AIAA Guidance, Navigation and Control Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-7441

Methods for Localization and Mapping Using Vision and Inertial Sensors

Allen D. Wu* and Eric N. Johnson †

Georgia Institute of Technology, Atlanta GA, 30332-0150

The problems of vision-based localization and mapping are currently highly active areas of research for aerial systems. With a wealth of information available in each image, vision sensors allow vehicles to gather data about their surrounding environment in addition to inferring own-ship information. However, algorithms for processing camera images are often cumbersome for the limited computational power available onboard many unmanned aerial systems. This paper therefore investigates a method for incorporating an inertial measurement unit together with a monocular vision sensor to aid in the extraction of information from camera images, and hence reduce the computational burden for this class of platforms. Feature points are detected in each image using a Harris corner detector, and these feature measurements are statistically corresponded across each captured image using knowledge of the vehicle's pose. The investigated methods employ an Extended Kalman Filter framework for estimation. Real-time hardware results are presented using a baseline configuration in which a manufactured target is used for generating salient feature points, and vehicle pose information is provided by a high precision motion capture system for comparison purposes.

I. Introduction

Traditionally, the task of determining position and attitude for an aircraft has been handled by the combination of an inertial measurement unit (IMU) with a global positioning system (GPS) receiver. In this configuration, accelerations and angular rates from the IMU can be integrated forward in time, and position updates from the GPS can be used to bound the resulting errors that result from this integration. This solution to the localization problem makes aircraft prone to certain modes of failure due to their reliance on the reception of external signals from the GPS satellite network. GPS signals can suffer from obstructions or multipath in cluttered environments, and the reception of these signals can furthermore be jammed or otherwise denied. Similarly, the task of mapping the surrounding environment is commonly approached by using ranging sensors to scan areas of interest. However, these sensors typically rely on the emission and reception of a signal to determine range which is sometimes undesirable if the vehicle needs to remain undetected.

Vision sensors have demonstrated immense potential for application to localization and mapping since they provide data about the surrounding environment, and simultaneously allow for the possibility of inferring own-ship information from these images. However, the majority of results presented in these areas have been applied to ground robots where size and payload considerations are often not a limitation. This means that most of the algorithms currently available for extracting information from the 2D images of an image sensor are often too computationally intensive to be handled by the limited processing power onboard many unmanned aerial systems (UAS's) such as the one shown in Figure 1. Over recent years, it has been proposed that adding an IMU to a vision system could help to alleviate the computational burden of such algorithms because the inclusion of inertial sensors allows for the prediction of camera motion from frame to frame and also helps with resolving scale ambiguity. A navigation and mapping system that uses only a combination of inertial and vision sensors would also be a fully self-contained one that would not be prone to jamming or detection.

Previous work in the application of inertial sensors to the problems of vision-based localization and mapping have included Kim and Sukkarieh who have successfully demonstrated simultaneous localization and mapping (SLAM) for a UAS using these sensors.[1] In their setup, targets of known size were placed on the ground, and the camera provided measurements of the bearing, elevation, and range to each target. The range information was computed from the size of the targets in each image. An Extended Kalman Filter (EKF) was used to estimate the vehicle's ego-motion as well

*Graduate Research Assistant, School of Aerospace Engineering, AIAA Student Member.

†Lockheed Martin Associate Professor of Avionics Integration, School of Aerospace Engineering, AIAA Member.

Figure 1. Small unmanned aerial systems, such as the Hornet UAS with a weight of 2.4 lbs and a 2.3 ft rotor diameter, are often not capable of carrying large computer systems.

as the locations of the targets. Langelaan also presented results where he used an IMU and a monocular camera for performing SLAM through an environment populated with obstacles.[2] Langelaan implemented an Unscented Kalman Filter (UKF) to estimate the states of a small vehicle along with the positions of obstacles located in a ground plane. Hardware results were presented using a ground robot, and simulation results were presented for a UAS in a 3D environment. Another approach to the SLAM problem was also presented by Mourikis and Roumeliotis.[3] In their approach, they estimated the motion of a ground vehicle through an environment using an EKF, but would perform the measurement corrections only after a feature has left the field of view or after a certain number of measurements of a feature have been obtained. They also proposed a residual that was independent of errors in the feature point location to within a first order accuracy. Lowe's Scale Invariant Feature Transform (SIFT) was used for feature point extraction and tracking. Additional efforts in this area also include those by Koch *et al.* who used a feature point tracker looking at the ground with an IMU for state estimation of a helicopter. They assumed a known altitude above the ground, and from the knowledge of the the helicopter's altitude, they could compute the inertial position of the feature point from the pixel position in the camera. An EKF framework was used as the estimation scheme with an image sensor, an IMU, a magnetometer, and a sonar as the sensors. A Lucas-Kanade feature point tracker was implemented for tracking the points on the ground. Software simulations and a flight test with the helicopter demonstrated the accuracy of the navigation system without GPS. Many others have also looked into the application vision-based localization and mapping for UAS's.[5−10]

The focus of this paper will be on methods that use an IMU and a monocular camera to track feature points. A single camera is used because systems with multiple cameras are more complex due to the spatial and temporal calibrations required. Furthermore, for many small UAS's, it is difficult to separate the cameras far enough to form a substantial baseline between the cameras. Feature points are used as targets because simple image processing methods exist for extracting these from images, and furthermore, no prior knowledge of the environment is required to obtain measurements of them. The estimation problem using a monocular camera to track feature points is a difficult problem because a point feature in a camera image only provides bearing and elevation angle measurements, and the range information is lost in the projection from the 3D world onto the 2D image. Further complicating this task is the fact that it is difficult to correspond features from one image to the next. This paper will look at an implementation based on the method proposed by Langelaan.[2] This method is implemented in a hardware setup, and is compared to a motion capture system as a baseline measurement to analyze the effectiveness of the method in a hardware configuration. This paper will first outline some background information regarding the estimation method used as well as an overview of the image processing. Then a description of the experimental hardware setup is provided, followed by an analysis of the results.

II. Background Information

This section provides some background information for the problem at hand. First a description of how the camera is modeled as a pinhole camera is provided. This is followed by a brief overview of the fundamental equations used in the Extended Kalman Filter. Some modifications are needed for the Extended Kalman Filter in this problem though to handle the uncertain number of measurements as well as the problem of corresponding measurements obtained to their actual point in inertial space. These issues are also discussed in the overview of the Extended Kalman Filter. Finally, a brief discussion of image processing techniques is provided to address how the feature points are actually extracted from a given image frame.

A. Relating 3D Position to 2D Images

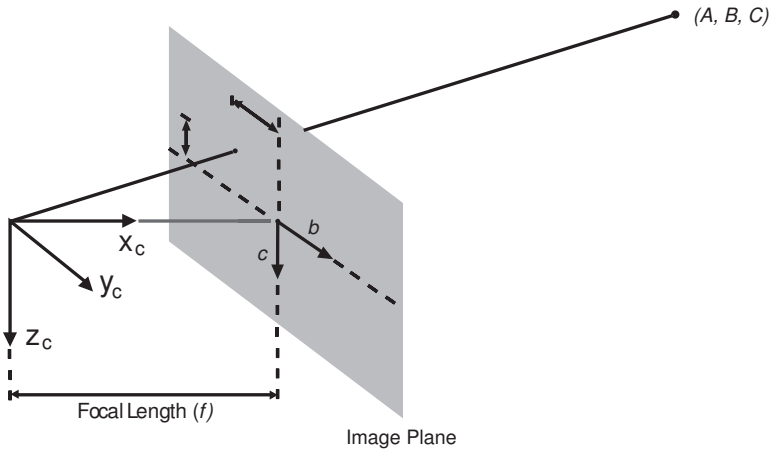

Figure 2. Camera perspective projection model used for relating 3D position to position in 2D images. The point (A, B, C) is projected onto the camera image plane to the point (b, c).

A perspective projection model of a pinhole camera allows position in a 2D camera image to be inferred from 3D position as shown in Figure 2. The model projects an arbitrary point (A, B, C) to a pixel point (b, c) on the image plane (the camera image) according to the following relations:

$$b = f\frac{B}{A} \tag{1}$$

$$c = f\frac{C}{A}, \tag{2}$$

where f is the focal length of the camera. The focal length can be computed from knowledge of the width of the camera image plane (w) and the angle of the horizontal field of view (γ), both of which are characteristics of the physical camera, according to

$$f = \frac{w}{2\tan\left(\frac{\gamma}{2}\right)} \tag{3}$$

B. Reference Frames

Three primary frames of reference are needed for this estimation problem. The inertial reference frame is a local inertial frame with its axes aligned in the North, East, and down directions. The camera frame has its origin at the camera's principal point with the x_c axis along the camera's optical axis and the z_c axis pointing downwards. The body frame is fixed to the vehicle center of mass with the x_b axis directed out the nose of the aircraft and the z_b axis pointing downwards. Vector components in the different reference frames can be transformed using direction cosine

matrix sequences as follows:

$$\mathbf{L}_{cb} = \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_c & 0 & \cos\theta_c \end{bmatrix} \begin{bmatrix} \cos\psi_c & \sin\psi_c & 0 \\ -\sin\psi_c & \cos\psi_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{L}_{bi} = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{5}$$

$$\mathbf{L}_{ci} = \mathbf{L}_{cb}\mathbf{L}_{bi} \; . \tag{6}$$

$\mathbf{L}_{cb}$ is a rotation matrix that converts vectors from components in the body frame to components in the camera frame by using the pan (ψ_c) and tilt (θ_c) angles of the camera. Note, however, that the transformation from the body to the camera frame accounts for only the orientation differences between the two frames. The fact that the camera frame is centered at the camera's location, whereas the body frame is centered at the vehicle center of mass, is neglected. $\mathbf{L}_{bi}$ is a standard rotation matrix from the body to the local inertial frame expressed in quaternions. The inverse rotations are obtained by swapping the matrix subscript indices and taking the transpose of the appropriate matrix.

C. Overview of the Extended Kalman Filter

The EKF formulation used for the vision-based estimation tasks is a mixed continuous-discrete time filter. The EKF algorithm can be broken up into two main phases: prediction and correction. In the prediction phase of the EKF, a nonlinear continuous-time process model is used to propagate the current best state estimate forward in time to come up with a new predicted state estimate. Meanwhile, the correction phase runs at discrete intervals and uses sensors to correct the estimate predicted by the process model. By comparing predicted values of the measurement vector with actual measurements from the image processor, the EKF is able to estimate the desired states.

1. Extended Kalman Filter Prediction

In the prediction phase of the EKF estimation algorithm, the state estimate $\hat{\mathbf{x}}$ and the covariance matrix $\mathbf{P}$ are updated using a nonlinear model of the vehicle dynamics. The following equations are used for these updates:

$$\dot{\hat{\mathbf{x}}} = f\left(\hat{\mathbf{x}}(t), t\right) \tag{7}$$

$$\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{Q} \tag{8}$$

where $f\left(\hat{\mathbf{x}}(t), t\right)$ is a nonlinear process model for the system dynamics, $\mathbf{A} = \left(\partial f / \partial \hat{\mathbf{x}}\right)|_{\hat{\mathbf{x}}}$ is a Jacobian matrix representing a linearization of the dynamics, and $\mathbf{Q}$ is a positive definite matrix representing the process noise inherent in the system.

