Available online at www.sciencedirect.com
ScienceDirect
Information Fusion 8 (2007) 316–330
INFORMATION FUSION
www.elsevier.com/locate/inffus

Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors

Ross W. Deming *, Leonid I. Perlovsky

Air Force Research Laboratory, SNHE, 80 Scott Drive, Hanscom AFB, MA 01731, USA

Received 2 March 2005; received in revised form 14 November 2005; accepted 16 November 2005
Available online 9 January 2006

Abstract

We are developing a probabilistic technique for performing multiple target detection and localization based on data from a swarm of flying sensors, for example to be mounted on a group of micro-UAVs (unmanned aerial vehicles). Swarms of sensors can facilitate detecting and discriminating low signal-to-clutter targets by allowing correlation between different sensor types and/or different aspect angles. However, for deployment of swarms to be feasible, UAVs must operate more autonomously. The current approach is designed to reduce the load on humans controlling UAVs by providing computerized interpretation of a set of images from multiple sensors. We consider a complex case in which target detection and localization are performed concurrently with sensor fusion, multi-target signature association, and improved UAV navigation. This method yields the bonus feature of estimating precise tracks for UAVs, which may be applicable for automatic collision avoidance. We cast the problem in a probabilistic framework known as modeling field theory (MFT), in which the pdf of the data is composed of a mixture of components, each conditional upon parameters including target positions as well as sensor kinematics. The most likely set of parameters is found by maximizing the log-likelihood function using an iterative approach related to expectation-maximization. In terms of computational complexity, this approach scales linearly with number of targets and sensors, which represents an improvement over most existing methods. Also, since data association is treated probabilistically, this method is not prone to catastrophic failure if data association is incorrect. Results from computer simulations are described which quantitatively show the advantages of increasing the number of sensors in the swarm, both in terms of clutter suppression and more accurate target localization.

Keywords: Sensor fusion; Unmanned aerial vehicles; Mixture model; Dynamic logic; Expectation-maximization; Maximum-likelihood; Concurrent data association and tracking; Photogrammetry

1. Introduction

This paper describes a probabilistic technique for performing multiple target detection and localization based on data from a swarm of flying optical sensors, for example to be mounted on a group of micro-UAVs (unmanned aerial vehicles). In this approach, target detection and mapping are performed concurrently with data association and sensor tracking. This fact gives our method a performance advantage, in principle, over most existing techniques in which detection and data association are performed as separate steps [1–3]. We cast the problem in a probabilistic framework known as modeling field theory (MFT), in which the pdf of the data is composed of a mixture of components, each conditional upon parameters including target positions as well as sensor kinematics [1,4–6]. The most likely set of parameters is found by maximizing the log-likelihood function using an iterative approach related to expectation-maximization. In terms of computational complexity, this approach scales linearly with number of targets and sensors, which represents a significant improvement over most existing methods. Also,

* Corresponding author. Tel.: +1 781 377 1736; fax: +1 781 377 8984.
E-mail address: ross.deming@hanscom.af.mil (R.W. Deming).

1566-2535/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2005.11.001

since data association is treated probabilistically, this method is not prone to catastrophic failure if data association is incorrect.

The problem falls under the broader area of "swarm intelligence", which is currently an active area of research. In military surveillance applications, progress in swarm intelligence is expected to revolutionize the ways in which unmanned aerial vehicles (UAVs) are used. The value and potential of UAVs have been demonstrated in recent military conflicts, where they have been used for dangerous and/or tedious missions to reduce the risk of human casualties. It is felt that UAV cooperation and swarming behavior may yield advantages that will make UAVs, in general, even more valuable [7–9]. The most obvious advantage would be an increase in mission success rates due to improved UAV survivability—hostile defenses would be taxed by the sheer numbers in the swarm. Also, swarms might be deployed in smart ways to increase the efficiency of the geographical coverage. Finally, having access to swarms of sensors may make it easier to detect and discriminate low signal-to-clutter (S/C) targets by exploiting correlations between different, complementary, sensor types and/or different aspect angles. In a sense, the collection of small sensors on individual UAVs would be equivalent to a wide aperture, which can be exploited to yield much better location and velocity estimates for targets, as well as better detection and discrimination performance.

In order to make deployment of UAV swarms feasible, it will be necessary for UAVs to operate more autonomously than is currently possible [7,9]. Presently UAVs operate more or less like "binoculars with wings" with human operators performing most duties, including low-level functions like image analysis/interpretation and obstacle/collision avoidance. Human operators (and data links from UAVs to operators) would become quickly overwhelmed attempting to control an entire swarm of UAVs. The approach discussed in this paper is designed to reduce the load on human operators by providing computerized interpretation of images from multiple sensors. A by-product of the approach is a set of precise tracks for both targets and UAVs that may be applicable to automatic collision avoidance.

One might think it unnecessary to compute UAV positions, since these can be measured directly using onboard inertial devices and global-positioning systems (GPS). However, the accuracy of GPS and inertial measurements may be too rough to allow a particular target's image (signature) in one frame to be reliably associated with its corresponding image in another frame (we discuss this "data association problem" below in more detail), especially if there are many closely spaced targets. For example, the typical accuracy of GPS is on the order of ± 10 m [10]. Also, while inertial devices and GPS measure absolute position, they do not measure position relative to potential obstacles. The algorithm described here provides a framework for fine-tuning information from a GPS using outputs from visual (or other) sensors. Thus, in this problem the term "sensor fusion" not only describes combining information from multiple visual sensors, but it also describes combining outputs from visual sensors with outputs from GPS sensors. For optimum performance, all functions need to be performed concurrently [2,3]: signature association requires accurate UAV tracking, while accurate localization of targets and UAVs requires signature association.

The algorithm works as follows. Consider the case in which multiple UAVs fly over a group of targets, acquiring digitized images (image frames) at multiple times. Within these images, we refer to the group of pixels associated with each target as a "signature". For each signature in each image frame the data set consists of (a) a vector of classification features computed from the signature, (b) the position of the signature on the image focal plane, (c) the time of acquisition for the corresponding image frame, and (d) (optional) the output of onboard GPS or inertial devices. From the aggregate data, we wish to identify targets of interest, and compute precise tracks for both UAVs and objects in the scene. This task is accomplished within the framework of *modeling field theory* (MFT) [1], as follows. First, a model of the data is developed, where model parameters include locations and features of targets, and coefficients of UAV equations of motion. By incorporating sensor errors, a statistical model is obtained. The parameters are then estimated by maximizing the log-likelihood function, which gives a quantitative measure of how well the model fits the set of measured data.[1] MFT maximizes the likelihood in the space of all parameters and all possible mappings between targets and data, to associate signatures with targets and iteratively solve for the parameters in an efficient manner. Thus, we are not plagued with a combinatorial search during data association like other optimum techniques such as *multiple hypothesis tracking* (MHT). In Sections 2 and 5 we further compare our technique with existing approaches.

Modeling field theory (MFT) is a general approach used to combine both physical and statistical models [1]. Whereas statistical analysis is a standard tool for analyzing a single physical process, standard techniques are not appropriate when multiple, competing, physical processes are involved, including statistical uncertainty and unknown physical parameters. However, MFT is explicitly designed for these types of problems. Historically, MFT describes biological systems in which neuronal fields are determined by physical and statistical models [1]. In practice, the convergence of MFT can be proven using expectation-maximization (EM), although prior to MFT EM was never before applied in this manner. Section 2 discusses in more detail how MFT relates to other EM-based approaches.

The paper is organized as follows. Section 2 provides a review of the relevant literature, and a comparison of our

[1] Of course, the log-likelihood will only measure the "goodness of fit" between the model and the data to the extent that the general form of the model is correct.

technique with some related existing ideas. The geometry of the problem is described in Section 3, as well as the organization of the data and the physical model that relates three-dimensional coordinates of objects in the environment to their corresponding two-dimensional image positions in the focal plane of the camera. In Section 4 an approach is developed for performing target detection concurrently with data association and the estimation of parameters related to target localization and UAV tracks. Some practical issues are discussed in Section 5, including computational complexity and communication between the multiple platforms. In Section 6 we present results from computer simulations. The results clearly show that the algorithm was able to automatically identify trackable objects (''targets'') in the presence of clutter, and that sensor fusion resulted in tracking and localization performance that improved significantly with an increase in the size of the UAV swarm. Also, the results show the effects of clutter level on performance. Finally, in Section 7 we discuss conclusions and directions for further research.

2. Related work

The problem we consider in this paper is quite similar to the *simultaneous localization and mapping* (SLAM) problem widely discussed in the robotics literature [15–33]. In SLAM the idea is to generate a model of the environment via sensor inputs acquired by a mobile robot. As the robot moves through the environment and acquires data, it seeks to develop an increasingly accurate estimate of the locations of surrounding landmarks, and also an estimate its own position in relation to the landmarks.

A popular SLAM approach is based on the extended Kalman filter (EKF) [15–27,29]. Here, the motion of the robot is modeled by a discrete-time state transition equation, in which the state of the system includes the position and orientation of the vehicle and the positions of all landmarks. The EKF then provides a recursive solution for updating the state as new measurements are acquired. Typically, the EKF-based methods have been tested in relatively simple 2D environments in which the robot moves on a plane. Sensors in these experiments have included radar or sonar range finders as well as digital cameras. Assuming known data association, the computational cost of EKF-based SLAM is $O[n^2]$, where n is the number of variables needed to represent the landmark positions and the robot pose [21,27,29,31]. This requirement is due to the fact that a full covariance matrix needs to be maintained which includes the positions of all landmarks. There have been various methods proposed to decrease this cost using approximations to the method [15,18,21,24,32].

In [20,26,27,33] the authors have worked on extending EKF-based SLAM to the multiple robot problem, although from these studies it's not clear how feasible the approach would be for large swarms of robots, especially for clutter-rich data. For multiple robot EKF-based SLAM, the computational complexity would be $O[j^2n^2]$ (assuming known data association), where j is the number of robots. In [38], the authors present an impressive experimental system consisting of four actual UAVs, plus a software algorithm specially designed to make SLAM work on a distributed processing architecture. In their tests, using 20 plastic white sheets as targets, they were able to make the system work in real time, despite using an exhaustive search method for data association. However, as we discuss below, and in Section 5, there are important limitations to exhaustive search techniques that make them impractical for applications containing significant clutter.

