GPS-Denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft

Girish Chowdhary, Eric N. Johnson, Daniel Magree,

Allen Wu, Andy Shein [*]

Abstract

GPS-denied closed-loop autonomous control of unstable Unmanned Aerial Vehicles (UAVs) such as rotorcraft using information from a monocular camera has been an open problem. Most proposed Vision aided Inertial Navigation Systems (V-INS) have been too computationally intensive or do not have sufficient integrity for closed-loop flight. We provide an affirmative answer to the question of whether V-INS can be used to sustain prolonged real-world GPS-denied flight by presenting a V-INS that is validated through autonomous flight-tests over prolonged closed-loop dynamic operation in both indoor and outdoor GPS-denied environments with two rotorcraft UAS. The architecture efficiently combines visual feature information from a monocular camera with measurements from inertial sensors. Inertial measurements are used to predict frame-to-frame transition of online selected feature locations, and the difference between predicted and observed feature locations is used to bind in real-time the inertial measurement unit drift, estimate its bias, and account for initial misalignment errors. A novel algorithm to manage a library of features online is presented that can add or remove features based on a measure of relative confidence in each feature location. The resulting V-INS is sufficiently efficient and reliable to enable real-time implementation on resource constrained aerial vehicles. The presented algorithms are validated on multiple platforms in real-world conditions: through a 16 minute flight test, including an autonomous landing, of a 66 kg rotorcraft UAV operating in an unconctrolled outdoor environment without using GPS and through a Micro-UAV operating in a cluttered, unmapped,

[*] girish.chowdhary@gatech.edu, dmagree@gatech.edu, eric.johnson@ae.gatech.edu. G. Chowdhary, A. Wu, E. Johnson, D. Magree, are affiliated with Georgia Institute of Technology, Atlanta, GA. A. Shein is affiliated with CyPhy Works Inc., Danvers MA.

and gusty indoor environment.

1 Introduction

Unmanned Aerial Vehicles (UAVs) and Micro UAVs (M-UAVs) have enjoyed several successes in the past decade. The role of UAVs in the modern air-force and civilian disaster management and homeland security applications is expected to grow. In order to enable closed-loop feedback control for autonomous operation, UAVs need an accurate estimate of their position and attitude with respect to the surrounding environment. Traditionally, this has been reliably enabled through a sensor fusion architecture that integrates position information from a global positioning system (GPS) receiver and acceleration and angular rate measurements from inertial measurement units (IMUs) (see e.g. (Kayton and Fried, 1997)). In this inertial navigation system (INS) architecture, 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 fused with the solution to bound the errors that result from misalignment and bias errors of the IMU (see for example (Christophersen et al., 2006; Wendel et al., 2006; Kayton and Fried, 1997)). The GPS position update is critical to the functioning of such an architecture. However, GPS may not be available indoors and can suffer from obstruction and multipath errors and is also prone to the possibility of being jammed. Therefore, the problem of GPS-denied navigation is of great interest for M-UAVs that operate indoors and for outdoor UAVs that may need to operate in an environment where GPS may be jammed or unavailable (such as under bridges, in urban jungles, or in riverine environments (Scherer et al., 2012)).

Several researchers have been tackling the problem of GPS-denied flight using monocular vision and inertial measurements (see Section 1.1). The risks and challenges in this field of robotics come from the need to develop a reliable set of algorithms that generate a high integrity navigation solution that can be used to close the control loop to enable autonomous flight. In general, there have been three key open questions:

Q1 How to update an online library of reference visual features to enable inclusion/removal of features online to accommodate rapidly changing visual scenes in closed-loop autonomous flight without compromising the integrity of the navigation solution?

Q2 Is it possible to create a sufficiently fast and high integrity monocular camera aided navigation solution to enable closed-loop autonomous flight, including autonomous landing, of resource-constrained unstable rotorcraft UAVs using off-the-shelf hardware?

Q3 Is it possible to use monocular vision-based navigation for indoor navigation for autonomous flight of an unstable M-UAV with fast dynamics and with a controller in the loop?

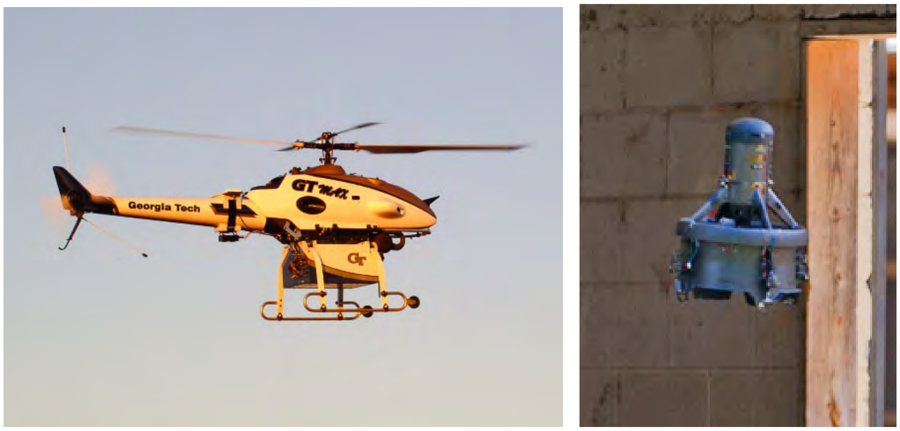
The main contribution of this paper is to answer the above three open questions. In particular, it contributes to answering Q1 by presenting a monocular vision aided inertial navigation system for providing accurate position and attitude estimates in GPS-denied environment which does not require a pre-mapped feature database. Rather, a monocular vision aided INS is presented in which feature locations are simultaneously estimated and tracked to provide position, velocity, and attitude estimates for real-time closed-loop control of unstable UAV platforms performing real-world missions, including waypoint flight in unmapped environments and landing on an unknown terrain. A Harris corner detector extracts feature points from each captured image, and a Mahalanobis distance is used to correspond features from frame-to-frame. The main contribution here is the real-time reference feature management algorithm. In particular, extracted features are stored in an online managed database of feature locations which ranks the features based on how often their predicted locations match measured location. Features that are not seen are dropped and replaced with new features, and the threshold at which features are dropped is dynamically changed. This enables the UAV to navigate while autonomously flying between waypoints in a completely GPS-denied unmapped dynamic environment without knowing feature locations *a priori*.

The paper also provides answers to Q2 and Q3 through extensive testing of the presented methods on real-world platforms operating in their intended domains. In order to stress the vision aided navigation system in real-world GPS-denied situations, multiple test runs have been performed on two very different platforms in their expected operating environments. The first one being a medium sized UAV (66 kg, see Figure 1), designed to operate in unmapped outdoor environments, and the second one being a small sized (Micro) M-UAV (1 kg, see Figure 1) designed to operate in cluttered and confined unmapped indoor and outdoor environments. The data presented from an autonomous landing of the 66 kg rotorcraft is to our knowledge the first demonstration of GPS-denied autonomous landing with an entirely self contained vision-based navigation system for a vehicle of this class.

The work presented here is on a path towards enabling seamless transition of M-UAVs from indoor to outdoor environments without relying on GPS (see Q3). This is a challenging problem, as the lighting and visual feature-structures in indoor and outdoor environments vary greatly, and it may be difficult for the integrated hardware or software to rapidly accommodate such changes. To our knowledge, there are currently no UAVs capable of performing such a transition without the aid of GPS. Therefore, as a first step to enable such transitions, the presented system is validated in closed-loop autonomous waypoint following flight and landing of unstable UAV platforms in both indoor and outdoor GPS-denied environments.

1.1 Related Work

Several authors have tackled the problem of navigation and control in GPS-denied environments. Researchers have investigated scanning range sensors for mapping the surrounding environment and use this information for navigation (see for example (Achtelik et al., 2009a; Pears, 2000; Chowdhary et al., 2011a; Chowdhary et al., 2011b; Achtelik et al., 2009b; Grzonka and Burgard, ; Shen and Kumar, ; Bachrach et al., 2011; Grzonka et al., 2012)). 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 a tremendous potential for localization and mapping applications since they provide data about features such as landmarks, discontinuities (corners), and patterns in the surrounding environment, which can be used to infer information about vehicle motion. Furthermore, a camera is a standard payload on any surveillance and reconnaissance UAV. However, with the exception of a few (notably (Ahrens et al., 2009; Conte and Doherty, 2009; Blösch et al., 2010; Strelow and Singh, 2004; Weiss et al., 2011; Achtelik et al., 2011; Weiss et al., 2012)), the majority of results presented in these areas have been applied to ground robots where size and payload considerations are often not a limitation, and odometry may be used. These algorithms for extracting information from the 2D images of an image sensor are often computationally intensive, and therefore, may not be feasible for M-UAV applications. On the other hand, a navigation system that integrates an IMU with a vision system can alleviate the computational burden by allowing frame to frame



(a) (b)

Figure 1: The GTMax helicopter (a) weighs 66 kg and has a rotor diameter of 3 m. For vision-based navigation, a downward-facing camera is mounted on the nose of the vehicle. The EASE (b) weighs 1 kg and has a duct diameter of 25 cm. The vehicle is designed for indoor and outdoor use. The camera is mounted to the rotor axel bearing underneath the vehicle.

prediction of camera motion, help in resolving scale ambiguity, and improve overall accuracy. An integrated vision-aided INS (V-INS) navigation system is passive, and hence not easily prone to jamming or detection. Due to these reasons, vision aided INS systems have received a great deal of attention for closed-loop control of UAVs.

The use of vision sensors for simultaneous localization and mapping (SLAM) is a very active research area, and several paradigms have emerged in recent years: Keyframe-based SLAM (Klein and Murray, 2007), EKF-SLAM (Davison, 2003; Davison et al., 2007), and particle filter SLAM (Eade and Drummond, 2006). The algorithms presented in these papers each provide a method of estimating the pose of a monocular camera and the location of features in the environment, and do so in real time. Additional extensions to these paradigms improve robustness and scalability, such as inverse depth parametrization presented in Civera et al. (Civera et al., 2008), and relative bundle adjustment presented in Sibley (Sibley et al., 2010). The results in these papers present open-loop data from cameras carried or driven by an operator, in particular, the SLAM provided state estimates are not used for closed-loop control on vehicles with fast dynamics. Additionally, the focus of (Klein and Murray, 2007) and (Eade and Drummond, 2006) was to highlight SLAM techniques for mapping, therefore the place a priority on the accuracy, correctness, and the size of the map. Dille et al. (Dille et al., 2009) and Kitt et al. (Kitt et al., 2011) showed that visual odometery techniques can result in navigation solutions comparable to high-end GPS-INS systems on ground robots, establishing the feasibility of this approach for ground robotics.

1.1.1 Monocular vision aided inertial navigation for GPS-denied operation of aerial platforms

The problem of UAV vision-aided navigation in aerial robotics, however, requires that the system have different priorities. Aerial vehicles have fast dynamics, are resource constrained, and rotorcraft/VTOL UAVs are unstable in position and/or attitude. Therefore, closed-loop control requires accurate and fast vehicle state estimation. Furthermore, since cameras are often rigidly mounted to the vehicle, errors in the navigation algorithm output can be exacerbated by the movement of the vehicle in response to the erroneous output, leading to eventual instability and loss of vehicle. This results in a deep potentially de-stabilizing coupling between the navigation and control problem, in which a poor navigation solution can rapidly destabilize the entire system. This coupling makes it desirable to incorporate inertial measurements from the IMU rather than relying on purely camera based monocular SLAM, such as MonoSLAM (Davison et al., 2007). Additionally, UAVs typically have strict payload and power constraints that limit the computational power available onboard, limiting the class of vision-based navigation techniques which can be practically applied. In the presence of these limitations and given the current state of available computational power onboard UAVs, the best course of action seems to be to first reliably solve the problem of closed-loop control of UAVs with vision aided inertial navigation before delving deeper into improving the global accuracy of the SLAM built map using advanced SLAM techniques. Therefore, the problem of vision-aided inertial navigation for closed-loop control is seen as a special subset of the overall vision-based SLAM problem.

Because of the difficulty of simultaneously mapping features while providing accurate state feedback to the vehicle controller, these steps have been separated in the past. This method of first-mapping-then-localizing has useful application for vehicles needing to maintain stability during periodic GPS loss, however does not easily generalize to situations where both these tasks need to be done simultaneously. Madison et al. presented an Extended Kalman Filter (EKF) based design where the vehicle states were estimated along with the inertial locations of features in 3D space (Madison and Bottkol, 2007). They presented a method in which a vehicle with a traditional GPS-aided INS estimates the locations of features in flight when GPS was active, and then use these features for aiding the INS in absence of GPS. Features were selected and tracked using a Lucas-Kanade feature tracker. Simulation results for a vehicle that momentarily loses GPS were provided.

More flexible navigation systems have been developed using SLAM techniques. Langelaan et al. used an unscented Kalman filter for simultaneously estimating vehicle states as well as feature locations in inertial space (Langelaan, 2007). A Mahalonobis norm was used as a statistical correspondence for data association from frame-to-frame for each estimated feature. New feature locations were initialized using a projection onto the ground plane. Simulation results for a UAV navigating through a 2D environment were presented, however no flight test results were reported.

