ON THE FORMULATION OF INERTIAL ATTITUDE ESTIMATION USING POINT CORRESPONDENCES AND DIFFERENTIAL CARRIER PHASE GPS POSITIONING: AN APPLICATION IN STRUCTURE FROM MOTION

By

SCOTT CLARK ROSENGREN

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2004

This thesis is dedicated to *BG*. I will always love you.

ACKNOWLEDGMENTS

First, I give all the glory and honor to my LORD and Saviour Jesus Christ for allowing and enabling me to complete this work.

I also thank my wife for putting up with countless late nights and early mornings, reading each copy of the manuscript, watching our daughter when she was *done*, and all the precious time that she sacrificed for me to work two jobs for the last four years. She has spent at least as much effort as I did in completing this thesis and all the coursework. I thank her for her never-ending love and support. My love will be forever hers.

I wish to thank my parents for supporting me, and convincing me that I can do anything I set my mind to accomplish.

Without the support and direction from Professor Eric Sutton, this project would not have even begun. I thank him for the time and energy he spent tutoring and mentoring me over the last two years.

My gratitude is extended to Ron Smith for developing the ufthesis document class for the LaTeX word processing program in which this document was prepared.

Finally, I wish to thank my company Dynetics, Inc. in Shalimar, FL for allowing me the time and resources to complete my coursework and thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

ON THE FORMULATION OF INERTIAL ATTITUDE ESTIMATION USING POINT CORRESPONDENCES AND DIFFERENTIAL CARRIER PHASE GPS POSITIONING: AN APPLICATION IN STRUCTURE FROM MOTION

By

Scott Clark Rosengren

May 2004

Chair: Eric Sutton
Major Department: Electrical and Computer Engineering

Combining the results from Structure from Motion (*SfM*) processing of two overlapping images with precise GPS differential carrier phase positioning yields an essentially "free" attitude estimate for the platform aircraft. This technique is "free," because it uses sensor systems already required on the platform aircraft for other reasons. The *SfM* processing results in a displacement vector in the body coordinate system between the two camera locations. The GPS differential carrier phase positioning is used to produce an extremely accurate displacement vector in a local level coordinate system. Once two vectors are found that are expressed in both coordinate systems, then the transformation between the two coordinate systems can be found. This corresponds directly to the platform vehicle's attitude. A detailed error analysis on a linearized model of the system shows that errors on the attitude estimate are on the order of tens of mrad for a measurement error standard deviation of one pixel.

Background information in both the *SfM* and GPS fields is given. Then, the problem is described, all assumptions are stated, and solution techniques are presented. Finally, a detailed error analysis is performed on a linearized model of the system. The key contribution of this thesis is the detailed error analysis of the attitude estimate.

CHAPTER 1
INTRODUCTION

Much research over the last 20 years in the Structure from Motion (*SfM*) field has focused on using information from multiple *2-D* images of the same scene taken from different camera locations to calculate a *3-D* rendering of the scene and the displacement of the camera, commonly called the reconstruction problem. This thesis presents an in-depth study of a new application for this field of study. By utilizing techniques from *SfM*, differential positioning information from GPS sensors can be combined with vision-system information to yield an estimate of vehicle heading.

Increasing the accuracy and robustness of airborne vehicle navigation is the objective driving the formulation of the problem presented in this thesis. The primary focus of this work is to examine an algorithm that uses data that will be available on the vehicle for other reasons. It is assumed that the airborne vehicle is small, low cost, mostly autonomous, and used for both attack and/or intelligence gathering. This vehicle is envisioned to be able to fly as part of multiple or single vehicle missions. Each vehicle or agent will almost certainly require the following four sensor systems: (1) a low grade inertial measurement unit (IMU), (2) a GPS receiver, (3) a vision system, and (4) an inter-agent communication system (if part of multi-agent mission). The IMU and GPS combination provides accurate navigation, and the IMU provides attitude sensors for vehicle control. The vision system is necessary for intelligence gathering or target acquisition. An inter-agent communication system is necessary for distributed intelligence and will provide secure communication with jamming resistance using technology such as spread spectrum or ultra wide band. Both of these technologies can be used to measure precise distances between transmitter and receiver, in addition to the communication functions.

The problem detailed in this thesis uses vision correspondences from multiple reference systems, combined with the corresponding inertial displacement vectors between the systems to obtain the inertial attitude of one of the systems. The vision correspondences of the reference systems could be from (1) different vehicles viewing parts of the same scene at the same time or (2) one moving vehicle that takes multiple images. The only constraint is that the point

correspondences remain constant in all images. As far as the formulation of the problem, it makes no difference whether it is thought of as (1) or (2) above.

As an example,[1] suppose vehicle one is directly behind vehicle two; and that vehicle one can sense, using its vision system, the angular location of vehicle two with respect to itself. If the relative position of vehicle two with respect to vehicle one is known using GPS, then the pitch and heading of vehicle one can be determined very precisely. This simplified scenario clearly imposes some unacceptable operational constraints, but these constraints can be removed using more sophisticated algorithms to process the sensor information. If both vehicles have downward-looking vision systems and the two fields of view overlap, then it should be possible to obtain the same information as would be obtained if vehicle two were directly visible to vehicle one. In addition, even if the two fields of view do not simultaneously overlap, if their point correspondences remain fixed in both images, that should provide enough information to calculate the attitude of one of the vehicles. The baseline between vehicles will be long enough to provide an extremely accurate pointing direction; the accuracy of this technique will depend primarily on the vision-system geometry. The primary objective of this study is a thorough error analysis of this technique.

This thesis is organized in the following manner. A brief overview of the Structure from Motion field that uses point correspondences to calculate spatial and orientation characteristics is given, along with a mathematical overview of the reconstruction problem given in the Euclidean framework. A chapter on the benefits gained from GPS differential carrier phase positioning is presented. Next, the problem of interest is mathematically defined, all simplifying assumptions are given, and solution techniques are presented. Then, an error analysis chapter develops a linearized model of the system and provides a detailed look at how various parameters as well as the overall scene geometry affect the heading estimate. Results for the other error terms are also shown. Finally, a conclusion section presents what the author believes are the important findings contained in this thesis, and further areas to explore.

[1] Originally given by E. Sutton in a 2001 white paper originating from the University of Florida Graduate Engineering and Research Center in Shalimar, FL.

CHAPTER 2
OVERVIEW OF EXPLOITATION OF POINT CORRESPONDENCES IN MULTIPLE IMAGES OF THE SAME SCENE

Using point correspondences between two different *2-D* views of the same scene to derive *3-D* information about that scene is not a new idea. As early as the mid-nineteenth century, the field of photogrammetry used point correspondences between two images of the same scene from different camera locations to aid in the production of topographical maps to assist the explorers of that generation [1]. With the advent of increased computing power in the 1980s, the techniques of photogrammetry have found application in robotics and computer vision. Research in robotics has emphasized vision as an aid to navigation, and research in computer vision has emphasized the reconstruction of *3-D* scenes from *2-D* data. The latter comprises the field known as Structure from Motion (*SfM*).

The original formulation of this problem used a technique called projective geometry to describe the relationships between image correspondences [2, 3]. Projective geometry uses ray tracing through the optical center of the camera system and epipolar transformations to describe the reconstruction of the *3-D* scene. More recently, *SfM* researchers have also developed and quantified the *3-D* reconstruction problem using Euclidean geometry [1, 4, 5, 6]. The Euclidean framework simplifies the problem to that of finding a solution to a set of nonlinear equations, and lends itself to an analytical solution. Maybank [1] gives the fundamental difference between these approaches on their different views of geometry: synthetic versus analytic. Projective geometry uses a synthetic approach that gives interpretations to the equations describing the scene based on the geometric shapes being seen (lines, planes, conics, etc.). The drawback to the synthetic approach is that it is difficult to make it completely rigorous. The analytic approach, used in the Euclidean framework, leads to the introduction of an essential matrix that concisely and robustly describes the motion of the camera. Maybank [1] gives a thorough background and outlines the mathematical complexities of both techniques. Before exploring some of the relevant research in *SfM*, a basic overview of the problem in the Euclidean framework is given.

2.1 General Structure from Motion (*SfM*) Problem Setup and Notation for Euclidean Framework

Before going into further findings about *2D-to-2D* point correspondences from *SfM* research, a general mathematical overview of the classical *SfM* problem needs to be given. General notation used throughout the community is not standardized, so the notation used in this thesis is given an elementary treatment.

2.1.1 General Mathematical Notations

Vectors and matrices. The notation of a boldfaced lowercase letter denotes a vector. The superscript denotes the coordinate system in which the components of the vector are expressed. For example, $\mathbf{p}^{c1}$ represents the vector $\mathbf{p}$ expressed in the (*c1*) *3-D* coordinate system. A unit vector is denoted by a carat on top of the boldfaced lowercase letter. For example, a unit vector in the same direction as the vector given above would be denoted as $\hat{\mathbf{p}}^{c1}$. A single capital letter is used to designate a matrix.

Rotation matrices. R_{c1}^{c2} represents an orthonormal rotation matrix where the subscript (*c1*) represents the coordinate system that the rotation will go from and the superscript (*c2*) represents the coordinate system that the rotation will go to. The orthonormal rotation matrix can be expressed as the product of successive rotations about the three body axes that are fixed to the system undergoing the rotation. The three angles associated with the three successive rotations are called Euler angles. For this thesis, a 3-2-1 sequence of rotations is used in reference to Euler angles, where 3 stands for the z-axis, 2 for the resulting y-axis, and 1 for resulting x-axis. All rotations are right-handed. That is, the system will first yaw (ψ) about the z-axis (3), pitch (θ) about the resulting y-axis (2), and then roll (ϕ) about the resulting x-axis (1). This is represented mathematically as

$$R_1^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0) \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-1}$$

Also, the inverse of an orthonormal rotation matrix is equal to its transposition.

$$R^{-1} = R^T \tag{2-2}$$

So, for a rotation matrix that goes from system 1 to system 2,

$$R_1^2 = (R_2^1)^T. \tag{2-3}$$

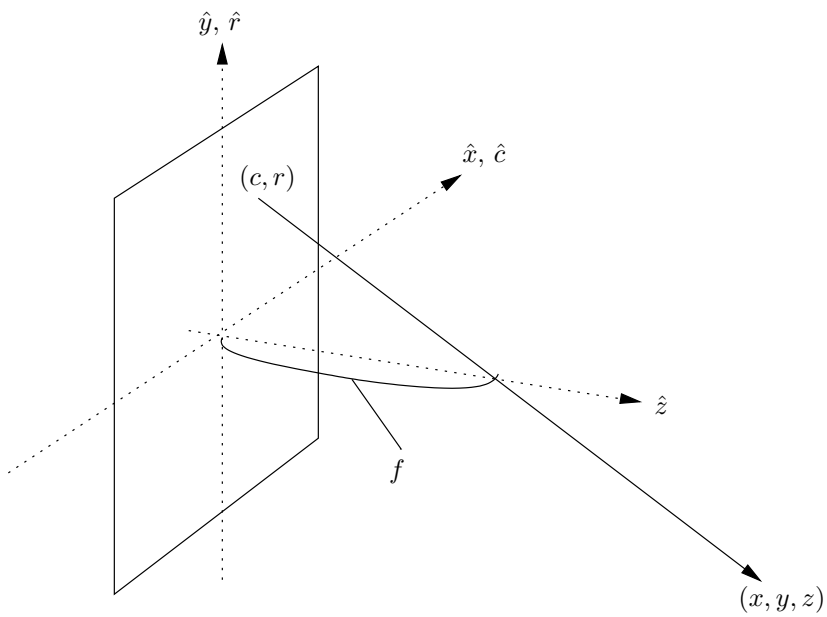

Figure 2–1: General *SfM* problem notation for Euclidean Framework

2.1.2 Imaging System

Let an imaging system have length f from the optical focus point to the focal plane. This is commonly referred to as the focal length. Image coordinates c and r are the *2-D* coordinates on the focal plane used to represent the projection of a point in the three-dimensional (*3-D*) scene given by (x,y,z). The origin of the *3-D* scene is collocated with the origin of the imaging system's focal plane. The x-axis and y-axis of the *3-D* scene correspond to the c and r axes of *2-D* focal plane coordinates. The z-axis of the *3-D* scene is orthogonal to the focal plane (i.e., directly in the line of sight or pointing direction). By using analytical geometrical relationships, the imaging system coordinates are related to the *3-D* scene coordinates by Equations 2-4 and 2-5, as shown in Figure 2–1.

$$c = f\,\frac{x}{z} \tag{2-4}$$

$$r = f\,\frac{y}{z} \tag{2-5}$$

For the *2D-to-2D* solution, two different views or images of the same scene with n correspondences are used to reconstruct the *3-D* scene in either one of the camera *3-D* coordinate systems. Let the origin of the first camera system be given by $\mathbf{o}^{c1}$ and the origin of the second camera system be given by $\mathbf{o}^{c2}$ and both of their values are $(0, 0, 0)$ in their respective *3-D* coordinate systems. The displacement between the camera systems is given by a translation $\mathbf{a}^{c1}$ and a rotation

R_{c1}^{c2}. Note that it is arbitrary to which *3-D* camera system is called *c1* and which is called *c2* when using this notation. This detail is pointed out because the solution technique of chapter 4 and the error analysis technique of chapter 5 solve for the attitude of *c1* and *c2* respectively, as given in Figure 2–2.

For this particular example, if the translation between the two camera coordinate systems is zero:

$$\mathbf{q}^{c2} = R_{c1}^{c2}\,\mathbf{p}^{c1} \tag{2-6}$$

So, adding in a non-zero translation gives the coordinate system conversion as:

$$\mathbf{q}^{c2} = R_{c1}^{c2}\,(\mathbf{p}^{c1} - \mathbf{a}^{c1}) \tag{2-7}$$

The lines $\mathbf{p}^{c1}$ and $\mathbf{q}^{c2}$ represent the *3-D* scene coordinates of a point correspondence expressed in the first and second camera system respectively. The geometry using two cameras is shown in Figure 2–2. If the magnitude of $\mathbf{p}^{c1}$ is taken to be p and the unit vector is denoted as $\hat{\mathbf{p}}^{c1}$, and likewise for the second coordinate system, then

$$q\,\hat{\mathbf{q}}^{c2} = R_{c1}^{c2}\,(p\,\hat{\mathbf{p}}^{c1} - \mathbf{a}^{c1}) \tag{2-8}$$

Equation 2-8 forms the basis for the Euclidean reconstruction problem used in computer vision.[1] The reconstruction problem can then be rewritten in a Euclidean framework as such: for n point correspondences, find the scalar values p_i and q_i along with the camera displacement described by R_{c1}^{c2} and $\mathbf{a}^{c1}$ that is valid for each $\hat{\mathbf{p}}_i^{c1}$, $\hat{\mathbf{q}}_i^{c2}$ point correspondence pair.

2.1.3 Presentation of the Essential Matrix to De-Couple Structure from Motion

Equation 2-8 takes into account the translation and rotation between the first and second camera coordinate systems when viewing a common point in real space. The following logic can be applied to separate the structure and motion components of the Euclidean *SfM* problem [1]:

- Take the limit as $\|\mathbf{p}^{c1}\| \to \infty$, $\|\mathbf{q}^{c2}\| \to \infty$, and $\frac{p}{q} \to 1$. This gives the relationship between $\hat{\mathbf{p}}_i^{c1}$ and $\hat{\mathbf{q}}_i^{c2}$ as

$$\hat{\mathbf{q}}^{c2} = R_{c1}^{c2}\,\hat{\mathbf{p}}^{c1} \tag{2-9}$$

- The dot product between $\hat{\mathbf{q}}^{c2}$ and $R_{c1}^{c2}\hat{\mathbf{p}}^{c1} \times R_{c1}^{c2}\mathbf{a}^{c1}$ yields

$$(R_{c1}^{c2}\hat{\mathbf{p}}^{c1} \times R_{c1}^{c2}\mathbf{a}^{c1}) \cdot \hat{\mathbf{q}}^{c2} = 0 \tag{2-10}$$

[1] Equivalent to Equation 2-1 in Maybank. [1]

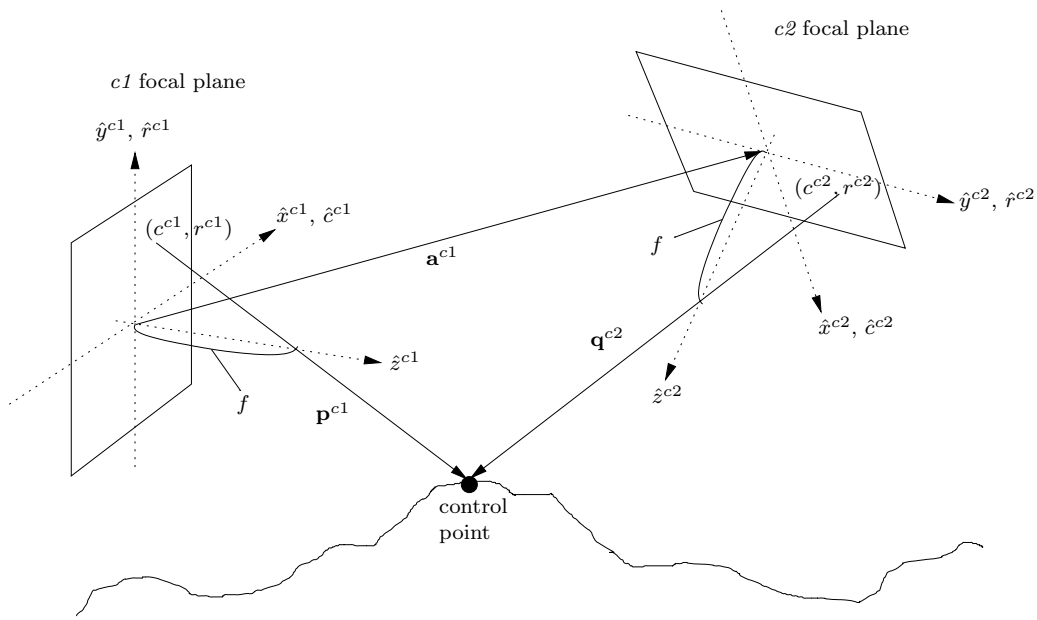

Figure 2–2: Geometry for *SfM 2D-to-2D* point correspondence problem.

This is referred to as the coplanar constraint and states in mathematical terms that two lines intersecting the same point in space must have a common plane with both lines being a member. If Equation 2-10 holds and $\hat{\mathbf{q}}^{c2} \times R_{c1}^{c2}\hat{\mathbf{p}}^{c1} \neq 0$ then the depth, or p and q can be found via Equation 2-8. If $\hat{\mathbf{q}}^{c2} \times R_{c1}^{c2}\hat{\mathbf{p}}^{c1} = 0$ then $\hat{\mathbf{q}}^{c2} = R_{c1}^{c2}\hat{\mathbf{p}}^{c1}$ and the depth is infinite.

- Write Equation 2-10 using the antisymmetric matrix $T_{a^{c1}}$ involving the vector components of $\mathbf{a}^{c1}$ where

$$T_{a^{c1}} \triangleq \begin{bmatrix} 0 & a_3 & -a_2 \\ -a_3 & 0 & a_1 \\ a_2 & -a_1 & 0 \end{bmatrix} . \tag{2-11}$$

- Noting that $T_{a^{c1}}\hat{\mathbf{p}}^{c1} = \hat{\mathbf{p}}^{c1} \times \mathbf{a}^{c1}$ gives an alternate representation for Equation 2-10 as

$$(\hat{\mathbf{q}}^{c2})^T \, R_{c1}^{c2} \, T_{a^{c1}} \, \hat{\mathbf{p}}^{c1} = 0 \ . \tag{2-12}$$

- An essential matrix is defined as a 3×3 matrix which is the product of an orthogonal rotation matrix and a non-zero antisymmetric matrix. Letting the essential matrix E be defined as $E = R_{c1}^{c2} \, T_{a^{c1}}$ gives

$$(\hat{\mathbf{q}}^{c2})^T \, E \, \hat{\mathbf{p}}^{c1} = 0 \ . \tag{2-13}$$

The most important property of essential matrices is that they can be defined as the zeros of a set of nine homogeneous polynomial equations of degree three in the coefficients of 3×3 matrices [1]. Another key property of essential matrices is that they are of rank two.[2]

A third statement of the problem based on the essential matrix formulation in the Euclidean framework is such: for n point correspondences, find the essential matrix E that is valid for all $\hat{\mathbf{p}}_i^{c1}$, $\hat{\mathbf{q}}_i^{c2}$ point correspondences. The advantage for this formulation is that motion and structure are now decoupled in the solution. That is, the camera displacement originally given by R_{c1}^{c2} and $\mathbf{a}^{c1}$ is summarized succinctly in E. Then, given the solution for E, the structure components of the solution given by p_i and q_i can be found by the relationships given in Equation 2-8. This approach lends itself to an algorithm that begins by solving for camera displacement independent of structure, which will be important later.

2.1.4 Note About Solution Ambiguities in the Euclidean Framework

For any solution to the reconstruction problem utilizing the Euclidean framework there are two ambiguities that will always be present: (1) a scaling ambiguity and (2) a twisted pair of solutions [1]. The scaling ambiguity is inherent because the absolute size of any object in a 2-D image is not known. That is, a larger object farther away may appear to be the same size as a smaller object that is nearer. The scaling ambiguity can be resolved by placing arbitrary constraints on the translation vector $\mathbf{a}^{c1}$ [1] (e.g., $\|\mathbf{a}^{c1}\| = 1$).

The twisted pair solution arises by rotating the camera 180 degrees around the axis of translation. After rotation around the translation axis, it can be shown that there is a second set of camera displacement terms, S_{c1}^{c2} and $\mathbf{b}^{c1}$ that is valid for each $\hat{\mathbf{p}}_i^{c1}$, $\hat{\mathbf{q}}_i^{c2}$ point correspondence pair [1]. That is,

$$E = R_{c1}^{c2}\, T_{a^{c1}} = S_{c1}^{c2}\, T_{b^{c1}}. \tag{2-14}$$

A single solution in the Euclidean framework is taken to be the sum of solutions over the scaling and twisted-pair ambiguities.

2.2 Problem Categorization and Refinement

There are different solution approaches based upon what information is available to the many applications of the *SfM* problem. Huang and Netravali [6] have placed these solution approaches into three categories:

[2] For further details on essential matrices the reader can look in Section 2.2 of Maybank's text [1] and other *SfM* literature [4, 5, 6].

- Three-dimensional (*3D*) to three-dimensional (*3D*) feature correspondences
- Two-dimensional (*2D*) to three-dimensional (*3D*) feature correspondences
- Two-dimensional (*2D*) to two-dimensional (*2D*) feature correspondences

This thesis explores the third of Huang's categories, specifically using point correspondences in a Euclidean framework. According to Huang [6], the applications that *2D*-to-*2D* feature correspondences are used to solve are

- Finding relative attitudes of two cameras observing the same scene
- Estimating motion and structure of objects moving relative to a camera
- Passive navigation (i.e., finding the relative attitude of a vehicle at two different time instants)
- Efficient coding and noise reduction of image sequences by estimating motion of objects.

The objective of this thesis corresponds to the first and third applications given above. As shown in section 2.1, it makes no difference mathematically whether the problem is stated as finding relative attitude of two cameras simultaneously viewing the same scene or a single camera viewing an overlapping scene at two different time instances.

An attempt to form a general framework for comparing the various flavors of solution algorithms is given by Soatto and Perona [5, 7]. They give five instances of their general model. One of these five is the *Essential Model*, which uses the coplanarity constraint introduced by Longuet-Higgens [8]. This formulation is essentially the Euclidean framework described earlier. Another of the five instances of their general model is the *Subspace model*. It consists of the subspace constraint introduced by Heeger [9] which interprets the model as a *dynamical system* rather than an algebraic constraint. This has the same advantage of the Euclidean framework in that it decouples the motion and structure components in the solution, but has an additional advantage of further decoupling the rotational velocity from the translational velocity. The other three instances of their model involve fixation of a various number of features on the focal plane and are not relevant to this thesis.

Soatto and Perona's research into generalizing the many *SfM* applications into a common framework looks promising because they present the idea of integrating information over time. If integration of information can be accomplished, then the effective baseline between images will be increased. A longer baseline should result in increased accuracy of the reconstruction. Soatto and Perona's work has the implication that any solution technique can be conformed to the general model and thus have the integration benefit provided therein. This caveat has the potential to

increase the accuracy of any application that is currently based on a *SfM* algorithm to perform reconstruction.

CHAPTER 3
BENEFITS GAINED FROM GLOBAL POSITIONING SYSTEM (GPS) DIFFERENTIAL CARRIER PHASE POSITIONING.

The Global Positioning System (GPS) is a technology consisting of a constellation of satellites in prescribed orbits transmitting known signals to be used for precise positioning via sophisticated triangulation techniques. Each satellite transmits pseudo-random codes at known frequencies. There are two ways to determine position from the GPS constellation: (1) code delay measurements and (2) carrier phase measurements. Typical user range errors (UREs) using code delay measurements are on the order of 5-10 meters and there is no ambiguity in the solution. Typical differential position errors using carrier phase measurements are two orders of magnitude less than the code delay UREs. However, there is an integer ambiguity inherent in using carrier phase measurements to determine differential position that must be resolved. [10]

The problem examined in this thesis, detailed in chapter 4, involves using a displacement vector between two airborne vehicles that is calculated in two coordinate systems to determine the rotation between those coordinate systems. One displacement vector will be calculated in the body coordinate system using *SfM* techniques. The effects of measurement error on this displacement vector in the calculation of the rotation between the two coordinate systems is the focus of chapter 5. The other coordinate system where the displacement vector is calculated is an inertial system. GPS differential carrier phase positioning will be used to calculate this displacement vector and will be treated as truth data in the error analysis given in chapter 5. This chapter is written to show the validity of using GPS differential carrier phase positioning as truth data.

This chapter begins with a general discussion of determining position from the GPS constellation using code delay and then using carrier phase measurements. Next, the process for determining the relative displacement vector for a single moving vehicle is discussed. Finally, the details of how the relative displacement vector between two vehicles is calculated using carrier phase measurements are presented. All material contained in this chapter was originally presented or derived from material given in chapters four and six of an excellent GPS text written by Misra and Enge [10].

3.1 General Discussion of GPS Positioning

There are two types of measurements that can be made to use GPS to determine position. The original design of the GPS system was to estimate the range to at least four satellites using measurements of the pseudo-random code to unambiguously determine the receiver position. The pseudo-random code generated by the satellite is compared to the receiver generated pseudo-random code to produce an estimated transit time. This estimated transit time multiplied by the speed of light in a vacuum determines the range to the satellite that transmitted the signal. The measurements from four satellites are used to determine the 3-D coordinates of the receiver and the receiver clock bias.

It was shown in the late 1970s that a second type of measurement could also be used to determine receiver position [11, 12]. The phase of the carrier signal transmitted by the satellite can be measured relative to a receiver generated carrier signal. The phase difference plus an unknown number of whole cycles gives another estimate of the range to the satellite. Measurements from at least four satellites must still be used to determine receiver position, but now the receiver position contains an integer ambiguity that cannot be resolved in a direct manner. Positioning using both measurement types is discussed in the next two sections.

3.1.1 Positioning with Code Delay Measurements

Positioning with code delay measurements relies on the satellite and receiver being able to create identical signals in time. The receiver-generated signal will then be a delayed version of the transmitted signal. The signal delay is proportional to the distance to the satellite and will be called the transit time. However, in order to compare the signals there must be a common time reference, t. GPS time (GPST) is used as the common time reference. Using the notation as given in Misra and Enge [10], let the transit time be denoted as τ.

Both the receiver and satellite clocks are not precisely aligned with GPST. Let the bias terms relative to GPST for the satellite and receiver be denoted as δt^s and δt_r respectively. Then, noting that t stands for GPST,

$$t^s(t-\tau) = (t-\tau) + \delta t^s(t-\tau) \tag{3-1}$$

$$t_r(t) = t + \delta t_r(t) \ . \tag{3-2}$$

So, the measured range to the satellites, or pseudo-range, is given by

$$\rho(t) = c[t_r(t) - t^s(t-\tau)] = c\,\tau + c[\delta t_r(t) - \delta t^s(t-\tau)] + \epsilon_\rho, \tag{3-3}$$

where ϵ_ρ is used to denote the random pseudo-range estimation error.

The transmission speed of the signal will be significantly less than the speed of light once it enters the earth's atmosphere. The transmission speed can be modeled to take into account the delays resulting from the ionosphere and troposphere by

$$c\,\tau = r(t, t-\tau) + I_\rho(t) + T_\rho(t), \tag{3-4}$$

where $I_\rho(t)$ and $T_\rho(t)$ represent the ionospheric and tropospheric transmission delays and $r(t, t-\tau)$ represents the true range from the receiver to the satellite. Dropping the reference to a specific time in GPST, t, gives the measurement equation for the pseudo-range from the receiver to the satellite as

$$\rho = r + c[\delta t_r - \delta t^s] + I_\rho + T_\rho + \epsilon_\rho\ . \tag{3-5}$$

The accuracy of Equation 3-5 is directly related to how well the bias and error terms are modeled or taken into account. The satellite clock bias at the signal generation time, $\delta t^s(t-\tau)$, is calculated by the GPS ground stations and contained in the navigation message that is sent with the pseudo-random signal. The receiver clock bias, $\delta t_r(t)$, varies for each receiver and can cause significant errors. Satellite ranges are on the order of 20000 - 26000 km, which correspond to transmission times of 70 to 90 ms [10]. So, measurement of an 80 ms transmission time with 1% error, or .8 ms, would result in approximately 240 km of range error. Or, in order to have less than 20 m of range error, the receiver clock bias with respect to GPST must be accurate to within 66.7 μs.

3.1.2 Positioning with Carrier Phase Measurements

A more accurate way of determining the range from a receiver to a satellite is to measure the phase of the carrier signal that carries the pseudo-random code generated from the satellite with respect to the carrier signal that the receiver generates. The phase of the carrier signal can be converted to transit time of the signal by adding the fractional cycle that is given by the phase plus an unknown number of whole cycles. This technique is ambiguous in the number of whole cycles that it takes to determine the range to the satellite.

The cycle, or wavelength, of the carrier signal is 19 cm for the L1 GPS signal and 24 cm for the L2 signal. By comparison, the length of one code chip of the C/A code that is transmitted

on the L1 frequency is 300 m. The different cycles of the carrier and code signals results in different resolutions in the distances that can be calculated from their respective measurements. The accuracy, or resolution, difference between these two types of measurements can be compared to the accuracy difference between making measurements with a ruler that has tick marks every half-centimeter to one that has tick marks every half-meter. In this analogy, the carrier phase measurements would be the ruler with tick marks every half-centimeter and the code phase measurements would be the ruler with tick marks every half-meter.

In an idealized case, the carrier phase measurement (in units of cycles) from a receiver to a satellite can be represented as

$$\phi(t) = \phi_r(t) - \phi^s(t-\tau) + N, \tag{3-6}$$

where the τ represents the transit time of the signal, N is the integer ambiguity, $\phi_r(t)$ represents the phase of the receiver generated signal, and $\phi^s(t-\tau)$ represents the phase of the carrier signal generated by the satellite at time $(t-\tau)$ that is received by the receiver at time t. Simplifying Equation 3-6 by writing phase as the product of frequency and time gives

$$\phi(t) = f\,\tau + N = \frac{c\,\tau}{\lambda} + N = \frac{r(t, t-\tau)}{\lambda} + N, \tag{3-7}$$

where f and λ are the frequency and wavelength of the carrier signal, c is the speed of light in a vacuum, and $r(t, t-\tau)$ is the geometric (or true) range from the receiver to the satellite (same notation as section 3.1.1).

Accounting for the error and bias terms inherent in the measurement equation and dropping the reference to a specific time gives Equation 3-7 to be

$$\phi = \frac{[r + I_\phi + T_\phi] + c\,(\delta t_r - \delta t^s)}{\lambda} + N + \epsilon_\phi, \tag{3-8}$$

where I_ϕ and T_ϕ are the ionospheric and tropospheric delays in meters, δt_r and δt^s account for the clock biases and initial phase offsets of the receiver and satellite clocks respectively, and ϵ_ϕ accounts for all other modeling and measurement errors. Note that Equation 3-8 is in units of cycles.

3.1.3 Positioning with Carrier Phase Absolute and Differential Position with Respect to an External Reference System

Sections 3.1.1 and 3.1.2 described how to use the GPS system to solve for absolute position with respect to an external reference system. Absolute position is the determination of a single receiver's location with respect to an external reference system. With the knowledge of the

absolute position of two receivers, the differential or relative position between those two receivers can be found. However, absolute position is not necessary to determine relative, or differential, position when using carrier phase measurements.