Data association is an open problem for EKF-based SLAM, and typical methods are borrowed from the multiple target tracking arena. For example, the gated nearest neighbor (NN) algorithm is a popular choice [19,34,14]. Unfortunately, although NN is relatively efficient—it requires $O[mn]$ computations, where m is the number of sensor measurements and n is the number of map features—it is widely recognized to become unreliable as the levels of clutter and sensor uncertainty increase [25]. This is a crucial problem for EKF-based SLAM methods, since they are brittle, i.e., prone to fail catastrophically, when data association is incorrect [27,31]. More robust (however, more computationally complex) data association methods have been borrowed from multi-target tracking applications, including *multiple hypothesis tracking* (MHT) [22,27,35], in which all possible combinations of tracks and data associations are exhaustively evaluated, and *joint probabilistic data association* (JPDA) [13,28] which is more efficient than MHT because one only needs to evaluate the association probabilities separately at each time step. Other robust methods have also been proposed [25]. It should be noted that methods like JPDA involve track maintenance but not track initialization. Therefore, since detection is performed separately from tracking, these methods are not optimal [3] and cannot initiate tracks at low signal-to-clutter (S/C) ratios. In contrast, MHT is optimal, but impractical since the number of mappings between data and targets will grow exponentially. It should also be noted that in the higher-clutter cases we consider in this paper, both JPDA and MHT would have combinatorial complexity.

An alternative to EKF-based methods is proposed in [30], in which SLAM is posed as a maximum-likelihood estimation problem, and a Hidden Markov Model is used to describe the motion of the robot. Here, data association is handled concurrently with landmark mapping. *Expectation-maximization* (EM) is used to perform data association in an iterative fashion. The authors implement their approach for an experimental 2D problem in which a robot moves through an indoor environment. Here, human intervention is used to help the robot detect and identify landmarks (corners, intersections, etc.). The authors show that the computational complexity of the algorithm scales as $O[n]$, including the data association, which is a significant improvement over the EKF-based approaches. In [31] the same authors show how the EM-based approach

can be extended for teams of robots acting cooperatively. One issue the authors do not address is how the algorithm might perform in high-clutter, three-dimensional, environments.

Multiple target tracking (MTT) is similar to SLAM in many aspects, however in MTT the sensors are assumed to have known, fixed, positions. Most approaches for SLAM borrow various ideas from the MTT arena. For example, the extended Kalman filter (EKF) together with JPDA has been used for years in MTT [13]. More recently, sequential Monte Carlo methods, a.k.a. "particle filters", have appeared as an alternative to the EKF-based methods [36,37]. Particle filters which, like the EKF, are based upon stochastic state equations, generalize the traditional Kalman Filter methods to situations involving non-linear state and measurement equations and non-Gaussian noise. Data association for these methods can be performed in various ways, including Gibbs sampling and expectation-maximization (EM) [36]. A good reference on the relationship between particle filters and EKF can be found in [37]. Particle filters have also been adapted for the SLAM problem [28].

The method we develop in this paper is inspired by—and most similar to—an alternative approach to MTT based upon modeling field theory (MFT), a relatively new idea which we discussed in the introduction. In this approach, detection (i.e., track initiation), data association, and tracking are performed concurrently [1–6] using a system of dynamic logic. Concurrent processing allows the tracker to approach ideal performance [3] and allows tracking in high clutter since no threshold is needed for detection. Here, the probability density of the data samples (including range plus azimuth plus classification features and, if available, Doppler) is modeled as a mixture. Each component of the mixture corresponds to a different target, and model parameters describe the kinematics of the target trajectories. Then, an iterative method related to expectation-maximization (EM) is used to efficiently hill climb in the space of all parameters and all possible mappings between data samples and targets. Our approach is similar to the EM-based SLAM methods [30,31] in the sense that data association is performed concurrently with landmark mapping, and because our approach also scales as O[n] (or O[jn] for j sensor platforms). Our approach is different, however, in the way we set up the likelihood function and incorporate the kinematics of the flying sensors and the target positions. Also, our sensors are not restricted to motion in the plane, and the sensor pose includes yaw, pitch, and roll. Finally, we assume there can be significant clutter and, accordingly, we provide a means to model the clutter distribution concurrently with landmark mapping, data association, etc. Our approach differs from the EKF-based SLAM methods because the track initiation and data association are performed concurrently with target mapping, and in this sense our approach is optimal. Also, it generates probabilistic, rather than deterministic, mappings between data and targets, and therefore is not brittle (i.e., it will not fail catastrophically if the data association is incorrect). In Section 5 we quantify the computational complexity of our approach relative to an existing optimal approach.

The data model used in this paper is a special type of Gaussian mixture model (GMM), where the unknown parameters determine the shape and structure of the components. Parameter estimation for GMMs is a fairly well-studied problem, and EM is a widely used method [1,11]. GMMs have been used for applications such as tracking objects in a room based upon color [12], which are similar to the application discussed in this paper. Our approach is different from typical GMM-based tracking approaches primarily because of the way that the dynamics of motion are incorporated. Here, the models are more complicated, having a shape and form that depends explicitly on the physical dynamics of motion. Thus, the object tracks are estimated during the EM iterations concurrently with the other model parameters. This structures allows detection, data association, and tracking to be performed concurrently, theoretically resulting in ideal performance [1,3]. In contrast, in typical GMM-based approaches the dynamics of motion are separate from the form of the model, appearing as links between model components that are computed apart from the EM iterations. Thus detection, association, and tracking are performed separately.

Finally, our problem is also related to the field known as *photogrammetry* [23,40–42], which is quite similar to SLAM, but is traditionally concerned with aerial photography. The problem here is to estimate the three-dimensional locations of terrain and objects of interest based upon their positions in the photographic planes in multiple photographs, each taken from a different vantage point. Traditionally, the image interpretation and feature association have been performed manually, by a human. However, there is currently much research being done on automatic feature extraction and data association [41,42], as well as ways to adapt SLAM techniques to photogrammetry [23]. As with SLAM, reliable and efficient automatic data association remains an open problem.

3. Description of geometry and data

The details of the problem are the following. Multiple UAVs located at the coordinates $\mathbf{X}_j = (X_j, Y_j, Z_j)$, $j = 1, 2, \ldots, J$, are flying over a group of objects ("targets") located at coordinates $\mathbf{x}_k = (x_k, y_k, z_k)$, $k = 1, 2, \ldots, K$, where z denotes the elevation and (x, y) denotes the horizontal position (throughout the discussion, vector quantities are indicated in bold type). Note that the term "target" is used loosely, referring both to potential threats and simply to landmarks and geographical features to be tracked for the purposes of navigation and registration between multiple images. Each UAV is equipped with an optical sensor (a digital camera which records, for example, a matrix of visible or infrared information) and, optionally, a GPS and/or inertial navigation instrument. The GPS measures the UAV position directly, although with significant random error. We denote the coordinate data output

by the GPS as $\hat{\mathbf{X}}_j = (\hat{X}_j, \hat{Y}_j, \hat{Z}_j)$. Fig. 1 shows a diagram of one of the UAVs flying over the group of targets.

Generally, features (''targets'') in an image can be distributed objects, for example vehicles, trees, buildings, or blobs of infrared energy. The types of landmarks extracted from extended objects for machine vision applications may include, for example, edges, corners, texture boundaries, or centers of mass [16,23]. However, the manner in which landmarks are extracted from images and assigned coordinates is beyond the scope of this paper and, in fact, is an active area of research in its own right. Here we set up the problem as if the targets were point reflectors, a simplification that is commonly made for SLAM approaches (e.g., [17]). In the future, we plan to address how the approach will handle more realistic targets.

The sensors on the UAVs record replicas of three-dimensional scenes onto two-dimensional images, for example mapping an object located at $\mathbf{x}_k = (x_k, y_k, z_k)$ to a (horizontal + vertical) position (α, β) on the camera's focal plane. Because the mapping goes from 3D to 2D, we cannot reverse the mapping to compute a target position uniquely from a single image, even if we know the UAV position. However, from multiple views of the same target it would be possible to triangulate the position, and this illustrates an advantage of having a swarm of sensors. In fact, the problem of localizing objects in 3D based on their image locations in a set of spatially separated photographs is well studied, and is discussed in detail in standard treatments of ''photogrammetry'', including [40,41]. Of course, when performing the process automatically, the difficulty lies in enabling a computer to associate a target signature from one digital photograph with its counterparts in the other photos acquired by other UAVs. This problem is especially acute when the photos contain many targets, some partially obstructed, and significant clutter. This ''association problem'' is addressed using MFT within a probabilistic framework, as we will discuss.

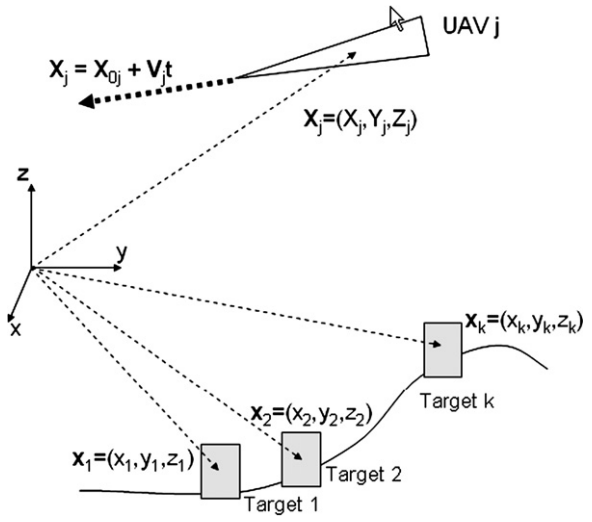

Fig. 1. Geometry, including a single UAV flying over a group of ''targets''.

