Bearing-Only SLAM for an Airborne Vehicle

Mitch Bryson and Salah Sukkarieh
ARC Centre for Excellence in Autonomous Systems
Australian Centre for Field Robotics
The Rose St. Building, J04
University of Sydney
m.bryson,salah@cas.edu.au

Abstract

This paper presents results of an application of vision-aided bearing-only Simultaneous Localisation And Mapping (SLAM) for an Unmanned Aerial Vehicle (UAV) while operating over unstructured, natural environments. A single colour vision camera is used to observe the terrain from which image points corresponding to both man-made and natural features are extracted. The SLAM algorithm estimates the complete 6-DoF motion of the UAV along with the three-dimensional position of the features in the environment. An Extended Kalman Filter (EKF) approach is used where a technique of delayed initialisation is performed to initialise the 3D position of features from bearing-only observations. Results are presented by running the algorithm using inertial sensor and vision data collected during a flight test of a UAV.

1 Introduction

The Simultaneous Localisation and Mapping (SLAM) problem has been an active research area within the robotic community over the last few years. Most demonstrated implementations of SLAM have involved low speed indoor robots or outdoor ground vehicles operating in structured, artificial environments. In applications such as search and rescue, surveillance/picture compilation and planetary exploration [Braun et al., 2004], where a high degree of maneuverability/vehicle speed and large area coverage ability is required, the use of autonomous airborne platforms has become a promising alternative to ground based vehicles. In these applications the vehicle is often required to operate over unknown terrain or where existing navigation infrastructure such as the Global Positioning System (GPS) is unavailable or unreliable. Even in applications in which satellite navigation is available, methods such as GPS-aided inertial navigation [Gebre-Egziabher et al., 1998] can suffer from several drawbacks such as lack of sufficient satellites/signal dropouts, poor vertical positioning and errors such as multi-path. Thus the ability to perform SLAM on an airborne platform results in a more robust and reliable localisation and mapping system.

Although laser or radar sensors can be used as a range and bearing sensor for SLAM, the weight and cost of these units for airborne vehicles is often prohibitive of their use in most applications. Vision sensors are of greater interest due to their light-weight and low cost. Visual sensors also provide rich information about the colour and texture of the environment that is often desired as an end product in most information gathering tasks such as picture compilation and terrain classification.

Bearing-only SLAM using a vision sensor has been demonstrated before for ground-based vehicles (see Section 2). Performing vision-aided SLAM in unstructured environments on an aerial vehicle poses several additional challenges:

1. **Finding and Recognising Environment Features** - Feature extraction algorithms must often deal with large deviations in image brightness (looking at the ground w.r.t the sky). An aerial vehicle can perform rapid manoeuvres and high-speed flight which affects the sensor pointing direction, features are often in view of the sensor for only a small number of frames at a time.
2. **Filter Conditioning and Stability** - A single bearing-only observation provides insufficient information alone to localise a feature, instead observations from two sufficiently different poses are required. Data association is complicated by bearing only observations and spurious observations in the feature extraction process. The filter must also contend with highly non-linear process and observation models from the nature of the vehicle's motion, resulting in risk of filter inconsistency and divergence.
3. **Computational Burden** - Localisation updates must be computed at a high-rate due to the control

requirements of the vehicle, vehicle motion must be estimated in 6-DoF and the position of features in 3D, adding to the size of the estimated state.

SLAM on an airborne vehicle has been demonstrated in the past: in [Kim and Sukkarieh, 2003], the authors demonstrate SLAM using artificial targets randomly scattered in the environment. A vision camera is used to compute the bearing to each feature along with an estimate of the range based on the known size of each target. The available range information improves the stability of the SLAM solution from the bearing-only case, but limits the applications and environments in which the system can be deployed. In this paper we extend the airborne SLAM algorithms to operate without the need for range observations to features. The resulting system can thus be used in environments where range information cannot be inferred from an image such as in most of the above mentioned applications.

The SLAM algorithm is run using an Extended Kalman Filter (EKF) in which the position, velocity and attitude of the vehicle along with the 3D positions of the features in the environment are estimated. A delayed initialisation technique is used to store information from bearing-only observations until there exists two observations with a sufficient baseline from which to initialise the 3D position of the feature.

Section 2 provides an overview of existing methods for solving the problems faced in bearing-only observations in regards to feature initialisation and provides the reasoning for the approach taken for the aerial vehicle case in this paper. Section 3 details the SLAM algorithms used, bearing-only feature initialisation and data association process. Section 4 describes the physical system and sensors used to drive the SLAM algorithm. Results of the algorithm are shown in Section 5. Conclusions and future work are covered in Section 6.

2 Overview of Bearing-only SLAM

This sections offers a background into existing solutions to aspects of the problems faced in implementing bearing-only SLAM and presents an overview to our approach for an aerial vehicle.

2.1 Overview of Previous Approaches

Once a well conditioned estimate of the feature position is available in SLAM, bearing-only tracking of the feature from subsequent observations can be tackled in the standard EKF framework. The issue however with implementing SLAM using a bearing-only sensor is that the initial 3D position of a feature can not be determined from a single observation and thus the estimate of the feature location is ill-conditioned when represented as a Gaussian. Instead several measurements are required with sufficiently different baseline to determine the initial position accurately.

