Tracking Multiple Ground Objects Using a Team of Unmanned Air Vehicles

Joshua Y. Sakamaki, Randal W. Beard and Michael Rice

Abstract This paper proposes a system architecture for tracking multiple ground-based objects using a team of unmanned air systems (UAS). In the architecture pipeline, video data is processed by each UAS to detect motion in the image frame. The ground-based location of the detected motion is estimated using a geolocation algorithm. The subsequent data points are then process by the recently introduced Recursive RANSAC (R-RANSASC) algorithm to produce a set of tracks. These tracks are then communicated over the network and the error in the coordinate frames between vehicles must be estimated. After the tracks have been placed in the same coordinate frame, a track-to-track association algorithm is used to determine which tracks in each camera correspond to tracks in other cameras. Associated tracks are then fused using a distributed information filter. The proposed method is demonstrated on data collected from two multi-rotors tracking a person walking on the ground.

1 Introduction

The objective of this paper is to describe a new approach to real-time video tracking of multiple ground objects using a team of multi-rotor style unmanned air systems (UAS). Many UAS applications involve tracking objects of interest with an on-board camera. These applications include following vehicles [15], visiting designated sites of interest [12], tracking wildlife [17], monitoring forest fires [8, 13], and inspecting infrastructure [24]. Current approaches to real-time video tracking from UAS can be brittle, and state-of-the-art techniques often require fiducial markings, or extensive human oversight.

There are numerous challenges in developing an object tracking system. For example, object tracking often involves finding image features and tracking those features from frame to frame. However, feature matching has a relatively high error rate, and

J.Y. Sakamaki · R.W. Beard (✉) · M. Rice
Brigham Young University, Provo, UT, USA
e-mail: beard@byu.edu

T.I. Fossen et al. (eds.), *Sensing and Control for Autonomous Vehicles*,
Lecture Notes in Control and Information Sciences 474,
DOI 10.1007/978-3-319-55372-6_12

the errors introduced by incorrect matches do not follow a Gaussian distribution. In addition, each measurement cycle, or image pair, produces many measurements where false measurements occur at a relatively high rate. Another challenge is distinguishing tracks from the background, especially when they stop moving, or have color and features similar to the background. Furthermore, even when objects of interest are correctly identified in each frame, the data association problem, or the problem of consistently associating the measurements with the correct object, can be difficult. Finally, many applications require that the system track many objects in the environment.

In this paper, we introduce a complete solution for tracking multiple ground-based objects using cameras on-board a team of UAS. Our solution draws upon several distinct technologies including geolocation [4, 7, 11], multiple target tracking [6, 28], track-to-track data correlation [1, 2], and distributed sensor fusion [16, 18].

In our framework we assume that there are multiple air vehicles each carrying an on-board camera. The computer vision algorithm on each vehicle returns a set of points in the image frame that may correspond to ground-based objects. In the implementation reported in this paper, we look for moving objects. The set of potential measurements are then processed using a geolocation algorithm and the GPS and IMU measurements on-board each vehicle, to project the measurements onto the ground plane. The geolocation algorithm is described in Sect. 2.2. The data is then processed using a newly introduced multiple target tracker called Recursive RANSAC [15, 21, 22]. The R-RANSAC algorithm produces a set of tracks that are communicated between vehicles on the team. The R-RANSAC algorithm is described in Sect. 2.3. Unfortunately, the geolocation process is imprecise, introducing potential biases between vehicles. For every pair of tracks, the bias must be determined, and our approach to this problem is described in Sect. 2.4. Since each object may not be seen by every UAS, it is necessary to determine whether tracks seen by UAS a, are also seen by UAS b. Our approach to track-to-track association is given in Sect. 2.5. Associated tracks are then fused using an information filter as described in Sect. 2.6. Finally, flight results using the complete system are described in Sect. 3.

2 System Architecture

The architecture for the tracking system that will be described in this paper is shown in Fig. 1. Each UAS instantiates a tracking pipeline that includes six key components. The first component in the pipeline, as shown in Fig. 1, is the UAS and gimbaled camera. We assume that the UAS contains an autopilot system, as well as a pan-tilt camera that can be automatically controlled to point along a desired optical axis. In this paper, we will assume an RGB camera and enough processing power on-board to process images at frame rate, and to implement the other components in the system. The second component shown in Fig. 1 is the Geolocation block. The purpose of the geolocation block is to transform the image coordinates into world coordinates based on the current pose of the UAS. A detailed description of the geolocation block is

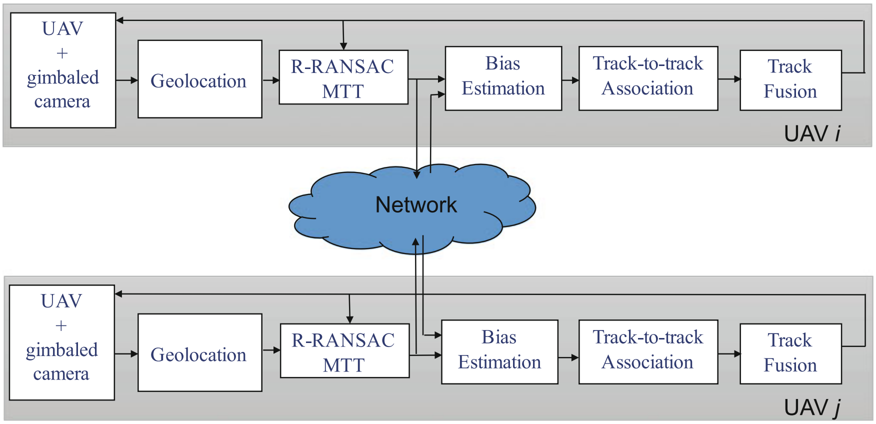


Fig. 1 Architecture for tracking multiple ground-based objects of interest using a team of UAS