2. Extended Kalman Filter Correction

The EKF makes use of a measurement model $h(\hat{\mathbf{x}}^-)$ that takes in the current best state estimate and computes an expected measurement vector for that given state. By comparing this expected measurement with the actual measurement from the sensor, corrections for the state estimate and the covariance matrix from the prediction phase of the filter are computed. The equations for these corrections are as follows:

$$\mathbf{K} = \mathbf{P}^-\mathbf{C}^T(\mathbf{C}\mathbf{P}^-\mathbf{C}^T + \mathbf{R})^{-1} \tag{9}$$

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{K}[\mathbf{z} - h(\hat{\mathbf{x}}^-)] \tag{10}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{P}^- \tag{11}$$

where $\mathbf{K}$ is the Kalman gain, $\mathbf{R}$ is a diagonal matrix representing measurement noise in the sensor, and $\mathbf{C} = \left(\partial \mathbf{z} / \partial \hat{\mathbf{x}}\right)|_{\hat{\mathbf{x}}}$ is the Jacobian of the measurement vector with respect to the state vector. Minus superscripts in the above equations denote a priori values obtained from the prediction phase equations. The results from (9) - (11) are used by the prediction phase in the next time step to further propagate the state vector and the covariance matrix, and the procedure is repeated.

The EKF corrections can also be performed using a sequential processing of the measurement update. Given the time updated state $\hat{\mathbf{x}}^-$ and error covariance matrix $\mathbf{P}^-$, and a measurement vector $\mathbf{z}(t_k) = [\mathbf{z}^1(t_k)^T \cdots \mathbf{z}^r(t_k)^T]^T$, it may happen that the different components of the measurement vector may come in at different rates or that they may not all be available at a given time step. Applying a sequential measurement update allows each component of the measurement vector to be applied independently of the others as they become available. For $l = 1, 2, \cdots, r$ (r r different measurements at time t),

$$\mathbf{K}_k^l = \mathbf{P}_k^{l-1}\mathbf{C}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1})\left[\mathbf{C}_k^l(\hat{\mathbf{x}}_k^{l-1})\mathbf{P}_k^{l-1}\mathbf{C}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) + \mathbf{R}_k^l\right] \tag{12}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l\left[\mathbf{z}_k^l - h(\hat{\mathbf{x}}_k^{l-1})\right] \tag{13}$$

$$\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l\mathbf{C}_k^l\right]\mathbf{P}_k^{l-1} \tag{14}$$

where the starting initial conditions for the sequential measurement update at time $t = t_k$ are $\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^-$, $\mathbf{P}_k^0 = \mathbf{P}_k^-$, and the final result of the measurement updates are $\hat{\mathbf{x}}_k^r = \hat{\mathbf{x}}_k$ and $\mathbf{P}_k^r = \mathbf{P}_k$. For every measurement not available at time $t = t_k$, the measurement update for that step l can be skipped. Whenever a measurement is available at any time instant $t = t_k$, that measurement can be included for this sequential update processing.

3. The Correspondence Problem

The correspondence problem of relating target measurements to their states or for correlating measurements from frame to frame can be solved using the statistical z-test. The z-test uses the state error covariance matrix, $\mathbf{P}$, and the measurement noise matrix, $\mathbf{V}$, to define a Z value that ranks the correlation between the measurement and target state estimate. The Z value is defined for the EKF as

$$Z = \mathbf{e}^T\left(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}\right)^{-1}\mathbf{e} \tag{15}$$

where the residual is defined as

$$\mathbf{e} = \mathbf{z} - h(\hat{\mathbf{x}}) \tag{16}$$

so that good matches are indicated by small Z values. Note that the magnitude of Z depends not only on the residual, but also on the covariance matrices, and will be small when $\mathbf{P}$ and $\mathbf{R}$ are large. Therefore, even if the residual is large, great uncertainties in the estimates and the measurements will help to keep the value of Z small. In other words, when the accuracy of the state estimate is poor, the z-test allows for larger residuals because of the high uncertainty. The z-test correlates the measurements and estimates by comparing the magnitude of Z to a critical value. If Z is larger than the critical value, then they do not correspond. Otherwise, the best matching pairs with the lowest Z values are used.

D. Detecting Feature Points

Three of the most common methods for detecting feature points are the Kanade-Lucas-Tomasi (KLT) Tracker,[11] Lowe's Scale-Invariant Feature Transform (SIFT),[12] and the Harris corner detector.[13] The KLT and SIFT formulations handle both the detection and tracking of feature points. However, they are more computationally intensive than the simple Harris corner detector, and for now, the correspondence (or tracking) of points is handled by the statistical z-test described above. The Harris corner detector works by first computing the Harris matrix (denoted $\mathbf{M}$) which is given by

$$\mathbf{M} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \tag{17}$$

In other words, the Harris matrix for a given pixel consists of summations of products of the horizontal and vertical image intensity gradients over a window surrounding the pixel. These summations are taken over 3x3 windows in this work. The following measure is then computed for each pixel

$$M_c = \det(\mathbf{M}) - \kappa\left[\mathrm{Tr}(\mathbf{M})\right]^2 \tag{18}$$

which indicates a feature point if the M_c value is greater than a certain threshold value. This measure looks for strong eigenvalues in more than one direction for corner detection without actually needing to explicitly compute the eigenvalues, thereby reducing the computational requirements of the image processing. Each image is separated into a uniform grid so that feature points are selected uniformly across each image. A minimum separation of distance is also enforced between each selected feature point.

III. Estimating Vehicle States Using Known Feature Points

This section addresses the problem of determining vehicle state using vision-aided inertial navigation when the positions of the feature points are already known in inertial space. This problem tackles part of the SLAM problem in that if GPS is available and feature point positions are being estimated, then in the case of a GPS outage, the combined inertial and vision state estimator could be used for stabilizing the vehicle until GPS returns. Alternatively, this formulation has some usefulness in its own right since the vehicle could have access to a database of positions of recognizable patterns to allow for navigation around a previously marked area. With this a priori knowledge of features, the vehicle could navigate autonomously through the area.

A. Extended Kalman Filter Formulation

The states to be estimated are as follows:

- vehicle position in inertial space: $\mathbf{p}_i = \begin{bmatrix} p_{x_i} & p_{y_i} & p_{z_i} \end{bmatrix}^T$
- vehicle velocity in inertial space: $\mathbf{v}_i = \begin{bmatrix} v_{x_i} & v_{y_i} & v_{z_i} \end{bmatrix}^T$
- vehicle attitude in quaternions: $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$

The measurements we have available from sensor information are:

- body-axis angular rates from the IMU: $\omega_b = \begin{bmatrix} p & q & r \end{bmatrix}^T$
- body-axis specific forces from the IMU: $\mathbf{f}_{sp_b} = (\mathbf{a}_b - \mathbf{g}_b) = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$
- pixel position for feature point n: $\mathbf{z}_n = \begin{bmatrix} X_n & Y_n \end{bmatrix}^T$

Therefore the true state vector for the vehicle is

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \mathbf{p}_i & \mathbf{v}_i & \mathbf{q} \end{bmatrix}^T \\ &= \begin{bmatrix} p_{x_i} & p_{y_i} & p_{z_i} & v_{x_i} & v_{y_i} & v_{z_i} & q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \end{aligned}$$

and the estimated state vector is

$$\begin{aligned} \hat{\mathbf{x}} &= \begin{bmatrix} \hat{\mathbf{p}}_i & \hat{\mathbf{v}}_i & \hat{\mathbf{q}} \end{bmatrix}^T \\ &= \begin{bmatrix} \hat{p}_{x_i} & \hat{p}_{y_i} & \hat{p}_{z_i} & \hat{v}_{x_i} & \hat{v}_{y_i} & \hat{v}_{z_i} & \hat{q}_1 & \hat{q}_2 & \hat{q}_3 & \hat{q}_4 \end{bmatrix}^T \end{aligned}$$

B. Process Model

In this work, the process model comes from the integration of the accelerations and the angular rates as provided by the IMU. The following equations constitute the process model for the state update:

$$\dot{\hat{\mathbf{p}}}_i = \hat{\mathbf{v}}_i \tag{19}$$

$$\begin{aligned} \dot{\hat{\mathbf{v}}}_i &= \hat{\mathbf{L}}_{ib}\mathbf{a}_b && (20) \\ &= \hat{\mathbf{L}}_{ib}\left(\mathbf{f}_{sp_b} + \mathbf{g}_b\right) && (21) \\ &= \hat{\mathbf{L}}_{ib}\mathbf{f}_{sp_b} + \mathbf{g}_i && (22) \end{aligned}$$

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\hat{\mathbf{q}} \tag{23}$$

Writing out these equations gives:

$$\dot{\hat{p}}_{x_i} = \hat{v}_{x_i} \tag{24}$$
$$\dot{\hat{p}}_{y_i} = \hat{v}_{y_i} \tag{25}$$
$$\dot{\hat{p}}_{z_i} = \hat{v}_{z_i} \tag{26}$$
$$\dot{\hat{v}}_{x_i} = \left(\hat{q}_1^2 + \hat{q}_2^2 - \hat{q}_3^2 - \hat{q}_4^2\right) f_x + 2\left(\hat{q}_2\hat{q}_3 - \hat{q}_1\hat{q}_4\right) f_y + 2\left(\hat{q}_2\hat{q}_4 + \hat{q}_1\hat{q}_3\right) f_z \tag{27}$$
$$\dot{\hat{v}}_{y_i} = 2\left(\hat{q}_2\hat{q}_3 + \hat{q}_1\hat{q}_4\right) f_x + \left(\hat{q}_1^2 - \hat{q}_2^2 + \hat{q}_3^2 - \hat{q}_4^2\right) f_y + 2\left(\hat{q}_3\hat{q}_4 - \hat{q}_1\hat{q}_2\right) f_z \tag{28}$$
$$\dot{\hat{v}}_{z_i} = 2\left(\hat{q}_2\hat{q}_4 - \hat{q}_1\hat{q}_3\right) f_x + 2\left(\hat{q}_3\hat{q}_4 + \hat{q}_1\hat{q}_2\right) f_y + \left(\hat{q}_1^2 - \hat{q}_2^2 - \hat{q}_3^2 + \hat{q}_4^2\right) f_z \tag{29}$$
$$\dot{\hat{q}}_1 = 0.5\left(-p\hat{q}_2 - q\hat{q}_3 - r\hat{q}_4\right) \tag{30}$$
$$\dot{\hat{q}}_2 = 0.5\left(p\hat{q}_1 + r\hat{q}_3 - q\hat{q}_4\right) \tag{31}$$
$$\dot{\hat{q}}_3 = 0.5\left(q\hat{q}_1 - r\hat{q}_2 + p\hat{q}_4\right) \tag{32}$$
$$\dot{\hat{q}}_4 = 0.5\left(r\hat{q}_1 + q\hat{q}_2 - p\hat{q}_3\right) \tag{33}$$