In the target tracking community, initialisation of 3D position from bearing observations has been tackled for example by representing the initial feature position as a non-Gaussian distribution such as a Gaussian-sum [Alspach and Sorenson, 1972] or with the use of particles [Gordon et al., 1993]. These approaches are less popular for SLAM in which the initial feature position is correlated to vehicle and the rest of the map, thus resulting in high computational burden when correctly applying these representations to a high-dimensional state.

In [Davidson, 2003] and [Fitzgibbons and Nebot, 2002] the authors tackle the initialisation problem by representing the position of the feature initially using particles. In both of these cases the authors maintain the particle distribution decoupled from the estimates of the vehicle and the rest of the map. Observation made of the feature during the particle stage will however be correlated to one another through the vehicle errors and thus ignoring this coupling will result in a loss of information to the vehicle states and can lead to an inconsistent initialisation of the feature.

In [Kwok and Dissanayake, 2004] the authors use a multi-hypothesis filtering approach in which several hypotheses of the position of a landmark are created based along the line of sight of the first observation of a feature. Each of the hypotheses is then integrated into the filter and treated as a separate feature. Subsequent observations as the vehicle moves around the feature will eventually allow all but one of the hypotheses to be pruned out of the filter. Although computationally efficient, this approach is losing information from observations made before initialisation. In [Sola et al., 2005] the authors similarly use a multi-hypothesis approach where the information from further observations of a feature before initialisation is transferred to each hypothesis using federated information sharing. In using this approach however there is no guarantee that estimates will be consistent due to updating for hypotheses that do not really exist.

Another approach to the bearing-only initialisation problem has been delayed initialisation of the feature position into the filter by storing observation and vehicle pose information. In [Bailey, 2003] the author stores the vehicle pose and observation data in the state vector for a single observation and later uses constrained initialisation to compute the feature position when a second observation is available from a sufficiently different vehicle pose. By correctly maintaining correlations between the stored pose and the current vehicle pose, information from the first observation is transferred the current vehicle pose estimate at initialisation in a consistent manner. The disadvantage of this approach is that observations of the feature after the first sighting but before the initialisation are rejected.

[Deans and Herbert, 2000] demonstrates a delayed approach to localisation and mapping with bearing-only observations by applying an adaptation of bundle adjustment. The idea of the bundle adjustment approach is to estimate the vehicle pose and landmark positions by running a batch update with all of the stored observations. The advantage of this approach is that estimates are well conditioned, the disadvantage being that the computational complexity of the algorithm scales with the number of observations and is thus not suitable to real-time applications.

2.2 Overview of the Current Approach

In this paper we take a delayed approach to feature initialisation by storing observations and vehicle poses and recovering this information in a batch update step when sufficient base-line exists between two observations. In our approach the correlations between stored vehicle poses and the current vehicle pose along with the rest of the map is maintained in a consistent manner. It is our belief that initialising the feature in a consistent manner and recovering all of the information from observations made before initialisation is of most importance in SLAM, particularly in the aerial vehicle case where reliability of the navigation system is paramount.

The aim of our approach is to benefit from well conditioned SLAM estimates by storing observations and delaying our update while at the same time maintaining the ability of the SLAM filter to be run in real-time, necessary for navigation feedback for autonomous vehicle control.

In most of the previous approaches to bearing-only SLAM, data association has either been ignored or in the case of using vision-based sensors has been based on matching the visual properties (i.e. colour, texture) of a feature between observations. There are however some situations in which data association using visual properties may not be ideal such as multiple features in an environment looking the same or changes in lighting or appearance of features from different viewing angles. The approach in this paper is therefore to develop a method for data association that does not depend upon visual characteristics of a feature thus extending the potential environments the vehicle can operate within.

3 Airborne SLAM Algorithm

This section describes the SLAM algorithm, feature initialisation and data association process.

3.1 Extracting Features from Image Data

Figure 1 shows a sample image from the on-board camera taken while in flight. The feature extraction process finds the normalised intensity of each pixel in the image, applies a intensity threshold and finds which pixels lie above the threshold (image on right-hand side of Figure 1). The feature extraction process then finds groups of interconnected pixels and generates an observation for each group whose pixel count and dimensions fall within given bounds. The values of the bounds are manually tuned based on finding features that are useful for the current task.

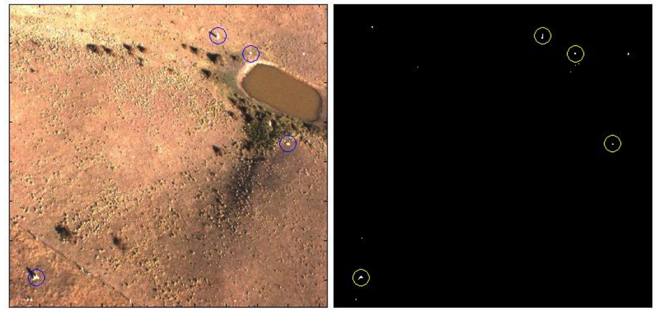


Figure 1: Feature Extraction: A sample vision frame alongside intensity thresholded image. Features are extracted as appropriately sized groups of connected pixels that pass the threshold.