given in Sect. 2.2. The next component shown in Fig. 1 is the Recursive RANdom SAmple Consensus Multiple Target Tracking (R-RANSAC MTT) block. This block uses image features in world coordinates to create and manage object tracks. This block performs several key tasks including data association, new track formation, track propagation, track collation, and track deletion. We define a *track* to be the time history of the system state (position, velocity, acceleration, etc.), as well as the associated covariance matrix. A more detailed description of this block will be given in Sect. 2.3. The current tracks maintained by each UAS is shared across the network with other UAS. The current collection of tracks is used in the Bias Estimation block shown in Fig. 1 to estimate the translational and rotational bias between each pair of tracks in the network, and thereby place all tracks in the coordinate system of the i^{th} UAS. Additional details about this process will be described in Sect. 2.4. The collection of tracks are then processed by the Track-to-track Association block shown in Fig. 1. This block uses a statistical test on a past window of the data to determine which tracks maintained by the i^{th} UAS are statistically similar to the tracks maintained by the j^{th} UAS. The details of this block are described in Sect. 2.5. When tracks are determined to be similar, they are fused in the Track Fusion block shown in Fig. 1. Track fusion is accomplished using an information consensus filter, as described in Sect. 2.6.

2.1 UAS and Gimbaled Camera

The techniques that are outlined in this paper are applicable to both multi-rotor systems and fixed wing vehicles. Independent of the type of aircraft used, we will assume that the sensor suite on-board the aircraft consists of a GPS aided IMU and associated filter algorithms that are able to estimate the 3D world position of the

UAS, as well as the inertial attitude of the UAS. We will also assume an altimeter that estimates the current height above ground of the aircraft. The altimeter may be a laser altimeter, or it may be an absolute pressure sensor that estimates the height above ground using the pressure difference between the take-off position and the current position. We will assume a flat earth model to simplify the discussion and equations. When an elevation map of the environment is known, the extension of these ideas to more complex terrain is conceptually straightforward.

We will assume that the UAS carries a gimbaled camera, where the gimbal can both pan and tilt, and possibly roll. For fixed wing vehicles, pan-tilt gimbals are common. For multi-rotor systems, pan-roll and pan-tilt-roll gimbal systems are common.

2.2 Object Geolocation

The UAS and gimbaled camera block shown in Fig. 1 produces a video stream, as well as state estimates obtained by filtering the on-board sensors. We will assume that the video stream is processed to produce a list of pixels, or image coordinates that represent possible measurements of objects on the ground. The task of the Geolocation block shown in Fig. 1 is to transform each image coordinate in the feature list into an inertial position on the ground. Object tracking can be performed in either the camera frame, or in the inertial frame. In order to perform object tracking using a team of UAS, the measurements of the objects need to be in a common reference frame. In this paper, we assume that all UAS on the team have GPS, therefore it makes sense to use GPS to define the common inertial reference frame, and to track the objects in the inertial frame. Transforming image coordinates to inertial coordinates is called geolocation in the literature. Geolocation algorithms for small UAS are described in [4, 5, 7, 10, 11, 23, 29].

Let $\mathscr{I}$ denote the inertial frame, let U_a denote UAS a, and let F_k denote the k^{th} feature of interest. Let $p_{U_a}^{\mathscr{I}}$ denote the inertial position of UAS a, and $p_{F_k}^{\mathscr{I}}$ denote the inertial position of the k^{th} feature. Define the line of sight vector between UAS a and the k^{th} feature, expressed in the camera frame as $\ell_{U_aF_k}^{\mathscr{C}_a} = p_{F_k}^{\mathscr{C}_a} - p_{U_a}^{\mathscr{C}_a}$. If $R_{\mathscr{C}_a}^{\mathscr{G}_a}$ denotes the rotation matrix from the camera frame to the gimbal frame of UAS a, $R_{\mathscr{G}_a}^{\mathscr{B}_a}$ denotes the rotation matrix from the gimbal frame to the body frame of UAS a, and $R_{\mathscr{B}_a}^{\mathscr{I}}$ denotes the rotation of the body frame of UAS a to the inertial frame, then the basic geolocation equation is given by [5]

$$p_{F_k}^{\mathscr{I}} = p_{U_a}^{\mathscr{I}} + R_{\mathscr{B}_a}^{\mathscr{I}} R_{\mathscr{G}_a}^{\mathscr{B}_a} R_{\mathscr{C}_a}^{\mathscr{G}_a} \ell_{U_aF_k}^{\mathscr{C}_a}. \tag{1}$$

The only element that is not available in Eq. (1) is the line of sight vector $\ell_{U_aF_k}^{\mathscr{C}_a}$. If we assume a pin-hole model for the camera, and that the focal length of the camera is f, and that the pixel location of the k^{th} feature is $(\varepsilon_{x_k}, \varepsilon_{y_k})$, then the line of sight vector is given by

$$\ell^{\mathscr{C}_a}_{U_aF_k} = L_{U_aF_k}\boldsymbol{\lambda}^{\mathscr{C}_a}_{U_aF_k}, \tag{2}$$

where $L_{U_aF_k}$ is the unknown length of the line of sight vector, and

$$\boldsymbol{\lambda}^{\mathscr{C}_a}_{U_aF_k} = \frac{1}{\sqrt{\varepsilon_{x_k}^2 + \varepsilon_{y_k}^2 + f^2}}\begin{pmatrix}\varepsilon_{x_k}\\ \varepsilon_{y_k}\\ f\end{pmatrix}$$

is the direction of the line of sight vector expressed in the camera frame. To determine $L_{U_aF_k}$ additional information about the terrain needs to be available. If an elevation map of the terrain is known, then $L_{U_aF_k}$ is determined by tracing the ray given by the right hand side of Eq. (1) to find the first intersection with the terrain. In other words, find the first $\alpha > 0$ such that