Fowers et al. implemented a Harris feature detector along with a random sample consensus (RANSAC) algorithm for the correspondence of features in an FPGA processor (Fowers and Archibald, 2008). Their architecture provided drift corrections to an INS for stabilizing 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 IMU-drift correction measurements.

Ahrens (Ahrens et al., 2009) presented a vision aided navigation system for drift free navigation in indoor environments. They combined emulated inertial measurements (using a motion capture system) with monocular vision sensors. Flight test results in indoor environment were presented.

Recently, navigation in the presence of intermittent GPS signals was explored by Scherer et al. (Scherer et al., 2012). This work uses a stereo camera sensor along with a laser range finder and intermittent GPS to map riverines, in the presence of obstacles such as trees, bridges, and embankments. The results are presented from a surrogate data collection system mounted on a small boat, and show an impressive ability to navigate in adverse conditions. While their goals are similar to those presented in this paper, our work is distinguished by our use of a monocular camera sensor, which is a more common payload and can be more practical for lightweight vehicles. Furthermore, the flight test results of the stereo system, presented in Haines (Haines, 2011), do not demonstrate the extended flight distances and durations demonstrated in our paper.

Wu et al. developed a method for fusing feature information from a monocular vision sensor in an extended Kalman filter framework (Wu and Johnson, 2010; Wu et al., 2012). The approach in those papers relied on tracking features whose locations were estimated when GPS was active. When GPS was inactive, they demonstrated that this method ensures bounded hover of a rotorcraft UAV through flight test results. However, this method could not sustain waypoint based flight without *a priori* knowledge of feature locations and estimate the position of new feature points without GPS. The key missing element was an algorithm for safely selecting and removing features in an online feature-reference database. This is remedied here by creating a real-time reference feature management algorithm which ranks the features based on how often their predicted locations match measured location. Features that are not seen consistently over a threshold number of images are dropped and replaced with new features, this enables the UAV to navigate while autonomously flying between waypoints in a completely GPS-denied unmapped dynamic environment without knowing feature locations *a priori*.

Weiss et al. (Weiss et al., 2011), an impressive monocular slam implementation and LQR controller is described. The SLAM algorithm used is based on PTAM algorithm (Klein and Murray, 2007), a keyframe based method using periodic global map optimization and localization from the resulting map. The results presented show the vehicle is capable of waypoint following flight and accurate position hold. These results are expanded in (Achtelik et al., 2011) and (Weiss et al., 2012). Several important modifications are made to the original PTAM algorthm, including throwing out distant keyframes to allow the vehicle to cover greater distances and the implementation of a speed estimation system for minimizing drift during map initialization. The resulting waypoint following results demonstrate a highly capable autonomous system able to operate in indoor and outdoor environments. However, it is reported in (Weiss et al., 2011) that the system results in overshoots during waypoint following flight. Furthermore, it was reported that the algorithm could overwhelm available computational resources in the bundle adjustment phase, resulting in limited computational power available for other onboard activities in resource constrained environments onboard UAVs. In addition, their approach results in a slow drift in attitude, which can be concerning for long duration flights. Hence, while the results in this work are impressive, this approach does not conclusively provide a comprehensive solution to the vision aided navigation and control problem.

The V-INS system proposed in this paper differs from that of Weiss and collaborators (Weiss et al., 2011; Achtelik et al., 2011; Weiss et al., 2012) in the fact that a fundamentally different SLAM approach is used: we use EKF-SLAM. The benefits of EKF-SLAM are that it makes it relatively straightforward to integrate measurements from inertial sensors leading to our V-INS architecture, and that it fits well with commonly used aerospace navigation and filtering frameworks which tend to be INS based (Kayton and Fried, 1997). There is an ongoing debate in the SLAM community about the superiority of EKF-SLAM versus PTAM and Keyframe based algorithms. Attempts have been made to resolve which approach is better (Strasdat et al., 2010), but the verdict is far from conclusive (IMU integration is not considered, and filtering is acknowledged as a possible option for low-computational-power systems, of which our system is an example). We believe that this successful demonstration of an EKF filtering-based approach will constructively contribute to this debate, especially considering how few successful implementations of vision aided SLAM navigation there are for rotorcraft UAVs (Kendoul, 2012). Furthermore, we do feel that a combination of keyframe based techniques and EKF based multi-sensor fusion based implementations such as that used in this paper, can potentially result in even more powerful algorithms as available onboard computational power increases.

1.1.2 UAV vision-aided autonomous landing in GPS-Denied environments

Several previous works have tackled the challenge of autonomous landing of a UAV in unknown GPS-Denied environments. Leading techniques for autonomous landing site selection and landing, such as in Chamberlain et al. (Chamberlain et al., 2011), rely on using GPS-INS navigation solutions and scanning range sensors for landing site selection. Frietsch et al. presented an application of vision to the hovering and landing of a quad-rotor UAV (Frietsch and Trommer, 2008). In that paper, 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. Autonomous landing of rotorcraft in GPS-denied environment was achieved by Merz et al. (Merz et al., 2004). The system required the use of a specially marked landing pad, and a pan-tilt camera was used to maintain view of the pad over long distances.

A related application for vision-aided navigation is that of spacecraft descent and landing. Mourikis et al. presented an application of vision-aided navigation to the problem of planetary landing (Mourikis et al., 2009). Images from a downward-facing camera were processed with a Harris corner detector to generate trackable features. These features were then matched to features with known 3-D positions, generated from a topographical map. In addition, features are matched frame-to-frame to generate velocity measurements, which extended the system to handle lower altitude than was possible with map-matching. The position and velocity measurements are integrated with IMU data in an EKF framework. The system was validated with image sequences and IMU data from a sounding rocket, which was post-processed to estimate the vehicle location during descent.

Theodore et al. (Theodore et al., 2006) presented a flight-tested vision aided autonomous landing system for landing at unprepared sites. Their work comes probably the closest to what has been achieved here, and highlights the key difficulty in performing vision aided landing: ensuring consistent vision aided navigation solution from flight altitude (e.g. 50m Above Ground Level (AGL)) to ground (0m AGL). However, they relied on GPS until reaching an altitude of 12 meters AGL, after which monocular SLAM based pseudo-GPS was used until 2 meters AGL, from where the landing was performed using only inertial measurements until weight on wheels switch was triggered. In contrast to their work, the vision aided navigation system presented in our work overcomes the difficulties in using vision aided solution in closed-loop control to perform landings from hover altitude (50 m AGL) to landing (0 meters AGL).

1.2 Contributions and Overview of Our Work

In summary, most of the works discussed above were limited due to 1) vision aided navigation was not used in GPS-denied closed-loop way-point following flight and/or GPS-denied landing, 2) they were validated through simulation only, 3) most results were implemented on ground vehicles, which have slower and stable dynamics and any closed-loop flight tests were conducted in controlled environments. This paper contributes to the state-of-the-art in vision aided GPS-denied navigation by lifting these limitations, and providing answers to the three open questions posed in 1. An overview of the main contributions follows:

- A monocular V-INS for GPS-denied navigation and real-time closed-loop control of unstable UAVs in both outdoor and indoor environments is presented,

- A novel feature database management algorithm is presented, which ranks features based on a confidence index and removes features if they fall below a dynamic threshold related to the number of feature correspondences,
- Flight test results where the navigation solution from the presented V-INS for real-time long-term closed-loop control of unstable resource-constrained rotorcraft platforms are presented in real-world situations. The algorithms are validated on two classes of rotorcraft UAVs in closed-loop control: a 66 kg helicopter UAV operating outdoors and a 1 kg small ducted-fan UAV operating in cluttered unmapped environments.

We begin with describing a monocular camera model in Section 2. The reference frames utilized are discussed in Section 3. In Section 4 some preliminary details of the EKF based implementation are provided. In Section 5 the V-INS algorithm is presented along with the feature location database management algorithm. Flight test results on an outdoor 66 kg rotorcraft UAV and an indoor M-UAV are presented in Section 6. The paper is concluded in Section 7.

2 Pinhole Camera Model

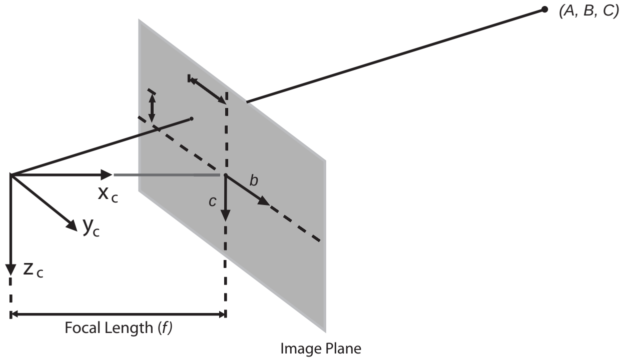


Figure 2: Camera perspective projection model used for relating 3D position to position in 2D images. The point (A, B, C) is projected onto the camera image plane to the point (b, c).

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 (Wu and Johnson, 2008). 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

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

Additionally, a camera calibration is performed to remove the effects of distortion.

3 Reference Frames

In the following, two standard aerospace right handed coordinate systems are defined: the inertial frame (denoted by the superscript i) is a local frame that is aligned to the local North, East, and down directions, and the body frame (denoted by the superscript b) is aligned such that the x axis points forward and the z axis pointing downwards (Kayton and Fried, 1997). Let $q = [q_1, q_2, q_3, q_4]^T$ denote the rotation quaternion, where q_1 defines the rotation angle and $[q_2, q_3, q_4]$ the axis of rotation. Then the orthonormal rotation matrix operator $L_{i\to b}$ transports vectors from the inertial to body frame:

$$L_{i\to b} = \begin{bmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2(q_2q_3+q_1q_4) & 2(q_2q_4-q_1q_3) \\ 2(q_2q_3-q_1q_4) & q_1^2-q_2^2+q_3^2-q_4^2 & 2(q_3q_4+q_1q_2) \\ 2(q_2q_4+q_1q_3) & 2(q_3q_4-q_1q_2) & q_1^2-q_2^2-q_3^2+q_4^2 \end{bmatrix}. \tag{4}$$

It follows that the orthonormal rotation matrix for transporting vectors from the body to inertial frame is given by $L_{b\to i} = L_{i\to b}^T$.

The camera reference frame (denoted by superscript c) 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 vector components can be transformed from the body frame to the camera frame by defining the pan (ψ_c), tilt (θ_c), and roll (ϕ_c)

angles, and using the rotation matrix:

$$L_{cb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_c & \sin\phi_c \\ 0 & -\sin\phi_c & \cos\phi_c \end{bmatrix} \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_c & 0 & \cos\theta_c \end{bmatrix} \begin{bmatrix} \cos\psi_c & \sin\psi_c & 0 \\ -\sin\psi_c & \cos\psi_c & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

It is usually the case that the camera is not located at the origin of the body reference frame. In this case, appropriate translation needs to be applied.

4 The Extended Kalman Filter

The EKF is a widely used Bayesian filtering tool for obtaining suboptimal solutions to nonlinear filtering problems (see for example (Gelb, 1974; Kayton and Fried, 1997)). A brief review of the EKF is provided here for completeness. EKF employs instantaneous linearization at each time step to approximate the nonlinearities, the propagation of the error covariance matrix is assumed to be linear. The architecture presented here is of the mixed continuous-discrete type. Let x denote the state of the system, u denote an input to the system, y denote the output, z denote the sampled output, w denote additive process noise, v denote additive measurement noise, and consider the following generic representation of the system dynamics:

$$\dot{x}(t) = f(x(t), u(t)) + Fw(t), \quad x(t_0) = x_0 \tag{6}$$

$$y(t) = g(x(t), u(t)), \tag{7}$$

$$z(k) = y(k) + Gv(k). \tag{8}$$

In the above equation, k denotes the discrete sampling time when a measurement is available. Note that different sensors may update at different rates. Furthermore, it is assumed that f and g are differentiable functions. The matrices F and G determine the variance of the process and measurement noise which are assumed to be Gaussian. The EKF algorithm consists of a prediction and a correction step. In the prediction step, the nonlinear process model is integrated forward from the last state estimate $\hat{x}(k-1)$ to yield the predicted state $\tilde{x}(k)$ until a measurement is available at time k. The error covariance matrix P is also propagated to yield the predicted error covariance matrix $\tilde{P}$ by linearizing the process model using the Jacobian matrix $A = \left[\frac{\partial f}{\partial x}\right]_{x=\hat{x}(k)}$ around the instantaneous state estimate from the previous step $\hat{x}(k-1)$. This results in the following approximation for the error covariance which is also integrated forward from

$P(k-1)$ to yield $\tilde{P}(k)$:

$$\dot{\tilde{P}} \approx AP + PA^T + Q, \tag{9}$$

where $Q = FF^T$. The EKF correction step occurs when a sensor measurement has been received. The expected system output $\tilde{y}(k)$ is predicted using the predicted state estimate $\tilde{x}(k)$ and the nonlinear measurement model $g(\tilde{x}, u)$. The measurement model Jacobian matrix $C(k) = \left[\frac{\partial g}{\partial x}\right]_{x=\tilde{x}(k)}$ is computed around the predicted state $\tilde{x}(k)$ and used to yield the following correction equations

$$K(k) = \tilde{P}(k)C^T(k)\left[C(k)\tilde{P}(k)C^T(k) + R\right]^{-1}, \tag{10}$$

$$\hat{x}(k) = \tilde{x}(k) + K(k)\left[z(k) - \tilde{y}(k)\right], \tag{11}$$

$$P(k) = \left[I - K(k)C(k)\right]\tilde{P}(k). \tag{12}$$

In the above equation $R = GG^T$, it is assumed that G is such that R is positive definite (Gelb et al., 1974).