In our case, surveyed 1x1 meter white plastic squares have been placed in the environment to act as a ‘truth’ from which to test the accuracy of the final SLAM map. The feature extraction process however also finds other features in the environment both man-made (e.g. rain water tanks) and natural (tree stumps, tree branches, rocks and patches of dirt) that are integrated into the map. Current work in progress is looking at feature extraction for more interesting natural features such as trees, shrubs, lakes and landscape features such as drainage scars.

3.2 The Inertial SLAM Algorithm

The inertial SLAM algorithm, as shown in [Kim and Sukkarieh, 2003], is formulated using an Extended Kalman Filter (EKF) in which feature locations and the vehicle’s position, velocity and attitude are estimated using relative observations between the vehicle and each feature.

Process Model

The estimated state vector $\hat{x}(k)$ contains the three-dimensional vehicle position (p^n), velocity (v^n) and Euler angles ($\Psi^n = [\phi, \theta, \psi]$) and the N feature locations (m_i^n) in the environment where $i = 1, ..., N$. The state estimate $\hat{x}(k)$ is predicted forward in time from $\hat{x}(k-1)$ via the process model:

$$\hat{x}(k) = \mathbf{F}(\hat{x}(k-1), u(k), k) + w(k) \tag{1}$$

where $\mathbf{F}(.,.,k)$ is the non-linear state transition function at time k, $u(k)$ is the system input at time k and $w(k)$ is uncorrelated, zero-mean vehicle process noise errors of covariance $\mathbf{Q}$. The vehicle process model uses the six-degree of freedom equations of inertial navigation to predict the position, velocity and attitude of the vehicle. An inertial-frame mechanization is used where position, velocity and attitude are found via:

$$\begin{bmatrix} p^n(k) \\ v^n(k) \\ \Psi^n(k) \end{bmatrix} = \begin{bmatrix} p^n(k-1) + v^n(k)\Delta t \\ v^n(k-1) + [C_b^n(k-1)f^b(k) + g^n]\Delta t \\ \Psi^n(k-1) + E_b^n(k-1)\omega^b(k)\Delta t \end{bmatrix} \tag{2}$$

where f^b and ω^b are the body-frame referenced vehicle accelerations and rotation rates which are provided by inertial sensors on the vehicle and g^n is the acceleration due to gravity. The direction cosine matrix C_b^n and rotation rate transformation matrix E_b^n between the body and navigation frames are given by:

$$C_b^n = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3}$$

$$E_b^n = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi sec_\theta & c_\phi sec_\theta \end{bmatrix} \tag{4}$$

where $s_{(.)}$, $c_{(.)}$ and $t_{(.)}$ represent sin(.), cos(.) and tan(.) respectively. Feature locations are assumed to be stationary and thus the process model for the position of the i^{th} feature is given as:

$$m_i^n(k) = m_i^n(k-1) \tag{5}$$

where $i = 1, ..., N$, N is the number of features.

Observation Model

An on-board vision sensor makes relative bearing observations $z_i(k)$ to features within the camera frame. The observations are related to the estimated states via:

$$z_i(k) = \mathbf{H}_i(p^n(k), \Psi^n(k), m_i^n(k), k) + v(k) \tag{6}$$

where $\mathbf{H}_i(.,.,.,k)$ is a function of the feature location, vehicle position and Euler angles and $v(k)$ is uncorrelated, zero-mean observation noise errors of covariance $\mathbf{R}$. The observation model is given by:

$$z_i(k) = \begin{bmatrix} \varphi_i \\ \vartheta_i \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{y^s}{x^s}\right) \\ \tan^{-1}\left(\frac{z^s}{\sqrt{(x^s)^2+(y^s)^2}}\right) \end{bmatrix} \tag{7}$$

where φ_i and ϑ_i are the observed azimuth and elevation angles to the feature and x_s, y_s and z_s are the cartesian co-ordinates of p_{ms}^s, the relative position of the feature w.r.t the sensor, measured in the sensor frame. p_{ms}^s is given by:

$$p_{ms}^s = C_b^s C_n^b [m_i^n - p^n - C_b^n p_{sb}^b] \tag{8}$$

where C_b^s is the transformation matrix from the body frame to sensor frame and p_{sb}^b is the sensor offset from the vehicle centre of mass, measured in the body frame.

A pinhole camera model is used to determine the azimuth and elevation angles from the pixel co-ordinates (u,v) of the feature in the image:

$$\begin{bmatrix} \varphi \\ \vartheta \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{(u-u_0)}{f_u} \\ \tan^{-1}(\tan(\frac{(v-v_0)}{f_v}) \cos\varphi) \end{bmatrix} \tag{9}$$

where u_0,v_0,f_u and f_v are calibration parameters for the camera which are determined before flight.

