Bearings-Only Localisation of Targets from Low-Speed UAVs*

Mohibullah W.
Department of Computer Science
University College London
Gower Street, London
WC1E 6BT, UK
W.Mohibullah@cs.ucl.ac.uk

Simon J. Julier
Department of Computer Science
University College London
Gower Street, London
WC1E 6BT, UK
S.Julier@cs.ucl.ac.uk

Abstract – *We perform a simulation study to compare the performance of several of single-camera mapping and localisation algorithms for a search and rescue application using low-speed, low-altitude UAVs. We investigate the effects of a range of conditions including target location, nadir angle of the camera and the trajectory of the UAV on three bearings-only SLAM algorithms: Delayed Initialisation (DI), Inverse Depth Point (IDP), and Anchored Homogeneous Point (AHP). Our results show that DI is robust but there can be significant delays before a landmark is initialised. IDP can produce landmark estimates of similar quality, but without the delays. However, this performance can only be achieved through the use of log parameterisations of depth and second order filters. AHP does not yield consistent estimates under any circumstance and is not appropriate for our application.*

Keywords: SLAM, UAV, Kalman filter, bearings-only tracking.

1 Introduction

In this paper we consider how Unmanned Aerial Vehicles (UAVs) can be used to aid Wilderness Search and Rescue (WiSAR). One of the most critical phases of WiSAR is *search* — until a missing person has been found, they cannot be rescued or recovered. Key to this task is the airborne collection of *evidence* [1] — according to the SAR Spotlight Forum, a missing person leaves around 2000 features (or clues) for every mile walked [2]. Although many of these clues — such as broken twigs and bruised vegetation — are probably too small to be automatically detected from the air, a small subset (such as discarded clothing or food wrappers) might be sufficient to infer the trajectory or intent of a missing person. However, critical to this inference is the georeferencing of the clues within the environment. Therefore, in this paper we consider the problem of accurately locating point-like objects using camera-equipped UAVs.

The problem of localising targets from UAVs is a heavily studied problem. Because of the high altitude and platform constraints, much of this work has been posed as bearings-only simultaneous localisation and mapping (MonoSLAM) problems. When applied to UAVs, the assumption is of a fixed wing UAV which flies at a high altitude and a constant velocity. However, in our target application we use the Ascending Technologies Hummingbird quadrotor helicopters illustrated in Figure 1 [3]. These are lightweight, low cost, highly agile and stable platforms that have the ability to loiter and readily change altitude even in strong gusts of wind. As such, their trajectories can be very different from fixed wing UAVs and thus their observability properties could be very different. However, very few studies have explored these properties in our domain. Although some papers consider the problems of mapping using range-bearing sensors indoors, the only work we are aware of for outdoor environments was a feasibility study [4, 5]. This showed that, for a given choice of algorithm, consistent results could be attained. No systematic study of the performance of the system was carried out.

Figure 1: An AscTec Hummingbird quadrotor helicopter.

In this paper, we perform a detailed comparison of the performance of three MonoSLAM algorithms: Delayed Initialisation (DI) [6], Inverse Depth Parameterisation (IDP) [7], and the Anchored Homogeneous Point algorithm (AHP) [8]. Since both second order [9] and iterated [10] filters can have a significant effect, we assess the performance of these algorithms as well. Further, the nadir angle of the camera on the UAV body is modified. This tradeoffs coverage resolution with area. Finally, since the problem is fundamentally one of bearings-only tracking, the trajectory

*This work was supported by the SUAAVE project. More information on this project can be found at http://www.suaave.org.

of the platform can have a significant effect on observability [11–13] and so we also investigate the impact of the trajectory on performance.

Our results show that the behaviour of these algorithms differ greatly from those published before. From an algorithmic perspective, we find that second order filters have a significant impact upon performance, but iterated forms of the filters do not. We find that the AHP gives inferior response in all cases, and is not recommended. DI gives the most robust performance, but at the cost of significant latency in initialising the features. Furthermore, a sizeable fraction of features are never fully initialised. The IDP appears to give the best performance overall both in terms of immediate availability of landmarks and estimation accuracy. However, this robustness can only be achieved using a log parameterisation of depth.

The structure of the paper is as follows. The problem statement is discussed in Section 2. In Section 3 we describe the three algorithms — delayed initialisation, IDP and AHP — which are used. The simulation scenario is described in Section 4 and the analysis and results are presented in Section 5. We summarise and conclude in section 6.

2 Problem Statement

2.1 System Description

Consider the situation shown in Figure 2: a UAV, equipped with a GPS, an IMU and a camera, flies over a wildereness terrain and searches for spatially-distributed clues about a missing person's whereabouts. The clues, such as discarded clothing or individual footprints, are assumed to be sufficiently small that they can be adequately modelled as point-like features. When the UAV detects a new feature, it might engage in a manoeuvre to optimise the location of the feature [11]. However, since its trajectory is controlled by high-level decision-making algorithms [14], it might conclude that the new feature is of low priority and does not warrant immediate investigation. However, the UAV might return to the feature at a later data. Therefore, a consistent estimate of the feature location must be estimated.