Only differential position is needed for the application in this thesis. This simplifies the processing of the carrier phase measurements. The number of whole cycles between the receiver and the satellite must be determined in order to calculate the receiver position when solving for absolute position using carrier phase measurements. This is normally accomplished by relying on geometric changes in the satellite constellation to rule out all the erroneous solutions to the integer ambiguity until there is only one valid solution. However, when only the differential position between two receivers is needed, as is the case for this application, the number of whole cycles corresponding to what is referred to as the *delta pseudo-range* can be used [10].

The delta pseudo-range is determined by the change in carrier phase measurements over a time interval.

$$\Delta\phi = \phi(t_1) - \phi(t_0) \tag{3-9}$$

If the baseline between the two measurements is small,[1] then the delay due to ionospheric and tropospheric effects on both measurements will be very similar and will essentially cancel out. Also, the satellite clock bias terms, δt^s, will cancel out between the two measurements. This leaves only differences in the two receiver clock bias terms, the true range difference, and the number of whole cycles between the two receivers as the only three unknowns left from Equation 3-8. That is,

$$\Delta\phi = \frac{\Delta r + c\left(\delta t_{r2} - \delta t_{r1}\right)}{\lambda} + N' + \tilde{\epsilon}_\phi \ . \tag{3-10}$$

where Δr is the difference in the ranges to the satellite between the two receivers, δt_{r2} and δt_{r1} represent the clock bias terms for receiver $r2$ and $r1$ respectively, N' is the number of whole cycles between the two measurements, and $\tilde{\epsilon}_\phi$ represents the measurement error. Note that N from Equation 3-8 is much greater than N' from Equation 3-10 as shown in Figure 3–1. Also, note that ϵ_ϕ will be greater than $\tilde{\epsilon}_\phi$ when the baseline between the receivers is very short, so differential position estimates from carrier phase measurements will be even more accurate than absolute position estimates using carrier phase measurements.

[1] Ranges less than 10 km can be classified as small for purposes of ionospheric and tropospheric purposes [10]. So this assumption is especially true for the application in this thesis since baselines will be on the order of tens of meters.

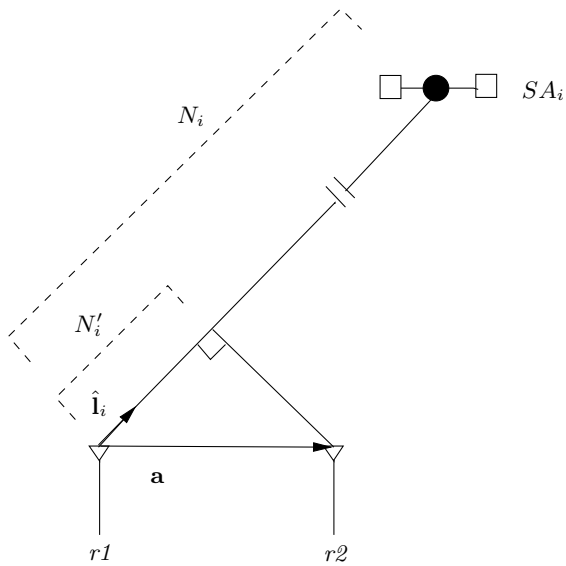

Figure 3–1: Geometry for determination of differential position from carrier phase measurements.

3.1.4 Determination of Differential Position from Carrier Phase Measurements

All the previous discussions in this chapter dealt with a single satellite and one or two receivers. However, in order to determine differential position from carrier phase measurements, the two receivers must track at least four satellites to determine the three position unknowns (x, y, z) in the external reference system and the clock bias term $(\delta t_{r2} - \delta t_{r1})$ for the two receivers. Subscripts will now be used to denote satellite number in all respective symbols.

Referring to Figure 3–1: $\hat{\mathbf{l}}_i$ represents the unit line-of-site (ULOS) vector in the external reference system that points in the direction from receiver $r1$ to the satellite denoted by SA_i, $\mathbf{a}$ represents the vector from receiver $r1$ to receiver $r2$, N_i' represents the number of whole cycles of the carrier signal transmitted by SA_i between the two receivers, and N_i represents the number of whole cycles between receiver $r1$ and SA_i. Using this notation,

$$\Delta r = \hat{\mathbf{l}}_i \cdot \mathbf{a} \ , \tag{3-11}$$

where Δr is the difference in the ranges to the satellite between the two receivers, and the ULOS vector, $\hat{\mathbf{l}}_i$, is a known quantity because the receiver has knowledge of the i-th satellite location in the external coordinate system and it can sense the azimuth and elevation angles to that satellite via its signal. Let $\mathbf{n} = [N_1 \ N_2 \ \ldots \ N_M]^T$ where M represents the number of satellites that are tracked by both $r1$ and $r2$, and $\delta t = (\delta t_{r2} - \delta t_{r1})$. This lets the relationship for determining the differential position between receivers $r1$ and $r2$ using carrier phase measurements from M

satellites as

$$\begin{bmatrix} \triangle\phi_1 \\ \triangle\phi_2 \\ \vdots \\ \triangle\phi_M \end{bmatrix}_{M\times 1} = \begin{bmatrix} \hat{\mathbf{l}}_1^T & 1 \\ \hat{\mathbf{l}}_2^T & 1 \\ \vdots & \vdots \\ \hat{\mathbf{l}}_M^T & 1 \end{bmatrix}_{M\times 4} \begin{bmatrix} \mathbf{a} \\ c\,\delta t \end{bmatrix}_{4\times 1} + \mathbf{n}_{M\times 1} \tag{3-12}$$

Notice that there are, in general, $M + 4$ unknowns in Equation 3-12, so this problem cannot be directly solved as written. Other knowledge of the situation must be used to account for the integer ambiguity and solve for the position and clock bias terms.

3.2 Displacement Vector for One Vehicle from Carrier Phase: Integer Ambiguity Not a Problem

The differential position between a single moving receiver at two different times is simply the displacement vector of the receiver. If the receiver can continuously track a minimum of four satellites for a given duration and count up the number of whole cycles occurring in that duration for each satellite, then Equation 3-12 will have four equations and four unknowns. Therefore, it can be used to solve unambiguously for the differential position and clock bias terms, $[\mathbf{a}^T\ c\delta t]^T$.

If the baseline between the two measurements is on the order of tens of meters, then the ionospheric and tropospheric delays will be practically identical and the accuracy of the differential position vector $\mathbf{a}$ will be given by the accuracy of the phase measurements. The carrier phase can typically be measured with an accuracy of 0.01-0.05 cycles (2 mm - 1 cm) [10]. This means that by continuously measuring the phase of a minimum of four satellites for a given duration, differential position (i.e., the $\mathbf{a}$ vector in an external reference system) can be estimated with millimeter-to-centimeter accuracy. The simple method which achieves this accuracy is quite remarkable!

3.3 Displacement Vector for Two Vehicles from Carrier Phase: Integer Ambiguity is a Problem

In order to use carrier phase measurements to calculate the displacement vector between two different receivers, the integer ambiguity must be resolved. The time it takes to resolve the integer ambiguity, or initialization time, is what needs to be reduced in order to use carrier phase measurements for airborne vehicle navigation. There are several methods for resolving the integer ambiguity in differential carrier phase positioning available in present technology. Three methods are discussed in the following sections: Real time kinematic (RTK), Wide laning, and Kalman filtering. However, once the integer ambiguity is resolved correctly, the differential position is still

in the millimeter-to-centimeter accuracy just as the single vehicle case. The only difference is that the two vehicle case requires an initialization time to resolve the integer ambiguity problem.

3.3.1 Real Time Kinematic (RTK) with Roving Reference Receiver

Differential GPS (DGPS) is a proven technique that takes advantage of the slowly changing nature of the standard GPS errors. If two receivers are within a reasonable distance from each other, differencing the correlated errors can reduce their effects and result in a more accurate position estimate. A mode available from many receiver manufacturers is called Real Time Kinematic (RTK). RTK mode combines DGPS and carrier phase measurements to produce precise relative positioning. RTK mode uses reference and rover receivers, a communication link, and software to precisely determine the rover receiver's position. The initialization time for integer ambiguity resolution for baselines of several kilometers (for 2001 technology) is 30-60 s [10]. Once the initialization time is complete, the two receivers must continue to track the same set of satellites to have knowledge of the *integer ambiguity*. There is no reason why both the reference and rover receivers cannot be moving.

3.3.2 Wide Laning Using Dual Frequency

It can be shown that the total number of integer values which satisfy the integer ambiguity decreases as the carrier wavelength increases [10]. There are two related effects which cause this to occur: 1) increased wavelength decreases the number of nodes in the search space of possible solutions, and 2) increased wavelength increases the distance between adjacent nodes in the search space. The GPS satellites are currently transmitting signals at two frequencies: L1 and L2 (1 cycle = 19 cm at L1 and 24 cm at L2). So, in order to improve integer estimation, the wide laning method combines both the L1 and L2 frequencies to "create" a new signal with a longer wavelength. The new signal, L12, results in a wavelength of 0.862 m [10]. This results in a more accurate estimate of the integer ambiguities, but sacrifices accuracy in the estimation of receiver position.

3.3.3 Kalman Filter

A Kalman filter approach can also be used to estimate the integer ambiguities. Both code and carrier measurements are used in this method. The course code measurements are used to give a guess at the initial position. Then, the integer ambiguities are treated as floating point states in a Kalman filter. Whenever the "integers" converge (i.e., the filter approaches steady state), they are rounded to the nearest integer to determine the solution.

CHAPTER 4
PROBLEM DEFINITION AND SOLUTION

The usual solution to the reconstruction problem contains both motion and structure components. However, for the purpose of vehicle attitude determination, the structure component is not required. The motion solution component, specifically the translation vector between two camera systems, $\mathbf{a}^{c1}$, when combined with the GPS differential carrier phase positioning between the two systems, $\mathbf{a}^{LL}$, gives all the information needed to know the inertial attitude of the vehicle up to rotation about the translation axis (this limitation is explained further in the next section). Stating the problem in a more formal form: for n point correspondences between two camera systems, *c1* and *c2*, and an inertial translational vector $\mathbf{a}^{LL}$, find the inertial orthonormal rotation matrix R_{LL}^{c1} or equivalently R_{LL}^{c2}. Two solution approaches are given based on section 6.1 of Maybank's text [1]: an SVD based approach and an error minimization approach. The SVD based approach is given a complete solution, while the error minimization approach is dealt with in general terms.

4.1 Problem Notes and Assumptions

4.1.1 Rotation Ambiguity about Translational Axis

The inertial rotation around the translational axis between the two camera systems cannot be found from the information contained in the problem statement. This can be illustrated by looking at a trivial example: If camera one and camera two were both pointed due north and their displacement was also due north, then using a 1-2-3 (or roll-pitch-yaw) Euler angle representation of the orthonormal rotation matrix would mean that the roll is completely ambiguous. Likewise, if the pointing and displacement vector were in any other direction comprised of more than one vector component, then the roll-pitch-yaw angles would be somehow interrelated. In other words, using only two vision systems in the formulation of this problem allows solution for only two degrees of freedom. A third vision system would need to be added to allow solution for a third degree of freedom.

4.1.2 No GPS Attitude Determination Available

GPS attitude determination will not be available on the vehicle of interest. This assumption is reasonable because the size of the vehicle being considered in this application does not support

a long enough baseline to provide accurate measurements. A baseline of 0.5-5 meters is needed to provide useful results from GPS attitude determination [13].

A second approach to GPS attitude determination is to use successive positional measurements and make the assumption that the vehicle attitude corresponds directly to the direction of motion. This would be possible with a single GPS receiver and could use the accurate results associated with the differential carrier phase measurements; however, as soon as the vehicle's flight path contained any rotational velocity, or any non-zero side-slip or angle-of-attack, this approach produces fundamentally erroneous results.

4.1.3 Specification of Control Points

Control points can be specified or tracked between images such that the projections of the point correspondences onto the focal plane are known. How the control points are identified is outside the scope of this thesis. This is a classic recognition or tracking problem in image processing and all that is assumed in this thesis is that the control point values (i.e., c and r) can be measured with known error statistics.

4.1.4 GPS Differential Carrier Phase Positioning Data Used as Truth

The errors associated with the GPS differential carrier phase positioning are considered negligible compared to other system errors and these position estimates are therefore used as truth data. This is reasonable since the baseline used in this problem is 10 meters and the errors associated with the GPS measurements are on the order of centimeters, thus corresponding to angular errors on the order of 5 mrad (e.g., $\tan^{-1}(\frac{5cm}{10m})$). See chapter 3 for detailed discussion of this assumption.

4.2 Specific *SfM* Problem Setup to Find Vehicle Inertial Attitude

The same imaging system is used as given in Chapter 2, noting that the *3-D* scene coordinate system used here is commonly called the camera or sensor system in navigational terms. Figure 4–1 is a reproduction of the corresponding figure in Chapter 2, where the following equations remain valid:

$$c = f\,\frac{x}{z}, \tag{4-1}$$

$$r = f\,\frac{y}{z}. \tag{4-2}$$

For the *2D-to-2D SfM* solution, two different views or images of the same scene with n correspondences are used to reconstruct the *3-D* scene in either one of the camera *3-D* coordinate

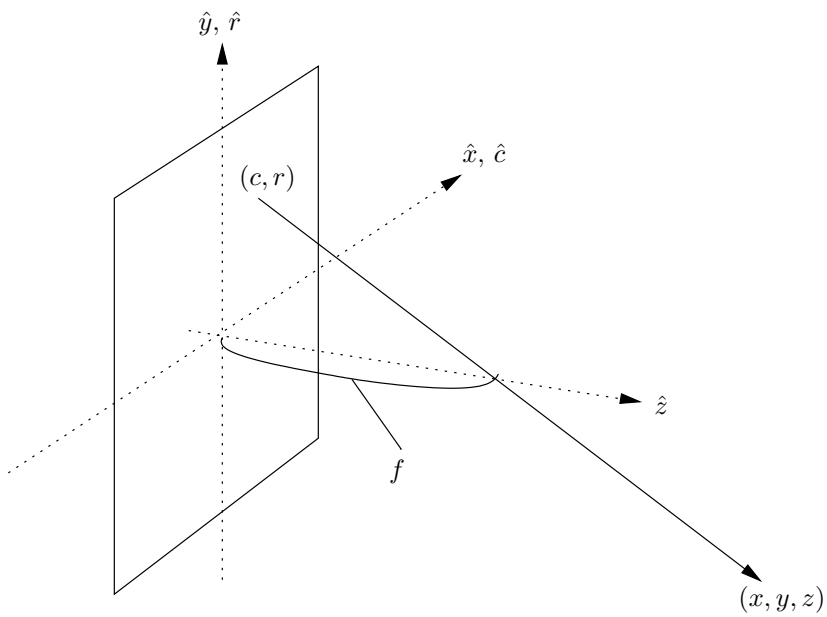

Figure 4–1: Imaging system notation used in finding vehicle's inertial attitude.

systems. Using the notation given earlier, the translation and rotation between the two camera systems is $\mathbf{a}^{c1}$ and R_{c1}^{c2} respectively. The origins of the two camera systems are given by $\mathbf{o1}^{LL}$ and $\mathbf{o2}^{LL}$, where LL stands for a local level coordinate system such as North-East-Down (*NED*) or East-North-Up (*ENU*). These values cannot be obtained from the imaging system, but are supplied via the GPS sensor and used as truth data.[1] The lines $\mathbf{p}^{c1}$ and $\mathbf{q}^{c2}$ represent the *3-D* scene coordinates of one correspondence expressed in the first and second camera system respectively. This is shown in Figure 4–2. Note that it is arbitrary which *3-D* camera system is called *c1* and which is called *c2* when solving this problem.

4.3 Singular Value Decomposition (SVD) Solution

The SVD solution approach to the vehicle attitude determination from point correspondences problem is adapted from section 6.1.1 of Maybank's text [1]. The reader is referred to this text if there are any areas of this algorithm that are not presented to the level of detail that is desired.

Let the point correspondence pairs, $\hat{\mathbf{p}}_i^{c1}$ and $\hat{\mathbf{q}}_i^{c2}$ where $i = 1...n$ such that $n \geq 9$, be described by the Euclidean framework notation utilizing the essential matrix E described earlier such that

$$E = R_{c1}^{c2} T_{a^{c1}} \tag{4-3}$$

[1] see section 4.1.4 for details.

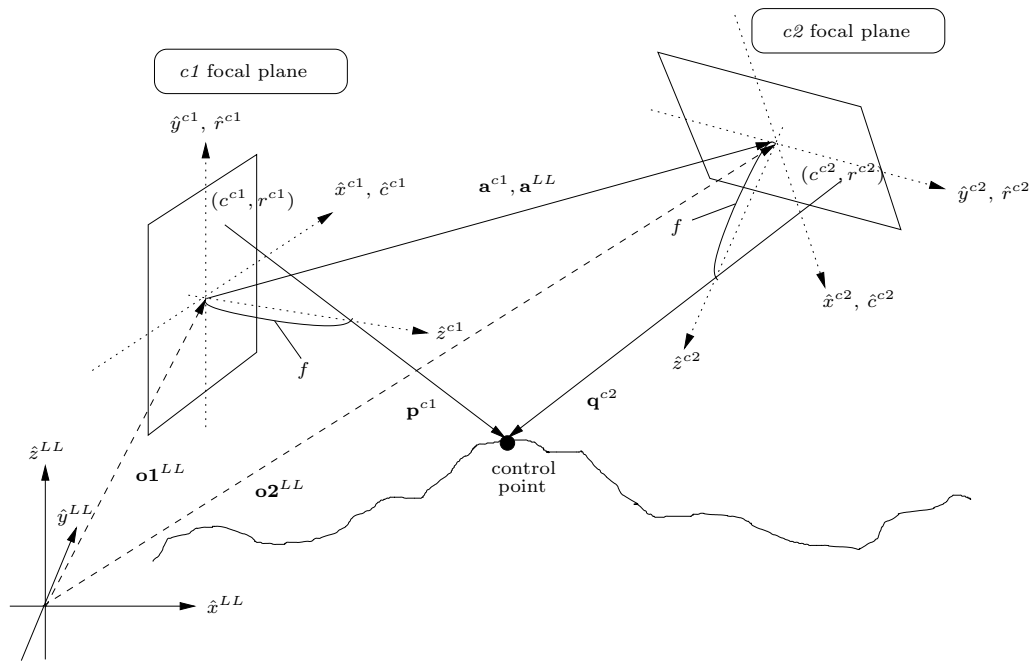

Figure 4–2: Single *2D-to-2D* point correspondence used in finding vehicle's inertial attitude.

where the translation and rotation between the camera systems are given by $\mathbf{a}^{c1}$ and R_{c1}^{c2} respectively. An error function $V(E)$ is defined on the space of 3×3 matrices such that

$$V(E) = \sum_{i=1}^{n} ((\hat{\mathbf{q}}_i^{c2})^T \, E \, \hat{\mathbf{p}}_i^{c1})^2 \tag{4-4}$$

Find the 3×3 matrix $\widehat{E}$ with unit Frobenius norm that minimizes $V(\widehat{E})$, and then find the nearest essential matrix to $\widehat{E}$ (the unit Frobenius norm constraint is imposed to exclude the trivial solution $\widehat{E} = 0$).[2] An outline of the SVD solution is given below, with details following background sections on some of the mathematical details needed for the solution.

- Form the $n \times 9$ matrix A from the n point correspondence pairs.
- Perform SVD on A via $A = U \Sigma V^T$
- Set $\widehat{\mathbf{e}}$ = last column of V
- Reshape $\widehat{\mathbf{e}}$ into the estimate of the essential matrix $\widehat{E}$.
- Perform SVD on $\widehat{E}$ via $\widehat{E} = U' \Sigma' V'^T$
- Set $\widehat{\hat{\mathbf{a}}}^{c1}$ = last column of V'
- Define $\mathbf{a}^{LL} = \mathbf{o2}^{LL} - \mathbf{o1}^{LL}$

[2] Note that the wide hat accent, $\widehat{E}$, is used to denote an estimate of a variable. This is not to be confused with the hat accent symbol used to denote a unit vector, $\hat{\mathbf{p}}$. With this nomenclature, $\widehat{\hat{\mathbf{a}}}^c$ indicates an estimate of the unit vector $\hat{\mathbf{a}}^c$.

- Determine a second linearly independent vector that is expressed in both the *c1* and *LL* coordinate systems ($\hat{\hat{\mathbf{b}}}^{c1}$ and $\mathbf{b}^{LL}$).
- Calculate a third linearly independent vector that is expressed in both the *c1* and *LL* coordinate systems by taking the cross product between (1) $\hat{\hat{\mathbf{b}}}^{c1}$ and $\hat{\hat{\mathbf{a}}}^{c1}$, and (2) $\hat{\mathbf{b}}^{LL}$ and $\hat{\mathbf{a}}^{LL}$.
- Use the three vectors expressed in both the *c1* and *LL* coordinate systems to compute R^{c1}_{LL}.

The following subsections give a brief background in the linear algebraic mathematical details needed before describing the outlined steps given above.

4.3.1 Brief Overview of Linear Algebra Techniques used in SVD Solution

Singular value decomposition. Any $m \times n$ matrix A can be written as the product of an $m \times n$ column-orthogonal matrix U, an $n \times n$ diagonal matrix Σ with positive or zero elements, and the transpose of an $n \times n$ orthogonal matrix V [14]

$$A = U\,\Sigma\,V^T \tag{4-5}$$

where

$$\Sigma = diag(\sigma_1\sigma_2...\sigma_{n-1}\sigma_n) \text{ for } \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{n-1} \geq \sigma_n \geq 0 \tag{4-6}$$

and

$$U^T\,U = I \tag{4-7}$$

$$V^T\,V = I \tag{4-8}$$

Matrix norm calculations. There are two matrix normalization calculations used in this thesis. As given in Maybank's text [1], the Euclidean norm, $\|.\|$, is subordinate to the Euclidean vector norm and is defined as

$$\|A\| = \sup(\|A\mathbf{x}\| \mid \|\mathbf{x}\| = 1), \tag{4-9}$$

where A is an $m \times n$ matrix. The Frobenius norm, $\|.\|_f$, is defined as

$$\|A\|_f^2 = \sum_{i=1,j=1}^{m,n} A_{ij}^2. \tag{4-10}$$

Performing the SVD in each of these matrix norms results in [1]

$$\|A\| = \|U\Sigma V\| = \|\Sigma\| = \sigma_1, \tag{4-11}$$

$$\|A\|_f^2 = \|U\Sigma V\|_f^2 = \|\Sigma\|_f^2 = \sum_{i=1}^{k} \sigma_i^2. \tag{4-12}$$

4.3.2 Detailed Look at SVD Solution Algorithm

Form the $n \times 9$ A matrix. Looking at a single point correspondence pair, $\hat{\mathbf{p}}^{c1}$ and $\hat{\mathbf{q}}^{c2}$, and writing out the essential matrix formulation in its component form gives

$$\begin{bmatrix} \hat{q}_x^{c2} & \hat{q}_y^{c2} & \hat{q}_z^{c2} \end{bmatrix} \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} \hat{p}_x^{c1} \\ \hat{p}_y^{c1} \\ \hat{p}_z^{c1} \end{bmatrix} = 0 \tag{4-13}$$

Writing Equation 4-13 in scalar form gives:

$$\begin{aligned} \hat{q}_x^{c2}\hat{p}_x^{c1}E_{11} + \hat{q}_y^{c2}\hat{p}_x^{c1}E_{21} + \hat{q}_z^{c2}\hat{p}_x^{c1}E_{31} + \hat{q}_x^{c2}\hat{p}_y^{c1}E_{12} + \hat{q}_y^{c2}\hat{p}_y^{c1}E_{22} + \hat{q}_z^{c2}\hat{p}_y^{c1}E_{32} \\ + \hat{q}_x^{c2}\hat{p}_z^{c1}E_{13} + \hat{q}_y^{c2}\hat{p}_z^{c1}E_{23} + \hat{q}_z^{c2}\hat{p}_z^{c1}E_{33} = 0\ . \end{aligned} \tag{4-14}$$

Letting the $\hat{\mathbf{p}}^{c1}$ and $\hat{\mathbf{q}}^{c2}$ terms form a row of the $n \times 9$ A matrix. The i-th row of A is then given by

$$A_i \triangleq \begin{bmatrix} \hat{q}_{ix}^{c2}\hat{p}_{ix}^{c1} & \hat{q}_{iy}^{c2}\hat{p}_{ix}^{c1} & \hat{q}_{iz}^{c2}\hat{p}_{ix}^{c1} & \hat{q}_{ix}^{c2}\hat{p}_{iy}^{c1} & \hat{q}_{iy}^{c2}\hat{p}_{iy}^{c1} & \hat{q}_{iz}^{c2}\hat{p}_{iy}^{c1} & \hat{q}_{ix}^{c2}\hat{p}_{iz}^{c1} & \hat{q}_{iy}^{c2}\hat{p}_{iz}^{c1} & \hat{q}_{iz}^{c2}\hat{p}_{iz}^{c1} \end{bmatrix}. \tag{4-15}$$

Rewrite the E terms as a 9×1 column vector as

$$\mathbf{e} \triangleq \begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{21} \\ E_{22} \\ E_{23} \\ E_{31} \\ E_{32} \\ E_{33} \end{bmatrix}. \tag{4-16}$$

So for nine point correspondences the corresponding matrix equation is

$$\begin{bmatrix} \hat{q}^{c2}_{1x}\hat{p}^{c1}_{1x} & \hat{q}^{c2}_{1y}\hat{p}^{c1}_{1x} & \hat{q}^{c2}_{1z}\hat{p}^{c1}_{1x} & \hat{q}^{c2}_{1x}\hat{p}^{c1}_{1y} & \hat{q}^{c2}_{1y}\hat{p}^{c1}_{1y} & \hat{q}^{c2}_{1z}\hat{p}^{c1}_{1y} & \hat{q}^{c2}_{1x}\hat{p}^{c1}_{1z} & \hat{q}^{c2}_{1y}\hat{p}^{c1}_{1z} & \hat{q}^{c2}_{1z}\hat{p}^{c1}_{1z} \\ \hat{q}^{c2}_{2x}\hat{p}^{c1}_{2x} & \hat{q}^{c2}_{2y}\hat{p}^{c1}_{2x} & \hat{q}^{c2}_{2z}\hat{p}^{c1}_{2x} & \hat{q}^{c2}_{2x}\hat{p}^{c1}_{2y} & \hat{q}^{c2}_{2y}\hat{q}^{c1}_{2y} & \hat{q}^{c2}_{2z}\hat{p}^{c1}_{iy} & \hat{q}^{c2}_{2x}\hat{p}^{c1}_{2z} & \hat{q}^{c2}_{2y}\hat{p}^{c1}_{2z} & \hat{q}^{c2}_{2z}\hat{p}^{c1}_{2z} \\ \hat{q}^{c2}_{3x}\hat{p}^{c1}_{3x} & \hat{q}^{c2}_{3y}\hat{p}^{c1}_{3x} & \hat{q}^{c2}_{3z}\hat{p}^{c1}_{3x} & \hat{q}^{c2}_{3x}\hat{p}^{c1}_{3y} & \hat{q}^{c2}_{3y}\hat{p}^{c1}_{3y} & \hat{q}^{c2}_{3z}\hat{p}^{c1}_{3y} & \hat{q}^{c2}_{3x}\hat{p}^{c1}_{3z} & \hat{q}^{c2}_{3y}\hat{p}^{c1}_{3z} & \hat{q}^{c2}_{3z}\hat{p}^{c1}_{3z} \\ \hat{q}^{c2}_{4x}\hat{p}^{c1}_{4x} & \hat{q}^{c2}_{4y}\hat{p}^{c1}_{4x} & \hat{q}^{c2}_{4z}\hat{p}^{c1}_{4x} & \hat{q}^{c2}_{4x}\hat{p}^{c1}_{4y} & \hat{q}^{c2}_{4y}\hat{p}^{c1}_{4y} & \hat{q}^{c2}_{4z}\hat{p}^{c1}_{4y} & \hat{q}^{c2}_{4x}\hat{p}^{c1}_{4z} & \hat{q}^{c2}_{4y}\hat{p}^{c1}_{4z} & \hat{q}^{c2}_{4z}\hat{p}^{c1}_{4z} \\ \hat{q}^{c2}_{5x}\hat{p}^{c1}_{5x} & \hat{q}^{c2}_{5y}\hat{p}^{c1}_{5x} & \hat{q}^{c2}_{5z}\hat{p}^{c1}_{5x} & \hat{q}^{c2}_{5x}\hat{p}^{c1}_{5y} & \hat{q}^{c2}_{5y}\hat{p}^{c1}_{5y} & \hat{q}^{c2}_{5z}\hat{p}^{c1}_{5y} & \hat{q}^{c2}_{5x}\hat{p}^{c1}_{5z} & \hat{q}^{c2}_{5y}\hat{p}^{c1}_{5z} & \hat{q}^{c2}_{5z}\hat{p}^{c1}_{5z} \\ \hat{q}^{c2}_{6x}\hat{p}^{c1}_{6x} & \hat{q}^{c2}_{6y}\hat{p}^{c1}_{6x} & \hat{q}^{c2}_{6z}\hat{p}^{c1}_{6x} & \hat{q}^{c2}_{6x}\hat{p}^{c1}_{6y} & \hat{q}^{c2}_{6y}\hat{p}^{c1}_{6y} & \hat{q}^{c2}_{6z}\hat{p}^{c1}_{6y} & \hat{q}^{c2}_{6x}\hat{p}^{c1}_{6z} & \hat{q}^{c2}_{6y}\hat{p}^{c1}_{6z} & \hat{q}^{c2}_{6z}\hat{p}^{c1}_{6z} \\ \hat{q}^{c2}_{7x}\hat{p}^{c1}_{7x} & \hat{q}^{c2}_{7y}\hat{p}^{c1}_{7x} & \hat{q}^{c2}_{7z}\hat{p}^{c1}_{7x} & \hat{q}^{c2}_{7x}\hat{p}^{c1}_{7y} & \hat{q}^{c2}_{7y}\hat{p}^{c1}_{7y} & \hat{q}^{c2}_{7z}\hat{p}^{c1}_{7y} & \hat{q}^{c2}_{7x}\hat{p}^{c1}_{7z} & \hat{q}^{c2}_{7y}\hat{p}^{c1}_{7z} & \hat{q}^{c2}_{7z}\hat{p}^{c1}_{7z} \\ \hat{q}^{c2}_{8x}\hat{p}^{c1}_{8x} & \hat{q}^{c2}_{8y}\hat{p}^{c1}_{8x} & \hat{q}^{c2}_{8z}\hat{p}^{c1}_{8x} & \hat{q}^{c2}_{8x}\hat{p}^{c1}_{8y} & \hat{q}^{c2}_{8y}\hat{p}^{c1}_{8y} & \hat{q}^{c2}_{8z}\hat{p}^{c1}_{8y} & \hat{q}^{c2}_{8x}\hat{p}^{c1}_{8z} & \hat{q}^{c2}_{8y}\hat{p}^{c1}_{8z} & \hat{q}^{c2}_{8z}\hat{p}^{c1}_{8z} \\ \hat{q}^{c2}_{9x}\hat{p}^{c1}_{9x} & \hat{q}^{c2}_{9y}\hat{p}^{c1}_{9x} & \hat{q}^{c2}_{9z}\hat{p}^{c1}_{9x} & \hat{q}^{c2}_{9x}\hat{p}^{c1}_{9y} & \hat{q}^{c2}_{9y}\hat{p}^{c1}_{9y} & \hat{q}^{c2}_{9z}\hat{p}^{c1}_{9y} & \hat{q}^{c2}_{9x}\hat{p}^{c1}_{9z} & \hat{q}^{c2}_{9y}\hat{p}^{c1}_{9z} & \hat{q}^{c2}_{9z}\hat{p}^{c1}_{9z} \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ E_{21} \\ E_{22} \\ E_{23} \\ E_{31} \\ E_{32} \\ E_{33} \end{bmatrix} = 0. \tag{4-17}$$

Equation 4-17 can be represented compactly for the set of n point correspondence pairs by

$$A_{n\times 9}\,\mathbf{e}_{9\times 1} = \mathbf{0}_{n\times 1}\ . \tag{4-18}$$

Note that the $\|\mathbf{e}\| = 1$ constraint is placed on $\mathbf{e}$ to exclude the trivial solution.

Perform SVD on A. From the above reformulation of the reconstruction problem, it follows that

$$((\hat{\mathbf{q}}^{c2}_i)^T\, E\, \hat{\mathbf{p}}^{c1}_i)^2 = (A_i\,\mathbf{e})^2 \tag{4-19}$$

and also that

$$V(E) = \|A\,\mathbf{e}\|^2 \tag{4-20}$$

Then, performing the singular value decomposition on A gives

$$V(E) = \|U\Sigma V^T\mathbf{e}\| = \|\Sigma V^T\mathbf{e}\| \geq \sigma_9^2\|V^T\mathbf{e}\| = \sigma_9^2\|\mathbf{e}\| = \sigma_9^2 \tag{4-21}$$

Noting that the constraint $\|\mathbf{e}\| = 1$ is invoked in the last step.