$$p^{\mathscr{I}}_{F_k} = p^{\mathscr{I}}_{U_a} + \alpha R^{\mathscr{I}}_{\mathscr{B}_a} R^{\mathscr{B}_a}_{\mathscr{G}_a} R^{\mathscr{G}_a}_{\mathscr{C}_a}\boldsymbol{\lambda}^{\mathscr{C}_a}_{U_aF_k}$$

intersects the terrain model. If the terrain is flat and the altitude h is known, then the equation for the length of the line of sight vector is given by [5]

$$L_{U_aF_k} = \frac{h}{\left(k^{\mathscr{I}}\right)^{\top} R^{\mathscr{I}}_{\mathscr{B}_a} R^{\mathscr{B}_a}_{\mathscr{G}_a} R^{\mathscr{G}_a}_{\mathscr{C}_a}\boldsymbol{\lambda}^{\mathscr{C}_a}_{U_aF_k}},$$

where $k^{\mathscr{I}} = (0, 0, 1)^{\top}$ is the unit vector pointing to the center of the earth.

Therefore, the geolocation block projects all 2D features in the image plane, into 3D features in the world frame, and returns a set of features in the inertial frame.

2.3 Multiple Object Tracking

The next step in the tracking pipeline shown in Fig. 1 is Recursive Random Sample Consensus Multiple Target Tracking (R-RANSAC MTT). The function of this block is to process the inertial frame measurements and to produce a set of tracks that correspond with objects on the ground. The R-RANSAC algorithm was recently introduced in [20] for static signals with significant gross measurement error, and extended in [21] to multiple target tracking, and in [15] to video tracking.

A graphic that highlights key elements of the algorithm are shown in Fig. 2. Figure 2a shows a single object on the ground, where the black dots represent current and past measurements. The small box around the object is a decision or measurement gate. As shown in Fig. 2b a set of trajectories that are consistent with the current measurement are created. For a given object, there many be many trajectories that are consistent with the current measurement. A set of these trajectories with the largest number of inlier measurements are retained in memory, and the trajectories that continue to be consistent with the measurements are retained. When other objects

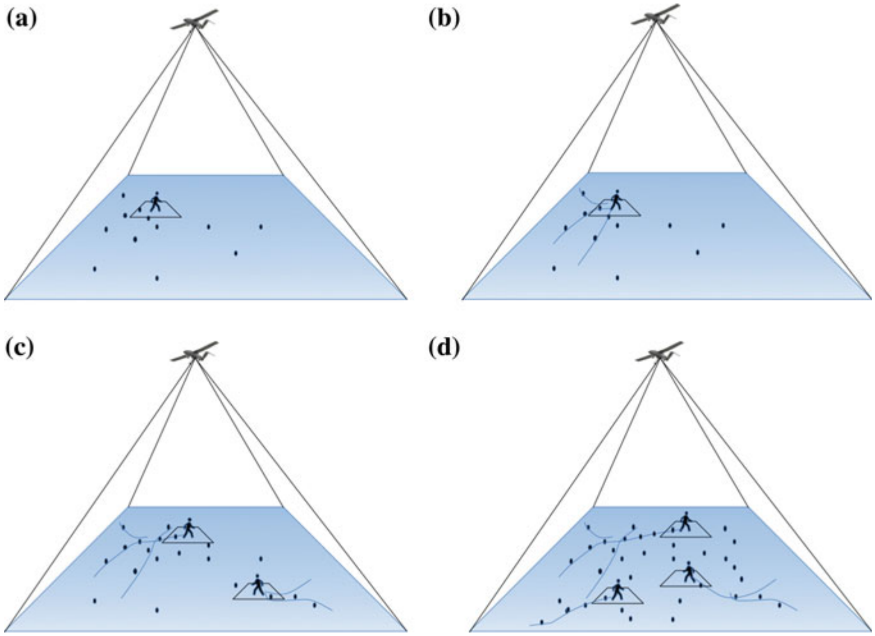


Fig. 2 Multiple object tracking using the R-RANSAC algorithm

appear, as shown in Fig. 2c they will generate measurements that are not in the measurement gate of existing tracks. When that happens, the initialization step is repeated, and a set of trajectories consistent with that measurement are added to memory. The R-RANSAC algorithm will have a bank of M possible trajectories in memory, and so pruning, merging, and spawning operations are key to its operation. In theory, the algorithm is capable of tracking $M - 1$ objects.

The R-RANSAC algorithm assumes a motion model for the objects of the form

$$x[t+1] = Ax[t] + \eta[t] \tag{3}$$

$$y[t] = Cx[t] + \nu[t], \tag{4}$$

where the size of the state is N, and where $\eta[t]$ and $\nu[t]$ are zero mean Gaussian random variables with covariance Q and R, respectively. We have found that for moving objects like pedestrians and vehicles on a road, constant acceleration and constant jerk models tend to work well [14]. The algorithm requires that all past measurements be retained in memory for the past D samples. The R-RANSAC initialization process begins by randomly selecting $N - 2$ time delays in the interval $[1, D - 1]$ denoted as $\{d_1, \ldots, d_{N-1}\}$. At each time delay, one measurement is randomly selected and denoted as $\{y_{d_1}, \ldots, y_{d_{N-1}}\}$. A measurement is also randomly selected at time $t - D$ and denoted $y[t - D]$. The state at time $t - D$ can then be reconstructed from the equation

$$\begin{pmatrix} y[t] \\ y[d_1] \\ \vdots \\ y[d_{N-1}] \\ y[t-D] \end{pmatrix} = \begin{pmatrix} CA^D \\ CA^{D-d_1} \\ \vdots \\ CA^{D-d_{N-1}} \\ C \end{pmatrix} \hat{x}[t-D]. \tag{5}$$