Using the notation from [6], the system state is composed of the vehicle state $\mathbf{x}_v(k)$ and the map state $\mathbf{m}(k)$. The vehicle state is composed of the platform position $\mathbf{p}^n(k) = [x^n\ y^n\ z^n]^T$, velocity $\mathbf{v}^n(k) = [u^n\ v^n\ w^n]^T$ and the attitude $\mathbf{\Psi}^n(k)$, all expressed in the navigation frame n. The physical constraints of the platform mean that the pitch and roll angles cannot exceed $19.4°$. Therefore, the attitude is parameterised as the standard roll-pitch-yaw Euler parameterisation, $\mathbf{\Psi}^n(k) = [\phi^n\ \theta^n\ \psi^n]$. The map state $\mathbf{m}(k)$ is a collection of the positions of each feature $\mathbf{m}_i^n(k)$.

The discrete-time process model is

$$\mathbf{x}(k) = \mathbf{f}\left[\mathbf{x}(k-1), \mathbf{u}(k), \mathbf{v}(k), \Delta T_k\right],$$

where $\mathbf{f}(\cdot,\cdot,\cdot)$ is the state transition equation, $\mathbf{u}(k)$ is the control input (the linear and angular accelerations measured by an onboard tri-axial IMU), $\mathbf{v}(k)$ is the process noise vector (dominated by the measurement errors in the IMU), and ΔT_k is the length of the timestep.

There are two classes of observation model. For the GPS, the observation is a function of the platform state only,

$$\mathbf{z}_{\text{GPS}}(k) = \mathbf{h}\left[\mathbf{p}^n(k), \mathbf{w}_{\text{GPS}}(k)\right]$$

where $\mathbf{z}_{\text{GPS}}(k)$ is the obsevation (position expressed in the navigation frame) and $\mathbf{w}_{\text{GPS}}(k)$ is the GPS observation noise. For the ith feature on the ground, the observation is

$$\mathbf{z}_i(k) = \mathbf{h}\left[\mathbf{p}^n(k), \mathbf{\Psi}^n(k), \mathbf{m}_i^n(k), \mathbf{w}(k)\right],$$

where $\mathbf{z}_i(k)$ is the observation of the ith feature (expressed in pixel coordinates) and $\mathbf{w}(k)$ is the observation error.

The full equations for constructing this observation model can be found in [6].

2.2 Estimation Problem

We address this problem through the use of the stochastic map formulation of the Simultaneous Localisation and Mapping (SLAM) problem [15, 16]. The estimate of $\mathbf{x}(k)$, using observations up to time step j, is $\{\hat{\mathbf{x}}(k \mid j),\ \mathbf{P}(k \mid j)\}$ where $\hat{\mathbf{x}}(k \mid j)$ is the estimated mean and $\mathbf{P}(k \mid j)$ the estimated covariance. This can be decomposed into the platform and map states,

$$\hat{\mathbf{x}}(k \mid j) = \begin{bmatrix} \hat{\mathbf{x}}_v^n(k \mid j) \\ \hat{\mathbf{m}}^n(k \mid j) \end{bmatrix},$$

$$\mathbf{P}(k \mid j) = \begin{bmatrix} \mathbf{P}_{vv}(k \mid k) & \mathbf{P}_{vm}(k \mid j) \\ \mathbf{P}_{mv}(k \mid j) & \mathbf{P}_{mm}(k \mid j) \end{bmatrix}.$$

The error in the estimate is $\tilde{\mathbf{x}}(k \mid j) = \mathbf{x}(k) - \hat{\mathbf{x}}(k \mid j)$. The estimate is said to be *covariance consistent* if it obeys the condition

$$\mathbf{P}(k \mid j) - \mathrm{E}\left[\tilde{\mathbf{x}}(k \mid j)\,\tilde{\mathbf{x}}^T(k \mid j)\right] \geq \mathbf{0},$$

where $\geq \mathbf{0}$ means that the difference is positive semidefinite.

3 MonoSLAM Algorithms Used

The main challenge with MonoSLAM algorithms is that the position of a feature is not globally observable from a single measurement; as a result, its initial position is poorly known. Qualitatively, a feature can be considered to be in one of two conditions: *poorly-localised*, and *well-localised*. In the poorly localised condition, the position uncertainty is sufficiently large that a Kalman-filter based representation of Cartesian position performs poorly. In the well-localised condition, the feature uncertainty is sufficiently small that regular Kalman filter updates can be applied.

The three algorithms we consider here — Delayed Initialisation, Inverse Depth Point and Anchored Homogeneous Point — all utilise this basic structure, but the way in which they handle the poorly-localised condition is different.

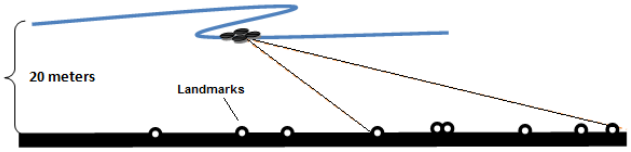


Figure 2: The experimental scenario. A UAV flies over ground and observes a set of sparse, point-like features using an onboard camera.

3.1 Delayed Initialisation (DI)

In this approach, it is assumed that a feature cannot be used when it is poorly localised. Instead, the initialisation of the feature is delayed until it can be directly initialised into the well-localised condition. This is achieved by waiting until the feature has been observed from a sufficiently large base line that its position can be estimated accurately using triangulation [4, 6, 17]. The algorithm has three steps: *storing feature observations*, *initialising the feature*, and *utilising the stored observations*. We summarise and simplify the steps here; a full description can be found in [6].