Following the discussion in [40, Sections 2.3.1 and 2.3.2], the mapping from the 3D world coordinate $\mathbf{x}_k = (x_k, y_k, z_k)$ to the 2D focal plane coordinate (α, β) of a camera located at $\mathbf{X}_j = (X_j, Y_j, Z_j)$ is given by the well-known pair of photogrammatic equations (the ''sensor model'')

$$\alpha = d_f \frac{(x_k - X_j)m_{11} + (y_k - Y_j)m_{12} + (z_k - Z_j)m_{13}}{(x_k - X_j)m_{31} + (y_k - Y_j)m_{32} + (z_k - Z_j)m_{33}} \tag{1}$$

and

$$\beta = d_f \frac{(x_k - X_j)m_{21} + (y_k - Y_j)m_{22} + (z_k - Z_j)m_{23}}{(x_k - X_j)m_{31} + (y_k - Y_j)m_{32} + (z_k - Z_j)m_{33}}, \tag{2}$$

where d_f is the camera focal distance, and the quantities m_{rs} are the elements of the 3×3 direction cosine matrix $\mathbf{M}$ relating the global coordinate frame to the coordinate frame local to the camera on the UAV. Explicitly, the direction cosine elements are given as (e.g., see Section 2.3.1 in [40])

$$\begin{aligned}
m_{11} &= \cos\phi\cos\kappa,\\
m_{12} &= \cos\omega\sin\kappa + \sin\omega\sin\phi\cos\kappa,\\
m_{13} &= \sin\omega\sin\kappa - \cos\omega\sin\phi\cos\kappa,\\
m_{21} &= -\cos\phi\sin\kappa,\\
m_{22} &= \cos\omega\cos\kappa - \sin\omega\sin\phi\sin\kappa,\\
m_{23} &= \sin\omega\cos\kappa + \cos\omega\sin\phi\sin\kappa,\\
m_{31} &= \sin\phi,\\
m_{32} &= -\sin\omega\cos\phi,\\
m_{33} &= \cos\omega\cos\phi,
\end{aligned}$$

where (ω, ϕ, κ) are the rotational angles (yaw, pitch, and roll) for the coordinate frame of the camera. For simplicity, we will assume these angles can be measured precisely using onboard sensors,[2] although the method in this paper can be extended in a straightforward manner to include estimation of (ω, ϕ, κ) along with the other parameters. If we define the vectors $\mathbf{M}_i \equiv [\, m_{i1} \quad m_{i2} \quad m_{i3} \,]$, we can rewrite Eqs. (1) and (2) using the compact notation

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = d_f \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix} \frac{(\mathbf{x}_k - \mathbf{X}_j)^{\mathrm{T}}}{\mathbf{M}_3(\mathbf{x}_k - \mathbf{X}_j)^{\mathrm{T}}}, \tag{3}$$

where T denotes the vector transpose. Note that $\mathbf{M}_i$, $\mathbf{x}_k$, and $\mathbf{X}_j$ are all row vectors, and therefore the vector product $\mathbf{M}_i(\mathbf{x}_k - \mathbf{X}_j)^{\mathrm{T}}$ is a scalar quantity.

As the jth UAV flies, it captures image frames at intervals along its path, and we wish to combine the information from these frames and from the sets of frames from the other UAVs. Models for the UAV flight trajectories will facilitate this task. Let us assume that UAV j flies at a constant velocity $\mathbf{V}_j = (\dot{X}_j, \dot{Y}_j, \dot{Z}_j)$ so that its equation of motion is

$$\mathbf{X}_j(t) = \mathbf{X}_{0j} + \mathbf{V}_j t, \tag{4}$$

[2] Small and inexpensive tilt sensors are available having an angular resolution of as little as 0.03°.

where $\mathbf{X}_{0j}$ is its time-zero position and t denotes time. If necessary, our approach will allow more complicated UAV trajectory models to be incorporated in a straightforward manner. However, although simple, the constant-velocity model is useful in many instances (note, for example, that arbitrary tracks can be approximated as being piecewise-linear). Using the constant-velocity model, we can rewrite Eq. (3) as

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = d_{\mathrm{f}}\begin{bmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{bmatrix}\frac{(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t)^{\mathrm{T}}}{\mathbf{M}_3(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t)^{\mathrm{T}}}. \tag{5}$$

The position (α, β) of a target signature in the image is only one piece of the data collected by the cameras. The other piece is the target signature itself, i.e., the array of pixel intensities (red, blue, and green) in the vicinity of the target's image on the focal plane. Most automatic target recognition algorithms make use of a preprocessing step in which a manageable set of classification features are computed from the signature [16,23,43,44]. These features are specially designed to allow signatures from threats to be automatically separated from signatures of clutter objects. In our case, the features will also help in the association problem, as we will discuss. How the features are computed is not the subject of this paper—we simply assume a set of features $\mathbf{f} = (f_{(1)} \quad f_{(2)} \quad \cdots \quad f_{(d)})$ has been computed at multiple locations within each image frame. It should be noted that more recent tracking algorithms often do not treat feature extraction entirely as a preprocessing step but rather incorporate this step in a more model-driven manner.

The data from each target signature include the set of classification features $\mathbf{f}$ plus the signature location (α, β) on the focal plane. Thus, the information from an image frame is reduced to a set of data samples $(\alpha_{jn}, \beta_{jn}, \mathbf{f}_{jn})$, where $n = 1, 2, \ldots, N$ is the index of the sample,[3] and $j = 1, 2, \ldots, J$ denotes which UAV acquired the image. Each of these samples was produced by a particular object (target or clutter). Also recorded with each sample is the time t_{jn} at which the corresponding image frame was acquired. In addition to the data from the camera, we have the data $\hat{\mathbf{X}}_{jn}$ from the GPS (to make things simple, we assume a GPS data point is acquired simultaneously with each photo). Therefore, the total set of data is contained in the set of samples $\mathbf{w}_{jn} = (\hat{\mathbf{X}}_{jn}, \alpha_{jn}, \beta_{jn}, \mathbf{f}_{jn})$ and their corresponding times t_{jn}. Since the rotational angles of each UAV change with time, we will henceforth indicate this dependence in the directional cosine vectors using the notation $\mathbf{M}_i^{jn}$.

4. Parameter estimation and data association

In this section we describe how the data can be modeled probabilistically using a Gaussian mixture model, and we describe a method for extracting the model parameters which include target locations and types, and UAV locations and velocities.

As described in the previous section, each of the samples jn is characterized by a set of measurements

$$\mathbf{w}_{jn} = \mathbf{w}_{jn}(t) = (\hat{\mathbf{X}}_{jn}, \alpha_{jn}, \beta_{jn}, \mathbf{f}_{jn}). \tag{6}$$

Each sample was produced by a target numbered $k = 1, 2, \ldots, K$, or by image clutter. Given that the data samples were produced by object k, the expected value of the data is

$$E\{\mathbf{w}_{jn}|k\} = E\{(\hat{\mathbf{X}}_{jn}, \alpha_{jn}, \beta_{jn}, \mathbf{f}_{jn})|k\} = (\bar{\mathbf{X}}_{jn}, \bar{a}_{jnk}, \bar{b}_{jnk}, \bar{\mathbf{F}}_k).$$

The function $\bar{\mathbf{F}}_k$ is simply the mean value for the classification feature vector assuming target class k. We assume feature statistics depend only on target k and not on j (which UAV) or n (which instant in time or space). The functions $\bar{\mathbf{X}}_{jn}$, $\bar{a}_{jnk}$, $\bar{b}_{jnk}$ are intended to incorporate all of the deterministic (i.e., physical and geometrical) information in the data. For example, based upon Eq. (4), the expected value of the GPS data is

$$\bar{\mathbf{X}}_{jn}(\mathbf{X}_{0j}, \mathbf{V}_j) = \mathbf{X}_{0j} + \mathbf{V}_j t_{jn}, \tag{7}$$

which is independent of target k (not related to targets at all). Also, based upon Eq. (5), the expected values of the (horizontal and vertical) signature position data are

$$\begin{bmatrix}\bar{a}_{jnk}(\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j)\\ \bar{b}_{jnk}(\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j)\end{bmatrix} = \begin{bmatrix}\mathbf{M}_1^{jn}\\ \mathbf{M}_2^{jn}\end{bmatrix}\frac{d_{\mathrm{f}}(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t_{jn})^{\mathrm{T}}}{\mathbf{M}_3^{jn}(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t_{jn})^{\mathrm{T}}}. \tag{8}$$

Note that the term in the denominator is the product of the row vector $\mathbf{M}_3^{jn}$ and the column vector $(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t_{jn})^{\mathrm{T}}$, and is therefore a scalar quantity.

Random GPS and sensor errors, as well as uncertainty in classification feature values, are modeled probabilistically. Thus, the probability density of the data from sensor j, sample n, given that it originated from object k, is

$$p(\mathbf{w}_{jn}|k) = p_1(\hat{\mathbf{X}}_{jn}) \cdot p_2(\alpha_{jn}, \beta_{jn}|k) \cdot p_3(\mathbf{f}_{jn}|k).$$

Here, we have broken down the total pdf into the product of pdfs for $\hat{\mathbf{X}}_{jn}$, $(\alpha_{jn}, \beta_{jn})$, and $\mathbf{f}_{jn}$ since, given the kth target, the components of the measurement vector are independent. We use Gaussian pdfs for modeling sensor errors, and therefore the explicit formula for the pdf of the GPS data is

$$p_1(\hat{\mathbf{X}}_{jn}) = \frac{1}{(2\pi)^{3/2}\sigma_{\mathrm{g}}^3}\mathrm{e}^{\left[-\frac{1}{2\sigma_{\mathrm{g}}^2}(\hat{\mathbf{X}}_{jn} - \bar{\mathbf{X}}_{jn})(\hat{\mathbf{X}}_{jn} - \bar{\mathbf{X}}_{jn})^{\mathrm{T}}\right]}, \tag{9}$$

where $\bar{\mathbf{X}}_{jn}$ is given in Eq. (7) and σ_{g} is the GPS error standard deviation. The pdf of the signature position data is

$$p_2(\alpha_{jn}, \beta_{jn}|k) = \frac{1}{2\pi\sigma_{\mathrm{a}}^2}\mathrm{e}^{-\frac{1}{2\sigma_{\mathrm{a}}^2}\left[(\alpha_{jn} - \bar{a}_{jnk})^2 + (\beta_{jn} - \bar{b}_{jnk})^2\right]}, \tag{10}$$

where $\bar{a}_{jnk}$ and $\bar{b}_{jnk}$ are given in Eq. (8) and σ_{a} is the standard deviation of the error in signature position, which includes errors in feature localization as well as errors in

[3] The index n counts over the entire set of (signatures × frames) acquired by a particular UAV.

camera alignment. Finally, the pdf for the classification feature data is

$$p_3(\mathbf{f}_{jn}|k) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{C}_{fk}|}} \mathrm{e}^{\left[-\frac{1}{2}(\mathbf{f}_{jn}-\bar{\mathbf{F}}_k)\mathbf{C}_{fk}^{-1}(\mathbf{f}_{jn}-\bar{\mathbf{F}}_k)^{\mathrm{T}}\right]}, \tag{11}$$

where $\mathbf{C}_{fk}$ is the covariance matrix of the features and d is the number of features.