5 Vision aided Inertial Navigation Algorithm

In this section we describe the vision aided inertial navigation algorithm presented in Algorithm 1 and Figure 3. The algorithm simultaneously estimates the location of visual feature points and the state of the vehicle. The idea is to use a monocular camera to detect feature points using a Harris corner detection algorithm, predict frame-to-frame transitions of the feature points, use the estimated feature point locations to populate and update online a database of feature point locations, and track the movement of the feature points in the image plane to bind the IMU drift, estimate IMU bias, and correct for initial misalignment errors. Therefore, the navigation algorithm fuses measurements from inertial sensors (accelerometer and gyroscopes), a monocular downward facing camera, and an altitude sensor (such as a sonar altimeter) to form an accurate estimate of the vehicles state to enable stable hover and waypoint following flight. For this purpose, it is necessary to estimate the following states: the vehicle's position ${}^ip = [x, y, z]^T$, velocity ${}^iv = [v_x, v_y, v_z]^T$, attitude quaternion $q = [q_1, q_2, q_3, q_4]^T$, and the IMU acceleration and gyroscope biases $b_a, b_g \in \Re^3$. It is beneficial to use a minimal representation of the attitude quaternion for describing the orientation of the aircraft (Lefferts et al., 1982). This is achieved by defining an error quaternion as $\delta q \approx [1, s]^T$ such that $\delta q \otimes \hat{q} = q$, where $\otimes$ is the quaternion product operator. The error quaternion is propagated during EKF updates, and is used to correct the attitude estimates after each measurement update. Note that in practice the error quaternion takes on very small values, therefore, the update equation for error quaternions can be approximated by $\dot{\mathbf{s}} = 0$. The vector $\mathbf{s} \in \Re^3$ to be tracked as the minimal representation of the attitude error.

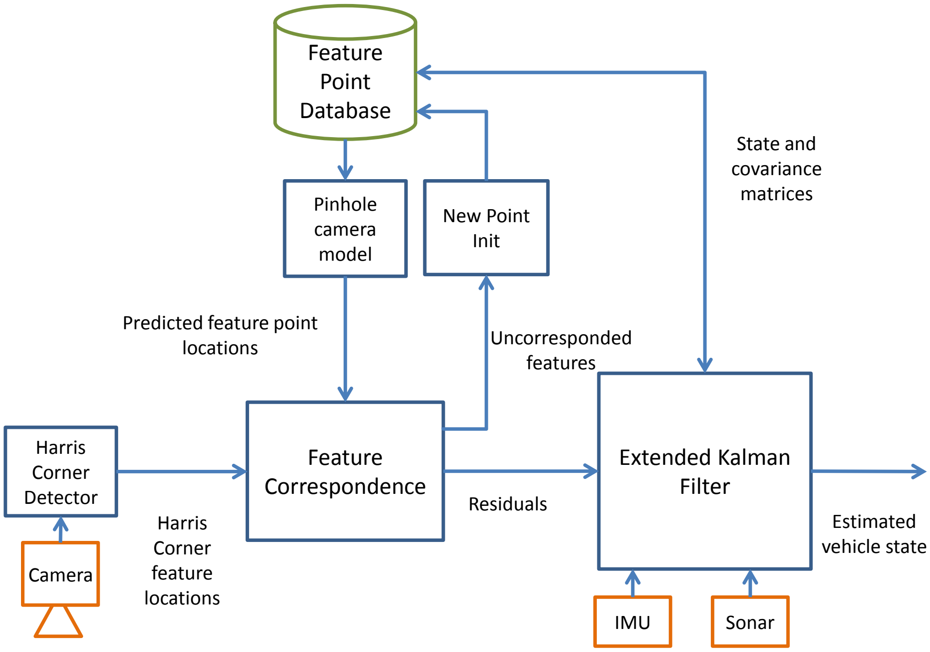


Figure 3: A block diagram describing the navigation system. Pixel location of features are corresponded to predicted locations of database points. The resulting residuals are fused with inertial data and sonar altimeter data to estimate the vehicle state, feature locations and associated covariance matrices. Uncorresponded features can be used to update the database as current points leave the field of view. The resulting architecture simultaneously estimates feature point locations and vehicle states without requiring the vehicle to maintain view of particular landmarks.

The accelerometer measurements ${}^{b}a = [a_x, a_y, a_z]^T$ and the gyroscope measurements ${}^{b}\omega = [p, q, r]$ from the IMU are modeled to account for sensor biases as follows: ${}^{b}a = a_{raw} - b_a$ and ${}^{b}\omega = \omega_{raw} - b_g$. The navigation filter is designed to estimate the biases $b_a \in \Re^3$ and $b_g \in \Re^3$ of the accelerometer and gyroscope respectively. This is important, as measurements from low-cost off-the-shelf IMUs are often subject to an unknown and variable bias.

Along with the vehicle states, the filter also needs to simultaneously estimate the position of the j^{th} feature points in the inertial frame, ${}^{i}p_{f_j} = [x_{f_j}, y_{f_j}, z_{f_j}]^T$. The tracked and corresponded feature points are stored in a database which contains a maximum of N slots.

In combination, the state vector to be estimated by the navigation filter is $\hat{x}(t) \in \Re^{15+3\times N}$ and is given as

$$\hat{x}(t) = \begin{bmatrix} \hat{s} & \hat{p} & \hat{v} & \hat{b}_a & \hat{b}_g & \hat{p}_f \end{bmatrix}^{\mathrm{T}} \tag{13}$$

where $\hat{:}$ represents estimated variables.

Next the process models and measurement models are described.

Algorithm 1 Navigation update and confidence-based database management algorithm

```
while new IMU packets are available do
  Propagate state and covariance matrix
  if new image available then
    Extract Harris corner features

    //Point correspondence
    for All features $j$ do
      for All database points $i$ do
        Project database point onto image plane
        Calculate Mahalanobis distance $Z_{ij} = e_{ij}^{\mathrm{T}}(C_j P C_j^{\mathrm{T}} + R)^{-1} e_{ij}$
      end for
    end for
    for All features $j$ do
      Find database point $i_{min}$ such that { $Z_{i_{min}j} \leq Z_{ij}$ for all $i$
      and $Z_{i_{min}j} \leq Z_{max}$ }
      if database point $i_{min}$ already corresponded to feature $j_{prev}$
      and $Z_{i_{min}j} < Z_{i_{min}j_{prev}}$ then
        Correspond feature $j$ to database point $i_{min}$ and mark previous database point $j_{prev}$ as uncorresponded
      end if
    end for

    //Kalman filter update
    for All database points $i$ do
      if Database point $i$ was corresponded then
        Increment confidence in database point $i$
        Perform Kalman filter update
      else
        Decrement confidence in database point $i$
      end if
    end for

    //Point initialization
    for All features $j$ do
      if Feature $j$ was uncorresponded then
        Get inertial frame point location based on altitude, Eq. (39)
        Set confidence of new point based on number of corresponded database points during previous update
        for All database points $i$ do
          if New point confidence > database point confidence then
            Insert new point in database location $i$
            break
          end if
        end for
      end if
    end for
  end if
end while
```

5.1 Navigation Algorithm: Process Model

The following nonlinear process model is used for predicting state estimates:

$$\dot{\hat{b}}_a = 0, \qquad \dot{\hat{b}}_g = 0, \qquad \dot{\hat{s}} = 0 \tag{14}$$

$${}^i\dot{\hat{p}}_{f_j} = 0,\ j = 1..N, \qquad {}^i\dot{\hat{p}} = \hat{v}, \qquad {}^i\dot{\hat{v}} = \hat{L}_{b \to i}({}^b a - \hat{b}_a) \tag{15}$$

The following equation is used to predict the quaternion estimate using bias corrected measurements from the gyroscopes:

$$\dot{\hat{q}} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \tag{16}$$

The error covariance P is propagated using the following equation

$$\dot{P} = AP + PA^T + Q, \tag{17}$$

where A is the Jacobian of the process model with respect to the states, and Q is the process covariance matrix. Appropriately initializing the covariance matrix is important. This topic has been dealt with in detail in (Wu et al., 2011). The following closed-form equations are the non-zero components of the Jacobian matrix A, these well-known equations are repeated here to facilitate real-time implementation:

$$\begin{gathered} \tfrac{\partial^i \dot{\hat{p}}}{\partial^i \hat{v}} = I_{3\times3}, \qquad \tfrac{\partial^i \dot{\hat{v}}}{\partial \hat{s}} = -\hat{L}_{b\to i}{}^b\widetilde{a}, \qquad \tfrac{\partial^i \dot{\hat{v}}}{\partial \hat{b}_a} = -\hat{L}_{b\to i}, \\ \tfrac{\partial \dot{\hat{s}}}{\partial \hat{s}} = -{}^b\widetilde{\omega}, \qquad \tfrac{\partial \dot{\hat{s}}}{\partial \hat{b}_g} = -I_{3\times3} \end{gathered} \tag{18}$$

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

$$\widetilde{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{19}$$

5.2 Navigation Algorithm: Measurement Model

The measurement model $g(x)$ here is used to predict the expected measurement using the states propagated through the process model. In the following the relative position of the feature point j in the inertial frame will be given as: ${}^i r_j = {}^i p_{f_j} - {}^i p$. The relative vector r_j will be denoted in the camera frame as ${}^c r_j = [X_{f_{jc}}, Y_{f_{jc}}, Z_{f_{jc}}]^T$, in the body frame as ${}^b r = [X_{f_{j_b}}, Y_{f_{j_b}}, Z_{f_{j_b}}]^T$, and in the inertial frame as ${}^i r = [X_{f_{j_i}}, Y_{f_{j_i}}, Z_{f_{j_i}}]^T$.

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

$$\hat{X}_j = f_x \frac{\hat{Y}_{f_{c_j}}}{\hat{X}_{f_{c_j}}}, \qquad \hat{Y}_j = f_y \frac{\hat{Z}_{f_{c_j}}}{\hat{X}_{f_{c_j}}}. \tag{20}$$

This nonlinear measurement model needs to be linearized at every time step. This is achieved by computing the Jacobian of the measurement model using the following closed-form equations (equations given here explicitly to facilitate real-time implementation):

$$\frac{\partial \hat{z}}{\partial^i \hat{p}} = \left(\frac{\partial \hat{z}}{\partial^c \hat{r}}\right)\left(\frac{\partial^c \hat{r}}{\partial^i \hat{p}}\right) \tag{21}$$

$$\frac{\partial \hat{z}}{\partial^c \hat{r}} = \begin{bmatrix} \frac{\partial \hat{X}}{\partial \hat{X}_{f_{c_j}}} & \frac{\partial \hat{X}}{\partial \hat{Y}_{f_{c_j}}} & \frac{\partial \hat{X}}{\partial \hat{Z}_{f_{c_j}}} \\ \frac{\partial \hat{Y}}{\partial \hat{X}_{f_{c_j}}} & \frac{\partial \hat{Y}}{\partial \hat{Y}_{f_{c_j}}} & \frac{\partial \hat{Y}}{\partial \hat{Z}_{f_{c_j}}} \end{bmatrix} \tag{22}$$

$$\frac{\partial \hat{X}}{\partial \hat{X}_{f_{c_j}}} = -f_x \frac{\hat{Y}_{f_{c_j}}}{\hat{X}^2_{f_{c_j}}} = -\frac{\hat{X}}{\hat{X}_{f_{c_j}}}, \qquad \frac{\partial \hat{X}}{\partial \hat{Y}_{f_{c_j}}} = \frac{f_x}{\hat{X}_{f_{c_j}}}, \qquad \frac{\partial \hat{X}}{\partial \hat{Z}_{f_{c_j}}} = 0\,, \tag{23}$$

$$\frac{\partial \hat{Y}}{\partial \hat{X}_{f_{c_j}}} = -f_y \frac{\hat{Z}_{f_{c_j}}}{\hat{X}^2_{f_{c_j}}} = -\frac{\hat{Y}}{\hat{X}_{f_{c_j}}}, \qquad \frac{\partial \hat{Y}}{\partial \hat{Y}_{f_{c_j}}} = 0\,, \qquad \frac{\partial \hat{Y}}{\partial \hat{Z}_{f_{c_j}}} = \frac{f_y}{\hat{X}_{f_{c_j}}}, \tag{24}$$