The following equations are the non-zero components of the Jacobian matrix $\mathbf{A}$ obtained by linearizing (24)-(33):

$$\frac{\partial \dot{\hat{p}}_{x_i}}{\partial \hat{v}_{x_i}} = 1, \qquad \frac{\partial \dot{\hat{p}}_{y_i}}{\partial \hat{v}_{y_i}} = 1, \qquad \frac{\partial \dot{\hat{p}}_{z_i}}{\partial \hat{v}_{z_i}} = 1 \tag{34}$$

$$\frac{\partial \dot{\hat{v}}_{x_i}}{\partial \hat{q}_1} = 2\left(\hat{q}_1 f_x - \hat{q}_4 f_y + \hat{q}_3 f_z\right) \qquad \frac{\partial \dot{\hat{v}}_{x_i}}{\partial \hat{q}_2} = 2\left(\hat{q}_2 f_x + \hat{q}_3 f_y + \hat{q}_4 f_z\right) \tag{35}$$
$$\frac{\partial \dot{\hat{v}}_{x_i}}{\partial \hat{q}_3} = 2\left(-\hat{q}_3 f_x + \hat{q}_2 f_y + \hat{q}_1 f_z\right) \qquad \frac{\partial \dot{\hat{v}}_{x_i}}{\partial \hat{q}_4} = 2\left(-\hat{q}_4 f_x - \hat{q}_1 f_y + \hat{q}_2 f_z\right) \tag{36}$$
$$\frac{\partial \dot{\hat{v}}_{y_i}}{\partial \hat{q}_1} = 2\left(\hat{q}_4 f_x + \hat{q}_1 f_y - \hat{q}_2 f_z\right) \qquad \frac{\partial \dot{\hat{v}}_{y_i}}{\partial \hat{q}_2} = 2\left(\hat{q}_3 f_x - \hat{q}_2 f_y - \hat{q}_1 f_z\right) \tag{37}$$
$$\frac{\partial \dot{\hat{v}}_{y_i}}{\partial \hat{q}_3} = 2\left(\hat{q}_2 f_x + \hat{q}_3 f_y + \hat{q}_4 f_z\right) \qquad \frac{\partial \dot{\hat{v}}_{y_i}}{\partial \hat{q}_4} = 2\left(\hat{q}_1 f_x - \hat{q}_4 f_y + \hat{q}_3 f_z\right) \tag{38}$$
$$\frac{\partial \dot{\hat{v}}_{z_i}}{\partial \hat{q}_1} = 2\left(-\hat{q}_3 f_x + \hat{q}_2 f_y + \hat{q}_1 f_z\right) \qquad \frac{\partial \dot{\hat{v}}_{z_i}}{\partial \hat{q}_2} = 2\left(\hat{q}_4 f_x + \hat{q}_1 f_y - \hat{q}_2 f_z\right) \tag{39}$$
$$\frac{\partial \dot{\hat{v}}_{z_i}}{\partial \hat{q}_3} = 2\left(-\hat{q}_1 f_x + \hat{q}_4 f_y - \hat{q}_3 f_z\right) \qquad \frac{\partial \dot{\hat{v}}_{z_i}}{\partial \hat{q}_4} = 2\left(\hat{q}_2 f_x + \hat{q}_3 f_y + \hat{q}_4 f_z\right) \tag{40}$$

$$\frac{\partial \dot{\hat{q}}_1}{\partial \hat{q}_2} = -0.5p, \qquad \frac{\partial \dot{\hat{q}}_1}{\partial \hat{q}_3} = -0.5q, \qquad \frac{\partial \dot{\hat{q}}_1}{\partial \hat{q}_4} = -0.5r, \tag{41}$$
$$\frac{\partial \dot{\hat{q}}_2}{\partial \hat{q}_1} = 0.5p, \qquad \frac{\partial \dot{\hat{q}}_2}{\partial \hat{q}_3} = 0.5r, \qquad \frac{\partial \dot{\hat{q}}_2}{\partial \hat{q}_4} = -0.5q, \tag{42}$$
$$\frac{\partial \dot{\hat{q}}_3}{\partial \hat{q}_1} = 0.5q, \qquad \frac{\partial \dot{\hat{q}}_3}{\partial \hat{q}_2} = -0.5r, \qquad \frac{\partial \dot{\hat{q}}_3}{\partial \hat{q}_4} = 0.5p, \tag{43}$$
$$\frac{\partial \dot{\hat{q}}_4}{\partial \hat{q}_1} = 0.5r, \qquad \frac{\partial \dot{\hat{q}}_4}{\partial \hat{q}_2} = 0.5q, \qquad \frac{\partial \dot{\hat{q}}_4}{\partial \hat{q}_3} = -0.5p, \tag{44}$$

C. Measurement Model

The measurement model here describes how the expected measurement $\hat{\mathbf{z}} = h(\hat{\mathbf{x}})$ is computed from the propagated state estimate. In order to describe these equations in a succinct manner, the vectors in Figure 3 are first introduced where $\mathbf{p}$ is the position of the vehicle, $\mathbf{p}_{fp}$ is the position of a feature point, and $\mathbf{r}$ is the relative position of the feature point with respect to the vehicle. We will denote the relative position vector $\mathbf{r}$ in the camera frame as $\mathbf{r}_c = [X_{fp_c} \;\; Y_{fp_c} \;\; Z_{fp_c}]^T$, and similarly in the body frame as $\mathbf{r}_b = [X_{fp_b} \;\; Y_{fp_b} \;\; Z_{fp_b}]^T$ and in the local inertial frame as $\mathbf{r}_i = [X_{fp_i} \;\; Y_{fp_i} \;\; Z_{fp_i}]^T$.

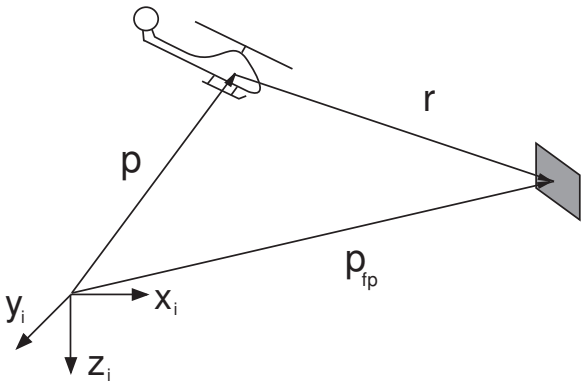

Figure 3. Vectors used in describing the EKF measurement model.

For each feature point, the expected measurement is computed as $\hat{\mathbf{z}} = [\hat{X} \quad \hat{Y}]^T$, where from the relations given in (1) and (2) we have

$$\hat{X} = f\frac{\hat{Y}_{fp_c}}{\hat{X}_{fp_c}} \tag{45}$$

$$\hat{Y} = f\frac{\hat{Z}_{fp_c}}{\hat{X}_{fp_c}} \tag{46}$$

The following describes the calculations of the partial derivatives needed for computing the Jacobians in the Kalman update based off of this measurement model. The partial derivatives of the measurement vector with respect to vehicle position in the inertial reference frame is computed as

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{p}}_i} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_i}\right) \tag{47}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \begin{bmatrix} \frac{\partial \hat{X}}{\partial \hat{X}_{fp_c}} & \frac{\partial \hat{X}}{\partial \hat{Y}_{fp_c}} & \frac{\partial \hat{X}}{\partial \hat{Z}_{fp_c}} \\ \frac{\partial \hat{Y}}{\partial \hat{X}_{fp_c}} & \frac{\partial \hat{Y}}{\partial \hat{Y}_{fp_c}} & \frac{\partial \hat{Y}}{\partial \hat{Z}_{fp_c}} \end{bmatrix} \tag{48}$$

$$\frac{\partial \hat{X}}{\partial \hat{X}_{fp_c}} = -f\frac{\hat{Y}_{fp_c}}{\hat{X}^2_{fp_c}} = -\frac{\hat{X}}{\hat{X}_{fp_c}}, \qquad \frac{\partial \hat{X}}{\partial \hat{Y}_{fp_c}} = \frac{f}{\hat{X}_{fp_c}}, \qquad \frac{\partial \hat{X}}{\partial \hat{Z}_{fp_c}} = 0 \tag{49}$$

$$\frac{\partial \hat{Y}}{\partial \hat{X}_{fp_c}} = -f\frac{\hat{Z}_{fp_c}}{\hat{X}^2_{fp_c}} = -\frac{\hat{Y}}{\hat{X}_{fp_c}}, \qquad \frac{\partial \hat{Y}}{\partial \hat{Y}_{fp_c}} = 0, \qquad \frac{\partial \hat{Y}}{\partial \hat{Z}_{fp_c}} = \frac{f}{\hat{X}_{fp_c}} \tag{50}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \frac{1}{\hat{X}_{fp_c}}\begin{bmatrix} -\hat{X} & f & 0 \\ -\hat{Y} & 0 & f \end{bmatrix} \tag{51}$$

$$\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_i} = \frac{\partial\left(\hat{\mathbf{p}}_{fp_c} - \hat{\mathbf{p}}_c\right)}{\partial \hat{\mathbf{p}}_i} = \frac{\partial \hat{\mathbf{p}}_{fp_c}}{\partial \hat{\mathbf{p}}_i} - \frac{\partial \hat{\mathbf{p}}_c}{\partial \hat{\mathbf{p}}_i} = 0 - \frac{\partial\left(\hat{\mathbf{L}}_{ci}\hat{\mathbf{p}}_i\right)}{\partial \hat{\mathbf{p}}_i} = -\hat{\mathbf{L}}_{ci} \tag{52}$$

The partial derivatives of the measurement vector with respect to the attitude quaternions are similarly computed as