In addition to the models for the trackable objects, we also need a model component for background clutter. Here we choose a single clutter model, corresponding to $k = 0$, which is uniform across sensors and positions. This model is Gaussian over the features $\mathbf{f}_{jn}$ and uniform over the other data types, and can be written

$$p(\mathbf{w}_{jn}|0) = p_3(\mathbf{f}_{jn}|k=0) = \frac{1}{\sqrt{(2\pi)^d|\mathbf{C}_{f0}|}} \mathrm{e}^{\left[-\frac{1}{2}(\mathbf{f}_{jn}-\bar{\mathbf{F}}_0)\mathbf{C}_{f0}^{-1}(\mathbf{f}_{jn}-\bar{\mathbf{F}}_0)^{\mathrm{T}}\right]}.$$

It should be noted that this simple model for clutter may not be adequate in general, since the background may contain transitions between different types of terrain, etc. However, the approach is flexible enough to include more complicated multi-modal clutter models as discussed, for example, in [1, Section 5.5.3].

The total probability density function for the data from sensor m is the sum over contributions from alternate targets

$$p(\mathbf{w}_{jn}) = \sum_{k=0}^{K} r_k p(\mathbf{w}_{jn}|k), \tag{12}$$

where r_k is the weighting for the kth component. These weights are non-negative, and sum to 1. The above equation expresses the pdf of the data as a Gaussian mixture model. Given this pdf, the association probability between sample jn and target (or clutter) k is

$$P(k|jn) = \frac{r_k p(\mathbf{w}_{jn}|k)}{p(\mathbf{w}_{jn})}. \tag{13}$$

It should be noted that Eq. (13) is simply a form of Bayes' rule, where r_k is the a priori probability for class k. Next, the total log-likelihood ([43, Chapter 3] and [1, Section 7.4]) of the observed data is the sum of log-likelihood for individual sensors (UAVs) and samples, i.e.,

$$\mathrm{LL} = \sum_{j,n} \ln p(\mathbf{w}_{jn}) = \sum_{j,n} \ln\left[\sum_{k=0}^{K} r_k p(\mathbf{w}_{jn}|k)\right]. \tag{14}$$

Estimation of the model parameters amounts to finding the parameter set, as described below, that maximizes the log-likelihood LL. However, not all of the parameters need to be estimated—for example the component weights r_k for $k > 0$ are simply proportional to the number of time frames for each UAV, while r_0 is easily estimated from the clutter. Also, the standard deviations σ_{g} (GPS error) and σ_{a} (error in the alignment of the camera axis) are known a priori. This leaves the remaining set of unknown parameters $\{\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j, \bar{\mathbf{F}}_k, \mathbf{C}_{fk}\}$, which need to be estimated.

Expressions for the parameters $\bar{\mathbf{F}}_k$ and $\mathbf{C}_{fk}$ are found directly, in the usual way (refer to [1, Chapter 5]), by finding the values that satisfy $\partial\mathrm{LL}/\partial\bar{\mathbf{F}}_k = 0$ and $\partial\mathrm{LL}/\partial(\mathbf{C}_{fk}^{-1}) = 0$. Thus, we obtain the expressions

$$\bar{\mathbf{F}}_k = \frac{1}{r_k}\sum_{j,n} P(k|jn)\mathbf{f}_{jn}, \tag{15}$$

$$\mathbf{C}_{fk} = \frac{1}{r_k}\sum_{j,n} P(k|jn)(\mathbf{f}_{jn}-\bar{\mathbf{F}}_k)^{\mathrm{T}}(\mathbf{f}_{jn}-\bar{\mathbf{F}}_k), \tag{16}$$

with $P(k|jn)$ given in Eq. (13). Note that the above equations are similar to the usual expressions for sample mean and covariances (e.g., [43, Chapter 2]), but here each sample is weighted by its probability of belonging to class k, $P(k|jn)$ from Eq. (13).

For the remaining parameters $\{\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j\}$, it is not convenient to solve $\partial\mathrm{LL}/\partial(\cdot) = 0$ directly due to the complicated form of the expected value functions $\bar{a}_{jnk}(\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j)$ and $\bar{b}_{jnk}(\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j)$ given in Eq. (8). Instead the parameters are changed incrementally along the direction of the gradient[4] of LL, as described in [1, Chapter 4]. At the Ith iteration the parameters are computed by adjusting their values from the $(I-1)$th iteration, as follows:

$$\mathbf{x}_k^{(I)} = \mathbf{x}_k^{(I-1)} + s\frac{\partial\mathrm{LL}^{(I-1)}}{\partial\mathbf{x}_k}, \tag{17}$$

$$\mathbf{X}_{0j}^{(I)} = \mathbf{X}_{0j}^{(I-1)} + s\frac{\partial\mathrm{LL}^{(I-1)}}{\partial\mathbf{X}_{0j}}, \tag{18}$$

$$\mathbf{V}_j^{(I)} = \mathbf{V}_j^{(I-1)} + s\frac{\partial\mathrm{LL}^{(I-1)}}{\partial\mathbf{V}_j}, \tag{19}$$

where s is the step size, which can be selected from experience. Note that a very small step size always results in convergence, but may lead to a larger number of iterations than are necessary. Explicitly, the partial derivatives from above are found to be

$$\frac{\partial\mathrm{LL}}{\partial\mathbf{x}_k} = \sum_{j,n} P(k|jn)\left[\left(\frac{\alpha_{jn}-\bar{a}_{jnk}}{\sigma_{\mathrm{a}}^2}\right)\frac{\partial\bar{a}_{jnk}}{\partial\mathbf{x}_k} + \left(\frac{\beta_{jn}-\bar{b}_{jnk}}{\sigma_{\mathrm{a}}^2}\right)\frac{\partial\bar{b}_{jnk}}{\partial\mathbf{x}_k}\right],$$

$$\frac{\partial\mathrm{LL}}{\partial\mathbf{X}_{0j}} = \sum_{n,k} P(k|jn)\left[\left(\frac{\hat{\mathbf{X}}_{jn}-\bar{\mathbf{X}}_{jn}}{\sigma_{\mathrm{g}}^2}\right) + \left(\frac{\alpha_{jn}-\bar{a}_{jnk}}{\sigma_{\mathrm{a}}^2}\right)\frac{\partial\bar{a}_{jnk}}{\partial\mathbf{X}_{0j}} + \left(\frac{\beta_{jn}-\bar{b}_{jnk}}{\sigma_{\mathrm{a}}^2}\right)\frac{\partial\bar{b}_{jnk}}{\partial\mathbf{X}_{0j}}\right],$$

$$\frac{\partial\mathrm{LL}}{\partial\mathbf{V}_j} = \sum_{n,k} P(k|jn)\left[\left(\frac{\hat{\mathbf{X}}_{jn}-\bar{\mathbf{X}}_{jn}}{\sigma_{\mathrm{g}}^2}\right)t_{jn} + \left(\frac{\alpha_{jn}-\bar{a}_{jnk}}{\sigma_{\mathrm{a}}^2}\right)\frac{\partial\bar{a}_{jnk}}{\partial\mathbf{V}_j} + \left(\frac{\beta_{jn}-\bar{b}_{jnk}}{\sigma_{\mathrm{a}}^2}\right)\frac{\partial\bar{b}_{jnk}}{\partial\mathbf{V}_j}\right],$$

[4] The procedure we use is not standard gradient ascent, due to complications discussed in the next paragraph.

where

$$\frac{\partial \bar{a}_{jnk}}{\partial \mathbf{x}_k} = \frac{d_f\left(\mathbf{M}_1^{jn} - \mathbf{M}_3^{jn}\bar{a}_{jnk}\right)}{\mathbf{M}_3^{jn}\left(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t_{jn}\right)^{\mathrm{T}}},$$

$$\frac{\partial \bar{b}_{jnk}}{\partial \mathbf{x}_k} = \frac{d_f\left(\mathbf{M}_2^{jn} - \mathbf{M}_3^{jn}\bar{b}_{jnk}\right)}{\mathbf{M}_3^{jn}\left(\mathbf{x}_k - \mathbf{X}_{0j} - \mathbf{V}_j t_{jn}\right)^{\mathrm{T}}},$$

$$\frac{\partial \bar{a}_{jnk}}{\partial \mathbf{X}_{0j}} = -\frac{\partial \bar{a}_{jnk}}{\partial \mathbf{x}_k},$$

$$\frac{\partial \bar{b}_{jnk}}{\partial \mathbf{X}_{0j}} = -\frac{\partial \bar{b}_{jnk}}{\partial \mathbf{x}_k},$$

$$\frac{\partial \bar{a}_{jnk}}{\partial \mathbf{V}_j} = -t_{jn}\frac{\partial \bar{a}_{jnk}}{\partial \mathbf{x}_k},$$

$$\frac{\partial \bar{b}_{jnk}}{\partial \mathbf{V}_j} = -t_{jn}\frac{\partial \bar{b}_{jnk}}{\partial \mathbf{x}_k}.$$

We have now described equations [Eqs. (15)–(19)] for estimating the parameters $\{\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j, \bar{\mathbf{F}}_k, \mathbf{C}_{fk}\}$ as functions of the conditional probabilities $P(k|jn)$. However, because the reverse is also true, i.e., the probabilities $P(k|jn)$ are functions of the unknown parameters, an analytical solution for the parameters is intractable. However, Eqs. (15)–(19), together with Eq. (13) form an iterative, convergent system that works as follows. First, an initial guess is made for the parameters $\{\mathbf{x}_k, \mathbf{X}_{0j}, \mathbf{V}_j, \bar{\mathbf{F}}_k, \mathbf{C}_{fk}\}$. Using the initial parameter values, the probabilities $P(k|jn)$ are then computed for each sample using Eq. (13). Next, these probabilities are used to update the parameter estimates using Eqs. (15)–(19). The updated parameters are now used *again* to update the probabilities $P(k|jn)$, and so on. In this manner, the parameters are estimated concurrently with data association using a system of dynamic logic. The system is convergent in the sense that, for each successive iteration, the log-likelihood is proven to be non-decreasing (refer to [1, Section 4.6]). It should be noted that the convergence proof involves a form of *generalized expectation-maximization* (EM) (refer, for example, to [43, Section 3.9]), although the way we utilize EM here has not been done previously. In the simulations described in the next section, we stopped after a predetermined number of iterations, on the order of 100, although this number could have easily been reduced.