It can be shown that if the system is observable, then there is a unique solution for $\hat{x}[t-D]$ [19]. The state $\hat{x}[t-D]$ is propagated forward to the current time t using the discrete-time steady-state Kalman filter

$$\begin{aligned} \hat{x}^-[\tau+1] &= A\hat{x}[\tau] \\ \hat{x}[\tau+1] &= \begin{cases} \hat{x}^-[\tau+1] + L(y[\tau] - C\hat{x}[\tau]) & \tau \in \{t, t-d_1, \ldots, t-d_{N-1}, t-D\} \\ \hat{x}^-[\tau+1] & \text{otherwise,} \end{cases} \end{aligned} \tag{6}$$

where

$$L = P_p C^\top S^{-1}, \tag{7}$$

is the Kalman gain, $S = (CP_pC^\top + R)$ is the innovation covariance and P_p is the steady state prediction covariance that satisfies the algebraic Riccati equation

$$P_p = AP_pA^\top + Q - AP_pC^\top S^{-1} CP_pA^\top. \tag{8}$$

The quality of the initialized track is then scored by counting the number of measurements that are consistent with that track. Let $\mathscr{Y}_\tau$ be the set of all measurements received at time τ, and let $Y_\tau(z, \gamma)$ be the set of measurements that are a Mahalanobis distance of γ from z at time τ, i.e.,

$$Y_\tau(z, \gamma) = \{y \in \mathscr{Y}_\tau : (z-y)^\top S^{-1}(z-y) \leq \gamma\},$$

then the *consensus set* at time t for the j^{th} track $\{\hat{x}^j[\tau]\}_{\tau=t-D}^t$, is defined to be

$$\chi^j[t] = \bigcup_{\tau=t-D}^{t} Y_\tau(C\hat{x}^j[\tau], \gamma).$$

The *inlier ratio* $\rho^j(t)$ for track j is defined to be the size of the consensus set divided by the total number of measurements, i.e.,

$$\rho^j[t] = \frac{\left|\chi^j[t]\right|}{\sum_{\tau=t-D}^t |\mathscr{Y}_\tau|}. \tag{9}$$

The inlier ratio is a measure of the quality of the the track.

After a set of tracks have been initialized, the R-RANSAC algorithm processes the set of measurements $\mathscr{Y}[t]$ as follows. Let $G^j[t]$ be a defined gate for the j^{th} track at time t where

$$G^j[t] = \{z \in \mathbb{R}^p : (z - C\hat{x}^j[t])^\top S^{-1}(z - C\hat{x}^j[t]) \leq \gamma\}.$$

The set of measurements $\mathscr{Y}[t] \cap G^j[t]$ are combined using the probabilistic data association (PDA) algorithm [3], and then used to update the associated Kalman filter. Measurements that are outside of the gate for every existing track, are used to spawn new tracks, based on the previously defined initialization method. Two tracks are combined when their outputs are within a certain threshold of each other over a specified window of time.

2.4 Track Alignment

Object tracks produced by the R-RANSAC MTT algorithm are communicated across the network to other team members as shown in Fig. 1. When UAS a receives a track from UAS b, UAS a must determine if the track corresponds to any of its existing tracks. We call this the problem of track-to-track association. However, before testing for track-to-track association, the tracks from UAS a must be aligned with the track from UAS b.

Let $\hat{x}^j_{m|n}[t]$ represent the j^{th} track estimated by UAS m at time t, where the state is represented in the coordinate frame of UAS n. Each UAS will maintain a set of tracks in their own coordinate frame. In other words, UAS a will maintain the track of its j^{th} object as $\hat{x}^j_{a|a}$. The track can be transformed into the coordinate frame of UAS b using

$$\hat{x}^j_{a|b}[t] = R^b_a \left(\hat{x}^j_{a|a}[t] + d^b_a\right),$$

where R^b_a is the transformation matrix that rotates the coordinate frame of UAS a into the coordinate frame of UAS b, and $d^j_{a|b}$ is the associated translation. For example, when the state consists of the 2D ground position, velocity, and acceleration, then

$$R^b_a = I_3 \otimes \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$
$$d^b_a = \begin{pmatrix} \beta_n & \beta_e & 0 & 0 & 0 & 0 \end{pmatrix}^\top$$

where $\otimes$ represents the Kronecker product and θ is defined as the relative rotational bias angle between the two tracks about the negative down axis, and where β_n and β_e are constants representing the north and east translational bias.

Two tracks $\hat{x}^j_{a|a}$ and $\hat{x}^k_{b|b}$ are aligned over a window of length D by solving the optimization problem

$$(R_a^{b*}, b_a^{b*}) = \arg\min_{(R_a^b, b_a^b)} \sum_{\tau=t-D+1}^{t} \left\| \hat{x}_{b|b}^k[\tau] - R_a^b(\hat{x}_{a|a}^j[\tau] + d_a^b) \right\|. \tag{10}$$

It should be noted that obtaining a solution from the optimizer does not guarantee that the tracks are associated. To determine whether the tracks originate from the same object requires solving the track-to-track association problem, which is discussed in the next section.