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{q}}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{q}}}\right) = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\frac{\partial\left(\mathbf{L}_{cb}\hat{\mathbf{r}}_b\right)}{\partial \hat{\mathbf{q}}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\mathbf{L}_{cb}\left(\frac{\partial \hat{\mathbf{r}}_b}{\partial \hat{\mathbf{q}}}\right) \tag{53}$$

$$\frac{\partial \hat{X}_{fp_b}}{\partial \hat{q}_1} = 2\begin{pmatrix} \hat{q}_1\hat{X}_{fp_i} + \hat{q}_4\hat{Y}_{fp_i} - \hat{q}_3\hat{Z}_{fp_i} \end{pmatrix} \qquad \frac{\partial \hat{X}_{fp_b}}{\partial \hat{q}_2} = 2\begin{pmatrix} \hat{q}_2\hat{X}_{fp_i} + \hat{q}_3\hat{Y}_{fp_i} + \hat{q}_4\hat{Z}_{fp_i} \end{pmatrix} \tag{54}$$

$$\frac{\partial \hat{X}_{fp_b}}{\partial \hat{q}_3} = 2\left(-\hat{q}_3\hat{X}_{fp_i} + \hat{q}_2\hat{Y}_{fp_i} - \hat{q}_1\hat{Z}_{fp_i}\right) \qquad \frac{\partial \hat{X}_{fp_b}}{\partial \hat{q}_4} = 2\left(-\hat{q}_4\hat{X}_{fp_i} + \hat{q}_1\hat{Y}_{fp_i} + \hat{q}_2\hat{Z}_{fp_i}\right) \tag{55}$$

$$\frac{\partial \hat{Y}_{fp_b}}{\partial \hat{q}_1} = 2\left(-\hat{q}_4\hat{X}_{fp_i} + \hat{q}_1\hat{Y}_{fp_i} + \hat{q}_2\hat{Z}_{fp_i}\right) \qquad \frac{\partial \hat{Y}_{fp_b}}{\partial \hat{q}_2} = 2\begin{pmatrix} \hat{q}_3\hat{X}_{fp_i} - \hat{q}_2\hat{Y}_{fp_i} + \hat{q}_1\hat{Z}_{fp_i} \end{pmatrix} \tag{56}$$

$$\frac{\partial \hat{Y}_{fp_b}}{\partial \hat{q}_3} = 2\begin{pmatrix} \hat{q}_2\hat{X}_{fp_i} + \hat{q}_3\hat{Y}_{fp_i} + \hat{q}_4\hat{Z}_{fp_i} \end{pmatrix} \qquad \frac{\partial \hat{Y}_{fp_b}}{\partial \hat{q}_4} = 2\left(-\hat{q}_1\hat{X}_{fp_i} - \hat{q}_4\hat{Y}_{fp_i} + \hat{q}_3\hat{Z}_{fp_i}\right) \tag{57}$$

$$\frac{\partial \hat{Z}_{fp_b}}{\partial \hat{q}_1} = 2\begin{pmatrix} \hat{q}_3\hat{X}_{fp_i} - \hat{q}_2\hat{Y}_{fp_i} + \hat{q}_1\hat{Z}_{fp_i} \end{pmatrix} \qquad \frac{\partial \hat{Z}_{fp_b}}{\partial \hat{q}_2} = 2\begin{pmatrix} \hat{q}_4\hat{X}_{fp_i} - \hat{q}_1\hat{Y}_{fp_i} - \hat{q}_2\hat{Z}_{fp_i} \end{pmatrix} \tag{58}$$

$$\frac{\partial \hat{Z}_{fp_b}}{\partial \hat{q}_3} = 2\begin{pmatrix} \hat{q}_1\hat{X}_{fp_i} + \hat{q}_4\hat{Y}_{fp_i} - \hat{q}_3\hat{Z}_{fp_i} \end{pmatrix} \qquad \frac{\partial \hat{Z}_{fp_b}}{\partial \hat{q}_4} = 2\begin{pmatrix} \hat{q}_2\hat{X}_{fp_i} + \hat{q}_3\hat{Y}_{fp_i} + \hat{q}_4\hat{Z}_{fp_i} \end{pmatrix} \tag{59}$$

IV. Estimating Feature Point Positions With Known Vehicle States

This section addresses the problem of determining the positions of feature points when the vehicle states are known. This problem tackles the other part of the SLAM problem in that when GPS is available, the position and attitude information of the vehicle can be used to figure out the inertial positions of the feature points. With the estimated positions of the feature points, the vehicle can use this information to navigate in the event that GPS is lost. This ties in with the previous estimation filter in that this part comes up with the positions of feature points so that we can assume they are known.

A. Extended Kalman Filter Formulation

In this problem, it is assumed that the position ($\mathbf{p}_i$), velocity ($\mathbf{v}_i$), and orientation ($\mathbf{q}$) of the vehicle are known. The states to be estimated are $\mathbf{p}_{fp_i}$ for each feature point to be estimated. The measurements that are used in the EKF are the pixel positions of the feature points in the image plane ($\mathbf{z}_n = [X_n \;\; Y_n]$)

B. Process Model

Since the feature points are assumed to be stationary, the process model for this problem is very simple. The stationary points have no dynamics, so we have that $f(\hat{\mathbf{x}}) = 0$ and likewise $\mathbf{A} = 0$. This means that the only update that occurs in the prediction phase for this problem is the update of the covariance matrix according to

$$\dot{\mathbf{P}} = \mathbf{Q} \tag{60}$$

C. Measurement Model

As before, the equations for the expected measurement for each feature point is computed as $\hat{\mathbf{z}} = [\hat{X} \;\; \hat{Y}]^T$, where

$$\hat{X} = f\frac{\hat{Y}_{fp_c}}{\hat{X}_{fp_c}} \tag{61}$$

$$\hat{Y} = f\frac{\hat{Z}_{fp_c}}{\hat{X}_{fp_c}} \tag{62}$$

The following describes the calculations of the partial derivatives needed for computing the Jacobians in the Kalman update based off of the above measurement model equations.

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{p}}_{fp_i}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_{fp_i}}\right) \tag{63}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \frac{1}{\hat{X}_{fp_c}}\begin{bmatrix} -\hat{X} & f & 0 \\ -\hat{Y} & 0 & f \end{bmatrix} \tag{64}$$

$$\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \left(\hat{\mathbf{p}}_{fp_c} - \mathbf{p}_c \right)}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \hat{\mathbf{p}}_{fp_c}}{\partial \hat{\mathbf{p}}_{fp_i}} - \frac{\partial \mathbf{p}_c}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \left(\mathbf{L}_{ci} \hat{\mathbf{p}}_{fp_i} \right)}{\partial \hat{\mathbf{p}}_{fp_i}} - 0 = \mathbf{L}_{ci} \tag{65}$$

V. Results

Results for the above method were obtained using a hardware setup where an NTSC camera and an IMU were attached to a rigid aircraft body for testing purposes as shown in Figure 4. This vehicle is marked with reflective markers so that it can be tracked with a high precision Vicon motion capture system. The Vicon motion capture system provides position and attitude measurements of the vehicle at a rate of 100 Hz, and it has been claimed to have on the order of millimeter accuracy. This vehicle was then moved around by hand to simulate motion during these tests.

Figure 4. The experimental setup used for testing the localization and mapping algorithms. The Vicon system consists of the infrared cameras shown in the background. The vision sensor and the IMU are on the black vehicle marked with the reflective markers. A sample target is shown on the ground with black rectangles against a white background.

The IMU used was a Microstrain 3DM-GX1 IMU. Targets of black rectangles against a white background were placed on the ground for the camera to look at. The camera and IMU were directly connected to a desktop computer with a dual core 2.4 GHz processor with 2GB of RAM. A framegrabber card digitizes interlaced images from the camera at a rate of 30 frames per second, and the IMU data is read directly over a RS-232 serial port at a rate of 100 Hz. A sample output of the framegrabber and image processor is shown in Figure 5.

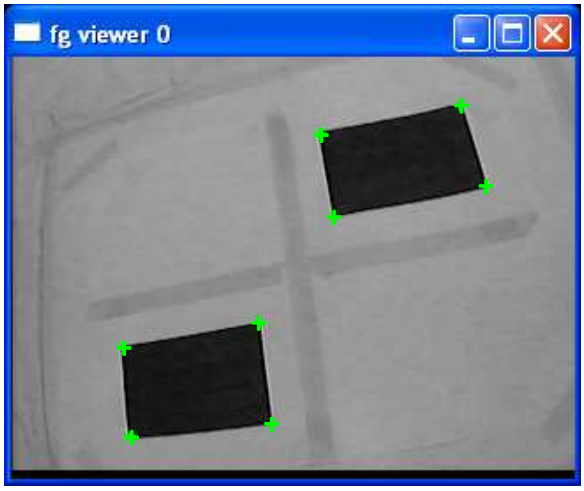

Figure 5. Sample output image from the framegrabber and image processing. The detected features are marked by the green crosses.

A. Results for Localization

In this setup, the targets used are two black rectangles with dimensions 4.5 inches in height and 7 inches in width. The positions of these targets are assumed to be known so that the vehicle can use the locations of the eight total corners for localization. Figures 6 to 14 show results for estimating the vehicle position and attitude using only measurements from the IMU and the monocular camera. Figures 6 to 12 show the estimated positions and attitude angles of the aircraft compared with those provided by the Vicon motion capture system. The results from the Vicon system have been bias shifted by $[0.425 \ -0.175 \ 0.25]^T$ ft in position and -5.7 degrees in heading angle. The filter is initialized to have a position of $[0.00.0 - 3.0]^T$ ft with zero velocity and a 90 degree downwards pitch angle.

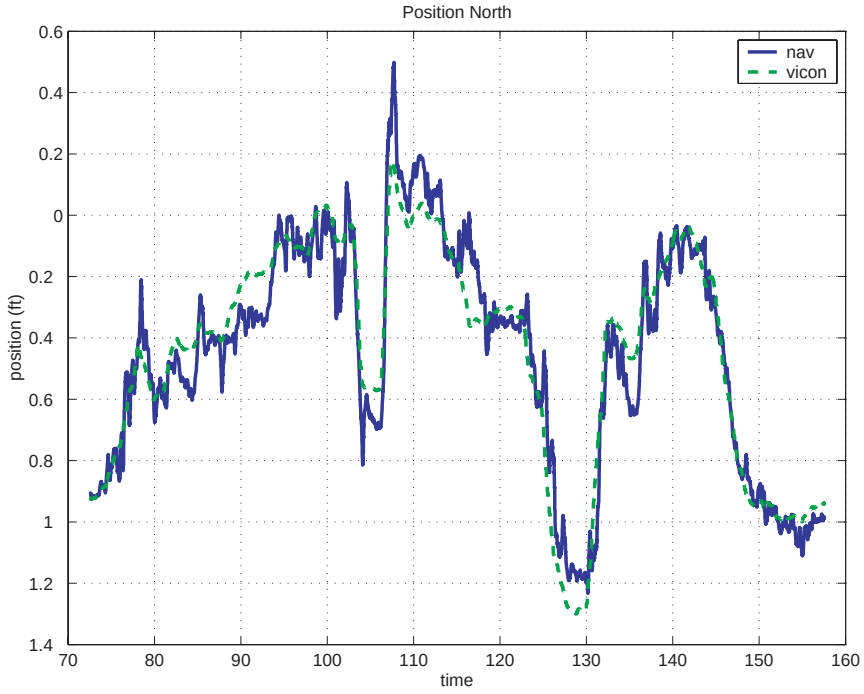

Figure 6. Position North for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by 0.425 ft.

