

Towards Autonomous UAV Flight in Forests

Jack Langelaan* and Steve Rock†

Stanford University, Stanford, CA 94305, USA

This paper presents results demonstrating real-time six degree of freedom localization, mapping, navigation and obstacle avoidance in an outdoor environment using only a low-cost off-the-shelf inertial measurement unit and a monocular camera. This navigation system is intended for operation of small unmanned aerial vehicles when GPS signals are unavailable due to obstructions or jamming and when operating in cluttered environments such as urban canyons or forests. A small radio-controlled car is used as a test bed.

A bearings-only Simultaneous Localization and Mapping algorithm was implemented to localize both the vehicle and obstacles. This paper describes: (a) the hardware used; (b) a two-step approach for data association (image frame to image frame followed by image frame to map); (c) a technique for landmark initialization for the case where landmarks are located on the ground. Hardware test results demonstrating navigation to a goal in an obstacle strewn environment are presented. The effect of unmodelled sensor biases is examined in simulation.

I. Introduction

SUCCESSFUL OPERATION OF autonomous mobile systems in obstacle strewn environments (such as urban areas, the sea floor, boulder strewn ground or the forest shown in Figure 1) requires both navigation and obstacle avoidance. A particularly challenging and complex subset of autonomous mobile systems is small autonomous Unmanned Aerial Vehicles (UAVs). They undergo 6 degree of freedom motion, are subject to significant external disturbances, require high bandwidth control and their small payload capacity limits on-board sensing. At the same time the missions envisioned for such vehicles are very challenging, involving low-altitude flight in obstacle strewn terrain.

Available sensing is limited both by the payload capacity of the vehicle and by the environment. A typical sensor suite on a small UAV may include GPS, an inertial measurement unit (IMU) and a monocular camera. However, in cluttered environments GPS signals may be obscured by obstacles (either natural or man-made) or may be jammed. Direct measurements of vehicle position are therefore unavailable. Additionally, since small UAVs are typically intended to be inexpensive the use of high-quality inertial measurement units (IMU) is precluded. Thus a purely inertial navigation solution is infeasible.

Vision systems provide an information-rich stream of data which is useful to a human operator and can be used by the navigation system. A stereo camera provides both range and bearing to features. However, vehicle dimensions impose limits on the baseline between cameras, and range accuracy is strongly dependent both on the baseline and on the actual range to the feature. Obtaining accurate range measurements to far away features can therefore be



Figure 1. The forest.

*Ph.D. Candidate, Department of Aeronautics and Astronautics, Durand Building, Senior Member AIAA.

†Professor, Department of Aeronautics and Astronautics, Durand Building, Fellow AIAA.

Copyright © 2005 by Jack Langelaan and Steve Rock. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

problematic. A monocular camera provides bearings to objects, and by fusing bearings to fixed landmarks with data from the low-cost IMU we can obtain a localization solution both for the UAV and for the landmarks.

Fusing inertial measurements with bearings to known features has been applied to the problem of autonomous ship-board landing,¹ indoor navigation by humans² and navigation of robots.³⁻⁵ When features are not known their locations must be included in the vector of states to be estimated. This is known as Simultaneous Localization and Mapping (SLAM), sometimes referred to as Concurrent Localization and Mapping.

SLAM has been the subject of a great deal of research. It enables navigation by simultaneously estimating the state of the vehicle and of landmarks in the environment using only on-board sensing, which typically includes odometry (for wheeled robots) or inertial measurements to measure vehicle motion and relative measurements (i.e. range and bearing) to landmarks. SLAM has been applied both in simulation and on hardware in situations as diverse as navigation of wheeled ground vehicles⁶ and subsea navigation of autonomous underwater vehicles.⁷ An algorithm has even been developed for a fixed wing UAV.⁸ In most SLAM implementations measurements to landmarks include both bearing and range, typically provided by a scanning laser range finder in land-based applications and scanning sonar in underwater applications. In the UAV application cited above the large size of the vehicles (10kg payload capacity) enabled use of a laser range finder for range measurements and a camera for bearing measurements.

In the application presented here only bearings to landmarks are available. This provides unique challenges that cause Extended Kalman Filter (EKF) estimation techniques described in the above references to diverge.⁹ These include: limited observability of states, which results from having only inertial and bearing measurements available; increased effect of non-linearities in the measurement model, which results from the close proximity of the vehicle to the objects being observed; and finally the significant uncertainty in predicted vehicle state, which results both from the measurement noise and drift error induced by the low cost IMU and the likelihood of significant external disturbances.

In previous work we described an implementation based on an Unscented Kalman Filter (UKF), demonstrating successful navigation through an obstacle-strewn environment in 2D simulation⁹ and demonstrating localization and mapping using hardware in a laboratory environment.¹⁰ In this work we present results demonstrating full 3D localization and mapping in an outdoor environment using only a low-cost off-the-shelf IMU and a monocular camera.

The remainder of this paper is organized as follows: Section II defines the problem under consideration, Section III describes the navigation system, Section IV gives results of testing and Section V discusses the results and presents conclusions.

II. Navigation Problem

THE PROBLEM OF flying a small UAV through an unsurveyed forest is considered (Figure 2). In addition to avoiding collisions with obstacles the vehicle must navigate to a goal, hence accurate position information must be obtained. Measurements are limited to vehicle accelerations and angular rates in the body frame (provided by an IMU) and bearings to obstacles (provided by a monocular camera).

Trees are located at \mathbf{x}_i in frame O , an inertial NED frame. A transformation matrix \mathbf{T} resolves a vector in frame O to the body frame B . $\boldsymbol{\omega}$ represents the rotational velocity of the vehicle expressed in the body frame and $[u \ v \ w]^T$ represents the velocity of the aircraft in the body frame. The position of the aircraft is defined by $[x \ y \ z]^T$, expressed in frame O .

