Autonomous Flight in GPS-Denied Environments Using Monocular Vision and Inertial Sensors

Allen D. Wu*, Eric N. Johnson†,
Michael Kaess‡, Frank Dellaert§, and Girish Chowdhary¶

A vision-aided inertial navigation system that enables autonomous flight of an aerial vehicle in GPS-denied environments is presented. Particularly, feature point information from a monocular vision sensor are used to bound the drift resulting from integrating accelerations and angular rate measurements from an Inertial Measurement Unit (IMU) forward in time. An Extended Kalman filter framework is proposed for performing the tasks of vision-based mapping and navigation separately. When GPS is available, multiple observations of a single landmark point from the vision sensor are used to estimate the point's location in inertial space. When GPS is not available, points that have been sufficiently mapped out can be used for estimating vehicle position and attitude. Simulation and flight test results of a vehicle operating autonomously in a simplified loss-of-GPS scenario verify the presented method.

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 accelerometers and gyroscopes of the IMU can be integrated forward in time, and position updates from the GPS can be used to bound the errors that result from misalignment and bias errors of the IMU (see for example references 1–3). This solution to the navigation problem makes the navigation solution prone to certain modes of failure due to the reliance on the reception of external signals from the GPS satellite network. GPS signals can suffer from obstructions or multipath in cluttered environments, furthermore the reception of these signals can be jammed or otherwise denied. Researchers have investigated scanning range sensors to mapping the surrounding environment and use this information for navigation (see for example 4, 5). 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 information about vehicle motion from these images. However, with the exception of a few (notably 6–8),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 Vehicles (UAVs) 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 passive navigation and mapping system that uses only a combination of inertial and vision sensors would also be fully self-contained and would not be easily prone to jamming or detection.

Extensive literature covering the use of vision for mapping and navigation for aerial vehicles is currently available. One of the earlier works to appear in the area of vision-based navigation for UAVs was Amidi[9] which presented the development of a visual odometer for flying an autonomous helicopter over flat ground. The visual odometer used a stereo pair of cameras pointed at the ground together with angular rate measurements from gyroscopes to determine

*Graduate of the School of Aerospace Engineering, AIAA Member.
†Lockheed Martin Assoc. Professor of Avionics Integration, School of Aerospace Engineering, Georgia Institute of Technology, AIAA Member.
‡Research Scientist, Massachusetts Institute of Technology
§Assoc. Professor, School of Interactive Computing, Georgia Institute of Technology
¶Research Engineer II, School of Aerospace Engineering, Georgia Institute of Technology, AIAA Member.

Figure 1. Small unmanned aerial vehicles, such as the GTFirstResponder UAV shown above, are often not capable of carrying large computer systems. A monocular downward facing camera is a typical payload on such UAVs.

the position of the aircraft. The method involved first finding a template, and then tracking the template from frame to frame. Since a stereo pair was used, the relative location of the template with respect to the aircraft could be found from the range information. Position information was backed out from the template track by first removing the vehicle's rotation by using the gyroscopes, and then using the displacement of the template in each image. Flight results using a constrained testbed were presented in addition to results from an outdoor helicopter.

In 10, Langelaan used an unscented Kalman filter for simultaneously estimating vehicle states as well as landmark locations in inertial space. A Mahalonobis norm was used as a statistical correspondence for data association from frame-to-frame for each estimated landmark. New landmarks were initialized in the filter by performing a projection onto the ground plane. Simulation results for a UAV navigating through a 2D environment were presented.

In 11, Madison *et al.* presented an EKF design where the vehicle states were being estimated as well as the inertial locations of features in 3D space. Their research presented a method which would allow a vehicle with a traditional GPS-aided inertial navigation system (INS) to fly along and estimate the locations of features, and then in the absence of GPS, use these features for aiding the INS. Features were selected and tracked using a Lucas-Kanade feature tracker. A few different methods for initializing the estimated locations of tracked features were implemented. The authors opted for a method of motion stereo where multiple observations of a new feature are first taken, and then the 3D location of the point is computed using a least-squares solution to find the intersection of the observation rays. Simulation results for a vehicle that momentarily loses GPS were provided.

The authors in 12 implemented a Harris feature detector along with a random sample consensus (RANSAC) algorithm for the correspondence of features in an FPGA. In this work, Fowers *et al.* used the FPGA to also read in digital images from a digital CMOS camera. This was used for providing drift corrections to an INS to stabilize a quad-rotor vehicle for short-term hover in an indoor flight environment. By assuming that the vehicle remains relatively level, the RANSAC algorithm was used to provide estimates of the translation, yaw rotation, and change in scale. RANSAC is a model fitting algorithm where a collection of points are fitted against a model, and the points are sorted into groups of inliers and outliers. These estimated values were then used to provide drift correction measurements to the integration of the IMU.

Frietsch *et al.* in 13 presented an application of vision to the hovering and landing of a quad-rotor UAV. These authors presented a method for estimating the above ground level altitude without the need for ranging sensors by utilizing the optical flow from a vision sensor in conjunction with a barometric altimeter. The proposed method used a Lucas-Kanade tracker to perform the detection and tracking of features from frame to frame. A RANSAC algorithm was then used to estimate a homography that relates points on the ground plane as viewed from two different perspectives. To figure out the distance of the ground plane from the vehicle, the net optical flow of the scene was combined with readings from a barometric altimeter. Since this work was intended for the applications of hovering and landing, it was assumed that the camera's motion was dominated by rotation and that the scene was planar. Simulation results of the method were provided.

In this work, a method for fusing feature information from a monocular vision sensor with an inertial measurement unit (IMU) in an extended Kalman filter framework for separately tackling the navigation and mapping problems is

proposed. A Harris corner detector extracts feature points from each captured image, and a statistical z-test is used to correspond features from frame to frame. A method to initialize a database of detected features and an associated covariance matrix is presented. Simulation and flight test results using the mapping and localization filters demonstrate how the architecture can be used in a loss-of-GPS scenario. It should be noted that the main focus of this work is to present an algorithm for monocular vision aided GPS denied navigation and establish through flight testing the feasibility of this algorithm in sustaining autonomous flight of unstable rotorcraft in GPS denied environment. The work in this paper therefore addresses the relative lack of outdoor flight-test validated monocular vision aided navigation algorithms. As such, integration of long-term vision aided mapping using techniques established in the widely studied field of Simultaneous Localization and Mapping are not pursued in this paper.

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. Modifications to the extended Kalman filter are described to enable the handling of the uncertain number of measurements as well as the problem of corresponding measurements obtained to their actual point in inertial space. These issues are discussed in the section on 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 Camera Images

A basic pinhole model is often used to model how a point in 3D space is projected onto a 2D camera image. This model, however, often does not accurately reflect the actual behavior of cameras due to the physical optical distortions of the camera lens. In this section, an overview of the pinhole camera model is first presented, followed by an overview of some corrections for camera calibration and lens distortion.

1. The Basic Pinhole Camera Model

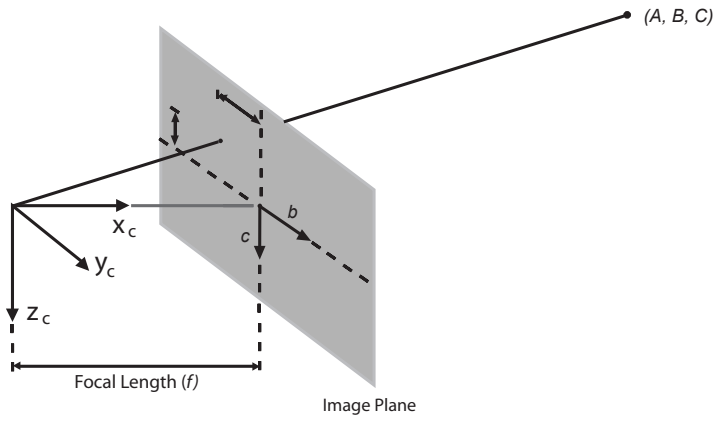

Figure 2. Camera perspective projection model used for relating 3D position to position in 2D images. The point (A, B, C) is projected onto the camera image plane to the point (b, c).