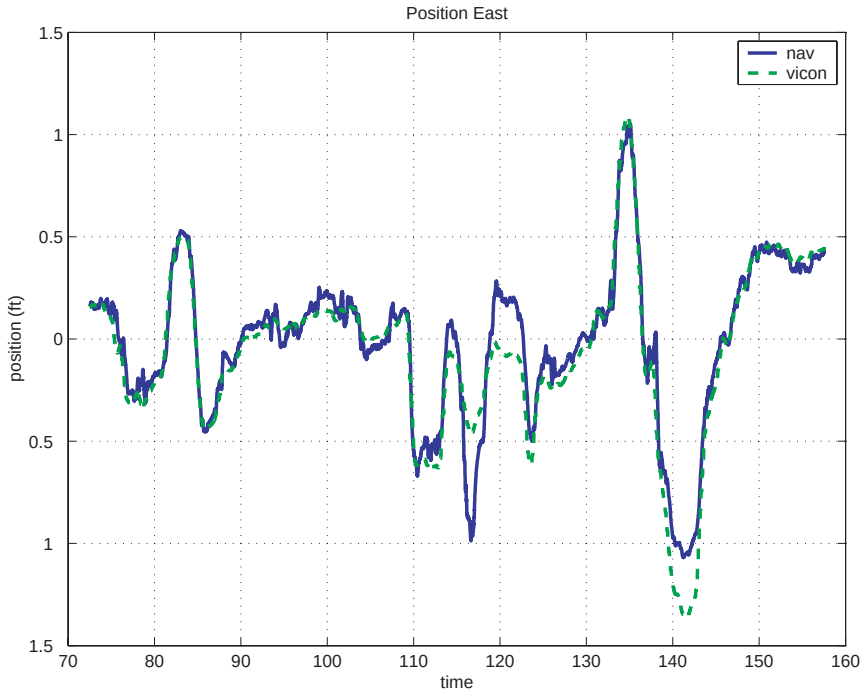

Figure 7. Position East for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by -0.175 ft.

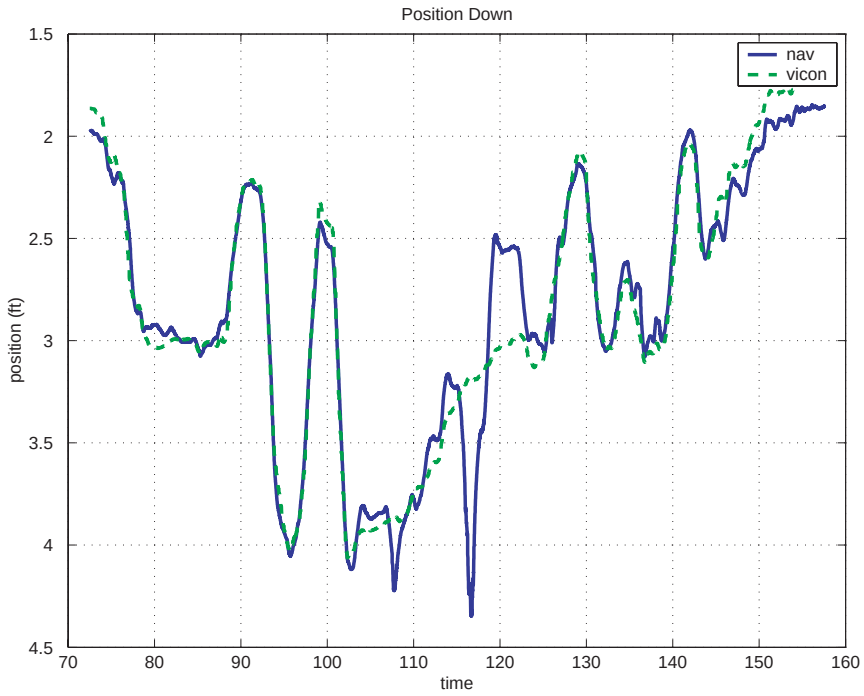

Figure 8. Position Down for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by 0.25 ft.

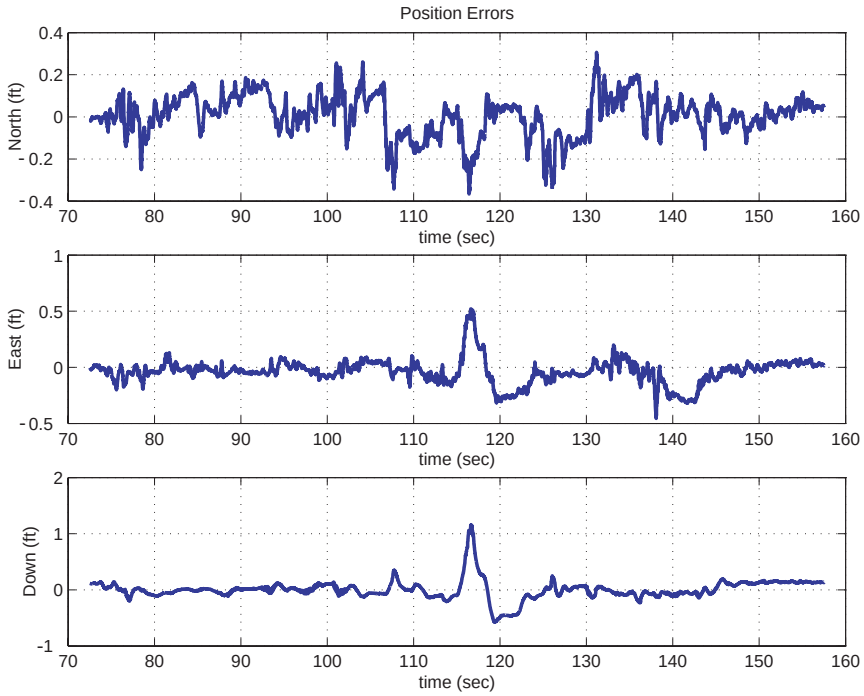

Figure 9. Position errors.

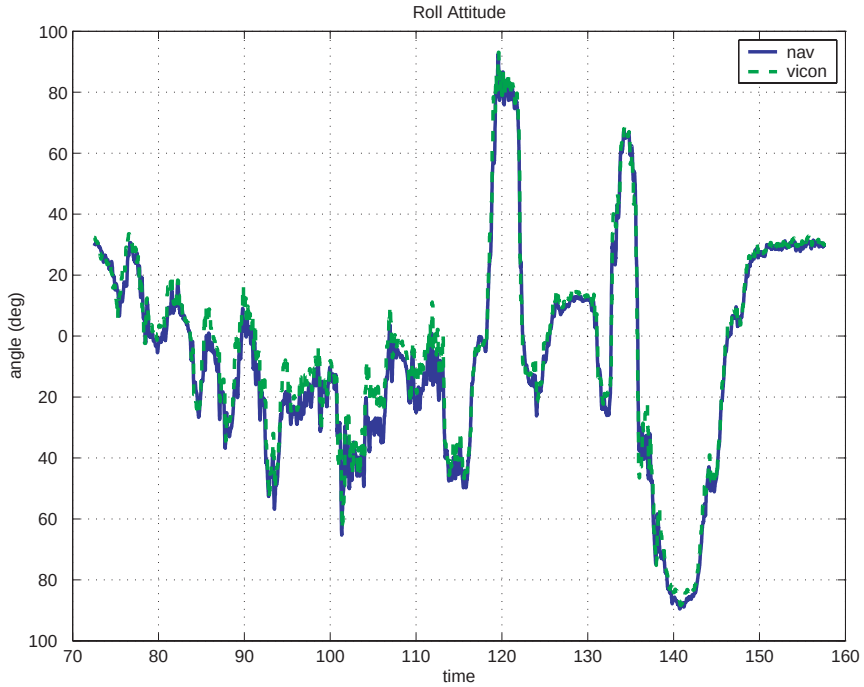

Figure 10. Roll attitude for the vehicle when using vision and IMU only for vehicle pose estimation.

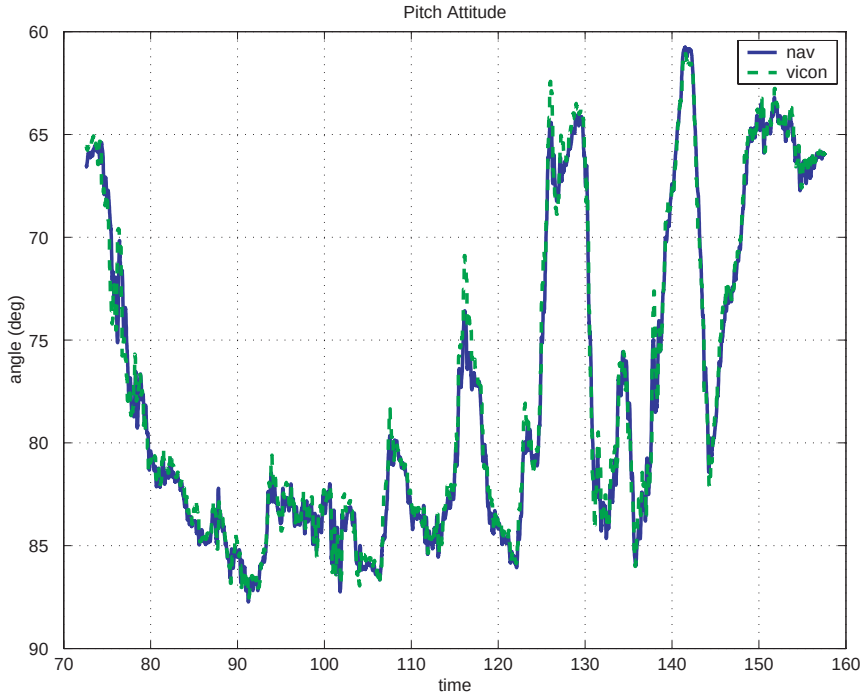

Figure 11. Pitch attitude for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have not been biased.

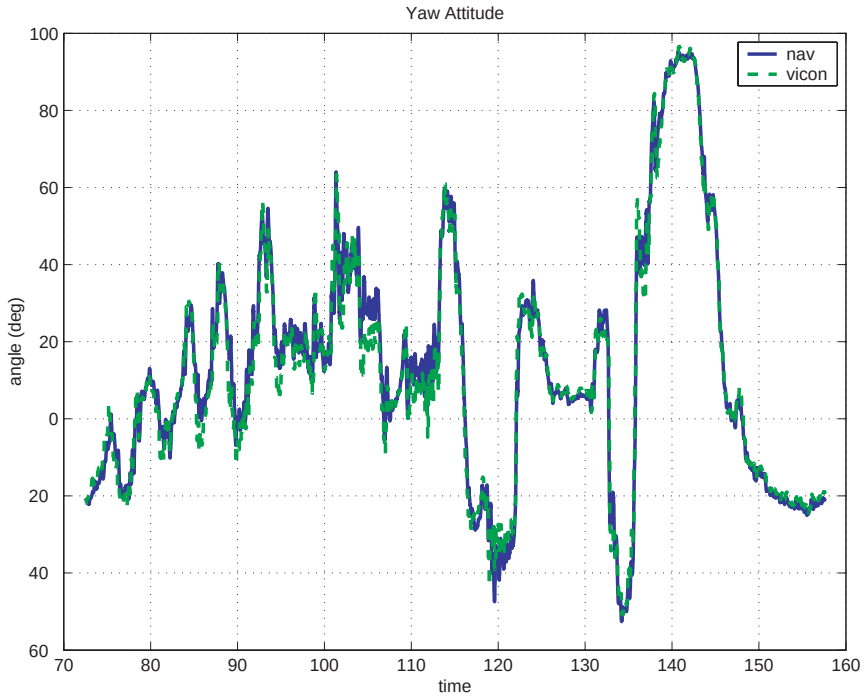

Figure 12. Yaw attitude for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by -5.7 degrees.