A salient aspect of this approach is that variances in estimated probabilities $P(k|jn)$ should match uncertainties in model parameters. This is achieved as follows. As we stated previously, the standard deviation σ_a of the error in signature position (e.g., due to errors in the alignment of the camera axis) is known a priori. However, to reflect our initial uncertainty regarding target positions, this parameter is initially set to a large value, say $\sigma_{a0} = \Delta_\alpha/K$, where Δ_α is the extent of the focal plane and K is the number of targets. Then σ_a is made to evolve (shrink) with each iteration according to a predetermined schedule until it settles at the a priori known uncertainty value σ_{ar}. An example of such a schedule is $\sigma_a = \sigma_{ar} + (\sigma_{a0} - \sigma_{ar})\exp\{-\gamma(I-1)\}$ where I denotes the iteration number and γ is a (positive) constant governing the rate of decay (this should be matched to the expected number of total iterations).

Upon convergence of this iterative association–estimation procedure, a decision needs to be made whether or not each of the target objects k is a threat. This is accomplished using a likelihood ratio test (e.g., Chapter 2 of [43]), precisely in the manner described for the CAT problem (refer to [2,3], or Section 7.2.9 of [1]).

There are several practical issues not yet discussed, including initialization of parameter values and choosing the best number of models K. Choosing the best number of models K is a complex issue [1] and, in practice, we choose K to be larger than the expected number of targets. Excess models will generally be weeded out automatically during the likelihood ratio test, although this issue needs further study. Parameter initialization is based upon a combination of random selection and a priori knowledge. The UAV position and velocity parameters can be initialized based on GPS measurements, while initial guesses for target positions $\mathbf{x}_k$ are random. Initial estimates for the feature-based parameters $\{\bar{\mathbf{F}}_k, \mathbf{C}_{fk}\}$ are based on rough estimates from the data—generally the covariance is initialized to a large value to reflect our initial uncertainty, then the covariance shrinks automatically during the iterations.

5. Practical issues

The crux here is data association, which is especially difficult in our problem because we may have many sensors. Our approach is efficient in the sense that it scales linearly with the size of the problem. For example, if we have K targets (including clutter), J sensors, and N data samples, the computational complexity is O[KJN]. In contrast, multiple hypothesis tracking (MHT), which is also optimal, requires an exhaustive search over mappings between data and targets, and therefore is O[K^{JN}]. For a moderate sized problem having $K = 10$, $J = 10$, and $N = 500$, our approach requires on the order of 50,000 operations while MHT would require roughly 10^{5000} operations! Of course, there are approximations that can be used to reduce the number of computations in MHT, however not enough to make it feasible without performance sacrifice.

It should be noted that the computational complexity of our approach is also proportional to the number of iterations in the parameter estimation/data association loop. We have found in our simulations that roughly 30–50 iterations are required for adequate convergence. This factor will affect the feasibility of the approach for real time processing, however this should not be a "show stopper" given the rapid advances in computational speed. The real barrier to real time processing will be in the data association step, which we can solve efficiently, as discussed above. The feasibility of the approach for real time processing is important, and we will study this issue further in a future work.

Another practical issue needing attention is the amount of information that must be communicated between the multiple sensor platforms. Since the data from all sensors needs to be processed together, it must be communicated

between UAVs, whether we choose a centralized or distributed processing architecture. However, this does not mean that all pixel values from the digital images need to be shared. Rather, only the preprocessed data $\mathbf{w}_{jn}(t) = (\hat{\mathbf{X}}_{jn}, \alpha_{jn}, \beta_{jn}, \mathbf{f}_{jn}, t_{jn})$ needs to be communicated. How, exactly, the data is communicated is not in the scope of this paper, but it will require further study. The interested reader may refer to [38], in which the authors describe an experiment in which SLAM was implemented on a decentralized architecture consisting of four UAVs. Also, in [39], the authors describe a communications and scheduling framework with which multiple UAVs may communicate with each other. This work mainly discusses the software and hardware specifications rather than mathematical algorithms for UAV navigation or target mapping.

Finally, we have described how the approach works over a relatively short time interval, over which the sensor (UAV) trajectories are approximately linear. Over larger time intervals, we envision the method working in a manner similar to the (related) multiple target tracking (MTT) approach [1,2,4,5], in which a sliding window is used to process separately each of the multiple overlapping groups of frames which overlap in time. For example, the first group would consist of frames acquired at t_1–t_6, the second would consist of t_4–t_9, etc. Then, legitimate tracks established in the first group would be propagated to the second group, while also allowing new tracks to be initiated. Thus, the sensor (UAV) trajectories are approximated as piecewise-linear functions. Alternatively, more complicated trajectory models may be used including polynomials, or link-track models [1].

6. Results from simulated data

Results of computer simulations are now presented to demonstrate the algorithm. We first consider the simplified two-dimensional case, which highlights some important issues. We then present results for the fully three-dimensional case, which more closely approximates an experimental system.

We now discuss the two-dimensional case in which $\mathbf{x} = (x, z)$, and in which the UAVs fly at a constant elevation with their cameras axes pointing directly downward, i.e., $(yaw, pitch, roll) = (\omega, \phi, \kappa) = (0, 180°, 0)$. Thus, the equation of UAV motion is [compare with Eq. (4)]

$$\mathbf{X}_j(t) = \begin{bmatrix} X_j(t) \\ Z_j(t) \end{bmatrix} = \begin{bmatrix} X_{0j} + V_j t \\ Z_{0j} \end{bmatrix}, \tag{20}$$

and the equation relating a two-dimensional target position $\mathbf{x}_k = (x_k, z_k)$ to the one-dimensional position α of its signature on the focal plane is [compare with Eq. (5)]

$$\alpha = d_f \left(\frac{x_k - X_{0j} - V_j t}{z_k - Z_{0j}} \right). \tag{21}$$

Furthermore, all equations in Section 4 relating to parameter estimation can be simplified by noting that $\mathbf{M}_1^{jn} = [-10]$, $\mathbf{M}_3^{jn} = [0 - 1]$. The full list of parameters we need to estimate is thus $\{x_k, z_k, X_{0j}, Z_{0j}, V_j, \bar{F}_k, C_{fk}\}$.

For this simulation we did not perform preprocessing on actual images, rather we assumed that preprocessing was already performed, and generated our data samples by random draw. Signatures from both "targets" (i.e., specific objects in the environment) and "clutter" were generated, with 100 clutter signatures per image frame, and various numbers of target signatures per image frame. For the target signatures, the positions α were generated according to Eq. (21), while for clutter signature positions were generated by uniform random draw. For classification features we used a one-dimensional vector (single feature) f having mean $\bar{F}_k$ and variance C_{fk}. For these simulations we considered the low-clutter case with clutter statistics $(\bar{F}_0, C_{f0}) = (0, 1)$ vs. target signature statistics having means $\bar{F}_k$ for each target k randomly drawn within the range of values (5–10), and variances $C_{fk} = 1$.

The first set of simulations was designed to study tracking and localization performance vs. number of UAVs in the swarm. The geometry was selected with targets distributed at random over the ranges $(0 \leqslant z_k \leqslant 3.5)$, $(-10 \leqslant x_k \leqslant 10)$ m, and UAV parameters distributed at random within the ranges $(16 \leqslant Z_{0j} \leqslant 24)$, $(-10 \leqslant X_{0j} \leqslant 10)$ m and $(20 \leqslant |V_j| \leqslant 26)$ m/s. Also, velocity components V_j were randomly chosen to have either $\pm$polarity. It was assumed that the error in the GPS measurements was normally distributed, zero-mean, with standard deviation $\sigma_g = 4$ m in both x and z. In addition, a random error with standard deviation $\sigma = 1°$ was introduced into the alignment angle of the camera axes. Each UAV acquired three image frames as it flew, at times $[0, 0.5, 1]$ s. At these intervals the GPS also acquired position estimates. For parameter estimation, the step size was chosen so that roughly 30–50 iterations were required for parameter estimates to settle to steady states.

Various combinations of (number of targets) and (number of UAVs) were considered and, for each combination, 1000 Monte Carlo type simulations were run with random draws on (a) target and UAV positions, (b) GPS measurements, (c) misalignment in angles of camera axes, and (d) classification features. Fig. 2 shows a plot of the error standard deviation for the UAV elevation Z_{0j} estimate, as a function of both the number of UAVs in the swarm and the number of targets being tracked (the error shown in the plot represents the average over UAVs). In Fig. 3 we compare, on the same set of axes, the error in both horizontal X_{0j} and elevation Z_{0j} estimates for the 2-target case. Also shown in the figures are the errors that would be obtained by simply performing linear regression on the GPS measurements, separately for each UAV. The figures indicate that with a single UAV in the swarm, the estimates are no better than the estimates obtained by performing linear regression on the GPS measurements, which would be the best possible result if you discarded the information from the image sensors. However, as long as at least two targets are tracked, the error standard deviation decreases

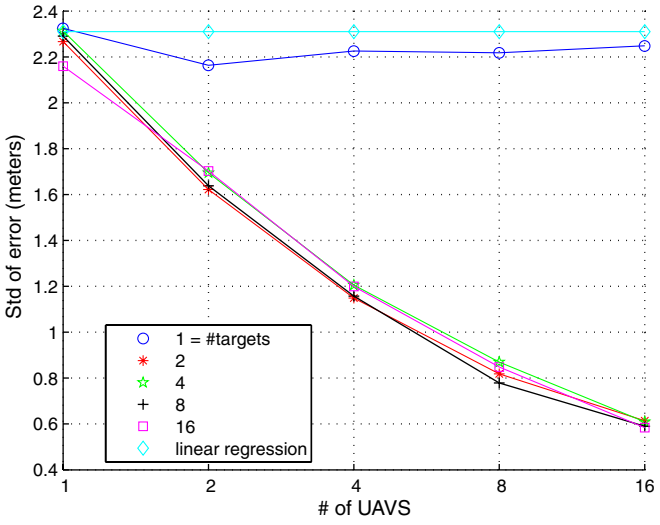

Fig. 2. Standard deviation of error in the estimate for UAV elevation Z_{0j} vs. number of UAVs in the swarm and number of targets.