2.5 Track-to-Track Association

After the tracks have been aligned, the next step shown in Fig. 1 is to test whether the two tracks do in fact originate from the same source. This is the classical track-to-track association problem [1]. The problem is formulated as a hypothesis test, where the two hypotheses are

$$\begin{aligned} &\mathrm{H}_0 : \text{The two tracks originate from the same object.}\\ &\mathrm{H}_1 : \text{The two tracks do not originate from the same object.} \end{aligned}$$

The association problem is solved over the past D measurement. Define the error vector as

$$\tilde{x}^{k_b j_a}[t] = \begin{pmatrix} \hat{x}_{b|b}^k[t-D+1] - R_a^b(\hat{x}_{a|a}^j[t-D+1] + d_a^b) \\ \hat{x}_{b|b}^k[t-D+2] - R_a^b(\hat{x}_{a|a}^j[t-D+2] + d_a^b) \\ \vdots \\ \hat{x}_{b|b}^k[t] - R_a^b(\hat{x}_{a|a}^j[t] + d_a^b) \end{pmatrix}. \tag{11}$$

Under the null hypothesis $\tilde{x}^{k_b j_a}[t]$ is a zero mean Gaussian random variable with covariance P_0, and under the alternative hypothesis $\tilde{x}^{k_b j_a}[t]$ is a zero mean Gaussian random variable with covariance P_1. The covariance matrix P_0 is known and will be discussed below. On the other hand, the covariance matrix P_1 is not known, and depends on the unknown true difference between the two unassociated tracks.

In the ideal case, where both P_0 and P_1 are known, the test statistic that follows from the log-likelihood ratio is [26]

$$L = \tilde{x}^{k_b j_a}[t]^\top \left(P_0^{-1} - P_1^{-1}\right) \tilde{x}^{k_b j_a}[t]. \tag{12}$$

However, because P_1 is unknown, this test statistic is unusable. We instead adopt the test statistic

$$\mathscr{D}[t] = \tilde{x}^{k_a j_b}[t]^\top P_0^{-1} \tilde{x}^{k_a j_b}[t]. \tag{13}$$

Under H_0, $\mathscr{D}[t]$ is a central chi-square random variable with DN degrees of freedom [26]. Under H_1, we use the Cholesky factorization $P_1^{-1} = W^\top W$ to write

$$\mathscr{D}[t] = \tilde{x}^{k_a j_b}[t]^\top W^\top W \tilde{x}^{k_a j_b}[t] = (W\tilde{x}^{k_a j_b}[t])^\top W\tilde{x}^{k_a j_b}[t]. \tag{14}$$

Here, $W\tilde{x}^{k_a j_b}[t]$ is a zero mean Gaussian random variable with covariance $WP_1W^\top$. Depending on the relationship between W and P_1, $\mathscr{D}[t]$ may or may not be a chi-square random variable [26]. In the event that $\mathscr{D}[t]$ is a chi-square random variable, it has less than DN degrees of freedom.

Because the likelihood ratio is an increasing function of $\mathscr{D}[t]$, by the Karlin–Rubin theorem, the following test is a uniformly most powerful test for testing H_0 against H_1 [26]:

$$\phi(\mathscr{D}[t]) = \begin{cases} 1, & \text{if } \mathscr{D}[t] > \mathscr{D}_\alpha \\ 0, & \text{if } \mathscr{D}[t] \le \mathscr{D}_\alpha \end{cases} \tag{15}$$

where $\phi(\mathscr{D}[t]) = 1$ means H_0 is rejected and $\phi(\mathscr{D}[t]) = 0$ means H_0 is not rejected. The decision threshold is found as follows. For a given false alarm probability

$$\alpha = \mathrm{P}\left(\phi(\mathscr{D}[t]) = 1 \mid H_0\right) = \mathrm{P}\left(\mathscr{D}[t] > \mathscr{D}_\alpha \mid H_0\right), \tag{16}$$

$\mathscr{D}_\alpha$ is computed from

$$\begin{aligned} \alpha &= 1 - F_{\mathscr{D}|H_0}(\mathscr{D}_\alpha) \\ &= 1 - \int_0^{\mathscr{D}_\alpha} \frac{1}{\Gamma(Nn_x/2)2^{(Nn_x/2)}} x^{(Nn_x/2)-1} e^{x/2} dx. \end{aligned} \tag{17}$$

Note that under H_0 this produces a probability of detection $P_d = 1 - \alpha$.

Under H_0, $\tilde{x}^{k_a j_b}[t]$ is a zero mean Gaussian random variable with covariance P_0 where the covariance is expressed as

$$P_0 = \lim_{t\to\infty} E\left\{\tilde{x}^{k_a j_b}[t]\tilde{x}^{k_a j_b}[t]^\top\right\}, \tag{18}$$

and where

$$\tilde{x}^{k_a j_b}[t] = \hat{x}^{k_a}[t] - \hat{x}^{j_b}[t],$$

and where $\hat{x}^{k_a}[t] \in \mathscr{R}^{DN\times 1}$ is the stacked vector associated with the past D estimates of the k^{th} track as observed by UAS a. Define the true track to be x^{k_a} and the track estimation error to be $\tilde{x}^{k_a} = \hat{x}^{k_a} - x^{k_a}$. Then P_0 can be written as

$$\begin{aligned} P_0 &= \lim_{t\to\infty} E\{\tilde{x}^{k_a j_b}[t]\tilde{x}^{k_a j_b}[t]^\top\} \\ &= \lim_{t\to\infty} E\{(\hat{x}^{k_a}[t] - \hat{x}^{j_b}[t])(\hat{x}^{k_a}[t] - \hat{x}^{j_b}[t])^\top\} \\ &= \lim_{t\to\infty} E\{(\tilde{x}^{k_a}[t] + x^{k_a}[t] - \tilde{x}^{j_b}[t] - x^{j_b}[t])(\tilde{x}^{k_a}[t] + x^{k_a}[t] - \tilde{x}^{j_b}[t] - x^{j_b}[t])^\top\}. \end{aligned}$$