Inserting (23) and (24) into (22) gives

$$\frac{\partial \hat{z}}{\partial^c \hat{r}} = \frac{1}{\hat{X}_{f_{c_j}}} \begin{bmatrix} -\hat{X} & f_x & 0 \\ -\hat{Y} & 0 & f_y \end{bmatrix}, \tag{25}$$

The derivative of the camera frame relative location of the feature point with respect to vehicle position is given by

$$\begin{aligned}\frac{\partial^c \hat{r}}{\partial^i \hat{p}} &= \frac{\partial\left(\hat{p}_{f_{c_j}} -^{c} \hat{p}\right)}{\partial^i \hat{p}}\\ &= \frac{\partial \hat{p}_{f_{c_j}}}{\partial^i \hat{p}} - \frac{{}^c\partial \hat{p}}{\partial^i \hat{p}}\\ &= 0 - \frac{\partial\left(\hat{L}_{i\rightarrow c}{}^i\hat{p}\right)}{\partial^i \hat{p}}\\ &= -\hat{L}_{i\rightarrow c}. \end{aligned} \tag{26}$$

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

$$\begin{aligned}\frac{\partial \hat{z}}{\partial \hat{R}} &= \left(\frac{\partial \hat{z}}{\partial^c \hat{r}}\right)\left(\frac{\partial^c \hat{r}}{\partial \hat{R}}\right)\\ &= \left(\frac{\partial \hat{z}}{\partial^c \hat{r}}\right)\frac{\partial\left(L_{b\rightarrow c}{}^b\hat{r}\right)}{\partial \hat{R}}\\ &= \left(\frac{\partial \hat{z}}{\partial^c \hat{r}}\right) L_{b\rightarrow c}\left(\frac{\partial^b \hat{r}}{\partial \hat{R}}\right), \end{aligned} \tag{27}$$

where $\frac{\partial^b \hat{r}}{\partial \hat{R}} =^b \widetilde{r}$.

The partial derivatives with respect to the feature point locations are calculated as

$$\frac{\partial \hat{z}}{\partial^i \hat{p}_{f_j}} = \left(\frac{\partial \hat{z}}{\partial^c \hat{r}}\right)\left(\frac{\partial^c \hat{r}}{\partial^i \hat{p}_{f_j}}\right) \tag{28}$$

$$\frac{\partial \hat{z}}{\partial^c \hat{r}} = \frac{1}{\hat{X}_{f_{c_j}}}\begin{bmatrix} -\hat{X} & f_x & 0 \\ -\hat{Y} & 0 & f_y \end{bmatrix} \tag{29}$$

$$\begin{aligned}
\frac{\partial^c \hat{r}}{\partial^i \hat{p}_{f_j}} &= \frac{\partial \left(\hat{p}_{f_c} - {}^c p\right)}{\partial^i \hat{p}_{f_j}} \\
&= \frac{\partial^c \hat{p}_{f_j}}{\partial^i \hat{p}_{f_j}} - \frac{\partial^c p}{\partial^i \hat{p}_{f_j}} \\
&= \frac{\partial \left(L_{i\to c}{}^i \hat{p}_{f_j}\right)}{\partial^i \hat{p}_{fp_j}} - 0 \\
&= L_{i\to c} \qquad (30)
\end{aligned}$$

The nonlinear and linearized measurement models are used in the Kalman filter update equations presented in Section 4. Note that the feature point update is performed sequentially, which drastically reduces computational load by requiring N inversions of 2×2 matrices, rather than a single inversion of an $2N \times 2N$ matrix, and assumes a statistical independence of the feature points.

5.3 Feature Point Detection and Tracking

In the literature, there are many methods for the detection of suitable features in an image and tracking those features across multiple frames. Examples of combined feature detectors and trackers are the Kanade-Lucas-Tomasi (KLT) tracker (Tomasi, 1991), and Lowe's Scale-Invariant Feature Transform (SIFT)(Lowe, 1999). Additionally, algorithms are available for feature detection alone, such as the Harris and Stevens combined corner and edge detector(Harris and Stephens, 1988). Of course, feature tracking must then be performed by some other means.

The system presented here uses a Harris and Stevens corner detector for feature point detection, and tracking is performed by corresponding new measurements to the database of stored points using a Mahalanobis distance. The advantages of this method are its computational efficiency and its natural relationship to the statistical information available through the use of the Kalman filter.

The Harris and Stevens corner detector uses a local second order moment matrix of the intensity gradients to determine directions of large intensity gradients. Large gradients in two directions indicate a corner, while small gradients indicate a region of uniform intensity. Let $I(u, v)$ be the intensity of the image at pixel location (u, v). The image gradient in the x and y directions at (u, v) are given by

$$I_x(u, v) = \frac{I(u+1, v) - I(u-1, v)}{2} \tag{31}$$

$$I_y(u, v) = \frac{I(u, v+1) - I(u, v-1)}{2} \tag{32}$$

The moment matrix for pixel location (b, c) is calculated as

$$M(X, Y) = \sum_{(u,v) \in W} w(u, v, X, Y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_x^2 \end{bmatrix} (u, v) \tag{33}$$

where $w(u, v, X, Y)$ is a gaussian kernel around image point (X, Y) given by

$$w(u, v, b, c) = \exp\left(-\frac{(u - X)^2 + (v - Y)^2}{2\sigma^2} \right) \tag{34}$$

and W defines the window around (X, Y) over which the summation occurs. In this system, summation occurred over a 3×3 window.

The corner score is calculated from this matrix

$$M_c = \det(M) - \kappa \left[\mathrm{Tr}(M)\right]^2 \tag{35}$$

$$= \alpha\beta - \kappa \left[\alpha + \beta\right]^2 \tag{36}$$

where α and β are the eigenvalues of M. Therefore, the corner detector gives high scores for locations with both eigenvalues large without performing an explicit eigenvalue decomposition.

During feature extraction, the image is divided into bins and a max number of features per bin is set to ensure features are spread over the image. In addition, a minimum distance between features is required. The extracted Harris corners are sorted by corner score before being used in the navigation algorithm.

Feature tracking is performed by corresponding measured features to the predicted location of features in the database. The measurement is considered to correspond to a stored point if its Mahalanobis distance to the stored point is minimal with respect to all other pairings and is below a threshold. The Mahalanobis distance is a Euclidian distance weighted by the covariance of the predicted measurement. The Mahalanobis distance between measurement z_i and database point p_{f_j} is given by

$$Z_{ij} = e_{ij}^{\mathrm{T}}(C_j P C_j^{\mathrm{T}} + R)^{-1} e_{ij} \tag{37}$$

$$e_{ij} \triangleq z_i - g_j(\hat{x}) \tag{38}$$

where $g_j(\hat{x})$ is the nonlinear measurement model for p_{f_j}, C_j is the linearized measurement matrix for point p_{f_j}, P is the state covariance matrix, and R is the measurement noise covariance.

If a measurement z_i does not meet the maximum Mahalanobis distance threshold for any database point, it is considered to be a measurement of no currently stored point and is uncorresponded. Uncorresponded points can be used to initialize new database points, as described in the following section.

5.4 Adding and Deleting Elements of the Feature Point Location Database

Typical operational scenarios for UAVs, such as aggressive maneuvers and following position commands, make it impossible for a fixed camera to maintain sight of a fixed set of points. Therefore it is necessary to develop a method of initializing new database points online and remove points from the database that are no longer in view.

New points are initialized from uncorresponded features, and features with the highest corner score are initialized first. Because of the lack of depth information from a single monocular image, additional information is necessary to estimate the measurements' three-dimensional coordinates from their two-dimensional image plane location. In many common environments for vehicles with a camera aligned with the body z-axis, it is a reasonable assumption that the points lie on a plane at a distance of the altitude of the vehicle.This assumption would be valid, for instance, for an vehicle flying indoors over an uncluttered floor, or a outdoor vehicle flying over a plain. By initializing points with a suitable uncertainty in altitude, points that are in fact not on the ground plane will converge to the correct altitude if they remain in view (Wu and Johnson, 2010). Inverting the camera model presented in section 2, we get

$$L_{c\to i}\begin{bmatrix} 1 \\ b/f_x \\ c/f_y \end{bmatrix} X_{f_{j_i}} = \begin{bmatrix} X_{f_{j_i}} \\ Y_{f_{j_i}} \\ Z_{f_{j_i}} \end{bmatrix} = {}^{i}r_j. \tag{39}$$

Distance $X_{f_{j_i}}$ is the altitude of the camera, and image plane locations b and c and focal lengths f_x and f_y are known. The covariance of the new feature is set to fixed value scaled by the distance of the vehicle to that feature:

$$P_{f_j} = L_{c\to i} \begin{bmatrix} R_0||{}^i r_j||^2 & 0 & 0 \\ 0 & R_0||{}^i r_j||^2/f_x^2 & 0 \\ 0 & 0 & R_0||{}^i r_j||^2/f_y^2 \end{bmatrix} L_{c\to i}^{\mathrm{T}}, \tag{40}$$

where P_{f_j} is the 3×3 block within the full covariance matrix P corresponding to the covariance of point f_j with itself, and R_0 is a scalar initialization value. All other cross-correlation terms in P are set to zero.

One of the key contributions of this work is to present a novel method for online management of the database of feature locations. In particular, the addition of new features to the database based on the number of corresponded features ensures that the database is quickly refilled with new points when a large number of features are lost.

Feature removal from the database is determined by comparing a "confidence index" of the current database features with the initialization confidence index of a new feature. The confidence index for current database features is limited to a minimum and maximum of 0 and 100, and is incremented for every iteration a feature is corresponded to a measurement, and decremented if it is not corresponded. The index can be thought of as a measure of how often a database feature is used (corresponded) for navigation in the recent past. The initialization confidence index is related to the number of database features which were corresponded on the current iteration. A database feature is replaced if its confidence index falls below the initialization confidence index. When few features are corresponded, the initialization confidence index is high, and features are replaced quickly. When many features are corresponded, the initialization confidence index is low and features are replaced slowly. The initialization confidence index represents a dynamic point removal threshold modified by the number of correspondences in the current iteration.

To understand the effect of this initialization method, let us consider a few cases. First, consider a slow translation of the vehicle when there are strong features from the image, and most of the database features are matched. As the vehicle translates, some features will leave the frame and their confidence index will drop. Since most features are still matched, the features leaving the frame are replaced slowly, because the confidence index takes time to fall below the initialization confidence index.

Now consider the case when a large change in pitch attitude has occurred, and for a short time no database features can be seen. Therefore the initialization confidence index is very high and it does not take long for some features to fall below this value and be replaced. As these new features are seen in later iterations, however, the initialization confidence index lowers, and features are not replaces as quickly. The best of the old features are maintained longer, and can be used again if the vehicle returns to the original orientation.

In this way, the feature database is quickly replaced if the features visible through the camera are not being matched with measurements, but the features are replaced in such a way that the most historically reliable features are saved longest and can be used if they come back into view. This property of the algorithm improves reliability for platforms subject to abrupt attitude changes or changes in the environment, such as helicopters.

To demonstrate the effectiveness of the proposed method, it is compared to a fixed threshold feature removal method. The fixed threshold method is implemented by setting the initialization confidence index to 50 regardless of the number of features corresponded. This fixed threshold method is equivalent to the method proposed in (Davison, 2003) and (Davison et al., 2007), with the number of iterations equal to 50 and the percent correspondence equal to 50%.

The simulated vehicle was flown over a highly textured surface which was instantaneously changed to a low textured surface. This is similar to the effect of instantaneous lighting changes, such as a light being switched off in a room. The texture removal was performed 8 times for each method, and the average, minimum and maximum number of feature correspondences is shown in Figure 4. It can be seen that the dynamic threshold does a better job maintaining a steady number of consistent features throughout the test. When the texture is removed, the fixed threshold takes a longer time to throw out the now-unobservable features, as the confidence index for these points fall. In the dynamic thresholding case, the threshold is raised when few features correspond, and so unseen features are removed quickly. The dynamic threshold method allows the navigation algorithm to rapidly adapt to dynamic environment.