Suppose a new feature is first observed at time step k_1 but the baseline is not sufficiently large to initialise the landmark until time step k_2. Rather than discard all the feature observations $\mathbf{z}_i(k)$, $k_1 < k < k_2$, they are stored for future processing. Furthermore, because they are platform-centric, the state of the platform when the observations were taken must be recorded. This is achieved through the use of an *augmentation operator*, which constructs a new estimate by appending the platform's current pose to the state vector. Let

$$\hat{\mathbf{x}}_p\left(k|k-1\right) = \begin{bmatrix} \hat{\mathbf{p}}^n(k|k-1) \\ \hat{\boldsymbol{\Psi}}^n(k|k-1) \end{bmatrix}.$$

Then

$$\hat{\mathbf{x}}\left(k \mid k\right) = \begin{bmatrix} \hat{\mathbf{x}}_v\left(k|k-1\right) \\ \hat{\mathbf{m}}^n\left(k|k-1\right) \\ \hat{\mathbf{x}}_p\left(k|k-1\right) \end{bmatrix}$$

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

where $\mathbf{P}_{pp}$ is the covariance of the pose states, $\mathbf{P}_{pm}$ is the cross correlation of the pose state and the map states and $\mathbf{P}_{pv}$ is the cross correlation between the current vehicle state and current pose states. Associated with each stored pose is an observation of the landmark. This is converted from a pixel coordinate via a pinhole projection model to a two-dimensional bearing observation.

Once a sufficiently large baseline has occurred, the initial position of the feature is estimated by the inverse observation function

$$\mathbf{m}_i^n = \mathbf{g}\left[\mathbf{x}_p\left(k_1\right), \mathbf{x}_p\left(k_2\right), \mathbf{z}_i\left(k_1\right), \mathbf{z}_i\left(k_2\right)\right].$$

This function constructs the epipolar lines are constructed at k_1 and k_2 and the feature is initialised at the point which lies at the shortest distance between the two lines. Specifically, the epipolar line at k_1 starts at $\mathbf{p}^n\left(k_1\right)$ and its direction is given by the unit vector

$$\overline{\mathbf{u}}_i^n(k_1) = \mathbf{C}_b^n(k_1)\mathbf{C}_s^b\overline{\mathbf{u}}_i^s(k_1), \tag{1}$$

where $\mathbf{C}_b^n(k_1)$ is the navigation-to-body frame transformation matrix (computed from $\boldsymbol{\Psi}^n(k_1)$), $\mathbf{C}_s^b$ is the transformation between the body and camera frame and encodes the nadir angle, and $\overline{\mathbf{u}}_i^s(k_1)$ is the unit vector from the camera centre to the feature, expressed in sensor-fixed coordinates. It is computed from $\mathbf{z}_i\left(k_1\right)$ and knowledge of the camera's intrinsic parameters. Similarly, the epipolar line at k_2 starts at $\mathbf{p}^n\left(k_2\right)$ and its direction is the unit vector $\overline{\mathbf{u}}_i^n(k_2)$.

The feature point is computed from

$$\mathbf{m}_i^n = \frac{1}{2}\left(\mathbf{p}^n(k_1) + \mathbf{p}^n(k_2) + p_1\overline{\mathbf{u}}_i^n(k_2) + p_2\overline{\mathbf{u}}_i^n(k_2)\right),$$

where

$$p_1 = \frac{\left[\left(\mathbf{p}^n(k_2) - \mathbf{p}^n(k_1)\right) \times \overline{\mathbf{u}}_i^n(k_1)\right] \cdot \left(\overline{\mathbf{u}}_i^n(k_1) \times \overline{\mathbf{u}}_i^n(k_2)\right)}{\mid \overline{\mathbf{u}}_i^n(k_1) \times \overline{\mathbf{u}}_i^n(k_2) \mid^2},$$

$$p_2 = \frac{\left[\left(\mathbf{p}^n(k_1) - \mathbf{p}^n(k_2)\right) \times \overline{\mathbf{u}}_i^n(k_1)\right] \cdot \left(\overline{\mathbf{u}}_i^n(k_2) \times \overline{\mathbf{u}}_i^n(k_1)\right)}{\mid \overline{\mathbf{u}}_i^n(k_2) \times \overline{\mathbf{u}}_i^n(k_1) \mid^2}.$$

After a feature has been initialised, all the remaining observations $\mathbf{z}_i\left(k\right)$, $k_1 < k < k_2$ are incorporated through a single, large update using conventional update equations.

The crucial property of this algorithm is that a feature is not defined in the state vector (and so cannot be used) until a sufficiently large baseline angle α has been observed[1]. Bryson suggests using the value $\alpha = 40°$, but offers no critical analysis of this choice [6]. Although such a large angle is readily achieved by a high-speed UAV flying close to a feature, it is unlikely that such a baseline will be quickly achieved for distant targets observed by low-speed UAVs.

This limitation has spurred authors to develop *undelayed* initialisation algorithms that can exploit a feature as soon as it is first observed. The best known of these is the Inverse Depth Point (IDP) algorithm [18].