Under hypothesis H_0, the two tracks originate from the same source, and since they are aligned in the same coordinate frame we have that $x^{k_a} = x^{j_b}$. Therefore

$$\begin{aligned} P_0 &= \lim_{t\to\infty} E\{(\tilde{x}^{k_a}[t] - \tilde{x}^{j_b}[t])(\tilde{x}^{k_a}[t] - \tilde{x}^{j_b}[t])^\top\} \\ &= \lim_{t\to\infty} E\{\tilde{x}^{k_a}[t]\tilde{x}^{k_a^\top}[t]\} + \lim_{t\to\infty} E\{\tilde{x}^{j_b}[t]\tilde{x}^{j_b^\top}[t]\} \\ &\quad - \lim_{t\to\infty} E\{\tilde{x}^{k_a}[t]\tilde{x}^{j_b^\top}[t]\} - \lim_{t\to\infty} E\{\tilde{x}^{j_b}[t]\tilde{x}^{k_a^\top}[t]\}. \end{aligned}$$

It can be shown using steady-state Kalman filter arguments, that P_0 has the structure

$$P_0 = \begin{pmatrix} P & PG^\top & P(G^2)^\top & \dots & P(G^{N-1})^T \\ GP & P & PG^\top & \dots & P(G^{N-2})^\top \\ G^2P & GP & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & PG^\top \\ G^{N-1}P & G^{N-2}P & \dots & GP & P \end{pmatrix}, \tag{19}$$

where $P = 2(P_e - P_c)$, and where P_e is the estimation covariance given by

$$P_e = P_p - P_pC^\top(CP_pC^\top + R)^{-1}CP_p, \tag{20}$$

and P_p is the prediction covariance given by the solution of the Riccati equation in Eq. (8), and where the cross covariance P_c satisfies

$$P_c = (I - LC)(AP_cA^\top + Q)(I - LC)^\top,$$

where L is the Kalman gain given in Eq. (7), and where

$$G = (I - LC)A.$$

The structure of (19) is convenient as it produces an inverse with a tridiagonal block form, as highlighted in the following theorem.

Theorem 1 *Consider the symmetric, positive definite block matrix P_0 defined by* (19)*. The inverse of the matrix P_0 is given by*

$$P_0{}^{-1} = \begin{pmatrix} U & V & 0 & \dots & 0 \\ V^T & W & V & \dots & 0 \\ 0 & V^T & \ddots & & \vdots \\ \vdots & \vdots & & W & V \\ 0 & 0 & \dots & V^T & Y \end{pmatrix} \tag{21}$$

where

$$
\begin{aligned}
U &= P^{-1} + G^T Y G \\
V &= -G^T Y \\
W &= Y + G^T Y G \\
Y &= (P - G P G^T)^{-1}.
\end{aligned}
$$

The proof of the theorem is in [25].

The theorem allows for a recursion equation to be developed for $\mathscr{D}$ in Eq. (13), which can be used to update the test statistic for a sliding window of data. Using (21), Eq. (13) can be expanded to

$$
\begin{aligned}
\mathscr{D}[t] &= \tilde{x}[t]^\top P_0^{-1} \tilde{x}[t] \qquad (22) \\
&= \Big[\tilde{x}[t-D+1]^\top U \tilde{x}[t-D+1] + \tilde{x}[t-D+2]^\top V^\top \tilde{x}[t-D+1]\Big] \\
&+ \Big[\tilde{x}[t-D+1]^\top V \tilde{x}[t-D+2] + \tilde{x}[t-D+2]^\top W \tilde{x}[t-D+2] + \tilde{x}[t-D+3]^\top V^\top \tilde{x}[t-D+2]\Big] \\
&+ \Big[\tilde{x}[t-D+2]^\top V \tilde{x}[t-D+3] + \tilde{x}[t-D+3]^\top W \tilde{x}[t-D+3] + \tilde{x}[t-D+4]^\top V^\top \tilde{x}[t-D+3]\Big] \\
&\qquad \vdots \\
&+ \Big[\tilde{x}[t-2]^\top V \tilde{x}[t-1] + \tilde{x}[t-1]^\top W \tilde{x}[t-1] + \tilde{x}[t]^\top V^\top \tilde{x}[t-1]\Big] \\
&+ \Big[\tilde{x}[t-1]^\top V \tilde{x}[t] + \tilde{x}[t]^\top Y \tilde{x}[t]\Big]. \qquad (23)
\end{aligned}
$$

Defining

$$
\begin{aligned}
d_1[\tau] &\overset{\triangle}{=} \tilde{x}[\tau]^\top U \tilde{x}[\tau] + \tilde{x}[\tau+1]^\top V^\top \tilde{x}[\tau] \\
d_2[\tau] &\overset{\triangle}{=} \tilde{x}[\tau-1]^\top V \tilde{x}[\tau] + \tilde{x}[\tau]^\top W \tilde{x}[\tau] + \tilde{x}[\tau+1]^\top V^\top \tilde{x}[\tau] \\
d_3[\tau] &\overset{\triangle}{=} \tilde{x}[\tau-1]^\top V \tilde{x}[\tau] + \tilde{x}[\tau]^\top Y \tilde{x}[\tau]
\end{aligned}
$$

the test statistic in (23) can be expressed as

$$
\begin{aligned}
\mathscr{D}[t] &= \tilde{x}[t]^\top P_0^{-1} \tilde{x}[t] \\
&= d_1[t-D+1] \\
&+ d_2[t-D+2] + d_2[t-D+3] + \cdots + d_2[t-2] + d_2[t-1] \\
&+ d_3[t].
\end{aligned}
$$

At the next time step the test statistic is

$$
\begin{aligned}
\mathscr{D}[t+1] &= d_1[t-D+2] \\
&+ d_2[t-D+3] + d_2[t-D+4] + \cdots + d_2[t-1] + d_2[t] \\
&+ d_3[t+1].
\end{aligned}
$$

$$\mathcal{D}[t] = d_1[t-D+1] + d_2[t-D+2] + d_2[t-D+3] + \cdots + d_2[t-2] + d_2[t-1] + d_3[t]$$

$$\cdots$$

$$\mathcal{D}[t+1] = d_1[t-D+2] + d_2[t-D+3] + d_2[t-D+4] + \cdots + d_2[t-1] + d_2[t] + d_3[t+1]$$