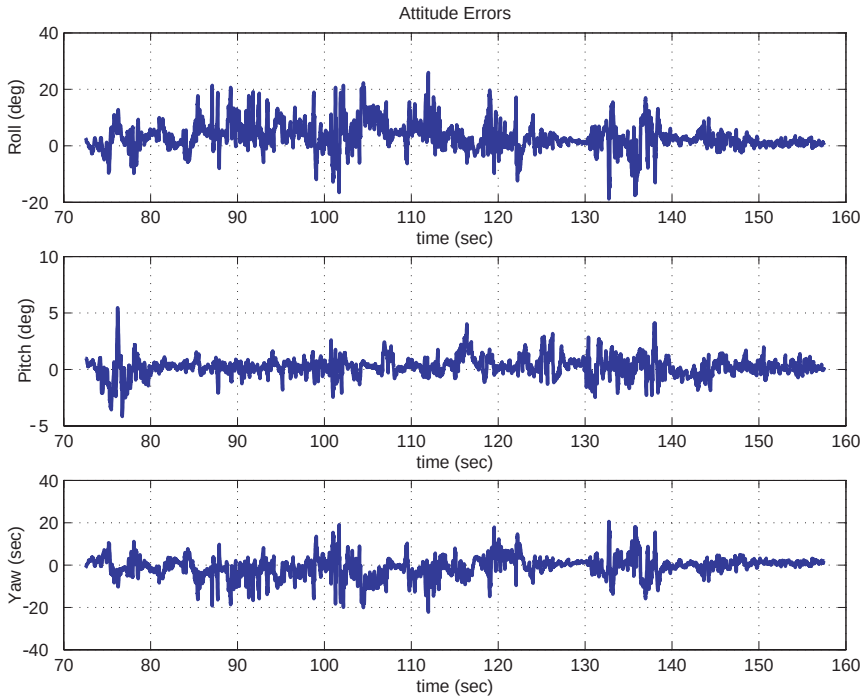

Figure 13. Euler angle errors for the attitude of the vehicle. The Vicon values in this plot have not been biased.

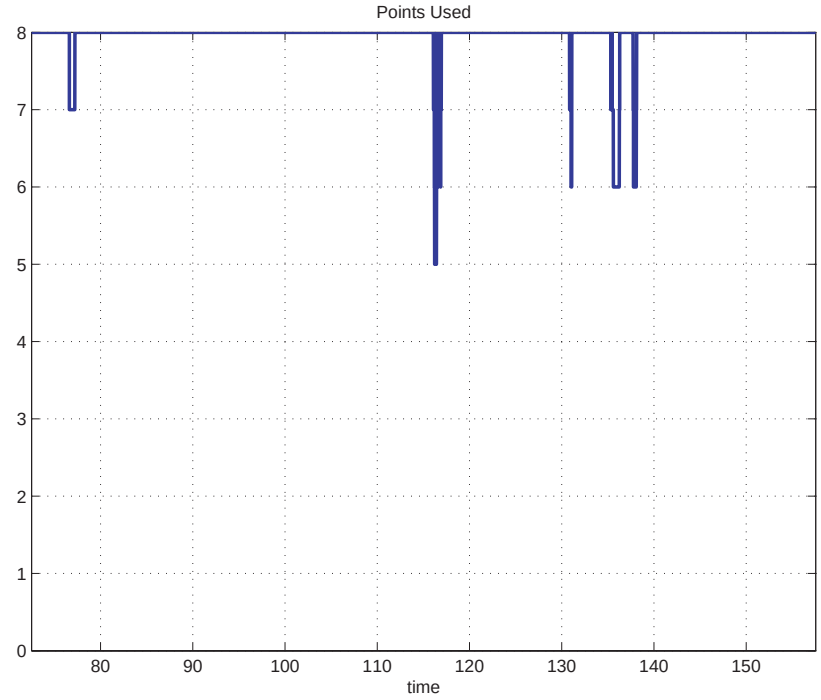

Figure 14. Number of points used in estimation. This depends on the number of feature points in the camera's field of view, and which points the state estimator expects to be in the field of view.

B. Results for Mapping

In this setup, the target used is a single black rectangle with a height of 22 inches and a width of 28 inches so that there are only four features to detect. The target is placed on the ground so that it's center is located approximately at the origin of the local inertial frame. The position and attitude of the aircraft are provided to the estimation computer by means of the Vicon motion capture system. In this setup, the camera starts out resting on the ground with the lens pointing downwards and no features visible. After resting on the ground for a period of time, the vehicle is then lifted off the ground to a position where all four feature points are visible within the camera's field of view. The vehicle is then moved around in small rapid motions at a roughly constant altitude to provide the estimator with as many different perspectives as possible to aid in the observability of the system. These movements, however, attempt to keep all four corners of the target within the camera's field of view whenever possible.

Figures 15 and 16 show the vehicle position and attitude throughout the test. Figures 17 through 20 show the estimates of the positions of the feature points. Even though we know the size of the target, the initial estimates are given to be a few inches off from the expected positions. Figure 21 shows the number of feature points being used at a given time in the estimation. The estimates of the location of the feature points take some time to converge, but the estimator is at least stable and the estimates remain bounded. Figure 22 shows the time evolution of the estimates of the feature points in the North-East plane. The circles represent the initial guesses for the features and the squares represent the most recent estimate of the features. The true locations of the features is not known, but the dimensions of the black target rectangle are known and can be used for comparison. Ideally, the estimates of the feature points should be such that they represent the corners of a 28"x22" rectangle. However, by looking at Figure 22, it can be seen that the estimated corners roughly represent a rectangle with dimensions 35"x28".

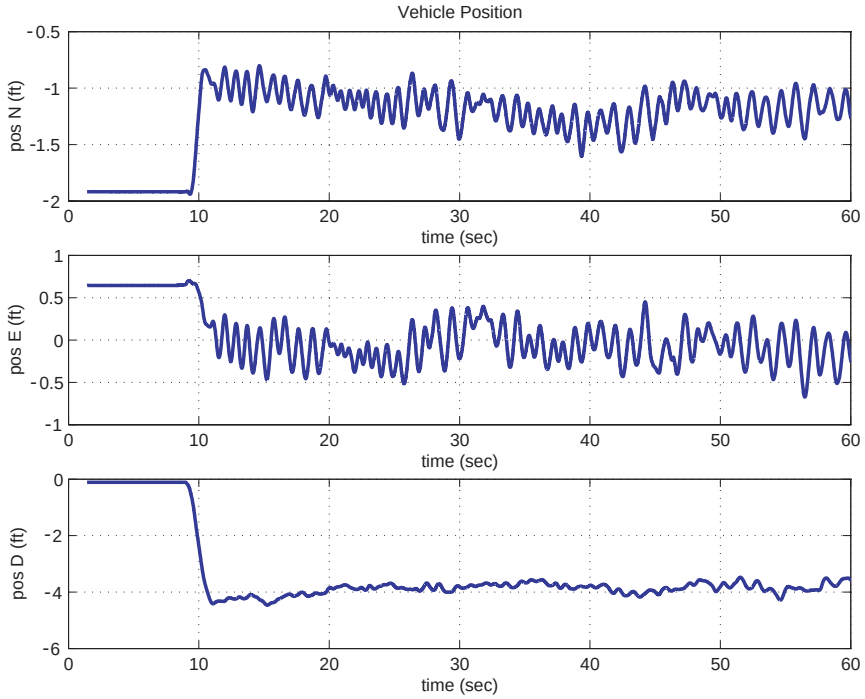

Figure 15. Vehicle position as given by the Vicon system while the vehicle is estimating the locations of the feature points.

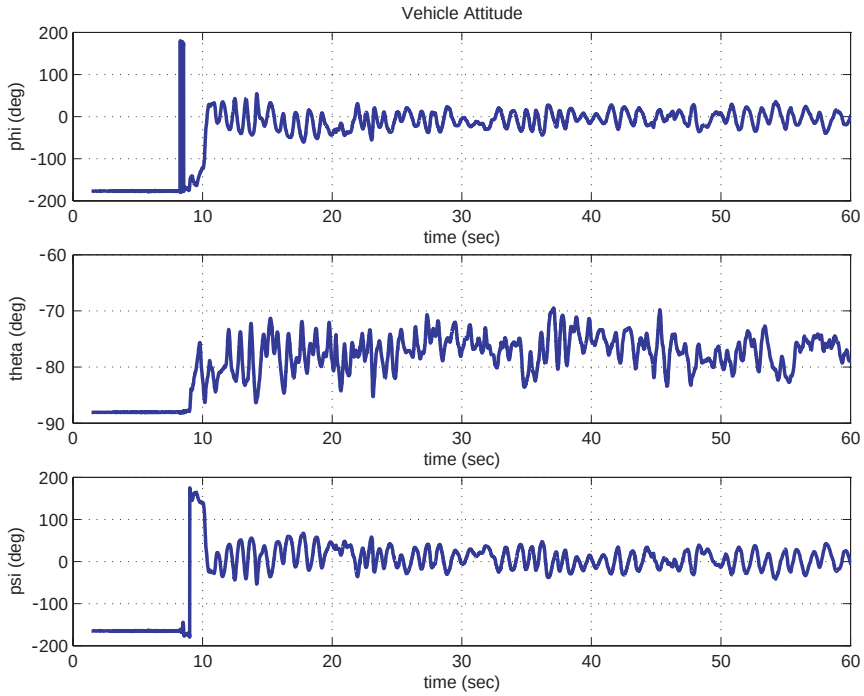

Figure 16. Vehicle attitude in Euler angles as given by the Vicon system while the vehicle is estimating the locations of the feature points.