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 $b = f\frac{B}{A}$ and $c = f\frac{C}{A}$ where f is the focal length of the camera. However, in reality, the focal lengths in the horizontal and vertical directions may be different. So these expressions will be

rewritten as

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

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

where f_x and f_y are the focal lengths in the horizontal and vertical directions respectively. These focal lengths can be computed from knowledge of the width and height of the camera image plane (w and h) and from the horizontal and vertical fields of view (γ_x and γ_y) according to

$$f_x = \frac{w}{2\tan\left(\frac{\gamma_x}{2}\right)}, \qquad f_y = \frac{h}{2\tan\left(\frac{\gamma_y}{2}\right)} \tag{3}$$

2. Camera Calibration and Lens Distortion

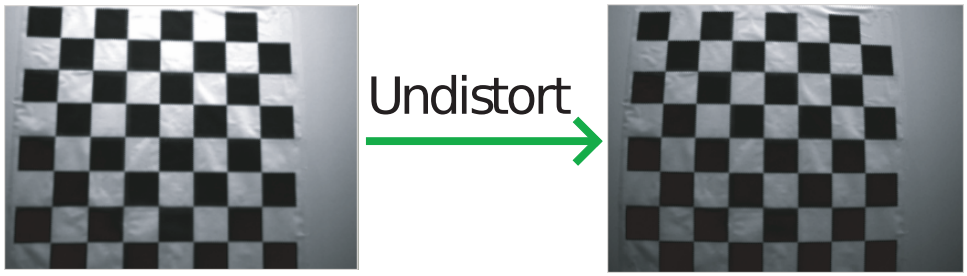

Figure 3. An example of the effect of distortion caused by a camera lens. The image on the left is the image obtained directly from a camera with a wide angle lens whereas the right image shows the undistorted version of the image.

The rectilinear perspective projection model for an ideal camera often does not accurately reflect the behavior of physical cameras due to the effects of lens distortions. Lens optics tend to warp the actual image captured by the camera sensor so that straight lines no longer appear straight in the image. Therefore, correcting of camera distortion is necessary. Reference 16 provides a model that is commonly used for representing the undistorted pixel locations after correcting for radial and tangential lens distortions:

$$x_u = x_d\left(1 + k_1 r^2 + k_2 r^4\right) 2p_1 x_d y_d + p_2\left(r^2 + 2{x_d}^2\right) \tag{4}$$

$$y_u = y_d\left(1 + k_1 r^2 + k_2 r^4\right) 2p_2 x_d y_d + p_1\left(r^2 + 2{y_d}^2\right) \tag{5}$$

where $r^2 = {x_d}^2 + {y_d}^2$ and $x_d = X - p_x$, $y_d = Y - p_y$. The k_1 and k_2 parameters are radial distortion coefficients whereas p_1 and p_2 are the tangential distortion coefficients. The offset of the image center is represented by the parameters p_x and p_y in the horizontal and vertical directions respectively. The effect of these distortions can be seen in Figure 3 where a sample image of a checkerboard pattern from a wide field of view camera was undistorted using the above correction. In this example, the OpenCV open source computer vision library was used to determine the distortion coefficients and to correct for the lens distortion. The camera calibration routines in OpenCV use different perspectives of a checkerboard pattern to determine the intrinsic parameters for the camera.[25] The parameters provided by the camera calibration are the distortion coefficients k_1, k_2, p_1, and p_2, as well as the focal lengths f_x and f_y and the offset of the image center as given by p_x and p_y.

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:

$$\mathbf{L}_{cb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_c & \sin\phi_c \\ 0 & -\sin\phi_c & \cos\phi_c \end{bmatrix} \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_c & 0 & \cos\theta_c \end{bmatrix} \begin{bmatrix} \cos\psi_c & \sin\psi_c & 0 \\ -\sin\psi_c & \cos\psi_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$\mathbf{L}_{bi} = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{7}$$

$$\mathbf{L}_{ci} = \mathbf{L}_{cb}\mathbf{L}_{bi} \ . \tag{8}$$

$\mathbf{L}_{cb}$ 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), tilt (θ_c), and roll (ϕ_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 Extended Kalman Filter (EKF) is a suboptimal but conventionally used Kalman filtering method for nonlinear process and measurement models (see for example[17]). 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 predict a state estimate. Meanwhile, the correction phase runs at discrete intervals and uses sensor measurements and a linearized process and measurement model to correct the estimate predicted by the process model. The EKF uses the error between the predicted values of the measurement vector and the actual measurements from the image processor 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:

$$\dot{\hat{\mathbf{x}}} = f\left(\hat{\mathbf{x}}(t), t\right) + \omega_n \tag{9}$$

$$\dot{\mathbf{P}} = \mathbf{AP} + \mathbf{PA}^T + \mathbf{Q} \tag{10}$$

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 tunable positive definite matrix representing the intensity of the additive process noise ω_n inherent in the system.

2. Extended Kalman Filter Correction

The EKF makes use of a nonlinear measurement model $h(\hat{\mathbf{x}}^-)$ that takes in the predicted best state estimate ($\mathbf{x}^-$) and computes an expected measurement vector. Minus superscripts here denote predicted values. 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:

$$\mathbf{K} = \mathbf{P}^-\mathbf{C}^T(\mathbf{CP}^-\mathbf{C}^T + \mathbf{R})^{-1} \tag{11}$$

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{K}[\mathbf{z} - h(\hat{\mathbf{x}}^-)] \tag{12}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{KC})\mathbf{P}^- \tag{13}$$

where $\mathbf{K}$ is the Kalman gain, $\mathbf{R}$ is a tunable diagonal matrix representing measurement noise in the sensor, and $\mathbf{C} = \left(\partial h(\hat{\mathbf{x}^-})/\partial \hat{\mathbf{x}}\right)|_{\hat{\mathbf{x}}}$ is the Jacobian of the measurement vector with respect to the state vector. The results from (11) - (13) 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 be received at different measurement rates and 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 different measurements at time t),

$$\mathbf{K}_k^l = \mathbf{P}_k^{l-1}\mathbf{C}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1})\left[\mathbf{C}_k^l(\hat{\mathbf{x}}_k^{l-1})\mathbf{P}_k^{l-1}\mathbf{C}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) + \mathbf{R}_k^l\right] \tag{14}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l\left[\mathbf{z}_k^l - h(\hat{\mathbf{x}}_k^{l-1})\right] \tag{15}$$

$$\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l\mathbf{C}_k^l\right]\mathbf{P}_k^{l-1} \tag{16}$$

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 are solved here using a statistical z-test. The proposed z-test uses the state error covariance matrix, $\mathbf{P}$, and the measurement noise matrix, $\mathbf{R}$, 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

$$Z = \mathbf{e}^T\left(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}\right)^{-1}\mathbf{e} \tag{17}$$

where the residual is defined as

$$\mathbf{e} = \mathbf{z} - h(\hat{\mathbf{x}}) \tag{18}$$

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. This critical value needs to be chosen. Note however that comparing Z values instead of just the norm of the residual (e^Te) is more robust due to the reasons mentioned above. Therefore, the algorithm was found to be fairly insensitive to the choice of a reasonable critical value.

D. Detecting Feature Points

As terminology, the detection of feature points is the determination of feature points in a given frame whereas the tracking of feature points refers to the corresponding of features from frame to frame. Three of the most common methods for detecting feature points are the Kanade-Lucas-Tomasi (KLT) Tracker,[18] Lowe's Scale-Invariant Feature Transform (SIFT),[19] and the Harris corner detector.[20] The KLT and SIFT formulations handle both the detection and tracking of feature points. However, these 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 a second-order moment matrix of the image intensity gradients (denoted $\mathbf{M}$) which is given by

$$\mathbf{M} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \tag{19}$$

In other words, for a given pixel, this matrix 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

$$M_c = \det(\mathbf{M}) - \kappa \left[\mathrm{Tr}(\mathbf{M})\right]^2 \tag{20}$$

which indicates a feature point if the M_c value is greater than a certain threshold value, which needs to be tuned to the image quality and operating environment. 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.