Set $\widehat{\mathbf{e}}$ = last column of V. So, the 3×3 matrix that minimizes the error function described by Equation 4-4 can be found from the 9×1 column vector

$$\widehat{\mathbf{e}} = V^T\,\mathbf{f}_9 \tag{4-22}$$

,

where $\mathbf{f}_9$ is the nine dimensional vector defined as $(0, 0, \ldots, 1)^T$. In the absence of noise, $\mathbf{e} = \widehat{\mathbf{e}}$. The consequence of this result is shown later to be that $V(\widehat{E}) = V(E) = 0$.

Reshape $\widehat{\mathbf{e}}$ into the essential matrix $\widehat{E}$. $\widehat{E}$ is found from $\widehat{\mathbf{e}}$ via

$$\widehat{E} = \begin{bmatrix} \widehat{e}_1 & \widehat{e}_2 & \widehat{e}_3 \\ \widehat{e}_4 & \widehat{e}_5 & \widehat{e}_6 \\ \widehat{e}_7 & \widehat{e}_8 & \widehat{e}_9 \end{bmatrix} \tag{4-23}$$

where $V(\widehat{E}) = \sigma_9$ which was shown before to be the minimum value. If $\sigma_9 = 0$, then $V(\widehat{E}) = V(E) = 0$ and therefore $\widehat{E} = E$.

Perform SVD on $\widehat{E}$. At this point in the algorithm, Maybank [1] discusses another minimization function using the least eigenvalue to derive rotation and translation terms from the essential matrix. The problem described in this thesis requires only the translational term, $\mathbf{a}^{c1}$, to form a solution for the vehicle attitude. That is,

$$E\,\mathbf{a}^{c1} = R_{c1}^{c2}\,T_{a^{c1}}\mathbf{a}^{c1} = R_{c1}^{c2}(\mathbf{a}^{c1} \times \mathbf{a}^{c1}) = R_{c1}^{c2}\,\mathbf{0} = \mathbf{0}. \tag{4-24}$$

Therefore, $\mathbf{a}^{c1}$ is in the null space of E. This results in the need to perform the singular value decomposition on $\widehat{E}$ such that

$$\widehat{E} = U'\Sigma'V'^T. \tag{4-25}$$

Set $\widehat{\hat{\mathbf{a}}}^{c1}$ = last column of V'^T. Using the same logic as finding $\widehat{\mathbf{e}}$,

$$\widehat{\hat{\mathbf{a}}}^{c1} = V'^T\mathbf{s}_3, \tag{4-26}$$

where $\mathbf{s}_3$ is defined as the three-dimensional vector $(0, 0, 1)^T$. Note that in the absence of noise, $\widehat{\hat{\mathbf{a}}}^{c1} = \hat{\mathbf{a}}^{c1}$, as would be expected.

This concludes the *SfM* contribution to this algorithm. The remaining steps are a consequence of using the available information from the GPS sensor and matrix manipulation via linear algebra.

Define $\mathbf{a}^{LL} = \mathbf{o2}^{LL} - \mathbf{o1}^{LL}$. Use GPS differential carrier phase positioning to mark the origin of each camera system, *c1* and *c2*. The translation displacement vector can then be obtained via

$$\mathbf{a}^{LL} = \mathbf{o2}^{LL} - \mathbf{o1}^{LL}. \tag{4-27}$$

Determine a second linearly independent vector that is expressed in both the *c1* and *LL* coordinate systems ($\widehat{\hat{\mathbf{b}}}^{c1}$ and $\mathbf{b}^{LL}$). There are at least two ways to find a second linearly independent vector that is expressed in both the *c1* and *LL* coordinate systems. The first way is to repeat steps (1) - (7) above for a second camera system pair *c1* and *c3* to find

$\widehat{\hat{\mathbf{b}}}^{c1}$ and $\mathbf{b}^{LL}$. The second possibility is to use the down vector obtained from the onboard inertial navigation system (INS) as the second vector expressed in both coordinate systems. This would take advantage of the down vector in LL as being $(0,0,1)_{NED}$ or $(0,0,-1)_{ENU}$. Whichever of the two methods is a more convenient choice should be used.

Derive a third linearly independent vector that is expressed in both the *c1* and *LL* coordinate systems by taking the cross product between (1) $\widehat{\hat{\mathbf{b}}}^{c1}$ and $\widehat{\hat{\mathbf{a}}}^{c1}$, and (2) $\hat{\mathbf{b}}^{LL}$and $\hat{\mathbf{a}}^{LL}$. Consider the two vectors used in the cross product as inputs and the resulting vector as output. If the two input vectors are linearly independent, then from the definition of a cross product, the output vector is linearly independent of both input vectors.

This result can be exploited to reduce the number of calculated vectors expressed in both the *c1* and *LL* coordinate systems from three to two. The third linearly independent vector can be derived from the first two vectors via

$$\widehat{\hat{\mathbf{c}}}^{c1} = \widehat{\hat{\mathbf{b}}}^{c1} \times \widehat{\hat{\mathbf{a}}}^{c1} \tag{4-28}$$

$$\hat{\mathbf{c}}^{LL} = \hat{\mathbf{b}}^{LL} \times \hat{\mathbf{a}}^{LL}. \tag{4-29}$$

Use the three vectors expressed in both the $c1$ and LL coordinate systems to compute R_{LL}^{c1} . Let each column in the 3×3 matrix C be defined as a linear independent vector expressed in the $c1$ coordinate system. From the results obtained earlier C is comprised as

$$C \triangleq \begin{bmatrix} \widehat{\hat{c}}_x^{c1} & \widehat{\hat{b}}_x^{c1} & \widehat{\hat{a}}_x^{c1} \\ \widehat{\hat{c}}_y^{c1} & \widehat{\hat{b}}_y^{c1} & \widehat{\hat{a}}_y^{c1} \\ \widehat{\hat{c}}_z^{c1} & \widehat{\hat{b}}_z^{c1} & \widehat{\hat{a}}_z^{c1} \end{bmatrix}. \tag{4-30}$$

And similarly, let each column of the 3×3 matrix L be defined as the corresponding vectors used to populate C expressed in the LL coordinate system.

$$L \triangleq \begin{bmatrix} \hat{c}_x^{LL} & \hat{b}_x^{LL} & \hat{a}_x^{LL} \\ \hat{c}_y^{LL} & \hat{b}_y^{LL} & \hat{a}_y^{LL} \\ \hat{c}_z^{LL} & \hat{b}_z^{LL} & \hat{a}_z^{LL} \end{bmatrix}. \tag{4-31}$$

Then, the rotation matrix R_{LL}^{c1} can be determined via

$$R_{LL}^{c1} = C\,L^{-1}. \tag{4-32}$$

This solution for R_{LL}^{c1} will be unique if the three vectors comprising C and L are linearly independent. Stated another way, if both L and C are of full rank, then R_{LL}^{c1} will be a unique solution.

4.4 Error Minimization Solution Approach

Another approach for finding the inertial attitude of a vehicle utilizing Euclidean *SfM* techniques is to find the minimum of an error equation using a descent or adaptive algorithm. The descent algorithm is an iterative approach to a solution. It searches the neighborhood around an initial point to determine a new point where the value of the error function is less than the initial point. The error function is approximated by a Taylor series expansion to simplify the search algorithm. This process repeats until a minimum of the error function is found.

Three solution approaches based on descent algorithms are presented. The first two approaches use the coplanarity constraint to obtain an error equation that can be minimized. The third approach minimizes the system error vector from the linearized measurement model derived in the next chapter. All three solutions are discussed in general terms. The two approaches utilizing the coplanarity constraint were originally given by Horn [15] and also discussed by Maybank [1]. Development of the third approach which minimizes the state error vector is left as a future area of study. All three approaches are discussed in a more general manner than the SVD solution given in the previous section.

4.4.1 Brief Overview of Rotations using Quaternions

The descent solution algorithms given in the section 4.4.2 make the subtle change of using quaternions to represent the rotation from the inertial to body coordinate system. This section gives a brief overview of the use of quaternions to represent rotations. The following information is derived from lecture notes given by E. Sutton at the University of Florida Graduate Engineering and Research Center in Shalimar, FL. Further information on quaternions can be found in Fraleigh [16] and Zipfel [17].

Quaternions. Quaternions can be thought of as a four element vector or as a sum:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{k} + q_4 \tag{4-33}$$

where $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ obey the following relationships: $\hat{\imath}\hat{\jmath} = \hat{k}$, $\hat{\jmath}\hat{k} = \hat{\imath}$, $\hat{k}\hat{\imath} = \hat{\jmath}$, $\hat{\jmath}\hat{\imath} = -\hat{k}$, $\hat{k}\hat{\jmath} = -\hat{\imath}$, and $\hat{\imath}\hat{k} = -\hat{\jmath}$. Quaternion multiplication (or composition) is defined as follows:

$$\mathbf{a} \cdot \mathbf{b} = \left(a_1\hat{\imath} + a_2\hat{\jmath} + a_3\hat{k} + a_4\right) \cdot \left(b_1\hat{\imath} + b_2\hat{\jmath} + b_3\hat{k} + b_4\right) \tag{4-34}$$

Expanding the terms on the right side of the Equation 4-34 and grouping by the $\hat{\imath}$, $\hat{\jmath}$, $\hat{k}$, and scalar parts give

$$\begin{aligned}\mathbf{a} \cdot \mathbf{b} = {} & (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)\,\hat{\imath} + (-a_1b_3 + a_2b_4 + a_3b_1 + a_4b_2)\,\hat{\jmath} \\ & + (a_1b_2 - a_2b_1 + a_3b_4 + a_4b_3)\,\hat{k} - (a_1b_1 - a_2b_2 - a_3b_3 + a_4b_4)\ .\end{aligned} \tag{4-35}$$

Quaternion conjugation is defined by

$$\mathbf{q}^* = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix} = -q_1\hat{\imath} - q_2\hat{\jmath} - q_3\hat{k} + q_4 \tag{4-36}$$

The multiplicative inverse is given by

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \tag{4-37}$$

where

$$\|\mathbf{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \tag{4-38}$$

Quaternion multiplication can also be performed using two equivalent matrix-vector forms:

$$\mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} a_4 & -a_3 & a_2 & a_1 \\ a_3 & a_4 & -a_1 & a_2 \\ -a_2 & a_1 & a_4 & a_3 \\ -a_1 & -a_2 & -a_3 & a_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} b_4 & b_3 & -b_2 & b_1 \\ -b_3 & b_4 & b_1 & b_2 \\ b_2 & -b_1 & b_4 & b_3 \\ -b_1 & -b_2 & -b_3 & b_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \tag{4-39}$$

Quaternions obey the following properties:

$$(\mathbf{q}^*)^* = \mathbf{q} \tag{4-40}$$

$$\|\mathbf{q}\|^2 = \mathbf{q}^* \cdot \mathbf{q} = \mathbf{q} \cdot \mathbf{q}^* \tag{4-41}$$

$$(\mathbf{p} \cdot \mathbf{q})^* = \mathbf{q}^* \cdot \mathbf{p}^* \tag{4-42}$$

$$\|\mathbf{p}\cdot\mathbf{q}\| = \|\mathbf{p}\|\,\|\mathbf{q}\| \tag{4-43}$$

$$(\mathbf{p}\cdot\mathbf{q})^{-1} = \mathbf{q}^{-1}\cdot\mathbf{p}^{-1} \tag{4-44}$$

Quaternions representing rotations. Quaternions can be used to represent rotations:

$$\mathbf{q} = \begin{bmatrix} \mathbf{e}\sin\frac{\beta}{2} \\ \cos\frac{\beta}{2} \end{bmatrix} = \begin{bmatrix} e_x\sin\frac{\beta}{2} \\ e_y\sin\frac{\beta}{2} \\ e_z\sin\frac{\beta}{2} \\ \cos\frac{\beta}{2} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4-45}$$

where $\mathbf{e} = (e_x,\, e_y,\, e_z)$ is the axis of rotation and β is the angle of rotation. A quaternion that represents a rotation must have unit length ($\|\mathbf{q}\| = 1$). Also, note that $-\mathbf{q}$ represents the same rotation as $\mathbf{q}$.

A rotation from coordinate system x to coordinate system y is accomplished as follows:

$$\mathbf{r}^y = \mathbf{q}_x^y\cdot\mathbf{r}^x\cdot(\mathbf{q}_x^y)^* \tag{4-46}$$

Expanding into matrix notation gives

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 0 \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} q_4 & -q_3 & q_2 & -q_1 \\ q_3 & q_4 & -q_1 & -q_2 \\ -q_2 & q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{bmatrix}$$

$$= \left[\begin{array}{ccc|c} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2\,(q_1q_2 - q_3q_4) & 2\,(q_1q_3 + q_2q_4) & 0 \\ 2\,(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2\,(-q_1q_4 + q_2q_3) & 0 \\ 2\,(q_1q_3 - q_2q_4) & 2\,(q_1q_4 + q_2q_3) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 & 0 \\ \hline 0 & 0 & 0 & q_1^2 + q_2^2 + q_3^2 + q_4^2 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \end{bmatrix} \tag{4-47}$$

noting that any three-dimensional vector can be rotated using quaternions by adding a zero for the scalar component. The rotation matrix C_x^y equivalent to $\mathbf{q}_x^y$ is the upper left 3×3 block from Equation 4-47.

$$C_x^y = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2\left(q_1 q_2 - q_3 q_4\right) & 2\left(q_1 q_3 + q_2 q_4\right) \\ 2\left(q_1 q_2 + q_3 q_4\right) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2\left(-q_1 q_4 + q_2 q_3\right) \\ 2\left(q_1 q_3 - q_2 q_4\right) & 2\left(q_1 q_4 + q_2 q_3\right) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{4-48}$$

If $\mathbf{q}_x^y$ represents a rotation from x to y, and $\mathbf{q}_y^z$ represents a rotation from y to z, then the rotation from x to z is given by

$$\mathbf{q}_x^z = \mathbf{q}_y^z \cdot \mathbf{q}_x^y. \tag{4-49}$$

4.4.2 Descent Algorithms using Coplanarity Constraint

Overview. Section 6.1.2 of Maybank's text [1] shows a first and second order descent algorithm which could be used to calculate an estimate of the displacement vector in the body coordinate system, $\hat{\mathbf{a}}^{c1}$. GPS differential carrier-phase positioning would then give a corresponding displacement vector in a local level coordinate system, $\mathbf{a}^{LL}$. Another vector expressed in both coordinate systems would then need to be found. Once two vectors expressed in both coordinate systems are found, then the solution for the vehicle attitude can be found via the same manner outlined in the SVD solution.

Calculation of displacement vector in body coordinate system. Both descent algorithms given in Maybank's text [1] use the error function $V(E)$ described by Equation 4-4 with two subtle changes: (1) expanding the essential matrix E to its implicit components of the camera displacement $\{R_{c1}^{c2},\, \hat{\mathbf{a}}^{c1}\}$ and (2) using a unit quaternion $\mathbf{z}_{c1}^{c2}$ to represent the rotation between the camera systems instead of the orthonormal rotation matrix R_{c1}^{c2}. Making the above two changes to Equation 4-4 results in [1]

$$V(E) = V(R_{c1}^{c2}, \hat{\mathbf{a}}^{c1}) = V(\mathbf{z}_{c1}^{c2}, \hat{\mathbf{a}}^{c1}) = \sum_{i=1}^{n} \left[(\mathbf{z}_{c1}^{c2} \cdot \hat{\mathbf{q}}_i^{c2} \cdot (\mathbf{z}_{c1}^{c2})^{-1}) \cdot (\hat{\mathbf{a}}^{c1} \times \ \hat{\mathbf{p}}_i^{c1}) \right], \tag{4-50}$$

where the Equation 4-50 is utilizing the coplanarity constraint. Note that the hat symbol on $\hat{\mathbf{a}}^{c1}, \hat{\mathbf{q}}_i^{c2},$ and $\hat{\mathbf{p}}_i^{c1}$ in Equation 4-50 is used to denote unit vectors.

The descent algorithm searches the neighborhood around $(\mathbf{z}_{c1}^{c2}, \hat{\mathbf{a}}^{c1})$ and determines the small perturbations $\Delta\mathbf{z}_{c1}^{c2}$ and $\Delta\mathbf{a}^{c1}$ which produce the greatest decrease in V for the given constraints

$$\|\mathbf{z}_{c1}^{c2} + \Delta\mathbf{z}_{c1}^{c2}\| = \|\mathbf{z}_{c1}^{c2}\| = 1 \ , \tag{4-51}$$

$$\|\hat{\mathbf{a}}^{c1} + \Delta\mathbf{a}^{c1}\| = \|\hat{\mathbf{a}}^{c1}\| = 1 \ . \tag{4-52}$$

The Taylor series expansion of Equation 4-50 about the point $(\mathbf{z}_{c1}^{c2}, \hat{\mathbf{a}}^{c1})$ is

$$\begin{aligned} V(\mathbf{z}_{c1}^{c2} + \delta\mathbf{z}_{c1}^{c2}, \hat{\mathbf{a}}^{c1} + \delta\mathbf{a}^{c1}) = V(\mathbf{z}_{c1}^{c2}, \hat{\mathbf{a}}^{c1}) + \mathbf{l} \cdot \delta\mathbf{z}_{c1}^{c2} + \mathbf{m} \cdot \delta\mathbf{a}^{c1} + (\delta\mathbf{z}_{c1}^{c2})^T L \delta\mathbf{z}_{c1}^{c2} \\ + 2\delta\mathbf{z}^T M \delta\mathbf{a}^{c1} + (\delta\mathbf{a}^{c1})^T N \delta\mathbf{a}^{c1} + \text{H.O.T.} \end{aligned} \tag{4-53}$$

where $\mathbf{l}$ and $\mathbf{m}$ are vectors and $L, M,$ and N are matrices which are functions of $\mathbf{z}_{c1}^{c2}, \hat{\mathbf{a}}^{c1}, \hat{\mathbf{q}}^{c2},$ and $\hat{\mathbf{p}}^{c1}$. Equation 4-53 is used to calculate V in Maybank's descent algorithms.[3] [1]

First order descent. The first order descent algorithm uses the first three terms of Equation 4-53 to calculate $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$. The values of $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$ are sought which make $\mathbf{l} \cdot \Delta\mathbf{z} + \mathbf{m} \cdot \Delta\mathbf{a}$ the most negative, subject to the constraints given by Equations 4-51 and 4-52. Also, a step size h needs to be defined to constrain $\|\Delta\mathbf{z}\|$ and $\|\Delta\mathbf{a}\|$ to be a small fixed value.

$$\|\Delta\mathbf{z}\| = \|\Delta\mathbf{a}\| = h \tag{4-54}$$

This constraint is imposed to ensure that the first order approximation is valid in the region around $V(\mathbf{z}, \mathbf{a})$ that is searched. [1]

The two delta terms, $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$, are found independently of each other. If the starting point of the descent algorithm is given as $(\mathbf{z}_1, \mathbf{a}_1)$, then the value of $\Delta\mathbf{z}$ is found via the error function

$$W(\lambda_1, \lambda_2, \Delta\mathbf{z}) = \mathbf{l} \cdot \Delta\mathbf{z} + \lambda_1(\|\Delta\mathbf{z}\|^2 - h^2) + \lambda_2(\|\mathbf{z}_1 + \Delta\mathbf{z}\|^2 - 1) \ , \tag{4-55}$$

where λ_1 and λ_2 are LaGrange multipliers used to enforce the constraints listed in Equations 4-51 and 4-54. Differentiating Equation 4-55 with respect to λ_1, λ_2, and $\Delta\mathbf{z}$, and setting the respective results to zero gives three equations and three unknowns. These three equations can be used to solve for the value of $\Delta\mathbf{z}$ which results in the most decrease in V. A similar method can be used to find the value of $\Delta\mathbf{a}$ which produces the most decrease in the value of V. [1]

[3] For ease of readability the unit vector hat symbol, superscripts and subscript will be dropped on the references to $\mathbf{z}_{c1}^{c2}$, $\Delta\mathbf{z}_{c1}^{c2}$, $\hat{\mathbf{a}}^{c1}$, and $\Delta\mathbf{a}^{c1}$ for the remainder of the discussion of Maybank's descent algorithms.

The updated values $(\mathbf{z}_2, \mathbf{a}_2)$ are defined as

$$(\mathbf{z}_2, \mathbf{a}_2) = (\mathbf{z}_1 + \Delta\mathbf{z}, \mathbf{a}_1 + \Delta\mathbf{a}) \ . \tag{4-56}$$

If the value of $V(\mathbf{z}_2, \mathbf{a}_2)$ is significantly smaller than the value of $V(\mathbf{z}_1, \mathbf{a}_1)$, then the descent algorithm is applied to $(\mathbf{z}_2, \mathbf{a}_2)$ in place of $(\mathbf{z}_1, \mathbf{a}_1)$. If $V(\mathbf{z}_2, \mathbf{a}_2)$ is not less than a defined delta from $V(\mathbf{z}_2, \mathbf{a}_2)$, then the algorithm is repeated on $(\mathbf{z}_1, \mathbf{a}_1)$ with a smaller step size, h, to determine $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$. If the step size reaches its smallest allowable value, then the algorithm terminates and returns the current value of $(\mathbf{z}, \mathbf{a})$ as an estimate of the global minimum of the error function V.

Second order descent. The second order descent algorithm can be applied if the first order descent algorithm does not produce a reasonable value for the global minimum of V. The second order algorithm uses all terms of Equation 4-53 to calculate $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$, with the additional constraints added,

$$\mathbf{z}_1 \cdot \Delta\mathbf{z} = 0 \ , \tag{4-57}$$

$$\mathbf{a}_1 \cdot \Delta\mathbf{a} = 0 \ . \tag{4-58}$$

The error function U is defined by

$$\begin{aligned} U(\lambda_1, \lambda_2, \Delta\mathbf{z}, \Delta\mathbf{a}) = \mathbf{l} \cdot \Delta\mathbf{z} + \mathbf{m} \cdot \Delta\mathbf{a} + (\Delta\mathbf{z})^T L \Delta\mathbf{z} \\ + 2(\delta\mathbf{z})^T M \delta\mathbf{a} + (\delta\mathbf{a})^T N \delta\mathbf{a} + \lambda_1 \mathbf{z}_1 \cdot \Delta\mathbf{z} + \lambda_2 \mathbf{a}_1 \cdot \Delta\mathbf{a} \end{aligned} \tag{4-59}$$

where the LaGrange multipliers λ_1 and λ_2 are used to enforce the constraints given in Equations 4-57 and 4-58. Differentiating Equation 4-59 with respect to $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$ and setting the results to zero gives two matrix equations with four unknowns. The two constraint equations, $\mathbf{z}_1 \cdot \Delta\mathbf{z} = 0$ and $\mathbf{a}_1 \cdot \Delta\mathbf{a} = 0$, can then be applied to solve for λ_1 and λ_2. Then, substituting the values of λ_1 and λ_2 into the two matrix equations found via setting the partial derivatives of Equation 4-59 to zero, allows the two equations to be solved for $\Delta\mathbf{z}$ and $\Delta\mathbf{a}$. [1]

The updated values $(\mathbf{z}_2, \mathbf{a}_2)$ are defined as

$$(\mathbf{z}_2, \mathbf{a}_2) = (\mathbf{z}_1 + \Delta\mathbf{z}, \mathbf{a}_1 + \Delta\mathbf{a}) \ . \tag{4-60}$$

If the value of $V(\mathbf{z}_2, \mathbf{a}_2)$ is significantly smaller than the value of $V(\mathbf{z}_1, \mathbf{a}_1)$, then the descent algorithm is applied to $(\mathbf{z}_2, \mathbf{a}_2)$ in place of $(\mathbf{z}_1, \mathbf{a}_1)$. If $V(\mathbf{z}_2, \mathbf{a}_2)$ is not less than a defined delta

from $V(\mathbf{z}_2, \mathbf{a}_2)$, then the algorithm terminates and returns the current value of $(\mathbf{z}, \mathbf{a})$ as an estimate of the global minimum of the error function V.[4]

4.4.3 Descent Algorithm Using Linearized Measurement Model

The descent or error minimization approach is less efficient than the SVD solution. But the efficiency in the SVD approach can be lost if the incorporation of statistics into the algorithm is desired. The descent algorithm allows the incorporation of statistics with the cost being a more complicated error equation. A linearized model of the system which incorporates focal plane measurement error, Equation 5-33, is derived in the next chapter. The state error vector, $\mathbf{v}_i$, of independent variables is a function of the point correspondence measurements $(\hat{\mathbf{q}}_i^{c2}, \hat{\mathbf{p}}_i^{c1})$ and the measurement errors $(\tilde{c}_i, \tilde{r}_i)$. An error equation based on minimizing the state error vector can be written in the form

$$J(\mathbf{v}_i) = \sum_{i=1}^{n} \mathbf{v}_i^T \, W \, \mathbf{v}_i \ , \tag{4-61}$$

where W is added as a weight matrix because the state error vector $\mathbf{v}$ is a combination of angular and range errors. Equation 4-61 can be used to form a descent solution algorithm similar to the ones described by Maybank's text in section 6.1.2 [1]. Development and implementation of this solution technique is left for a future area of study.

[4] Further details of the first and second order descent solution algorithms are given in section 6.1.2 of Maybank's text [1].

CHAPTER 5
ERROR ANALYSIS

The accuracy of the inertial attitude estimates is the litmus test for the viability of this application. This chapter starts with a presentation of the error model scenario. All assumptions and information considered as constants are outlined. Next, the derivation of a generalized system model is presented. Then, an algorithm simulating the generalized model is described and verification results are presented. Finally, the results of parameter studies utilizing this algorithm are presented for (1) camera focal length/field of view (FOV), (2) camera depression angle, (3) displacement vector magnitude between imaging systems, (4) number of control points, (5) focal plane location of control points, and (6) control point measurement error.

5.1 Generalized Linearized Error Model

To quantify the accuracy of this model, consider an aerial vehicle, or agent that contains an imaging system, GPS sensor and INS system. The imaging system consists of a camera with a given focal length, physical size, pixel count, and depression angle. Suppose the agent is flying horizontal so that the translation vector, $\mathbf{a}^{LL}$ does not have any component in the up or $\hat{\mathbf{z}}^{LL}$ direction[1] . While moving horizontally, the agent takes two images such that they have an overlapping FOV. A minimum of five image correspondences must be present in the two images. An example scenario is shown in Figure 5–1.

The measurement model equation is given by

$$\mathbf{q}^{c2} = R_{c1}^{c2}\mathbf{p}^{c1} - R_{LL}^{c2}\mathbf{a}^{LL}, \tag{5-1}$$

where $\mathbf{p}^{c1}$ is the vector from the first camera position to the control point, $\mathbf{q}^{c2}$ is the vector from the second camera position to the control point, and $\mathbf{a}^{LL}$ is the vector from the first camera position to the second camera position in the LL coordinate system. As a reminder, the notation $\mathbf{q}^x$ denotes a vector $\mathbf{q}$ expressed in coordinate system x, and R_x^y is the rotation matrix from

[1] The horizontal constraint is done to cancel out the translational axis ambiguity. The c, or center, coordinate system will be introduced later to explain the reasoning for this seemingly arbitrary constraint.

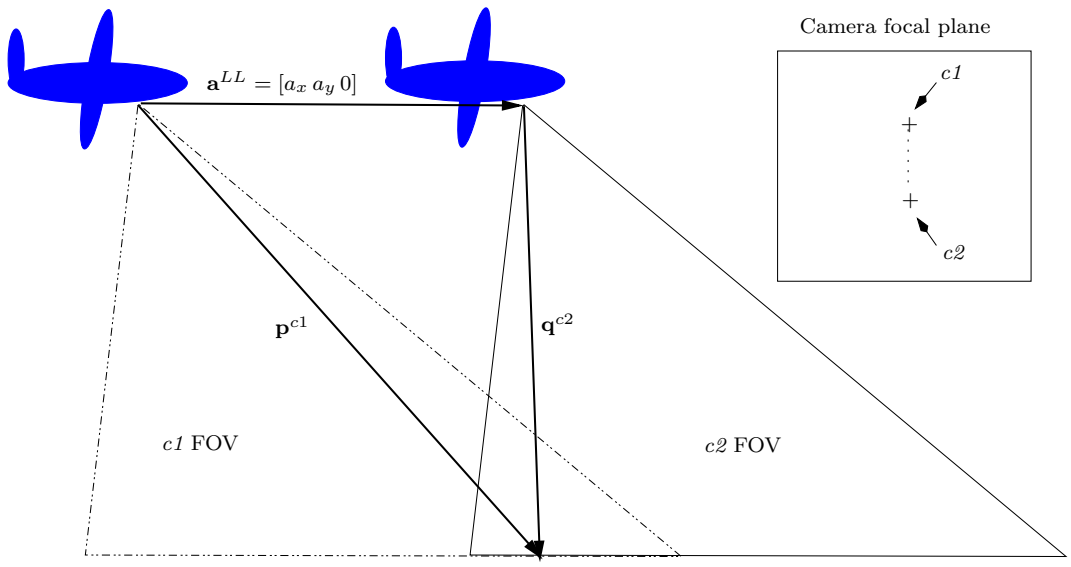

Figure 5–1: Error analysis scenario.

coordinate system x to coordinate system y. The coordinate systems used in this model are described in Table 5–1.

Table 5–1: Coordinate System Definitions

System	Definition
$c1$	Camera coordinates for camera in the first position; the x-y plane is the focal plane of the camera, and the z axis is along the camera boresite.
$c2$	Camera coordinates for camera in the second position; same convention as above.
c	Center coordinate system where $\hat{x}^c$ axis points along vector $\mathbf{a}^{LL}$ and the $\hat{z}^c$ axis is as close to vertical as possible (see appendix for discussion of the c coordinate system).
LL	Local level coordinates; north-east-down or east-north-up.

We will assume that the direction of $\mathbf{p}^{c1}$ is a perfect measurement and all errors in direction are associated with the second measurement $\mathbf{q}^{c2}$. However, the range to the control point is unknown and must be calculated from the measurements, so let

$$\mathbf{p}^{c1} = \widehat{\mathbf{p}}^{c1}(1 + \delta p), \tag{5-2}$$

where $\widehat{\mathbf{p}}^{c1}$ is an estimate of $\mathbf{p}^{c1}$, and δp is the proportion of error in the length of $\widehat{\mathbf{p}}^{c1}$.

In order to overcome the translation axis ambiguity, the horizontal constraint on the displacement vector's direction is used to define the coordinate system c such that the $\hat{\mathbf{x}}^c$ axis points along vector $\mathbf{a}^{LL}$, and the $\hat{\mathbf{z}}^c$ axis is as close to vertical as possible. As discussed in the appendix, the matrix R^c_{LL} is given by

$$R^c_{LL} = \frac{1}{\|\mathbf{a}^{LL}\|}\begin{bmatrix} a_x^{LL} & a_y^{LL} & a_z^{LL} \\ \frac{-a_y^{LL}\|a\|}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & \frac{a_x^{LL}\|a\|}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & 0 \\ \frac{-a_x^{LL}a_z^{LL}}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & \frac{-a_y^{LL}a_z^{LL}}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & \sqrt{(a_x^{LL})^2+(a_y^{LL})^2} \end{bmatrix}. \tag{5-3}$$

Note that when the rotation R^c_{LL} is applied to $\mathbf{a}^{LL}$, the following result is obtained:

$$\begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\|\mathbf{a}^{LL}\|}\begin{bmatrix} a_x^{LL} & a_y^{LL} & a_z^{LL} \\ \frac{-a_y^{LL}\|a\|}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & \frac{a_x^{LL}\|a\|}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & 0 \\ \frac{-a_x^{LL}a_z^{LL}}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & \frac{-a_y^{LL}a_z^{LL}}{\sqrt{(a_x^{LL})^2+(a_y^{LL})^2}} & \sqrt{(a_x^{LL})^2+(a_y^{LL})^2} \end{bmatrix}\begin{bmatrix} a_x^{LL} \\ a_y^{LL} \\ a_z^{LL} \end{bmatrix}, \tag{5-4}$$

where $a = \|\mathbf{a}^{LL}\|$.

Now, let

$$R^{c2}_{c1} = \delta R_1 \widehat{R}^{c2}_{c1}, \tag{5-5}$$

where $\widehat{R}^{c2}_{c1}$ is an estimate of R^{c2}_{c1}, and δR_1 is the misalignment in the estimate. Also, let

$$R^{c2}_{LL} = \widehat{R}^{c2}_{LL} R^{LL}_c \delta R_2 R^c_{LL}, \tag{5-6}$$

where $\widehat{R}^{c2}_{LL}$ is an estimate of R^{c2}_{LL}, and δR_2 is the misalignment in the estimate expressed in the c coordinate system. When δR_2 is expressed in the c coordinate system, one of the three misalignment angles is unobservable and will drop out of the calculations. This removes the translational axis ambiguity from this problem.