Estimation Process

The estimation process is recursive and is broken into two steps:

Prediction: The vehicle position, velocity and attitude are predicted forward in time using (1) and (2) with data supplied by the inertial sensors. The state covariance $\mathbf{P}$ is propagated forward via:

$$\begin{aligned} \mathbf{P}(k|k-1) =& \nabla\mathbf{F}_x(k)\mathbf{P}(k-1|k-1)\nabla\mathbf{F}_x^T(k) \\ &+ \nabla\mathbf{F}_w(k)\mathbf{Q}\nabla\mathbf{F}_w^T(k) \end{aligned} \tag{10}$$

where $\nabla\mathbf{F}_x$ and $\nabla\mathbf{F}_w$ are the jacobians of the state transition function w.r.t the state vector $\hat{x}(k)$ and the noise input $w(k)$ respectively.

Update: Assuming that we have already initialised the three-dimensional position of a feature, the state estimate is updated from further observations via:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \mathbf{W}(k)\nu(k) \tag{11}$$

where the gain matrix $\mathbf{W}(k)$ and innovation $\nu(k)$ are calculated as:

$$\nu(k) = z_i(k) - \mathbf{H}_i(\hat{x}(k|k-1)) \tag{12}$$

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\nabla\mathbf{H}_x^T(k)\mathbf{S}^{-1}(k) \tag{13}$$

$$\mathbf{S}(k) = \nabla\mathbf{H}_x(k)\mathbf{P}(k|k-1)\nabla\mathbf{H}_x^T(k) + \mathbf{R} \tag{14}$$

where $\nabla\mathbf{H}_x(k)$ is the jacobian of the observation function w.r.t the predicted state vector $\hat{x}(k|k-1)$. The state covariance $\mathbf{P}(k|k)$ after the observation is updated via:

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k) \tag{15}$$

3.3 Initialisation of Feature Positions from Bearing-only Observations

As mentioned in Section 3, a single bearing-only observation is insufficient to initialise the 3D position of a feature into the SLAM filter with Gaussian uncertainty. In the following subsection, we outline a method for delayed initialisation of a feature into the filter by using stored observations and vehicle pose information.

Storing Feature Observations and Vehicle Pose Information

When an observation of an un-initialised feature is made, the current bearing-only observation is stored and the SLAM state vector and covariance matrix are augmented to include the current vehicle pose (3 position states and 3 Euler angle states):

$$\hat{x}_v = \begin{bmatrix} p^n(k) \\ v^n(k) \\ \Psi^n(k) \end{bmatrix}, \hat{x}_p = \begin{bmatrix} p^n(k) \\ \Psi^n(k) \end{bmatrix} \tag{16}$$

$$\hat{x}_{aug} = \begin{bmatrix} \hat{x}_v(k) \\ m^n(k) \\ \hat{x}_p(k) \end{bmatrix} \tag{17}$$

$$\mathbf{P}_{aug}(k) = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} & \mathbf{P}_{vp} \\ \mathbf{P}_{mv} & \mathbf{P}_{mm} & \mathbf{P}_{mp} \\ \mathbf{P}_{pv} & \mathbf{P}_{pm} & \mathbf{P}_{pp} \end{bmatrix} \tag{18}$$

where $\hat{x}_v$ are the vehicle position, velocity and attitude states, $\hat{x}_p$ are the vehicle pose states (position and attitude) at the time of the observation and $\hat{x}_{aug}$ is the augmented state vector.

Deciding when to Initialise a Feature

A remaining issue with the delayed initialisation method is the question of when the initialisation should be made. In [Bailey, 2003], the author uses the Kullback-Leibler distance to determine whether to initialise a feature based on available measurements by comparing result of the initialisation with a non-gaussian approximation. This method is very computationally intensive due to the requirement for a numerical evaluation of the Kullback-Leibler distance. In our approach it is found that a heuristic such as setting the minimum angle between observations necessary to initialise a feature to a conservative threshold (in our case 40 degrees) provides a practical way of deciding when to initialise. In our case all of the information from observations is recovered at initialisation and hence the SLAM algorithm can afford to delay the initialisation (by setting the angular threshold at a conservatively large angle) provided that we continue to circle around a feature.

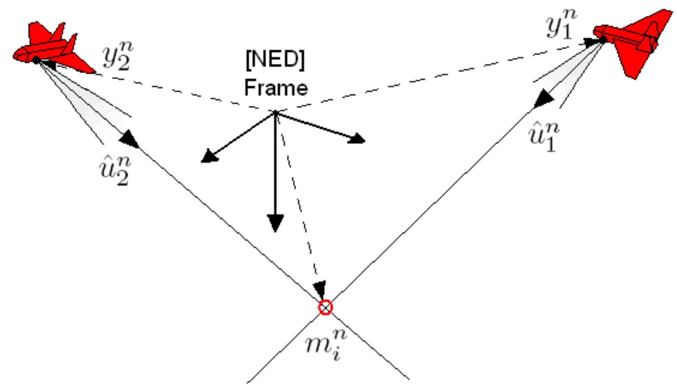


Figure 2: Initialising a 3D feature position from two observations from different vehicle poses: position is calculated as the closest point between 2 lines of sight in 3D space.