Implementations of a Harris feature detector typically need to do some further selection of the feature points to obtain practical measurements. Since the corner metric is computed over a window of pixels, it is possible for multiple windows around a single corner to all exceed the minimum threshold. In order to get a single reading for each individual corner, a minimum distance is enforced between each point output from the corner detector. Furthermore, it is typically desired that there is a maximum number of corners that are output from the image processor either for memory, processing, or bandwidth limitations. So to ensure that the points output from the corner detector are spread sufficiently throughout the image, the image can be broken up into a grid so that there is a limit on the number of points that can be output for each grid section in the image.

III. Estimator Formulation for Localization and Mapping

A. Estimating Vehicle States Given Landmark Locations

The following are the states that we wish to estimate for the closed-loop control of the aircraft

- 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: $\mathbf{s}_b = (\mathbf{a}_b - \mathbf{g}_b) = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix}^T$
- pixel position for feature point n: $\mathbf{z}_n = \begin{bmatrix} X_n & Y_n \end{bmatrix}^T$

However, to describe the orientation of the aircraft, an attitude error representation is employed to describe the small angle difference between a reference body frame and the true body-fixed frame. This is accomplished by defining an infinitesimal error quaternion $\delta\mathbf{q}$ according to[21]

$$\delta\mathbf{q} \simeq \begin{bmatrix} 1 & \mathbf{R} \end{bmatrix}^T \tag{21}$$

such that

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{22}$$

This allows the 3 dimensional vector $\mathbf{R}$ to be tracked as a minimal representation of the attitude state error. The accelerometer and gyroscope biases of the IMU are also included in the state vector such that $\omega_b = \omega_{b_{raw}} - \mathbf{b}_g$ and $\mathbf{a}_b = \mathbf{a}_{b_{raw}} - \mathbf{b}_a$. Therefore the estimated 15 element state vector is

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{p}}_i & \hat{\mathbf{v}}_i & \hat{\mathbf{R}} & \hat{\mathbf{b}}_a & \hat{\mathbf{b}}_g \end{bmatrix}^T$$

where the attitude quaternion $\hat{\mathbf{q}}$ is tracked as the reference body frame to which the attitude error states are compared.

1. Process Model

The following equations constitute the propagation of the necessary states:

$$\dot{\hat{\mathbf{b}}}_g = \dot{\hat{\mathbf{b}}}_a = 0; \tag{23}$$

$$\dot{\hat{\mathbf{p}}}_i = \hat{\mathbf{v}}_i \tag{24}$$

$$\begin{aligned} \dot{\hat{\mathbf{v}}}_i &= \hat{\mathbf{L}}_{ib}\mathbf{a}_b && (25)\\ &= \hat{\mathbf{L}}_{ib}\left(\mathbf{s}_b + \mathbf{g}_b\right) && (26)\\ &= \hat{\mathbf{L}}_{ib}\mathbf{s}_b + \mathbf{g}_i && (27) \end{aligned}$$

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\hat{\mathbf{q}} \tag{28}$$

The following equations are the non-zero components of the Jacobian matrix $\mathbf{A}$:

$$\frac{\partial\dot{\hat{\mathbf{p}}}_i}{\partial\hat{\mathbf{v}}_i} = \mathbf{I}_{3\times3}, \qquad \frac{\partial\dot{\hat{\mathbf{v}}}_i}{\partial\hat{\mathbf{R}}} = -\hat{\mathbf{L}}_{ib}\widetilde{\mathbf{s}}_b, \qquad \frac{\partial\dot{\hat{\mathbf{v}}}_i}{\partial\hat{\mathbf{b}}_a} = -\hat{\mathbf{L}}_{ib}, \qquad \frac{\partial\dot{\hat{\mathbf{R}}}}{\partial\hat{\mathbf{R}}} = -\widetilde{\omega}_b, \qquad \frac{\partial\dot{\hat{\mathbf{R}}}}{\partial\hat{\mathbf{b}}_g} = -\mathbf{I}_{3\times3} \tag{29}$$

where the tilde symbol denotes a skew symmetric matrix composed of the vector components such that for the vector $\mathbf{a} = [a_1 \;\; a_2 \;\; a_3]$, then $\widetilde{\mathbf{a}}$ is defined by

$$\widetilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{30}$$

2. 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 4 are first introduced.

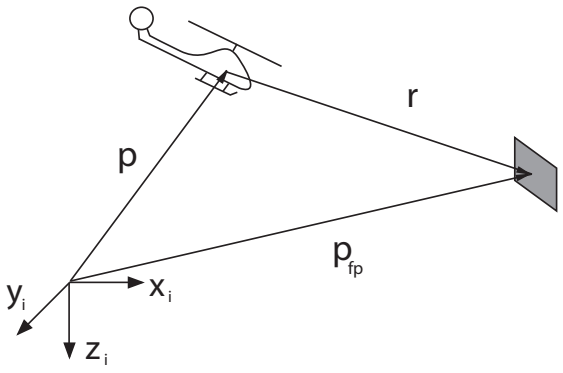

Figure 4. Vectors used in describing the EKF measurement model.

Here $\mathbf{p}$ 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 $\mathbf{r}_i = [X_{fp_i} \;\; Y_{fp_i} \;\; Z_{fp_i}]^T$.

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

$$\hat{X} = f_x \frac{\hat{Y}_{fp_c}}{\hat{X}_{fp_c}} \tag{31}$$

$$\hat{Y} = f_y \frac{\hat{Z}_{fp_c}}{\hat{X}_{fp_c}} \tag{32}$$

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

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{p}}_i} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_i}\right) \tag{33}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \begin{bmatrix} \frac{\partial \hat{X}}{\partial \hat{X}_{fp_c}} & \frac{\partial \hat{X}}{\partial \hat{Y}_{fp_c}} & \frac{\partial \hat{X}}{\partial \hat{Z}_{fp_c}} \\ \frac{\partial \hat{Y}}{\partial \hat{X}_{fp_c}} & \frac{\partial \hat{Y}}{\partial \hat{Y}_{fp_c}} & \frac{\partial \hat{Y}}{\partial \hat{Z}_{fp_c}} \end{bmatrix} \tag{34}$$

$$\frac{\partial \hat{X}}{\partial \hat{X}_{fp_c}} = -f_x \frac{\hat{Y}_{fp_c}}{\hat{X}^2_{fp_c}} = -\frac{\hat{X}}{\hat{X}_{fp_c}}, \qquad \frac{\partial \hat{X}}{\partial \hat{Y}_{fp_c}} = \frac{f_x}{\hat{X}_{fp_c}}, \qquad \frac{\partial \hat{X}}{\partial \hat{Z}_{fp_c}} = 0 \tag{35}$$

$$\frac{\partial \hat{Y}}{\partial \hat{X}_{fp_c}} = -f_y \frac{\hat{Z}_{fp_c}}{\hat{X}^2_{fp_c}} = -\frac{\hat{Y}}{\hat{X}_{fp_c}}, \qquad \frac{\partial \hat{Y}}{\partial \hat{Y}_{fp_c}} = 0, \qquad \frac{\partial \hat{Y}}{\partial \hat{Z}_{fp_c}} = \frac{f_y}{\hat{X}_{fp_c}} \tag{36}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \frac{1}{\hat{X}_{fp_c}} \begin{bmatrix} -\hat{X} & f_x & 0 \\ -\hat{Y} & 0 & f_y \end{bmatrix} \tag{37}$$

$$\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_i} = \frac{\partial \left(\hat{\mathbf{p}}_{fp_c} - \hat{\mathbf{p}}_c\right)}{\partial \hat{\mathbf{p}}_i} = \frac{\partial \hat{\mathbf{p}}_{fp_c}}{\partial \hat{\mathbf{p}}_i} - \frac{\partial \hat{\mathbf{p}}_c}{\partial \hat{\mathbf{p}}_i} = 0 - \frac{\partial \left(\hat{\mathbf{L}}_{ci}\hat{\mathbf{p}}_i\right)}{\partial \hat{\mathbf{p}}_i} = -\hat{\mathbf{L}}_{ci} \tag{38}$$