3.2 Inverse Depth Point (IDP)

The intuition behind the IDP algorithm is that a poorly-localised landmark can be maintained and updated in a Kalman filter, but *not* using a Cartesian representation. Rather, the feature position is represented as [18]

$$\mathbf{m}_i^n = \mathbf{p}_i^n + \overline{\mathbf{u}}_i^n/\rho_i. \tag{2}$$

In other words, the feature is encoded in terms of the epipolar line which it lies on (starting at $\mathbf{p}_i^n$ and with direction $\overline{\mathbf{u}}_i^n$) together with the *inverse* of the distance from the start of the line to the point of the feature (ρ_i). The reason for this formulation is that once a feature has been initialised, the unit vector of the observation (computed from $\mathbf{z}_i\left(k\right)$) can be computed from

$$\overline{\mathbf{u}}_i^s(k) = \mathbf{K}\mathbf{C}_b^s\mathbf{C}_n^b(k)\left[\overline{\mathbf{u}}_i^n - \rho_i\left(\mathbf{p}^n\left(k\right) - \mathbf{p}_i^n\right)\right].$$

[1] In other words, $|\overline{\mathbf{u}}_i^n(k_1) \cdot \overline{\mathbf{u}}_i^n(k_2)| \geq \cos\alpha$.

where $\mathbf{K}$ is the intrinsic camera calibration matrix. In other words, the effect of the parameterisation is to turn the large uncertain range estimate into a multiplicative uncertainty, which is is better-behaved than a large divisible uncertainty.

The IDP formulation actually represents $\bar{\mathbf{u}}_i^n$ in terms of its azimuth and elevation angles φ_i and υ_i respectively. Therefore, in summary, the poorly-localised state representation of the feature is

$$\hat{\mathbf{m}}_i^n\,(k \mid j) = \begin{bmatrix} \hat{\mathbf{p}}_i^n\,(k \mid j) \\ \hat{\varphi}_i(k|j) \\ \hat{\upsilon}_i(k|j) \\ \hat{\rho}_i(k|j) \end{bmatrix}.$$

The initial value of ρ_i is derived as follows. It is assumed that the distance, $d \geq d_{min}$. Therefore, ρ_i is chosen to have the mean and covariance

An estimate is determined to be well-localised according to a linearity index. If the index falls below a threshold, the feature is deemed to be well-localised, and the inverse depth parameterisation is converted into Cartesian coordinates using (2).

Various improvements to this algorithm include using a common ray origin for multiple features defined within a single frame. One issue is that such bearings-only algorithms have no sense of scale. This can be addressed through the use of inertial systems; in our case GPS works well. A more serious problem is that the algorithm can exhibit failure in which $\hat{\rho}_i$ can become negative [9, 19]. Although various strategies can be used to circumvent these effects [9], they are symptomatic of the problem that the parameterisation can become unstable especially when the feature is far from the platform. In consequence of this limitation, Solà recently proposed an alternative parameterisation, the Anchored Homogeneous Point (AHP) [8].

3.3 Anchored Homogeneous Point (AHP)

Solà argued that the formulation of the IDP in terms of azimuth and elevation angles introduces nonlinear transformations into the filter which can contribute towards the exhibited nonlinearities [8]. Therefore, he proposes to represent the poorly-localised map state by

$$\mathbf{m}_i^n = \mathbf{p}_i^n + \mathbf{u}_i^n / \rho_i. \tag{3}$$

The crucial difference from (2) is that the vector which encodes the ray is *not* constrained to be a unit vector.

Therefore, using this representation, a poorly-localised feature is encoded with the seven parameter state,

$$\hat{\mathbf{m}}_i^n\,(k \mid j) = \begin{bmatrix} \hat{\mathbf{p}}_i^n\,(k \mid j) \\ \hat{\mathbf{u}}_i^n\,(k \mid j) \\ \hat{\rho}_i(k|j) \end{bmatrix}. \tag{4}$$

As with IDP, this representation is converted into Cartesian coordinates if the value of a linearity index falls below a critical value. In this case, the conversion from Polar to Cartesian coordinates is carried out using (3).

Solà shows that the AHP provides significantly better performance than IDP for a scenario in which a ground vehicle drives between various features. This situation is very different from ours — in which a UAV flies over features and sees them from a considerable distance.

4 Simulator Setup

To assess performance of our SLAM systems, we use simulations, which is most widely used in literature and provides with ground truth and also because objects on the ground observed from platform would seem like point objects, and simulation's abstraction of image processing level would not affect our results.

4.1 Platform Description

The platform we use is the Ascending Technologies Hummingbird quadrotor helicopter (illustrated in Fig. 1) [3]. This platform can carry up to 200g payload that can consist of additional sensing devices or processing boards. It is equipped with a GPS, an IMU and a camera. The IMU measurements are thermal bias compensated. The path of the UAV is specified as a series of waypoints. Without vision-based aiding, position can be measured with an accuracy of 1.5–2.5m.