The problem is to estimate the locations of the obstacles and of the vehicle state given this limited sensor suite. The estimation problem contains two significant non-linearities: the first is due to the rotational degrees of freedom of the vehicle and the second is due to the vision system's projection of the three-dimensional world onto the two-dimensional image plane. Thus both the system dynamics and measurement equations are dependent on the actual state of the system.

Since range measurements are unavailable, observability of states is highly dependent on motion of the camera (i.e. the trajectory flown by the UAV). During motion

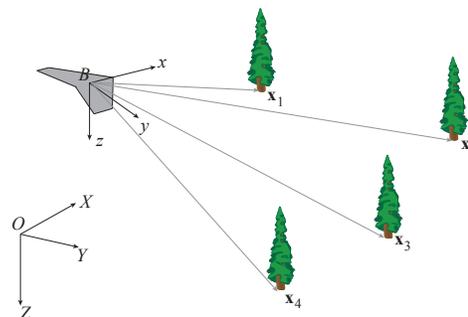


Figure 2. Navigation problem.

Table 1. Typical 1σ measurement noise for IMU portion of Phoenix-AX autopilot module (data transmitted at approximately 250Hz, averaged to 50Hz).

acceleration (m/s ²)			angular rate (rad/s)		
x	y	z	x	y	z
0.085	0.080	0.062	0.0084	0.0098	0.0078

directly towards or away from an object (along the bearing) there is no new information which improves the estimate of range. Transverse motion is required to produce a useful estimate of object position. In the case of obstacle avoidance, transverse motion has the added benefit of ensuring that a collision is avoided, but the presence of multiple obstacles places conflicting demands on the trajectory which must be flown. Additional demands on the trajectory may arise if certain areas are to be avoided or if a final goal location must be reached.

III. System Description

THE NAVIGATION SYSTEM can be loosely divided into four subsystems: a vision subsystem, which provides bearings to landmarks in the environment; an inertial subsystem, which provides accelerations and angular rates; the estimator; and a path planning and control subsystem (see Figure 3).

The camera and IMU are carried on board the vehicle and transmit data to a ground station via wireless links, where all computation is performed. Commands are subsequently sent to the vehicle through a modified RC transmitter.

A. Vision Subsystem

The camera is a Panasonic KX-131 CCD camera equipped with a 115° field of view lens. Resolution is 380 lines by 640 pixels per line. This is converted to NTSC before analog transmission over a 2.4GHz wireless link to the groundstation. A Matrox Meteor II framegrabber captures frames at 10Hz.

Each frame is blurred using a 5 pixel×5 pixel Gaussian filter and downsampled to 320×240 pixels. The downsampled image is further blurred with a 5 pixel×5 pixel flat filter prior to segmentation. Color segmentation is done in the HSV color space (Hue, Saturation, Value) to recognize red blobs, which have been defined as landmarks/obstacles. A standard lens distortion correction algorithm¹¹ is applied to the xy pixel location of the centroid of each blob.

The wireless link is subject to dropouts caused by interference, multipath and occlusion, hence the estimator must be robust to data losses in the vision subsystem.

B. Inertial Subsystem

The inertial measurement unit is part of the Phoenix-AX Autopilot Module.¹² In this application accelerations and angular rates (along with a packet checksum) are transmitted over an RS-232 link using a 900MHz radio modem to a ground station at a rate of approximately 100Hz. At the ground station this is downsampled to 50Hz by averaging. This reduces the effect of a lost packet.

Noise levels and starting values for biases are obtained prior to each run by keeping the unit approximately level and stationary for a period of time. The mean and standard deviation of the signals provide bias and noise, respectively. Typical noise levels are given in Table 1.

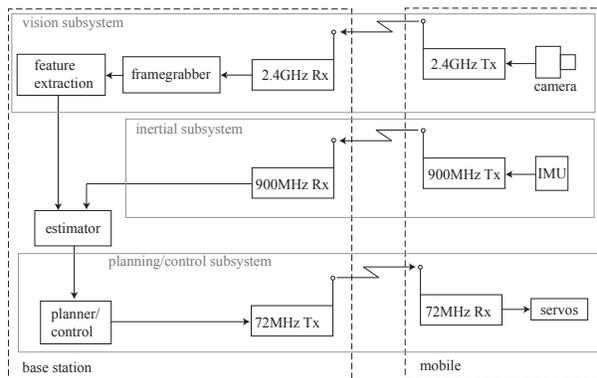


Figure 3. Schematic of mapping/navigation system.

C. Estimator

A detailed description of the algorithm used by the estimator is given in previous work.⁹ It is summarized here.

The vector of states to be estimated includes vehicle states and obstacle states and the associated covariance:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_v \\ \mathbf{x}_o \end{bmatrix} \quad (1)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vo} \\ \mathbf{P}_{ov} & \mathbf{P}_{oo} \end{bmatrix} \quad (2)$$

where

$$\mathbf{x}_v = \left[x \ y \ z \ \phi \ \theta \ \psi \ u \ v \ w \ \boldsymbol{\alpha}^T \ \mathbf{b}_a^T \ \mathbf{b}_\omega^T \right]^T \quad (3)$$

$$\mathbf{x}_o = \left[\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_m^T \right]^T \quad (4)$$

The vehicle states are composed of position in the inertial frame ($x \ y \ z$), Euler angles with respect to the inertial frame ($\phi \ \theta \ \psi$), velocity expressed in the body frame ($u \ v \ w$) and IMU biases, which must be included because of the possibility of drift. The obstacle state vector \mathbf{x}_o is the concatenation of all obstacle positions.