The measurement equation for the generalized error model, Equation 5-1, now becomes

$$\mathbf{q}^{c2} = \delta R_1 \widehat{R}^{c2}_{c1}\widehat{\mathbf{p}}^{c1}(1+\delta p) - \widehat{R}^{c2}_{LL} R^{LL}_c \delta R_2 R^c_{LL}\mathbf{a}^{LL}. \tag{5-7}$$

To simplify Equation 5-7, combine some constant factors:

$$\widehat{\mathbf{p}}^{c2} = \widehat{R}^{c2}_{c1}\widehat{\mathbf{p}}^{c1} \tag{5-8}$$

$$\widehat{R}^{c2}_{c} = \widehat{R}^{c2}_{LL} R^{LL}_c \tag{5-9}$$

$$\mathbf{a}^c = R^c_{LL}\mathbf{a}^{LL} \tag{5-10}$$

Then the measurement model equation becomes

$$\mathbf{q}^{c2} = \delta R_1 \widehat{\mathbf{p}}^{c2}(1+\delta p) - \widehat{R}^{c2}_c \delta R_2 \mathbf{a}^c. \tag{5-11}$$

Equation 5-11 written out with all vectors and matrices expanded is

$$\begin{bmatrix} q_x^{c2} \\ q_y^{c2} \\ q_z^{c2} \end{bmatrix} = \begin{bmatrix} 1 & \tilde{\psi}_1 & -\tilde{\theta}_1 \\ -\tilde{\psi}_1 & 1 & \tilde{\phi}_1 \\ \tilde{\theta}_1 & -\tilde{\phi}_1 & 1 \end{bmatrix} \begin{bmatrix} \widehat{p}_x^{c2} \\ \widehat{p}_y^{c2} \\ \widehat{p}_z^{c2} \end{bmatrix} (1+\delta p) - \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} a \\ -\tilde{\psi}_2 a \\ \tilde{\theta}_2 a \end{bmatrix} . \tag{5-12}$$

The solution for the cr-coordinates on the focal plane of the second imaging system $c2$ given by Equations 4-1 and 4-2 are

$$c = f\frac{q_x^{c2}}{q_z^{c2}} = f\frac{(\widehat{p}_x^{c2} + \tilde{\psi}_1\widehat{p}_y^{c2} - \tilde{\theta}_1\widehat{p}_z^{c2})(1+\delta p) - (r_{11}a - r_{12}\tilde{\psi}_2 a + r_{13}\tilde{\theta}_2 a)}{(\tilde{\theta}_1\widehat{p}_x^{c2} - \tilde{\phi}_1\widehat{p}_y^{c2} + \widehat{p}_z^{c2})(1+\delta p) - (r_{31}a - r_{32}\tilde{\psi}_2 a + r_{33}\tilde{\theta}_2 a)} \tag{5-13}$$

$$r = f\frac{q_y^{c2}}{q_z^{c2}} = f\frac{(-\tilde{\psi}_1\widehat{p}_x^{c2} + \widehat{p}_y^{c2} + \tilde{\phi}_1\widehat{p}_z^{c2})(1+\delta p) - (r_{21}a - r_{22}\tilde{\psi}_2 a + r_{23}\tilde{\theta}_2 a)}{(\tilde{\theta}_1\widehat{p}_x^{c2} - \tilde{\phi}_1\widehat{p}_y^{c2} + \widehat{p}_z^{c2})(1+\delta p) - (r_{31}a - r_{32}\tilde{\psi}_2 a + r_{33}\tilde{\theta}_2 a)} . \tag{5-14}$$

A summary of the above variables is given in Table 5–2 where $(\widehat{p}_x^{c2}, \widehat{p}_y^{c2}, \widehat{p}_z^{c2})$, a, and $\{r_{ij}\}$ are treated as constants; (c, r) are dependent variables or measurements subject to measurement errors; and the state vector of independent variables is $(\tilde{\phi}_1, \tilde{\theta}_1, \tilde{\psi}_1, \tilde{\theta}_2, \tilde{\psi}_2, \delta p)$. Note that $\tilde{\phi}_2$, which represents the roll about the generalized translational axis, $\hat{\mathbf{x}}^c$, does not appear in Equations 5-13 and 5-14. This resolves the translational axis ambiguity that results from only using two images to obtain the attitude estimate.

To linearize the measurement model, we calculate the partial derivatives of each of the measurements with respect to the independent variables about a nominal point. Let the vector notation

$$f(\nu) = f(\nu_{\mathbf{0}}) + \left.\frac{\partial f}{\partial b}\right|_{\nu_0} \triangle b + \left.\frac{\partial f}{\partial c}\right|_{\nu_0} \triangle c + \text{ higher order terms} \tag{5-15}$$

indicate a first order Taylor series expansion on the independent variables a, b, and c about a nominal point $\nu_{\mathbf{0}}$ where $\nu = [a\ b\ c]^T$ and $\nu_{\mathbf{0}} = [a\ b_0\, c_0]^T$. The nominal point $\nu_{\mathbf{0}}$ consists of a constant value a and nominal values for b and c. Extending this notation to the measurement equation gives the generalized linearized measurement model as

$$\begin{aligned} c(\nu) = c(\nu_0) + \left.\frac{\partial c}{\partial \tilde{\phi}_1}\right|_{\nu_0} \triangle\tilde{\phi}_1 + \left.\frac{\partial c}{\partial \tilde{\theta}_1}\right|_{\nu_0} \triangle\tilde{\theta}_1 + \left.\frac{\partial c}{\partial \tilde{\psi}_1}\right|_{\nu_0} \triangle\tilde{\psi}_1 + \left.\frac{\partial c}{\partial \tilde{\theta}_2}\right|_{\nu_0} \triangle\tilde{\theta}_2 \\ + \left.\frac{\partial c}{\partial \tilde{\psi}_2}\right|_{\nu_0} \triangle\tilde{\psi}_2 + \left.\frac{\partial c}{\partial \delta p}\right|_{\nu_0} \triangle\delta p \end{aligned} \tag{5-16}$$

$$\begin{aligned} r(\nu) = r(\nu_0) + \left.\frac{\partial r}{\partial \tilde{\phi}_1}\right|_{\nu_0} \triangle\tilde{\phi}_1 + \left.\frac{\partial r}{\partial \tilde{\theta}_1}\right|_{\nu_0} \triangle\tilde{\theta}_1 + \left.\frac{\partial r}{\partial \tilde{\psi}_1}\right|_{\nu_0} \triangle\tilde{\psi}_1 + \left.\frac{\partial r}{\partial \tilde{\theta}_2}\right|_{\nu_0} \triangle\tilde{\theta}_2 \\ + \left.\frac{\partial r}{\partial \tilde{\psi}_2}\right|_{\nu_0} \triangle\tilde{\psi}_2 + \left.\frac{\partial r}{\partial \delta p}\right|_{\nu_0} \triangle\delta p \end{aligned} \tag{5-17}$$

Table 5–2: Description of variables in the generalized error model

Variable(s)	Description
$(\widehat{p}_x^{c2}, \widehat{p}_y^{c2}, \widehat{p}_z^{c2})$	The position of the control point designated from the first camera position with the range estimated, transformed to the second camera system, *c2* via $\widehat{R}_{c1}^{c2}$.
a	Distance between the first camera position and the second camera position; the direction of displacement between camera positions appears in the measurement equations implicitly.
$\{r_{ij}\}$	Elements of matrix $\widehat{R}_c^{c2}$; a function of estimated or known rotations.
(r,c)	Measurement of the control point position from the second camera position.
$(\tilde{\phi}_1, \tilde{\theta}_1, \tilde{\psi}_1)$	Misalignment angles in the estimate of the rotation of the camera from first position to the second position.
$(\tilde{\theta}_2, \tilde{\psi}_2)$	Misalignment angles in the estimate of the roll and yaw 3-2-1 Euler angles from the *c* coordinate system to the *c2* system. Note that if the vehicle is flying horizontally (i.e., no $\hat{\mathbf{z}}^{LL}$ component to $\mathbf{a}^{LL}$) that these angles correspond directly to the inertial pitch and heading attitude errors of the vehicle in the second position.
δ p	Proportion of error in the estimate of range to the control point.

where $\nu = [\widehat{\mathbf{q}}^{c2}\ \widehat{\mathbf{p}}^{c1}\ r_{ij}\ a\ \tilde{\phi}_1\ \tilde{\theta}_1\ \tilde{\psi}_1\ \tilde{\theta}_2\ \tilde{\psi}_2\ \delta p]^T$ and $\nu_0 = [\widehat{\mathbf{q}}^{c2}\ \widehat{\mathbf{p}}^{c1}\ r_{ij}\ a\ \tilde{\phi}_{1,0}\ \tilde{\theta}_{1,0}\ \tilde{\psi}_{1,0}\ \tilde{\theta}_{2,0}\ \tilde{\psi}_{2,0}\ \delta p_0]^T$. Note that $\widehat{\mathbf{q}}^{c2}$, $\widehat{\mathbf{p}}^{c1}$, r_{ij}, and a are treated as constants in the Taylor series expansion of the measurement equations for $c(\nu)$ and $r(\nu)$.

Calculating the partial derivatives for each of the terms in Equations 5-16 and 5-17, and letting the all the nominal values be zero gives

$$\frac{\partial c}{\partial \tilde{\phi}_1} = f\frac{(\widehat{p}_x^{c2} - r_{11}a)\widehat{p}_y^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-18}$$

$$\frac{\partial c}{\partial \tilde{\theta}_1} = f\frac{-(\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_z^{c2} - (\widehat{p}_x^{c2} - r_{11}a)\widehat{p}_x^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-19}$$

$$\frac{\partial c}{\partial \tilde{\psi}_1} = f\frac{(\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_y^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-20}$$

$$\frac{\partial c}{\partial \tilde{\theta}_2} = f\frac{-(\widehat{p}_z^{c2} - r_{31}a)r_{13}a + (\widehat{p}_x^{c2} - r_{11}a)r_{33}a}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-21}$$

$$\frac{\partial c}{\partial \tilde{\psi}_2} = f\frac{(\widehat{p}_z^{c2} - r_{31}a)r_{12}a - (\widehat{p}_x^{c2} - r_{11}a)r_{32}a}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-22}$$

$$\frac{\partial c}{\partial \delta p} = f\frac{-r_{31}a\widehat{p}_x^{c2} + r_{11}a\widehat{p}_z^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-23}$$

$$\frac{\partial r}{\partial \tilde{\phi}_1} = f\frac{(\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_z^{c2} + (\widehat{p}_y^{c2} - r_{21}a)\widehat{p}_y^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-24}$$

$$\frac{\partial r}{\partial \tilde{\theta}_1} = f\frac{-(\widehat{p}_y^{c2} - r_{21}a)\widehat{p}_x^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-25}$$

$$\frac{\partial r}{\partial \tilde{\psi}_1} = f\frac{-(\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_x^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-26}$$

$$\frac{\partial r}{\partial \tilde{\theta}_2} = f\frac{-(\widehat{p}_z^{c2} - r_{31}a)r_{23}a + (\widehat{p}_y^{c2} - r_{21}a)r_{33}a}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-27}$$

$$\frac{\partial r}{\partial \tilde{\psi}_2} = f\frac{(\widehat{p}_z^{c2} - r_{31}a)r_{22}a - (\widehat{p}_y^{c2} - r_{21}a)r_{32}a}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-28}$$

$$\frac{\partial r}{\partial \delta p} = f\frac{-r_{31}a\widehat{p}_y^{c2} + r_{21}a\widehat{p}_z^{c2}}{(\widehat{p}_z^{c2} - r_{31}a)^2} \tag{5-29}$$

Define the error in the c and r measurements as

$$\begin{bmatrix} \tilde{c} \\ \tilde{r} \end{bmatrix} \triangleq \begin{bmatrix} c(\nu) \\ r(\nu) \end{bmatrix} - \begin{bmatrix} c(\nu_0) \\ r(\nu_0) \end{bmatrix} . \tag{5-30}$$

The complete linearized measurement equation is then given by

$$\begin{bmatrix} \tilde{c} \\ \tilde{r} \end{bmatrix} = \frac{f}{(\widehat{p}_z^{c2} - r_{31}a)^2} \left[\begin{array}{c|c|c|c|c|c} (\widehat{p}_x^{c2} - r_{11}a)\widehat{p}_y^{c2} & -(\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_z^{c2} - (\widehat{p}_x^{c2} - r_{11}a)\widehat{p}_x^{c2} & (\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_y^{c2} & -(\widehat{p}_z^{c2} - r_{31}a)r_{13}a + (\widehat{p}_x^{c2} - r_{11}a)r_{33}a & (\widehat{p}_z^{c2} - r_{31}a)r_{12}a - (\widehat{p}_x^{c2} - r_{11}a)r_{32}a & -r_{31}a\widehat{p}_x^{c2} + r_{11}a\widehat{p}_z^{c2} \\ (\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_z^{c2} + (\widehat{p}_y^{c2} - r_{21}a)\widehat{p}_y^{c2} & -(\widehat{p}_y^{c2} - r_{21}a)\widehat{p}_x^{c2} & -(\widehat{p}_z^{c2} - r_{31}a)\widehat{p}_x^{c2} & -(\widehat{p}_z^{c2} - r_{31}a)r_{23}a + (\widehat{p}_y^{c2} - r_{21}a)r_{33}a & (\widehat{p}_z^{c2} - r_{31}a)r_{22}a - (\widehat{p}_y^{c2} - r_{21}a)r_{32}a & -r_{31}a\widehat{p}_y^{c2} + r_{21}a\widehat{p}_z^{c2} \end{array} \right] \begin{bmatrix} \triangle\tilde{\phi}_1 \\ \triangle\tilde{\theta}_1 \\ \triangle\tilde{\psi}_1 \\ \triangle\tilde{\theta}_2 \\ \triangle\tilde{\psi}_2 \\ \triangle\delta p \end{bmatrix}. \quad (5\text{-}31)$$

Let C represent the submatrix consisting of the first five columns of the matrix in Equation 5-31, and let c represent the last column of the matrix in Equation 5-31. So, C is a 2×5 submatrix and c is a 2×1 submatrix or column vector. Note that there are six unknowns and two measurements. So, more measurements are needed to uniquely determine the set of independent variables. But, each measurement will introduce a different range error, δp_i. So, the minimum number of control points needed to uniquely determine the independent variables is five. For five control points, the system is represented by:

$$\begin{bmatrix} \tilde{c}_1 \\ \tilde{r}_1 \\ \tilde{c}_2 \\ \tilde{r}_2 \\ \tilde{c}_3 \\ \tilde{r}_3 \\ \tilde{c}_4 \\ \tilde{r}_4 \\ \tilde{c}_5 \\ \tilde{r}_5 \end{bmatrix}_{10x1} = \begin{bmatrix} C_1 & c_1 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ C_2 & 0 & c_2 & 0 & 0 & 0 \\ & 0 & & 0 & 0 & 0 \\ C_3 & 0 & 0 & c_3 & 0 & 0 \\ & 0 & 0 & & 0 & 0 \\ C_4 & 0 & 0 & 0 & c_4 & 0 \\ & 0 & 0 & 0 & & 0 \\ C_5 & 0 & 0 & 0 & 0 & c_5 \\ & 0 & 0 & 0 & 0 & \end{bmatrix}_{10x10} \begin{bmatrix} \triangle\tilde{\phi}_1 \\ \triangle\tilde{\theta}_1 \\ \triangle\tilde{\psi}_1 \\ \triangle\tilde{\theta}_2 \\ \triangle\tilde{\psi}_2 \\ \triangle\delta p_1 \\ \triangle\delta p_2 \\ \triangle\delta p_3 \\ \triangle\delta p_4 \\ \triangle\delta p_5 \end{bmatrix}_{10x1} \quad (5\text{-}32)$$

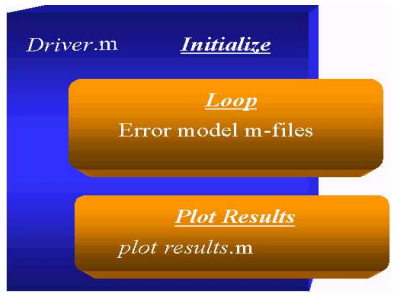

Figure 5–2: Overview of Matlab© implementation of the generalized linearized error model analysis (GLEMA) program.

where the subscripts denote the matrix sizes in Equation 5-32. Let the n control point measurements be denoted as the $2n \times 1$ column vector $\mathbf{m}$, the independent variables, or state error vector, be denoted as the $(5+n) \times 1$ column vector $\mathbf{v}$, and the $2n \times (5+n)$ square matrix of constant terms be G. Then, Equation 5-32 can be concisely written as

$$\mathbf{m}_{2n\times 1} = G_{2n\times(n+5)}\mathbf{v}_{(n+5)\times 1} \tag{5-33}$$

where $\mathbf{v}$ will be referred to as the state error vector hereafter. Adding more than five measurements allows Equation 5-33 to be solved by a least squares type approach.

5.1.1 Implementation of the Generalized Linearized Error Model Analysis (GLEMA) Matlab© Program

An analysis program for the generalized linearized error model, GLEMA, was created in the Matlab© programming language. The implementation contains three inter-dependent components: (1) initialization, (2) loop structure, and (3) display of results. The initialization component defines the fixed parameters, sets up the parameter that will be looped over (i.e., focal length, camera depression angle, displacement vector, etc.), and sets up and maintains the output containers that hold the analysis results. The looping component computes the error variances for the independent variables (given in the state error vector $\mathbf{v}$) for all the geometries of interest. The variances are then plotted versus the parameter of interest in the plot results component.

GLEMA consists of a combination of Matlab© functions and scripts.[2] The labels `driver.m` and `plot_results.m` in Figure 5–2 are used to illustrate the one-to-one correspondence between a driver file and a plotting file. This correspondence is shown in Table 5–3. The list below describes

[2] The difference between a function and a script is in the scope of the variables. A script sends the variables to the workspace and a function does not. Both files end in ".m" and are commonly called m-files.

each step of the implementation as it corresponds to the three components shown in Figure 5–2 and lists the m-files that correspond to each step.

initialize 1 `driver.m`

Set up fixed camera parameters, control points (cr–coordinates with LL heights), true attitude, range and attitude errors, and displacement vector between the cameras.

initialize 2 `driver.m`

Set up the parameter of interest and output container(s) to capture the data from the upcoming loop.

initialize 3 `driver.m, errorAnalysis.m`

Loop over parameter of interest.

loop 1 `errorAnalysis.m, inputGeometry_truth.m`

Get the geometry of for the truth system in LL.

loop 2 `errorAnalysis.m, checkIfControlPointInFOV.m`

Ensure that there are a minimum of five points that are in the FOV of both cameras.

loop 3 `errorAnalysis.m, inputGeometry_truth.m, crh2enu.m, computeControlPoint.m`

Calculate the true control point locations in the $c1$ and $c2$ coordinate systems.

loop 4 `errorAnalysis.m, inputGeometry_est.m`

Add in attitude and range errors to get the geometry in the estimated system in LL

loop 5 `errorAnalysis.m, calc_cr_truth.m`

Calculate the true control point cr–coordinates on the second camera's focal plane.

loop 6 `errorAnalysis.m, calc_cr_est.m`

Calculate the estimated control point cr–coordinates on the second camera's focal plane.

loop 7 `errorAnalysis.m`

Calculate the cr–coordinate error values on the second camera's focal plane.

loop 8 `errorAnalysis.m, reshape_c_and_r_err_into_M.m`

Form the $\mathbf{m}_{2n\times 1}$ vector.

loop 9 `errorAnalysis.m, formG.m, form_c_row_for_G.m, form_r_row_for_G.m`

Form the $G_{2n\times(n+5)}$ matrix.

loop 10 `errorAnalysis.m`

Compute the system error vector, $\mathbf{v}_{(n+5)\times 1}$, of independent state variables from $\mathbf{m}$ and G.

loop 11 `errorAnalysis.m`

Compute the $(n+5)\times(n+5)$ matrix via $(G^TG)^{-1}$.

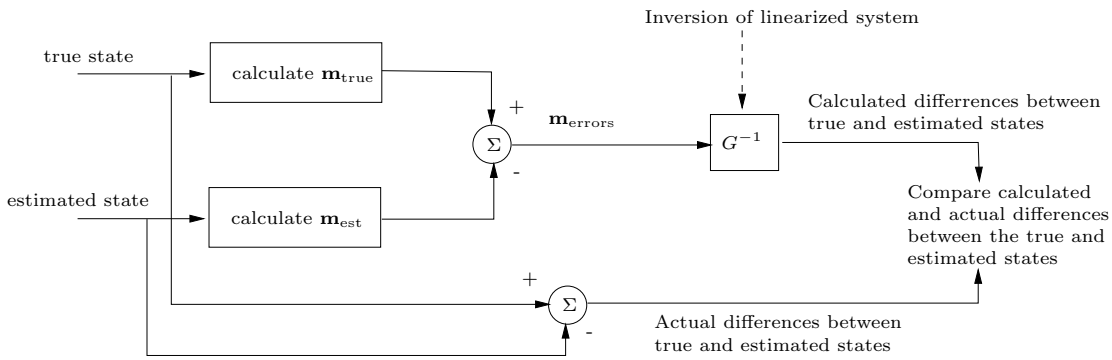

Figure 5–3: Graphical overview of verification of the GLEMA program

loop 12 `driver.m, errorAnalysis.m`

Read off the geometry terms for each independent state variable given by the diagonal terms of $(G^T G)^{-1}$.

loop 13 `driver.m`

Determine a pixel based error corresponding to the *cr*–coordinate measurements made on the second camera systems focal plane. Convert the pixel-based error to an area on the focal plane.

loop 14 `driver.m`

Calculate the error variances for each desired independent variable based on its geometry term and given focal plane error.

plot results 1 `driver.m, plot_results.m`

Plot the error variances versus the parameter of interest.

Table 5–3: Correspondence between driver.m and plot_results.m files for the GLEMA program.

driver.m	plot_results.m
varying_beta.m	plot_results_b.m
varying_displacement_vector.m	plot_results_a.m
varying_focal_length.m	plot_results_f.m
varying_last_CP_location.m	plot_results_cp.m
varying_num_cp_cols.m	plot_results_num_cp_cols.m
varying_num_cp_rows.m	plot_results_num_cp_rows.m
varying_pixel_error.m	plot_results_pixel_error.m
verification.m	Matlab© command window

Verification results. The goal of verifying the error model is to show that the known errors that are input to the system can be successfully calculated using the GLEMA program. To accomplish this goal, the driver file `verification.m` was created. A graphical overview of the verification process is shown in Figure 5–3.

Verifying the accuracy of the GLEMA program was accomplished in two steps. The first step was to verify that linearization of each independent variable in the system error vector, $\mathbf{v}$, with all coordinate systems aligned. A summary of the variables contained in the system error vector with their corresponding inputs is given in Table 5–4. This was verified by setting all input errors to zero except the error of interest, and then calculating the system error vector, $\mathbf{v}$, and visually verifying the output to the Matlab© command window corresponds to the input error. For example, to verify the successful linearization of the inertial heading estimate $\tilde{\psi}_2$, all input error terms were zero except ψ^2_{LL}[3] and the displacement vector was aligned such that no side-slip or angle-of-attack was present. A value of 0.01 radians for ψ^2_{LL} produced a value of 0.0099 rad for $\tilde{\psi}_2$ and values less than 10^{-14} for the rest of the values $\mathbf{v}$. A complete list of results comparing the input errors to the resulting terms of system error vector is given in Table 5–5.

Table 5–4: Summary of linearized independent variables contained in the system error vector $\mathbf{v}$ with their corresponding inputs in the GLEMA program.

Variable	GLEMA Input
$\tilde{\phi}_1$	`errors.phi_c1toc2`
$\tilde{\theta}_1$	`errors.theta_c1toc2`
$\tilde{\psi}_1$	`errors.psi_c1toc2`
$\tilde{\theta}_2$	`errors.theta_ned2c2`
$\tilde{\psi}_2$	`errors.psi_ned2c2`
δp_i	`errors.r[i]`

Table 5–5: Step one verification results for the GLEMA program

	Independent Variables										
	$\tilde{\phi}_1$	$\tilde{\theta}_1$	$\tilde{\psi}_1$	$\tilde{\theta}_2$	$\tilde{\psi}_2$	δp_1	δp_2	δp_3	δp_4	δp_5	δp_6
input	1e-2	0	0	0	0	0	0	0	0	0	0
output	1e-2	0	0	-1.1e-3	0	7e-4	9e-4	9e-4	4e-4	4e-4	4e-4
input	0	1e-2	0	0	0	0	0	0	0	0	0
output	0	1.02e-2	0	7e-4	-1e-4	1.6e-3	2.5e-3	1.4e-3	2e-3	2.4e-3	2e-3
input	0	0	1e-2	0	0	0	0	0	0	0	0
output	-1e-4	0	1e-2	6e-4	7e-4	-3e-4	3e-4	5e-4	0	2e-4	3e-4
input	0	0	0	1e-2	0	0	0	0	0	0	0
output	0	0	0	1e-2	0	0	0	-1e-4	0	0	-1e-4
input	0	0	0	0	1e-2	0	0	0	0	0	0
output	0	0	0	0	1e-2	0	0	0	0	0	0
input	0	0	0	0	0	0	0	1e-2	0	0	0
output	0	0	0	0	0	0	0	1e-2	0	0	0

[3] For convenience sake, the rotation matrix notation is used to denote the coordinate system that the input error term acts upon.

The second step of the verification process involves slewing the heading and velocity/displacement vector around 360°. If the coordinate system rotations and translations are correct, then there should be no change in the system error vector. The inertial heading is given a 0.01 rad error and the velocity and inertial heading are slewed around in 45° increments. The comparison results are shown in Table 5–6. Corresponding comparisons for the remaining independent variables are not shown, since step one produced a valid linearization for all independent variables.

Table 5–6: Step two verification results for the GLEMA program

		Independent Variables										
		$\tilde{\phi}_1$	$\tilde{\theta}_1$	$\tilde{\psi}_1$	$\tilde{\theta}_2$	$\tilde{\psi}_2$	δp_1	δp_2	δp_3	δp_4	δp_5	δp_6
$\psi = 45^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 90^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 135^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 180^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 225^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 270^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 315^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0
$\psi = 360^o$	input	0	0	0	0	1e-2	0	0	0	0	0	0
	output	0	0	0	0	1e-2	0	0	0	0	0	0

The results presented in Tables 5–5 and 5–6 show that the GLEMA program is a reasonable representation of the error model scenario given in Figure 5–1. Specifically, the G matrix is shown to be a valid model of the linearized system.

5.2 Parameter Study

The parameter study section uses the G matrix to transform measurement errors into solution or angular errors. The G matrix is formed using truth data. Given measurement errors $\mathbf{m}$ are then used to calculate the errors that result from the geometry of the scene (e.g., $\mathbf{v}$) from Equation 5-33. Figure 5–4 shows a graphical representation of this process. This study assumes that all measurement errors are uncorrelated and can be specified by a value for the pixel variance denoted as σ_p^2. Note that the units for σ_p^2 are [pixels^2]. The G matrix, camera size, and number of pixels is used to transform σ_p^2 into angular errors by

$$p_{\text{area}} \left[\frac{\text{m}^2}{\text{pixels}^2} \right] = \left(\frac{\text{camera size [m]}}{\text{num pixels [pixel]}} \right)^2 \tag{5-34}$$

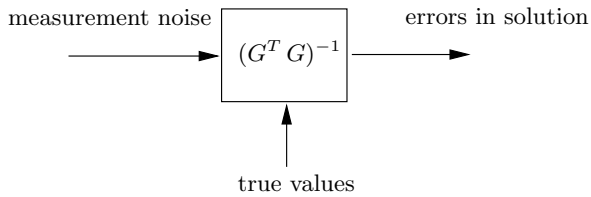

Figure 5–4: Graphical overview of the parameter study process using the GLEMA program

$$\sigma^2_{\text{soln}} = (G^T G)^{-1} \sigma_p^2 \, p_{\text{area}} \tag{5-35}$$

where the units for σ_{soln} are radians for $(\tilde{\phi}_1, \tilde{\theta}_1, \tilde{\psi}_1, \tilde{\theta}_2, \tilde{\psi}_2)$ and proportion of error in the estimate of range to the i-th control point for δp_i. For all the parameter study results, a one-pixel standard deviation is assumed. This simplifies Equation 5-35 to

$$\sigma^2_{\text{soln}} = (G^T G)^{-1} p_{\text{area}} \text{ for } \sigma_p = 1 \tag{5-36}$$

The δp results are not displayed in any of the parameter study results. This is the sacrifice that comes from ensuring that all control points are visible to both camera systems. That is, if a point is designated in the first camera system and cannot be seen by the second camera system then it is not used in the formulation of the G matrix.

In the following sections, the results for the angular terms $(\tilde{\phi}_1, \tilde{\theta}_1, \tilde{\psi}_1, \tilde{\theta}_2, \tilde{\psi}_2)$ are displayed in units of mrad. To following equation was used to convert each angular term of σ^2_{soln} to mrad:

$$\sigma_{\text{soln,i}}[\text{mrad}] = 1000 \left[\frac{\text{mrad}}{\text{rad}} \right] \cdot (\sigma^2_{\text{soln}})^{\frac{1}{2}} \tag{5-37}$$

The control points are designated in a pseudo-random fashion on the first cameras focal plane. The focal plane is divided up into an n row by m column grid. The control points are randomly placed within each grid box with the constraint that they cannot be within 25% of the grid boundaries. The allowed location of a control point within a single grid box is shown in Figure 5–5. An example focal plane with a 3 row by 3 column grid is shown in Figure 5–6. The control point designation in the GLEMA program is performed in the `markControlPoints.m` m–file.

5.2.1 Varying Camera Focal Length

The effect of varying the camera's focal length on the attitude estimates is shown in this section. The input parameters used for this study are shown in Table 5–7. The GLEMA driver file used in this study was `varying_focal_length.m`.

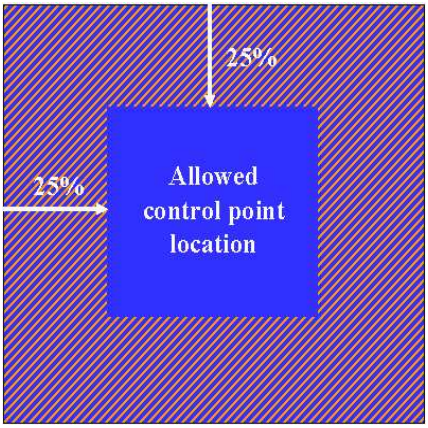

Figure 5–5: Allowed location of control points within each grid box of the focal plane.

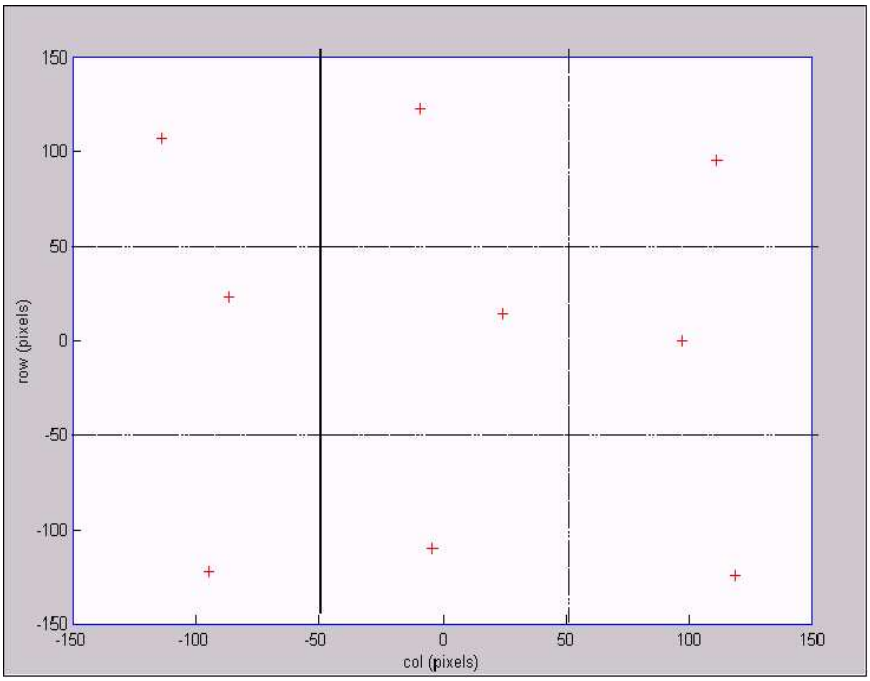

Figure 5–6: Control point locations on a focal plane with a 3 row by 3 column grid designation.