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

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{R}}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{R}}}\right) = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\frac{\partial \left(\mathbf{L}_{cb}\hat{\mathbf{r}}_b\right)}{\partial \hat{\mathbf{R}}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\mathbf{L}_{cb}\left(\frac{\partial \hat{\mathbf{r}}_b}{\partial \hat{\mathbf{R}}}\right) \tag{39}$$

where

$$\frac{\partial \hat{\mathbf{r}}_b}{\partial \hat{\mathbf{R}}} = \widetilde{\mathbf{r}}_b \tag{40}$$

B. Estimating Landmark Positions Given Vehicle States

In this problem, it is assumed that the position ($\mathbf{p}_i$), velocity ($\mathbf{v}_i$), and attitude ($\mathbf{q}$) 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 ($\mathbf{z}_n = [X_n \;\; Y_n]^T$). Estimates of the landmark positions require an initial guess which is obtained by taking an intersection of the observation ray with the ground plane assumed to be at $0\ ft$ altitude.

1. 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 $\mathbf{A} = 0$ (see equation 9). The process noise in this case is introduced due to possible movement in the image plane of the features due to effects such as unaccounted camera vibrations, or slight movement of the features. This is captured through the positive semidefinite process noise covariance matrix Q. Therefore, the only update that occurs in the prediction phase for this problem is the update of the covariance matrix according to

$$\dot{\mathbf{P}} = \mathbf{Q} \tag{41}$$

The Q matrix is typically chosen to be diagonal, and its values need to be tuned by the designer. Lower values of Q denote higher confidence in the assumption that the feature points are stationary in the image plane.

2. Measurement Model

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

$$\hat{X} = f_x \frac{\hat{Y}_{fp_c}}{\hat{X}_{fp_c}} \tag{42}$$

$$\hat{Y} = f_y \frac{\hat{Z}_{fp_c}}{\hat{X}_{fp_c}} \tag{43}$$

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.

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{p}}_{fp_i}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_{fp_i}}\right) \tag{44}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \frac{1}{\hat{X}_{fp_c}} \begin{bmatrix} -\hat{X} & f_x & 0 \\ -\hat{Y} & 0 & f_y \end{bmatrix} \tag{45}$$

$$\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \left(\hat{\mathbf{p}}_{fp_c} - \mathbf{p}_c\right)}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \hat{\mathbf{p}}_{fp_c}}{\partial \hat{\mathbf{p}}_{fp_i}} - \frac{\partial \mathbf{p}_c}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \left(\mathbf{L}_{ci}\hat{\mathbf{p}}_{fp_i}\right)}{\partial \hat{\mathbf{p}}_{fp_i}} - 0 = \mathbf{L}_{ci} \tag{46}$$

C. Initializing Points Located on Ground Plane

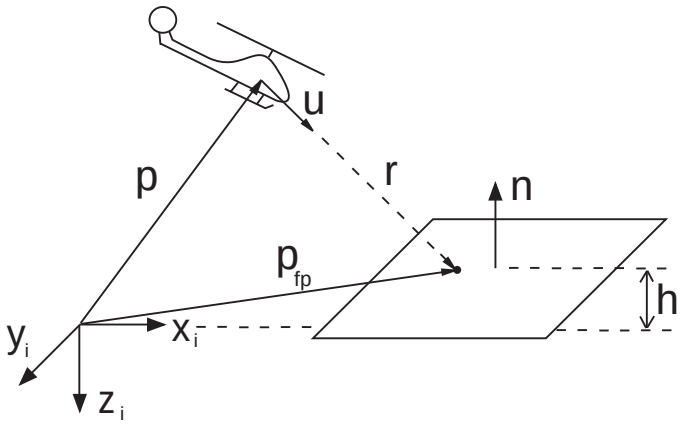

Figure 5. By assuming all landmarks to lie on the ground, points in the mapping database can be initialized by finding the intersection of the first observation ray with the ground plane.

A basic method that can be used to find an initial guess for landmark locations in the EKF is to assume that the points reside on the ground plane. This allows for the intersection of the ground plane with the ray from the first observation to act as the initial guess when the altitude of the aircraft above the ground can be measured using a rangefinder, or approximated with knowledge of the vehicle's altitude. From the first observation of a point, the vector $\mathbf{u}_i = \mathbf{L}_{ic}[1 \; X \; Y]^T$ can be constructed. With this, the relative vector from the vehicle to the feature point can be computed as $\mathbf{r}_i = t\mathbf{u}_i$ where t is the unknown scale factor for the initial observation. To compute the scale factor, if the ground plane is assumed to be at a height h above the ground, then

$$t = -\frac{(h + \mathbf{p}_{i_z})}{\mathbf{r}_{i_z}} \tag{47}$$

D. Initialization of Covariance Matrix for Mapping

An initial value for the covariance of the state estimate for each mapped landmark point is required for the EKF algorithm. This can be expressed as

$$\mathbf{P}_{0_c} = \mathit{Var}(\mathbf{r}_{c_x}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{R_x}{f_x^2} & 0 \\ 0 & 0 & \frac{R_y}{f_y^2} \end{bmatrix} \tag{48}$$

where $\mathit{Var}(\mathbf{r}_{c_x})$ is the variance of the initial range estimate to the landmark point. The initial range estimate is obtained by intersecting the ray given by the point observation with the known ground plane, see Section C. The variance of the range depends on the fixed uncertainty of the height estimate over ground as well as the angle of incidence with the ground plane. The variance of the range was modeled as being proportional to the distance to the landmark to reflect the higher uncertainty for points with smaller angle of incidence. The scaling factor on the range variance term can be tuned as an adjustable parameter. Note however that the initial covariance matrix in (48) is expressed in the camera frame, and since the location of the landmark points are being estimated in the local inertial frame this needs to be transformed into the inertial frame according to

$$\mathbf{P}_{0_l} = \mathbf{L}_{ic}\mathbf{P}_{0_c}\mathbf{L}_{ci} \tag{49}$$

IV. Generic Initialization of Database Points

In a monocular vision system, points are projected from 3D space onto a 2D image plane. With this projection comes the loss of range information between the camera and the landmark making it difficult to initialize the states in the mapping filter since the location of a given landmark in inertial space can only be determined along a ray. In this section we present a generic initialization algorithm that estimates the distance to a point instead of assuming that the point lies on a fixed ground plane. This removes some limitation of previous work,[14] in which points were assumed to lie on the ground. In that work the locations of the points in 3D inertial space were found by intersecting the first observation ray with the ground plane. This method can work well when the landmarks are located close to or on the ground and when the altitude above ground level is known. However it does not work well for initializing points that may lie above the ground such as the walls of buildings.

To overcome this limitation, multiple observations of a landmark from different perspectives can be used to triangulate its location for the initial estimate. The method employed for this was to correspond the measurements between different images using the knowledge of the change in rotation of the aircraft and to use a template match. Once sufficient observations of a landmark have been obtained, then a triangulation using the measurements and the vehicle pose at each point in time can be performed. With sufficient observations we mean that several observations are used to overcome noise, and that a sufficient baseline exists between measurements. The baseline is defined as the distance between observations perpendicular to the viewing direction, and a sufficiently large baseline allows obtaining of a reasonable depth estimate. Here, we perform an explicit initialization of point features once depth is observable, while methods such as the inverse depth parameterization[15] allow initialization with the first measurement. However, without a sufficient baseline, only rotational estimates can be obtained, and in our application those are already provided by the IMU, at least over short time frames.

