JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 5, September–October 2007

State Estimation for Autonomous Flight in Cluttered Environments

Jack W. Langelaan*
Pennsylvania State University, University Park, Pennsylvania 16802

DOI: 10.2514/1.27770

Safe, autonomous operation in complex, cluttered environments is a critical challenge facing autonomous mobile systems. The research described in this paper was motivated by a particularly difficult example of autonomous mobility: flight of a small unmanned aerial vehicle through a forest. The focus was on enabling the three critical tasks that comprise flight: 1) maintaining controlled flight while avoiding collisions (*aviate*); 2) flying from a known start location to a known goal location (*navigate*); and 3) providing information about the environment—a map—to a human operator or other robots in the team (*communicate*). Presented here is a solution to the problem of estimating vehicle state (its position, orientation, and velocity) as well as the positions of obstacles or landmarks in the environment using only inertial measurements and bearings to landmarks. This is a highly nonlinear estimation problem, and standard estimation techniques such as the extended Kalman filter are prone to divergence in this application. In this paper a sigma-point Kalman filter is implemented, resulting in an estimator which is able to cope with the significant nonlinearities in the system equations and uncertainty in state estimates while remaining tractable for real-time operation. In addition, the issues of data association and landmark initialization are addressed. Estimator performance is examined through Monte Carlo simulations in two dimensions for scenarios involving unmanned aerial vehicle flight in cluttered environments. Simulations show that convergent, consistent estimates of vehicle state and obstacle positions can be obtained and that the estimates can be used by a trajectory planner to generate a path through a cluttered environment.

Introduction

SMALL autonomous unmanned aerial vehicles (UAVs) are a particularly challenging subset of mobile robots and autonomous vehicles. They undergo 6 degrees of freedom motion, are subject to significant external disturbances, require high bandwidth control, and have limited onboard sensing due to their small payload capacity. At the same time the missions envisioned for such vehicles are very challenging, involving low-altitude flight in obstacle-strewn terrain such as natural and urban canyons or forests. The cluttered environment further complicates the problem of control and navigation by greatly reducing the reliability of global positioning system (GPS) signals. A system which enables obstacle avoidance and navigation using only onboard sensing is therefore required.

This paper describes the theoretical development of a self-contained system to enable both control and navigation of small autonomous vehicles using only a low-cost microelectromechanical system (MEMS) inertial measurement units (IMU) and monocular vision, extended work reported earlier [1]. Results of Monte Carlo simulations examining system performance and results of a proof of concept hardware demonstration using a small unmanned ground vehicle (UGV) as a test bed are reported elsewhere [2].

Microelectromechanical IMUs have been commercially available for some time and have been used for sensing and stabilization in many applications. Their small size and low power requirements make them well suited to small UAV applications. However, two factors preclude purely inertial navigation solutions: first, IMUs do not provide information about nearby obstacles; second, their noise and bias characteristics lead to rapid unbounded drift in the computed position. Additional sensors are therefore required.

Presented as Paper 5140 at the AIAA Guidance, Navigation and Controls Conference, Providence, RI, 16–19 August 2004; received 12 September 2006; revision received 24 January 2007; accepted for publication 25 January 2007.

*Assistant Professor, Department of Aerospace Engineering. Senior Member AIAA.

Autonomous Vehicle Control and Navigation

Vision for Control and Navigation

Vision is a particularly rich stream of data suitable both for sensing and providing data to a human operator. Charge-coupled device (CCD) cameras have become very small and lightweight and are thus suitable for use as a sensor on small vehicles. Vision systems can provide measurements to obstacles or landmarks in the environment to enable obstacle avoidance and to aid in computing vehicle position, orientation, and velocity for navigation.

A monocular vision system provides bearings to landmarks. By itself, a single bearing to a landmark does not provide enough information to localize it. However, multiple bearings taken from disparate vantage points allow triangulation to determine the landmark's position. Note that a stereo pair provides depth information, but the accuracy of the depth information is proportional to the separation of the cameras (baseline) and inversely proportional to the *square* of the actual distance to the feature. The size of the vehicle limits the baseline; hence range information to distant features will be highly uncertain. For example, a stereo pair with 30 cm baseline, 640 pixel horizontal resolution, 70 deg field of view, 600 mm focal length cameras has 1σ depth uncertainty of 1 m at 10 m, 1σ depth uncertainty of 4 m at 20 m.

Vision has been extensively studied for use as a sensor in estimation related applications. However past work has not dealt with estimating all the states necessary for flight in cluttered environments (i.e., vehicle state and obstacle states). Examples include structure from motion, vision augmented inertial navigation, real-time benthic navigation, and relative position estimation.

Structure from motion attempts to reconstruct the trajectory of the video camera and an unknown scene. An example of an application is described by Pollefeys [3], who describes reconstruction of archaeological sites using video from a hand-carried camera. However, structure from motion algorithms are typically formulated as batch processes, analyzing and processing all images in the sequence simultaneously. Although this will give the greatest accuracy of both the reconstructed scene and camera path, it does not

lend itself to real-time operation. Variants of structure from motion techniques to enable real-time operation have been applied to UAV control [4,5]. However, structure from motion is able to recover scene information and camera motion up to a scale factor, and an accurate dynamic model of the vehicle is required to help solve for the scale factor. These techniques are specific to the vehicle carrying the camera, and it is unclear how external disturbances will affect the result.

Research into vision augmented inertial navigation [6–8] is primarily concerned with estimating the vehicle state by fusing inertial measurements either with bearings to known fiducials or data from optical flow algorithms. A variant is presented by Diel [9], who uses epipolar constraints for vision-aided inertial navigation. Positions of unknown obstacles are not estimated.

Real-time benthic navigation using vision as the primary sensor is described by Marks [10]. Distance from a planar surface (i.e., the ocean floor or a canyon wall) is obtained using a sonar proximity sensor and texture correlation is used to determine position offsets relative to a reference image. This capability has been adapted to enable underwater station keeping [11] and has been extended to incorporate additional sensors [12]. However this technique only estimates vehicle position, not the position of obstacles in the environment.

Huster [13] demonstrated fusion of inertial and monocular vision measurements for an underwater object retrieval task. In this case only one object in the environment was considered and relative position estimation (between the vehicle and object) was performed, not absolute vehicle and obstacle position estimation.

The use of vision for aiding UAV navigation has become an active area of research. In many cases vision is not the primary navigation/control sensor but is used in conjunction with inertial navigation systems and GPS to increase situation awareness. For example, Amidi [14] describes vision-aided navigation for an autonomous helicopter where a stereo pair is used to aid in station keeping. Sinopoli [15] describes a system that uses data from fused GPS/INS (inertial navigation system) and a digital elevation map to plan coarse trajectories which are then refined using data from a vision system. Roberts [16] describes a flight control system for a helicopter that uses a stereo pair to determine altitude and optical flow to determine ground speed. Vision-aided landing on a pad of known size and shape is described by Saripalli [17]. A more complex system for identifying suitable terrain for landing an autonomous helicopter is described by Meingast [18], which uses fused GPS/INS for control and navigation and a stereo pair for determining local terrain characteristics.

Proctor [19] describes a vision-only landing system that performs relative state estimation with respect to a set of known fiducials. A constant velocity motion model is used to model camera motion. Wu [20] describes a vision-aided inertial navigation system that relies on measurements to a known target for vehicle state estimation. Both are examples of terrain aided navigation (TAN), where measurements to known landmarks are used to aid navigation. Initially unknown environments are not addressed.

Framework for Integrated Control and Navigation

In general, successful operation of a UAV (in any environment, cluttered or clear) involves three basic tasks:

1) The vehicle must maintain controlled flight while avoiding collisions with obstacles (the vehicle must *aviate*). This requires a means to determine the state of the vehicle and to detect and localize obstacles with enough accuracy that appropriate action can be taken.

2) It must find its way from the starting point to a goal location in a finite amount of time (the vehicle must *navigate*). This requires a means to localize the vehicle relative to the goal.

3) It must convey information about the environment to a human operator or other robots in the team (the vehicle must *communicate*). This requires a means of presenting data in a useful manner to human operators or other robots in the team.

The task of *aviation* could be accomplished by flying reactively: the vehicle maintains heading until an obstacle is detected, the vehicle maneuvers to avoid the obstacle, and then attempts to reacquire the desired heading. However, although this reactive flight is adequate for small numbers of well-spaced obstacles, intuition suggests that the limited field of view of most sensors will cause this approach to fail in more complex environments with densely packed obstacles. Obstacles which have been detected but which have then left the field of view may still be a potential source of collision as the aircraft maneuvers to avoid obstacles still in the field of view. Thus a means of accounting for obstacles which have left the field of view must be provided to plan safe maneuvers.

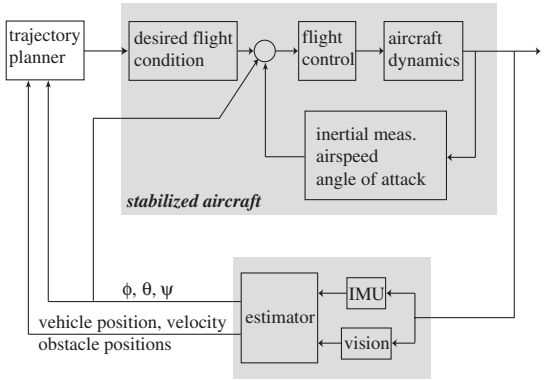

Fig. 1 Framework for a vision/inertial measurement navigation system. A *stabilized aircraft* is an aircraft that can maintain a desired flight condition. This may require measurements such as angular rates, angle of attack, sideslip angle, and airspeed. In addition, bank and pitch angles are required to stabilize certain longer-period dynamic modes, and more complex functions such as altitude hold require a measurement of altitude.