Fig. 3 The test statistic at two subsequent time steps. Notice the values that are carried over to the next time step, as indicated by the *arrows*. The test statistic at time $t+1$ is obtained by taking $D[t]$, subtracting the values in *red*, and adding the values in *blue*

Figure 3 illustrates the manner in which values from the test statistic at t are carried over to $t+1$. From this figure, it is clear to see that the complete recursion is

$$\begin{aligned}\mathscr{D}[t+1] = \mathscr{D}[t] &- (d_1[t-D+1] + d_2[t-D+2] + d_3[t]) \\ &+ (d_1[t-D+2] + d_2[t] + d_3[t+1]).\end{aligned} \tag{24}$$

It should be noted that the original window contains D time steps, however, Eq. (24) requires that a window of $D+1$ time steps be maintained. To avoid this change in the window size the recursion equation can be determined in two steps. First, after the hypothesis test at time t an intermediate value for $\mathscr{D}[t]$ is calculated as

$$\mathscr{D}^+[t] = \mathscr{D}[t] - (d_1[t-D+1] + d_2[t-D+2] + d_3[t]).$$

At time $t+1$ the test statistic can then be updated using

$$\mathscr{D}[t+1] = \mathscr{D}^+[t] + (d_1[t-D+2] + d_2[t] + d_3[t+1]).$$

The method can be extended to the case where during the construction of (11) the time difference between subsequent estimation errors is ℓ time steps. Doing so reduces the observed time correlation in the test statistic (which is theoretically zero, but nonzero in practice), which enhances the power of the test [27]. In that case, the recursion for the test statistic becomes

$$\begin{aligned}\mathscr{D}[t+\ell] = \mathscr{D}[t] &- (d_1[t-\ell(D-1)] + d_2[t-\ell(D-2)] + d_3[t]) \\ &+ (d_1[t-\ell(D-2)] + d_2[t] + d_3[t+\ell]).\end{aligned} \tag{25}$$

Note, the recursion requires that the window is slid by ℓ time steps.

2.6 Track Fusion

The final step in the architectures shown in Fig. 1 is to fuse tracks for which the test statistic exceeds the given threshold. Therefore, if H_0 is accepted for a given pair of tracks, $(\hat{x}^{k_a}, \hat{x}^{j_b})$, the objective is to combine or fuse the estimates. The method presented in this paper takes advantage of the fusion properties of information filters [18]. For the steady-state information filter, the information matrix is given by $J = P_e^{-1}$, where P_e is given in Eq. (20), and the information vector is given by $z^{j_b}[t] = P_p^{-1} x^{j_b}[t]$. When measurements are received locally, the information filter is updated using

$$z^{j_b}[t+1] = JAP_p z^{j_b}[t] + C^\top R^{-1} y_{j_b}[t].$$

When the state $\hat{x}^{k_a}$ is to be fused with z^{j_b} then the fusion equation becomes

$$z^{j_b}[t+1] = JAP_p z^{j_b}[t] + C^\top R^{-1} y_{j_b}[t] + R_a^b J \hat{x}^{k_a} + (J-I) d_a^b.$$

3 Simulation and Flight Results

The complete cooperative estimation system shown in Fig. 1 was tested in a tracking scenario that involved two stationary cameras (simulating a hovering scenario), each viewing the tracking area from a different point of view, as shown in Fig. 4.

For this test the position and orientation of the cameras were calculated by determining the mapping of known inertial coordinates to their corresponding locations in the image frame. A foreground detector based on the KLT method [9] was used to produce pixel measurements for each object of interest, which were geolocated in the inertial frame. Note that the inertial frame was specified using a north-east-down (NED) frame of reference. These measurements in the inertial frame were input to R-RANSAC, producing states for each object.

The tracks produced by each object can be seen in the Fig. 5, where the green tracks are from camera V_1, while the cyan tracks are from camera V_2. This figure illustrates the rotational and translational biases that separate the associated tracks.

A window of $N = 10$ state estimates was stored with $\ell = 10$. This window was used to calculate the rotational and translational biases as in Sect. 2.4. The rotation matrix and bias vector were then used to transform the tracks from one vehicle into the reference frame of the other vehicle. Applying the bias estimation to two associated tracks can be seen in Fig. 6.

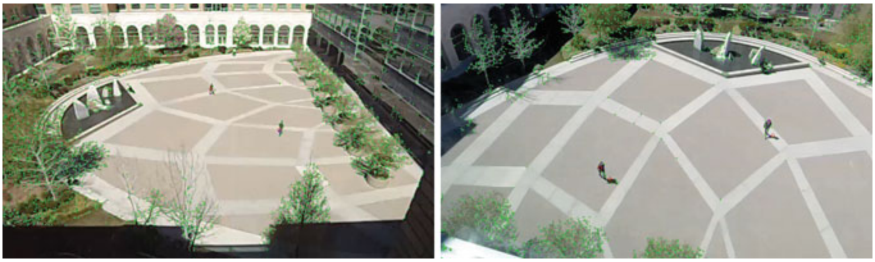


Fig. 4 Tracking scenario with two cameras and two ground objects. Each camera views the tracking area from a different angle