Measurements can be predicted into the next image using knowledge of the relative attitude change between observations. If the camera calibration matrix is given by

$$\mathbf{\Gamma} = \begin{bmatrix} c_x & f_x & 0 \\ c_y & 0 & f_y \\ 1 & 0 & 0 \end{bmatrix} \tag{50}$$

then the projection of a landmark point onto the image plane can be expressed using homogeneous coordinates. With homogenous coordinates, the coordinate triple (u, v, w) is equivalent to the 2D point $(u/w, v/w)$ and the coordinate (x, y, z, t) is equivalent to the 3D point $(x/t, y/t, z/t)$. With this representation, the last coordinate essentially acts as a divisor factor that also allows for points to lie at infinity if the divisor factor is 0. Denoting a homogeneous 2D image location of a feature point as $\bar{\mathbf{z}}$ and the homogeneous 3D inertial location of a landmark as $\bar{\mathbf{p}}_{fp_i}$, and by defining the camera matrix $\mathbf{\Phi}$ as

$$\mathbf{\Phi} = \mathbf{\Gamma}\left[\mathbf{L}_{ci} \mid -\mathbf{p}_c\right] \tag{51}$$

the projection of a point onto the image plane can be denoted as

$$\bar{\mathbf{z}} = \mathbf{\Phi}\bar{\mathbf{p}}_{fp_i} \tag{52}$$

An infinite homography can be used to predict the location of a feature point in the next image frame by assuming that the measurement lies on the plane out at infinity.[22] If the first observation is denoted as being in the c_1 frame, and the second observation as being in the c_2 frame, then the infinite homography is such that

$$\bar{\mathbf{z}}_{c_2} = \mathbf{H}_\infty \bar{\mathbf{z}}_{c_1} \tag{53}$$

where $\mathbf{z}_{c_1}$ is the measurement in the c_1 camera, $\mathbf{z}_{c_2}$ is the measurement as seen from the c_2 camera, and the homography matrix $\mathbf{H}_\infty$ is given by

$$\mathbf{H}_\infty = \mathbf{\Gamma}\mathbf{L}_{c_2c_1}\mathbf{\Gamma}^{-1} \tag{54}$$

The matrix $\mathbf{L}_{c_2c_1}$ is a relative rotation matrix from the orientation of the c_1 camera frame to the orientation of the c_2 camera frame so $\mathbf{L}_{c_2c_1} = \mathbf{L}_{c_2i}\mathbf{L}_{ic_1}$. This relation allows measurements to be predicted in the next image using only the relative rotation of the vehicle between the two camera poses, which could potentially be provided by gyroscopes, a magnetometer, or other attitude sensing device if a full state estimator is not available.

The direct linear transformation (DLT) algorithm can be used for triangulating the location of a feature point in 3D space from multiple observations using a linear least-squares minimization.[22] This is formulated by starting out with equation (52) which states that the image projection of a landmark point can be modeled as $\bar{\mathbf{z}} = \mathbf{\Phi}\bar{\mathbf{p}}_{fp_i}$. The homogeneous scale factor can first be removed from this equation using a cross product. For a single image measurement, this yields

$$\begin{bmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}^1\bar{\mathbf{p}}_{fp_i} \\ \mathbf{\Phi}^2\bar{\mathbf{p}}_{fp_i} \\ \mathbf{\Phi}^3\bar{\mathbf{p}}_{fp_i} \end{bmatrix} \tag{55}$$

where $\mathbf{\Phi}^i$ is the ith row of the camera matrix $\mathbf{\Phi}$. This can be rewritten as

$$x(\mathbf{\Phi}^3\bar{\mathbf{p}}_{fp_i}) - (\mathbf{\Phi}^1\bar{\mathbf{p}}_{fp_i}) = 0 \tag{56}$$

$$y(\mathbf{\Phi}^3\bar{\mathbf{p}}_{fp_i}) - (\mathbf{\Phi}^2\bar{\mathbf{p}}_{fp_i}) = 0 \tag{57}$$

$$x(\mathbf{\Phi}^2\bar{\mathbf{p}}_{fp_i}) - y(\mathbf{\Phi}^1\bar{\mathbf{p}}_{fp_i}) = 0 \tag{58}$$

However, the third equation can be dropped since it is just a linear combination of the first two. The equations for n different observations of the same landmark are then written in the form $\mathbf{A}_{dlt}\bar{\mathbf{p}}_{fp_i} = \mathbf{0}$, where $\mathbf{A}_{dlt} \in \mathbb{R}^{2n\times 4}$ is given by

$$\mathbf{A}_{dlt} = \begin{bmatrix} x_1\mathbf{\Phi}_1^3 - \mathbf{\Phi}_1^1 \\ y_1\mathbf{\Phi}_1^3 - \mathbf{\Phi}_1^2 \\ \vdots \\ x_n\mathbf{\Phi}_n^3 - \mathbf{\Phi}_n^1 \\ y_n\mathbf{\Phi}_n^3 - \mathbf{\Phi}_n^2 \end{bmatrix} \tag{59}$$

This can be solved by finding the $\bar{\mathbf{p}}_{fp_i}$ that minimizes $||\mathbf{A}_{dlt}\bar{\mathbf{p}}_{fp_i}||$ subject to $||\bar{\mathbf{p}}_{fp_i}|| = 1$. By the Singular Value Decomposition (SVD), it can be written that $\mathbf{A}_{dlt} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where $\mathbf{U} \in \mathbb{R}^{2n\times 2n}$ and $\mathbf{V} \in \mathbb{R}^{4\times 4}$ are orthogonal matrices, and $\mathbf{D} \in \mathbb{R}^{2n\times 4}$ is a diagonal matrix that contains the singular values of $\mathbf{A}_{dlt}$ in decreasing order. The goal is to minimize $||\mathbf{U}\mathbf{D}\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}|| = ||\mathbf{D}\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}||$ since $\mathbf{U}$ is orthogonal. Furthermore, it is also known that $||\bar{\mathbf{p}}_{fp_i}|| = ||\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}||$ since $\mathbf{V}$ is orthogonal. This means that the problem now becomes to minimize $||\mathbf{D}\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}||$ subject to $||\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}|| = 1$. A variable substitution can be made by letting $\mathbf{y} = \mathbf{V}^T\bar{\mathbf{p}}_{fp_i}$. This means that the problem can now be formulated as minimizing $||\mathbf{D}\mathbf{y}||$ such that $||\mathbf{y}|| = 1$. Since the singular values in the diagonal $\mathbf{D}$ matrix are sorted in decreasing order, the solution to this problem is for $\mathbf{y}$ to have its last component as 1, and all other components as 0. Solving for $\bar{\mathbf{p}}_{fp_i}$ gives that $\bar{\mathbf{p}}_{fp_i} = \mathbf{V}\mathbf{y} = \mathbf{V}[0\ 0\ 0\ 1]^T$ which means that the minimization problem is solved by having $\bar{\mathbf{p}}_{fp_i}$ equal to the last column of $\mathbf{V}$. Therefore, by solving the SVD of the $\mathbf{A}_{dlt}$ matrix, the last column of the $\mathbf{V}$ matrix provides the triangulated homogenous coordinates of the landmark in inertial space. The data used in the DLT algorithm needs to be normalized in order to improve the conditioning of the $\mathbf{A}_{dlt}$ matrix.[22] The data can be normalized by constructing a normalization matrix operator T such that the collection of the normalized 2-D measurements of landmark locations has its centroid at the coordinate origin $(0, 0)$ and average distance of $\sqrt{2}$ from

the origin. Similarly, normalization needs to be also performed on the position estimates such that the collection of 3-D points has its origin at $(0, 0, 0)$ and an average distance of $\sqrt{3}$ from the origin.[23]

The algorithm for the initialization is outlined here:

1. Collect all the measurements that were not corresponded from the statistical z-test method.
2. For all points currently stored in the initialization database, predict these points into the current frame using the infinite homography relation (54).
3. For each of the unmatched measurements, compare it to the list of predicted points from the last image, and find the N nearest neighbors within a box around each measurement.
4. Take the N nearest neighbors from 3 and perform a template match with the point in the initialization database. The template match score is the sum-square error of the image intensities from a square window around the feature point. The minimum score signifies a best match.
5. Perform 4, and repeat for all unmatched measurements until each measurement has been corresponded with the best match.
6. Points for which a match has been found can be used to update the initialization database, whereas measurements that were not matched can be added as new entries into the initialization database.
7. When an entry in the database has M measurements associated with it, then apply the DLT algorithm to the feature track for this landmark point. Compute the sample covariance of the measurements in the feature track with respect to the measurement expected from the computed point (using knowledge of the vehicle pose associated with each measurement in the feature track) according to

$$R_{init} = \sum_{j=1}^{M} \frac{(x_j - x_{j_{exp}})^2 + (y_j - y_{j_{exp}})^2}{M} \tag{60}$$

   and add the computed point to the mapping database if $R_{init} <= R_{max}$. Otherwise the point is rejected and the slot in the initialization database can be freed for another point to be tracked for initialization.
8. If a point in the initialization database has not had a new measurement associated with it within the last F frames, then remove the point from the initialization database.

Figure 6 shows a simulation run that was performed to validate the performance of the DLT initialization algorithm. The blue lines show the progression of the feature track over time through the image frame. Targets on the ground have a length of 12 ft and a width of 9 ft with their centers located at $[0, -10, 0]$ ft and $[0, 10, 0]$ ft in North, East, down coordinates. This was performed using the same simulation setup and flight path as was used to verify the mapping estimator. After being initialized, the estimates of the point continued to converge as before using the mapping estimator. Table 1 shows the initialized landmark coordinates in simulation.

Table 1. Initialized landmark coordinates in simulation using DLT.

[ft]	N	E	D	N	E	D
Actual	**6**	**-14.5**	**0**	**6**	**-5.5**	**0**
Estimated	6.77	-13.15	0.45	7.12	-6.20	-1.09
Actual	**6**	**5.5**	**0**	**6**	**14.5**	**0**
Estimated	6.38	8.33	1.95	5.53	17.86	2.70
Actual	**-6**	**-14.5**	**0**	**-6**	**-5.5**	**0**
Estimated	-7.87	-13.54	-1.89	-6.56	-2.28	2.56
Actual	**-6**	**5.5**	**0**	**-6**	**14.5**	**0**
Estimated	-5.94	9.94	4.48	-5.99	18.39	2.51

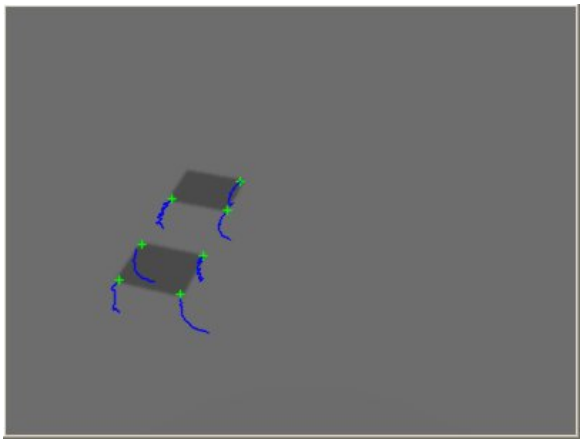

Figure 6. Visualization for the feature point tracks when initializing the 3D inertial location of landmarks using multiple observations for triangulation. The blue lines show the locations of the last 20 measurements for the associated feature point.

Using synthetically generated scenes as shown in Figure 6 resulted in frequent mismatches during the nearest neighbor template-matching portion of the algorithm due to the uniformity of the pixel intensities and lack of texture. However, these mismatches did not significantly affect the performance of the initialization method since these points were either rejected by the sample covariance test, or they resulted in an error in the initial guess of the states in the mapping filter which could later be corrected by the estimator. Testing with actual images and sensor data obtained from flight onboard a rotorcraft UAV, as shown in the sample feature tracks in Figure 7, suggested that images obtained from actual cameras are less prone to the occurrence of such mismatches. To date, this initialization method has been tested in simulation and offline using recorded data from actual flight tests, but not in real-time on actual flight hardware.

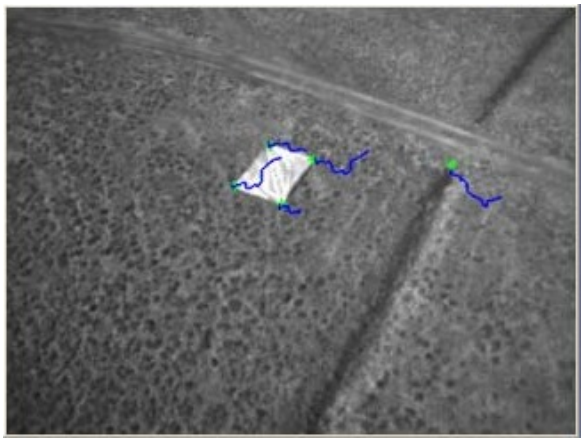

Figure 7. Visualization for the feature point tracks when initializing the 3D inertial location of landmarks using multiple observations for triangulation. The blue lines show the locations of the last 20 measurements for the associated feature point.

V. Simulation and Flight Test Results

A loss-of-GPS scenario was used to demonstrate the proposed architecture in both simulation and flight test. The scenario involved having a rotorcraft UAV flying around a target placed on the ground. While flying around the target, the rotorcraft has knowledge of its position and attitude by means of a baseline GPS-aided inertial navigation system (INS). In this initial stage of the scenario, the mapping estimator is running to figure out the locations of the landmarks in 3D inertial space. The method outlined in Section III.C is used for forming an initial guess for the position of landmarks. When the landmarks have been sufficiently mapped out, then the vehicle is commanded to a stop and the mapping estimator is disabled. With the camera looking at the target, the feature point measurements

are then incorporated into the EKF navigation filter. GPS is then ignored in the navigation filter so that the vehicle is performing autonomous closed-loop control using only IMU and the vision sensor. This initial scenario demonstrates the ability of the helicopter to fly around and map out landmarks that it can in turn use to figure out its own position and attitude should GPS be lost.

A. Simulation Results

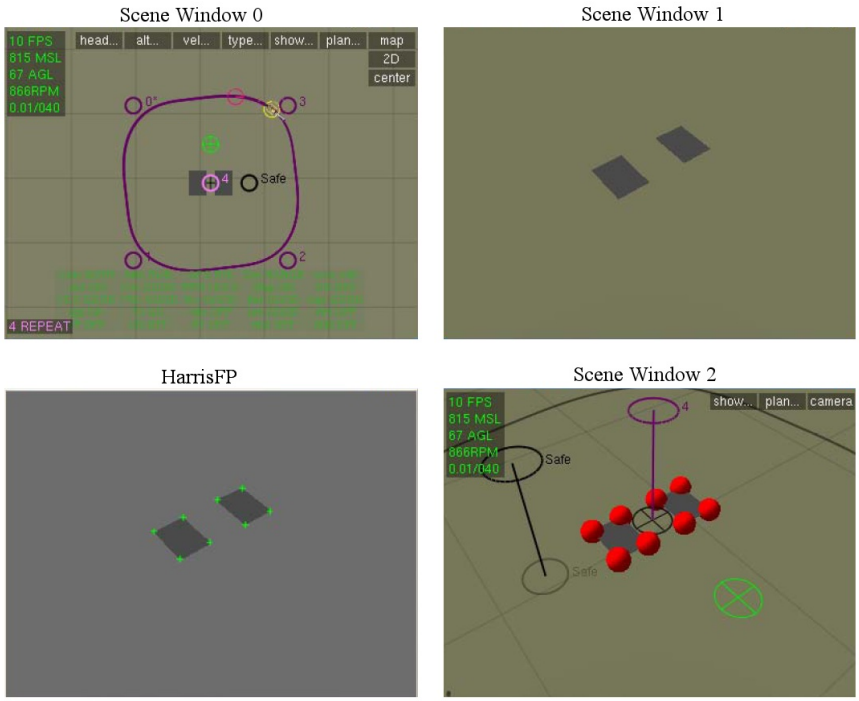

Figure 8. The simulation environment used for testing the loss-of-GPS scenario. "Scene Window 0" shows the trajectory of the helicopter as it flies around the target points. "Scene Window 1" shows the image viewed by the simulated camera. The "harrisFP" window shows the image processing results of the corner detector. The output of the estimated locations are replicated in the simulated camera image as red spheres in "Scene Window 2".