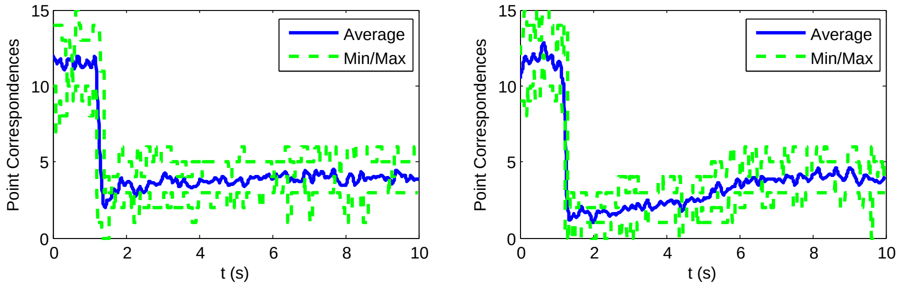
6 Results

Validation of the Kalman Filter vision-based navigation system was conducted on multiple unmanned aerial vehicles of different sizes, operating in indoor and outdoor conditions. The use of different vehicles illustrates the robustness and portability of the system design. The larger of the vehicles is the GTMAX (Johnson et al., 2004), a 66 kg modified Yamaha RMAX helicopter UAV with custom avionics and flight software designed by the Georgia Institute of Technology Unmanned Aerial Vehicle Research Facility (UAVRF). Also, the navigation system was tested on the EASE, a 1 kg ducted fan vehicle, in an indoor environment. Both vehicles used the same baseline adaptive controller described in (Johnson and Kannan, 2005).

Simulation results presented below were generated using the Georgia Tech UAV Simulation Tool (GUST). The GUST software package that combines a high-fidelity vehicle and environment model, onboard flight control software, and ground station software. The vehicle model is a six rigid body degree of freedom model with additional engine and rotor dynamics. The vehicle model simulates sensor noise, delay, location, orientation, and actuator dynamics and saturation.

6.1 GTMAX Helicopter

The GTMAX helicopter UAV is a modified Yamaha RMAX helicopter, weighing 66 kg and with a 3 m rotor diameter. Figure 1 shows a picture of the GTMAX. The helicopter is outfitted with a variety of sensors including an Inertial Science IMU sampled at 100 Hz, short-range sonar, magnetometer, and differential GPS (Johnson et al., 2004). The vehicle was equipped with a Prosilica GC 1380 camera for vision-based navigation, which features progressive scanning and gigabit ethernet connectivity. Two onboard computers with 1.6 GHz Intel Pentium M CPUs process information from these sensors for guidance, navigation and control of the vehicle. For the vision-based navigation described in this paper, the secondary computer performs image acquisition and feature detection, while the primary computer performs the navigation update and handles vehicle guidance and control. The computational power onboard was found to be sufficient to support real-time operation. In particular, the EKF described in Section 4 processes vision data and makes updates based on vision at at 20 Hz. It also features a sequential update (Christophersen et al., 2006), hence, the updates are performed as measurements arrive, allowing flexibility in handling sensors with different update rates.



(a)

(b)

Figure 4: The number of feature correspondences over time. The left plot data was gathered using the new dynamic thresholding presented in this paper. The right plot uses a fixed threshold and is shown for comparison. After the change in image texture at $t = 1$, the proposed method more quickly removes old points from the database.

In combination, the fused navigation solution is available to the controller at 100 Hz, which is sufficient to support real-time operation. All flight results presented below employ the GTMax's baseline adaptive flight controller that has been described in detail in (Johnson and Kannan, 2005).

6.1.1 GTMAX flight test results

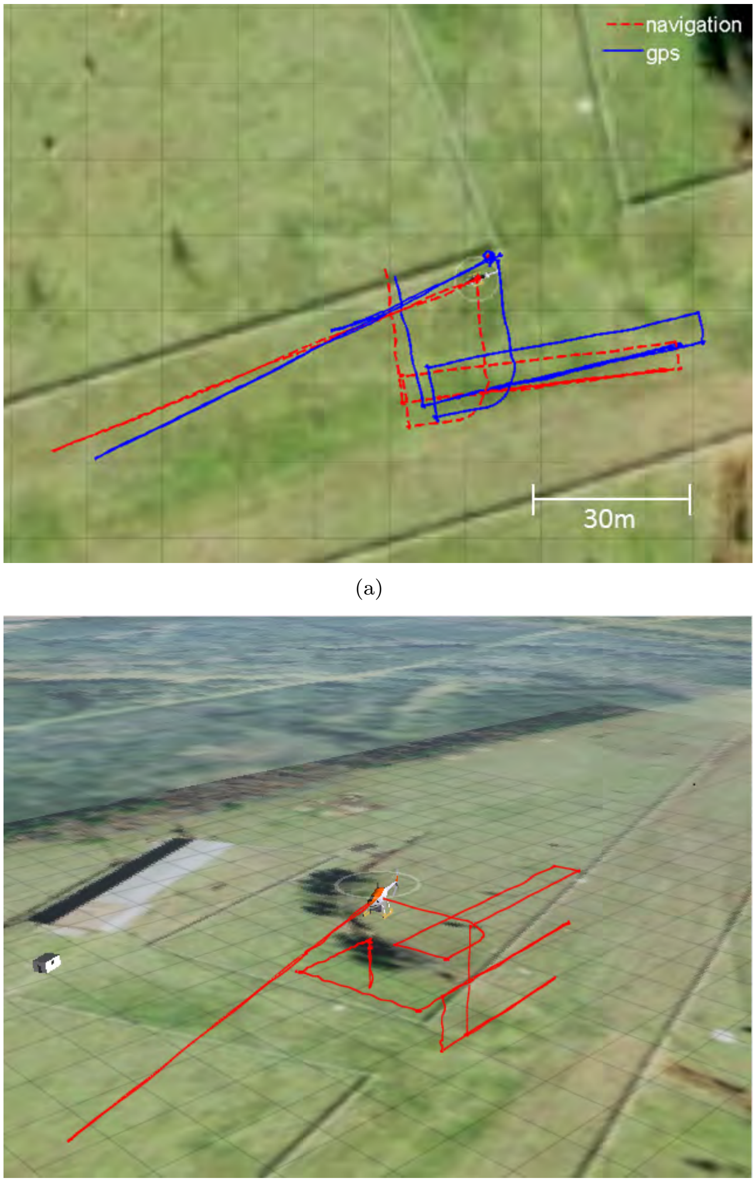
The data presented below show excerpts from a single flight during which there were 16 minutes of continuous operation of the vision-based navigation system (16 minutes is about 40% of the typical endurance of the GTMax). Figure 5 shows the navigation solution during the flight, and the top-down view also shows the location of the vehicle as given by GPS for comparison. Note that the vehicle is not drawn to scale. Speeds as high as 6 m/s were reached and the vehicle flew at multiple altitudes . While this test was performed in an area approximately 150 m by 50 m, the results were solid, and there is no reason why GPS-denied closed-loop autonomous flight with our V-INS flight could not be performed over greater distances or for longer time. The flight was carried out over a grassy area with a fence occasionally in view. The satellite imagery in Figure 5 gives a rough indication of the flight area.

For all tests, the camera was mounted to the nose of the aircraft pointing downward, and monochrome images were gathered at approximately 20 Hz and at 320 × 240 pixel resolution. The best features in each image, with a maximum of 20, were output for use in the navigation algorithm. A maximum of 50 feature locations were stored in the database. GPS data was gathered at 5 Hz and was used for altitude measurement in the navigation solution. The measurement is analogous to using a barometric pressure sensor, in that both measurements are relative to an initial datum, rather than the altitude above ground. Also, the GPS position and velocity output is presented below for comparison to the vision-based navigation in the horizontal plane.

Figures 6 and 7 show the results of a series of translational commands given to the GTMAX while using vision-based navigation. Figure 6 shows that the navigation solution corresponds well with the GPS output, despite the vehicle traveling over 90 m each way, and having to replace online the database of feature locations. At time 100-120 seconds, it can be seen that a velocity error causes the vehicle to drift. This drift is possibly caused by mis-corresponded measurement and database features, which can occur over areas with poor texture and poor feature distribution over the image. In poorly textured areas, features are less repeatable from frame to frame, and periodic dropouts of features could cause a database point to be incorrectly matched to a neighboring feature. Reducing the occurrence of feature mis-correspondences and navigation drift in general is the subject of ongoing research. An absolute position error of 7 m is incurred over the course of the maneuver.

Figures 8 and 9 show the results of a 90 second hover of the GTMAX. The vehicle remains within approximately a 1.6 m radius circle around the commanded position.

Figure 10 show the performance of the V-INS solution during autonomous descent and landing. This is the first time an autonomous landing has been performed under vision-based navigation for a vehicle of this class in completely unmapped environments, to the authors' knowledge. This maneuver demonstrates that



(a)

(b)

Figure 5: The ground station visualization of full 16 minute (40% of aircraft endurance) flight trajectory while using vision-based navigation. GTMAX flight test data presented in this paper, except for the landing maneuver, is exerpted from this flight. The navigation output is shown in red, and GPS data from the flight is show in blue, for comparison. Note that the image of the vehicle is not drawn to scale.

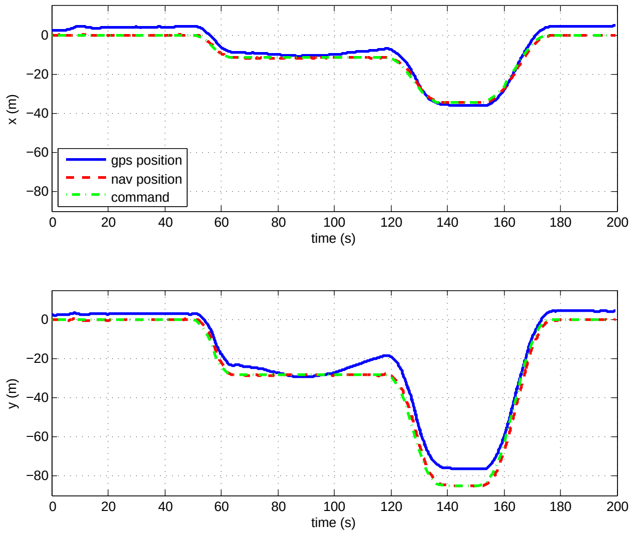


Figure 6: Comparison of vision-based navigation position states (red dashed line) with GPS location data (blue solid line) of the GTMAX during translational commands. GPS data was used in navigation solution for altitude only. The commands were given as indicated by the free dash-dot line. The vehicle tracked the commanded positions with 7 m drift of the vision-based navigation solution from GPS position data. These results are part of a 16 minute flight.

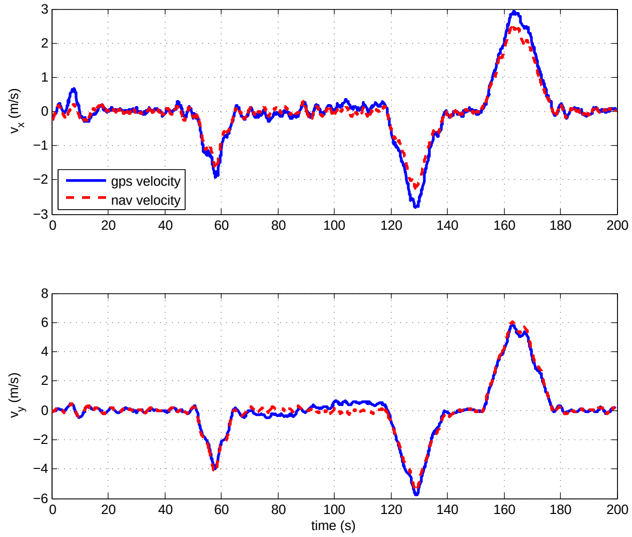


Figure 7: Comparison of vision-based navigation velocity states (red dashed line) with GPS velocity data (blue solid line) of the GTMAX during translational input commands. These results are part of a 16 minute flight.

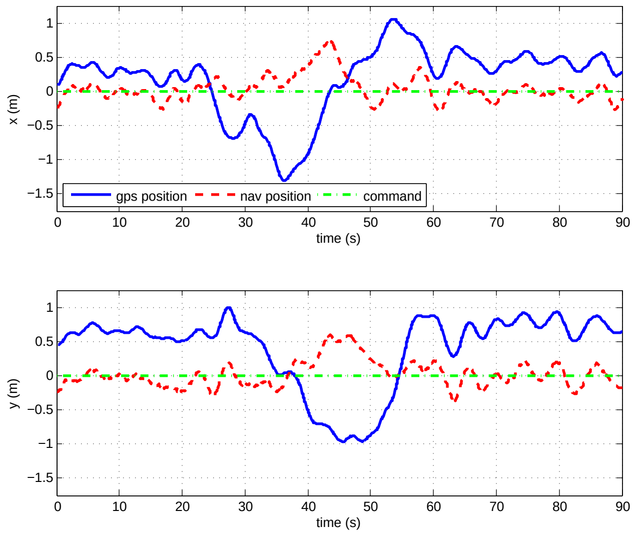


Figure 8: Comparison of vision-based navigation position states (red dashed line) with GPS location data (blue solid line) of the GTMAX during a 90 second excerpt of hovering flight approximately 100 ft above ground. GPS data was used in navigation solution for altitude only, and this can easily be replaced with a barometer or active range sensor. The vehicle remained within approximately 1.6 m of the commanded position despite a small amount of drift in the navigation solution from 22 seconds to 60 seconds. These results are part of a 16 minute flight.

the navigation solution is robust to large altitude changes and corresponding changes in the surface texture beneath the vehicle, and that the solution can be trusted during maneuvers with low velocity error tolerance.