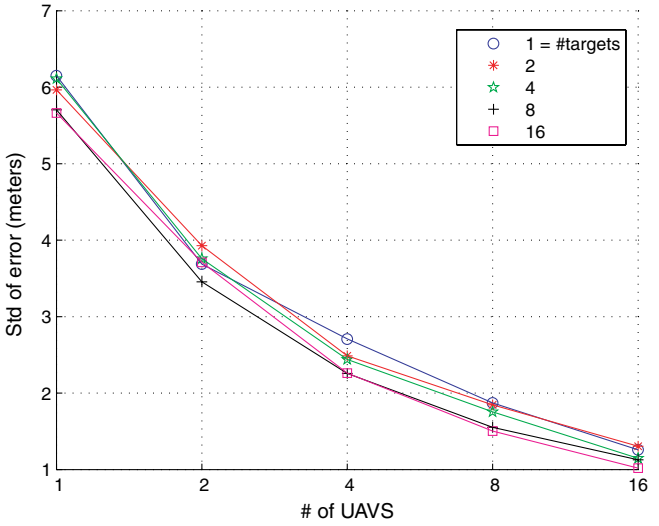

Fig. 4. Standard deviation of error in the estimate for target elevation z_k vs. number of UAVs in the swarm and number of targets.

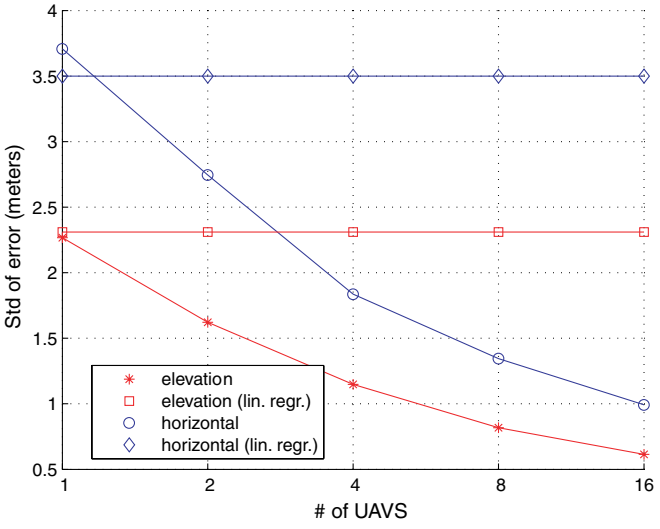

Fig. 3. Standard deviation of error in the estimate for UAV elevation Z_{0j} and horizontal position X_{0j} vs. number of UAVs in the swarm (two targets).

by a factor roughly proportional to $1/\sqrt{J}$, where J is the number of UAVs in the swarm. Here is clear evidence of the advantages of using swarms. The error in target elevation z_k estimates follows a similar pattern, as shown in Fig. 4. A plot of errors in x_k would be similar. By linking together the positions of the multiple UAVs using the target signatures, you are able to "beat down" the noise in the GPS measurements. The factor of $1/\sqrt{J}$ makes sense because the situation is similar, in a way, to taking multiple measurements (e.g., of a voltage) in the presence of Gaussian noise, in which case the error standard deviation decreases as one over the square root of the number of measurements.

It is intuitively evident why the error remains high if only a single target is present—the situation is similar to the orienteering problem in which you seek to pinpoint your position on a map using compass readings to various geological landmarks. The accuracy will increase as you move from one line of bearing to two lines (a "cut"), and then to three lines (a "fix"). Of course the advantage you gain is dependent upon how far apart (in angle) the landmarks are from each other.

Next we consider graphically how the parameter estimates evolve with successive iterations of the MFT process described in Section 4. Fig. 5 shows how the target position parameters x_k evolve over 30 iterations for a particular scenario having three targets and two UAVs. Fig. 6 is the analogous plot showing how the errors decrease with successive iterations. Note that in this case the errors settled

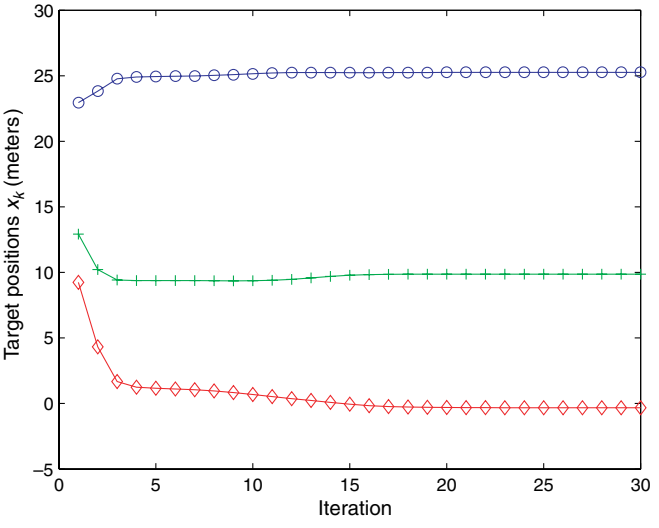

Fig. 5. Evolution of target position x_k estimates vs. iteration number for 3-target, 2-UAV case. Note that the true positions of the targets are 25 m for target 1 (curve marked by circles), 10 m for target 2 (curve marked by 'plus' signs), and 0 m for target 3 (curve marked by diamonds).

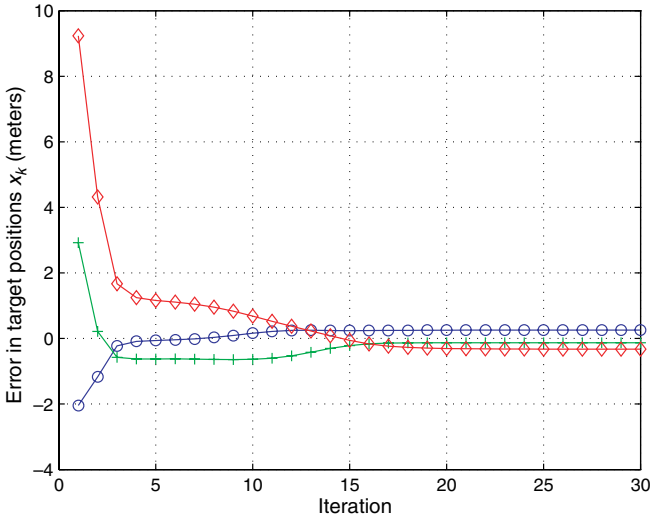

Fig. 6. Evolution of errors in target position estimates for the scenario shown in Fig. 5.

to a steady state after only around 18–20 iterations. The exact nature of parameter evolution depends upon how the parameters are initialized. In this case, target parameters were initialized at random, while UAV parameters were initialized using GPS measurements. The classification feature parameters were initialized by rough analysis of the data. One question that was not studied here was how to pick the number of targets in the model—for all our simulations we simply used the known quantity. Typically, in MFT, one will choose an excessive number of target components, and the unneeded components will (in many cases) tend to double up. In such cases it is a simple matter to check for redundancy. This issue requires further study, however.

The two-dimensional results shown above are valuable for illustrating some important concepts. However, it is also important to consider the full three-dimensional case, which more closely approximates reality. Thus, we extended the computer code to handle the case in which targets and UAVs have x, y, and z positions, and UAVs

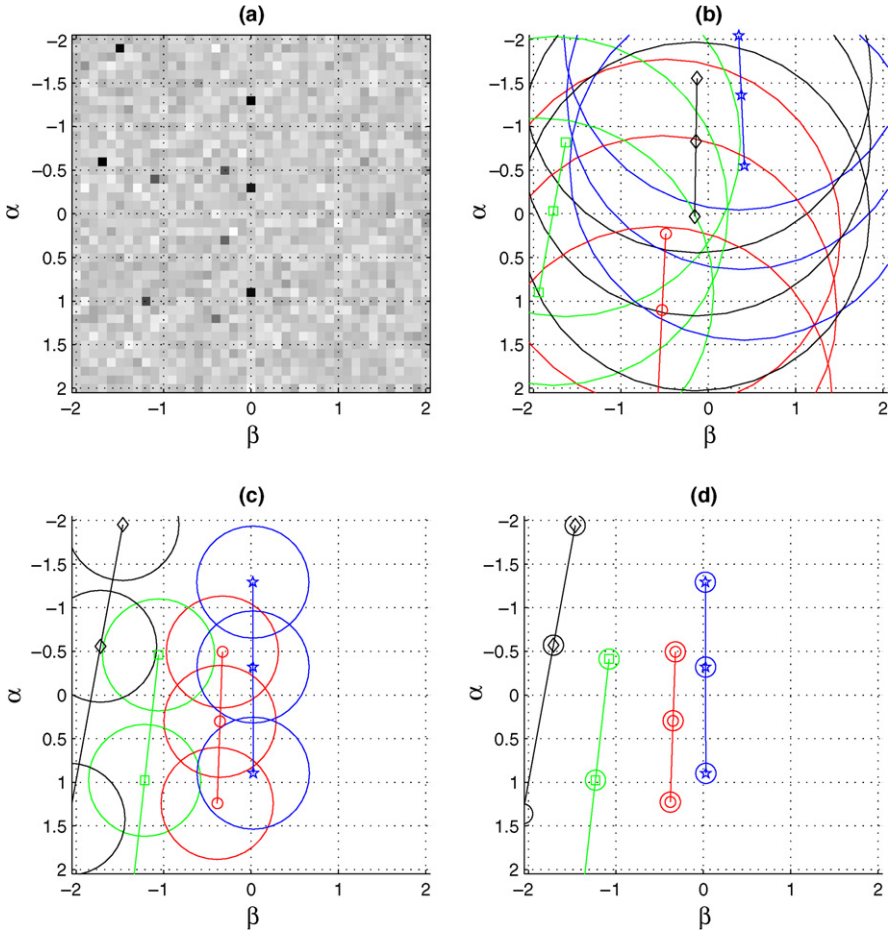

Fig. 7. Results from the low-clutter example, UAV 1, of 3 total. In (a) the preprocessed feature data is shown distributed over the sensor focal plane. Here the high values of the target features show up as relatively dark pixels over a lighter, speckled, clutter background. In (b) the initial, randomly selected, estimates for target signature positions are shown as symbols connected by lines; three symbols (corresponding to three different time instances) for each of four targets. The large circles around signature positions indicate the high initial uncertainty in the estimates. Plots (c) and (d) show the evolution of the signature position estimates at iterations 10 and 50, respectively. Here, the radii of uncertainty shrink with increasing iterations as the data association becomes less ambiguous.

have x, y, and z velocity components. Throughout these simulations the cameras were assumed to point directly downward. We first considered two examples having four targets distributed within the ranges $[-20 \leqslant (x_k, y_k) \leqslant 20]$, and $[0 \leqslant z_k \leqslant 10]$, and three UAVs distributed within the ranges $[-30 \leqslant (X_{0j}, Y_{0j}) \leqslant 30]$ and $[15 \leqslant Z_{0j} \leqslant 20]$. The UAV velocities were distributed within the ranges $[-10 \leqslant (\dot{X}_j, \dot{Y}_j) \leqslant 10]$ and $[-2 \leqslant \dot{Z}_j \leqslant 2]$. The full sensor model given by Eq. (5) was used to calculate the data at time samples $t = (0, 1.5, 3)$ (frame times). For example, in a realistic close-range scenario, all time units might be in seconds and all position units in m.