Although it is certainly aviating, purely reactive flight can hardly be said to be *navigation*: without knowledge of aircraft position there is no guarantee of reaching the goal. Thus to navigate in an obstacle-strewn environment some means of obtaining the position of the vehicle must be provided.

The ultimate purpose of exploratory flight is to communicate knowledge of the environment (i.e., a map) to a human operator or to other robots on the team. If the map is generated in real time as the vehicle flies through the environment it can also be used to aid aviation (because obstacle locations are computed) and navigation (because vehicle position in the map is computed).

A framework which enables aviation, navigation, and communication is introduced in Fig. 1. It comprises three parts: a trajectory planner, a stabilized aircraft, and an estimator. The trajectory planner uses knowledge of vehicle position and orientation and of the positions of nearby obstacles to compute a safe trajectory to the goal. A stabilized aircraft is one that can maintain a desired flight condition (determined based on the trajectory). This is enabled through measurements of variables such as angular rate and acceleration. Note that knowledge of other variables (such as the angle of bank to control spiral divergence) may be necessary to maintain controlled flight. Finally the estimator uses available sensing (in this research, an IMU and a monocular camera) to compute the data required for flight control and trajectory planning. In crewed aircraft the pilot provides the additional information required to maintain controlled flight and acts as trajectory planner. Position knowledge may be provided to the pilot by maps or navigation beacons such as GPS.

The framework presented in Fig. 1 can be generalized to other vehicles (such as autonomous underwater vehicles, or AUVs, and unmanned ground vehicles, or UGVs).

The problem of state estimation is directly tied to enabling a small UAV to aviate and navigate through the environment and to communicate its acquired knowledge. Hence, the primary focus of this paper is on developing an estimator which computes the variables necessary for control, obstacle avoidance, navigation, and mapping. Nonlinearities in the system models (both vehicle kinematics and the vision model) coupled with potentially large uncertainties in system states make this a particularly difficult

estimation problem. This is further exacerbated by the lack of observability in the system: a monocular vision system provides only bearings to obstacles, making multiple measurements from different vantage points necessary to localize it.

State Estimation

Estimating vehicle state as well as the positions of obstacles or landmarks in the environment is a simultaneous localization and mapping (SLAM) problem, a field of research which has received significant attention by the mobile robotics community in recent years. In a typical SLAM implementation the vehicle obtains measurements of ego motion (sometimes called *proprioceptive* measurements) and relative measurements (generally range and bearing) to nearby landmarks (called *exteroceptive* measurements). Extended Kalman filter (EKF) implementations of SLAM using range and bearing measurements have been applied both in simulation and on hardware in many scenarios including indoor navigation of small robots [21], subsea navigation by AUVs [22], outdoor navigation by wheeled robots [23], and navigation by aircraft [24]. In these cases range measurements to landmarks are available.

Bearings-only SLAM using cameras mounted on wheeled ground vehicles has also been investigated [25–27]. In these cases only pure 2-D motion is considered and estimation is performed for the planar environment.

Davison [28] describes a bearings-only SLAM implementation using only a monocular camera. Again an EKF is used to recursively estimate the state of the camera and of landmarks, and a constant velocity model with unknown acceleration is used to describe camera motion. By itself this system will be able to estimate motion and landmark positions up to a scale factor. To determine the scale factor the system is initialized by viewing a set of known landmarks. In unexplored environments, however, there are no known landmarks that can be used to determine the scale and some other means must be employed.

Burschka [29] describes a fused vision/inertial navigation system for off-road capable vehicles. Here the main focus is on vehicle pose estimation: a map of the environment is not maintained. Foxlin [30] describes a wearable vision/inertial system for self-tracking that uses unique coded fiducials for indoor tracking of humans. A range estimate is computed based on the size of the fiducial in the image plane.

State Estimation Problem

As discussed in the Introduction, the scenario considered here consists of a small UAV flying through an unsurveyed forest (Fig. 2) using only an inertial measurement unit and a monocular camera. The onboard camera obtains bearing measurements to obstacles (tree trunks) and the inertial measurement unit provides accelerations and angular rates in the body-fixed frame.

For the purpose of this research, flight control is taken to refer only to the maintenance of steady, controlled flight (the first part of the aviate task, the second part is obstacle avoidance). Navigation refers to directed motion toward a goal. In general, the information required for flight control differs from that required for navigation, and often the computations and actions required for flight control occur at a much higher rate than those for navigation. In general, angular rate, orientation, and speed are required for flight control, vehicle position is required for navigation, and obstacle relative position is required for obstacle avoidance.

Low-cost IMUs are subject to scale factor and bias errors that can drift with time; thus estimates of scale factor and bias are required in addition to vehicle position, orientation, and speed. Therefore the vehicle state vector is

$$v = \begin{bmatrix} x & y & z & \phi & \theta & \psi & u & v & w & \boldsymbol{\alpha}^T & \mathbf{b}_a^T & \mathbf{b}_\omega^T \end{bmatrix}^T \quad (1)$$

Referring to Fig. 2, $(x\ y\ z)$ represents position in the inertial frame, $(\phi\ \theta\ \psi)$ represent Euler angles with respect to the inertial frame,

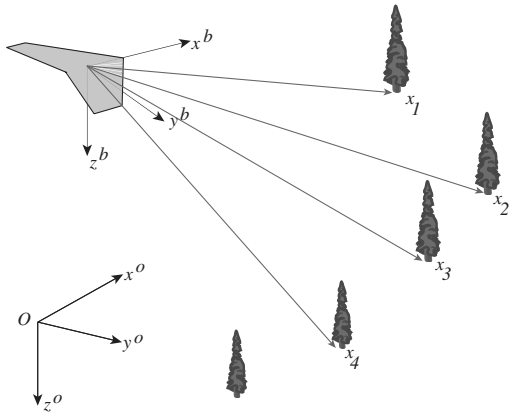

Fig. 2 Schematic of estimation problem. The aircraft obtains bearings to fixed landmarks (tree trunks) and measurements of acceleration and angular rate. Using these measurements an estimate of aircraft position, orientation, and velocity as well as obstacle positions must be obtained.

$(u\ v\ w)$ represents velocity expressed in the body frame, $\boldsymbol{\alpha}^T$ represents the IMU scale factor error, $\mathbf{b}_a^T$ represents accelerometer bias, and finally $\mathbf{b}_\omega^T$ represents rate gyro bias.

In addition to vehicle state, obstacle relative positions are required. In this research absolute obstacle positions in the inertial frame are estimated. This simplifies the mapping process and, as will be further discussed in the Estimator Design section, simplifies the computational requirements of the resulting estimator. Obstacle relative position can easily be computed from the absolute obstacle position and vehicle absolute position.

The final state vector is

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_v^T & \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_m^T \end{bmatrix}^T \quad (2)$$

where $\mathbf{x}_v$ is the vehicle state defined in Eq. (1), and $\mathbf{x}_i = [x_i\ y_i\ z_i]^T$ is the position of the ith obstacle in the inertial frame.

Given the noisy, limited measurements available from the IMU and vision system, the problem is to obtain the information required to control the aircraft, avoid collisions with obstacles, and to permit navigation. That is, the problem is to compute an estimate $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$ of the state vector $\mathbf{x}$ given a process model

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (3)$$

and a measurement model

$$\mathbf{z}_{\text{imu}} = g_1(\mathbf{x}, \mathbf{u}) \quad (4)$$

$$\mathbf{z}_{\text{cam}} = g_2(\mathbf{x}) \quad (5)$$

Here $\mathbf{u}$ represent inputs to the plant, $\mathbf{z}_{\text{imu}}$ represent inertial measurements, and $\mathbf{z}_{\text{cam}}$ represents bearing measurements. The following sections develop the process model f, the inertial measurement model g_1, the vision model g_2, and integrate the models to form the prediction and the vision update equations.

Sensor and System Models

Coordinate Frames

Navigation is done with respect to an inertial North–East–Down (NED) coordinate frame O. Sensors are fixed to the vehicle with known position and angular offsets with respect to a body-fixed frame B. Acceleration and angular rate are measured using a strapdown inertial measurement unit in the body frame B, and bearings to landmarks are obtained in a camera frame C. Transformation matrices $\mathbf{T}$ and $\mathbf{T}_{\text{cam}}$ define the transformation of a vector expressed in O to B and a vector expressed in B to C, respectively. Coordinate frames are shown schematically in Fig. 3.

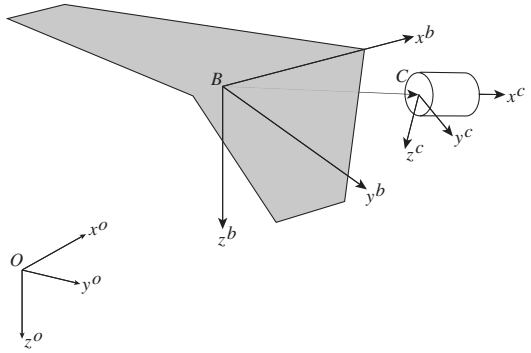

Fig. 3 Coordinate frames. Frame *O* is an inertial NED frame. *B* is the vehicle body-fixed frame, and the matrix T defines the transformation of a vector in *O* to its representation in *B*. Frame *C* is the camera-fixed frame, with the camera's optical axis aligned with x^c. The transformation $\mathbf{T}_{\text{cam}}$ between the camera frame and the body frame *B* is assumed known and the axes of the inertial measurement unit are assumed to be aligned perfectly with the body frame *B*.

Vehicle Kinematic Model

A dynamic model requires knowledge of all inputs, including disturbances. For small UAVs there is a very high degree of uncertainty associated with disturbances which act on the vehicle (i.e., gusts, which are extremely difficult to characterize in cluttered environments). In this case a standard technique is to use a kinematic model driven by inertial measurements as a process model.