4.2 Platform Dynamics Model

A quadrotor is a nonlinear 3D system governed by complex dynamic and aerodynamic effects. The ground truth process model is a simplified version of the formulation obtained in [20][2]. The model can be written as

$$\begin{bmatrix} \dot{x}^n \\ \dot{y}^n \\ \dot{z}^n \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} u^b \\ v^b \\ w^b \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi}^n \\ \dot{\theta}^n \\ \dot{\psi}^n \end{bmatrix} = \mathbf{E}_b^n \begin{bmatrix} p^b \\ q^b \\ r^b \end{bmatrix}$$

where

$$\begin{aligned} \dot{u}^b &= -q^b w^b + r^b v^b - g \sin\theta^n + K_{uv} u^b \\ \dot{v}^b &= -r^b u^b + p^b w^b + g \sin\phi^n \cos\theta^n + K_{uv} v^b \\ \dot{w}^b &= -p^b v^b + q^b u^b + K_{th} U_{th} + K_w w^b \\ \dot{p}^b &= K_{rl} U_{rl} + K_{pq} p^b \\ \dot{q}^b &= K_{pt} U_{pt} + K_{pq} q^b \\ \dot{r}^b &= K_{ya} U_{ya} + K_r r^b - U_{ya} (r^b)^2. \end{aligned}$$

U_{th}, U_{rl}, U_{pt} and U_{ya} are the thrust, roll, pitch and yaw control inputs, g is the acceleration due to gravity, $\mathbf{C}_b^n$ is the transformation matrix from the body frame to the navigation frame and $\mathbf{E}_b^n$ is the body to navigation frame rotation rate transformation matrix. The coefficients K_{uv}, K_w, K_{pq} and K_r are the parameters that define the dynamic response of our quadrotor and are defined in Table 1.

[2] We neglected the secondary terms and adjusting some of the parameters to make the model symmetric in the pitch and roll axis.

Symbol	Meaning	Value
K_{uv}	Horizontal damping constant	-0.6
K_w	Vertical damping constant	-1.0
K_{pq}	Rotational damping constant	-4
K_r	Yaw damping constant	-1.9

Table 1: Coefficients used in the quadrotor model.

Sensors	Parameters	Values	Rate	Delay
Gyro	$\sigma_{p^b}, \sigma_{q^b}, \sigma_{r^b}$	0.1°	30Hz	0.005s
Accelerometer	$(\sigma_x, \sigma_y, \sigma_z)$	$0.3m$	30Hz	0.005s
GPS	$(\sigma_x, \sigma_y, \sigma_z)$	$0.5m$	10Hz	0.01s
Camera	(σ_u, σ_v)	1pix	5Hz	0.001s

Table 2: Summary of the sensor properties.

The discrete time model is computed by integrating the equations with a second order Runge-Kutta integration scheme. The length of each time step is 0.01s.

The IMU measurements are $\mathbf{f}^b(k)$, and $\boldsymbol{\omega}^b(k)$. The GPS measurements are given by corrupting the platform position $[x^n\ y^n\ z^n]$. In both cases, the measurements are corrupted by zero-mean, Gaussian-distributed random variables.

A summary of the sensors and their settings are given in Table 2. These show the measurement rates, standard deviations and delays (average latencies) in the readings.

4.3 Filter Process Model

The filter does not use the quadrotor dynamics model. Instead, it uses a feedforward filter in which the IMU measurements are treated as control inputs [6]:

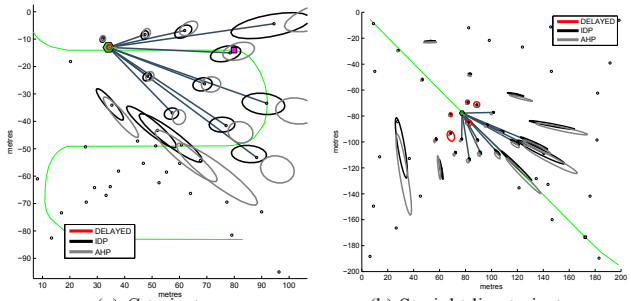
$$\begin{bmatrix} \mathbf{p}^n(k) \\ \mathbf{v}^n(k) \\ \boldsymbol{\psi}^n(k) \end{bmatrix} = \begin{bmatrix} \mathbf{p}^n(k-1)+\mathbf{v}^n(k)\Delta t_k \\ \mathbf{v}^n(k-1)+\left\{\mathbf{C}_b^n(k-1)[\mathbf{f}^b(k)+\mathbf{v}_v^n(k)+\mathbf{g}^n\right\}\Delta t_k \\ \boldsymbol{\psi}^n(k-1)+[\mathbf{E}_b^n(k-1)[\boldsymbol{\omega}^b(k)+\mathbf{v}_\psi^n(k)]\Delta t_k \end{bmatrix}$$

4.4 Trajectories

In bearings-only tracking, the platform trajectory is critical to ensure observability and to obtain an accurate target localisation. An optimal course is to proceed at a fixed deviated angle, hence an optimal trajectory is a deviated pursuit curve [12]. To formalise this, Fawcett has proposed two simple rules [11]:

1. The platform has to move towards the target
2. The platform has to manoeuvre in order to maximise bearing rate (parallax) during the tracking and/or change of bearing rate between manoeuvres

Keeping above points in mind, Passerieux suggests that an optimal trajectory for a platform is composed of two or three "legs" with approximately equal lengths [13]. However undelayed initialisation claims it has the advantage to initialise objects well even in the absence of large parallax or objects along the optic axis further away, which would normally take long time for delayed initialisation.



