	610.737	396.109	247.000	170002.740
22.472	418.862	254.000	521.508	403.326	384.000	170002.800
18.315	421.764	89.000	537.399	400.709	378.000	170002.820
27.903	418.296	277.000	592.309	396.787	333.000	170002.840
22.955	422.552	134.000	609.978	395.377	371.000	170002.940
24.194	421.930	129.000	538.873	405.587	315.000	170003.000
31.717	419.764	293.000	603.089	395.897	380.000	170003.020
32.492	419.516	122.000	576.188	387.733	1437.000	170003.240
41.375	415.833	618.000	583.663	392.128	392.000	170003.290
48.904	416.471	272.000	580.754	395.623	183.000	170003.340
45.974	408.145	311.000	577.922	393.391	243.000	170003.380
51.216	407.598	301.000	574.809	394.671	152.000	170003.430
57.311	413.689	122.000	577.822	393.673	214.000	170003.470
47.622	415.933	135.000	577.508	390.108	482.000	170003.500
52.181	413.349	149.000	524.980	393.238	608.000	170003.560
49.153	414.210	157.000	527.141	405.223	354.000	170003.580
51.277	407.041	390.000	547.831	387.289	692.000	170003.620
46	413.610	177.000	567.644	385.263	744.000	170003.660
54.158	410.385	234.000	583.540	390.241	278.000	170003.700
51.777	403.607	448.000	581.877	387.091	309.000	170003.760
52.223	407.791	206.000	555.867	383.290	555.000	170003.810
57.655	401.072	472.000	557.141	380.889	866.000	170003.820
59.06	405.046	216.000	548.688	392.563	391.000	170003.890
60.312	404.427	199.000	545.046	378.014	1162.000	170003.950
60.348	404.860	178.000	547.829	377.578	578.000	170004.000
320	240.000	0.000	556.462	380.353	340.000	170004.020
61.707	395.569	540.000	558.583	383.241	324.000	170004.050
60.049	399.866	224.000	534.212	373.529	1135.000	170004.100