Vehicle position x, y, z is expressed in the inertial frame; rotations are expressed as Euler angles ϕ, θ, ψ relative to the inertial frame; and velocity u, v, w is expressed in the body frame. The coordinate transform **T** [defined in Eq. (7)] projects a vector expressed in the inertial frame O into the body frame B. Vehicle kinematics are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{6}$$

The transformation matrix **T** is defined by the Euler angles of the aircraft with respect to the inertial frame. Following a roll–pitch–yaw convention,

$$\mathbf{T} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{7}$$

Body angular rates can be expressed as Euler angle rates by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{8}$$

Expanding Eqs. (6) and (8) gives

$$\dot{x} = \cos\theta\cos\psi u + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)v + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)w \tag{9}$$

$$\dot{y} = \cos\theta\sin\psi u + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)v + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)w \tag{10}$$

$$\dot{z} = -\sin\theta u + \sin\phi\cos\theta v + \cos\phi\cos\theta w \tag{11}$$

$$\dot{\phi} = p + \sin\phi\tan\theta q + \cos\phi\tan\theta r \tag{12}$$

$$\dot{\theta} = \cos\phi q - \sin\phi r \tag{13}$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r \tag{14}$$

Inertial Measurement Model

The inertial measurement unit includes accelerometers and rate gyros. The accelerometers measure specific force, which includes the acceleration of the vehicle and the projection of the acceleration due to gravity onto the body frame. The rate gyros measure the rotational velocity of the vehicle. Both sensors include sensor biases and zero-mean Gaussian random noise. Using the vehicle state vector defined in Eq. (1), the IMU measurement model g_1 can be written as

$$\mathbf{z}_a = \text{diag}(\boldsymbol{\alpha})\left[\frac{\text{d}}{\text{d}t}\mathbf{u}_a - \mathbf{T}\mathbf{g}\right] + \mathbf{b}_a + \mathbf{n}_a \tag{15}$$

$$\mathbf{z}_\omega = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{n}_\omega \tag{16}$$

where $\mathbf{z}_{\text{imu}} = [\mathbf{z}_a^T\ \mathbf{z}_\omega^T]^T$ is the IMU measurement vector and $\mathbf{u}_a$ represents the velocity of the accelerometer. The accelerometer scale factor correction is represented by $\text{diag}(\boldsymbol{\alpha}) \approx \mathbf{I}_{3\times3}$. The angular velocity $\boldsymbol{\omega}$ represents the components p, q, r of the vehicle angular velocity, expressed in the body frame.

The IMU is offset from the aircraft center of gravity (CG) by a known amount $\boldsymbol{\rho}$, hence

$$\mathbf{u}_a = \mathbf{u}_v + \boldsymbol{\omega} \times \boldsymbol{\rho} \tag{17}$$

where $\mathbf{u}_v = [u\ v\ w]^T$ denotes the velocity of the aircraft CG expressed in the body frame B. Taking the time derivative,

$$\frac{\text{d}}{\text{d}t}\mathbf{u}_a = \dot{\mathbf{u}}_v + \boldsymbol{\omega} \times \mathbf{u}_v + \dot{\boldsymbol{\omega}} \times \boldsymbol{\rho} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \boldsymbol{\rho} \tag{18}$$

The terms containing ρ can be collected into a single expression representing the accelerations induced by the offset of the IMU from the aircraft CG:

$$\frac{\text{d}}{\text{d}t}\mathbf{u} = \dot{\mathbf{u}}_v + \boldsymbol{\omega} \times \mathbf{u}_v + \mathbf{b}(\boldsymbol{\rho}) \tag{19}$$

Finally the accelerometer measurement model can be written as

$$\mathbf{z} = \text{diag}(\boldsymbol{\alpha})[\dot{\mathbf{u}}_v + \boldsymbol{\omega} \times \mathbf{u}_v + \mathbf{b}(\boldsymbol{\rho}) - \mathbf{T}\mathbf{g}] + \mathbf{b}_a + \mathbf{n}_a \tag{20}$$

Sensor biases and the accelerometer scale factor are assumed to vary by a random walk model with zero-mean Gaussian driving terms.

$$\dot{\boldsymbol{\alpha}} = \mathbf{n}_{\alpha} \tag{21}$$

$$\dot{\mathbf{b}}_{a} = \mathbf{n}_{b_a} \tag{22}$$

$$\dot{\mathbf{b}}_{\omega} = \mathbf{n}_{b_\omega} \tag{23}$$

that is, $\mathbf{n}_{(\cdot)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{(\cdot)})$.

Vision Model

The camera is assumed to be fixed to the aircraft with known offset $\Delta\mathbf{s}$ from the CG and known angular offset from the body-fixed frame, defined by a transformation $\mathbf{T}_{\text{cam}}$. The camera x axis is perpendicular to the image plane (coordinate frames are defined in Fig. 3).

A pinhole camera model describes the projection of a vector onto the image plane as

$$\mathbf{z} = \frac{f}{x}\begin{bmatrix} y \\ z \end{bmatrix} \tag{24}$$

where f is the focal length and $[x\ y\ z]^T$ is the vector (expressed in the camera frame). The focal length f can be normalized without loss of generality.

For cameras with "standard" field of view (less than approximately 70 deg) this model is sufficient. In wide field of view cameras ($\gtrsim$90 deg) this model becomes problematic. The pinhole projection model becomes ill conditioned for vectors which are close to 90 deg away from the optical axis (the component x of the vector expressed in the camera frame approaches 0). To improve conditioning and to express bearings as azimuth and depression (in the coordinate frames used here a positive angle is down with respect to the optical axis) measurements are modeled as arctangents of the projection onto the image plane (Fig. 4). For the ith landmark the vision measurement model $g_{2,i}$ is

$$\mathbf{z}_{\text{cam},i} = \begin{bmatrix} \arctan\frac{s_{i,y}}{s_{i,x}} \\ \arctan\frac{s_{i,z}}{s_{i,x}} \end{bmatrix} + \mathbf{n}_c \tag{25}$$

The measurement is corrupted by zero-mean Gaussian noise $\mathbf{n}_c$. $\mathbf{s}_i$ represents the vector from the camera to the ith tree, expressed in the camera frame:

$$\mathbf{s}_i = \mathbf{T}_{\text{cam}}\left[\mathbf{T}\begin{bmatrix} x_i - x \\ y_i - y \\ z_i - z \end{bmatrix} - \Delta\mathbf{s}\right] \tag{26}$$

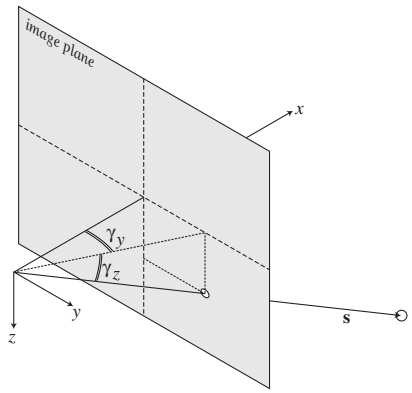

Fig. 4 Modified pinhole projection model.

When vision measurements to several landmarks are available the vision measurement vector is formed by concatenating the available measurements, that is, $\mathbf{z}_{\text{cam}} = [\mathbf{z}_{\text{cam},1}^T\ \mathbf{z}_{\text{cam},2}^T\ \cdots\ \mathbf{z}_{\text{cam},m}^T]^T$.

Estimator Design

Inertial/Vision Navigation Filter

In general, estimators follow a recursive process of prediction followed by correction. In many cases the prediction step is driven by a dynamic model of the system driven by known inputs and unknown disturbances, where it is assumed that the disturbances can be adequately modeled by random process noise. In this research the prediction step is performed by a kinematic model driven by accelerometer and angular rate measurements. Noise in these measurements is treated as process noise.

The correction step is driven by the bearing measurements obtained from the vision system.

Prediction Step

Incorporating the rate gyro measurement model [Eq. (16)] into Eqs. (12–14) relates measurements of angular rate to the Euler angle rates:

$$\dot{\phi} = (z_p - b_p) + \tan\theta\sin\phi(z_q - b_q) - \tan\theta\cos\phi(z_r - b_r) \tag{27}$$

$$\dot{\theta} = \cos\phi(z_q - b_q) - \sin\phi(z_r - b_r) \tag{28}$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta}(z_q - b_q) + \frac{\cos\phi}{\cos\theta}(z_r - b_r) \tag{29}$$

where b_p, b_q, and b_r are the rate gyro biases. The vehicle acceleration is obtained from the accelerometer model [Eq. (20)] and the rate gyro model [Eq. (16)]:

$$\dot{u} = \frac{z_x}{\alpha_x} - g\sin\theta - b_x - (z_q - b_q)w + (z_r - b_r)v - b_x(\boldsymbol{\rho}) \tag{30}$$

$$\dot{v} = \frac{z_y}{\alpha_y} + g\cos\theta\sin\phi - b_y - (z_r - b_r)u + (z_p - b_p)w - b_y(\boldsymbol{\rho}) \tag{31}$$

$$\dot{w} = \frac{z_z}{\alpha_z} + g\cos\theta\cos\phi - b_z - (z_p - b_p)v + (z_q - b_q)u - b_z(\boldsymbol{\rho}) \tag{32}$$

$z_{(\cdot)}$ and $b_{(\cdot)}$ represent the inertial measurement and bias, respectively, along a body-fixed axis. $\alpha_{(\cdot)}$ is the accelerometer scale factor error along a body-fixed axis, g is the acceleration due to gravity, and $b_{(\cdot)}(\boldsymbol{\rho})$ is the acceleration induced by the (known) offset of the IMU from the vehicle CG.