(a) S-trajectory (b) Straight line trajectory

Figure 3: The trajectories (curves) used in both scenarios together with partially estimated maps from the different implementations. The actual landmark positions are represented by the small dots, and the estimated landmark positions for the different implementations are shown using their 3σ covariance ellipse.

4.5 Scenario

To analyse the performance of the different initialisation methods for a range of conditions, the simulations are based on two different scenarios illustrated in Figure 3. The first is a square region whose sides are of length 200m and is populated with 50 landmarks. The trajectory is a straight line. This will provide the least parallax [18] especially if the target is located far along the line of the trajectory and is thus considered a "worst case". The second scenario consists of a square region whose sides are of length 100m and is populated with 40 landmarks. The trajectory is an S shape trajectory to provide high parallax [13]. Both trajectories are around 300 meters long.

Our simulations were designed to cover large trajectories of around 300 meters, which is around 1600 frames for S shaped trajectory and 1550 frames for straight line trajectory.

The landmarks are configured so that the environment has areas consisting of sparse, populated and no landmarks to simulate real world scenario. Neither scenario includes loop closing because we do not anticipate that the quadrotor will revisit part of the map. The quadrotor itself mostly flies at a fixed altitude of 20m. We investigated two choices of the nadir angle (which affect $\mathbf{C}_b^s$): whether the camera points straight down at the ground, or whether it is oriented at 45° to the vertical, pointing in the direction of travel. The latter makes it possible for the quadrotor to observe a larger part of the environment in any given frame, but trades off resolution to do so.

4.6 Comparison Criteria

The following criteria were used:

- **Estimation consistency.** Though NEES used a lot in literature provides some evidence that estimates are covariance consistent, it provides no evidence of the actual consistency of features on 3D. Since we are mainly concerned about feature estimation consistency, we use

Threshold	Baseline	Performance
0°	40°	Best but tremendously expensive
11.4°	40°	Best
22.9°	40°	Very similar to 11.4° threshold
28.6°	40°	Faster, reduced landmarks initialised
17.1°	34.5°	Well,conservative observation NEES

Table 3: Effect of threshold and baseline angles on the delayed initialisation method based on simulation performed for S shape trajectory with the camera pointing down.

Optimal Subpattern Assignment (OSPA) metric [21], which is a well established metric for map consistency,

$$\overline{d}_p^{(c)}(X,Y)=\left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m} d^{(c)}\left(x_i,y_{\pi(i)}\right)^p+c^p(n-m)\right)\right)^{1/p}$$

where $d^{(c)}(x,y) := \min(c,(x,y))$ is the distance between $x \in X = \{x_i, x_2, \cdots, x_m\}, y \in Y = \{y_1, y_2, \cdots, y_n\} \in W$ (bounded observation window) cut off at $c > 0$ and Π_k is the set permutations on $\{1, 2, \cdots, k\}$ for any $k \in N$. In our case X is the set of all well-localised landmarks and Y is the set of *all* landmarks detected by the platform so far. We used $p = 2$ and $c = 10m$ because these provide a good compromise between localisation and cardinality errors in this application.

- **Total number of features initialised.** This is the number of features which reach the well-localised state and are thus "well-behaved".
- **Time to initialise.** This is average number of frames required until a feature becomes well-localised.
- **The baseline between the initial and last observation of landmarks.** This is the average change in angle required to estimate the landmark's location.
- **Computational complexity.** As complexity of SLAM is $O\left(n^3\right)$ we have stored the size of the state at each update cycle for each filter.

5 Results and Discussion

Table 4 summarises the performance results for the different algorithms and Figures 4 and 5 illustrate the plots for OSPA distance metrics and maximum Eigenvalue respectively for each filter. All results were computed for 20 Monte Carlo runs. Since we found that, qualitatively, the performance and issues for DI are different from those of IDP and AHP, we discuss DI and the IDP/AHP algorithms separately.

5.1 Delayed Initialisation Algorithm

DI algorithm gave very consistent performance across all the scenarios and sensor values tried. It produced consistent estimates as it waits for a good baseline to triangulate location of landmarks illustrated in Figure 5 , hence with least localisation error and did not exhibit any sign of catastrophic failures. This behaviour could also be observed in Figure 4, where as oppose the IDP and AHP, DI does not have peaks caused by high uncertainty in landmarks at initialisation instants. However, we observed several issues with the algorithm, which are all caused by the large baseline which is required. As shown in Table 4, the large baseline means that the DI algorithm is much slower at initialising landmarks than the other methods. In the worst case, it can take an average of more than 770 frames, which is over 150s of simulation time. This has obviously implications in terms of both the computational and storage costs of the filter. Furthermore, many landmarks are not initialised due to the trajectories chosen. In the worst case (straight line, camera at 45°) less than two thirds of the landmarks are actually initialised as illustrated in Figure 4.