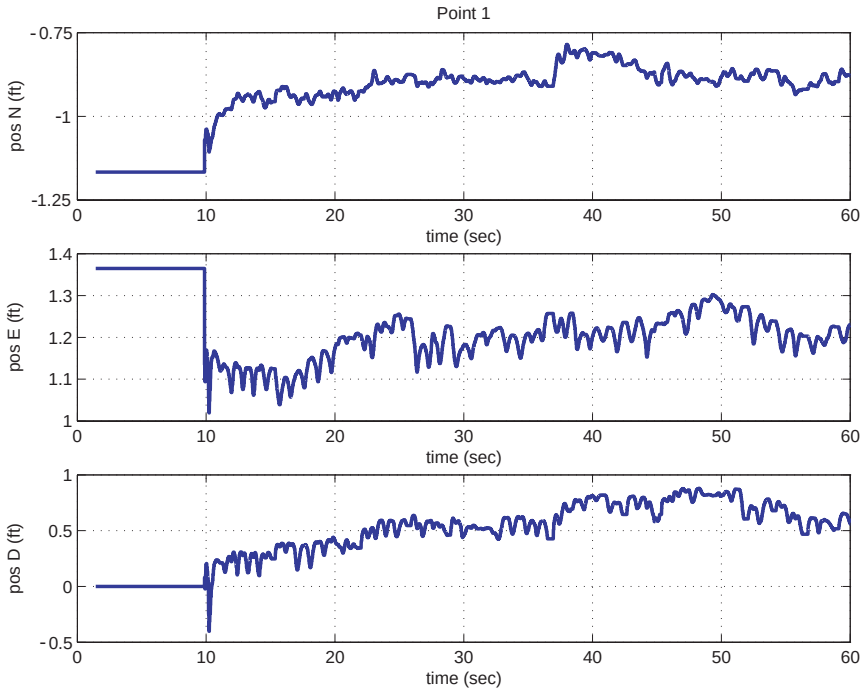

Figure 17. The estimated location of feature point 1 (lower right corner of target). The initial guess for the location is $[-1.167\ 1.365\ 0.0]^T$**.**

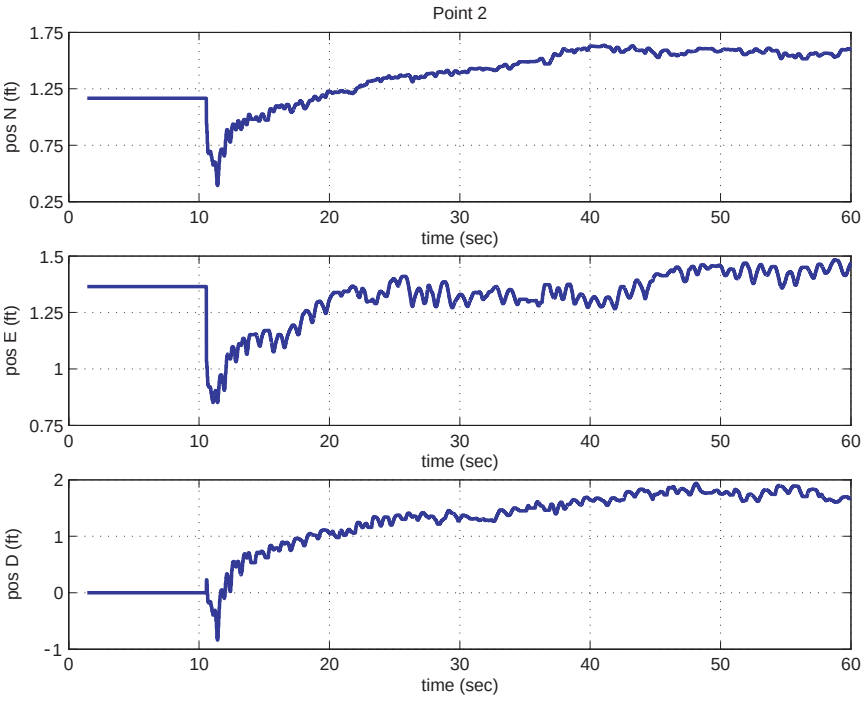

Figure 18. The estimated location of feature point 2 (top right corner of target). The initial guess for the location is $[1.167\ 1.365\ 0.0]^T$**.**

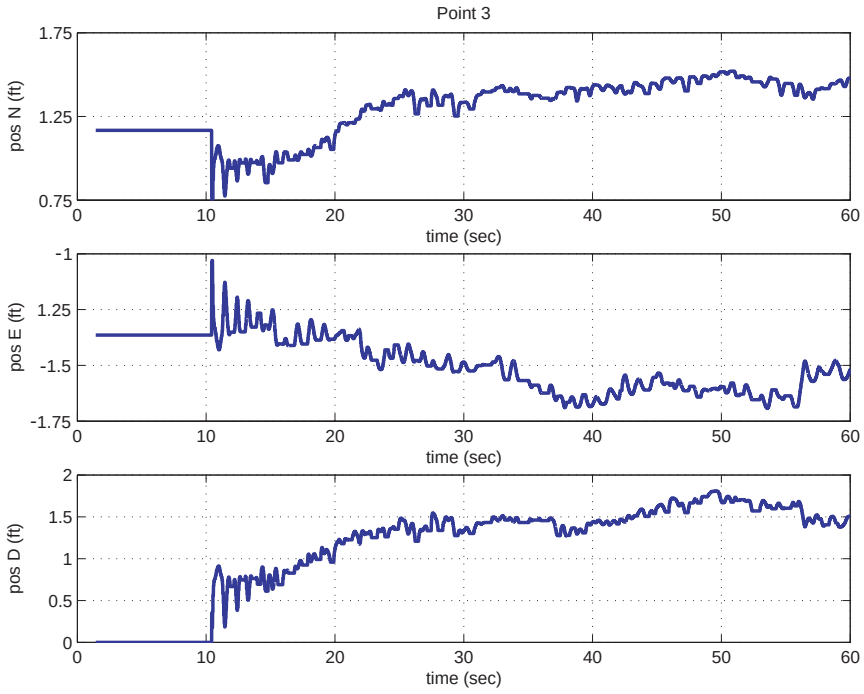

Figure 19. The estimated location of feature point 3 (top left corner of target). The initial guess for the location is $[1.167\ -1.365\ 0.0]^T$**.**

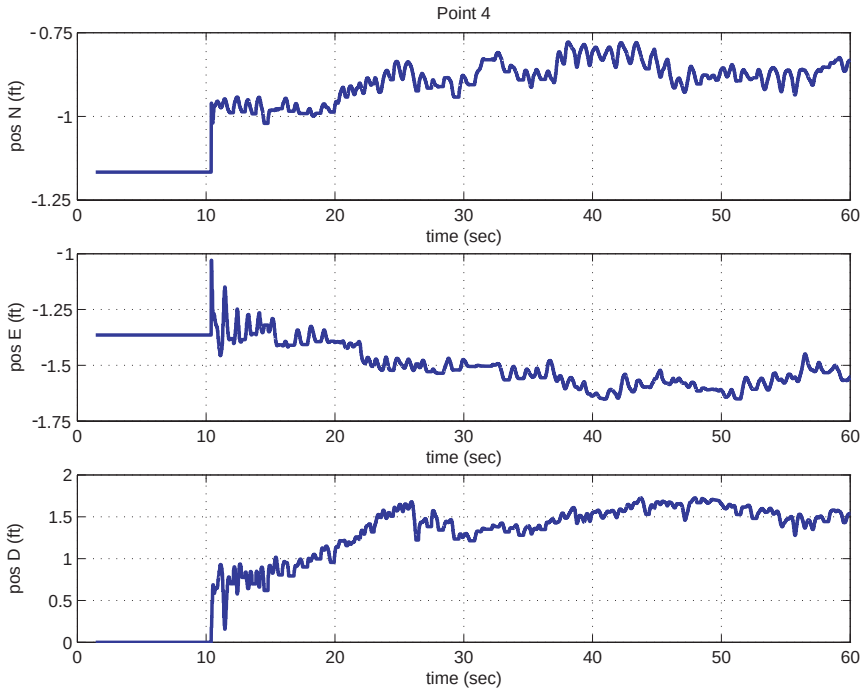

Figure 20. The estimated location of feature point 4 (lower left corner of target). The initial guess for the location is $[-1.167\ -1.365\ 0.0]^T$**.**

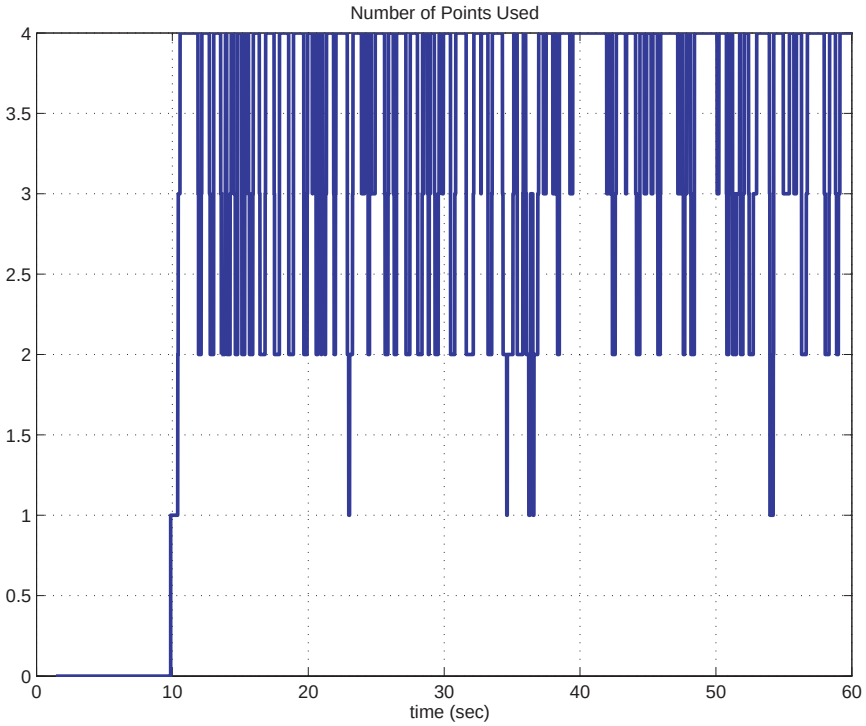

Figure 21. The number of feature points being estimated at a given moment. This depends on the number of feature points in the camera's field of view, and which points the state estimator expects to be in the field of view.

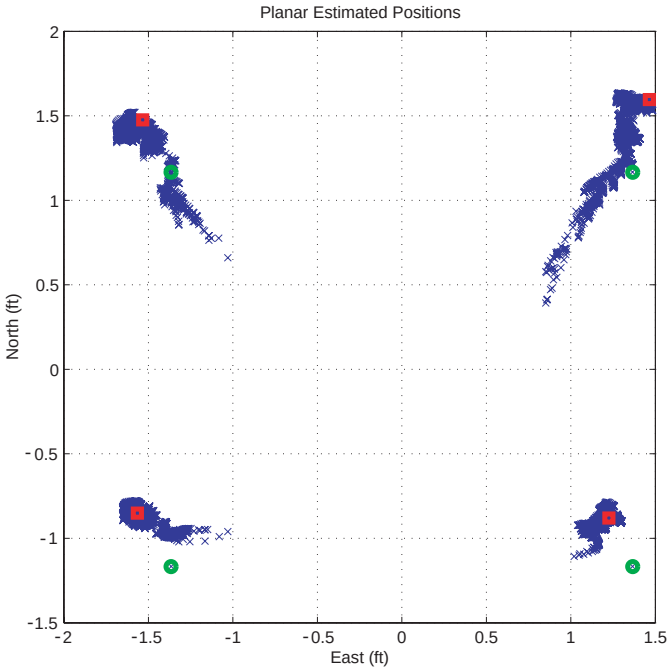

Figure 22. The North and East positions of the estimated points. The green circles represent the initial guesses and the red squares represent the most recent estimated locations of the points to show the progression of the estimates. Note that the actual target is 2.33 ft wide and 1.83 ft in height.