IMU biases vary as a random walk:

$$\dot{\boldsymbol{\alpha}} = \mathbf{n}_{\alpha} \tag{33}$$

$$\dot{\mathbf{b}}_{a} = \mathbf{n}_{b_a} \tag{34}$$

$$\dot{\mathbf{b}}_{\omega} = \mathbf{n}_{b_\omega} \tag{35}$$

where $\boldsymbol{\alpha} = [\alpha_x\ \alpha_y\ \alpha_z]^T$, $\mathbf{b}_a = [b_x\ b_y\ b_z]^T$, and $\mathbf{b}_\omega = [b_p\ b_q\ b_r]^T$.

Finally, obstacles are assumed to be stationary, hence

$$\dot{\mathbf{x}}_i = 0 \tag{36}$$

Equations (9–11) and (27–36) represent the kinematics of the vehicle, inertial measurement unit, and obstacles. These can be written compactly in discrete form as

$$\begin{bmatrix} \mathbf{x}_{v,k+1} \\ \mathbf{x}_{0,k+1} \end{bmatrix} = \begin{bmatrix} f_v^d(\mathbf{x}_{v,k}, \mathbf{z}_{\text{imu},k}) \\ \mathbf{x}_{0,k} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{\text{ext}} \\ 0 \end{bmatrix} \tag{37}$$

The nonlinear function f_v^d captures the discrete-form kinematics of the vehicle and the IMU bias states and is driven by the inertial measurements $\mathbf{z}_{\text{imu},k}$. The external noise term $\mathbf{n}_{\text{ext}}$ includes the assumed noise of the random walk model for the IMU scale factor and bias drift and may include an additional noise term to cover unmodeled vehicle kinematics. Inertial measurements are assumed to be corrupted by zero-mean Gaussian random noise.

The system kinematic equations are used in the prediction step of the unscented Kalman filter (UKF) algorithm given by van der Merwe [31]. The process noise for the time update must still be obtained. In this case process noise arises from the noisy measurements of acceleration and angular rate. Process noise is approximated as

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{F}_z \Sigma_{\text{imu}} \mathbf{F}_z^T + \Sigma_{\text{ext}} T & 0 \\ 0 & 0 \end{bmatrix} \tag{38}$$

where $\mathbf{F}_z = \nabla_{\mathbf{z}_{\text{imu}}} f_v^d$ (the Jacobian of the discrete process model with respect to the inertial measurements), Σ_{imu} is the covariance matrix describing the IMU measurement noise, and Σ_{ext} is the covariance describing the external noise, including drift of the IMU biases. This is identical to the EKF process noise approximation and is adequate for the application being considered. Because obstacles (i.e., trees) are assumed to be perfectly stationary only components corresponding to vehicle states have nonzero process noise.

Note that measurements from the IMU are likely to be available at a much higher rate than measurements from the camera, allowing the time update to proceed at a higher rate than the vision correction.

Vision Update

Measurements from the camera are incorporated in the measurement update step of the estimator. The vision model reflects the projection of a vector in 3-D (the vector from the vehicle to the feature) onto the 2-D image plane and is represented by Eqs. (25) and (26). This can be written compactly as

$$\mathbf{z}_{\text{cam},k} = g_2(\mathbf{x}_{v,k}, \mathbf{x}_{0,k}) + \mathbf{n}_c \tag{39}$$

Here $\mathbf{z}_{\text{cam}}$ represents bearings from the vehicle to stationary features. Measurement noise is represented by the zero-mean Gaussian random variable $\mathbf{n}_c$.

Choice of States

Recall from the State Estimation Problem section that only relative obstacle position is required to avoid collision. A different state vector including vehicle state and range and bearings to obstacles instead of absolute obstacle position would also provide the information required for control and navigation. Because the vision system provides a measurement of bearing to obstacles, this would result in a linear vision update step.

However, while absolute obstacle position is constant, relative position changes continuously and is a nonlinear function of vehicle motion, resulting in a more complex prediction step. The prediction step runs at the IMU measurement rate, which is typically much higher than the vision frame rate. Computationally, estimating obstacle relative position is more expensive than estimating absolute position. Absolute obstacle position estimation has the additional benefit of immediately providing a map of obstacle positions in the inertial frame.

Hence absolute obstacle position is estimated along with vehicle position, orientation, and velocity.

Estimation Algorithm

In many SLAM implementations an EKF is used. As will be shown later, in this particular example an EKF generally results in divergent estimates of vehicle and obstacle states.

Rather than approximate the system equations (as done by an EKF), sigma-point Kalman filters (SP-KFs) approximate the probability distribution of the state which is to be estimated [31,32]. In contrast with a particle filter, which makes no assumption about the probability distribution function (PDF), SP-KFs assume a Gaussian PDF of both state variables and noise. This allows a greatly reduced set of particles to model the PDF. Because they are more capable of coping with significant nonlinearities they have recently become popular in SLAM implementations [1,33,34]. Briefly, in an SP-KFs the distribution is modeled using a set of deterministically chosen *sigma points* which are propagated through the nonlinear system dynamic equations (the prediction step) and these propagated sigma points are used to compute a predicted mean and associated covariance. The propagated sigma points are then used to compute predicted measurement, which is used to compute both the innovations and the covariance associated with the innovations. These are then used in the measurement update step. The reader is referred to van der Merwe [31] for a more complete description.

Data Association

Inherent in any Kalman filter is an assumption of known data association. However, in this application (as in many SLAM implementations) landmarks are indistinguishable from one another, hence this must be computed explicitly. Note that here indistinguishable means that landmarks are not uniquely labeled. They are, however, geographically separated. This geographic separation is used to compute data association.

Typical data association algorithms are based on using a χ^2 test to compare an actual measurement with a prediction. After computing likelihoods of possible associations either a gated nearest neighbor approach is used to determine association on a landmark-by-landmark basis or a joint compatibility test [35] is used to determine the most likely overall set of associations.

The difficulty of data association in bearings-only SLAM lies in the small size of the measurement subspace (the 2-D image plane as opposed to the 3-D physical space). It is especially difficult when landmarks have not yet been localized to a high degree of accuracy. Previous work has proposed using additional information (e.g., in an indoor SLAM implementation described by Neira [27] the lengths of the vertical lines used as features are used as additional information for data association; Fitzgibbons [26] uses color) to assist in the process of data association. In this research additional identifying information is not available: the vision system provides only a bearing to a landmark.

A two-stage process is used for data association: first, the current bearings are compared with those obtained in a previous frame to check frame-to-frame correspondence; second, bearings to features not seen in the previous frame are compared with predicted bearings obtained from landmarks in the map to check if the features have been seen earlier in the run. Those bearings that are not associated in either step are assumed to come from a new, previously unseen landmark.

Frame-to-frame association for point features is used extensively in optical flow algorithms. Typically these operate at high frame rates compared with motion of the vehicle. However, when frame rates are low and vehicle motion (especially angular motion) is large, frame-to-frame correlation is more complex. However, additional information to aid frame-to-frame association is available.

Angular rate measurements available from the IMU are integrated to obtain the change in orientation between frames (shown schematically in Fig. 5). The change in orientation is used to calculate predicted bearings for features seen in the previous frame and these are compared with current measurements. Whereas changes in

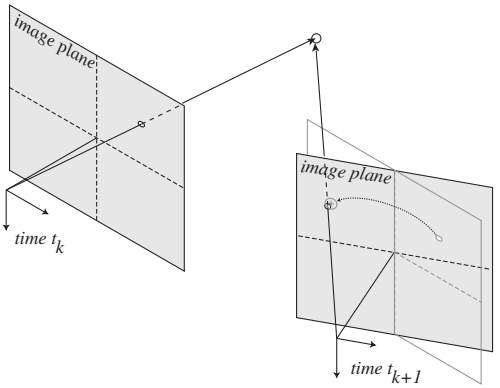

Fig. 5 Frame-to-frame data association. A bearing is obtained at t_k. The camera moves and another bearing is obtained at t_{k+1}. Rate gyro measurements are integrated to compute the camera rotation between frames and the expected bearing for t_{k+1} is computed.

position of the vehicle will also affect the predicted bearings, this is a much smaller effect than changes in orientation. Also, changes in bearing to a feature due to changes in vehicle position require a measurement of range (which is unavailable here) to the feature to compute. Hence bearing changes due to vehicle position change are not calculated but are covered by an increased uncertainty in the predicted bearing.

A bearing from the previous image is expressed as a vector in the camera frame. The rotation $\mathbf{T}_{f-f}$ between the previous frame and the current frame is computed by integrating angular rates and is used to compute the prediction of the bearing in the current frame:

$$\mathbf{c} = \mathbf{T}_{f-f}\begin{bmatrix} 1 \\ z_{y,k-1} \\ z_{z,k-1} \end{bmatrix} \tag{40}$$

The transformed vector is projected onto the image plane to determine the predicted bearing $\bar{\mathbf{z}}_k$.

$$\bar{\mathbf{z}}_k = \frac{1}{c_x}\begin{bmatrix} c_y \\ c_z \end{bmatrix} \tag{41}$$

Finally a gated nearest neighbor approach based on the Mahalonobis distance between the current bearing and predicted bearings is used to determine association:

$$d_{ij} = (\mathbf{z} - \bar{\mathbf{z}}_j)^T \mathbf{P}_{jj}^{-1} (\mathbf{z} - \bar{\mathbf{z}}_j) \tag{42}$$

The matrix $\mathbf{P}_{jj}$ is the covariance associated with the predicted bearing $\bar{\mathbf{z}}_j$ and includes measurement uncertainty, uncertainty induced by the transformation, and a term to cover the change in bearing due to vehicle position change, which is not incorporated explicitly.