In discrete form the system dynamics are:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{v,k+1} \\ \mathbf{x}_{o,k+1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_{v,k}, \mathbf{u}_k) \\ \mathbf{x}_{o,k} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{ext} \\ 0 \end{bmatrix} \quad (5)$$

Landmarks are assumed to be stationary. The non-linear function f captures the kinematics of the vehicle and the IMU bias states and is driven by the input u , which contains inertial measurements and a zero-mean Gaussian term for the IMU biases, whose drift is modelled by a random walk:

$$\mathbf{u}_k = \begin{bmatrix} \mathbf{z}_{imu,k} \\ \mathbf{n}_{\alpha,k} \\ \mathbf{n}_b,k \end{bmatrix} \quad (6)$$

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.

The vision model reflects the projection of a vector in 3D (the vector from the vehicle to the landmark) onto the 2D image plane. This has the form

$$\mathbf{z}_{cam,k} = h(\mathbf{x}_{v,k}, \mathbf{x}_{o,k}) + \mathbf{n}_c \quad (7)$$

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

Both the vehicle kinematics $f(\mathbf{x}_{v,k}, \mathbf{u}_k)$ and the projection function $h(\mathbf{x}_{v,k}, \mathbf{x}_{o,k})$ are highly non-linear, complicating the problem of accurately propagating a state estimate through the system equations. A Sigma Point Kalman Filter^{13,14} is used to estimate the vehicle and obstacle states.

Two critical components of the estimation process that were not addressed in previous work^{9,10} are data association and feature initialization.

1. Data Association

Inherent in any Kalman filter is an assumption of known data association. However, in many SLAM implementations landmarks are indistinguishable from one another, hence this must be computed explicitly. Typical data association algorithms are based on using a χ^2 test to compare an actual measurement with a prediction. The difficulty of data association in bearings-only SLAM lies in the small size of the measurement subspace (the 2D image plane as opposed to the 3D 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 Neira¹⁶ the lengths of the vertical lines

used as features is used as additional information for data association; Fitzgibbons¹⁵ uses color) to assist in the process of data association.

In the implementation discussed here 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.

Angular rate measurements available from the IMU can be integrated to obtain the change in orientation of the camera between frames. The change in orientation is used to calculate predicted bearings for features seen in the previous frame and these are compared with current measurements. While changes in 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 and then transformed by a matrix \mathbf{T} obtained from the change in camera orientation since the previous image:

$$\mathbf{c} = \mathbf{T} \begin{bmatrix} 1 \\ z_{y,k-1} \\ z_{z,k-1} \end{bmatrix} \quad (8)$$

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

$$\bar{\mathbf{z}}_k = \frac{1}{c_x} \begin{bmatrix} c_y \\ c_z \end{bmatrix} \quad (9)$$

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

$$d_{ij} = (\mathbf{z}_i - \bar{\mathbf{z}}_j)^T \mathbf{P}_{jj}^{-1} (\mathbf{z}_i - \bar{\mathbf{z}}_j) \quad (10)$$

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.

The bearing to map association compares a current bearing with the predicted bearings computed using the current state estimate. 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.

2. Landmark Initialization

Landmark initialization is a critical component of SLAM. Kalman filter based implementations assume that all measurements and states obey a Gaussian distribution, and a single bearing does not provide enough information to initialize a landmark correctly. When a landmark is initialized its position and covariance must be close enough to truth that the filter does not diverge.

Methods for feature initialization can be characterized as undelayed or delayed. Undelayed approaches¹⁷⁻¹⁹ 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. Bailey²⁰ 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 in Fitzgibbons¹⁵ and in Montesanto,²¹ 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). A sigma-point transform is used to determine the initial landmark position and its uncertainty: the sigma points associated with the uncertain bearing are projected onto the ground plane and the mean and covariance defining the initial landmark location are computed from the projected sigma points.

This method of landmark initialization can be extended to non-planar 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.

D. Obstacle Avoidance and Vehicle Control

A modified RC transmitter is used to control the vehicle. Desired control inputs are transmitted over a serial link to a PIC and converted to voltages in the range (0V,5V). The RC transmitter converts the voltages and transmits the signal to a standard RC receiver on the vehicle.

Control commands are generated based on the current map and vehicle location and the desired goal. Generic trajectory planning routines can be implemented (previous work⁹ showed 2D simulation results using a potential field approach and a randomized motion planner), in the current work a variant of a potential field obstacle avoidance routine was implemented. A cost function based on distance from the goal and both the Mahalanobis distance and Eulerian distance between the vehicle and the mapped obstacles is defined:

$$c = (\mathbf{x}_g - \mathbf{x}_v)^T (\mathbf{x}_g - \mathbf{x}_v) + \sum_{i=1}^n [(\mathbf{x}_i - \mathbf{x}_v)^T \mathbf{P}_{ii}^{-1} (\mathbf{x}_i - \mathbf{x}_v) + (\mathbf{x}_i - \mathbf{x}_v)^T (\mathbf{x}_i - \mathbf{x}_v)] \quad (11)$$

Here \mathbf{x}_g is the goal location, \mathbf{x}_v is current estimated vehicle position and \mathbf{x}_i is the estimated position of the i^{th} obstacle. To minimize the cost function the desired steering angle is computed based on the cross product of the vehicle’s velocity vector (in the inertial frame) and the gradient of the cost function computed at the current estimated vehicle position:

$$\gamma \propto \arcsin(\mathbf{v} \times \nabla c) \quad (12)$$

Vehicle speed is commanded to be constant.

Potential field obstacle avoidance algorithms are subject to problems such as local minima and non-optimality. However they are easy to implement and in certain situations (such as widely spaced convex obstacles) they can perform well.

IV. Test Results

THIS SECTION PRESENTS results of both simulation and hardware tests. The scenario used for both simulations and hardware tests is described; simulation results for a nominal localization and mapping run are presented; results of outdoor tests showing navigation to a goal are described; results of simulations to assess the effects of certain unmodelled biases are presented; finally results are summarized.