Table 4 23 October Flight 11:29:13

Appendix D: Generated Tracks



Figure D 30 November 8th Flight

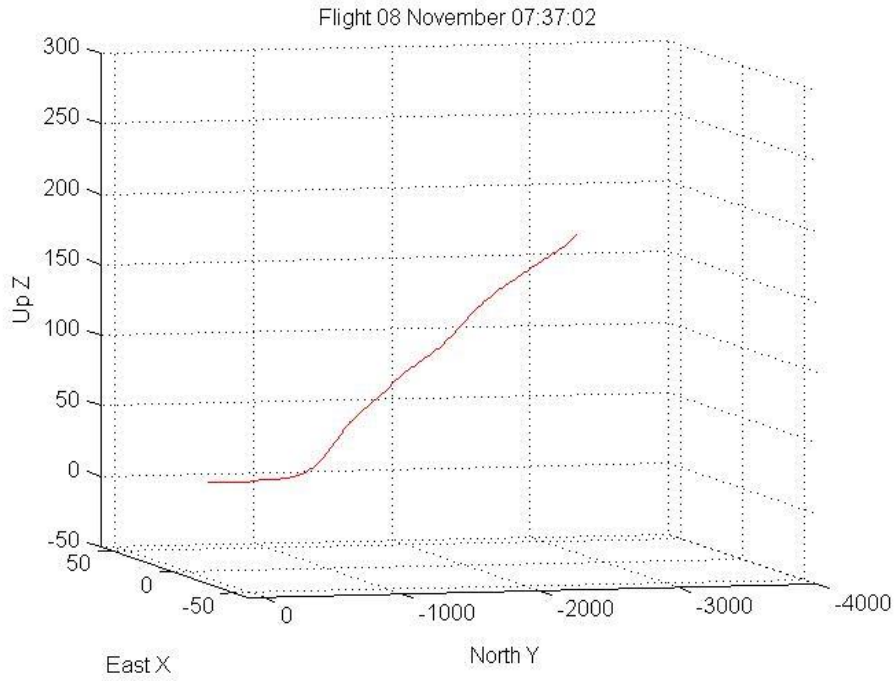


Figure D 31 November 8th Generated Track



Figure D 32 November 9th Flight

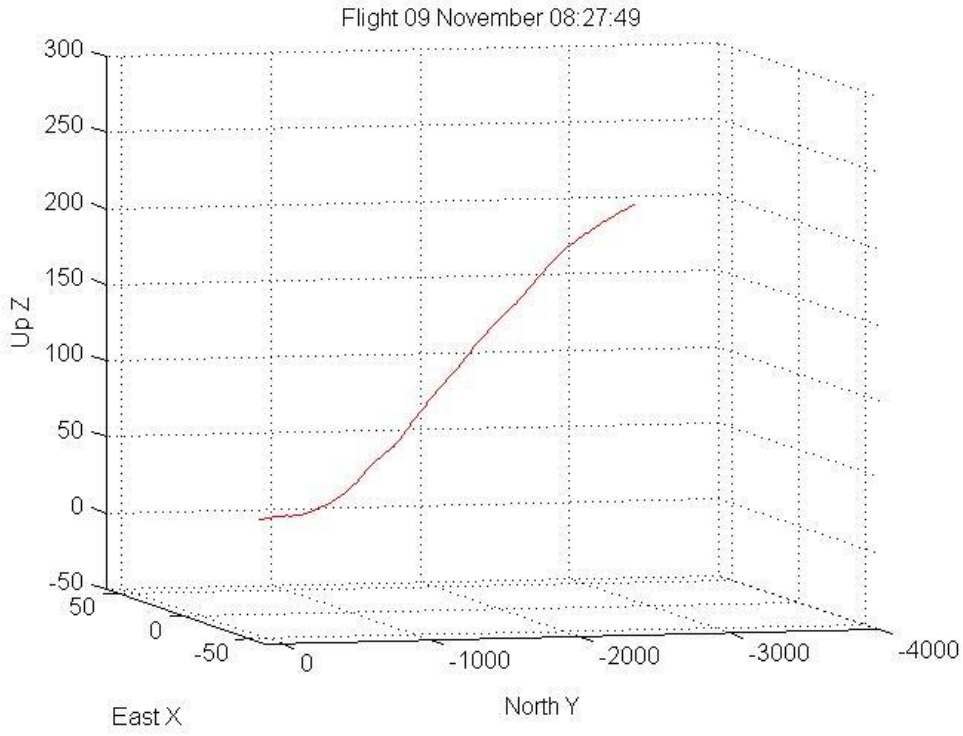


Figure D 33 November 9th Generated Track



Figure D 34 November 15th Flight