Features which are not associated with landmarks in the frame–frame association step are passed to the frame–map association step. The frame-to-map association compares a current bearing with the predicted bearings computed using the current state estimate.

$$d_{ij} = (\mathbf{z} - \hat{\mathbf{z}}_j)^T \mathbf{P}_{zz,jj}^{-1} (\mathbf{z} - \hat{\mathbf{z}}_j) \tag{43}$$

The Mahalanobis distance is computed to assess the likelihood that a bearing is associated with a particular prediction and a gated nearest neighbor approach is used to assign bearings to map landmarks.

This two-step approach to data association is more robust to dropouts of individual features (which may be due to occlusion or to features which are on the edge of the vision system's detection envelope) and enables more accurate association when landmarks have not yet been accurately localized.

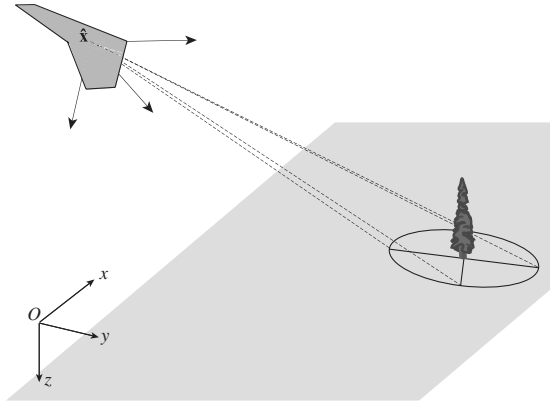

Fig. 6 Landmark initialization: A landmark is initialized at the intersection of the vector from the vehicle to the previously unseen landmark and the ground plane. The uncertainty is computed by projecting the uncertainty in the vector onto the ground plane.

Landmark Initialization

Features which are not associated with known landmarks are assumed to come from new, previously unseen landmarks. As with data association, landmark initialization is complicated by the lack of complete measurements to a new feature, and an algorithm for computing an initial estimate of landmark position is required.

Methods for feature initialization can be characterized as undelayed or delayed. Undelayed approaches [25,28,36] represent the conical probability distribution function of a single bearing as a series of Gaussians which are then pruned as more measurements become available. Delayed methods collect several bearings to a feature from different vehicle poses and compute a landmark position. It is difficult, however, to obtain an initial landmark position and covariance which is sufficiently Gaussian to prevent divergence of Kalman-type filters. Bailey [37] describes a method for constrained initialization which computes the "Gaussian-ness" by calculating the Kullback–Leibler distance. However this is expensive to compute and a threshold value had to be determined experimentally. Another approach is described by Fitzgibbons [26] and by Montesano [38], where particle filters are used for landmark placement until the distribution is sufficiently Gaussian to permit a switch to an EKF framework.

In the current implementation landmarks are assumed to lie on the ground plane. Using the estimate of the vehicle's altitude, a new landmark is initialized at the intersection of the ground plane and a vector from the vehicle along the bearing to the previously unseen landmark (obtained from the vision system). This is shown schematically in Fig. 6.

In the camera frame, a vector to the landmark is

$$\boldsymbol{\gamma}^C = \begin{bmatrix} 1 \\ \arctan z_y^c \\ \arctan z_z^c \end{bmatrix} \tag{44}$$

This can be transformed to the inertial frame by

$$\boldsymbol{\gamma}^O = \mathbf{T}^{-1}[\mathbf{T}_{\text{cam}}^{-1}\boldsymbol{\gamma}^C + \Delta\mathbf{s}] \tag{45}$$

where $\mathbf{T}_{\text{cam}}$ is known and $\mathbf{T}$ is computed from the estimates of Euler angles. Uncertainty is present in both the bearing measurement $[z_y\ z_z]^T$ and the vehicle orientation (described by $\mathbf{T}$). The vector $\boldsymbol{\gamma}^O$ points from the vehicle to the landmark. The landmark position can now be computed from the intersection of this vector with the ground plane, because

$$\boldsymbol{\gamma}^O = \gamma\begin{bmatrix} x_i - x \\ y_i - y \\ z_i - z \end{bmatrix} \tag{46}$$

where $[x\ y\ z]^T$ denotes the vehicle position in the inertial frame, $[x_i\ y_i\ z_i]^T$ denotes the landmark position in the inertial frame and γ is

a scale factor. All landmarks are assumed to lie on the ground plane, hence $z_i \sim \mathcal{N}(z_g, \Sigma_g)$. Therefore:

$$\gamma = \frac{\gamma_z^O}{z_g - z} \tag{47}$$

$$x_i = \frac{\gamma_x^O}{\gamma} + x \tag{48}$$

$$y_i = \frac{\gamma_y^O}{\gamma} + y \tag{49}$$

The uncertain parameters leading to the three equations above are vehicle position x, y, z; ground plane height z_g; vehicle orientation ϕ, θ, ψ [through the transformation in Eq. (45)]; and bearings z_y^C, z_z^C.

A sigma-point transform can be used to project the bearing and its associated uncertainty onto the ground plane. First a state vector is assembled from the current estimate of vehicle position and orientation, the ground plane height, and the bearing measurements:

$$\boldsymbol{\xi} = [\,x \quad y \quad z \quad \phi \quad \theta \quad \psi \quad z_g \quad z_y^C \quad z_z^C\,]^T \tag{50}$$

Similarly a covariance matrix is assembled:

$$\mathbf{P}_\xi = \begin{bmatrix} \mathbf{P}_{vv} & 0 & 0 \\ 0 & \Sigma_g & 0 \\ 0 & 0 & \mathbf{R} \end{bmatrix} \tag{51}$$

where $\mathbf{P}_{vv}$ is the 6×6 covariance associated with the current estimate of vehicle position and orientation, Σ_g is the variance of the uncertainty associated with the ground plane height, and $\mathbf{R}$ is the covariance of the uncertainty associated with the bearing measurement.

Using this state vector and associated covariance, a set of sigma points is computed. The sigma points are propagated through Eqs. (44–49) and finally the estimated landmark position $\hat{\mathbf{x}}_{\text{new}}$ and associated covariance $\mathbf{P}_{\text{new}}$ are computed. The system state vector is then augmented with the new landmark:

$$\hat{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{\text{new}} \end{bmatrix} \qquad \mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k} & 0 \\ 0 & \mathbf{P}_{\text{new}} \end{bmatrix} \tag{52}$$

While the augmented state covariance is initially block diagonal, the uncertainty in the new landmark position quickly becomes correlated with both the vehicle state uncertainty and the uncertainty in previously observed landmark positions.

This method of landmark initialization can be extended to nonplanar ground if the vehicle has a digital elevation map (DEM) of the surrounding terrain. A landmark would now get initialized at the intersection of the bearing vector and the ground as given by the DEM. There is one caveat however: as the grazing angle between the ground and the bearing becomes small the projected uncertainty will become very large. Additionally, the projected uncertainty will no longer be Gaussian, leading to potential problems when the landmark is added to the estimator.

Estimate Smoothing

Bearing measurements provide no information about vehicle absolute position or about landmark absolute position. Absolute position is therefore unobservable, and unless additional information in the form of absolute measurements are available the estimates of vehicle and landmark absolute position will drift.

In most SLAM implementations (including this one) the uncertainty in all the states becomes highly correlated over the course of vehicle motion. Hence additional information about any of the states can be used to improve the estimates of all the states. This information may come from many sources: loop closure, which consists of revisiting previously explored terrain; observation of a priori known landmarks; or sporadic GPS updates. Arguably observations of a priori known landmarks and GPS updates are equivalent: both are measurements to what are effectively beacons. With this additional information a smoothing algorithm can correct the drift in absolute vehicle and landmark positions.

Smoothing of the full vehicle trajectory provides the most accurate correction of the map. However, this requires enough data that the vehicle trajectory can be reconstructed: this may consist of all measurements (both inertial and bearings) obtained through the trajectory or it may consist of retaining vehicle state estimates and the associated covariance throughout the trajectory (i.e., rather than only including the current vehicle state estimate in the state vector, the current state and all previous states are included). The state vector will then grow with time, as current vehicle states are added, as well as with distance, as new landmarks are observed. Both techniques greatly increase both the storage and computational requirements of the smoothing step.

Rather than smoothing the full vehicle trajectory, a correction step can be applied using only the current estimate of vehicle state and obstacle positions. This results in a less effective map correction but does not require storage of all measurements or of all previous vehicle states.

Estimation happens in the state space (i.e., $\mathbf{x}$, $\mathbf{P}$). The correction happens in information space:

$$\mathbf{I} = \mathbf{P}^{-1} \qquad \mathbf{i} = \mathbf{I}\hat{\mathbf{x}} \tag{53}$$

where $\hat{\mathbf{x}}$ is the current estimate of the system state and $\mathbf{P}$ is the estimated covariance. The correction step is

$$\mathbf{I}_{\text{corr}} = \mathbf{I} + \mathbf{I}_{\text{add}} \tag{54}$$

$$\mathbf{i}_{\text{corr}} = \mathbf{i} + \mathbf{i}_{\text{add}} \tag{55}$$

where $\mathbf{I}_{\text{add}}$ is the information matrix and $\mathbf{i}_{\text{add}}$ is the information vector associated with the additional state knowledge. The corrected state and covariance are then recovered:

$$\mathbf{P}_{\text{corr}} = \mathbf{I}_{\text{corr}}^{-1} \qquad \hat{\mathbf{x}}_{\text{corr}} = \mathbf{P}_{\text{corr}}\mathbf{i}_{\text{corr}} \tag{56}$$

Data Flow

The flow of data through the estimation process is shown schematically in Fig. 7.