A. Scenario

The scenario consists of a small autonomous ground vehicle operating on level ground in a forest-like environment. While a ground vehicle driving on a flat surface has 3DOF kinematics, the navigation system was performing full six degree of freedom estimation for these tests, i.e. no information about altitude, roll or pitch were given to the estimator. The vehicle and test site are shown in Figure 4.



Figure 4. The vehicle (a) and the artificial forest (b) used for testing.

Three pylons were placed in a 3-4-5 triangle to define a coordinate frame. Thirteen artificial ‘trees’ were placed randomly with a minimum spacing of approximately 2 meters extending from $x=7\text{m}$ to $x=32\text{m}$ and from $y=-1\text{m}$ to $y=5\text{m}$ in the frame defined by the three pylons. A ‘tree’ consists of a 2m tall PVC pole topped with a red flag. This enabled the use of colour segmentation to identify obstacles.

Recall that landmark initialization uses an assumption of planar ground and the altitude estimate of the vehicle to compute the initial landmark location and uncertainty. With features located at an altitude of 2 meters above ground and the vehicle driving along the ground this is geometrically similar to an aircraft flying at an altitude of 2 meters with features on the ground.

The same forest geometry was used for all of the runs (both simulation and hardware) described in the following sections. In all cases the vehicle starts at $(-3.2,1)$ in the coordinate frame defined by the three pylons.

B. Localization and Mapping: Nominal Case (simulation)

Results for a nominal localization and mapping run are shown in Figure 5. The vehicle proceeds from the start position at a constant speed of 0.5 m/s in the $+x$ direction. The camera field of view is 115° and range is 9m, which approximately matches the landmark detection range for the hardware implementation.

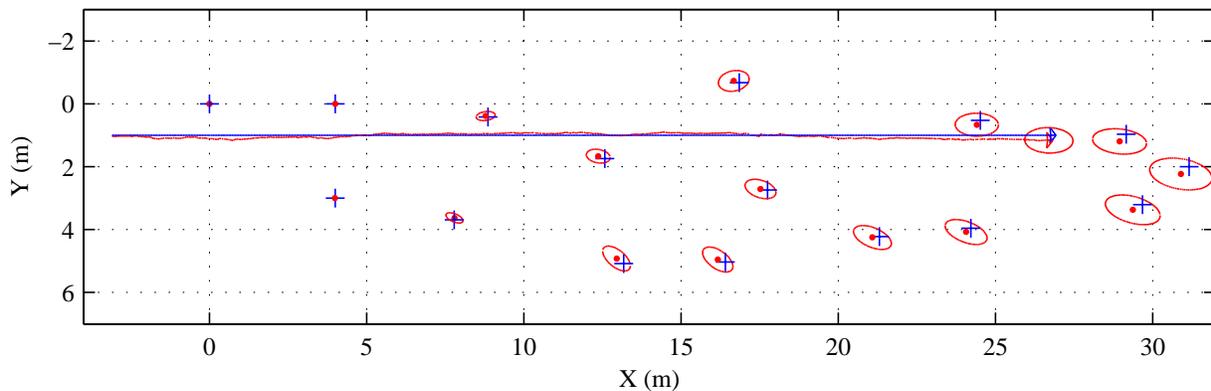


Figure 5. Nominal results for localization and mapping. True tree locations are shown as blue +, true vehicle path and final position shown with blue line and ‘wings’. Estimated tree locations shown as red dot with surrounding 3σ error ellipsoid, estimated path and final position shown with red line and ‘wings’ with surrounding 3σ error ellipsoid.

Locations of the pylons at $(0,0)$, $(4,0)$ and $(4,3)$ are incorporated into the estimator as a priori knowledge with 1σ uncertainty of 2cm.

There is very close agreement between estimated landmark positions and actual landmark positions and between the estimated vehicle position and true vehicle positions. The vehicle localization error for this

nominal run was 0.7% of the distance travelled, the 3σ error bound on position is approximately 2.5% of distance travelled.

In the nominal case the localization error grows with distance travelled. This is to be expected, because unless landmarks remain in constant view over the entire run the navigation system is essentially performing visual odometry. The localization and mapping error is well within 3σ bounds.

C. Navigation to a Goal: Hardware Tests

In these hardware tests the vehicle is commanded to travel from its start position at (-3.2,1) to a goal location at (22.1,1.8) in the coordinate frame defined by the three pylons. The positions of the three pylons were incorporated into the navigation estimator as a priori information with 1σ uncertainty of 2cm. The initial vehicle 1σ uncertainty was defined as 5cm in x and y and 2cm in z .

Upon activation of the navigation system, the vehicle initially remained stationary for 10 seconds to compute inertial biases. It then proceeded to the goal location, simultaneously mapping and avoiding obstacles along the way.

Three runs were performed in the same forest to assess the accuracy of the mapping algorithm in an outdoor scenario and to verify navigation and obstacle avoidance. Truth data for tree locations and for final vehicle positions were obtained by tape measure. Total distance travelled in each run was greater than 25m.

In each of the three runs the vehicle successfully avoided trees and drove within 1 meter of the goal. Figure 6 shows the estimated path taken and maps for each of the three runs. The three maps show generally good agreement, especially early in the run. Later in the trajectory the maps do not agree as closely with truth but they are consistent with each other, with estimated tree locations generally within 3σ error bounds across the three runs.