Initialising a 3D Feature Position Estimate

When two observations for a given un-initialised feature are separated by an angle greater than 40 degrees, these two observations are initially used to create an estimate of the feature position. Each bearing-only observation can be represented by a 3D point in space y^n (at the origin of the sensor) along with a unit vector $\hat{u}^n$ pointing in the direction of the feature at the time of observation, thus:

$$y^n = p^n + C_b^n p_{sb}^b \tag{19}$$

$$\hat{u}^n = C_b^n C_s^b \hat{p}_{ms}^s \tag{20}$$

where p^n and C_b^n are determined from the stored pose data associated to each observation and $\hat{p}_{ms}^s$ is determined from the observation data itself via:

$$\hat{p}_{ms}^s = \begin{bmatrix} \cos(\varphi_i)\cos(\vartheta_i) \\ \sin(\varphi_i)\cos(\vartheta_i) \\ \sin(\vartheta_i) \end{bmatrix} \tag{21}$$

The initial feature position is computed as the closest point between the two lines for each observation:

$$\begin{aligned} m_i^n &= \mathbf{G}(p_1^n, p_2^n, \Psi_1^n, \Psi_2^n, z_1, z_2) \\ &= \frac{1}{2}(y_1^n + y_2^n + p_1\hat{u}_1^n + p_2\hat{u}_2^n) \end{aligned} \tag{22}$$

$$p_1 = \frac{((y_2^n - y_1^n) \times \hat{u}_2^n) \cdot (\hat{u}_1^n \times \hat{u}_2^n)}{|\hat{u}_1^n \times \hat{u}_2^n|^2} \tag{23}$$

$$p_2 = \frac{((y_1^n - y_2^n) \times \hat{u}_1^n) \cdot (\hat{u}_2^n \times \hat{u}_1^n)}{|\hat{u}_2^n \times \hat{u}_1^n|^2} \tag{24}$$

The state vector and covariance matrix in the SLAM filter are then augmented to include the estimate of the new feature:

$$\hat{x}_{aug}(k) = \begin{bmatrix} \hat{x}(k) \\ m_i^n(k) \end{bmatrix} \tag{25}$$

$$\mathbf{P}_{aug}(k) = \begin{bmatrix} \mathbf{I} & 0 \\ \nabla\mathbf{G}_p & \nabla\mathbf{G}_z \end{bmatrix} \begin{bmatrix} \mathbf{P}(k) & 0 \\ 0 & \mathbf{R}_{2x2} \end{bmatrix} \times \begin{bmatrix} \mathbf{I} & 0 \\ \nabla\mathbf{G}_p & \nabla\mathbf{G}_z \end{bmatrix}^T \quad (26)$$

where $\nabla\mathbf{G}_p$ and $\nabla\mathbf{G}_z$ are the jacobians of the initialization function $\mathbf{G}(.)$ w.r.t the pose states $(p_1^n, p_2^n, \Psi_1^n, \Psi_2^n)$ and the observations (z_1, z_2) respectively and $\mathbf{R}_{2x2}$ is:

$$\mathbf{R}_{2x2} = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix} \quad (27)$$

Recovering the Information from Remaining Stored Observations

Once two observations have been used to initialise the 3D position of the feature into the filter, the remaining stored observations $(z_1, z_2, ..., z_j)$ are run through a batch Kalman filter update, in order to recover information towards the position of the feature and also the current vehicle pose states. The update follows similar steps as in Section 3.2. The innovation $\nu(k)$ is:

$$\nu(k) = \begin{bmatrix} z_1 - \mathbf{H}(p_1^n, \Psi_1^n, m_i^n) \\ z_2 - \mathbf{H}(p_2^n, \Psi_2^n, m_i^n) \\ \vdots \\ z_j - \mathbf{H}(p_j^n, \Psi_j^n, m_i^n) \end{bmatrix} \quad (28)$$

Equations 11 and 13-15 are then used to calculate the update where $\nabla\mathbf{H}_x$ is computed w.r.t all of the stored pose states corresponding to observation data used. Once the update has been completed, pose states that no longer have any associated stored observations are removed from the state vector and their corresponding rows and columns removed from the covariance matrix.

3.4 Data Association

Data association is the process of matching observations of features from the camera (which are generally provided as 2D points in the image which are not necessarily distinct from any other point) with the estimated 3D position of the feature within the map. The aim of this section is to develop a method of data association that does not rely on any visual information about a feature from the image data.

Starting a New Feature

When an observation is made in the image that cannot be associated to any other previously seen feature, initialised or un-initialised, a new feature is created using the observation. The feature is termed an 'un-initialised feature' as we have insufficient information to determine the 3D position of the feature and integrate the estimate of it's position into the 3D feature map. From our single observation we create a set of equally weighted hypotheses for where the feature could lie in 3D space along the line of sight. The mean $(\hat{x}_j)$ and covariance $(\mathbf{P}_j)$ for each hypothesis is calculated for several different values of range (r_j) in equal increments from an expected minimum and maximum sensor range by:

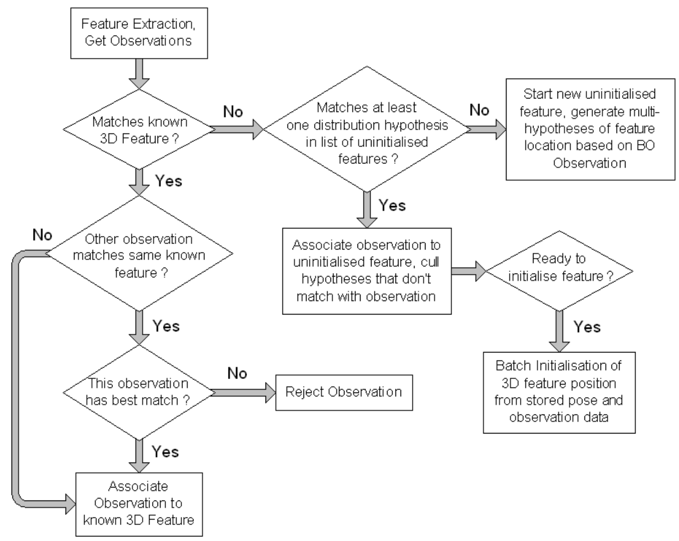


Figure 3: Data Association Procedure for initialised and un-initialised Features.

$$\begin{aligned} \hat{x}_j &= \mathbf{G}(p^n(k), \Psi^n(k), z_i(k), r_j) \\ &= p^n + C_b^n p_{sb}^b + r_j(C_b^n C_s^b p_{ms}^s) \quad (29) \\ \mathbf{P}_j &= \nabla\mathbf{G}_v \mathbf{P}_{vv} \nabla\mathbf{G}_v^T + \nabla\mathbf{G}_z \mathbf{R} \nabla\mathbf{G}_z^T \quad (30) \end{aligned}$$

where p_{ms}^s is calculated from the observation data using Equation 21 and $\nabla\mathbf{G}_v, \nabla\mathbf{G}_z$ are the jacobians of the function $\mathbf{G}(.)$ w.r.t vehicle states and the observation and range data respectively.

A record of the multi-hypothesis distribution is maintained separately from the state vector and is used only to assist in data association of un-initialised features.

Data Association Matching Test

The validity of potential associations between observations and features is assessed using the Mahalanobis distance (γ) [Neira and Tardos, 2002] in the sensor space (azimuth and elevation):

$$\gamma = \nu^T \mathbf{S}^{-1} \nu \quad (31)$$

where the distance is calculated for each possible matching of observation and feature. Matchings that fall within a defined threshold of γ corresponding to a 95% level of confidence are considered acceptable.

Matching Observations to Initialised or Un-initialised Features

At each time observations from the feature extraction process are received, Equation 31 is used to evaluate the potential matching between each observation and each of the 3D initialised features and each of the hypotheses for each of the un-initialised features. Observations that match 3D initialised features are associated and sent on to the SLAM filter to be updated. In the event of multiple features matching a single observation, the matching with the lowest value of γ will be accepted.

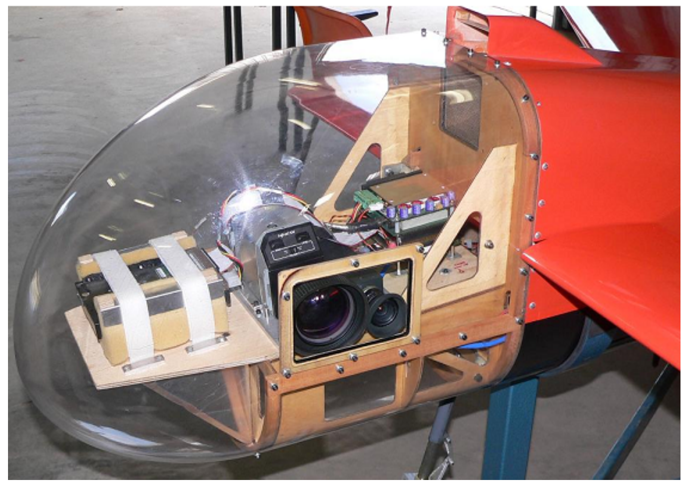
Observations that match with at least one hypothesis of an un-initialised feature are associated to this feature. Once an observation is associated to an un-initialised feature, any hypotheses for this feature that do not match the observation are removed. The observation itself is stored and the vehicle pose at the current time is then added to the state vector (see Section 3.3). In the event of multiple un-initialised features matching a single observation, all matchings to this observation are rejected.

The matching process is illustrated is Figure 3.

4 Experimental Setup

The experimental setup comprises of the flight vehicle, a low-cost IMU, colour vision camera and PC104 computer setup on the aircraft which is currently used to log data during flight tests (see Figure 4). The flight vehicle has an autonomous flight control system that follows a fixed path of orbits around features of interest on the ground. The current vehicle navigation system uses an on-board GPS receiver with differential corrections via a base-station located on the ground to aid the IMU. The current navigation system results provide a reasonable approximation to the true position, velocity and attitude of the vehicle (1-2 meter positioning accuracy and 1-2 degrees orientation accuracy) and will be used as a comparison to the SLAM navigation system results. Several 1x1 meter white plastic squares were placed in the environment to act as artificial features. The position of each white square was surveyed using differential GPS and is used to compare the accuracy of the SLAM map.