VI. Observations from the Hardware Experiments

There were a few interesting observations from the hardware implementation of the vision-aided inertial localization and mapping algorithms used in this paper that are worth noting. It was found that when using these two-dimensional targets, the performance of the localization algorithm was significantly better when the target was placed on the ground as opposed to mounting the target in an upright position such as on a wall (see Figure 23). In fact, it was difficult to get the filter to even converge when the target was mounted on a vertical surface. Conceptually, this occurs because when the vehicle is moving around and looking at the target, the measurements from the image processor may not necessarily change significantly because of the geometry of the problem. Rotation parallel to the target is well determined, but the assistance of the knowledge of the gravity vector is required for the other directions. So when the target is on the ground, gravity helps to resolve this discrepancy by providing information about the pitch and yaw of the vehicle. Similarly, if the target was placed on a vertical surface, a magnetometer could probably assist the filter in a similar manner.

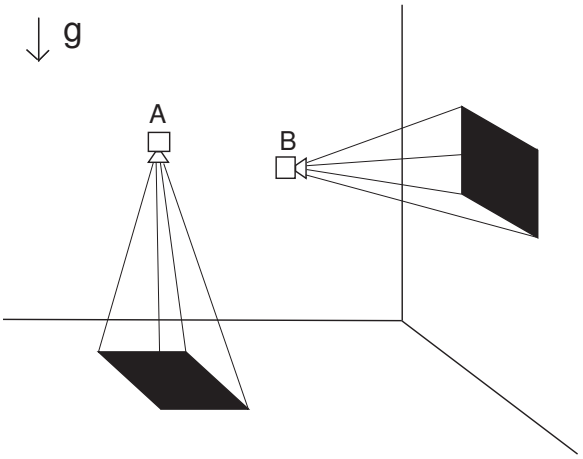

Figure 23. Placing the image processing targets on the ground (configuration A) provided markedly better results than when the target was placed upright on a wall (configuration B). This is because when the camera is looking down at the target, then the gravity vector assists in determining the relative pose of the vehicle.

With the target on the ground, the localization estimator proved to work reliably. Some care did however need to be taken in the initialization of the estimator since the initial position used by the estimator heavily affected the correspondence of the points. The vehicle needed to be in a position that would correctly associate the measurements with the correct points in the feature database by starting off with each point in the image processor being the closest measurement to the predicted measurement of its associated database point.

With regards to the results from the mapping algorithm, it was found that the filter would at least converge and bound the estimates of the locations of the feature points. However, convergence for these cases would sometimes take more than 10 seconds, and they would also drift towards incorrect values. Simulations were performed to investigate what properties of the vision sensors would most likely cause this type of problem to occur. Effects such as latency, errors in the field view, and errors in vehicle position and attitude from the Vicon system were investigated. Out of all these, it was found that errors in the camera's field of view (and equivalently focal length) could potentially cause this sort of a performance degradation. Figures 24 and 25 show results comparing situations where the field of view of the camera is known exactly (Figure 24) and where the assumed field of view is incorrect (Figure 25). These simulations simulate a camera which has a 50 degree field of view in both the horizontal and vertical directions (note that most cameras have a 4:3 aspect ratio, but for simplicities sake we just assume them to be identical in this simulation). The camera is pointing North the whole time and is moving in a circle around the origin in the East-Down plane with a radius of 10 ft at a velocity of 1 ft/s. The feature point is located at $[30\ 2\ 2]^T$ ft. The initial guess has a 5 ft error in all three axes. Figure 24 shows the estimation errors for when the correct field of view values are used. However, Figure 24 shows the estimation errors when it is assumed that the horizontal field of view is known correctly, but the vertical field of view is erroneously measure as 47 degrees (a 3 degree error). In the case of the incorrectly measured vertical field of view, convergence takes substantially longer and the steady state error is non-zero. This suggests that

a careful calibration of the camera or a more sophisticated camera model might be needed to improve the mapping performance.

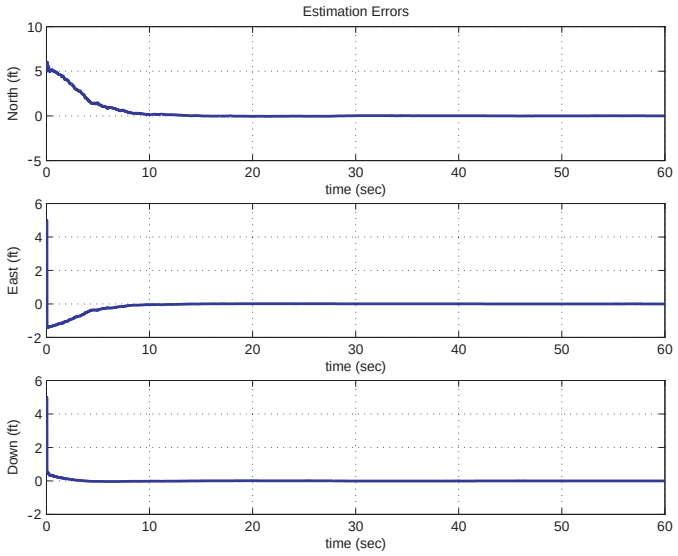

Figure 24. Estimation errors for the observation of feature point in a simulated case where the field of view parameters of the camera are exactly known. The vehicle maintains a constant orientation and moves in a circular motion while facing the feature point and maintaining it in sight at all times.

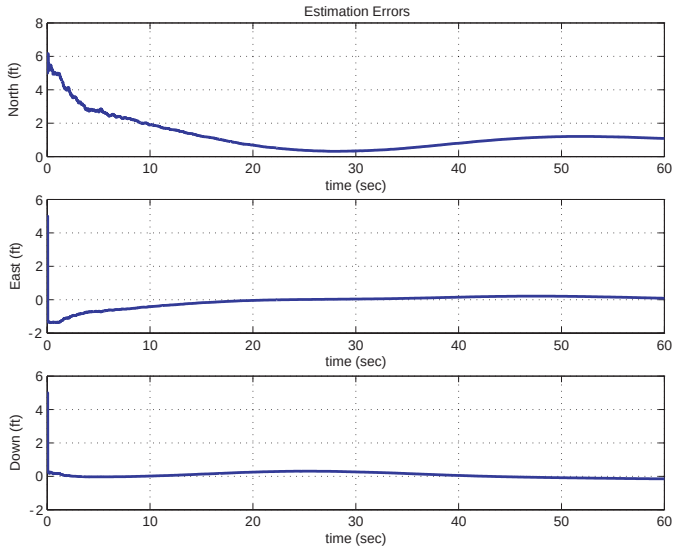

Figure 25. Estimation errors for the observation of feature point in a simulated case where the field of view parameters of the camera are not exactly known. A 3 degree error is induced in the vertical field of view in this case. The vehicle maintains a constant orientation and moves in a circular motion while facing the feature point and maintaining it in sight at all times.

VII. Conclusion

Preliminary results for separately performing localization and mapping were presented in this paper. The localization performance was compared with a high precision motion capture system for comparison purposes. In the mapping section, the vehicle pose was provided by the motion capture system, and the vehicle used this information to estimate the locations of four feature points. The obtained results demonstrate that vision sensors have the potential to provide accurate estimates for the position and attitude of a vehicle given good initial conditions, and suggest that the localization algorithm can potentially be applied to vision-aided inertial navigation relative to a known stationary target. Work still remains with getting the mapping portion to a satisfactory level. This includes investigating more sophisticated camera calibration models as well as possibly other estimation schemes that are more robust to these parametric uncertainties. It is hoped that these algorithms can be made robust enough to eventually be combined into an integrated method that allows for vision-aided inertial navigation in uncertain environments.

Acknowledgments

The authors would like to thank the following people for their contributions to the work presented in this paper: Dr. Suresh Kannan, Nimrod Rooz, Dr. Eric Feron, Dr. Frank Dellaert, and Michael Kaess. This work was supported in part by AFOSR MURI #F49620-03-1-0401 for Active Vision Control Systems for Complex Adversarial 3D Environments, Stealth Robotics, and a grant from the Institute Robotics and Intelligent Machines at Georgia Tech.

References

[1]Kim, J. and Sukkarieh, S., "Autonomous Airborne Navigation in Unknown Terrain Environments," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 3, July 2004, pp. 1031-1045.

[2]Jacob Willem Langelaan, *State Estimation for Autonomous Flight in Cluttered Environments*, PhD thesis, Stanford University, Stanford, CA, March 2006.

[3]Mourikis, A.I. and Roumeliotis S.I., "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," *IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007.

[4]Koch, A., Wittich, H., and Thielecke, F., "A Vision-Based Navigation Algorithm for a VTOL-UAV," *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.

[5]Call, B., Beard, R., Taylor, C., and Barber, B., "Obstacle Avoidance For Unmanned Air Vehicles Using Image Feature Tracking," *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.

[6]Wu, A.D., Johnson, E.N., and Proctor, A.P., "Vision-Aided Inertial Navigation for Flight Control," *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, No. 9, 2005, pp. 348-360.

[7]Ivey, G.F. and Johnson, E.N., "Investigation of Methods for Simultaneous Localization and Mapping Using Vision Sensor," *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.

[8]Toupet, O., Paduano, J.D., Panish, R., Sedwick, R., and Frazzoli, E., "Augmenting State Estimates with Multiple Camera Visual Measurements," *AIAA Infotech Conference*, Rohnert Park, CA, May 2007.

[9]Watkins, A.S., Kehoe, J.J., and Lind, R., "SLAM for Flight through Urban Environments using Dimensionality Reduction," *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.

[10]Prazenica, R.J., Watkins, A., Kurdila, A.J., Ke, Q.K., and Kanade, T., "Vision-Based Kalman Filtering for Aircraft State Estimation and Structure from Motion,"*AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, August 2005.

[11]Lucas, B.D. and Kanade, T., "An Interative Image Registration Technique with an Application to Stereo Vision," *International Joint Conference on Artificial Intelligence*, Vancouver, British Columbia, August 1981.

[12]Lowe, D.G., "Distinctive Image Feature from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp.91-110.

[13]Harris, C. and Stephens M., "A Combined Corner and Edge Detector," *Alvey Vision Conference*, 1988.