Table 5–7: Input parameters for varying camera focal length study

Parameter	Value	Description
β	$\frac{\pi}{4}$	Camera depression angle in radians.
f	**varied**	Camera focal length in meters.
ccd_pixels	384 288	Number of column and row pixels in the ccd array camera.
ccd_size	0.0042 0.0032	Size of the ccd array camera in meters.
cp cols	5	Number of control point columns on the ccd array camera.
cp rows	7	Number of control point rows on the ccd array camera.
$(\phi_{c1}^{c2}, \theta_{c1}^{c2}, \psi_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw of the relative attitude between *c1* and *c2*.
$(\phi_{ned}^{c2}, \theta_{ned}^{c2}, \psi_{ned}^{c2})$	(0,0,0)	Roll, pitch, yaw of the inertial attitude of the second camera system *c2*.
$(\tilde{\phi}_{c1}^{c2}, \tilde{\theta}_{c1}^{c2}, \tilde{\psi}_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors in the relative attitude between *c1* and *c2*.
$(\tilde{\phi}_{c}^{c2}, \tilde{\theta}_{c}^{c2}, \tilde{\psi}_{c}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors between the *c* (center) coordinate system and second camera system *c2*.
$\|\mathbf{a}\|$	10	Displacement vector magnitude in meters.
$\mathbf{v}^{enu}$	(0,1,0)	Direction of the displacement vector, or velocity vector.
pixel error	1	Measurement error of the control points in pixels.

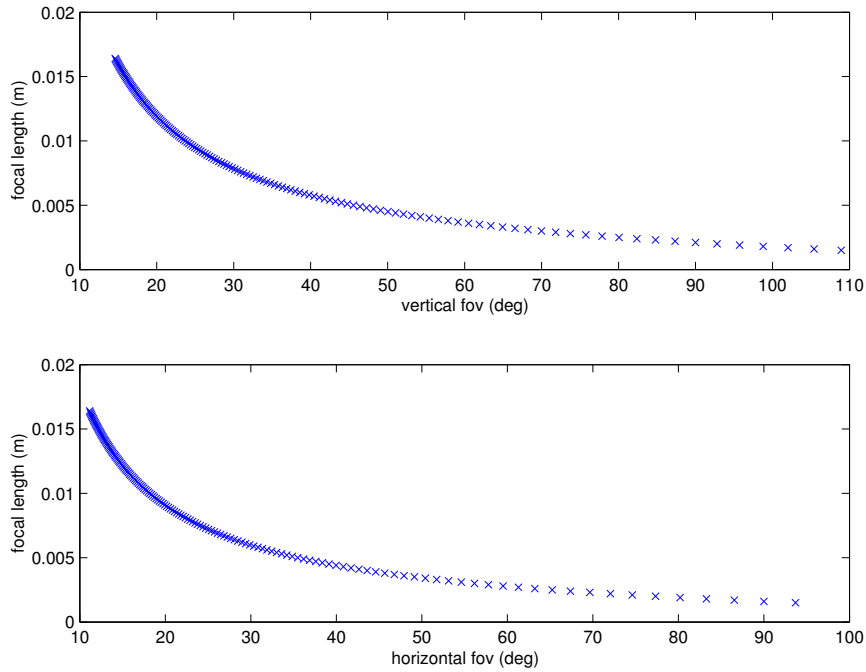

Figure 5–7: Varying FOV vs. focal length

The focal length of the camera system was varied from 0.0015 to 0.0164 meters. The lower limit, 0.0015 meters, corresponds to an unrealistic FOV region for the given camera size. The data generated in this realm can only be used for theoretical arguments. The camera FOV is related to the focal length of the camera by

$$\text{FOV[radians]} = 2 \cdot \arctan\left(\frac{\text{camera size [m]}}{2f\text{[m]}}\right) \tag{5-38}$$

The resulting data points used in this analysis is shown in Figure 5–7. The first iteration of the loop corresponds to the largest FOV and the last iteration corresponds to the smallest FOV. The vertical FOV has a maximum value of 108.92 degrees and a minimum value of 14.59 degrees. The horizontal FOV has a maximum value of 93.70 degrees and a minimum value of 11.14 degrees.

The control point locations on the focal planes of both cameras for the first and last iterations of the analysis loop are shown in Figures 5–8 and 5–9. The "x" symbols indicate the designated control point location on the first camera's focal plane, and the hexagon symbols indicate the calculated control point locations on the second camera's focal plane. A line is drawn to connect the corresponding control point locations in the first and second camera system. Notice that the last two rows of the control points (where the first row is located on the bottom of the

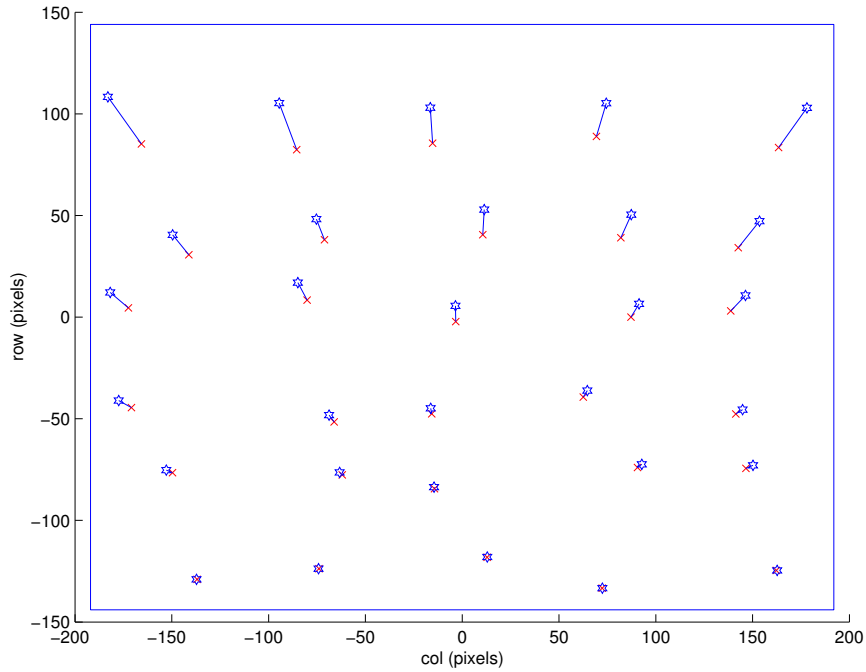

Figure 5–8: Control point locations on the focal planes of both cameras for the first iteration for varying camera FOV.

focal plane) disappear in focal plane of the last iteration. This is due to the control points going out of the second camera's FOV.

The *3-D* view of the scene for the first and last iterations is shown in Figures 5–10 and 5–11. The control points are marked by plus symbols. The first camera position is marked by a square symbol. The second camera position is marked by a triangle.

The results for the computed attitude standard deviations are shown in Figures 5–12 and 5–13. The discontinuities in the various plots are due to control points going out of the second camera's FOV.

5.2.2 Varying Camera Depression Angle

The effect of varying the camera's depression angle β on the attitude estimates is shown in this section. The input parameters used for this study are shown in Table 5–8. The GLEMA driver file used in this study was `varying_beta.m`.

The camera depression angle was varied from 0 to 179.9 degrees. The first iteration corresponds to a camera looking directly ahead of the vehicle. The last iteration corresponds to a camera looking almost directly behind the vehicle. Since the projection onto the $x-y$ plane of the velocity and camera line of sight vectors are identical in this study, a forward looking camera gets

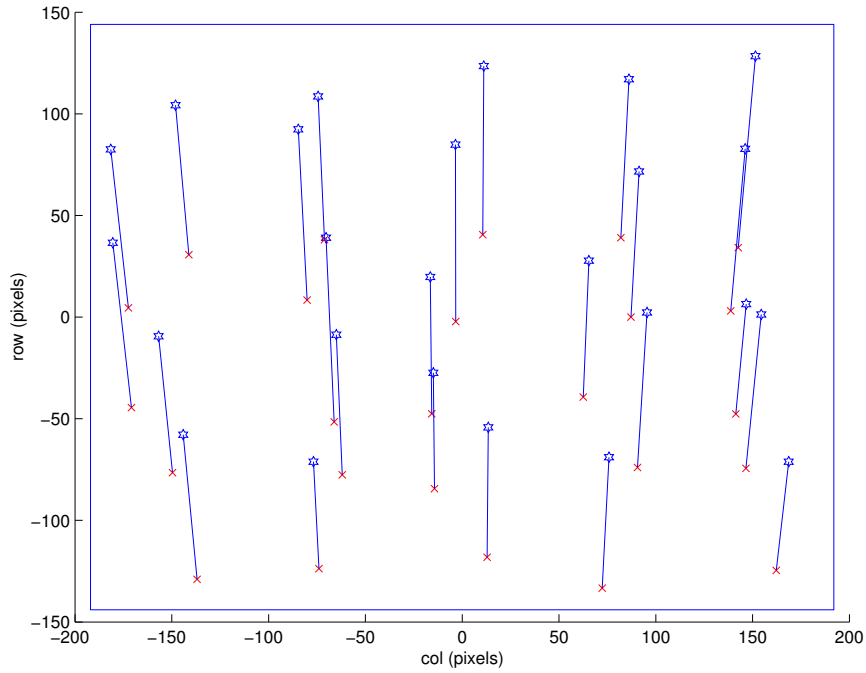

Figure 5–9: Control point locations on the focal planes of both cameras for the last iteration for varying camera FOV.

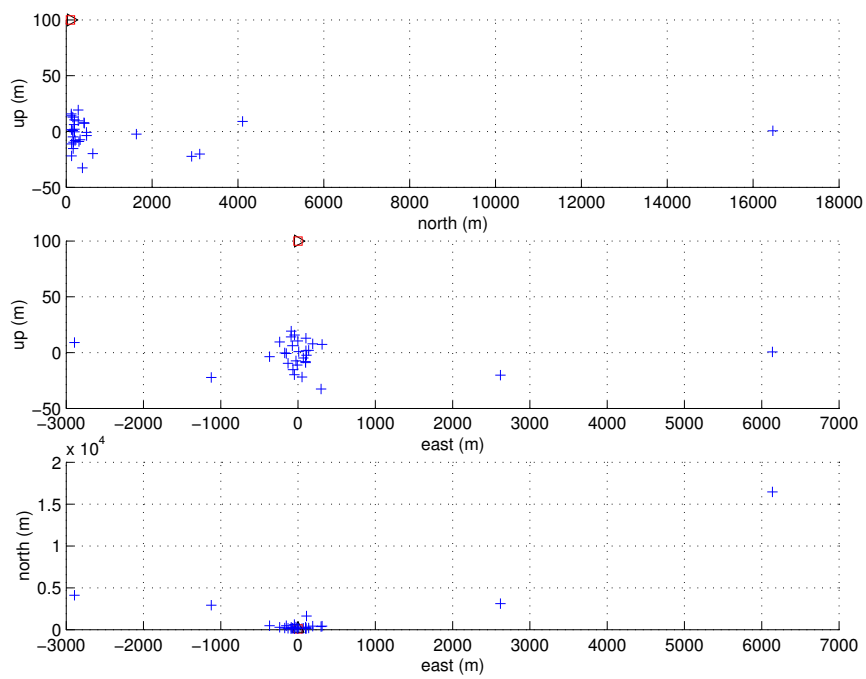

Figure 5–10: Three-dimensional view of the scene for the first iteration for a varying focal length.

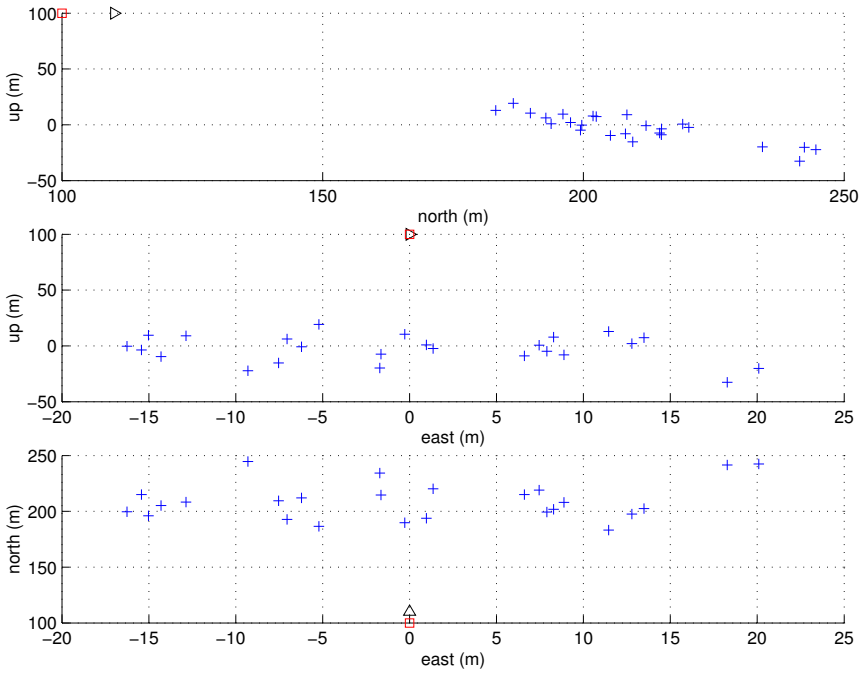

Figure 5–11: Three-dimensional view of the scene for the last iteration for a varying focal length.

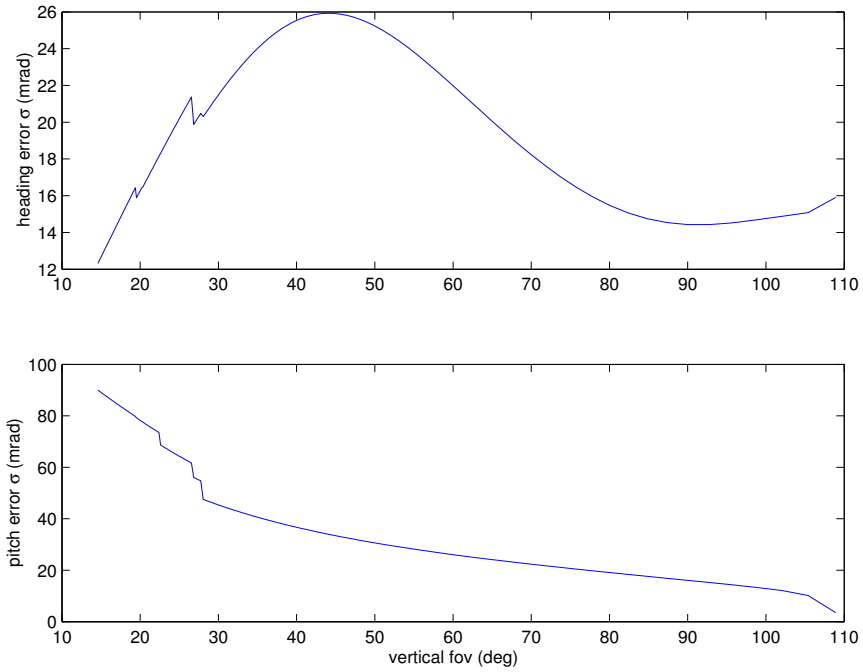

Figure 5–12: Varying camera focal length results - ned to c2

Table 5–8: Input parameters for varying camera depression angle β study

Parameter	Value	Description
β	**varied**	Camera depression angle in radians.
f	0.005	Camera focal length in meters.
ccd_pixels	384 288	Number of column and row pixels in the ccd array camera.
ccd_size	0.0042 0.0032	Size of the ccd array camera in meters.
cp cols	5	Number of control point columns on the ccd array camera.
cp rows	7	Number of control point rows on the ccd array camera.
$(\phi_{c1}^{c2}, \theta_{c1}^{c2}, \psi_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw of the relative attitude between *c1* and *c2*.
$(\phi_{ned}^{c2}, \theta_{ned}^{c2}, \psi_{ned}^{c2})$	(0,0,0)	Roll, pitch, yaw of the inertial attitude of the second camera system *c2*.
$(\tilde{\phi}_{c1}^{c2}, \tilde{\theta}_{c1}^{c2}, \tilde{\psi}_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors in the relative attitude between *c1* and *c2*.
$(\tilde{\phi}_{c}^{c2}, \tilde{\theta}_{c}^{c2}, \tilde{\psi}_{c}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors between the *c* (center) coordinate system and second camera system *c2*.
$\|\mathbf{a}\|$	10	Displacement vector magnitude in meters.
$\mathbf{v}^{enu}$	(0,1,0)	Direction of the displacement vector, or velocity vector.
pixel error	1	Measurement error of the control points in pixels.

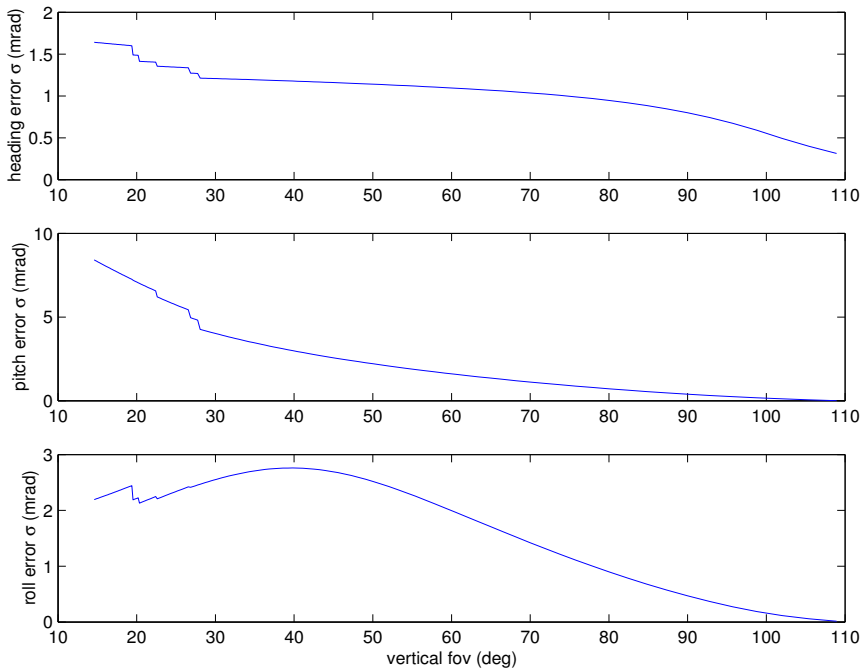

Figure 5–13: Varying camera focal length results - c1 to c2

closer to the points while a rear-looking camera gets farther away. The camera was allowed to be rear looking to show this difference.

The control point locations on the focal planes of both cameras for the β equaling 45, 90, and 135 degrees of the analysis loop are shown in Figures 5–14 through 5–16. The "x" symbols indicate the designated control point location on the first camera's focal plane, and the hexagon symbols indicate the calculated control point locations on the second camera's focal plane. A line is drawn to connect the corresponding control point locations in the first and second camera system. Notice that the last two rows of the control points (where the first row is located on the bottom of the focal plane) disappear in the focal plane of the last iteration. This is due to the control points going out of the second camera's FOV.

The *3-D* view of the scene for β equaling 45, 90, and 135 degrees is shown in Figures 5–17 through 5–19. The control points are marked by plus symbols. The first camera position is marked by a square symbol. The second camera position is marked by a triangle. The location of the control points move as expected from (1) in front to (2) directly below to (3) behind the camera systems when β equals 45, 90, and 135 degrees respectively.

The results for the computed attitude standard deviations are shown in Figures 5–20 and 5–21. The discontinuities in the various plots are due to control points going out of the second

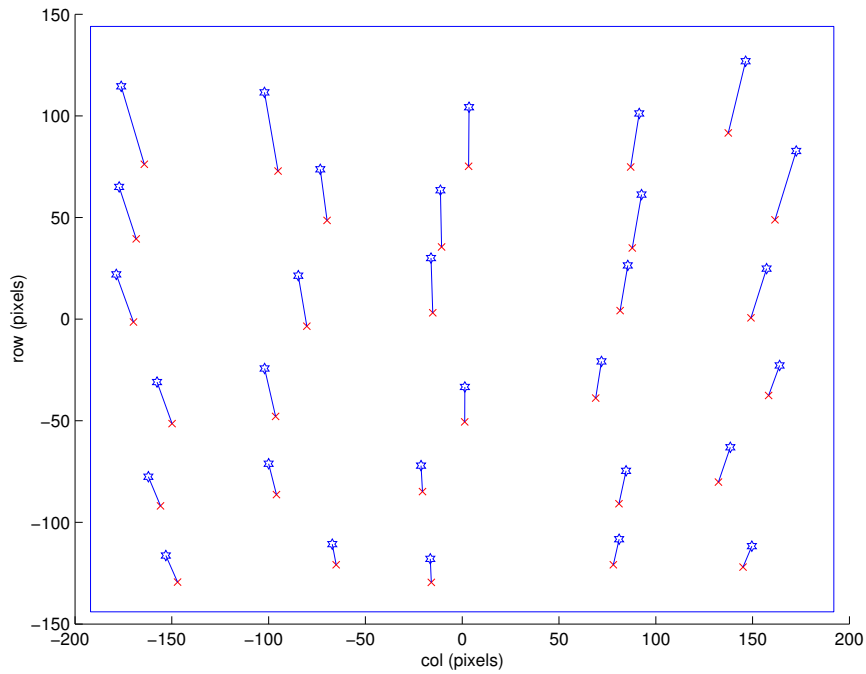

Figure 5–14: Control point locations on focal plane for it $\beta = 45$ degrees for the varying camera depression angle study

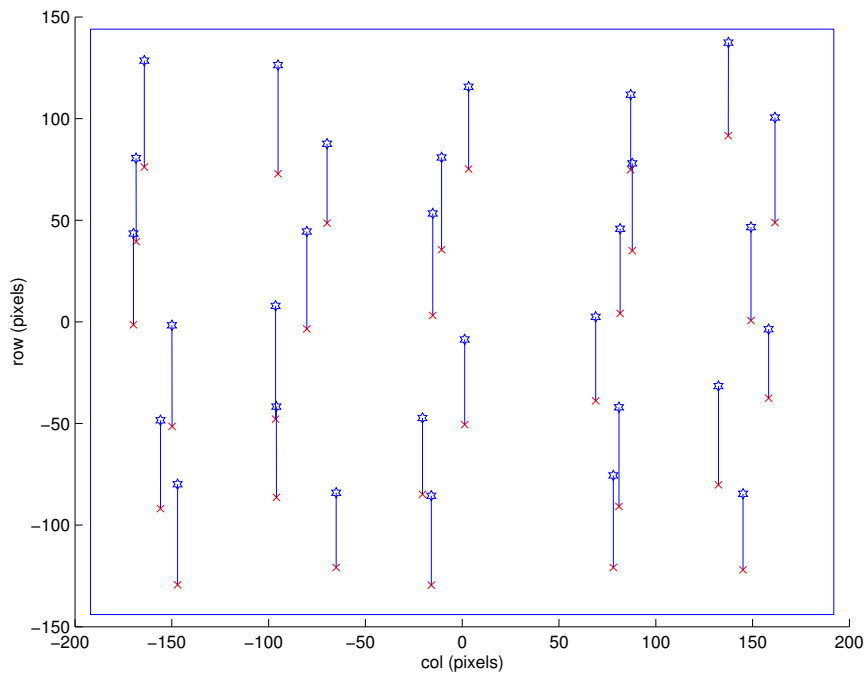

Figure 5–15: Control point locations on focal plane for it $\beta = 90$ degrees for the varying camera depression angle study

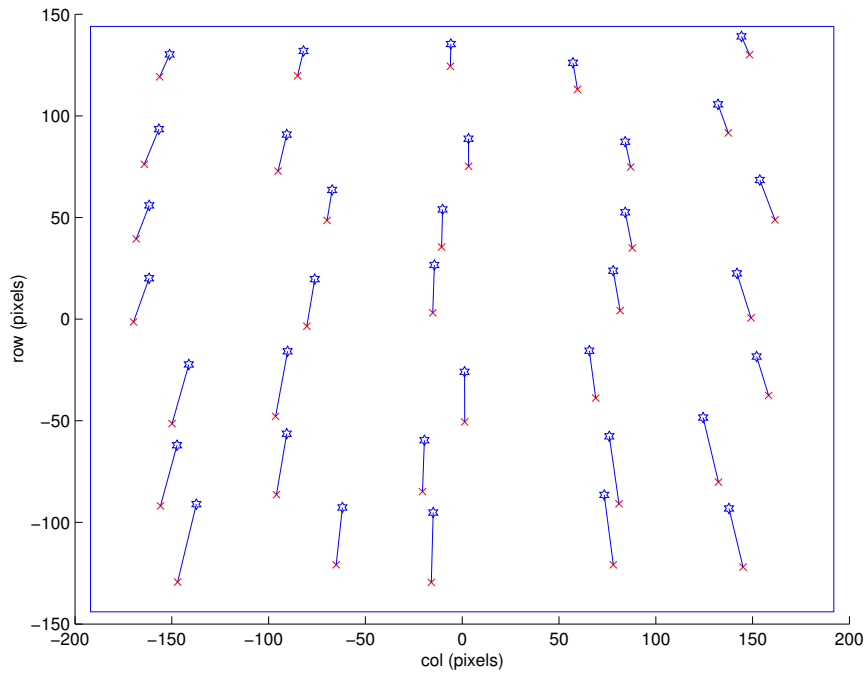

Figure 5–16: Control point locations on focal plane for it $\beta = 135$ degrees for the varying camera depression angle study

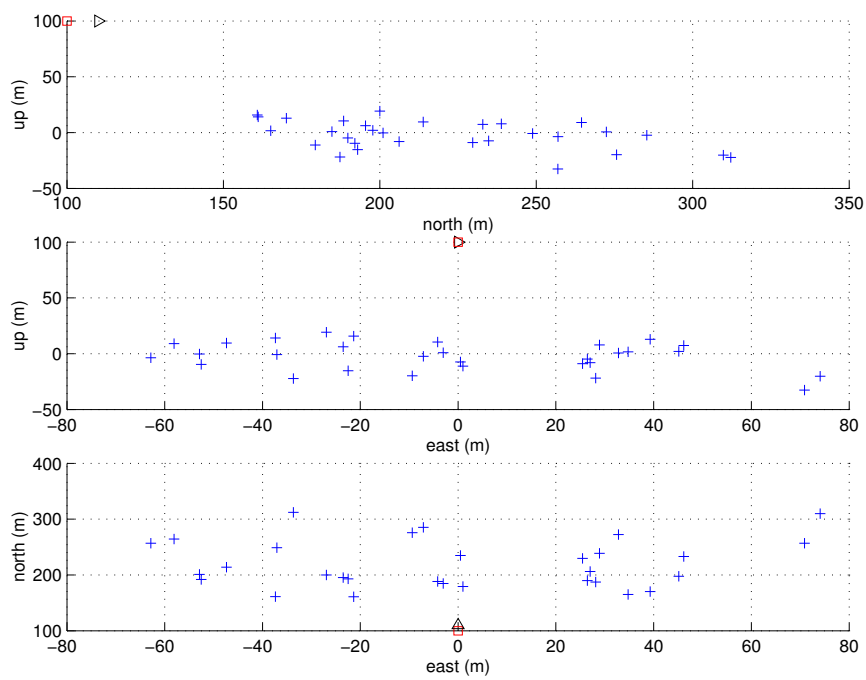

Figure 5–17: Three-dimensional view of the scene for $\beta = 45$ degrees for the varying camera depression angle study.

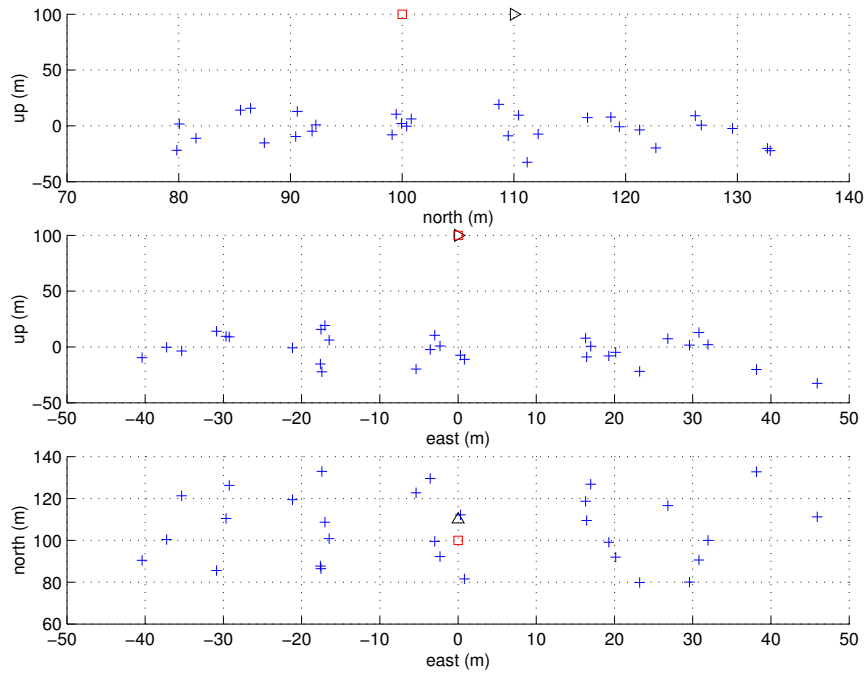

Figure 5–18: Three-dimensional view of the scene for $\beta = 90$ degrees for the varying camera depression angle study.

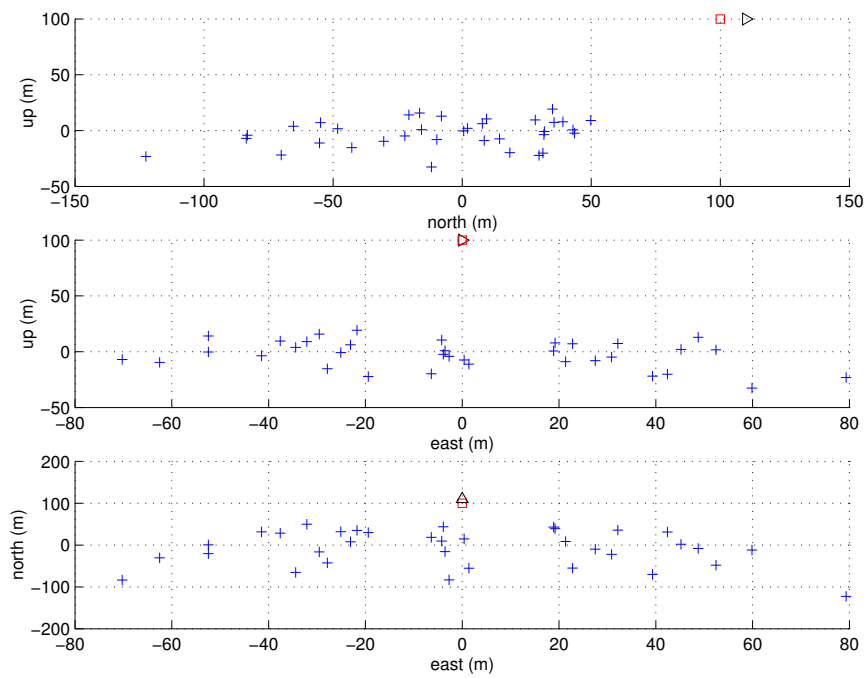

Figure 5–19: Three-dimensional view of the scene for $\beta = 135$ degrees for the varying camera depression angle study.

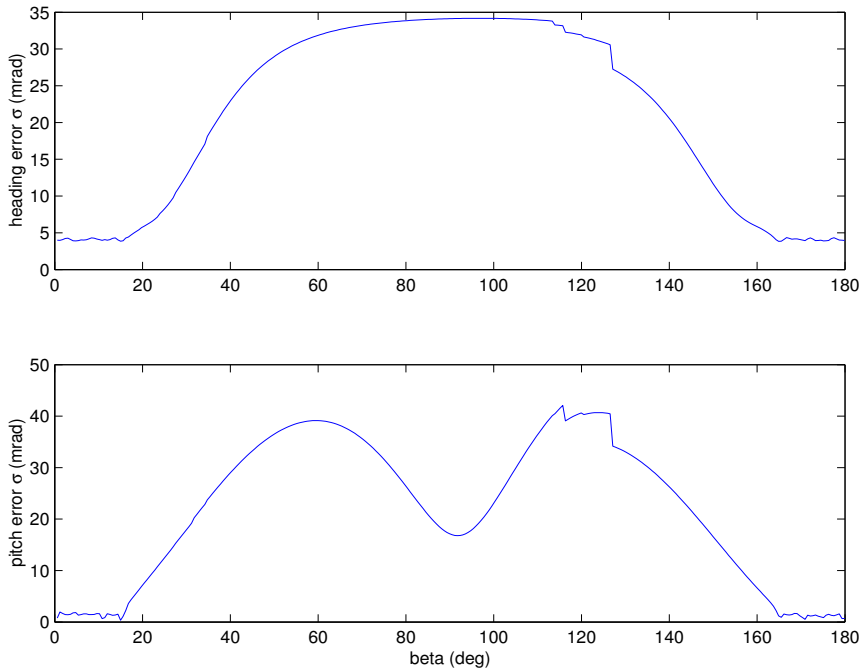

Figure 5–20: Varying camera depression angle results - ned to c2

camera's FOV. The non-symmetric behavior about the 90° camera depression angle is due to (1) the different control point locations which overlap in the two FOVs, and (2) the difference between forward and rear-looking camera (i.e., approaching and receding from control points).