The estimator is initialized at time t_0 with

$$\hat{\mathbf{x}}_{0|-} = \mathbf{x}_0 \tag{57}$$

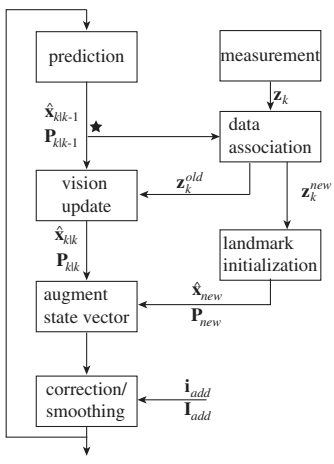

Fig. 7 Flow of estimation: The estimation process is initialized at the junction labeled ★.

$$\mathbf{P}_{0|-} = \mathbf{P}_0 \tag{58}$$

If there are any landmarks known a priori, they are included in $\mathbf{x}_0$ and $\mathbf{P}_0$, otherwise only vehicle states are present in the initial state vector.

At some time t_k:

1) Prediction

Predictions of system state $\hat{\mathbf{x}}_{k|k-1}$ and covariance $\mathbf{P}_{k|k-1}$ are computed using the previous estimates of state and covariance, system kinematics [Eq. (37)] and the UKF prediction step equations.

2) Measurement

Bearings are obtained from the vision system.

3) Data association

If there are any landmarks in the state vector, data association is performed, following the two-step process described in Data Association. Predicted measurements are computed using the vision model (39), predicted system state and covariance, and UKF correction step equations. Bearings associated with known landmarks (i.e., those already in the state vector) are placed in an array $\mathbf{z}_{\text{old}}$, unassociated bearings are placed in an array $\mathbf{z}_{\text{new}}$

4) Vision update

Bearings $\mathbf{z}_{\text{old}}$ associated with known landmarks are used to correct the prediction of the system state using UKF correction step equations, producing $\hat{\mathbf{x}}_{k|k}$ and the associated covariance $\mathbf{P}_{k|k}$.

5) Landmark initialization

Using the recently corrected system state and covariance and the unassociated bearings $\mathbf{z}_{\text{new}}$, initial landmark positions $\mathbf{x}_{\text{new}}$ and covariance $\mathbf{P}_{\text{new}}$ are computed using the landmark initialization process described in Landmark Initialization.

6) State augmentation

The state vector is augmented with the new landmarks so that

$$\hat{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathbf{x}_{\text{new}} \end{bmatrix} \qquad \mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k} & 0 \\ 0 & \mathbf{P}_{\text{new}} \end{bmatrix} \tag{59}$$

7) Correction/smoothing

If additional information is available, correct the current state estimate using Eqs. (54–56).

8) Return to prediction

The process continues recursively.

2-D Simulation Results

The simulations in this paper serve three main purposes: first, to demonstrate that standard estimation techniques (i.e., the EKF) often fail to provide a convergent solution to the problem of estimating vehicle and landmark states; second, to provide results showing that an unscented Kalman filter-based estimation technique can provide a convergent, consistent solution to the state estimation problem; and third, to show that the output from the state estimator can be used by a trajectory planning algorithm to plan safe, feasible trajectories through a cluttered environment.

For ease of visualization, simulations were conducted for navigation in a plane. The state vector $\mathbf{x} = [\mathbf{x}_v^T \ \mathbf{x}_1^T \ \cdots \ \mathbf{x}_m^T]^T$ contains vehicle states and obstacle positions. For the 2-D case the vehicle state vector is

$$\mathbf{x} = [\,x \quad y \quad \psi \quad u \quad v \quad \alpha_x \quad \alpha_y \quad b_x \quad b_y \quad b_\psi\,]^T \tag{60}$$

where x, y, ψ denote vehicle position and orientation in the inertial frame; u and v denote velocity expressed in the body frame; α_x and α_y denote accelerometer scale factor errors in the body x and y directions; and finally b_x, b_y, and b_ψ denote bias errors for the accelerometer x and y directions and the rate gyro. The obstacle position

$$\mathbf{x} = [\,x_i \quad y_i\,]^T \tag{61}$$

includes the obstacle location in the inertial frame.

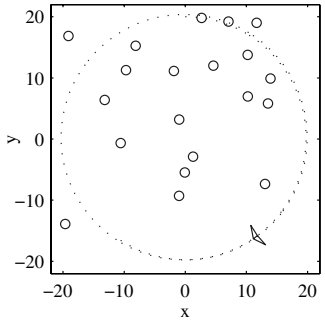

Fig. 8 Schematic of a single run of the Monte Carlo simulation. Aircraft speed is 10 m/s, and turn rate is 0.5 rad/s, resulting in a turn radius of 20 m. The aircraft travels a distance of 200 m, approximately 1.6 times around the circular path.

As with the 3-D case the time update step is represented by vehicle kinematics, driven by inertial measurements. The 2-D vision measurement is represented by a bearing to a fixed landmark.

In this section data association and landmark initialization are assumed to be known a priori. This isolates the estimation problem, allowing a focus on consistency and comparison with an EKF-based implementation.

Estimate Consistency

Kalman filters propagate an estimate of system state and an estimate of the covariance of the estimate error. A *consistent* estimator is both unbiased (i.e., the expected value of the estimate error is zero) and accurately estimates the covariance of the estimate error:

$$\mathbb{E}\,(\mathbf{x}_k - \hat{\mathbf{x}}_k) = 0 \tag{62}$$

$$\mathbb{E}\,((\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T) = \mathbf{P}_{kk} \tag{63}$$

Alternatively,

$$\mathbb{E}\,((\mathbf{x}_k - \hat{\mathbf{x}}_k)^T(\mathbf{x}_k - \hat{\mathbf{x}}_k)) = Tr(\mathbf{P}_{kk}) \tag{64}$$

Hence to evaluate consistency of the navigation solution one can compare the 2-norm of the estimate error with the square root of the trace of the covariance matrix. This can be done using Monte Carlo simulations.

Monte Carlo Simulation Results

The UAV is flown in a circular trajectory of radius 20 m at a velocity of 10 m/s through a forest of randomly distributed trees (see Fig. 8). A new forest was generated for each run. UAV position and velocity was assumed to be known accurately at the start of each run ($\sigma^2 = 1 \times 10^{-6}$) and landmark positions were assumed to be known with a standard deviation of 1 m (i.e., an initial, but uncertain, map was provided for each run). Estimates of IMU biases were initialized with a Gaussian random number using parameters given in Table 1.

Because the purpose of the estimator is to determine vehicle state and obstacle positions with an accurate assessment of the estimate error, success of a run is determined by comparing the estimated standard deviation of the error with the true standard deviation of the error. These are

$$\sigma_{k,\text{est}} = \sqrt{Tr\mathbf{P}_{kk}} \tag{65}$$

Table 1 IMU initialization parameters for 2-D simulations

	α_x	α_y	b_x	b_y	b_ψ
Mean	1	1	0.15	0.10	0.004
σ	0.01	0.01	0.1	0.1	0.02

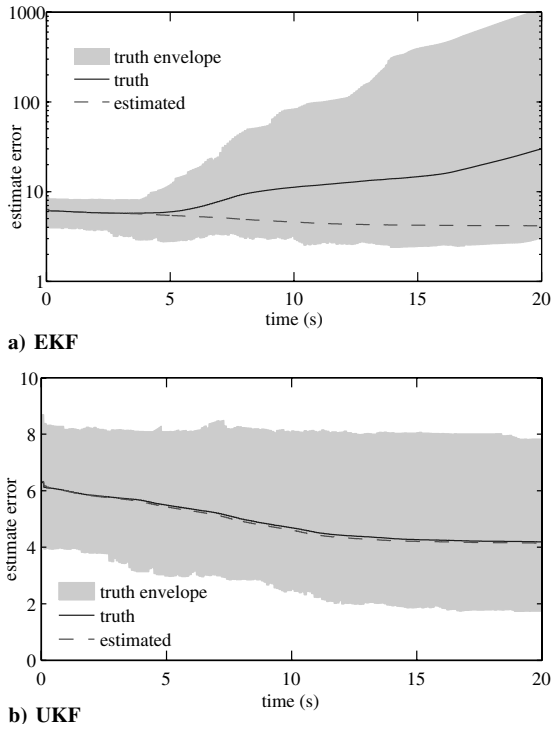

Fig. 9 Estimator consistency for 2-D navigation. A comparison of estimated standard deviation of the estimate error and true standard deviation of the estimate error for the EKF implementation a) and UKF implementation b). Note the difference in scales. Results are from a 1000 run Monte Carlo simulation.

$$\sigma_{k,\text{truth}} = \sqrt{(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T(\hat{\mathbf{x}}_k - \mathbf{x}_k)} \tag{66}$$

where for a consistent estimator $\mathbb{E}\sigma_{k,\text{est}} = \mathbb{E}\sigma_{k,\text{truth}}$ [Eq. (64)].

Results of a 1000 run Monte Carlo simulation are shown graphically in Fig. 9. Each plot shows the envelope of the true variance of estimate error (i.e., the minimum and maximum values of the 2-norm of the estimate error over all runs at each time step) as a gray region, the mean of the true variance of estimate error at each time step as a solid line, and the mean of the estimated error variance (i.e., $\sqrt{Tr\mathbf{P}_{kk}}$) at each time step as a dashed line. The EKF implementation (Fig. 9a) shows clear divergence of the estimate, with true variance an order of magnitude greater than the estimated variance. Conversely the UKF implementation (Fig. 9b) shows a well-behaved estimator with the expectations of true and estimated variance aligned and a narrow envelope of true variance.