6.1.2 GTMAX simulation results

A 350 second simulation was performed to analyze the performance of the navigation system over an area with a non-flat ground plane, and provide further validation of long-distance flight. During this simulation, the vehicle traveled over 560 m at an altitude of 50 m, over a group of buildings up to three stories tall (approximately 10 m). Images were captured from a simulated camera located on the vehicle. An image from the simulation environment, along with the true path of the vehicle and the waypoints of the trajectory, is shown in Figure 11.

Figure 12 show the GPS and navigation output in the x-y plane and with respect to time. The final navigation error over the 560 m flight path was 5 m. Note that no loop-closure was performed, indicating that similar performance can be expected over non-repeating paths. It can be seen that the navigation solution provides data which ensures the stability of the vehicle during the entire trajectory.

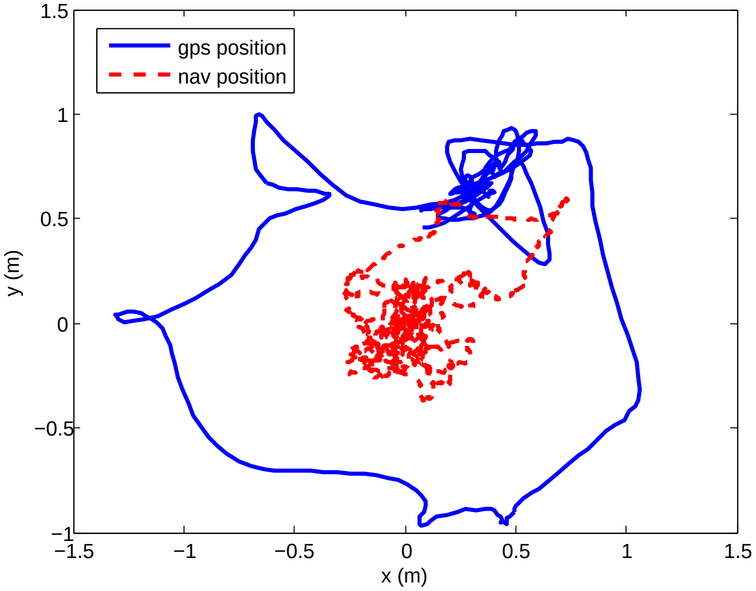


Figure 9: X-Y position of vehicle as given by vision-based navigation position states (red dashed line) and GPS location data (blue solid line) of the GTMax during a 90 second excerpt from extended hovering flight approximately 100 ft above ground. These results are part of a 16 minute flight.

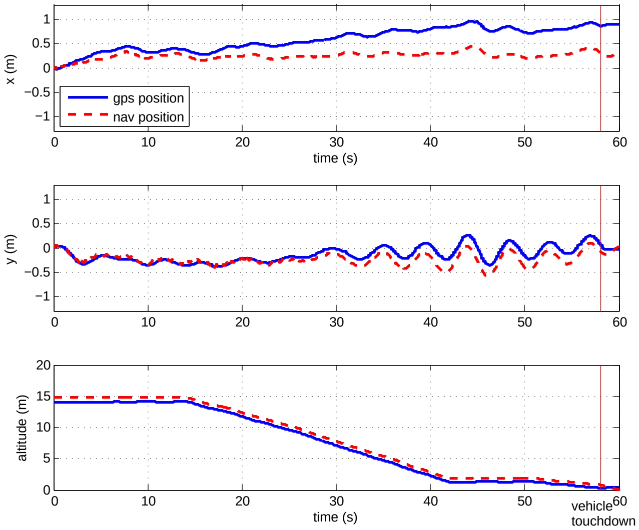


Figure 10: Time history of the position states of the GTMAX navigation solution during autonomous descent and landing. The red dashed line gives the navigation solution position, and the blue solid line gives the GPS position. These results are part of a 16 minute flight.

6.1.3 Comparison with existing V-INS techniques for GTMAX class UAVs

The approach presented here has significant improvements over that in (Wu and Johnson, 2010; Wu et al., 2012) in which comparable flight times were reported for a GTMAX class UAS with pre-mapped features. This is because our approach does not assume pre-mapped features and uses the computationally efficient feature-database management algorithm to evaluate and accommodate new features online. The approach is also more general than that presented by Amidi (Amidi and Fujita, 1999) who assumed flat ground, and more computationally efficient than that presented by Langelaan (Langelaan, 2007). Furthermore, these two approaches were not flight-tested. The autonomous landing demonstrated in this paper is performed without any artificial features, as were used in (Merz et al., 2004), and without GPS, as was used in (Chamberlain et al., 2011). The results presented in Scherer (Scherer et al., 2012) demonstrate impressive navigation performance in intermittant GPS conditions with a stereo camera sensor. However, a stereo camera sensor is not as standard a payload as a monocular camera, and may not always be practical. Additionally, the closed-loop flight results using vision aided navigation presented in (Weiss et al., 2011; Achtelik et al., 2011; Weiss et al., 2012) papers are limited to about 8 mins, whereas , the vision aided INS results presented in our work extend well over double of that time. Finally our results are also unique because there are no similar results for RMAX class UAVs, and over the time and distance demonstrated in this paper.

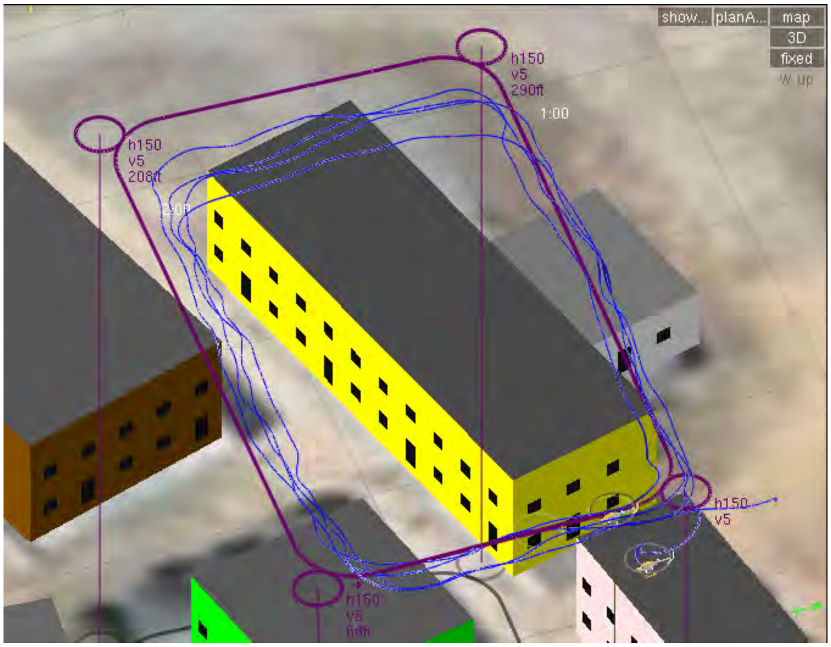
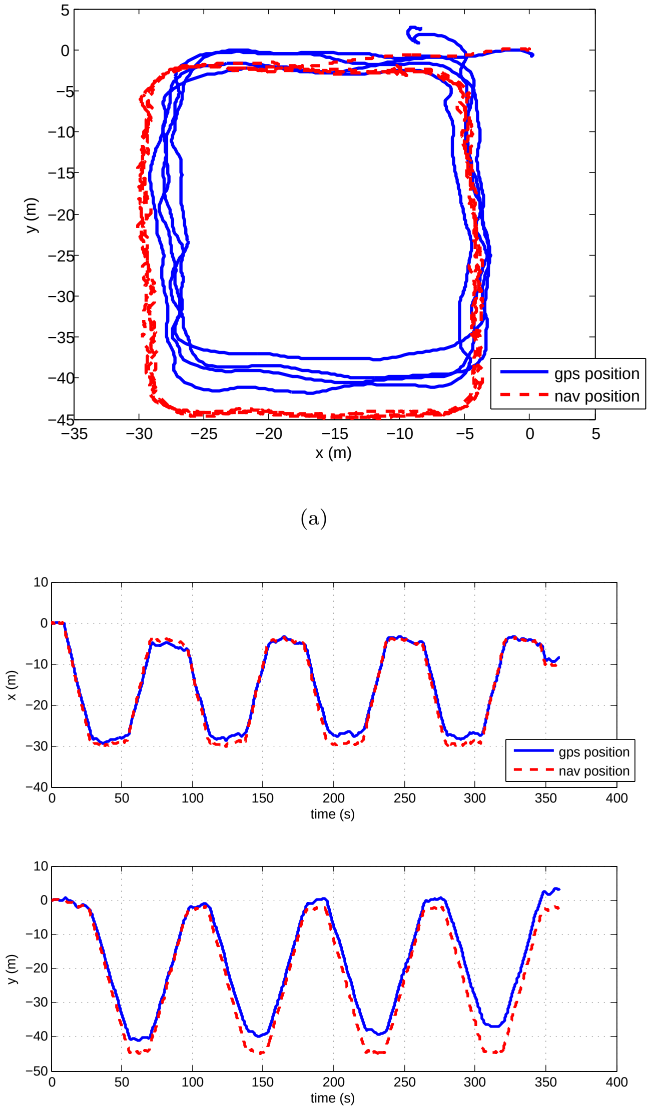


Figure 11: Flight path during a simulated flight over a group of buildings. The vehicle flew at an alittude of 50 m and and buildings were as tall as 10 m. The vehicle flew four times around the path indicated by the purple waypoints, for a total linear distance of 560 m. The true path of the vehicle is shown in blue, and the final error between the true position and navigation solution was 5 m.

6.2 EASE

The EASE is a ducted-fan electric vehicle with a mass of approximately 1 kg designed for use in both indoor and outdoor environments. The EASE has two independently controlled coaxial counter-rotating propellers as well as four independent control surfaces in the duct for generating control moments. The sensors on the vehicle are an Analog Devices 16405 IMU sampled at 100 Hz, a sonar for altitude, and a video camera. The video camera is mounted underneath the vehicle, with the optical axis aligned with the body z-axis. The camera outputs analog video at 30 frame per second and 640×480 resolution. The



(a)

(b)

Figure 12: Plots illustrating the navigation solution (red) and GPS position (blue) during a simulated flight over non-flat ground.

images are down-sampled to 320×240 resolution for use in the navigation algorithm. The camera field of view is 41 degrees, and distortion coefficients were calculated in MATLAB using the Camera Calibration Toolbox (Cam,). The vehicle receives power and communicates with the ground control station over a microfilament, which currently can be extended up to 50m. The microfilament also serves as an analog video link. The main reason for the microfilament is to overcome flight endurance limitations (less than 8 mins) imposed by state-of-the-art batteries. The power transferred through the cable allows the ease to operate over indefinite interval. Because this cable is already in place, the vision aided processing was performed off-board using an analog video link bootstrapped onto this cable. Since the video link is analog, it does not induce any more delay in transmitting the images than if the data were processed onboard. The V-INS and the position control algorithms are run on a computer on the ground and actuator commands are sent to the vehicle. The vehicle itself runs a high-frequency (>600 Hz) innerloop rate controller with a washout filter for attitude feedback. The specifications of the GTMAX, two single core 1.6 GHz pentium M computers, are comparable to the latest dual-core atom processors which could be easily carried by the EASE. Hence, the ability of the platform to carry required computational resources are not the limiting factor in implementing our algorithms onboard the EASE. Figure 1 shows a picture of the EASE. The baseline controller for the EASE consists of an adaptive attitude and velocity controller that is based on the GTMax adaptive flight controller (see (Johnson and Kannan, 2005)) and is implemented offboard. In addition, it features a high bandpass rate damping loop implemented onboard. Similar to GTMax, the EKF described in Section 4 processes vision data and makes updates based on vision at 20 Hz, and also features a sequential update.

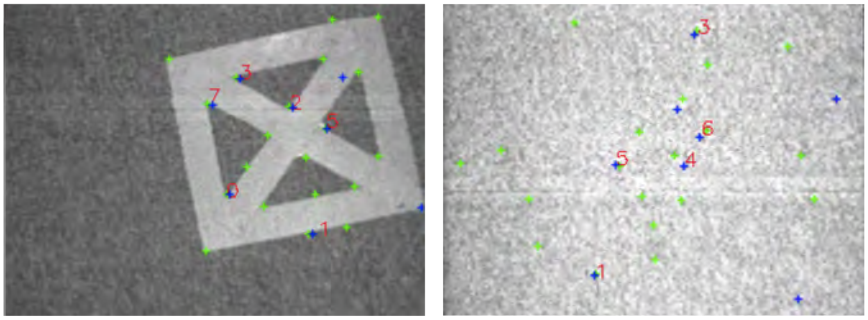
Potential applications of the EASE include indoor reconnaissance in unfriendly environments, first-response operations, and counter-terrorism operations. Therefore, the intended operating environment for EASE consists of gusty, unmapped, cluttered indoor environments without any external sensor support. In the results presented here, the EASE is evaluated under these conditions. In the following tests, the EASE performs autonomous hover and translations in still air, and hover in 2.4 m/s wind. The vehicle flew over a carpeted floor, and typical camera images are shown in Figure 13. Up to 16 points were stored in the database of feature locations, and the feature detector output up to 20 features for each image. For comparison, data was collected using a motion capture system (VICON) which gives sub-millimeter position and sub-degree attitude accuracy at 100 Hz. Note that for indoor operation, there is no direct mean of checking whether a feature is on the ground or on the wall when initializing a feature using Algorithm 1. However, since our approach incorporates the estimation of the altitude of the feature, it can converge to the correct location of features if they remain in view.