5.2.3 Varying Displacement Vector Magnitude Between Imaging Systems

The effect of varying the displacement vector magnitude between camera systems on the attitude estimates is shown in this section. The input parameters used for this study are shown in Table 5–9. The GLEMA driver file used in this study was `varying_displacement_vector.m`.

The magnitude of the displacement vector between camera systems $\|\mathbf{a}\|$ was varied from 1.0 to 20.9 meters. The first iteration corresponds to camera $\|\mathbf{a}\| = 1.0$ and the last iteration corresponds to $\|\mathbf{a}\| = 20.9$.

The control point locations on the focal planes of both cameras for the $\|\mathbf{a}\| = 1.0$, 5.9, 10.9, 15.9 and 20.9 meters are shown in Figures 5–22 through 5–26. The "x" symbols indicate the designated control point location on the first camera's focal plane, and the hexagon symbols indicate the calculated control point locations on the second camera's focal plane. A line is drawn to connect the corresponding control point locations in the first and second camera system. Notice that the last two rows of the control points (where the first row is located on the bottom of the

Table 5–9: Input parameters for varying displacement vector magnitude between imaging systems.

Parameter	Value	Description
β	$\frac{\pi}{3}$	Camera depression angle in radians.
f	0.005	Camera focal length in meters.
ccd_pixels	384 288	Number of column and row pixels in the ccd array camera.
ccd_size	0.0042 0.0032	Size of the ccd array camera in meters.
cp cols	5	Number of control point columns on the ccd array camera.
cp rows	7	Number of control point rows on the ccd array camera.
$(\phi_{c1}^{c2}, \theta_{c1}^{c2}, \psi_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw of the relative attitude between *c1* and *c2*.
$(\phi_{ned}^{c2}, \theta_{ned}^{c2}, \psi_{ned}^{c2})$	(0,0,0)	Roll, pitch, yaw of the inertial attitude of the second camera system *c2*.
$(\tilde{\phi}_{c1}^{c2}, \tilde{\theta}_{c1}^{c2}, \tilde{\psi}_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors in the relative attitude between *c1* and *c2*.
$(\tilde{\phi}_{c}^{c2}, \tilde{\theta}_{c}^{c2}, \tilde{\psi}_{c}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors between the c (center) coordinate system and second camera system *c2*.
$\|\mathbf{a}\|$	**varied**	Displacement vector magnitude in meters.
$\mathbf{v}^{enu}$	(0,1,0)	Direction of the displacement vector, or velocity vector.
pixel error	1	Measurement error of the control points in pixels.

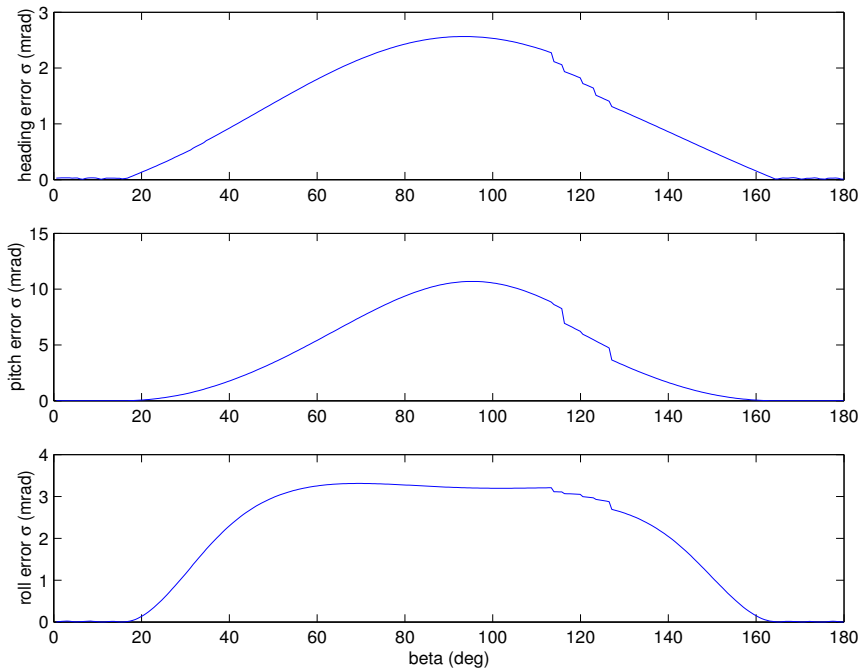

Figure 5–21: Varying camera depression angle results - c1 to c2

focal plane) disappear by the time $\|\mathbf{a}\| = 20.9$. This is due to the control points going out of the second camera's FOV.

The *3-D* view of the scene for $\|\mathbf{a}\| = 1.0$, 5.9, 10.9, 15.9 and 20.9 meters is shown in Figures 5–27 through 5–31. The control points are marked by plus symbols. The first camera position is marked by a square symbol. The second camera position is marked by a triangle. The second camera moves from right to left as the displacement vector is increased from 1 to 20.9 meters. Referring to the top portion of each figure, notice that the control points corresponding to the disappearing rows are seen to gradually disappear from the left side of the control point grouping.

The results for the computed attitude standard deviations are shown in Figures 5–32 through 5–35. The discontinuities in the various plots are due to control points going out of the second camera's FOV.

5.2.4 Varying Number of Control Points

The effect of varying the number of control points marked on the focal plane on the attitude estimates is shown in this section. The input parameters used for this study are shown in Table 5–10. The GLEMA driver files used in this study were `varying_num_cp_cols.m` and `varying_num_cp_rows.m`.

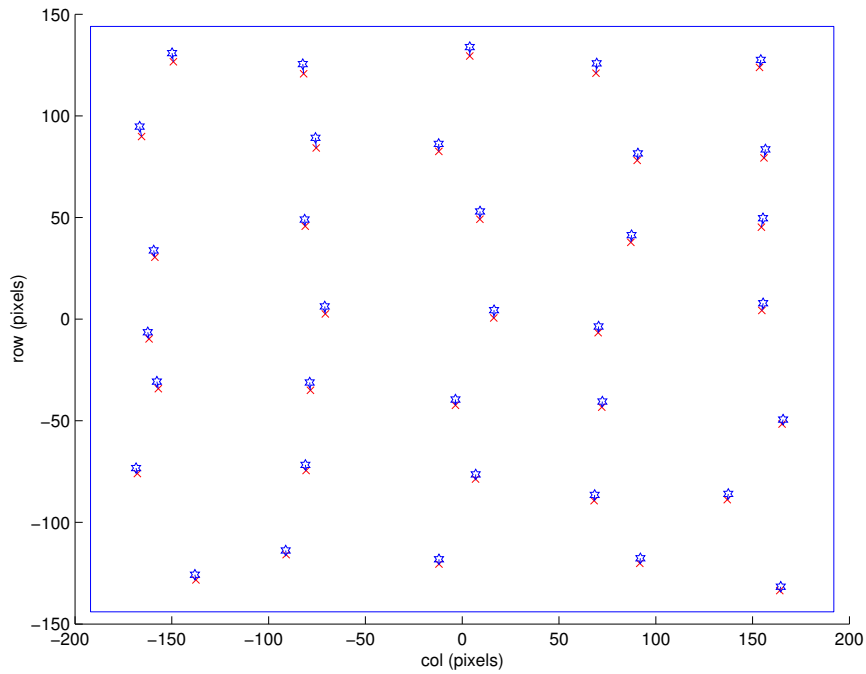

Figure 5–22: Control point locations on focal plane for a = 1.0 meters for the varying displacement vector magnitude study.

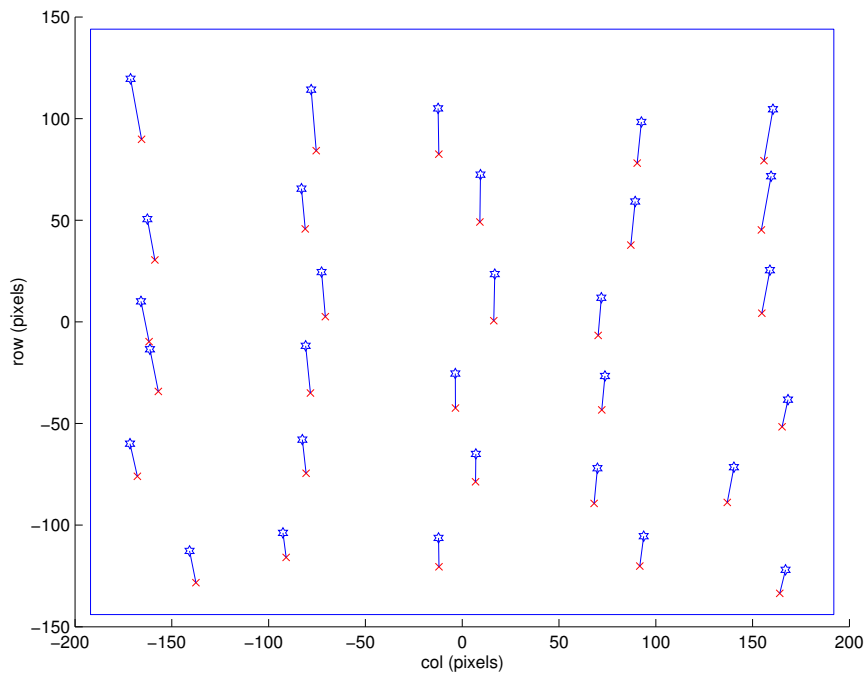

Figure 5–23: Control point locations on focal plane for a = 5.9 meters for the varying displacement vector magnitude study.

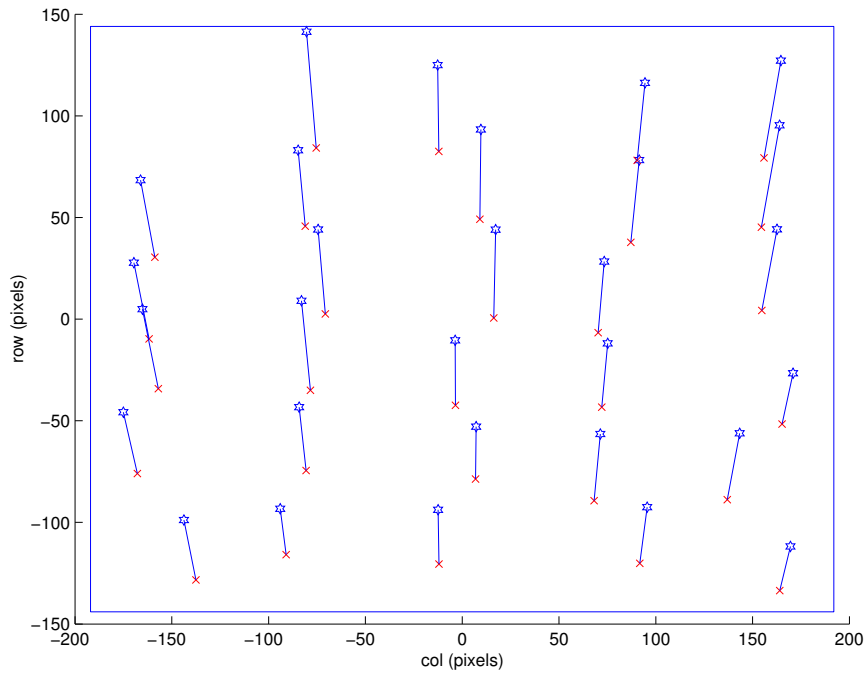

Figure 5–24: Control point locations on focal plane for a = 10.9 meters for the varying displacement vector magnitude study.

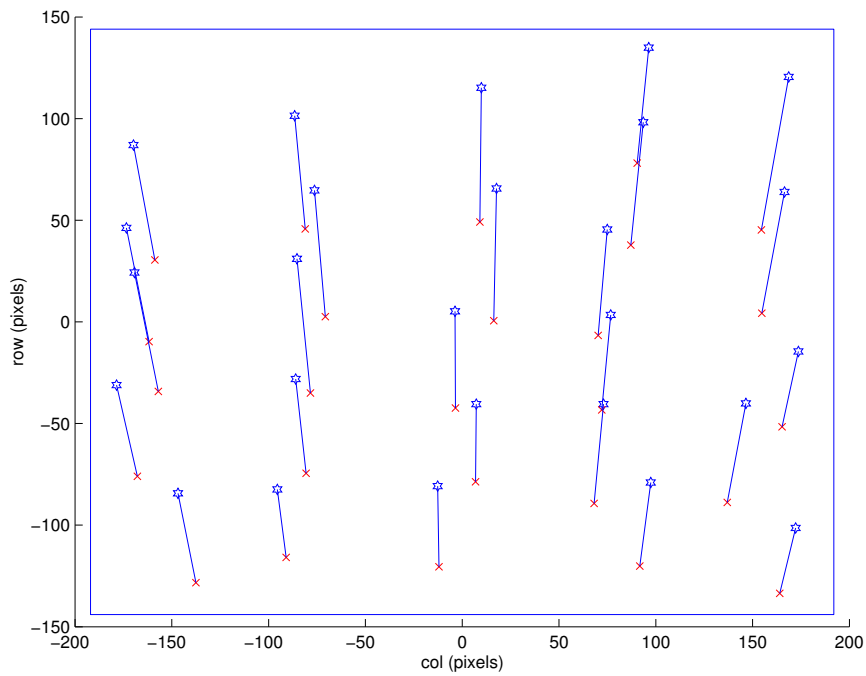

Figure 5–25: Control point locations on focal plane for a = 15.9 meters for the varying displacement vector magnitude study.

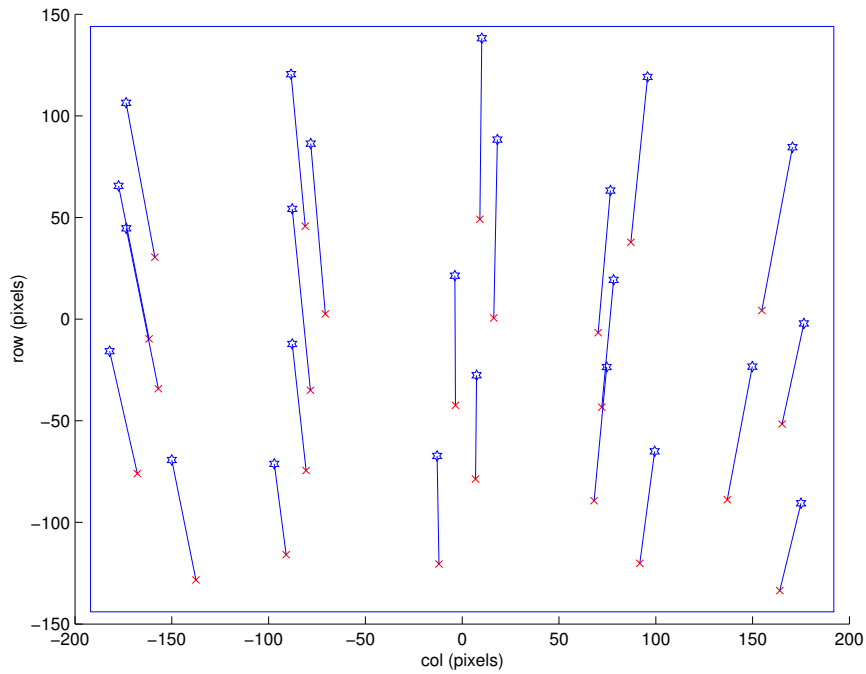

Figure 5–26: Control point locations on focal plane for a = 20.9 meters for the varying displacement vector magnitude study.

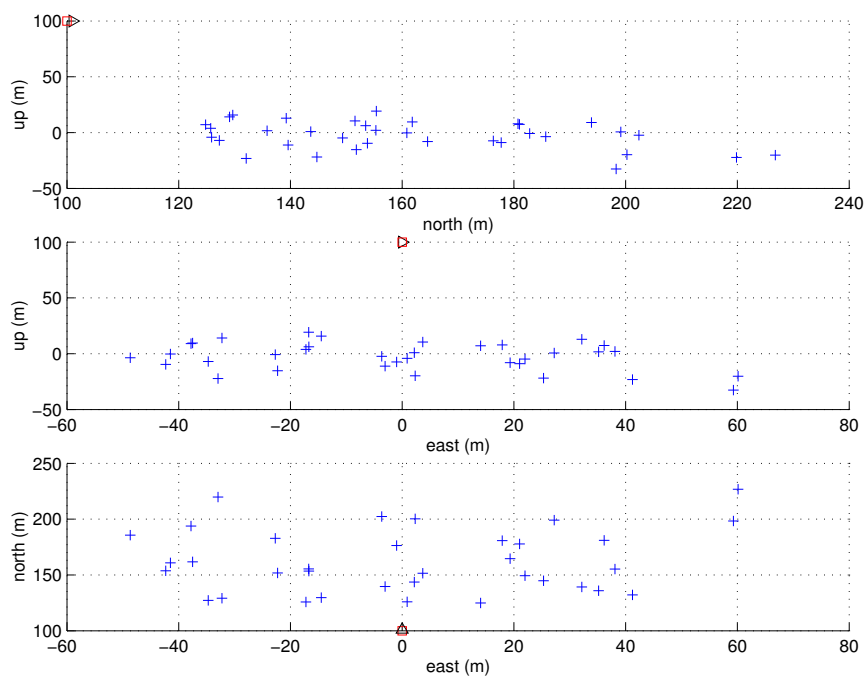

Figure 5–27: Three-dimensional view of the scene for a = 1.0 meter for the varying displacement vector magnitude between imaging systems study.

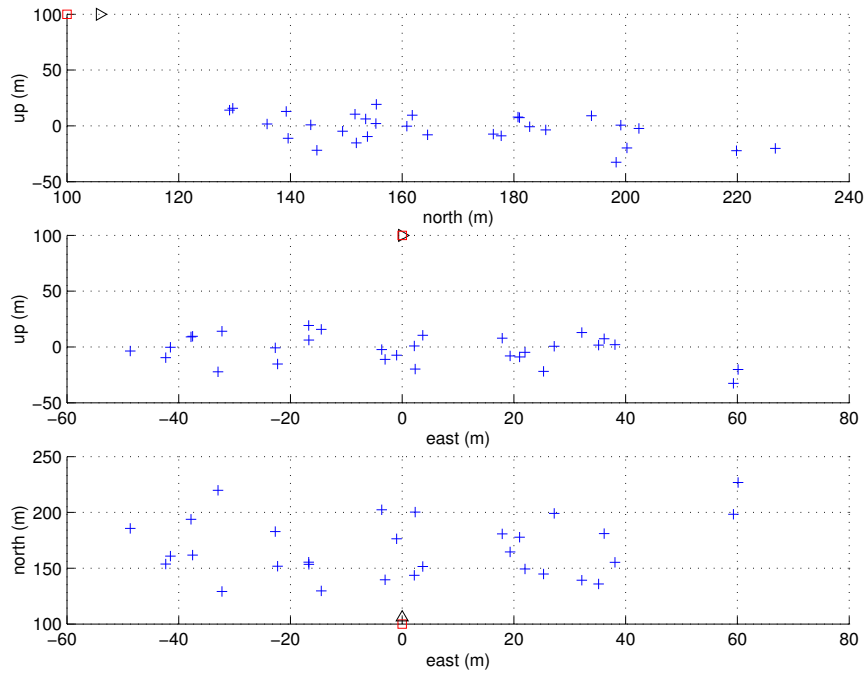

Figure 5–28: Three-dimensional view of the scene for a = 5.9 meter for the varying displacement vector magnitude between imaging systems study.

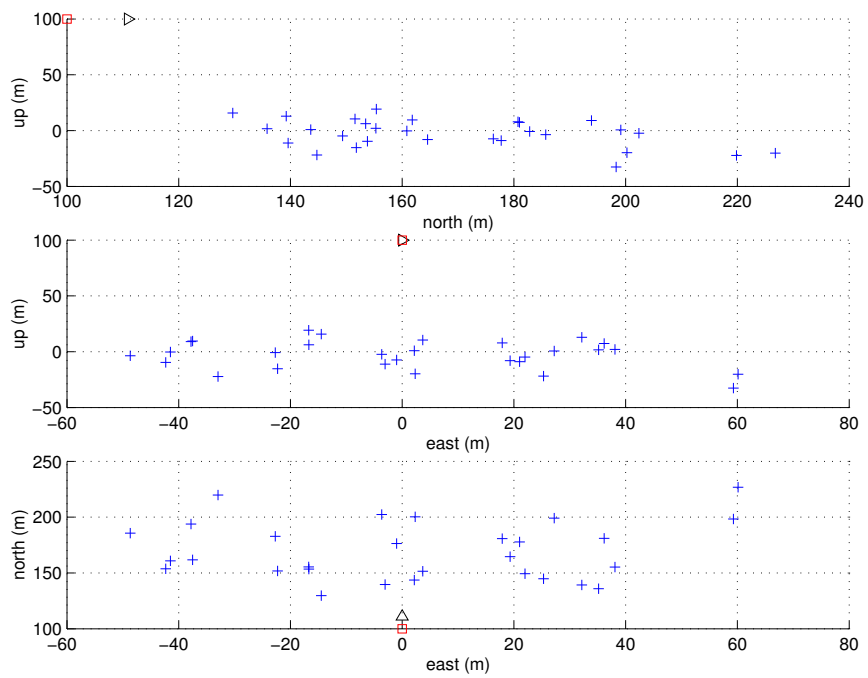

Figure 5–29: Three-dimensional view of the scene for a = 10.9 meter for the varying displacement vector magnitude between imaging systems study.

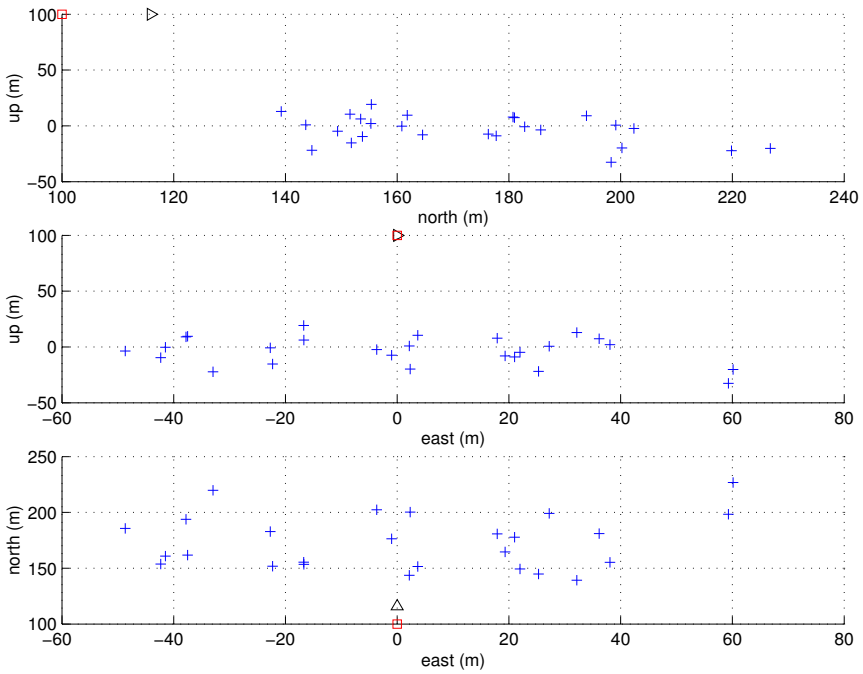

Figure 5–30: Three-dimensional view of the scene for a = 15.9 meter for the varying displacement vector magnitude between imaging systems study.

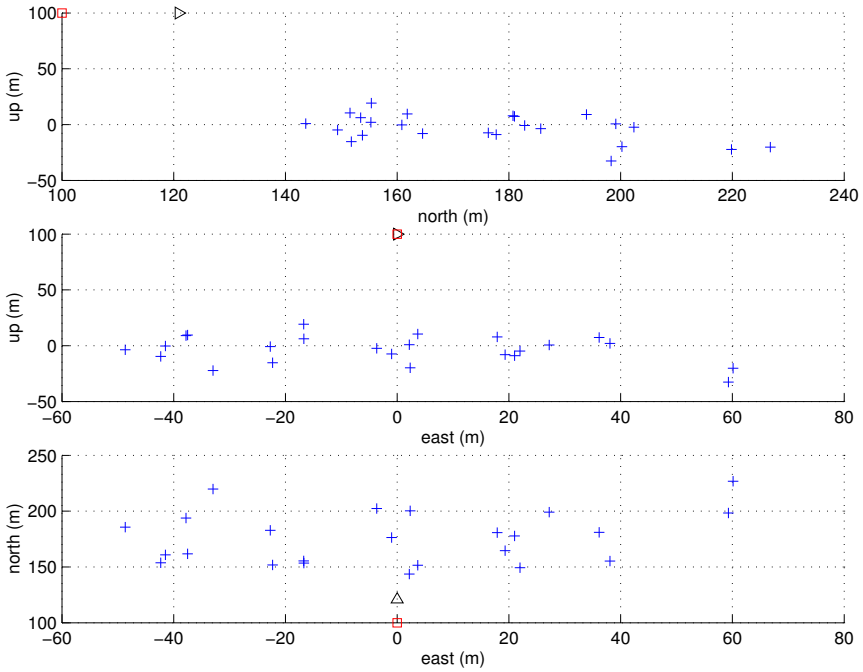

Figure 5–31: Three-dimensional view of the scene for a = 20.9 meter for the varying displacement vector magnitude between imaging systems study.

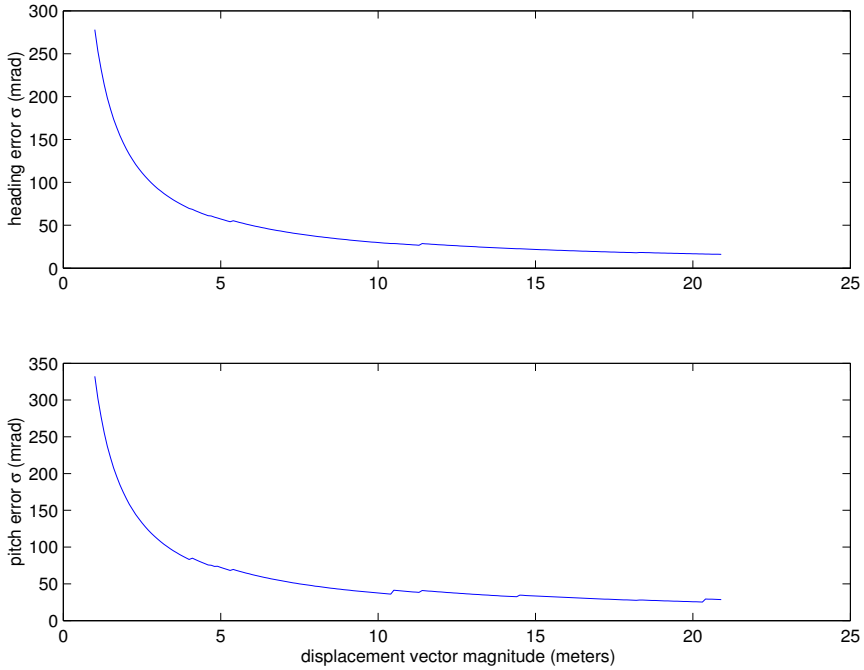

Figure 5–32: Varying displacement vector magnitude between imaging systems results for a = 1.0 to 20.9 meters - ned to c2

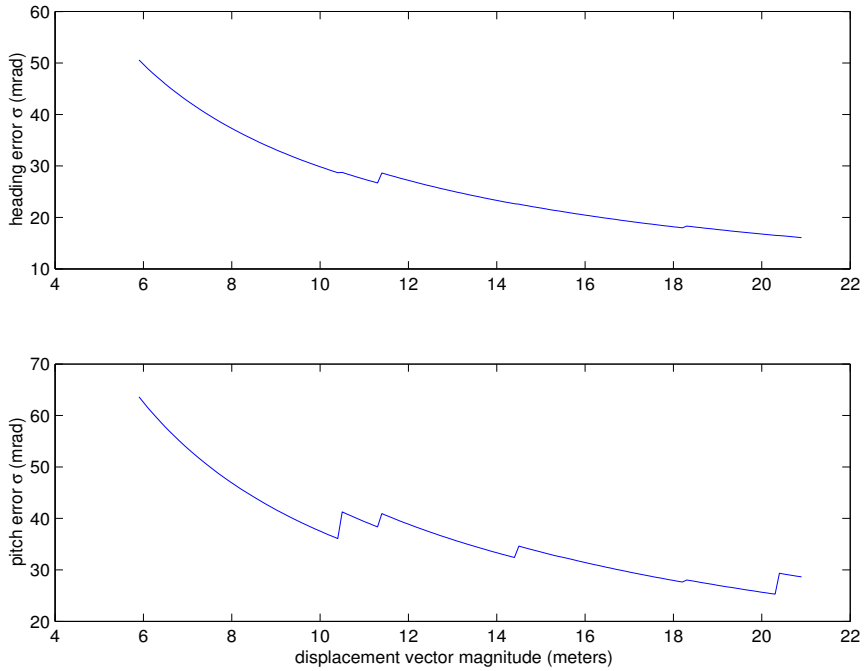

Figure 5–33: Varying displacement vector magnitude between imaging systems results for a = 5.9 to 20.9 meters - ned to c2

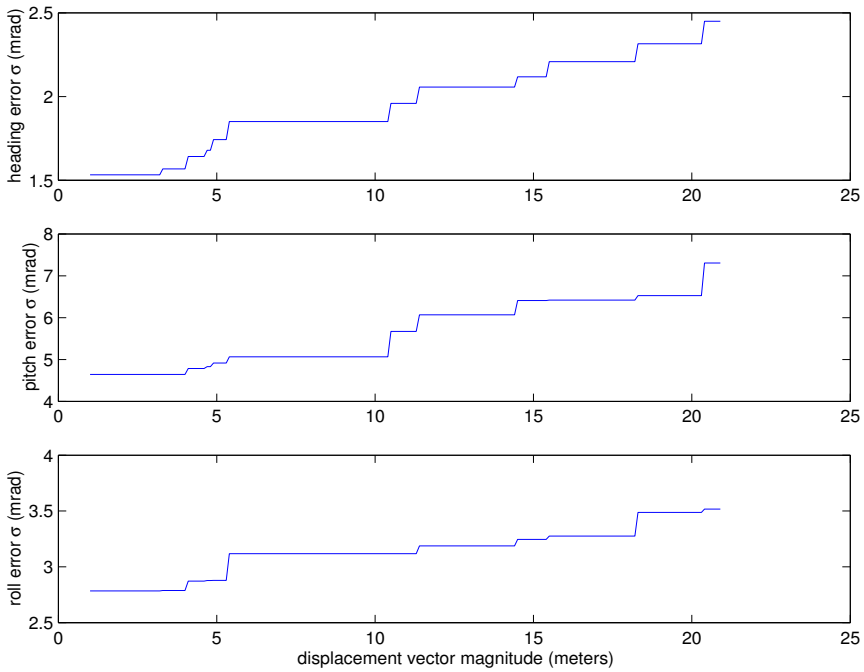

Figure 5–34: Varying displacement vector magnitude between imaging systems results for a = 1.0 to 20.9 meters - c1 to c2

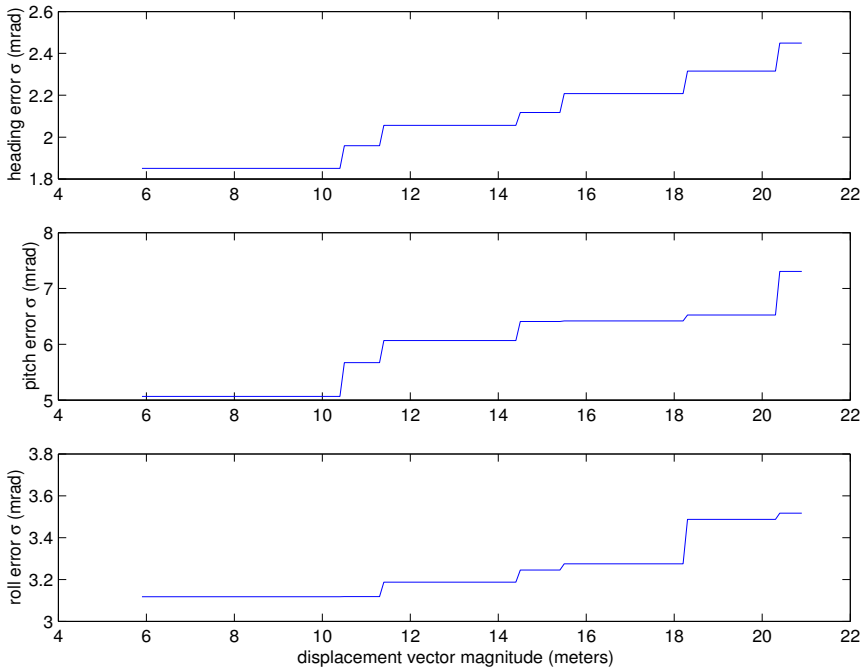

Figure 5–35: Varying displacement vector magnitude between imaging systems results for a = 5.9 to 20.9 meters - c1 to c2

Table 5–10: Input parameters for varying the number of control points study.