A run is ultimately deemed successful if the estimate error at the end of the run is less than the estimate error at the state of the run (i.e., information has been gained about the vehicle and the environment):

$$\sigma_{K,\text{truth}} < \sigma_{1,\text{truth}} \tag{67}$$

Figure 10 shows a cumulative distribution plot of the error ratio $\sigma_{K,\text{truth}}/\sigma_{1,\text{truth}}$ for the UKF-based implementation (solid line) and the EKF-based implementation (dashed line) for 1000 runs. All of the UKF-based estimators resulted in improved knowledge of the environment, and only 205 EKF-based estimators resulted in improved knowledge of the environment.

The estimated state covariance is an assessment of the uncertainty in the estimates of the state variables. It can be used in trajectory planning to generate paths which minimize the probability of collisions with obstacles; hence accurate estimation of the estimate error covariance is an important characteristic of a successful estimator.

The standard deviation of the estimate error at the end of a run can be used as an assessment of the best-case and worst-case performance. If a run is successful (i.e., $\sigma_{K,\text{truth}} < \sigma_{1,\text{truth}}$)

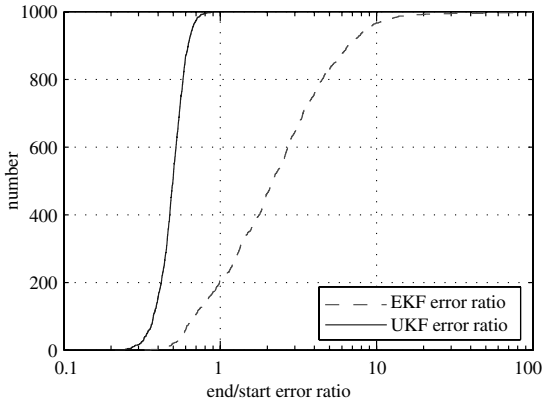

Fig. 10 Change in estimate error over course of run. A cumulative distribution plot of the number of runs which resulted in a reduction in estimate error over the course of each run. All UKF runs resulted in a reduction in true error (i.e., error ratio<1), only 205 EKF runs resulted in reduced true error.

$$\frac{\sigma_{K,\text{truth}}}{\sigma_{K,\text{est}}} = 1 + \delta \quad \text{where } \delta > -1 \tag{68}$$

For a "good" estimator $\delta \approx 0$, indicating that the estimated state uncertainty accurately represents the true state uncertainty. $\delta > 0$ indicates that the estimated state uncertainty underpredicts the true state uncertainty, increasing the likelihood of collisions with obstacles. $\delta < 0$ indicates that the estimated state uncertainty is overpredicted, leading to inefficient trajectories. Note that if the estimate did not converge then δ would continue to grow without bound, and thus would not be a meaningful assessment of performance. In any case, a diverged estimate indicates failure.

Maximum, minimum, and mean values of $(1 + \delta)$ are given in Table 2 for successful runs. Figure 9 showed that all runs for the UKF-based estimator converged. The largest value of $(1 + \delta)$ for the UKF was 1.73, indicating that the end-of-run true standard deviation of the estimate error does not exceed the end-of-run estimated standard deviation of the estimate error by more than 73%. The mean value of $(1 + \delta)$ is 1.05, indicating that on average the true error was accurately predicted by the estimated covariance. Conversely, the EKF-based implementation was unsuccessful for 795 runs, indicated by the increase in estimate error. For the remaining 205 successful runs, the worst-case value of $(1 + \delta)$ was 2.62, indicating that the true error is 162% greater than the estimated error. On average (for successful runs only) the true error exceeded the estimated error by 46%.

It should be noted that performance improvement of the UKF over the EKF is a result of the close proximity of the vehicle to the landmarks. In this case the uncertainty in estimated vehicle and landmark positions is significant when compared with the range to the landmark, increasing the effect of the nonlinearity of the measurement equations. When the vehicle is operating at significant distance away from landmarks (which may occur when the vehicle is at higher altitude) very little difference is observed between the UKF and EKF implementations of the SLAM estimator. In that case the uncertainty in vehicle and landmark positions is small compared with the range to the landmark. The difference between the UKF and EKF may be further reduced if a range measurement is provided. In the current application, however, the UKF showed clear improvement over EKF-based methods.

Table 2 Maximum and minimum standard deviation ratios. Only results for successful runs [as defined in Eq. (67)] are presented

	$\max(1+\delta)$	$\min(1+\delta)$	$\text{mean}(1+\delta)$
UKF (1000 runs)	1.73	0.56	1.05
EKF (205 runs)	2.62	0.75	1.46

Navigation and Obstacle Avoidance

The previous subsection showed that a UKF-based estimator provides a convergent, consistent estimate of vehicle position, orientation, and speed of obstacle positions. Flight in an obstacle-strewn environment also requires an algorithm for obstacle avoidance and navigation. In this section a randomized motion planner [39] is used: a sequence of random inputs is generated and the cost of each trajectory is computed based on the probability of collision with obstacles (computed from the estimated obstacle positions and the associated covariance) and final distance from the goal.

To illustrate flight in a cluttered environment Fig. 11 shows a sequence of images of a UAV navigating through a forest to a goal. True obstacle positions are shown as dots, and true vehicle positions are shown as "wings." Estimated obstacle positions are shown as + with associated 3σ error ellipsoid, and estimated vehicle positions are shown as "wings" with associated 3σ error ellipsoid. The dotted lines show the planned trajectory, the solid lines show the actual trajectory flown. The planned trajectory only accounts for obstacles that have been "seen" by the camera: unobserved obstacles remain unknown until they enter the field of view.

Early in the flight (upper left image) estimated obstacle positions are highly uncertain, as shown by the large 3σ uncertainty ellipsoids. As the aircraft approaches an obstacle it is localized with greater accuracy, reflecting both the fact that more measurements to that obstacle have been obtained and that the measurement geometry permits more accurate localization at shorter range (the "spread" of the uncertainty in the bearing is smaller at close range). The increase in positioning certainty has the consequence of increased safety during operations close to an obstacle: with greater certainty in obstacle position come greater certainty that a planned collision-free path is actually collision free.

Conclusions

This paper has presented a framework to enable control and navigation of small autonomous UAVs operating in cluttered environments, using only a monocular camera and inertial measurements as sensing. In this framework, an estimator computes vehicle state (position, orientation, and velocity) and the positions of obstacles in the environment.

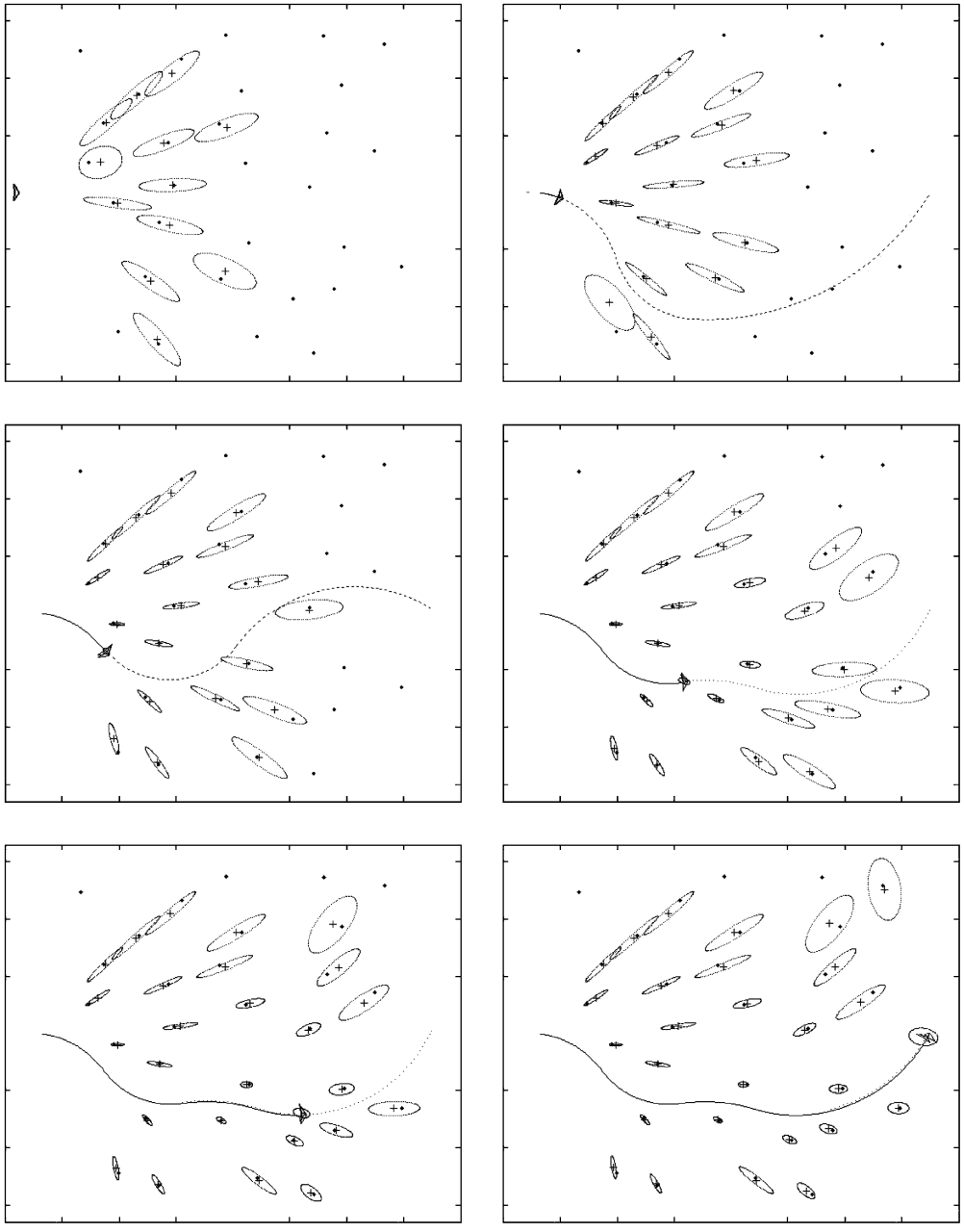

Fig. 11 Obstacle avoidance and navigation in 2-D environment. The sequence of images shows flight through a 2-D forest to a goal position.