Figures 14 and 15 show the results of a 40 second hover without wind. The vehicle remains within a 16 cm diameter circle during the test, and the navigation position and motion capture position coincide closely throughout. This test is useful in establishing a baseline for the closed-loop control performance with the V-INS.

Figures 16 and 17 show the hovering performance of the vehicle in the presence of external disturbances (wind). The additional vehicle motion caused by the wind makes vision-based navigation challenging, hence both algorithms have to work in unison for the system to keep hovering. The wind was generated indoors using a 3 m diameter fan, and wind speed approximately 2.4 m/s, +/- .4 m/s. The navigation solution is again in agreement with the motion capture output, despite more vehicle and hence camera motion. As expected, the vehicle holds position less precisely, having a diameter of motion of approximately .76 m. Note in Figure 17 the approximately -10 degree pitch angle needed to hold position in the wind, and the smooth transition to a 10 degree roll angle as the vehicle follows the yaw command.

Figure 18 illustrates the response of the vehicle to a commanded translation. The 1.2 m command moved the vehicle such that no points in the original database were visible, and the entire database was replaced with points initialized in real time during the maneuver. As can be seen in the plots, the navigation solution incurs a small amount of offset in the y direction and in yaw due to accumulated error.

Figure 19 shows a sequence of images from a flight of the EASE in a cluttered indoor environment. The vehicle is flown by an operator who gives position commands to the vehicle using a joystick, and the vehicle autonomously tracks these position commands. It can be seen that the navigation solution is robust to ground clutter, such as the extension cord on the floor in (b) and robust to the change in carpet texture as the vehicle enters the room between (b) and (c). Furthermore, this demonstrates the ability of the EASE to



(a)

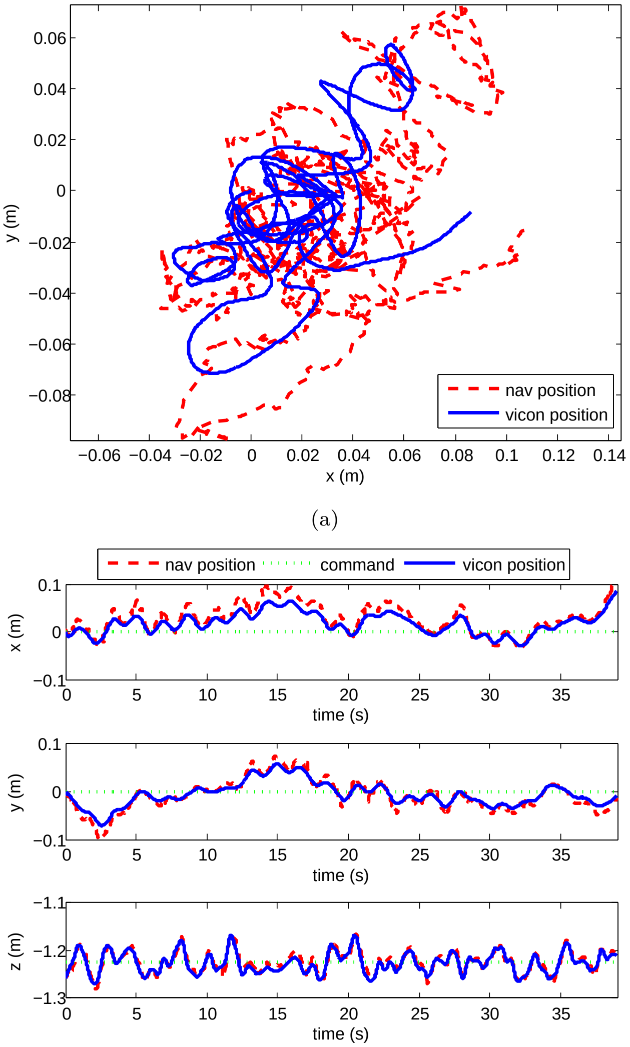
(b)

Figure 13: Typical images from the EASE camera used in navigation algorithm. The green crosses are features output from the Harris corner detector, and the blue crosses are database feature locations projected onto the image plane.

be controlled in confined spaces such as narrow hallways and doorways. The vehicle traveled approximately 30 m over the course of this flight.

6.2.1 Comparison with existing V-INS techniques for EASE class UAVs

The results presented on the EASE demonstrate several important advances over other work in the literature. Work such as Ahrens et al. (Ahrens et al., 2009) and Madison et al. (Madison and Bottkol, 2007) demonstrate comparable hover results, but in a laboratory setting (IMU data is simulated with a motion capture system)



(a)

(b)

Figure 14: Position data from an excerpt of 40 second hover of the EASE. Plots show that the navigation solution (red dashed line) is in good agreement with the motion capture system output (blue solid line).

and they do demonstrate flight over distances we have shown. The impressive work done in the robotics community in SLAM (Klein and Murray, 2007) (Davison et al., 2007) (Eade and Drummond, 2006) (Civera et al., 2008) (Sibley et al., 2010) promises to greatly increase the capability of vision-based navigation by generating accurate maps in addition to localization, but none of these papers demonstrate closed-loop control on a vehicle with fast dynamics. The visual odometry system presented in (Dille et al., 2009) shows results of extended range and duration, however the system is implemented on a ground vehicle. Frietsch et al. (Frietsch and Trommer, 2008) present a visual odometry system that enables vehicle to maintain stability in the absence of GPS, but their system does not demonstrate the long distance flight capability that we show in this paper. Furthermore, Frietsch et al. did not present flight-test results.

7 Conclusion

A Unmanned Aerial Vehicle vision aided Inertial Navigation System (V-INS) for sustained dynamic operation in GPS-denied environments was presented and validated in autonomous flight. Visual feature locations were estimated using a Harris corner detector, their frame-to-frame transitions were predicted using inertial measurements, and compared with observed locations to bound the INS drift, correct for inertial sensor biases and misalignment errors using an Extended Kalman Filter. A novel and efficient algorithm to manage a library of features online was presented and was designed to add or remove features based on a measure of relative confidence in each feature location. The ability of the system to maintain a database of trustworthy navigable features allowed the system to sustain dynamic operation during waypoint following autonomous

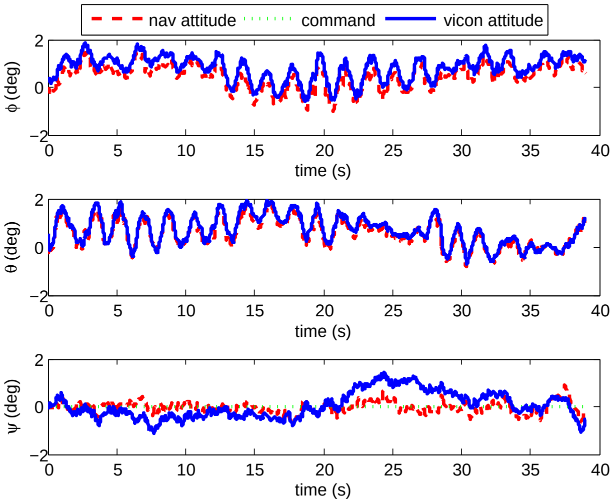


Figure 15: Attitude data from a 40 second hover of the EASE.

flight in a completely GPS-denied dynamic environment without any *a priori* known feature locations. The presented algorithms are validated through flight tests on a M-UAV operating indoors and a 66 kg rotorcraft operating outdoors in simulated GPS-denied environments. The method presented here was shown to overcome the shortcomings of existing V-INS algorithms such as computational inefficiency, lack or reliability, or lack of proven flight-test results. Therefore, these results establish the feasibility of using V-INS for sustained autonomous operation in both indoor and outdoor GPS-denied environments.

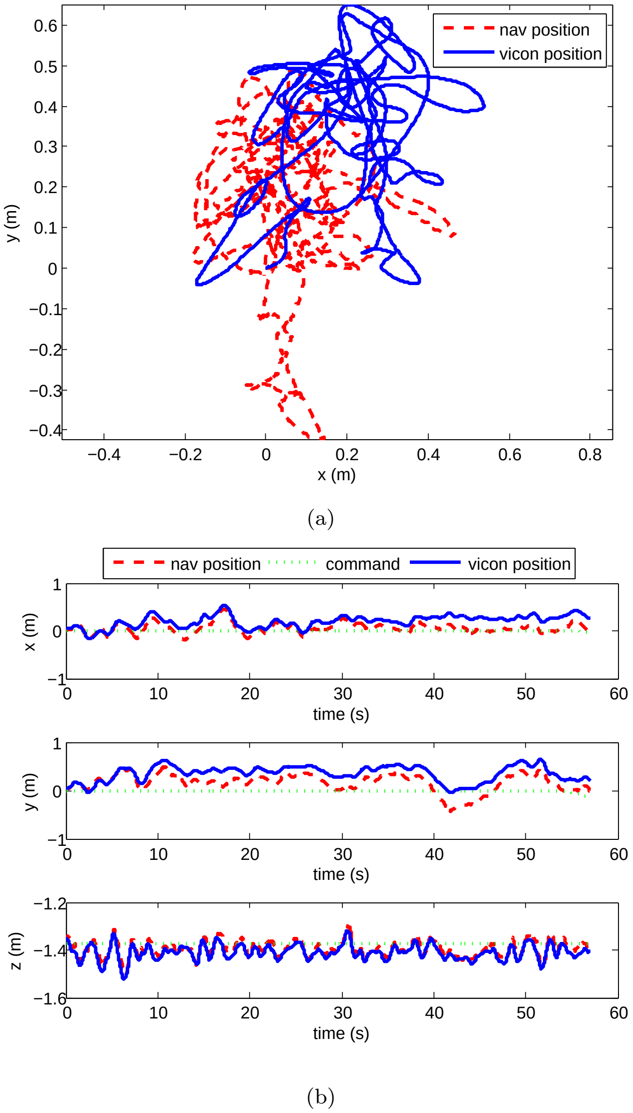


Figure 16: Position data from an excerpt of 57 second hover of the EASE in 2.4 m/s wind.

Acknowledgments

This work was performed under the NIST grant no. $70NANB10H013$. The authors wish to thank Eohan George, and the team at CyPhy works, including Helen Greiner, Jason Walker, Alexey Zaparoavvny, and Felipe Bohorquez.

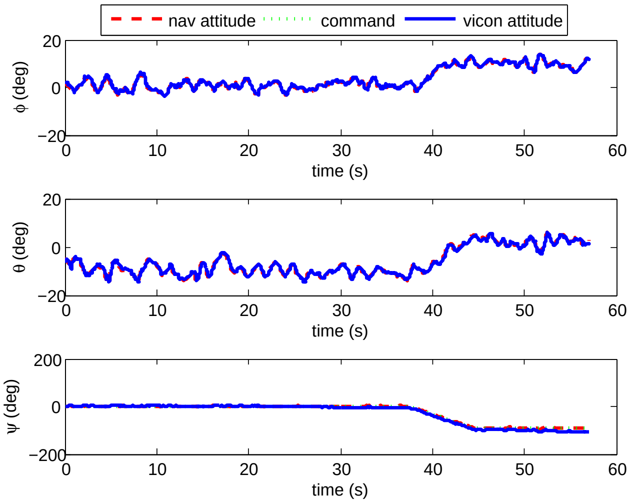
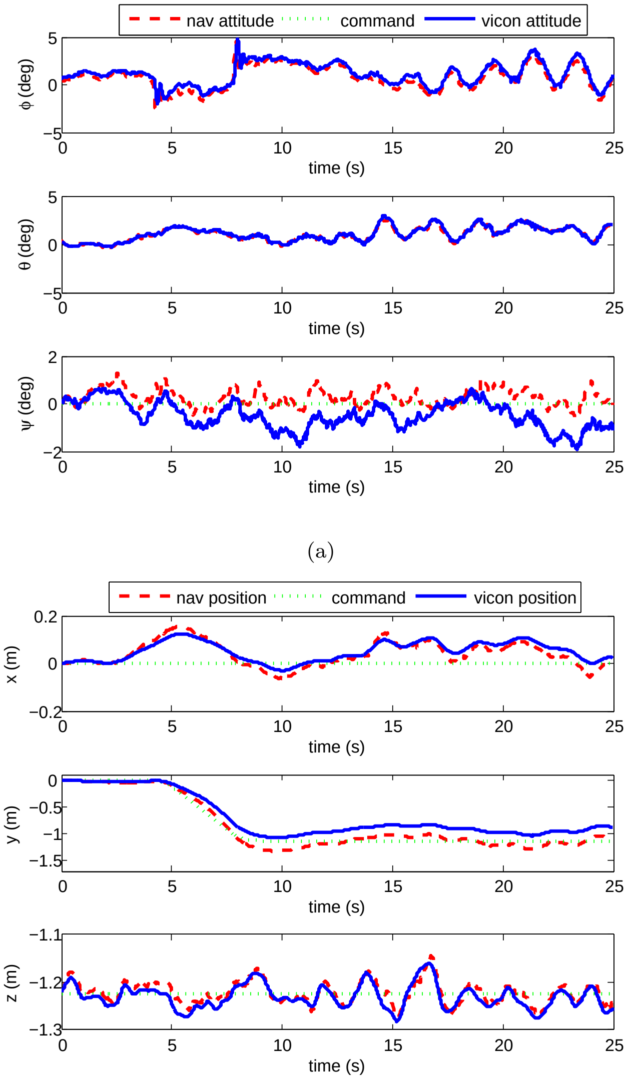


Figure 17: Attitude data from an excerpt of 57 second hover of the EASE in 2.4 m/s wind. The wind causes considerably more variation in the attitude than was seen in still air. Note the commanded yaw at 37 seconds, and the corresponding change in pitch angle to a roll angle.