Parameter	Value	Description
β	$\frac{\pi}{3}$	Camera depression angle in radians.
f	0.005	Camera focal length in meters.
ccd_pixels	384 288	Number of column and row pixels in the ccd array camera.
ccd_size	0.0042 0.0032	Size of the ccd array camera in meters.
cp cols	**varied**	Number of control point columns on the ccd array camera.
cp rows	**varied**	Number of control point rows on the ccd array camera.
$(\phi_{c1}^{c2}, \theta_{c1}^{c2}, \psi_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw of the relative attitude between *c1* and *c2*.
$(\phi_{ned}^{c2}, \theta_{ned}^{c2}, \psi_{ned}^{c2})$	(0,0,0)	Roll, pitch, yaw of the inertial attitude of the second camera system *c2*.
$(\tilde{\phi}_{c1}^{c2}, \tilde{\theta}_{c1}^{c2}, \tilde{\psi}_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors in the relative attitude between *c1* and *c2*.
$(\tilde{\phi}_{c}^{c2}, \tilde{\theta}_{c}^{c2}, \tilde{\psi}_{c}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors between the c (center) coordinate system and second camera system *c2*.
$\|\mathbf{a}\|$	10	Displacement vector magnitude in meters.
$\mathbf{v}^{enu}$	(0,1,0)	Direction of the displacement vector, or velocity vector.
pixel error	1	Measurement error of the control points in pixels.

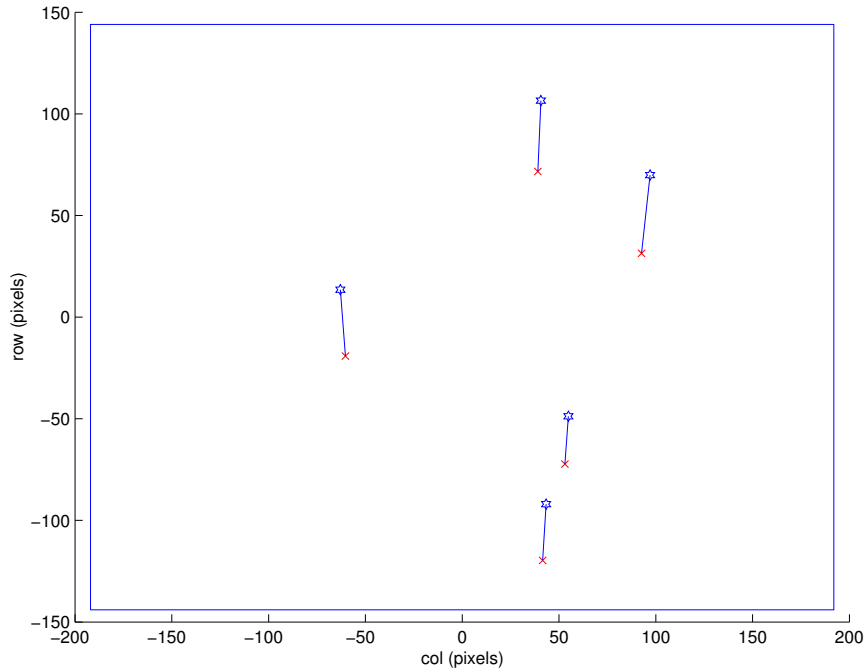

Figure 5–36: Control point locations on focal plane for a column count of 1 and a row count of 6.

The number of rows and number of columns of control points were varied separately to show the impact that each had on the accuracy of the attitude estimate.

Varying number of control point column count

The number of control point column counts was varied from 1 to 20 while the number of rows was fixed at 6. The first iteration corresponds to a column count or 1 and the last iteration corresponds to a column count of 20. The GLEMA driver file used for this part of the study was `varying_num_cp_cols.m`.

The control point locations on the focal planes of both cameras for the column count of 1, 10, and 20 are shown in Figures 5–36 through 5–38. The "x" symbols indicate the designated control point location on the first camera's focal plane, and the hexagon symbols indicate the calculated control point locations on the second camera's focal plane. A line is drawn to connect the corresponding control point locations in the first and second camera system. Notice that the last column in all three of the focal plane views is missing. This is due to the control points going out of the second camera's FOV.

The *3-D* view of the scene for column counts of 1, 10, and 20 and a row count of 6 is shown in Figures 5–39 through 5–41. The control points are marked by plus symbols. The first camera position is marked by a square symbol. The second camera position is marked by a triangle.

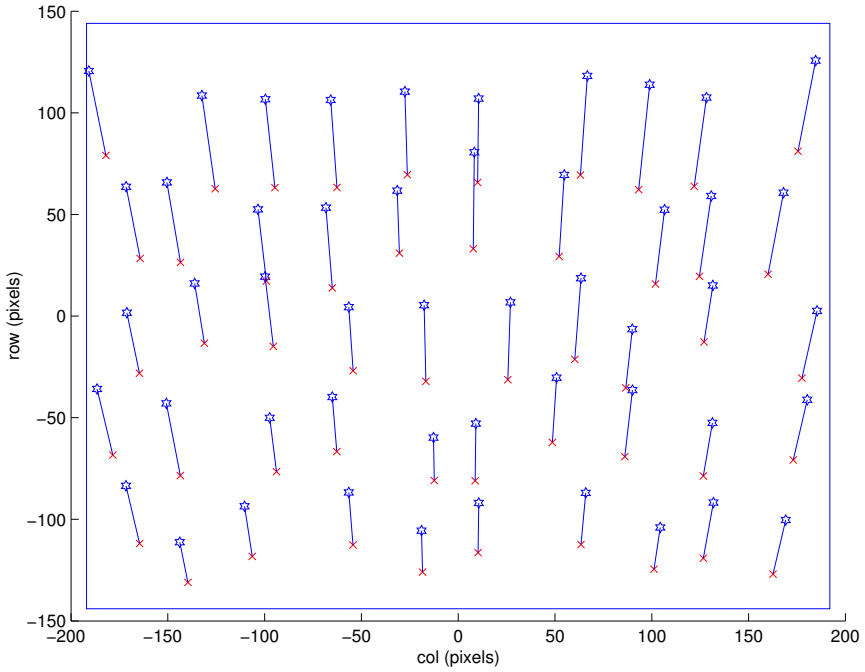

Figure 5–37: Control point locations on focal plane for column count of 10 and a row count of 6.

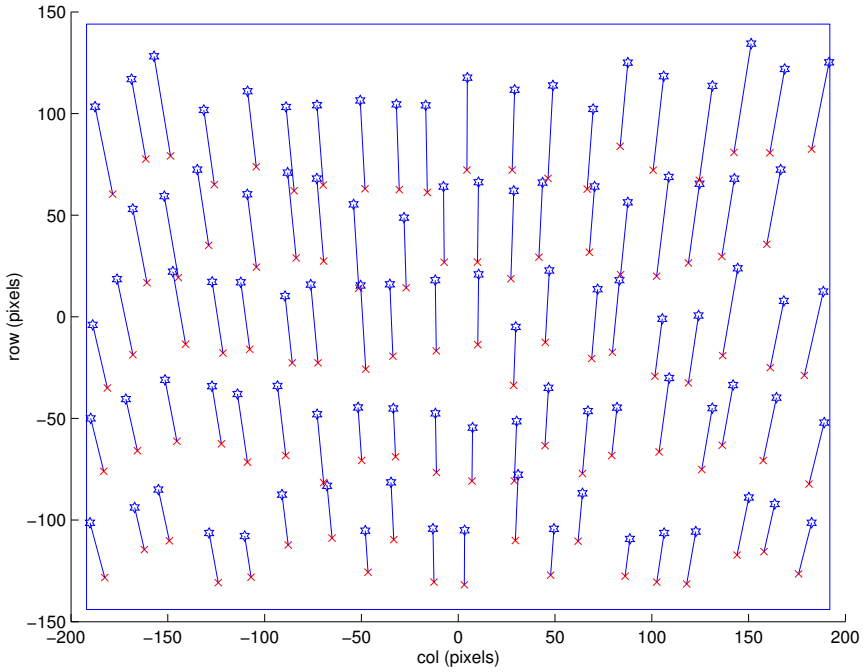

Figure 5–38: Control point locations on focal plane for column count of 20 and a row count of 6.

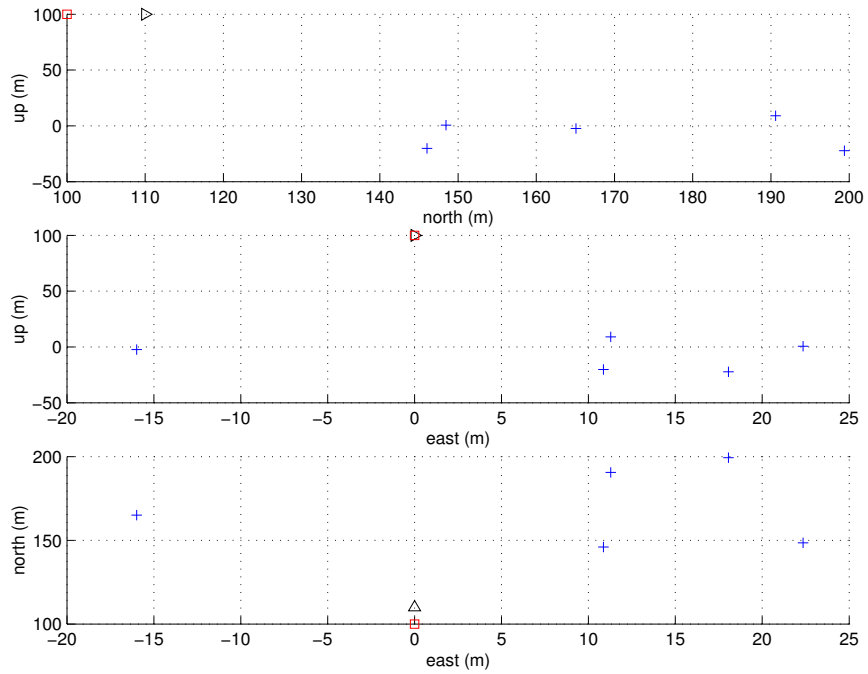

Figure 5–39: Three-dimensional view of the scene for a column count of 1 and a row count of 6 for the varying number of control points study.

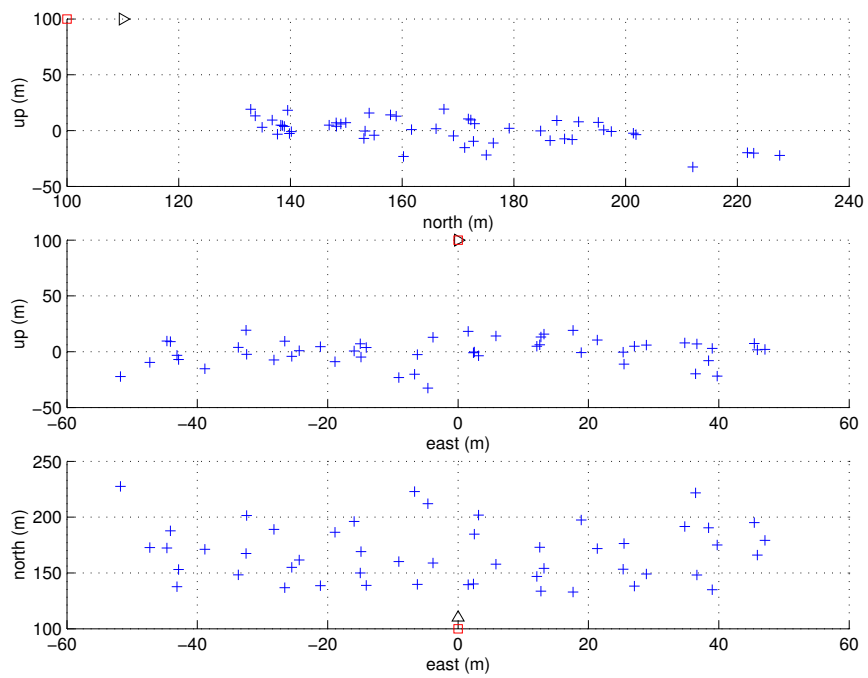

Figure 5–40: Three-dimensional view of the scene for a column count of 10 and a row count of 6 for the varying number of control points study.

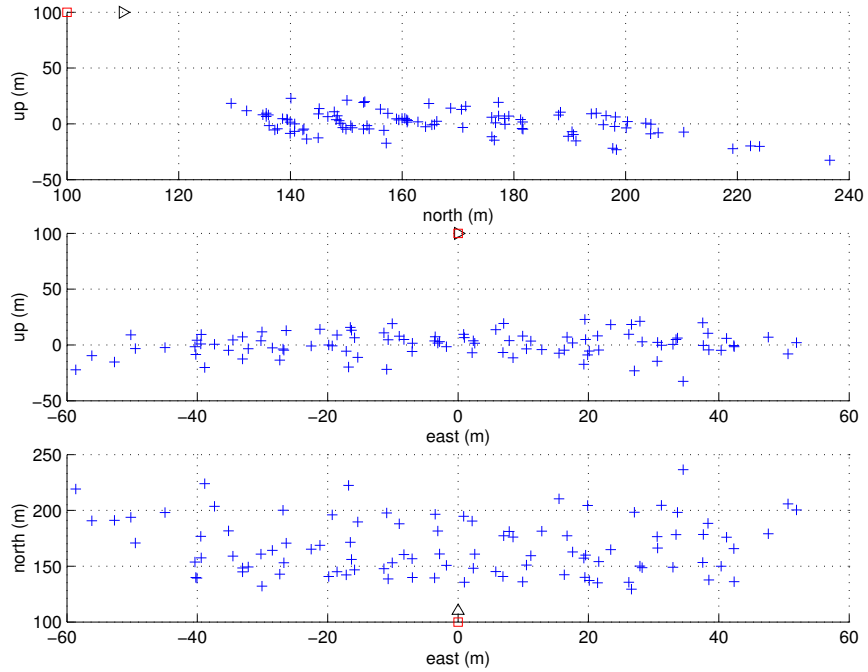

Figure 5–41: Three-dimensional view of the scene for a column count of 20 and a row count of 6 for the varying number of control points study.

The results for the computed attitude standard deviations are shown in Figures 5–42 and 5–43. As expected, increasing the number of control points decreases the error values.

Varying number of control point row count

The number of control point row count was varied from 1 to 20 while the number of columns was fixed at 6. The first iteration corresponds to a row count or 1 and the last iteration corresponds to a row count of 20. The GLEMA driver file used for this part of the study was `varying_num_cp_rows.m`.

The control point locations on the focal planes of both cameras for the row counts of 1, 10, and 20 are shown in Figures 5–44 through 5–46. The "x" symbols indicate the designated control point location on the first camera's focal plane, and the hexagon symbols indicate the calculated control point locations on the second camera's focal plane. A line is drawn to connect the corresponding control point locations in the first and second camera system. Notice that the last column in all three of the focal plane views is missing. This is due to the control points going out of the second camera's FOV.

The *3-D* view of the scene for row counts of 1, 10, and 20 and a column count of 6 is shown in figures 5–47 through 5–49. The control points are marked by plus symbols. The first camera position is marked by a square symbol. The second camera position is marked by a triangle. Note

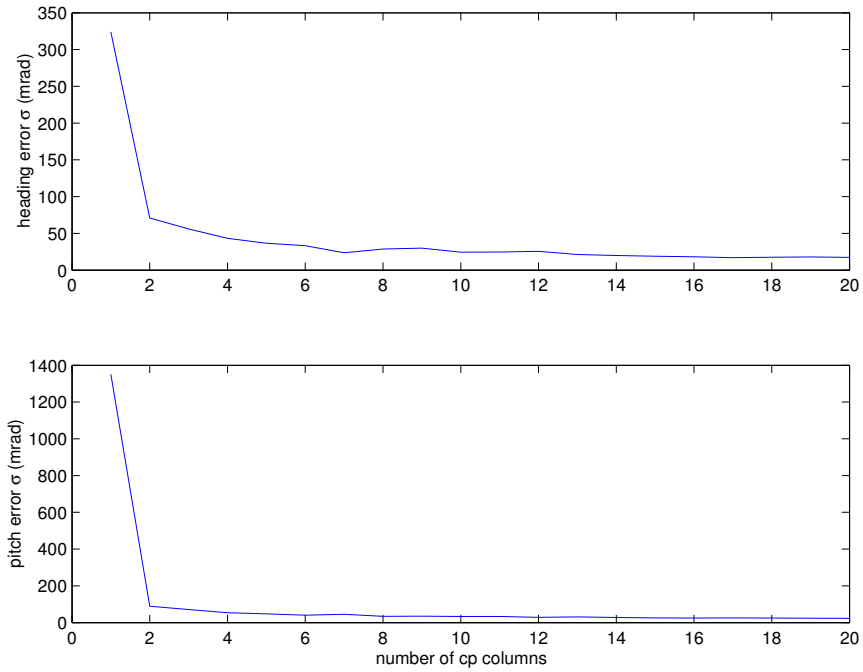

Figure 5–42: Varying number of control points study results for column counts 1-20 and a row count of 6 - ned to c2

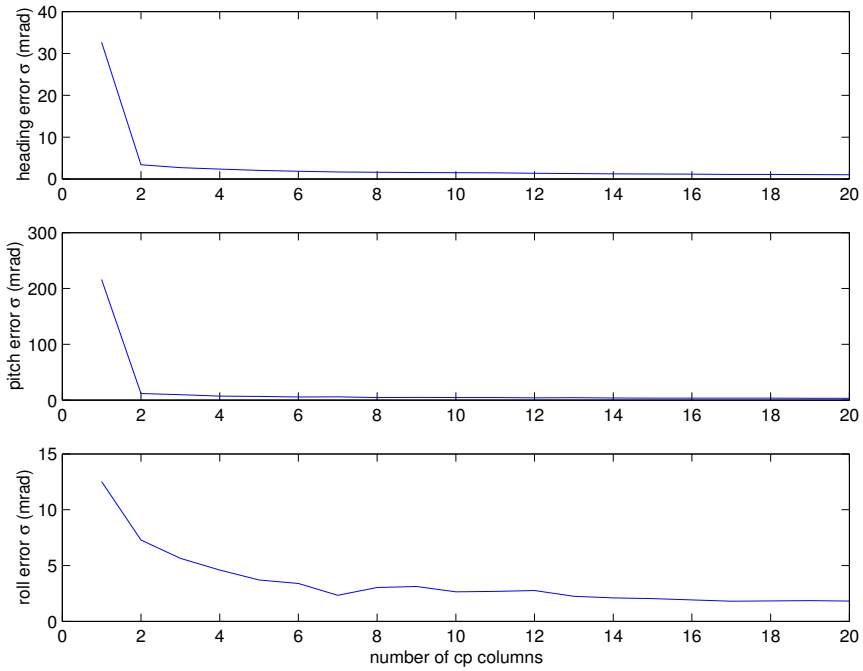

Figure 5–43: Varying number of control points study results for column counts 1-20 and a row count of 6 - c1 to c2

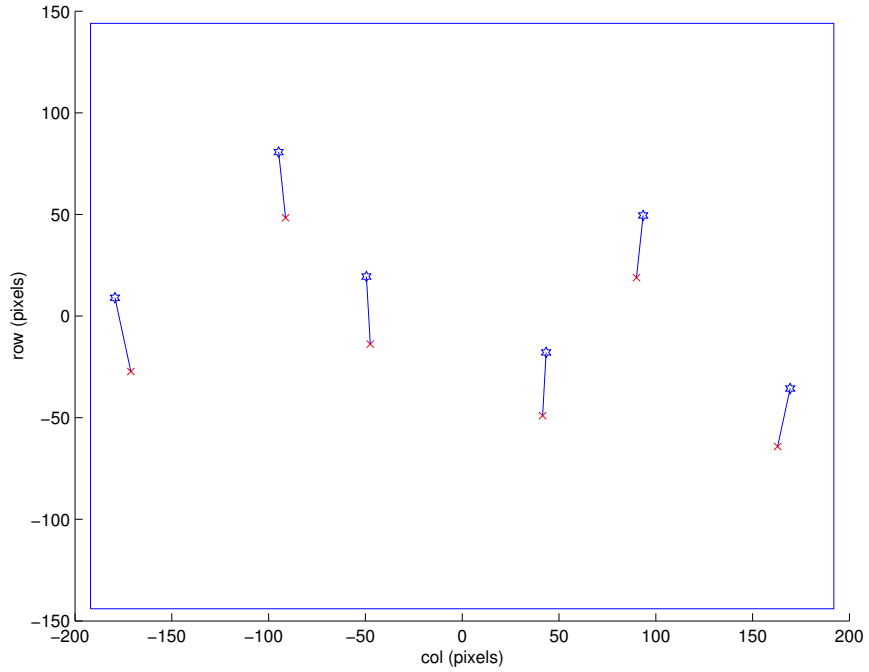

Figure 5–44: Control point locations on focal plane for row count of 1 and a column count of 6.

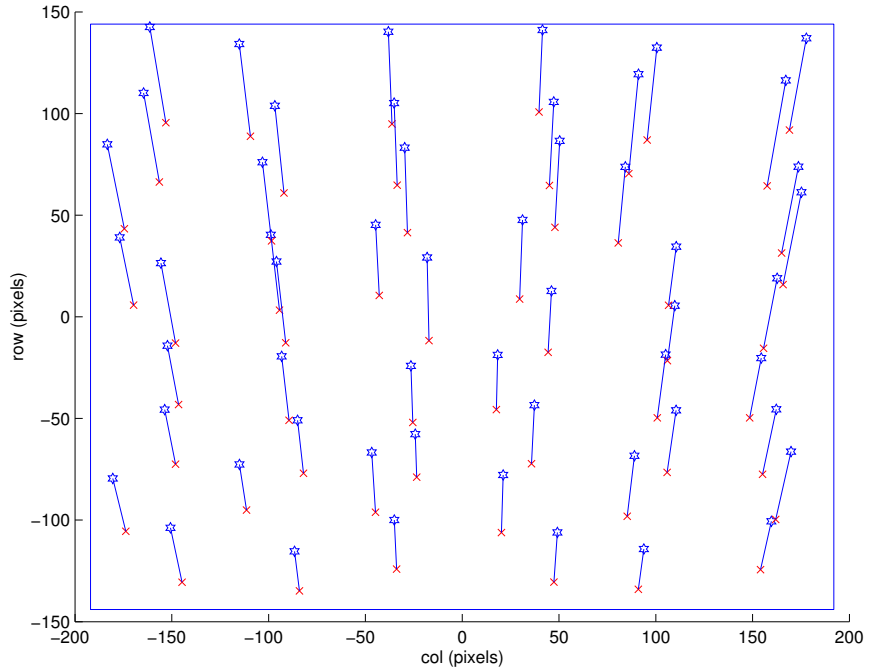

Figure 5–45: Control point locations on focal plane for row count of 10 and a column count of 6.

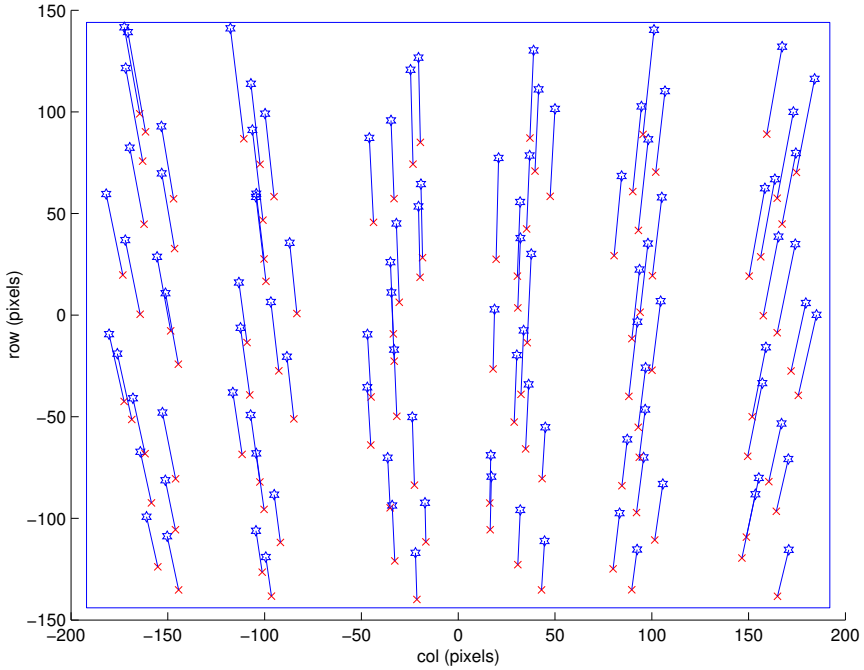

Figure 5–46: Control point locations on focal plane for row count of 20 and a column count of 6.

that there are only 5 points in figure 5–44 due to the last control point going out of the second camera's FOV.

The results for the computed attitude standard deviations are shown in Figures 5–50 and 5–51.

5.2.5 Varying Location of Last Control Point

The effect of varying the location of the last control point on the attitude estimates is shown in this section. The input parameters used for this study are shown in Table 5–11. The GLEMA driver file `varying_last_CP_location.m` was used for this part of the study.

The original 3×3 control point configuration is shown in Figure 5–52 with the control point that was varied being designated by a hexagram. The varying control points original location corresponds to the upper right grid location on the focal planes of Figures 5–53 and 5–54; its location was varied from the lower left hand corner to the upper right hand corner of the focal plane for those two figures. The eight stationary control points are designated by a diamond in Figures 5–53 and 5–54. For the 384×288 pixel array camera, that corresponds to 111,592 iterations. The inertial heading estimate error for each location is displayed in Figure 5–53.

The terrain used for the attitude variance calculation was formed using the

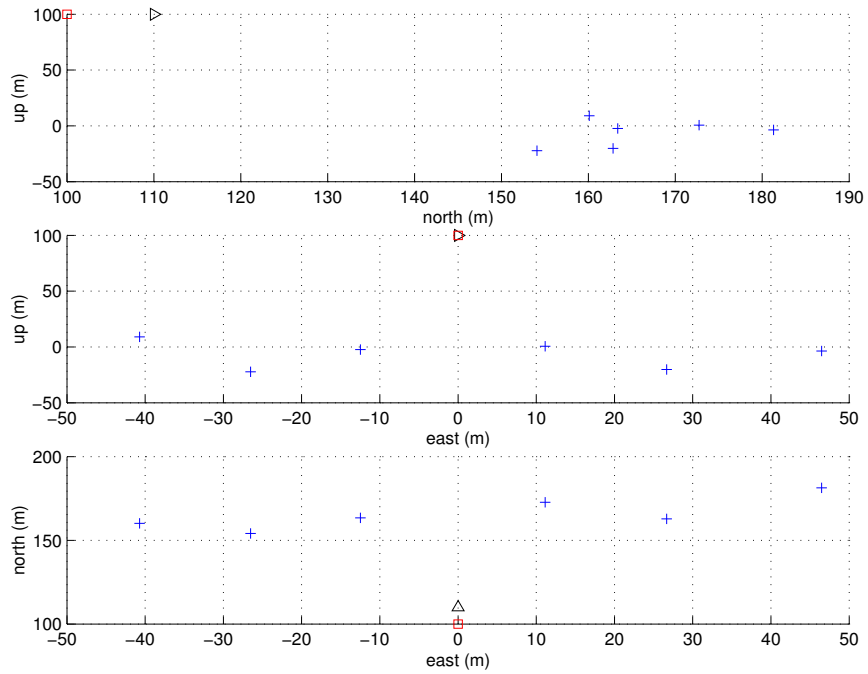

Figure 5–47: Three-dimensional view of the scene for a row count of 1 and a column count of 6 for the varying number of control points study.

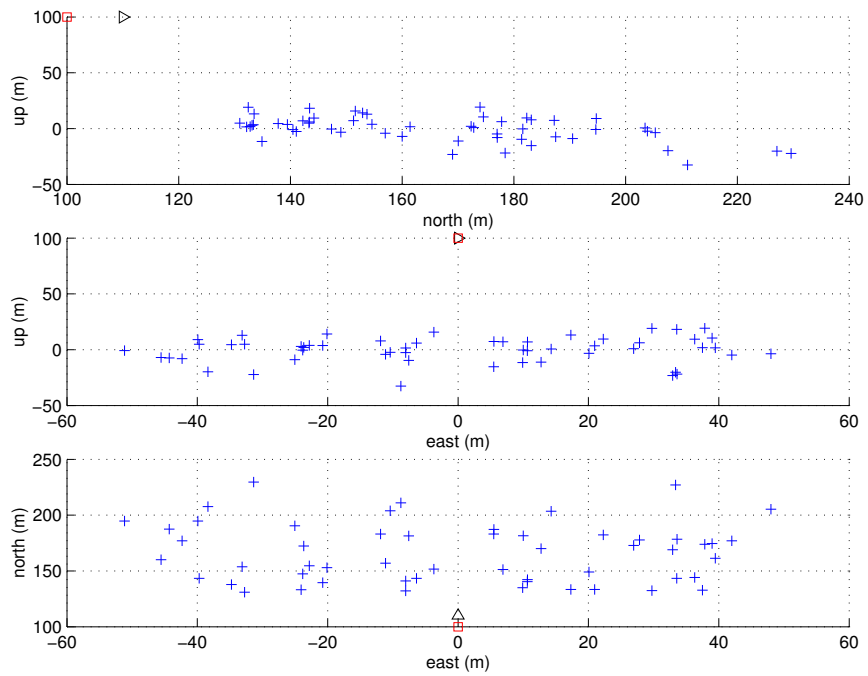

Figure 5–48: Three-dimensional view of the scene for a row counts of 10 and a column count of 6 for the varying number of control points study.

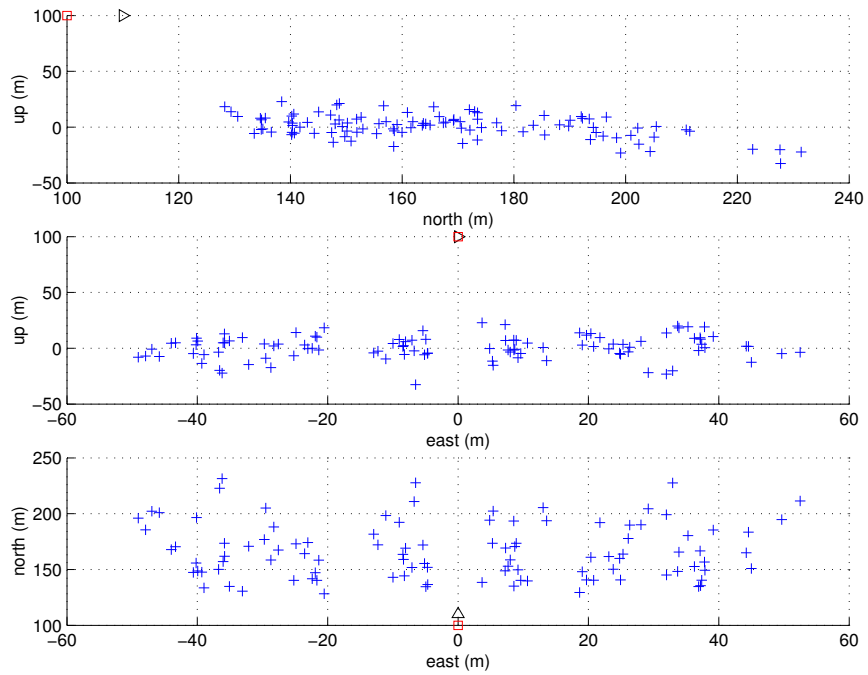

Figure 5–49: Three-dimensional view of the scene for a row count of 20 and a column count of 6 for the varying number of control points study.