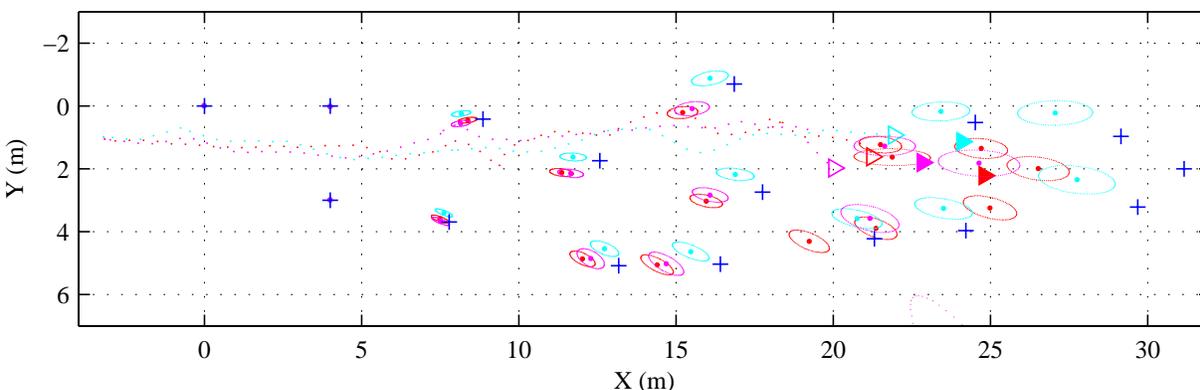


Figure 6. Estimated path and maps generated for three runs through the forest. The common coordinate frame is defined by pylons at (0,0),(4,0) and (4,3). True tree locations are shown by blue +. Vehicle start position was (-3,1) for all runs. Run 1 is shown in red, run 2 is shown in cyan, run 3 is shown in magenta. Solid \triangleright denotes true final vehicle position, open \triangleright denotes estimated final vehicle position. The goal is at (22.1,1.8). Figure 7 shows mapping at various points along the trajectory for run 2.

Table 2 presents errors in final vehicle position estimates and error as a percentage of distance from the start position for each of the three runs. The average value of error as a percentage of distance from the start position is approximately 11%. Note that error as percentage of distance travelled will be somewhat better since the vehicle did not follow a straight line from the start position to the final position.

Figure 7 shows vehicle progress for run 2. Initial mapping is quite accurate, but as the run progresses an offset is seen between truth and the estimated map.

Closer examination of this run shows that while there is an offset between the estimate and truth, the relative positions between the estimated vehicle position and tree positions closely matches the relative positions between the true vehicle position and the true tree positions. This is shown in Figure 8.

The vehicle was driving on an approximately flat, level field, hence approximate truth data is available for altitude. The vehicle's short wheelbase makes a constant pitch assumption problematic and applying throttle caused the vehicle's body to roll a few degrees, hence truth for roll and pitch is not available to a useful accuracy. Figure 9 shows the error in estimated altitude (assuming level ground) and 3σ error bounds