The state estimation problem was cast in a framework of bearings-only SLAM, with inertial measurements providing information about ego motion and bearings to landmarks (obtained from the vision system) providing information about the surroundings. The nonlinearities in system models combined with the likelihood of significant uncertainties in system states resulted in an estimation problem that could not be solved reliably using standard techniques. To address the problem of estimation in the face of nonlinearities and significant uncertainty, a SP-KF was implemented and shown to provide consistent, reliable estimates of vehicle state and obstacle position, along with the associated covariance.

Landmarks are not uniquely labeled: therefore data association must be computed explicitly. In typical SLAM implementations data association is computed based on a comparison of the actual measurements with predictions computed from the current best estimate of vehicle position and landmark positions. In bearings-only SLAM applications (such as this one) data association is made more complex by the lack of full information provided by the measurements: the vision system projects a 3-D world onto a 2-D image plane. This measurement to map approach of data association is especially prone to failure when uncertainty in either vehicle state or landmark position is high (a landmark's position uncertainty is at its greatest when it is seen for the first time), and correct data association is a critical component of successful state estimation.

This paper has presented a two-step approach to data association. The first step compares current bearings with those obtained in the previous image frame to check frame-to-frame correspondence. The second step compares bearings with predictions based on landmarks in the current map. This approach is better able to perform data association when landmark position uncertainty is high and it is more robust to dropouts of both individual features (which may occur when a feature is occluded) and dropouts of an entire frame. Any bearing which is not associated with a mapped landmark is assumed to come from a new, previously unseen landmark.

When a landmark is seen for the first time, an initial estimate of range must be obtained. Like data association, this is complicated by the lack of sufficient measurements: a single bearing provides only enough information to localize a landmark along a ray but gives no information about distance. In this research landmarks are assumed to lie on the ground plane. Using the estimate of aircraft position (including altitude) and orientation along with the measured bearing to the new landmark, an initial position estimate is computed at the intersection of the bearing and the ground plane.

The UKF-based estimator was combined with a trajectory planner to demonstrate in simulation flight through a cluttered environment.

References

[1] Langelaan, J., and Rock, S., "Navigation of Small UAVs Operating in Forests," *AIAA Guidance, Navigation and Controls Conference*, AIAA, Reston, VA, Aug. 2004.

[2] Langelaan, J. W., "State Estimation for Autonomous Flight in Cluttered Environments," Ph.D. Thesis, Stanford University, Stanford, CA, March 2006.

[3] Pollefeys, M., Gool, L. V., Vergauwen, M., Cornelis, K., Verbiest, F., and Tops, J., "3D Capture of Archaeology and Architecture with a Hand-Held Camera," *The International Archives of Photogrametry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV, Part 5/W12, July 2003, pp. 262–267.

[4] Webb, T. P., Prazenica, R. J., Kurdila, A. J., and Lind, R., "Vision-Based State Estimation for Uninhabited Aerial Vehicles," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, August 2005; also AIAA Paper 2005-5869.

[5] Prazenica, R. J., Watkins, A., Kurdila, A. J., Ke, Q. F., and Kanade, T., "Vision-Based Kalman Filtering for Aircraft State Estimation and Structure from Motion," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, Aug. 2005; also AIAA Paper 2005-6003.

[6] Roumeliotis, S. I., Johnson, A. E., and Montgomery, J. F., "Augmenting Inertial Navigation with Image-Based Motion Estimation," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, 2002.

[7] Lobo, J., and Dias, J., "Integration of Inertial Information with Vision," *24th Conference of IEEE Industrial Electronics Society (IECON)*, IEEE, Piscataway, NJ, 1998, pp. 1263–1267.

[8] Netter, T., and Franceschini, N., "A Robotic Aircraft that Follows Terrain Using a Neuromorphic Eye," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway, NJ, 2002.

[9] Diel, D. D., DeBitetto, P., and Teller, S., "Epipolar Constraints for Vision-Aided Inertial Navigation," *IEEE Workshop on Motion and Video Computing*, IEEE, Piscataway, NJ, 2005.

[10] Marks, R. L., "Experiments in Visual Sensing for Automatic Control of an Underwater Robot," Ph.D. Thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA, June 1995; also published as SUDAAR 681.

[11] Leabourne, K. N., Rock, S. M., Fleischer, S. D., and Burton, R. L., "Station Keeping of an ROV Using Vision Technology," *OCEANS 97 Conference*, Vol. 1, MTS/IEEE, Piscataway, NJ, Oct. 1997, pp. 634–640.

[12] Richmond, K., and Rock, S., "A Real-Time Visual Mosaicking and Navigation System," *Proceedings of the Unmanned Untethered Submersible Technology Conference*, Autonomous Undersea Systems Institute, Lee, NH, Aug. 2005.

[13] Huster, A., and Rock, S., "Relative Position Sensing by Fusing Monocular Vision and Inertial Rate Sensors," *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, IEEE, Piscataway, NJ, July 2003.

[14] Amidi, O., "An Autonomous Vision Guided Helicopter, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, Aug. 1996.

[15] Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., "Vision Based Navigation for an Unmanned Aerial Vehicle," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, May 2001.

[16] Roberts, J. M., Corke, P. I., and Buskey, G., "Low-Cost Flight Control System for a Small Autonomous Helicopter," *2002 Australasian Conference on Robotics and Automation*, ARAA, Brisbane, Australia, Nov. 2002.

[17] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Vision-Based Autonomous Landing of an Unmanned Aerial Vehicle," *International Conference on Robotics and Automation*, IEEE, Piscataway, May 2002, pp. 2799–2804.

[18] Meingast, M., Geyer, C., and Sastry, S., "Vision Based Terrain Recovery for Landing Unmanned Aerial Vehicles," *Conference on Decision and Control*, IEEE, Piscataway, NJ, Dec. 2005.

[19] Proctor, A., and Johnson, E. N., "Vision-Only Approach and Landing," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, Aug. 2005; also AIAA Paper 2005-5871.

[20] Wu, A. D., Johnson, E. N., and Proctor, A. A., "Vision-Aided Inertial Navigation for Flight Control," *AIAA Guidance, Navigation and Control Conference*, AIAA, Reston, VA, Aug. 2005; also AIAA Paper 2005-5998.

[21] Thrun, S., Burgard, W., and Fox, D., "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, 2000.

[22] Williams, S. B., Dissanayake, G., and Durrant-Whyte, H., "Field Deployment of the Simultaneous Localisation and Mapping Algorithm," *15th IFAC World Congress on Automatic Control*, Elsevier, Oxford, U.K., June 2002.

[23] Liu, Y., and Thrun, S., "Results for Outdoor-SLAM Using Sparse Extended Information Filters," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, 2003.

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

[25] Kwok, N. M., and Dissanayake, G., "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," *Interational Conference on Intelligent Robots and Systems*, IEEE, Piscataway, NJ, 2004.

[26] Fitzgibbons, T., and Nebot, E., "Bearing Only SLAM Using Colour-Based Feature Tracking," *2002 Australasian Conference on Robotics and Automation*, ARAA, Brisbane, Australia, 2002.

[27] Neira, J., Ribeiro, M. I., and Tardós, J. D., "Mobile Robot Localization and Map Building Using Monocular Vision," *Fifth International Symposium on Intelligent Robotic Systems*, IAS, Amsterdam, 1997.

[28] Davison, A. J., "Real-Time Simultaneous Localisation and Mapping with a Single Camera," *International Conference on Computer Vision*, IEEE, Piscataway, NJ, Oct. 2003.

[29] Burschka, D., and Hager, G. D., "V-GPS(SLAM): Vision Based Inertial System for Mobile Robots," *IEEE International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, April 2004.

[30] Foxlin, E., and Naimark, L., "VIS-Tracker: A Wearable Vision-Inertial

Self-Tracker," *IEEE VR2003*, IEEE, Piscataway, NJ, March 2003.

[31] van der Merwe, R., and Wan, E., "The Square-Root Unscented Kalman Filter for State and Parameter-Estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Piscataway, NJ, 2001.

[32] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, March 2000, pp. 477–482.

[33] Classon, J., "Map Building Using Mobile Robots," M.S. Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2006.

[34] Chekhlov, D., Pupilli, M., Mayol-Cuevas, W., and Calway, A., "Real-Time and Robust Monocular SLAM Using Predictive Multi-Resolution Descriptors," *2nd International Symposium on Visual Computing*, Springer, New York, Nov. 2006.

[35] Neira, J., and Tardós, J. D., "Data Association in Stochastic Mapping Using the Joint Compatibility Test," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 6, Dec. 2001, pp. 890–897.

[36] i Ortega, J. S., Lemaire, T., Decy, M., Lacroix, S., and Monin, A., "Delayed vs Undelayed Landmark Initialization for Bearing Only SLAM," *Workshop on Simulateous Localisation and Mapping, International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, 2005.

[37] Bailey, T., "Constrained Initialisation for Bearing-Only SLAM," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NJ, 2003.

[38] Montesano, L., Gaspar, J., Santos-Victor, J., and Montano, L., "Fusing Vision-Based Bearing Measurements and Motion to Localize Pairs of Robots," *International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, 2005.

[39] Hsu, D., Kindel, R., Latombe, J. C., and Rock, S., "Randomized Kinodynamic Motion Planning with Moving Obstacles," *International Journal of Robotics Research*, Vol. 21, No. 3, March 2002, pp. 233–255.