Figure 8 shows a simulation environment and results that demonstrate the functionality of the framework. In this sample simulation run, a rotorcraft UAV is circling around some objects of interest located on the ground using a GPS-aided INS. A monocular vision camera is looking out the left side of the aircraft and downwards at a $45\ degree$ angle. As it circles these objects, it is using the measurements from the monocular camera as well as the knowledge of its position and attitude as given by the GPS-aided INS to map out where the landmarks are located in 3D space. The estimates of the landmark locations are represented by the red spheres in the scene windows. After these points have been sufficiently mapped out, the vehicle is put into a stationary hover, and then GPS readings are ignored so that the vision sensor takes the place of the GPS in the navigation filter. The vehicle is able to maintain a stable closed-loop hover using the vision-aided INS. The helicopter also maneuvers in different directions to demonstrate the transient response of the system. The plot in Figure 9 compares the output from the vision-based estimator with the commanded position as well as the true position as given by the dynamic model of the simulation. It should also be noted that the state output of the vision-based estimator is used in the closed loop control of the aircraft in this simulation test. Figure 10 shows the error between the estimated and actual positions of the helicopter in simulation.

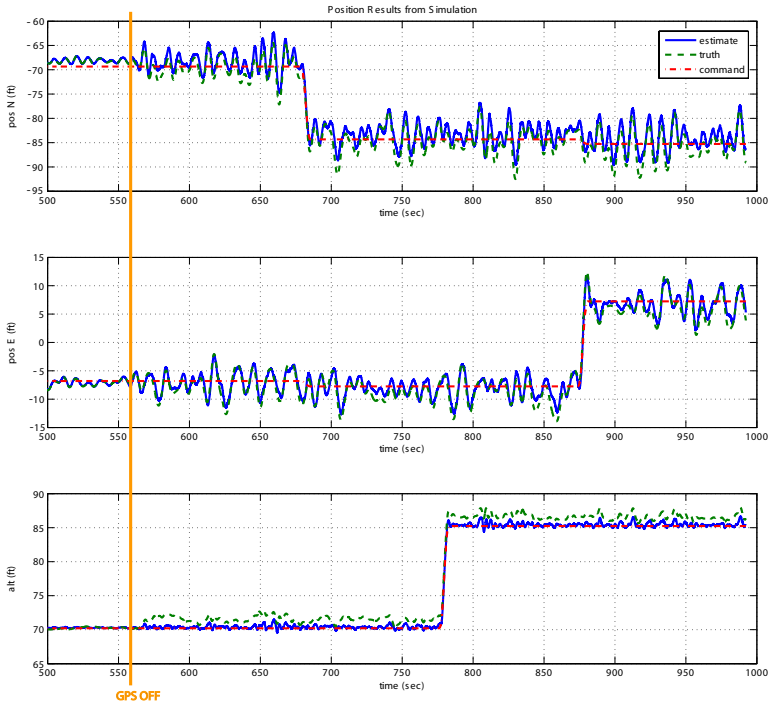

Figure 9. The estimated, actual, and commanded positions for the aircraft during a loss-of-GPS scenario expressed in North, East, and altitude coordinates during the simulation run. At t = 555 sec, the GPS is ignored from the navigation solution so that only the IMU and monocular vision sensor are being used for determining the location of the helicopter for closed-loop control.

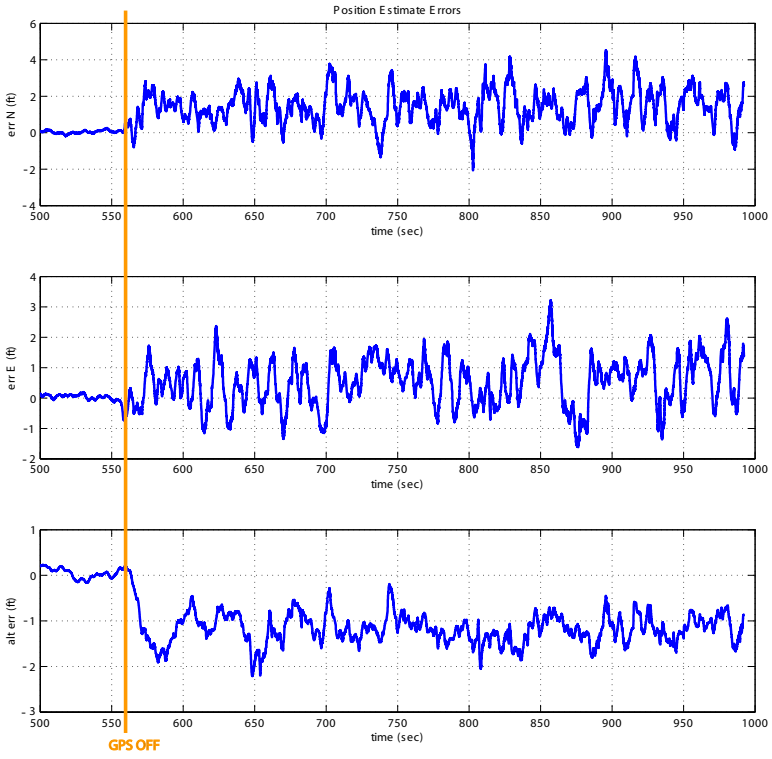

Figure 10. The error between the estimated and actual position of the helicopter during the simulation run.

B. Flight Test Results

Figure 11. The GTMax rotorcraft UAV is a modified Yamaha R-Max helicopter equipped with custom avionics. For testing the vision-based algorithms, a machine vision progressive scan camera is used for acquiring images. Two onboard computers divide up the tasks of image processing and estimation.

The vision-based mapping and localization algorithms were also verified in real-time onboard a rotorcraft UAV during a flight test to demonstrate feasibility of the proposed architecture. Georgia Tech's GTMax UAV was used for these tests and is shown in Figure 11. The GTMax is a modified Yamaha R-Max helicopter UAV that is equipped with dual flight computers, an IMU, and a differential GPS receiver. An adaptive neural-network feedback controller uses EKF state estimates, obtained from the sensor array, to guide the helicopter appropriately.[24] The GTMax is under $200\ lbs$ in weight and has a $6\ ft$ rotor diameter. For these tests, an Ethernet progressive scan camera designed for machine vision is also included as an onboard sensor as shown in Figure 11. The use of a progressive scan camera helps to eliminate any potential problems that may arise from the interlaced images of lower quality video cameras. This camera is mounted on the nose of the aircraft pointing towards the front and angled downwards at a $45\ degree$ angle. The camera provides monochrome images at a resolution of 320×240 pixels and a rate of $20\ Hz$. Furthermore, the GTMax has two onboard computers, so that the computation can be divided by having one computer perform the image processing while the other computer uses the outputs from the feature detector to perform the actual estimation.

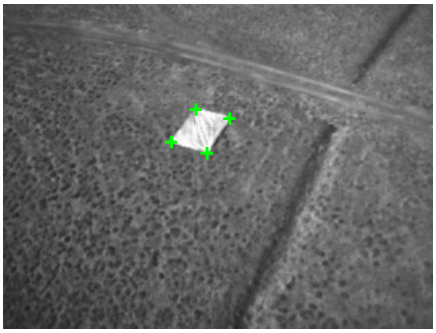

Figure 12. The view from the camera onboard the helicopter with the image processing results overlayed on top of the monochrome image. The target is a 12×9 ft rectangle laid out flat on the ground.