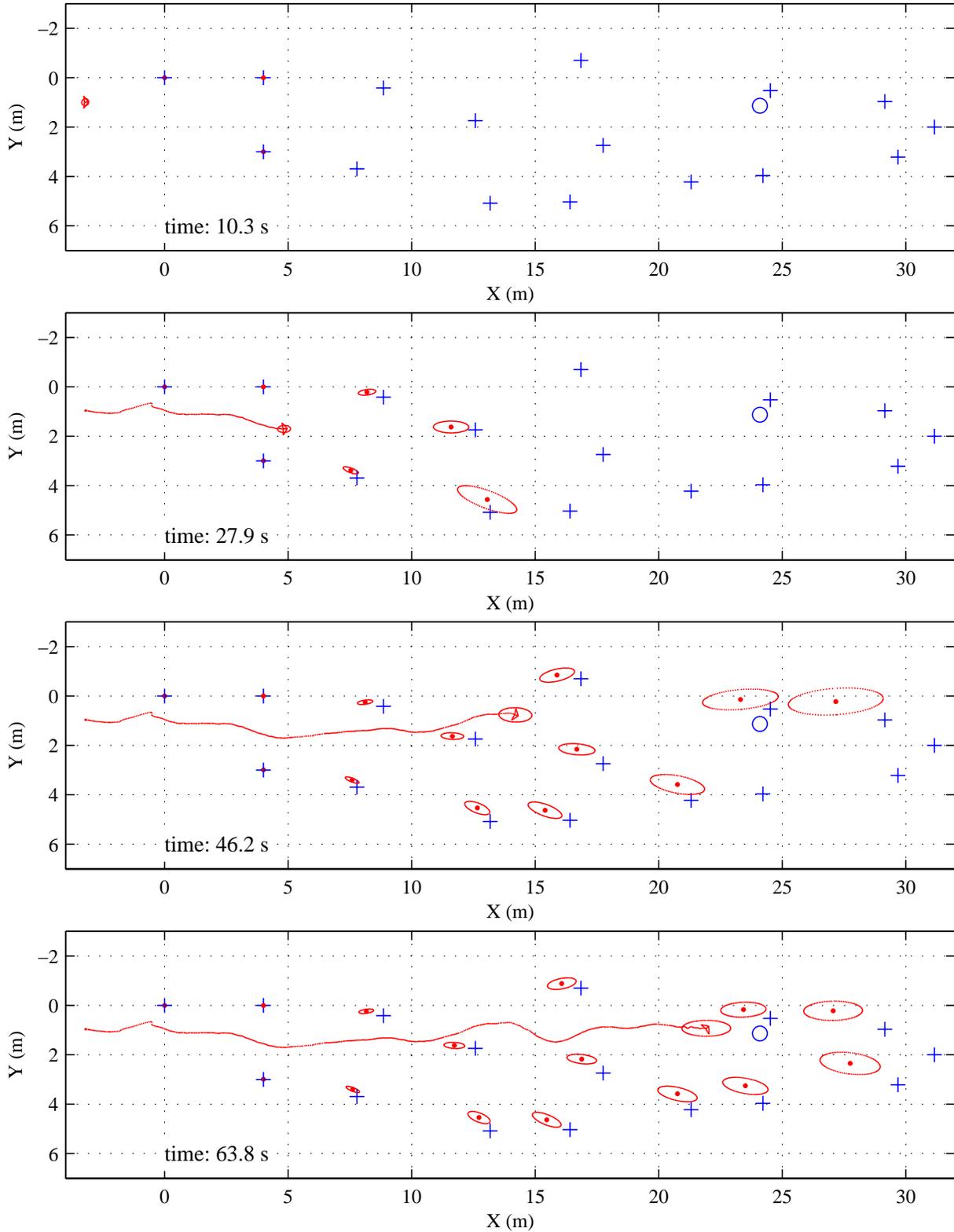


Figure 7. Estimated path taken through the forest. Blue + show true tree locations, blue o shows true final position. Red ellipses denote estimated position, 3σ uncertainty of estimated tree positions, red 'wings' shows estimated vehicle position along with 3σ error ellipsoid. Motion began at $t=10.3$ seconds (top image) and ended at $t=63.8$ seconds (bottom image).

Table 2. Error in final position as percentage of distance from start position. ‘Distance’ denotes true final distance from start position; ‘ Δ distance’ denotes distance between true final position and estimated final position; ‘ $n\sigma$ ’ is number of standard deviations between true final position and estimated final position (computed using estimated covariance); ‘%’ is final position error as percentage of true distance from start position; and ‘ $3\sigma\%$ ’ is the major axis of the 3σ error ellipsoid as a percentage of true distance from the start position.

run	distance (m)	Δ distance (m)	$n\sigma$	%	$3\sigma\%$
1	28	3.65	16.5	13	2.72
2	27.3	2.19	7.1	8	3.69
3	26.1	2.88	11.5	11	3.64
average	27.1	2.9	11.7	10.7	3.35

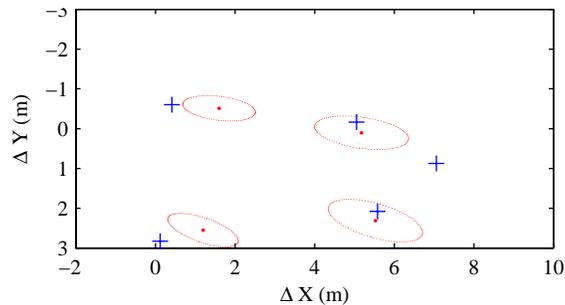


Figure 8. Relative distances from vehicle final position to nearby trees. Truth shown in blue, estimate shown in red with 3σ uncertainty.

for one of the runs. The error is generally within 3σ bounds. There is a strong correlation between the error bounds and the number of landmarks in view at a particular time.

D. Effect of Unmodelled Biases (simulation)

It is through the vision system that measurements (bearings) to stationary landmarks in the environment are obtained. Indeed, this is the only means of obtaining measurements to external variables. Intuitively one would expect biases in the vision system to have a large effect on both the map and navigation solution.

Figure 10 shows the final map and vehicle location for a simulated localization and mapping run with an unmodelled 1° offset in camera pitch. Note the steadily increasing error in estimated landmark positions as the run progresses. This is matched by an increasing error in estimated vehicle position as the run progresses. Comparing Figures 6 and 10 shows that the behaviour observed in the hardware runs is almost identical to that observed in this simulation. By the end of the run a localization error of 9.5% of distance traveled has occurred and the estimated vehicle position is 12 standard deviations (calculated using the estimated error covariance) away from the true position.

E. Summary of Results

The nominal simulation run showed estimated vehicle position error of less than 1% of distance travelled, the 3σ bounds were approximately 2.5% of distance travelled. This agrees with results presented earlier.¹⁰ Hardware test results demonstrated navigation to within 1m of a goal location over multiple >25 m traverses. Vehicle estimated position error was 10% of distance travelled and the position error is also reflected in the map generated. This error in estimated position is virtually identical to that observed in simulations with an unmodelled 1° offset in camera pitch.