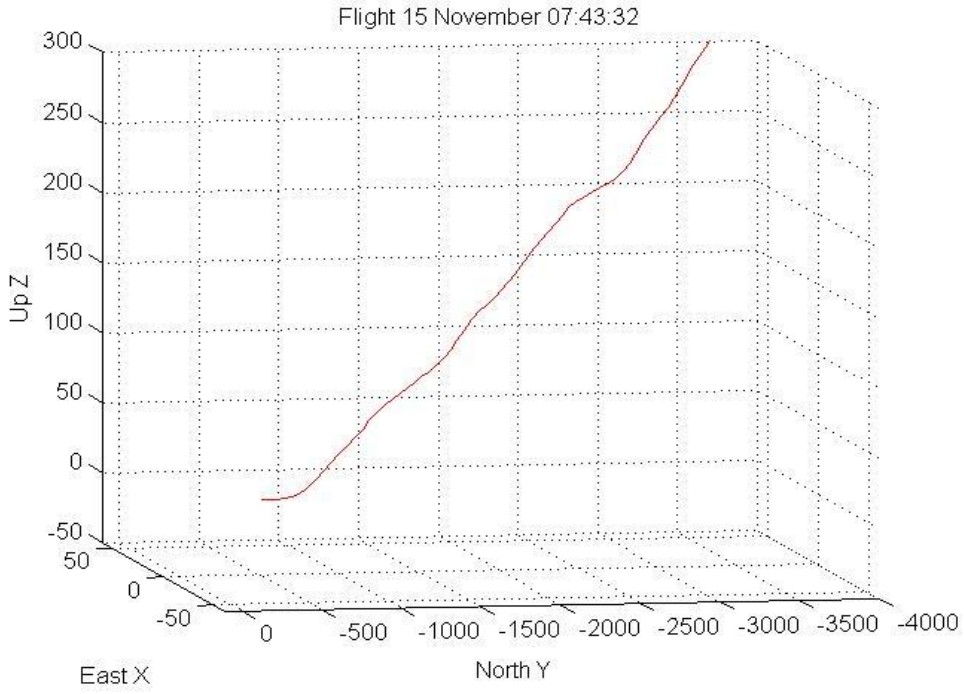


Figure D 35 November 15th Generated Track



Figure D 36 November 15th Flight

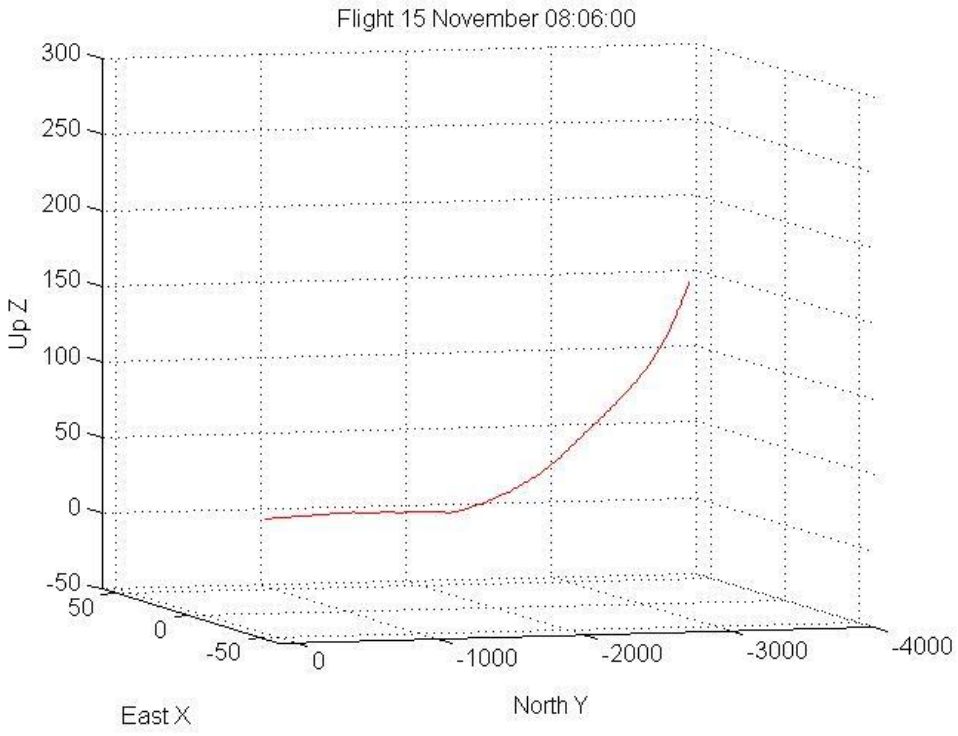


Figure D 37 November 15th Generated Track



Figure D 38 November 9th Flight

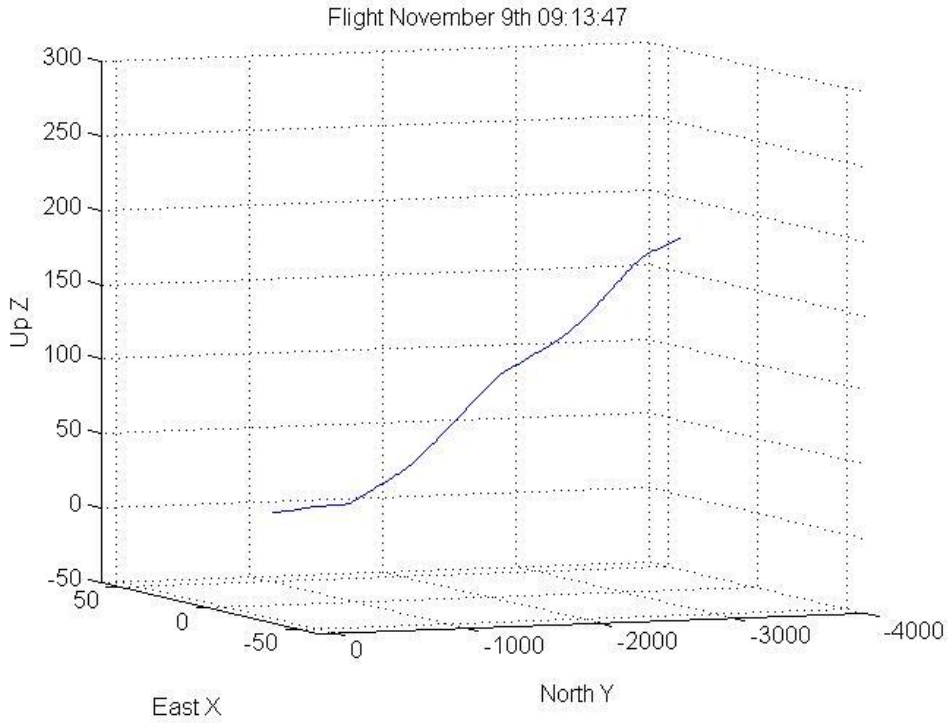


Figure D 39 November 9th Generated Track