A similar scenario to the simulation setup was performed with the GTMax in flight. A single $12 \times 9\ ft$ silver rectangle was placed on the ground to act as a target and provide 4 relatively consistent landmarks. The helicopter flies in a circular pattern around the target at an altitude of $90\ ft$. Once the points have been sufficiently mapped out, then the helicopter is commanded to a stop, and the vision-based estimator is switched from mapping mode to navigation mode. The GPS is then ignored from the EKF so that the helicopter is using only the IMU and vision sensor for determining the states needed for closed-loop control. Figure 13 shows the estimated and commanded positions of the aircraft compared to the position provided by the GPS during this test. A $5\ ft$ translational motion is also commanded to the aircraft while under vision and IMU only control.

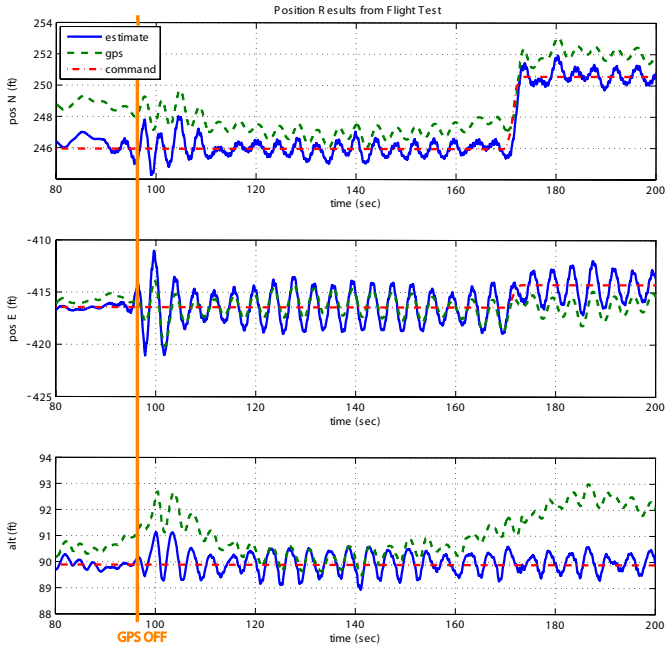

Figure 13. The error between the estimated and actual positions of the helicopter during flight test.

VI. Conclusion

A framework for including a monocular vision sensor into a GPS-aided INS for assisting in state estimation of the aircraft should GPS be lost was presented. When GPS is available, the aircraft can use feature point observations of landmarks in the surrounding environment to map out where these landmarks are located in 3D inertial space. Once these points have been sufficiently mapped (as indicated by the covariance of the estimates), then the monocular vision observations of the landmarks can in turn be used to determine the pose of the aircraft in the absence of GPS. A method to initialize a database consisting of feature points to be mapped was presented. The method used multiple observation of a landmark from different perspectives to form a good initial position guess for points that do not lie on the ground plane. Simulation and real-time flight test results demonstrating autonomous closed-loop control of an aircraft in a loss-of-GPS scenario using only vision and IMU were presented to demonstrate feasibility of the method.

It should be noted that this method requires known features to be in the image for enabling GPS - denied operation. However, given that the inertial solution would drift when GPS is not available, the ability of the algorithm to provide a sufficiently reliable solution when GPS is denied but when features are visible is an improvement. Therefore, this work provides a technique to allow safe autonomous flight when GPS is temporarily unavailable. Furthermore, this work establishes the feasibility of using monocular vision based navigation solutions onboard resource-constrained rotorcraft for GPS-denied flight and forms a foundation for ongoing work in combining the problem of mapping and localization for closed-loop flight control.

References

[1]Christophersen, H. B. and Pickell, W. R. and Neidoefer, J. C. and Koller, A. A. and Kannan, S. K. and Johnson, E. N, " A compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles", *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, May 2006.

[2]Wendel, J. and Metzger, J. and Trommer, G. F., "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration", *Proceedings of AIAA GNC*, San Francisco, CA, August 2005.

[3]Kayton, M. and Fried, W. R., *Avionics Navigation Systems*, John Wiley and Sons, 1997.

[4]M. Achtelik and A. Bachrach and R. He and S. Prentice and N. Roy, " Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments", *Proc. of the 1st Symposium on Indoor Flight*, Mayaguez, PR, 2009.

[5]Pears, N. E., "Feature extraction and tracking for scanning range sensors", *Robotics and Autonomous Systems*, Vol. 33, no.1, pp 43 - 58, 2000.

[6]Ahrens, S. and Levine, D. and Andrews, G. and How, J.P.; , "Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments", *Robotics and Automation, IEEE International Conference on*, pp.2643-2648, 12-17 May 2009, doi: 10.1109/ROBOT.2009.5152680.

[7]Conte, G. and Doherty, P. "Vision-based unmanned aerial vehicle navigation using geo-referenced information". *EURASIP J. Adv. Signal Process*, Article 10 (January 2009), 18 pages. DOI=10.1155/2009/387308

[8]Blösch, M. and Weiss, S. and Scaramuzza, D. and Siegwart, R., "Vision based MAV navigation in unknown and unstructured environments", R*obotics and Automation (ICRA), 2010 IEEE International Conference on*, pp.21-28, 3-7 May 2010 doi: 10.1109/ROBOT.2010.5509920

[9]Amidi, O., Kanade, T., and Fujita, K., "A Visual Odometer for Autonomous Helicopter Flight," *Robotics and Autonomous Systems*, Vol. 29, pp. 185-193, 1999.

[10]Langelaan, J.W., "State Estimation for Autonomous Flight in Cluttered Environments," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, pp. 1414-1426, Sep-Oct 2007.

[11]Madison, R., Andrews, G., DeBitetto, P., Rasmussen, S., and Bottkol, M., "Vision-Aided Navigation for Small UAVs in GPS-Challenged Environments", *AIAA Infotech@Aerospace Conference*, 2007.

[12]Fowers, S.G., Tippetts, B.J., Lee, D.J., and Archibald, J.K., "Vision-Guided Autonomous Quad-rotor Helicopter Flight Stabilization and Control," *AUVSI Unmanned Systems North America Conference*, 2008.

[13]Frietsch, N., Meister, O., Schlaile, C., Seibold J., and Trommer, G.F., "Vision Based Hovering, and Landing System for a VTOL-MAV with Geo-Localization Capabilities," *AIAA Guidance, Navigation, and Control Conference*, 2008.

[14]Wu, A.D., Johnson, E.N., "Methods for Localization and Mapping Using Vision and Inertial Sensors," *AIAA Guidance, Navigation, and Control Conference*, Honolulu, HI, August 2008.

[15]Montiel, J.M.M., Civera, J, and Davison A.J., "Unified Inverse Depth Parametrization for Monocular SLAM," *Robotics: Sciene and Systems (RSS)*, Philadelphia, PA, August 2006.

[16]Brown, D.C., "Close-Range Camera Calibration," *Photogrammetric Engineering*, Vol. 37, No. 8, pp. 855-866, 1971.

[17]Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

[18]Kanade, T., Amidi, O., and Ke, Q., "Real-Time and 3D Vision for Autonomous Small and Micro Air Vehicles", *IEEE Conference on Decision and Control*, 2004.

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

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

[21]Lefferts, E.J., Markley, F.L., and Shuster, M.D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sep - Oct 1982.

[22]Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, March 2004.

[23]Wu, A. D., *Vision-Based Navigation and Mapping for Flight in GPS-Denied Environments*, Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, December 2010.

[24]Johnson,E. and Kannan,S., "Adaptive Trajectory Control for Autonomous Helicopters", *Journal of Guidance Control and Dynamics*, Vol. 28, No. 3, pp 524-538, May 2005.

[25]Bradski, G., "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, citeulike-article-id 2236121, 2000.