(a)

(b)

Figure 18: Attitude and position data illustrating the response of the EASE to a commanded translation of 1.2 m.

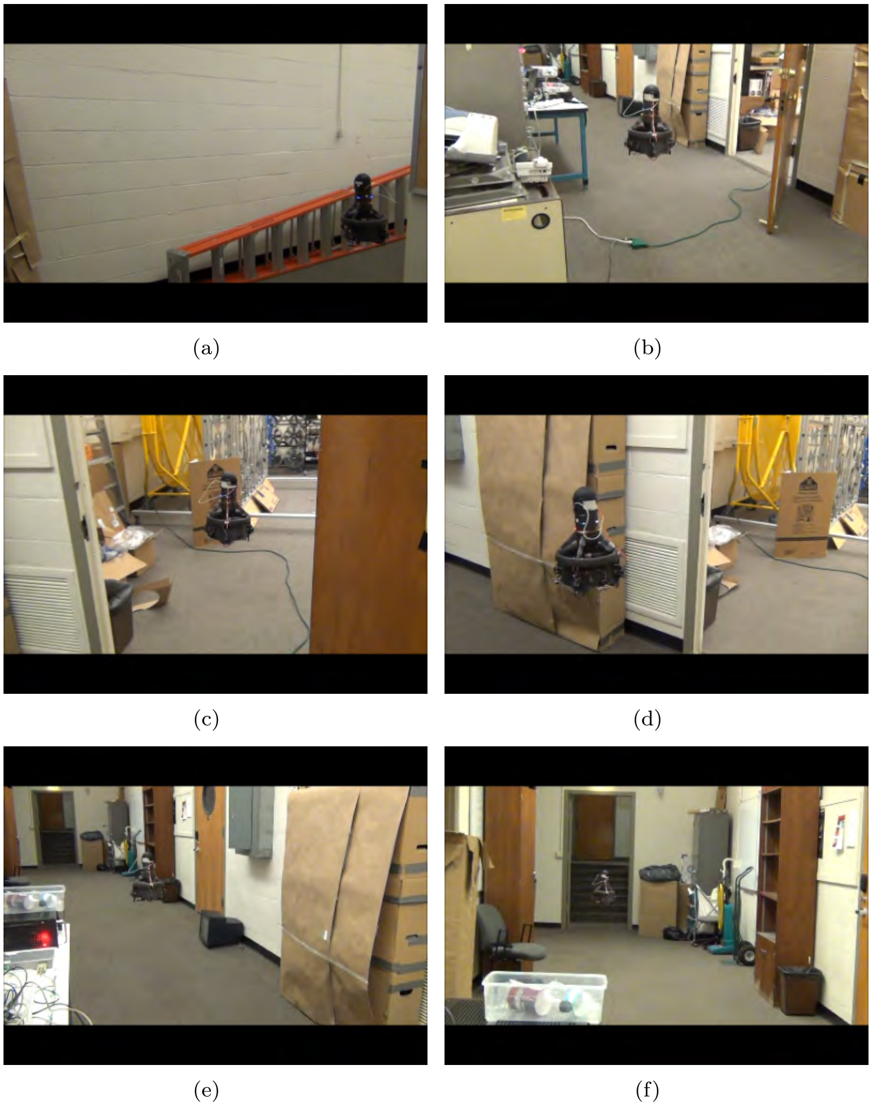


Figure 19: Sequence of images from a flight of the EASE in a cluttered hallway. Note the extension cord on the floor in (b), the change in carpet textures between (b) and (c), and the narrow corridors throughout

References

Achtelik, M., Achtelik, M., Weiss, S., and Siegwart, R. (2011). Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3056 –3063.

Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N. (2009a). Autonomous navigation and exploration of a quadrotor helicopter in gps-denied indoor environments. In *Proc. of the 1st Symposium on Indoor Flight, International Aerial Robotics Competition*.

Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N. (2009b). Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-Denied Indoor Environments. In *Proceedings of SPIE*, volume 7332.

Ahrens, S., Levine, D., Andrews, G., and How, J. (2009). Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments. In *International Conference on Robotics and Automation*, pages 2643–2648. IEEE.

Amidi, O., K. T. and Fujita, K. (1999). A visual odometer for autonomous helicopter flight. *Robotics and Autonomous Systems*, 29:185–193.

Bachrach, A., Prentice, S., He, R., and Roy, N. (2011). Range - robust autonomous navigation in gps-denied environments. *Journal of Field Robotics*, 28(5):644–666.

Blösch, M., Weiss, S., Scaramuzza, D., and Siegwart, R. (2010). Vision based mav navigation in unknown and unstructured environments. In *International Conference on Robotics and Automation*, pages 21–28. IEEE.

Chamberlain, L. J., Scherer, S., and Singh, S. (2011). Self-aware helicopters: Full-scale automated landing and obstacle avoidance in unmapped environments. In Whaley, M., editor, *AHS Forum 67*.

Chowdhary, G., Sobers, D. M., Pravitra, C., Christmann, C., Wu, A., Hashimoto, H., Ong, C., Kalghatgi, R., and Johnson, E. N. (2011a). Integrated guidance navigation and control for a fully autonomous indoor uas. In *Guidance Navigation and Control Conference*, Portland, OR. AIAA.

Chowdhary, G., Sobers, D. M., Pravitra, C., Christmann, C., Wu, A., Hashimoto, H., Ong, C., Kalghatgi, R., and Johnson, E. N. (2011b). Self contained ranging sensor aided autonomous guidance, navigation, and control for indoor flight. Portland, OR. AIAA.

Christophersen, H. B., Pickell, W. R., Neidoefer, J. C., Koller, A. A., Kannan, S. K., and Johnson, E. N. (2006). A compact guidance, navigation, and control system for unmanned aerial vehicles. *Journal of Aerospace Computing, Information, and Communication*, 3.

Civera, J., Davison, A., and Montiel, J. (2008). Inverse depth parametrization for monocular slam. *Robotics, IEEE Transactions on*, 24(5):932 –945.

Conte, G. and Doherty, P. (2009). Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP Journal of Advanced Signal Process*, 10.

Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403 –1410 vol.2.

Davison, A., Reid, I., Molton, N., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052 –1067.

Dille, M., Grocholsky, B. P., and Singh, S. (2009). Outdoor downward-facing optical flow odometry with commodity sensors. In *Proceedings Field & Service Robotics (FSR '09)*.

Eade, E. and Drummond, T. (2006). Scalable monocular slam. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 469 – 476.

Fowers, S.G., T. B. L. D. and Archibald, J. (2008). Vision-guided autonomous quad-rotor helicopter flight stabilization and control. In *Unmanned Systems North America Conference*. AUVSI.

Frietsch, N., M. O. S. C. S. J. and Trommer, G. (2008). Vision based hovering, and landing system for a vtol-mav with geo-localization capabilities. In *Guidance Navigation and Control Conference*. AIAA.

Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press, Cambridge.

Gelb, A., Joseph F. Kasper, J., Raymond A. Nash, J., Price, C. F., and Arthur A. Sutherland, J. (1974). *Applied Optimal Estimation*. The M.I.T. Press.

Grzonka, S., G. G. and Burgard, W. Towards a navigation system for autonomous indoor flying. In *Robotics and Automation, 2009. Proceedings IEEE International Conference on*, pages 2878–2883.

Grzonka, S., Grisetti, G., and Burgard, W. (2012). A fully autonomous indoor quadrotor. *Robotics, IEEE Transactions on*, 28(1):90 –100.

Haines, J. (2011). Vision-based control of an aerial vehicle. *Master's thesis, Carnegie Mellon University*.

Harris, C. and Stephens, M. (1988). *A combined corner and edge detector*, volume 15, pages 147–151. Manchester, UK.

Johnson, E. and Kannan, S. (2005). Adaptive trajectory control for autonomous helicopters. *Journal of Guidance Control and Dynamics*, 28(3):524–538.

Johnson, E. N., Proctor, A. A., Ha, J., and Tannenbaum, A. R. (2004). Development and test of highly autonomous unmanned aerial vehicles. *AIAA Journal of Aerospace Computing, Information and Communication*, 1(12):485–501.

Kayton, M. and Fried, W. R. (1997). *Avionics Navigation Systems*. John Wiley and Sons.

Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378.

Kitt, B. M., Rehder, J., Chambers, A. D., Schonbein, M., Lategahn, H., and Singh, S. (2011). Monocular visual odometry using a planar road model to solve scale ambiguity. In *Proc. European Conference on Mobile Robots*.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225 –234.

Langelaan, J. (2007). State estimation for autonomous flight in cluttered environments. *AIAA Journal of Guidance Navigation and Control*, 30(5):1414–1426.

Lefferts, E., Markley, F., and Shuster, M. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150 –1157 vol.2.

Madison, R., A. G. D. P. R. S. and Bottkol, M. (2007). Vision-aided navigation for small uavs in gps-challenged environments. In *Infotech@Aerospace Conference*. AIAA.

Merz, T., Duranti, S., and Conte, G. (2004). Autonomous landing of an unmanned helicopter based on vision and inertial sensing. In *In Proc. of the 9th International Symposium on Experimental Robotics*.

Mourikis, A., Trawny, N., Roumeliotis, S., Johnson, A., Ansar, A., and Matthies, L. (2009). Vision-aided inertial navigation for spacecraft entry, descent, and landing. *Robotics, IEEE Transactions on*, 25(2):264 –280.

Pears, N. E. (2000). Feature extraction and tracking for scanning range sensors. *Robotics and Autonomous Systems*, 33(1):43 – 58.

Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., and Singh, S. (2012). River mapping from a flying robot: state estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33:189–214. 10.1007/s10514-012-9293-0.

Shen, S., M. N. and Kumar, V. Autonomous multi-floor indoor navigation with a computationally constrained micro aerial vehicle. In *Robotics and Automation, 2011. Proceedings IEEE International Conference on.*

Sibley, G., Mei, C., Reid, I., and Newman, P. (2010). Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980.

Strasdat, H., Montiel, J., and Davison, A. (2010). Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657 –2664.

Strelow, D. and Singh, S. (2004). Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157 – 1195.

Theodore, C., Rowley, D., Ansar, A., Matthies, L., Goldberg, S., Hubbard, D., and Whalley, M. (2006). Flight trials of a rotorcraft unmanned aerial vehicle landing autonomously at unprepared sites. In *American Helicopter Society 62nd Annual Forum*, Phoenix, AZ. American Helicopter Society.

Tomasi, C. (1991). Detection and tracking of point features technical report cmu-cs-91-132. *Image Rochester NY*, 104(April):1–22.

Weiss, S., Achtelik, M., Lynen, S., Chli, M., and Siegwart, R. (2012). Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 957 –964.

Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874.

Wendel, J., Meister, O., Schlaile, C., and Trommer, G. F. (2006). An integrated gps/mems-imu navigation system for an autonomous helicopter. *Aerospace Science and Technology*, 10(6):527 – 533.

Wu, A. and Johnson, E. (2008). Methods for localization and mapping using vision and inertial sensors. In *Proceedings of AIAA GNC.*

Wu, A. and Johnson, E. N. (2010). Autonomous flight in gps-denied environments using monocular vision and inertial sensors. In *Infotech@AIAA*, Atlanta, GA. AIAA.

Wu, A. D., Johnson, E. N., Kaess, M., Dellaert, F., and Chowdhary, G. (2012). Autonomous flight in gps-denied environments using monocular vision and inertial sensors. *AIAA journal of aerospace computing, information, and communication*. accepted.

Wu, A. D., Johnson, N. E., Kaess, M., Dellaert, F., and Chowdhary, G. (2011). Autonomous flight in gps-denied environments using monocular vision and inertial sensors. *Journal of Aerospace Computing, Information, and Communication*. submitted.