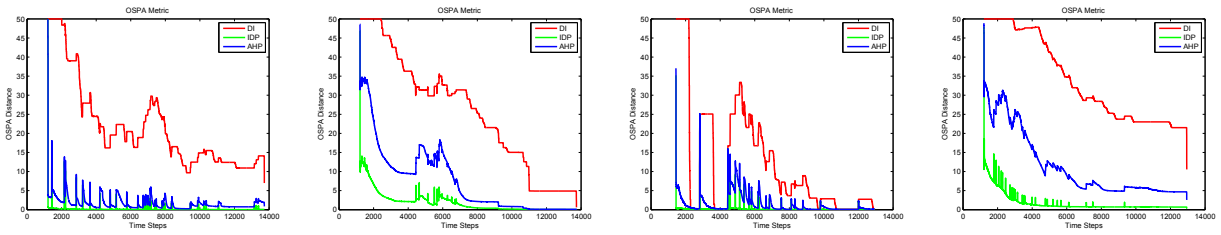
To consider these effects, we investigated two changes to the algorithm. First, we reduced the size of the baseline before a feature is declared well-localised. Second, we modified the augmentation logic so that the state is augmented and the observation is stored only if there is some minimal baseline between successive observations. The results are given in Table 3. The affect of minimal baseline was observed when we run the system using a minimal baseline of 11.4° and 0°, with baseline 40°, giving average sate size of around 300 and 2000 respectively, which obviously deeply affects the performance, the later making the filter really slow.

However it was proven that DI is able to both initialise landmarks and stay stable even in the absence of immediate utilisation of bearing information. In all scenarios DI was able to produce good results.

5.2 IDP and AHP

In all the scenarios, both IDP and AHP algorithms suffered from periodic negative depth problems, indicating that the filters had failed. We observed that this predominantly happened in two cases. The first case arose when the UAV observed distant landmarks. Because of the camera angle and the relatively low speed of the UAV, there is little parallax and observation noise terms dominated. The second case arose when the platform observed a landmark with high uncertainty from a number of very different angles. These difficulties were exacerbated when the angle to the nadir was 45°. To overcome these difficulties, we found it was necessary to use truncated second order filters. Furthermore, we had to modify the parameterisation so that the state includes the *log* of the inverse of the depth [9]. However, we did not observe significant improvements when iterated Kalman filters (using either linearisation or second order terms) were used.

With these changes, we found that the IDP gave significantly better performance and, indeed, its performance is better than the DI algorithm. As Table 4 shows, all the landmarks were intialised, and the number of frames required to get well localised landmarks were at least an order of magnitude less than that required for DI. However, we found that the AHP performed extremely poor in almost all circumstances with high OSPA distances shown in Figure 4 where



(a) S trajectory, nadir=0°. (b) S trajectory, nadir=45°. (c) Straight trajectory, nadir=0°. (d) Straight trajectory, nadir=45°.

Figure 4: Time histories of OSPA Metric. As could be observed the DI OSPA distance is mainly due to cardinality, as it holds many observations of landmarks before they are initialised, hence the sharp drops of the plot at initialisation instants. But the OSPA distance for IDP and AHP are due to localisation error, as both of the methods hold only one observation of the landmarks, that are initialised immediately, and the peaks and distances are as a result of high uncertainty in observed landmarks, which reduces with re-observations, hence the curves

peaks are as a result of very uncertain landmark estimates illustrated in Figure 5[3]. Since it is possible that this could be caused by an inaccurate initial depth estimate, we modified the initialisation of depth to exploit the fact that the landmark features lie close to the ground. However, we found that even with these accurate depth estimates, the qualitative performance did not change significantly. We found that if we further limited the maximum range that a landmark could lie at would be 60m, then consistent behaviour could be obtained. However, this is at a cost of greatly reducing the utility of the algorithm, and suggests that AHP is extremely poor at dealing with distant targets and little motion parallax.

6 Summary and Conclusion

In this paper we have considered the problem of estimating the position of interest points on the ground based on observations taken from a UAV. Our studies have shown that the DI and IDP algorithms can both be used with suitable implementation strategies. However, the AHP algorithm can only be used when the maximum distance to a landmark is strongly curtailed. Furthermore, using an angle to the nadir of 45° is a more demanding situation for the algorithms. The results also show that although the shaped of the trajectory posed various challenges for the DI and IDP, once these had been overcome the overall change in performance in each algorithm was marginal. Therefore, the results suggest that the IDP algorithm is most suitable.

There are several avenues for potential future work. First, terrain elevation databases exist for almost any part of the world and these could be used to apply constraints on the locations of targets. Second, there have been significant recent developments in multi-frame smoothing algorithms. Although we do not believe that the form of these algorithms might be appropriate here, a crucial property of linearising over multiple time steps could have the potential to significantly improve performance.

[3]The max Eigenvalue for IDP and AHP remain constant towards the end of scenarios (straight trajectory with nadir=45° and S trajectory shaped with nadir=0°) does not change as there are landmarks detected earlier that are not re-observed.

Acknowledgements

The work in this paper was supported under the EPSRC-funded project "SUAAVE: Sensing Unmanned Autonomous Aerial VEhicles" (EP/F064179/1).

References

[1] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting Wilderness Search and Rescue Using a Camera-Equipped Mini UAV: Research Articles," *Journal of Field Robotics*, vol. 25, no. 1–2, pp. 89–110, 2008.