(a)



(b)

Figure 4: (a) The Brumby MkIII UAV, weighing 40kg with a wing span of 2.8 meters, capable of carrying a payload of 13.5 kg and flying at 100kts (b) Sideways mounted colour camera and PC104 computer stack

5 Results

The following results were generated by using logged vision and IMU data to drive the SLAM algorithms. Figure 5 shows four captured vision frames with observations, the projected positions of 3D initialised features and multiple hypotheses of un-initialised features overlaid on the image. As features are seen for the first time, an array of hypotheses represented by the green ellipses is projected into the image. Further observations of the feature begin to cull unmatched hypotheses before finally the angular threshold for different observations of a feature is reached and the 3D position of the feature is initialised into the map and SLAM filter.

The trajectory of the vehicle and the position of point features in the map is shown in Figure 6 with results from both the GPS-aided INS solution and the un-aided INS shown for comparison. The blue numbered points on the map represent the estimated position of features in the environment from the SLAM filter and the red points indicate the surveyed position of the artificial white plastic targets. Only a handful of the estimated map features correspond to white targets where the other features are collections of tree stumps, rocks, bright patches of dirt and a rain water tank. Although the mapped features may be of little interest to a human wishing to build a picture of the environment, these features can be repeatedly extracted from the vision data and associated by the SLAM algorithm and thus their usefulness for navigation purposes.

Plots of the position and attitude of the vehicle from

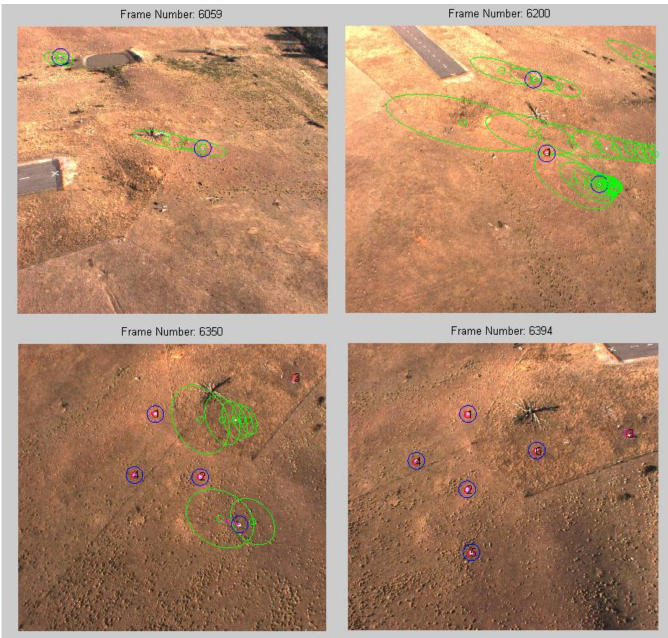


Figure 5: Four captured vision frames with observations (blue), the projected positions of 3D initialised features (red) and multiple hypotheses of un-initialised features (green) overlaid on the image.

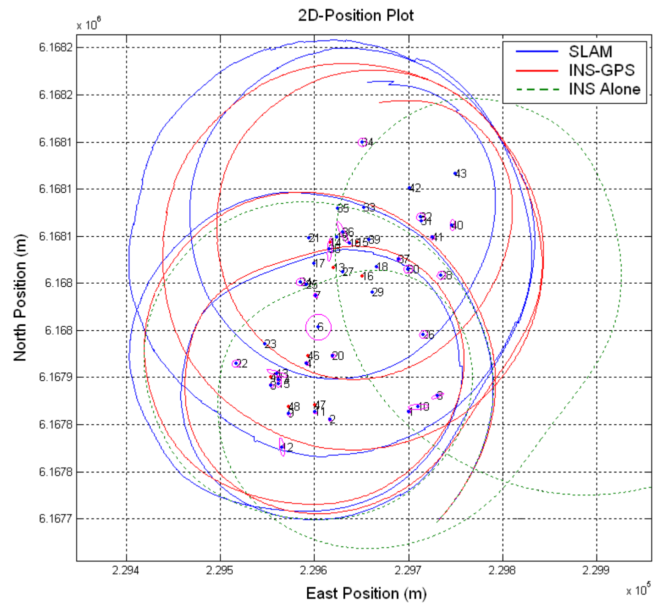


Figure 6: Vehicle trajectory from SLAM results (blue) and INS-GPS system (red) and un-aided INS (green), SLAM mapped features (blue points) and surveyed white plastic target locations (red points).

the SLAM solution are shown in Figures 7 and 8. The estimates of the vehicle Euler angles show little deviation from the GPS-aided results (within about 1-2 degrees), however the SLAM position solution deviates as much as 15 meters in vertical positioning and as much as 30 meters in horizontal positioning.

6 Conclusions and Future Work

This paper has demonstrated an implementation of vision-aided, bearing-only SLAM on an aerial vehicle using data logged during a flight test. Bearing-only initialisation of features in the map has been tackled using a delayed, batch Kalman filter update using stored observation and vehicle pose data. A method for data association that does not rely on visual properties of features from the image data has been shown. It has been shown that the SLAM algorithm can constrain the errors in the inertial navigation system while operating without GPS in an unstructured environment without the need for range observations. The algorithm builds a 3D point feature map of the environment consisting of both man-made artificial features and natural features.

Current work in progress is looking at integrating more sophisticated feature extraction algorithms for finding more interesting natural features such as trees, shrubs, lakes and landscape features such as drainage scars. Future work will examine methods for further improving the consistency of the SLAM algorithms for operation over large scale areas.