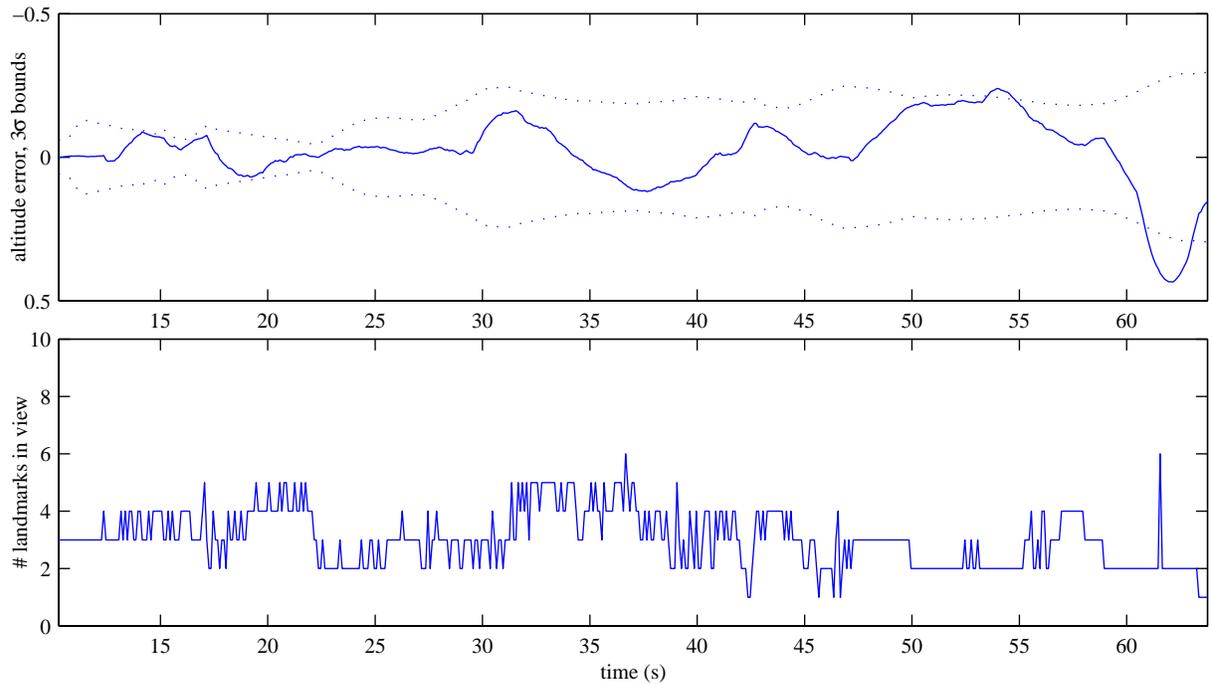


Figure 9. Altitude error (top) and number of landmarks in view (bottom) during the run. Note the correlation between the 3σ error bounds and the number of landmarks in view at a particular time.

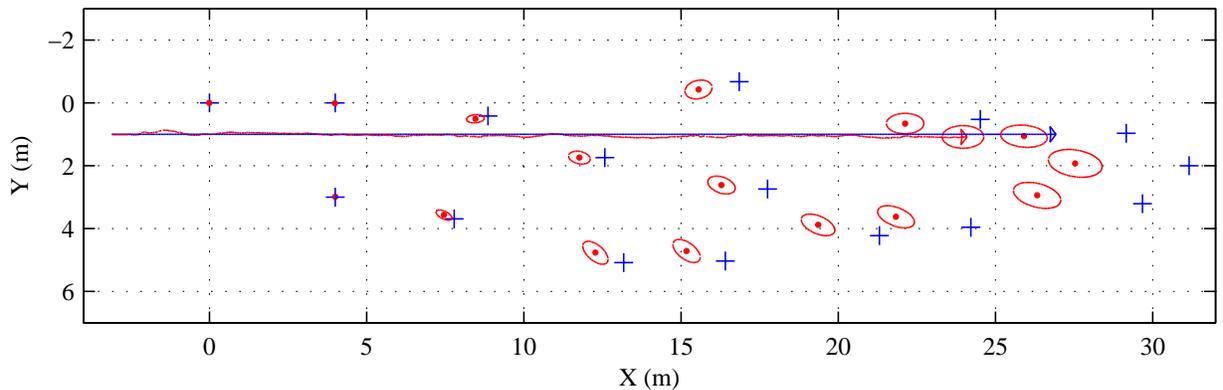


Figure 10. Localization and mapping with an unmodelled 1 degree pitch offset in camera orientation. True tree locations are shown as blue +, true vehicle path and final position shown with blue line and ‘wings’. Estimated tree locations shown as red dot with surrounding 3σ error ellipsoid, estimated path and final position shown with red line and ‘wings’ with surrounding 3σ error ellipsoid.

V. Discussion and Conclusion

A METHOD FOR 6DOF navigation and mapping using only a monocular camera and inertial measurement unit has been demonstrated. A small radio-controlled car was used as a test bed. The vehicle successfully navigated to within 1m of a goal location while avoiding obstacles over multiple >25m traverses in an outdoor environment.

In the outdoor tests estimated vehicle position error of 10% of distance travelled was observed. Previous experience with simulation results showed that estimated vehicle position errors of approximately 2% of distance travelled are achievable and indoor testing showed vehicle position errors of approximately 5% of distance travelled. The vehicle position error observed in the outdoor runs is very similar to that observed in a simulation incorporating an unmodelled 1 degree pitch offset of the camera.

The strong affect of unmodelled vision system biases agrees with intuition: since the vision system provides the only source of measurements to external variables we would expect a bias in these measurements to have a large effect on the mapping (and hence localization) solution. It is through fusing bearings to fixed landmarks (obtained through the vision system) that the drift associated with the inertial navigation solution is reduced.

Furthermore, the accuracy of the navigation solution is strongly dependent on the number of landmarks in view at any given time. The bottom graph in Figure 9 shows that for run 2 an average of only 3 features were in view at a given time. In this case an assumption about one of the landmark states (altitude) reduces the number of landmarks that must in in view for an observable solution: a scale is fixed by the assumption of feature altitude and the estimate of vehicle altitude. With few features in view, however, there is very little robustness to errors.

There is the possibility of other significant unmodelled biases in the vision system. A camera with a wide field of view (approximately 115°) was used, and these can be difficult to calibrate correctly. Using a narrower field of view camera would reduce the likelihood of large errors associated with camera calibration, but a narrower field of view would potentially reduce the length of time a particular landmark is in view. This would reduce the information that can be obtained from a landmark.

The initial IMU bias was calculated by keeping the vehicle stationary and approximately level at the beginning of the run. If the vehicle is not exactly level, the projection of gravity onto the x and y axes will corrupt the bias measurement for these axes (and also for the z axis, since it will now only sense that part of gravity projected onto it). A roll of 2 degrees will affect the y axis bias calculation by $0.035g$, or approximately $0.35m/s^2$, 2 degrees pitch will affect the x axis bias by the same amount.

IMU biases are included in the vector of states to be estimated, so in principle the estimator should be robust to errors in the initial bias estimate. However, preliminary investigations indicate that certain of the IMU bias states (specifically scale factor error and bias in the body x direction) appear to be rather weakly observable. Hence any initial error in these states is unlikely to be improved significantly over time. This is important because the IMU allows us to compute changes in vehicle position between measurements of bearing to landmarks, giving a baseline to triangulate landmark positions. Errors in the IMU bias estimate will affect the estimate of the distance travelled between images obtained through the vision system. This will in turn affect the estimates of landmark positions. There is a potential, however, for designing trajectories to optimize observability in the bias states.