[2] R. Skip and B. C. Stoffel, "SAR Spotlight Forum," *http://www.eri-online.com/*.

[3] "Ascending Technologies Hummingbird." `http://www.asctec.de`. Last accessed 6th April, 2010.

[4] R. Steffen and W. Förstner, "On Visual Real Time Mapping for Unmanned Aerial Vehicles," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVII, (Beijing), pp. 57–62, 2008 2008.

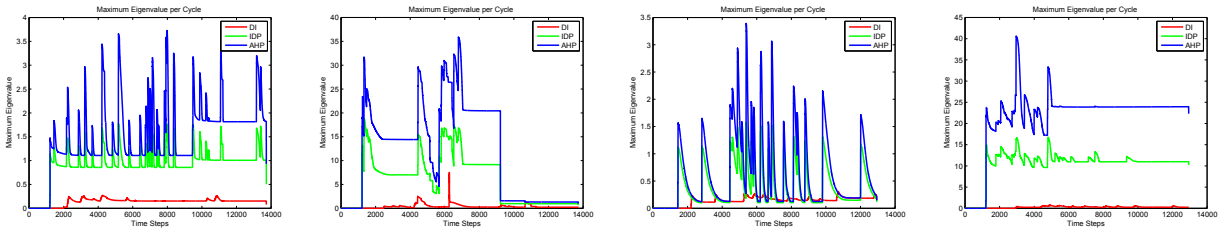
[5] W. Förstner and R. Steffen, "Online Geocoding and Evaluation of Large Scale Imagery Without GPS," *Photogrammetric Week, Heidelberg*, vol. Wichmann Verlag, 2007.

[6] M. T. Bryson and S. Sukkarieh, "Building a Robust Implementation of Bearing-Only Inertial SLAM for a UAV," *Field Robotics*, vol. 24, pp. 113–143, February 2007.

[7] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardós, "Inertial Aiding of Inverse Depth SLAM using a Monocular Camera," in *IEEE International Conference on Robotics and Automiation (ICRA 2007)*, (Rome, Italy), pp. 2797–2802, IEEE, 10–14 April 2007.

Parameters	S shaped with Camera Down			S shaped With Camera at 45°			Straight with camera down			Straight with camera at 45°		
Method	AHP	IDP	Delayed	AHP	IDP	Delayed	AHP	IDP	Delayed	AHP	IDP	Delayed
Cost/update	64	64	215	101	101	338	41	41	76	87	87	433
Landmarks	37	37	27	40	40	34	19	19	18	33	33	20
Frames	33	42	711	41	94	771	25	43	693	45	118	630
Baseline	3°	7.2°	46.5°	11.5°	18.9°	69°	1.18°	1.3°	47.5°	0.45°	0.74°	47°
Performance	OK	V.Good	V.Good	OK	V.Good	V.Good	OK	V.Good	Good	Bad	Good	Good

Table 4: Average MC results (with performance based on state NEES and OSPA metric).



(a) S trajectory, nadir=0°. (b) S trajectory, nadir=45°. (c) Straight trajectory, nadir=0°. (d) Straight trajectory, nadir=45°.

Figure 5: Maximum of landmark Eigenvalues at each cycle.

[8] J. Solà, "Consistency of the Monocular EKF-SLAM Algorithm for Three Different Landmark Parametrizations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2010)*, (Anchorage, AK, USA), 3–8 May 2010.

[9] M. P. Parsley and S. J. Julier, "Avoiding Negative Depth in Inverse Depth Bearing-Only SLAM," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, (Nice, France), pp. 2066–2071, 22–26 September 2008.

[10] S. Tully, H. Moon, G. Kantor, and H. Choset, "Iterated Filters for Bearing-only SLAM," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, (Pasadena, CA, USA), pp. 1442–1448, 19–23 May 2008.

[11] J. A. Fawcett, "Effects of Course Maneuvers on Bearing Only Range Estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1193–1199, August 1998.

[12] P. T. Liu, "An optimum approach in target tracking with bearing measurements," *Journal of Optimization Theory and Applications*, vol. 56, pp. 205–214, February 1988.

[13] J. M. Passerieux and D. van Cappel, "Optimal Observer Maneuver for Bearings-Only Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 777–788, July 1998.

[14] S. Waharte, N. Trigoni, and S. J. Julier, "Coordinated Search with a Swarm of UAVs," in *Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON) Workshops*, (Rome, Italy), pp. 1–3, 22–26 June 2009.

[15] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[16] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez, and M. Olivares, "Visual 3-D SLAM from UAVs," *Journal of Intelligent and Robotic Systems*, vol. 55, pp. 299–321, August 2009.

[17] T. Bailey, "Constrained Initialisation for Bearing-Only SLAM," in *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA'03)*, vol. 2, (Taipei, Taiwan), pp. 1966–1971, 14–19 September 2003.

[18] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, pp. 932–945, October 2008.

[19] N. Sünderhauf, S. Lange, and P. Protzel, "Using the Unscented Kalman Filter in Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control," in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2007)*, (Rome, Italy), pp. 1–6, 27–29 September 2007.

[20] R. De Nardi and O. Holland, "Coevolutionary modelling of a miniature rotorcraft," in *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS10)*, 2008.

[21] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3447–3457, August 2008.