Acknowledgment

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

[Braun et al., 2004] R.D.Braun, H.S. Wright, M.A. Croom, J.S. Levine, D.A. Spencer, "The Mars Airplane: a Credible Science Platform," IEEE Aerospace Conference, Big Sky, MT, 2004.

[Gebre-Egziabher et al., 1998] D. Gebre-Egziabher, R.C. Hayward, J.D. Powell, "A Low-cost GPS/inertial Attitude Heading Reference System (AHRS) for General Aviation Applications," IEEE Position Location and Navigation Symposium, 1998.

[Kim and Sukkarieh, 2003] J.H. Kim, S. Sukkarieh, "Airborne Simultaneous Localization and Map Building," IEEE International Conference on Robotics and Automation, Taipei, 2003.

[Alspach and Sorenson, 1972] D.L. Alspach, H.W. Sorenson, "Nonlinear Bayesian Estimation using Gaussian Sum Approximations," IEEE Trans. on Automatic Control, vol. 17, no. 4, pp. 439-448, 1972.

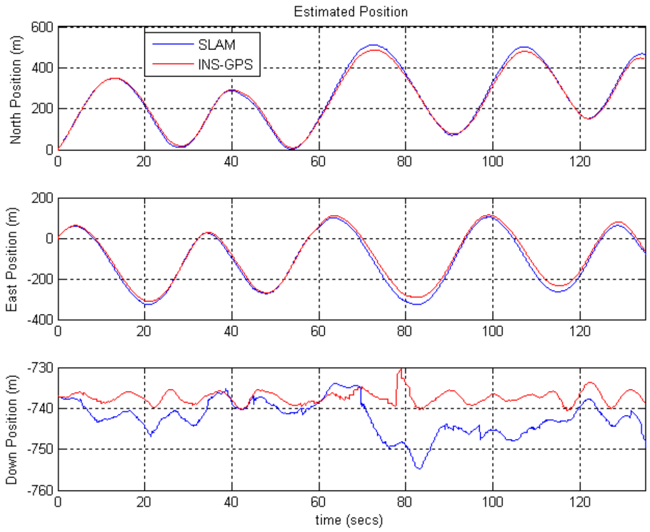


Figure 7: Comparison of SLAM (blue) and INS-GPS (red) Estimated Position.

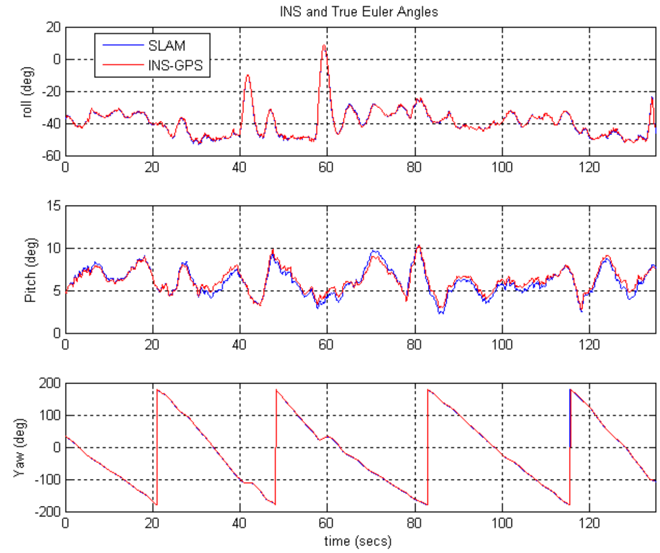


Figure 8: Comparison of SLAM (blue) and INS-GPS (red) Estimated Euler Angles.

[Gordon et al., 1993] N. J Gordon, D.J. Salmond and A.F.M. Smith, "A Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," IEE Proceedings-F, Radar and Signal Processing, vol. 140, no. 2, pp.107-113, 1993.

[Kwok and Dissanayake, 2004] N.M. Kwok, G. Dissanayake, "An Efficient Multiple Hypothesis Filter for Bearing Only SLAM," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.

[Davidson, 2003] A.J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," IEEE Conference on Computer Vision, 2003.

[Sola et al., 2005] J. Sola, A. Monin, M. Devy, T. Lemaire, "Undelayed Initialisation in Bearing-Only SLAM," IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, 2005.

[Fitzgibbons and Nebot, 2002] T. Fitzgibbons, E. Nebot, "Bearing-Only SLAM using Colour-based Feature Tracking," Australasian Conference on Robotics and Automation, Auckland, 2002.

[Bailey, 2003] T. Bailey, "Constrained Initialisation for Bearing-Only SLAM," IEEE International Conference on Robotics and Automation, Taipei, 2003.

[Deans and Herbert, 2000] M. Deans and M. Hebert, "Experimental Comparison of Techniques for Localization and Mapping using a Bearings Only Sensor," 7th International Symposium on Experimental Robotics, Honolulu, December, 2000.

[Neira and Tardos, 2002] J. Neira, J.D. Tardos, "Data association in stochastic mapping using the joint compatibility test," IEEE Trans. on Robotics and Automation, vol. 17, no. 6, pp. 890-897, December 2001.