First, a relatively low-clutter example was investigated. Here, 1600 clutter samples per frame were randomly generated having a single classification feature f with mean and variance $(\bar{F}_0, C_{f0}) = (0, 0.75)$. The target features were also randomly drawn from distributions having variance $C_{fk} = 0.75$ and means of $\bar{F}_k = [5.5, 7.5, 9.5, 11.5]$, respectively for $k = 1, 2, 3, 4$. The K-factor is a commonly used quantitative measure of the degree of separation between two distributions having equal variances. If σ^2 is the variance and ΔM is the separation between the means, then $K = \Delta M/\sigma$. Thus, for this example the K-factors of each of the four targets vs. the clutter are roughly $K = [6, 9, 11, 13]$. Also, the standard deviations of the GPS and signature position errors were set to $\sigma_g = 4$ and $\sigma_{ar} = 0.1$, respectively.

Fig. 7a–d show the results of the simulations, plotted over the space of the UAV 1 sensor focal plane. In (a) the distribution of preprocessed feature data is shown. Here the high values of the target classification features show up as relatively dark pixels over a lighter, speckled, clutter background. For display purposes, the target signatures from all three frame times are shown superimposed onto a single frame of clutter, thus for each of the four targets there are (potentially) three dark pixels corresponding to the three time instances. In (b) the initial, randomly selected, estimates for target signature positions are shown

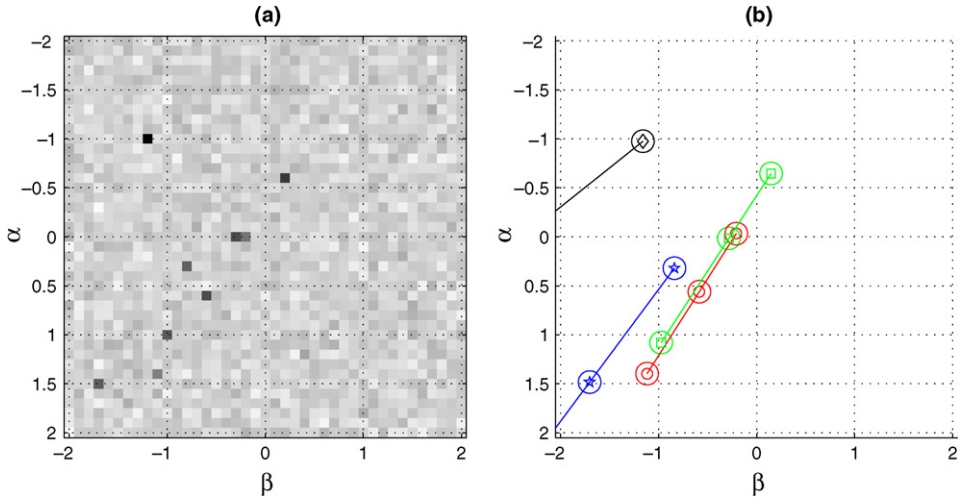

Fig. 8. Results from the low-clutter example, UAV 2, of 3 total. These plots are analogous to Fig. 7a and d, respectively.

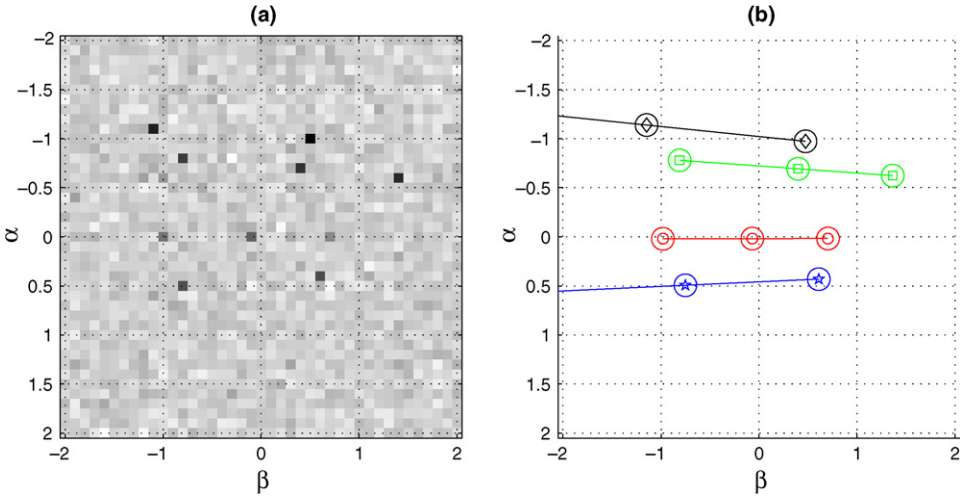

Fig. 9. Results from the low-clutter example, UAV 3, of 3 total. These plots are analogous to Fig. 7a and d, respectively.

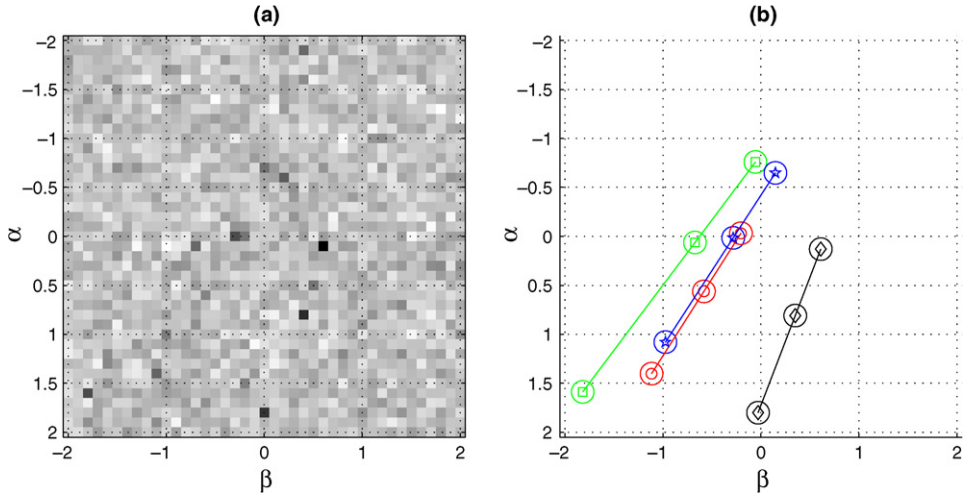

Fig. 10. Results from the higher-clutter example in which target and clutter feature standard deviations have been doubled. These plots correspond to UAV 2, of 3 total, and are analogous to Fig. 7a and d, respectively. Please note that the target and UAV positions are different than in the low-clutter example.

as symbols connected by lines; three symbols for each of four targets. The large circles around signature positions indicate the high initial uncertainty in the estimates—in fact the circle radii are simply the initial value of σ_a. Plots (c) and (d) show the evolution of the signature position estimates at iterations 10 and 50, respectively. Here, the radii of uncertainty σ_a shrink with increasing iterations as the data association becomes less ambiguous. In (d), the data has been properly associated, and the signatures for all four targets have been identified at all frame times. Figs. 8 and 9 show the analogous information for UAVs 2 and 3. Note that this experiment was set up such that various targets drift out of the sensor fields-of-view during data collection, as might very well be the case in a true scenario.

Next we repeat the experiment for a higher-clutter example. Here the mean statistics for the clutter and target classification features remain the same, however the standard deviations are doubled. Therefore, now the K-factors of each of the four targets vs. the clutter are roughly $K = [3, 4, 5, 7]$. Fig. 10a and b show the results plotted for UAV 2 (the analogous plots for UAVs 1 and 3 are not shown). It is difficult to pick out, by eye, the signatures corresponding to the lower signal-to-clutter targets. However, the algorithm was able to automatically detect and track all of the four targets, despite the relatively high level of clutter.

Finally, we generated Monte Carlo results to study the effects of the clutter level on algorithm performance. The

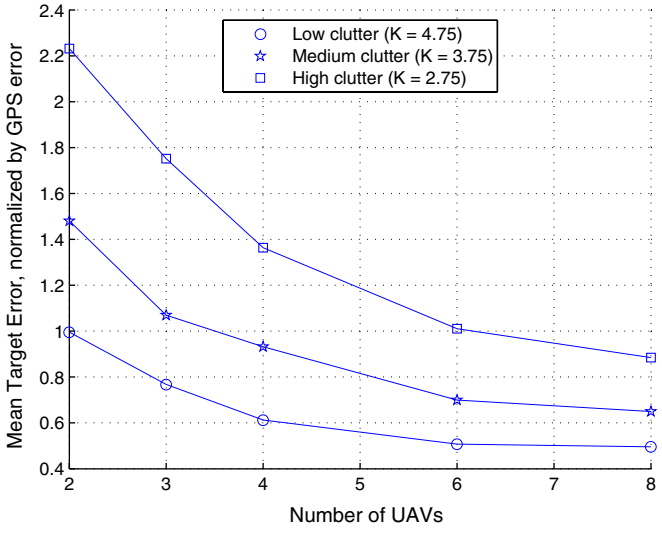

Fig. 11. Errors in estimated target position vs. signal-to-clutter (proportional to the K-factor) and the number of UAVs in the swarm. The vertical axis in these plots indicates the average in radial error $\sqrt{\text{error}_x^2 + \text{error}_y^2 + \text{error}_z^2}$, normalized by GPS error, and averaged over 100 Monte Carlo iterations for each data point.