For a wheeled ground vehicle the terrain greatly affects the vibration experienced by the IMU. Isolating the unit (which is very lightweight) is difficult and some of the vibration frequencies were quite low (around 10-20Hz). Hence the measurement noise of the device is significantly less than the magnitude of the vibrations experienced by the device during a run. No filter tuning was performed prior to the runs presented here: this may also help reduce the errors observed.

Even with perfect system models we would expect to see some error growth (although in this case the error would be described by the estimated covariance). Unless the vehicle can maintain view of the same set of features for the entire trajectory this navigation system is essentially performing visual odometry. The error growth associated with odometry can be reduced by periodically revisiting previously explored regions. This technique of loop closure (both recognizing when a loop has closed and incorporating this information in the map) is a subject of research in the SLAM community.

Data association is a key problem in bearings-only SLAM. It must be robust to full vision dropouts and to dropouts of individual features (which may be due to occlusion or to failure to detect a particular feature in a frame). Examining the bottom graph in Figure 9 shows rapid changes in the number of features in view, indicating that in some cases individual landmarks are rapidly entering and leaving view. This is in

part caused by the feature extraction routine: coloured blobs are detected as features and a minimum blob size is used to filter spurious features. When a blob is near the limit of detection it exhibits this behaviour of entering and leaving view.

To summarize, this system has been used to successfully navigate through an obstacle-strewn environment to a goal. Simulations showed that mapping and localization is strongly affected by unmodelled biases in the vision system and preliminary investigations have shown that certain IMU biases are only weakly observable. This requires further investigation.

References

- ¹Kaminer, I., Pascoal, A., and Kang, W., "Integrated Vision/Inertial Navigation System Design Using Nonlinear Filtering," *Proceedings of the American Control Conference*, San Diego, California, June 1999.
- ²Foxlin, E. and Naimark, L., "VIS-Tracker: A Wearable Vision-Inertial Self-Tracker," *IEEE VR2003*, IEEE, IEEE, Los Angeles, California USA, March 2003.
- ³Roumeliotis, 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, Washington, DC, 2002.
- ⁴Lobo, J. and Dias, J., "Integration of Inertial Information with Vision," *24th Conference of IEEE Industrial Electronics Society (IECON)*, IEEE, Aachen, Germany, 1998, pp. 1263–1267.
- ⁵Netter, T. and Franceschini, N., "A Robotic Aircraft that Follows Terrain Using a Neuromorphic Eye," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.
- ⁶Thrun, 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, San Francisco, CA, 2000.
- ⁷Williams, S. B., Dissanayake, G., and Durrant-Whyte, H., "Field Deployment of the Simultaneous Localisation and Mapping Algorithm," *15th IFAC World Congress on Automatic Control*, Barcelona, Spain, June 2002.
- ⁸Kim, J. H. and Sukkarieh, S., "Recasting SLAM—Towards Improving Efficiency and Platform Independency," *11th International Symposium of Robotics Research (ISRR)*, Siena, Italy, 2003.
- ⁹Langelaan, J. and Rock, S., "Navigation of Small UAVs Operating in Forests," *AIAA Guidance, Navigation and Controls Conference*, American Institute of Aeronautics and Astronautics, Providence, RI USA, August 2004.
- ¹⁰Langelaan, J. and Rock, S., "Passive GPS-Free Navigation for Small UAVs," *IEEE Aerospace Conference*, IEEE, Big Sky, Montana USA, March 2005.
- ¹¹Bouget, J.-Y., "Camera Calibration Toolbox for Matlab," <http://www.vision.caltech.edu/bougetj/calib.doc/index.html>.
- ¹²O-Navi, LLC, PO Box 27213, San Diego CA 92198-1213, *Phoenix/AX 6DOF/GPS Sensor Module Quick Start Guide*, rev. x10 ed., April 2004.
- ¹³Julier, 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.
- ¹⁴van der Merwe, R., Wan, E., and Julier, S., "Sigma Point Kalman Filters for Nonlinear Estimation and Sensor Fusion: Applications to Integrated Navigation," *AIAA Guidance, Navigation and Controls Conference*, American Institute of Aeronautics and Astronautics, Providence, RI USA, August 2004.
- ¹⁵Fitzgibbons, T. and Nebot, E., "Bearing Only SLAM using Colour-based Feature Tracking," *2002 Australasian Conference on Robotics and Automation*, Auckland, New Zealand, 2002.
- ¹⁶Jose Neira, Maria Isabel Ribeiro, J. D. T., "Mobile Robot Localization and Map Building using Monocular Vision," *Fifth International Symposium on Intelligent Robotic Systems*, Stockholm, Sweden, 1997.
- ¹⁷Davison, A. J., "Real-Time Simultaneous Localisation and Mapping with a Single Camera," *International Conference on Computer Vision*, Nice, France, October 2003.
- ¹⁸Joam Sola i Ortega, Thomas Lemaire, M. D. S. L. and Monin, A., "Delayed vs Undelayed Landmark Initialization for Bearing Only SLAM," *Workshop on Simultaneous Localisation and Mapping, International Conference on Robotics and Automation*, Barcelona, Spain, 2005.
- ¹⁹Kwok, N. M. and Dissanayake, G., "An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM," *International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.
- ²⁰Bailey, T., "Constrained Initialisation for Bearing-Only SLAM," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Taipei, Taiwan, 2003.
- ²¹Luis Montesano, Jose Gaspar, J. S.-V. and Montano, L., "Fusing Vision-Based Bearing Measurements and Motion to Localize Pairs of Robots," *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.