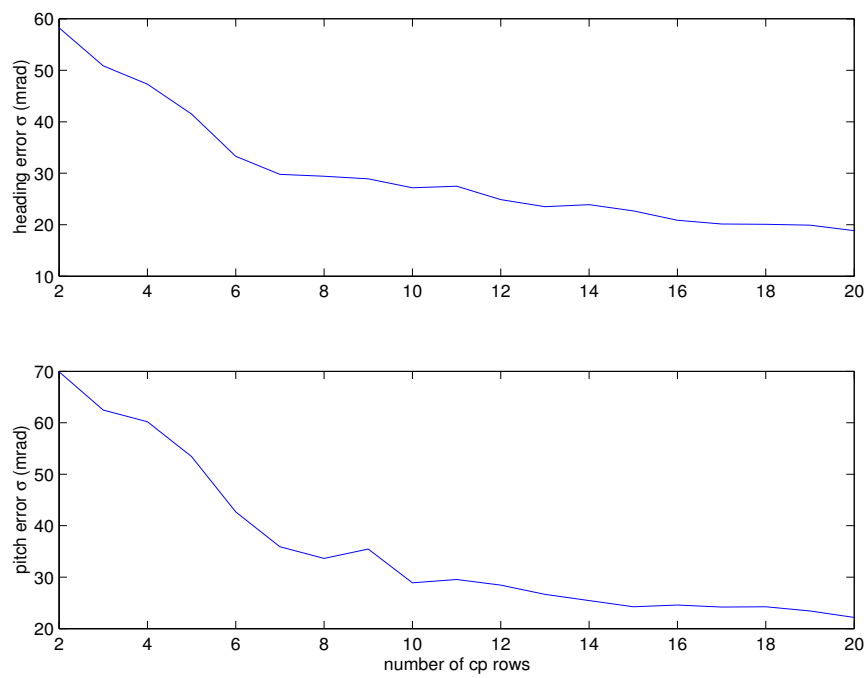

Figure 5–50: Varying number of control points study results for row counts 2-20 and a column count of 6 - ned to c2

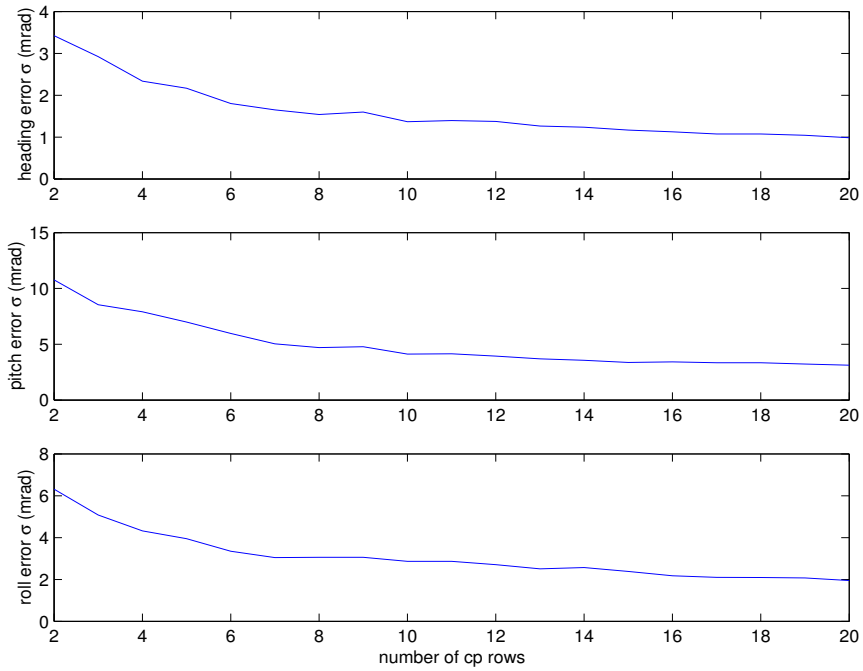

Figure 5–51: Varying number of control points study results for row counts 2-20 and a column count of 6 - c1 to c2

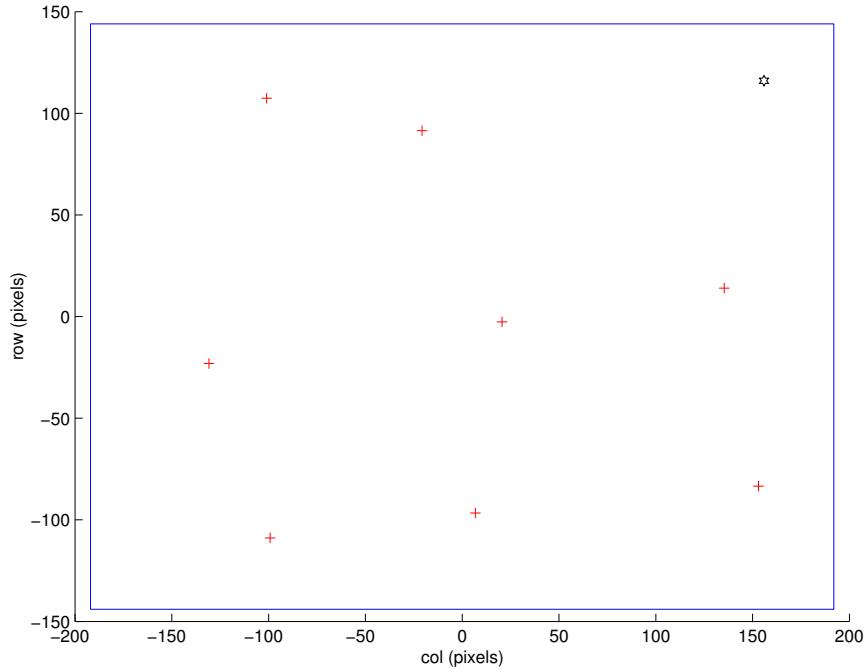

Figure 5–52: Original focal plane view of control point locations for the varying last control point study.

Table 5–11: Input parameters for varying location of last control point.

Parameter	Value	Description
β	$\frac{\pi}{3}$	Camera depression angle in radians.
f	0.005	Camera focal length in meters.
ccd_pixels	384 288	Number of column and row pixels in the ccd array camera.
ccd_size	0.0042 0.0032	Size of the ccd array camera in meters.
cp cols	3	Number of control point columns on the ccd array camera.
cp rows	3	Number of control point rows on the ccd array camera.
$(\phi_{c1}^{c2}, \theta_{c1}^{c2}, \psi_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw of the relative attitude between *c1* and *c2*.
$(\phi_{ned}^{c2}, \theta_{ned}^{c2}, \psi_{ned}^{c2})$	(0,0,0)	Roll, pitch, yaw of the inertial attitude of the second camera system *c2*.
$(\tilde{\phi}_{c1}^{c2}, \tilde{\theta}_{c1}^{c2}, \tilde{\psi}_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors in the relative attitude between *c1* and *c2*.
$(\tilde{\phi}_{c}^{c2}, \tilde{\theta}_{c}^{c2}, \tilde{\psi}_{c}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors between the *c* (center) coordinate system and second camera system *c2*.
$\|\mathbf{a}\|$	10	Displacement vector magnitude in meters.
$\mathbf{v}^{enu}$	(0,1,0)	Direction of the displacement vector, or velocity vector.
pixel error	1	Measurement error of the control points in pixels.

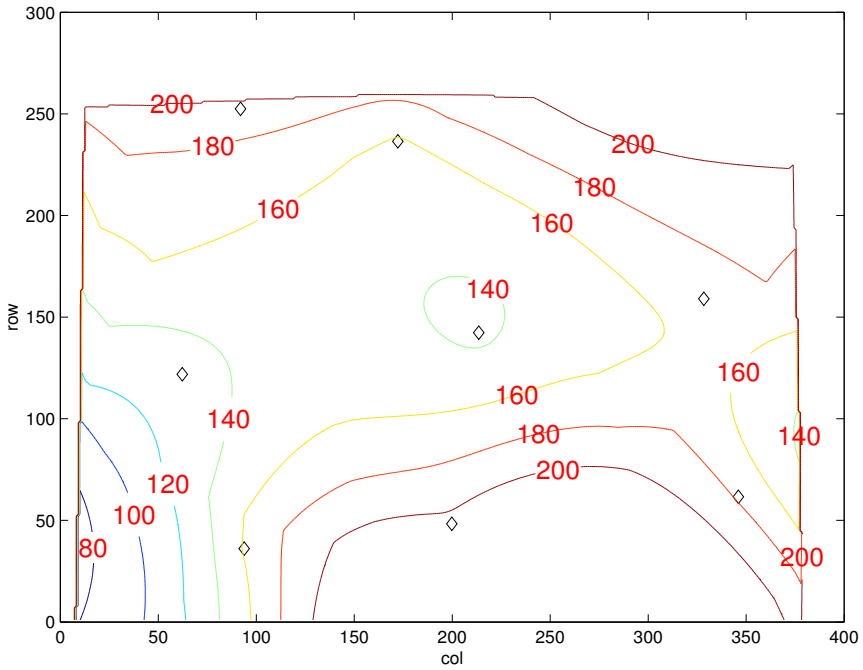

Figure 5–53: Inertial heading estimate errors for varying last control point study.

`computeLastCPRange_better.m` m-file from the GLEMA program. It used all the ranges of the designated control points within a 200-pixel threshold to interpolate the range at each pixel location on the focal plane. The corresponding terrain from each calculation is shown mapped onto the focal plane in Figure 5–54.

5.2.6 Varying Control Point Measurement Error

The effect of varying the control point's measurement error on the attitude estimates is shown in this section. The input parameters used for this study are shown in Table 5–12. The GLEMA driver file used in this study was `varying_pixel_error.m`.

The measurement error of the control points was varied from 1.0 to 2.99 pixels. The first iteration corresponds to measurement error of 1.0 pixels and the last iteration corresponds to a measurement error of 2.99 pixels.

The control point locations on the focal planes of both cameras was not varied for this study and are shown in Figure 5–55. The "x" symbols indicate the designated control point location on the first camera's focal plane, and the hexagon symbols indicate the calculated control point locations on the second camera's focal plane. A line is drawn to connect the corresponding control point locations in the first and second camera system.

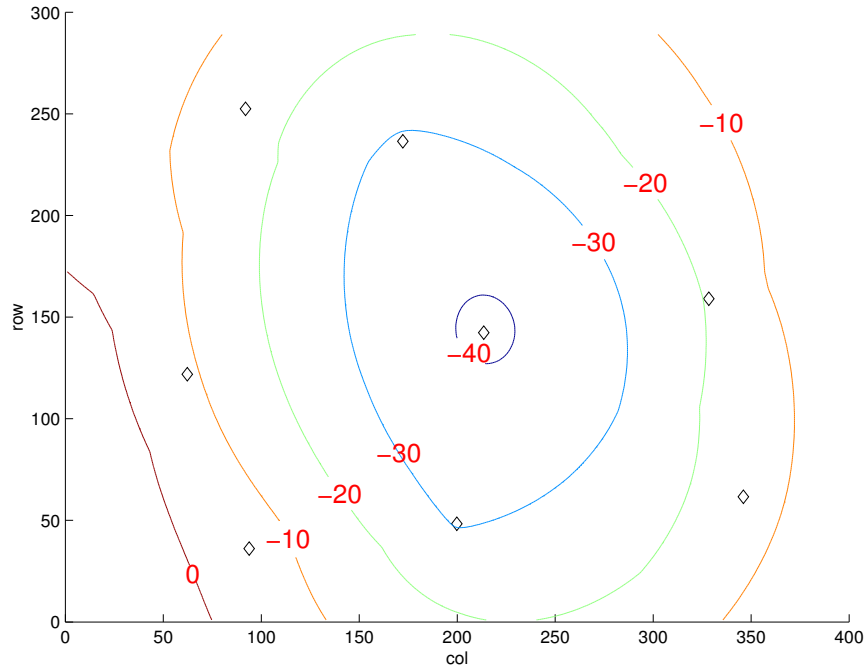

Figure 5–54: The calculated terrain mapped onto the focal plane for varying last control point study.

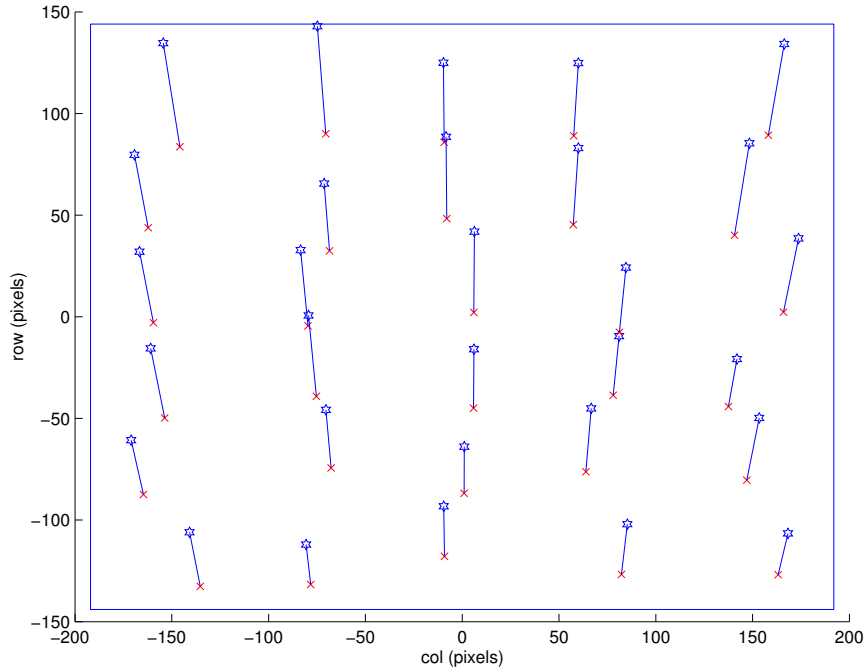

Figure 5–55: Control point locations on focal plane for the varying pixel error study.

Table 5–12: Input parameters for varying control point measurement error.

Parameter	Value	Description
β	$\frac{\pi}{3}$	Camera depression angle in radians.
f	0.005	Camera focal length in meters.
ccd_pixels	384 288	Number of column and row pixels in the ccd array camera.
ccd_size	0.0042 0.0032	Size of the ccd array camera in meters.
cp cols	5	Number of control point columns on the ccd array camera.
cp rows	7	Number of control point rows on the ccd array camera.
$(\phi_{c1}^{c2}, \theta_{c1}^{c2}, \psi_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw of the relative attitude between *c1* and *c2*.
$(\phi_{ned}^{c2}, \theta_{ned}^{c2}, \psi_{ned}^{c2})$	(0,0,0)	Roll, pitch, yaw of the inertial attitude of the second camera system *c2*.
$(\tilde{\phi}_{c1}^{c2}, \tilde{\theta}_{c1}^{c2}, \tilde{\psi}_{c1}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors in the relative attitude between *c1* and *c2*.
$(\tilde{\phi}_{c}^{c2}, \tilde{\theta}_{c}^{c2}, \tilde{\psi}_{c}^{c2})$	(0,0,0)	Roll, pitch, yaw misalignment errors between the *c* (center) coordinate system and second camera system *c2*.
$\|\mathbf{a}\|$	10	Displacement vector magnitude in meters.
$\mathbf{v}^{enu}$	(0,1,0)	Direction of the displacement vector, or velocity vector.
pixel error	**varied**	Measurement error of the control points in pixels.

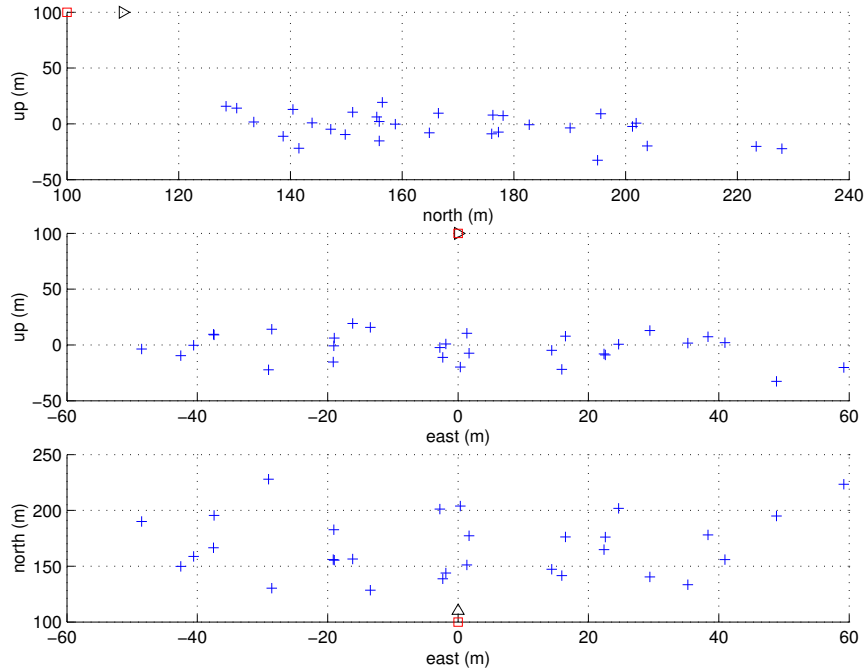

Figure 5–56: Three-dimensional view of the scene for the varying control point measurement error study.

The *3-D* view of the scene did not change for this study as well. It is shown in Figure 5–56. The control points are marked by plus symbols. The first camera position is marked by a square symbol. The second camera position is marked by a triangle.

The results for the computed attitude standard deviations are shown in Figures 5–57 and 5–58. The attitude errors are shown to be directly proportional to the measurement errors as expected.

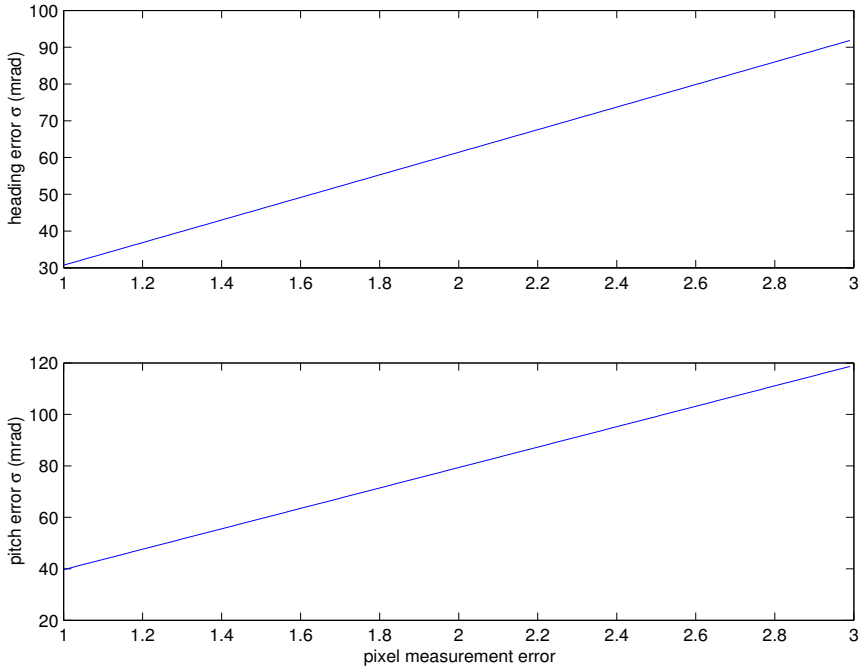

Figure 5–57: Varying control point measurement errors results for 1.0 to 2.99 pixels - ned to c2

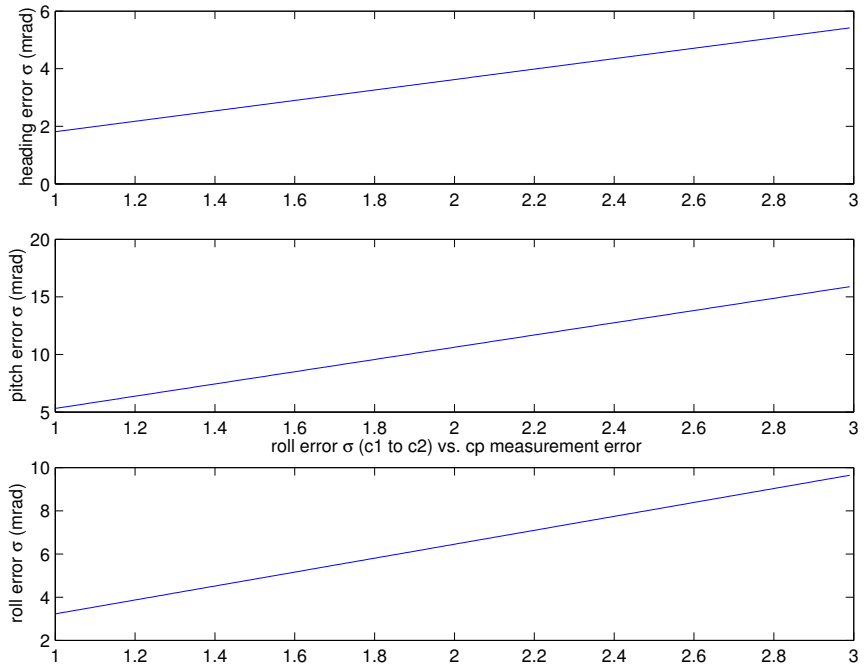

Figure 5–58: Varying control point measurement errors results for 1.0 to 2.99 pixels - c1 to c2

CHAPTER 6
CONCLUSIONS

It has been shown in this thesis that a "free" attitude estimate is available for (1) a vehicle that contains a downward-looking vision system and a GPS receiver capable of differential carrier-phase positioning, and (2) two or more vehicles containing an overlapping FOV and GPS receivers capable of differential carrier-phase positioning. *SfM* techniques on two images with point correspondences produce a displacement vector between the camera locations in the body coordinate system. GPS differential carrier-phase positioning produces the displacement vector between the two cameras in an inertial reference frame. Once two vectors are found that are expressed in both coordinate systems, the rotation between the two coordinate systems can be found. This corresponds to the vehicle attitude.

This thesis began with a literature review of the *SfM* field and detailed how to set up the reconstruction problem using Euclidean geometry techniques. Background in GPS technology showed the accuracy and ambiguity differences between positioning with code measurements and positioning with carrier-phase measurements. Also, it was shown that when using carrier-phase measurements for positioning it is less ambiguous and more accurate to compute differential positions between two receivers than the absolute position of each receiver position. Two solution approaches for finding an attitude estimate were presented: (1) an SVD approach and (2) a descent approach. Finally, a linearized model of the system was developed in which a detailed error analysis was performed. The key contribution that this thesis presents is the detailed error analysis of the attitude estimate.

The error analysis showed how the accuracy of the attitude estimate depends on the geometry of the scene, camera parameters, displacement between camera locations, number and location of control points specified on the focal plane, and measurement error associated with specifying the control points. The accuracy of the attitude estimate was shown to be on the order of tens of mrad.

6.1 Optimal Camera Parameters for Heading Estimation

Table 6–1 shows the optimal values for computing the inertial heading estimate based on the error analysis parameter study of chapter 5. Based on these findings the following values

were chosen for the optimum heading estimate: 20° FOV, 20° camera depression angle, ten meter camera displacement vector magnitude, six columns by ten rows of control points, a measurement error of one pixel, and a 384×288 ccd array camera[1] with a 0.0042×0.0032 meter focal plane. Inputting this configuration into the GLEMA Matlab© program resulted in a heading estimate accuracy of 8.6836 mrad or approximately half a degree. The driver file used to generate this results was `optimal_heading_estimate.m`.

Table 6–1: Optimal parameter values for heading estimate.

Optimal Parameter Value	Description/Notes
FOV $\leq 20^o$ or FOV $\geq 70^o$	Camera FOV in degrees (note that for the pitch attitude estimates larger FOV values are desired).
$\beta \leq 20^o$ or $\beta \geq 160^o$	Camera depression angle in degrees.
$\|\mathbf{a}\| \geq 10$	Displacement vector magnitude in meters.
cp cols ≥ 6	Number of control point columns on the ccd array camera.
cp rows ≥ 10	Number of control point rows on the ccd array camera.
pixel error ≤ 2	Measurement error of the control points in pixels.

6.2 Worst Camera Parameter Values for Attitude Estimation

The error analysis presented in chapter 5 also showed that there are worst case values for the parameters in regards to formulating the "free" attitude estimate. A focal length of 45° produced the largest error in the heading estimate. A camera depression angle of 90° (pointing straight down) produced the largest error in the heading estimate, and a value of approximately 65° produced the largest pitch error value. A displacement vector magnitude of less than 5 meters produced errors in both pitch and heading estimates greater than 100 mrad and increasing exponentially. Finally, less than 12 control points also produced significant errors in both the heading and pitch estimates. Inputting the configuration for the worst heading estimate into the GLEMA Matlab© program resulted in an accuracy of 405.8933 mrad or approximately 23°. The driver file used to generate this result was `worst_heading_estimate.m`.

[1] The ccd array size and number of pixels was not part of the parameter study. If a finer resolution on the focal plane can be achieved (i.e., increased number of pixels per focal plane area), then that would obviously decrease the magnitude of the heading estimation error.

6.3 Future Areas to Explore

There are several areas throughout this thesis that were noted as future areas to explore. Those areas, as well as some others not previously mentioned are listed below.

- Implementation and characterization (i.e., via Monte Carlo analysis) of an algorithm for the SVD solution.
- Implementation and characterization of algorithms for the first and second order coplanarity constraint based descent algorithms.
- Development, implementation, and characterization of a descent algorithm using the generalized linearized error model presented in chapter 5.
- Incorporation of range information into the error analysis from laser radar or synthetic aperture radar (SAR) visioning systems.
- Experimental verification of error analysis results.
- Expansion of application to tactical far target identification devices.
- Derive a fundamental equation that expresses the accuracy of the attitude estimate as a function of the number of control points (i.e., is there a theoretical limit at which adding more control points has no positive effect on the accuracy of the attitude estimate).

APPENDIX
CENTER (c) COORDINATE FRAME DISCUSSION

We need a transformation that maps $\mathbf{a}^{LL}$ to the x-axis of the c coordinate frame, and keeps the z-axis of the c coordinate frame as close to the z-axis of the LL coordinate frame as possible. Assume this form:

$$\begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \tag{A-1}$$

where $a = \|\mathbf{a}\|$.

The row vectors $\mathbf{r}_1^T$, $\mathbf{r}_2^T$, and $\mathbf{r}_3^T$ are the unit vectors of the axes of the c coordinate frame expressed in the LL coordinate frame. It immediately follows that

$$\mathbf{r}_1^T = \frac{\mathbf{a}^T}{\|\mathbf{a}\|}. \tag{A-2}$$

The requirement that the z-axis of the c coordinate frame be as close as possible to the z-axis of the LL coordinate frame is satisfied if the y-axis of the c coordinate frame lies in the xy-plane of the LL coordinate frame. It immediately follows that $r_{23} = 0$.

For $\mathbf{r}_2$, we need a vector in the xy-plane of the LL coordinate frame that is orthogonal to $\mathbf{a}$. There are two possibilities: $[-a_y \quad a_x \quad 0]$ and $[a_y \quad -a_x \quad 0]$. The first is the correct choice, since $\mathbf{a}^{LL} = [a \quad 0 \quad 0]^T$ should lead to $R_{LL}^c = I$. Therefore,

$$\mathbf{r}_2^T = \frac{1}{\sqrt{a_x^2 + a_y^2}} \begin{bmatrix} -a_y & a_x & 0 \end{bmatrix}. \tag{A-3}$$

Finally, $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$, so

$$\mathbf{r}_3^T = \frac{1}{\|\mathbf{a}\|\sqrt{a_x^2 + a_y^2}} \begin{bmatrix} -a_x a_z & -a_y a_z & a_x^2 + a_y^2 \end{bmatrix}. \tag{A-4}$$

Putting it all together:

$$R_{LL}^c = \frac{1}{\|\mathbf{a}\|\sqrt{a_x^2 + a_y^2}} \begin{bmatrix} a_x\sqrt{a_x^2 + a_y^2} & a_y\sqrt{a_x^2 + a_y^2} & a_z\sqrt{a_x^2 + a_y^2} \\ -a_y\|\mathbf{a}\| & a_x\|\mathbf{a}\| & 0 \\ -a_x a_z & -a_y a_z & a_x^2 + a_y^2 \end{bmatrix} \tag{A-5}$$

REFERENCES

[1] S. Maybank, *Theory of Reconstruction from Image Motion*, Springer–Verlag, Berlin, Germany, 1993.

[2] R. Sturm, "Das Problem per Projektivität undseine Anwendung auf die Flächen zweiten Grades," *Math Annalen*, vol. 1, pp. 533–573, 1869.

[3] O. Hesse, "Die cubische Gleichung von welcher die Lösung des Problems der Homographie von M. Chasles abhängt," *J. Reine Angre Math*, vol. 62, pp. 188–192, 1863.

[4] Q.-T. Luong and T. Vieville, "Canonical Representations for the Geometries of Multiple Projective Views," *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 193–229, 1996.

[5] S. Soatto and P. Perona, "Reducing Structure from Motion: A General Framework for Dynamic Vision Part 1: Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 933–942, Sept. 1998.

[6] T. Huang and A. Netravali, "Motion and Structure from Feature Correspondences: A Review," in *Proc. of the IEEE*, Feb. 1994, vol. 82, pp. 252–268.

[7] S. Soatto and P. Perona, "Reducing Structure from Motion: A General Framework for Dynamic Vision Part 2: Implementation and Experimental Assesment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 943–960, Sept. 1998.

[8] H. Longuet-Higgens, "A Computer Algorithm for Reconstructing a Scene From Two Projections," *Nature*, vol. 293, pp. 133–135, 1981.

[9] D. Heeger and A. Jepson, "Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation," *Int'l J. Computer Vision*, vol. 7, no. 2, 1992.

[10] P. Misra and P. Enge, *GLOBAL POSITIONING SYSTEM Signals, Measurements, and Performance*, Ganga–Jamuna Press, Lincoln, Massachusetts, 2001.

[11] C.C. Counselman III, I.I. Shapiro, R.L. Greenspan, and D.B. Cox, Jr., "Backpack VLBI Terminal with Subcentimeter Capability," in *Proc. Radio Interferometric Techniques for Geodesy*. 1979, vol. 2115, pp. 409–413, NASA Converence Publications.

[12] C.C. Counselman III and S. Gourevitch, "Miniature interferometer terminals for earth surveying: Ambiguity and multipath with global positioning system," *IEEE Transactions on Geosciences and Remote Sensing*, vol. GE–19, no. 4, pp. 244–252, 1981.

[13] R. Brown, "Instantaneous GPS attitude determination," in *Proc. of IEEE Position, Location, and Navigation Symposium (PLANS '92)*, Monterey, California, Mar. 1992, pp. 113–120.

[14] G. Golub and C. V. Loan, *Matrix Computations*, North Oxford Publishing Co. Ltd., Oxford, 1983.

[15] B.K.P. Horn, "Relative Orientation," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 59–78, 1990.

[16] J. Fraleigh, *A First Course in Abstract Algebra*, Addison Wesley, Reading, Massachusetts, 1973.

[17] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA, Reston, VA, 2000.

BIOGRAPHICAL SKETCH

Mr. Rosengren was born the son of an Air Force officer in Camp Springs, MD in 1975. He graduated from A.C. Mosley High School in Panama City, FL, in 1994. After spending the next 3 years studying space physics at the United States Air Force Academy, Mr. Rosengren finished his undergraduate studies at the University of Florida. He graduated in the Summer of 1998, earning his B.S. in physics with high honors. Upon graduation, Mr. Rosengren spent 2 years as a member of the 28th Communications Squadron at Ellsworth AFB, working as a 3C0x1 (Computer Operator). In the Fall of 2000, Mr. Rosengren began working for Dynetics, Inc. in Shalimar, FL. He works as a Systems Analyst, supporting tasks such as the development of the Multiple Ordinance Air Burst (MOAB) GPS-guided munition, Exploitation of *3-D* Data (E3D) Defense Advanced Research Projects Agency (DARPA) program, Post Mission Miss Distance Scoring System (PMMDSS) for the Open Air Range (OAR), Target Acquisition and Tracking System (TATS) Upgrade also for the Open Air Range (OAR), and modeling and simulation of a tactical unmanned air vehicle (TUAV) and an organic air vehicle (OAV) for use in the DARPA Jigsaw sensor development program.

While in Gainesville finishing his undergraduate studies, Mr. Rosengren met the love of his life, the former Dawn Tammy Fries of Floral City, FL. Somehow, Mr. Rosengren managed to convince the lovely Dawn to marry him. They were married in the Summer of 1999 and following an all too short honeymoon in the Bahamas, Mrs. Rosengren moved up to South Dakota to join her husband. A precious little girl, Miss Gabriella Nicole, was given to Mr. and Mrs. Rosengren on Feb 4, 2002. They also have a *scruffy* Maltese named Daisy Mae and a floppy-eared bunny named Flopsy.

Mr. Rosengren enjoys playing tennis, volleyball, and golf in his spare time. He also enjoys playing jazz and classical trumpet, and serves as a member of the Rocky Bayou Baptist Church Orchestra.

Mr. Rosengren began his graduate studies at the University of Florida Graduate Engineering and Research Center in the Spring of 2001. He is scheduled to complete his Master of Science degree in electrical engineering in the Spring of 2004. Much celebration from the Rosengren family will commence at that time.