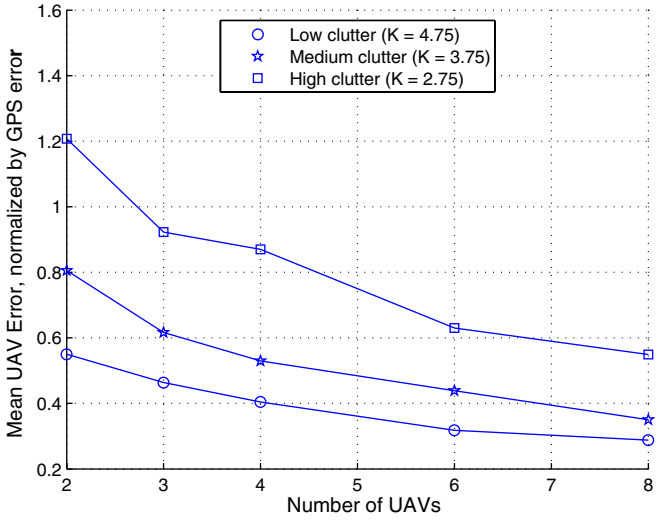

Fig. 12. Errors in estimated UAV position over the same conditions as the analogous plot in Fig. 11.

error distributions were chosen as in the preceding examples, and target and UAV positions and UAV velocities were generated randomly within the ranges specified above. Figs. 11 and 12 plot the errors in estimated target and UAV positions as functions of K-factor (related to the S/C ratio as described above) and number of UAVs in the swarm. The vertical axis in these plots indicates the average in radial error $\sqrt{\text{error}_x^2 + \text{error}_y^2 + \text{error}_z^2}$, normalized by GPS error, and averaged over 100 Monte Carlo iterations for each data point. From these plots it is apparent that both target and UAV position errors increase roughly linearly with decreasing S/C. Also, the errors decrease as roughly $1/\sqrt{J}$, as J ranges from 2 to 8, where J is the number of UAVs in the swarm. This trend was also observed in the results from the two-dimensional experiments.

7. Conclusions and directions for future work

Based upon preliminary tests, the approach described in this paper appears to offer a feasible solution to the problem of concurrent multi-target localization and navigation for a swarm of flying optical sensors. Using simulated data, we tested a fully three-dimensional version of the method over various scenarios in which the clutter level, GPS error, and number of sensors in the swarm were varied. From the results, it is apparent that increasing the size of the swarm will significantly boost performance, especially in lower signal-to-clutter (S/C) situations.

The potential advantages of this method vs. other approaches are discussed in the paper. Most importantly, the approach is not subject to combinatorial complexity during data association. This issue is particularly important for low S/C data and for larger swarms of sensors. Also, since data association is performed probabilistically rather than deterministically, the approach is not subject to catastrophic failure if data association is imperfect—an important limitation in some existing methods [27,31]. Finally, since all functions are performed concurrently, the approach is optimum [1,3] and can, in principle, be used to track very low S/C targets. In contrast, alternative methods in which detection is performed separately are likely to discard lower S/C samples based upon an arbitrary threshold.

The next step will be to design and build an experimental system. Practical issues that must be resolved include real time processing and efficient data communication between platforms. Although these issues are outside the scope of this paper, we have mentioned them in some detail in Section 5.

References

[1] L.I. Perlovsky, Neural Networks and Intellect: Using Model-based Concepts, Oxford University Press, New York, NY, 2001.
[2] L.I. Perlovsky, W.H. Schoendorf, L.C. Garvin, W. Chang, J. Monti, Development of concurrent classification and tracking, J. Underwater Acoust. 47 (1) (1997) 202–210.
[3] L.I. Perlovsky, Cramer-Rao bound for tracking in clutter and tracking multiple objects, Pattern Recogn. Lett. 18 (3) (1997) 283–288.
[4] L.I. Perlovsky, Tracking multiple objects in visual imagery, Conference on Neural Networks for Vision and Image Processing, Tyngsboro, MA, 1991.
[5] L.I. Perlovsky, V.H. Webb, S.R. Bradley, C.A. Hansen, Improved ROTHR detection and tracking using MLANS, AGU Radio Sci. 33 (4) (1998) 1034–1044.
[6] R. Deming, L. Perlovsky, R. Brocket, Sensor fusion for swarms of unmanned aerial vehicles using modeling field theory, 2005 IEEE Int'l Conf. on Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Evolution, and Engineering (KIMAS 2005), Waltham, MA, April 18–21, 2005.
[7] Office of the Secretary of Defense, Unmanned Aerial Vehicles Roadmap 2002–2027, 2002.
[8] R. Wall, The latest leap, Aviat. Week Space Technol. 6 (September) (2004) 46–47.
[9] R. Wall, On the offensive, Aviat. Week Space Technol. 6 (September) (2004) 49–50.
[10] S. Winkler, P. Vorsmann, Bird's-eye view: GPS and micro aerial vehicles, GPS World (October) (2004) 14–22.
[11] G.J. McLachlan, K.E. Baskford, Mixture Models, Marcel Dekker, Inc., New York, NY, 1988.
[12] S.J. McKenna, Y. Raja, S. Gong, Tracking colour objects using adaptive mixture models, Image Vision Comput. 17 (1999) 225–231.
[13] Y. Bar-Shalom, T.E. Fortmann, Tracking and Data Association, Academic Press, New York, 1988.
[14] J.A. Castellanos, J.M.M. Montiel, J. Neira, J.D. Tardos, The SPmap: a probabilistic framework for simultaneous localization and map building, IEEE Trans. Robotic. Autom. 15 (Oct.) (1999) 948–953.
[15] M. Csorba, H.F. Durrant-White, New approach to map building using relative position estimates, in: SPIE, Orlando, FL, April 1997, pp. 115–125.
[16] A. Davison, Y.G. Cid, N. Kita, Real time 3D SLAM with wide-angle vision, in: Intelligent Autonomous Vehicles, Lisbon Portugal, July 2003, IEEE.
[17] M. Dissanayaka, P. Newman, S. Clark, H. Durrant-White, M. Csorba, A solution to the simultaneous localisation and map building problem, IEEE Trans. Robotic. Autom. 17 (3) (2001) 229–241.
[18] H.F. Durrant-White, M. Dissanayake, Toward deployment of large scale simultaneous localisation and map building, Control Theory and Applications, IEEE Proceedings, vol. 142, pp. 385–400.
[19] H.J.S. Feder, J.J. Leonard, C.M. Smith, Adaptive mobile robot navigation and mapping, Int. J. Robot. Res. 18 (7) (1999) 650–668.
[20] J. Fenwick, P. Newman, J. Leonard, Collaborative concurrent mapping and localization, in: IEEE Conf. on Robotics and Automation, Washington, DC, May 2002.
[21] J. Guivant, E. Nebot, Improving computational and memory requirements of simultaneous localization and map building algorithms, in: IEEE Conf. on Robotics and Automation, Washington, DC, May 2002, pp. 2731–2736.
[22] P. Jensfelt, S. Kristensen, Active global localisation for a mobile robot using multiple hypothesis tracking, IEEE Trans. Robotic. Autom. 17 (5) (2001) 748–760.
[23] I.K. Jung, S. Lacroix, High resolution terrain mapping using low altitude aerial sterio imagery, in: ICCV, Nice, France, July 2003, IEEE.
[24] J.J. Leonard, H.J.S. Feder, Decoupled stochastic mapping, MIT Marine Robotics Technical Memorandum, December 1999.
[25] J. Neira, J.D. Tardos, Data association in stochastic mapping using the joint compatibility test, IEEE Trans. Robotic. Autom. 17 (6) (2001) 890–897.
[26] E. Nettleton, H. Durrant-White, P. Gibbens, A. Goktogan, Multiple platform localisation and map building, in: G.T. McKee, P.S. Schenker (Eds.), Sensor Fusion and Decentralised Control in Robotic Systems III, Bellingham, vol. 4196, 2000, pp. 337–347.

[27] J. Nieto, J.E. Guivant, E.M. Nebot, S. Thrun, Real time data association for FastSLAM, in: ICRA, Taiwan, September 2003, IEEE.
[28] D. Schultz, W. Burgard, D. Fox, A.B. Cremers, Tracking multiple moving objects with a mobile robot, in: CVAP-01, Kauai, HW, December 2001, IEEE.
[29] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics, in: I.J. Cox, G.T. Wilfon (Eds.), Autonomous Robot Vehicles, Springer Verlag, New York, 1990, pp. 167–193.
[30] S. Thrun, W. Burgrad, D. Fox, A probabilistic approach to concurrent mapping and localization for mobile robots, Mach. Learning 31 (1998) 29–53.
[31] S. Thrun, A probabilistic online mapping algorithm for teams of mobile robots, Int. J. Robot. Res. 20 (5) (2001) 335–363.
[32] J.K. Uhlmann, S.J. Julier, M. Csorba, Nondivergent simultaneous map building and localization using covariance intersection, in: SPIE, Orlando, FL, April 1997, pp. 2–11.
[33] S. Williams, Efficient solutions to autonomous mapping and navigation problems, PhD Thesis, University of Sydney, 2001.
[34] Z. Zhang, O.D. Faugeras, A 3D world model builder with a mobile robot, Int. J. Robot. Res. 11 (4) (1992) 269–285.
[35] D. Reid, An algorithm for tracking multiple targets, IEEE Trans. Automat. Contr. 24 (AC-6) (1979) 84–90.
[36] C. Hue, J.P. Le Cadre, P. Perez, Sequential Monte Carlo methods for multiple target tracking and data fusion, IEEE Trans. Signal Process. 50 (2) (2002) 309–326.
[37] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[38] S. Sukkarieh, E. Nettleton, J.H. Kim, M. Ridley, A. Goktogan, H. Durrante-White, The ANSER project: data fusion across multiple uninhabited air vehicles, Int. J. Robot. Res. 22 (7–8) (2003) 505–539.
[39] V. Crespi, W. Chung, A.B. Jordan, Decentralized sensing and tracking for UAV scheduling, Proc. SPIE (2004).
[40] M.M. Thompson (Ed.), Manual of Photogrammetry, vol. 1, American Society of Photogrammetry, Falls Church, VA, 1966, Chapter II.
[41] W. Linder, Digital Photogrammetry: Theory and Application, Springer, 2003.
[42] E. Gulch, Information extraction from digital images: a KTH approach, in: A. Gruen, O. Kuebler, P. Agouris (Eds.), Automatic Extraction of Man-Made Objects from Aerial and Space Images, Birkhauser Verlag, Basel, 1995, pp. 73–82.
[43] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, NY, 2001.
[44] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1986 (Chapter 9).