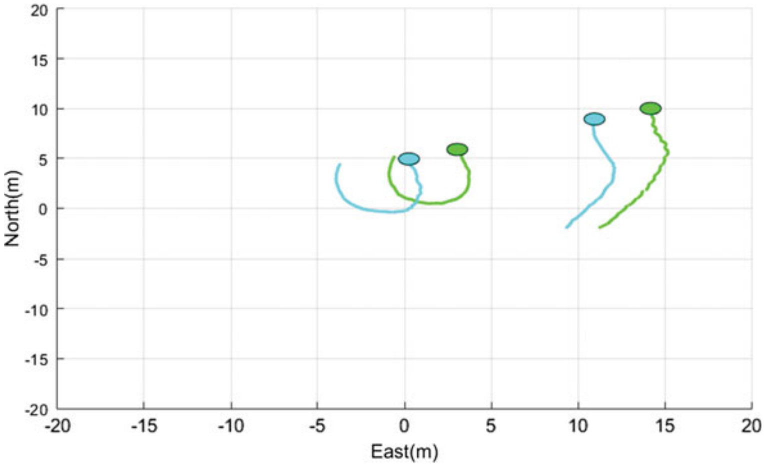


Fig. 5 The two objects from Fig. 4 are geolocated by two cameras. The *circles* represent the object locations at the current time step, while the trails represent the track history. *Green* denotes the tracks from V_1, and *cyan* represents the tracks from V_2. Due to sensor biases, the associated tracks are biased from each other

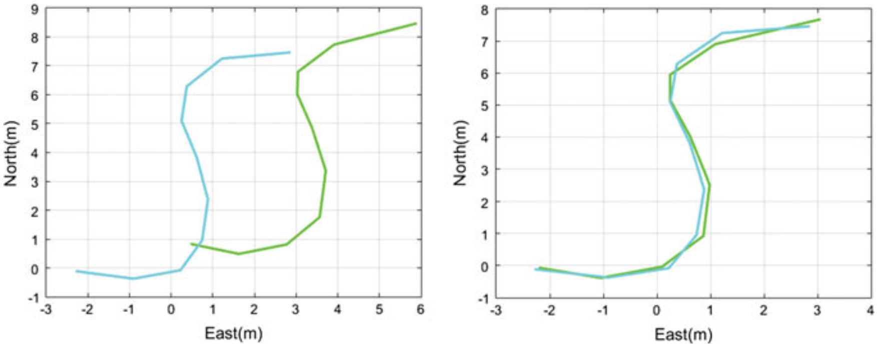


Fig. 6 Two associated tracks, before (*left*) and after (*right*) the bias estimation

It is clear that the bias estimation technique was effective in transforming both tracks into a common reference frame, which is vital for performing the track-to-track association. The application of the bias estimation to two unassociated tracks is shown in Fig. 7. Notice that despite the tracks being unassociated the optimizer still returned a rotation matrix and bias vector that minimized the squared error.

At every time step the window was slid, the bias estimation applied, and the track-to-track association performed. The threshold for the test statistic was based on $\alpha = 0.05$. The results of the track-to-track association over the entire video sequence can be seen in Fig. 8. Note that the track-to-track association was performed between a single track from the first camera with the two tracks from the other camera, which yielded an associated track pair as well as an unassociated track pair. For each

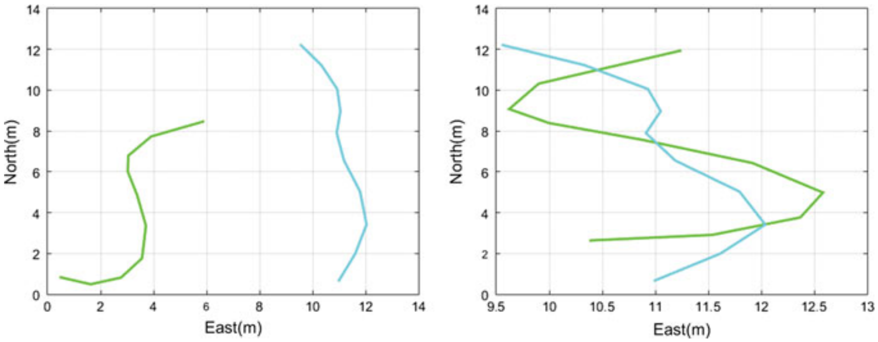


Fig. 7 Two unassociated tracks, before (*left*), and after (*right*) the bias estimation

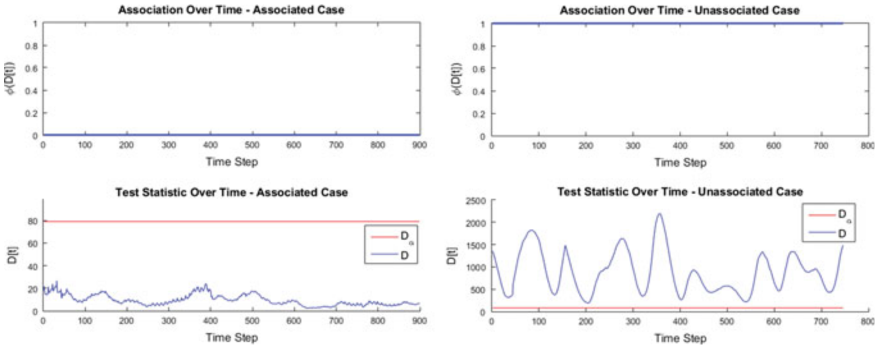


Fig. 8 Track-to-track association between two associated tracks and two unassociated tracks. For each column the *top plots* represent the determined association over time, where a 0 indicates that H_0 was accepted and a 1 indicates that H_0 was rejected. The *bottom plots* show the test statistic over time (*blue*) compared to the threshold (*red*)

column (the left and right columns representing the associated and unassociated cases, respectively) the top plots represent the determined association over time; 0 meaning that H_0 was accepted for the track pair, 1 meaning that H_0 was rejected. The bottom plots show the test statistic over time, compared to the threshold. As seen, over the entire video sequence the track-to-track association algorithm was able to correctly accept and reject H_0 with $P_D = 1$ and $P_R = 1$.

After H_0 was accepted for a given track pair the tracks were fused. The effect of the track fusion can be seen in Fig. 9. The left plot shows two associated tracks that were aligned using the bias estimation technique, with no track fusion. Notice that there were several areas where the two tracks did not fully line up. Over the entire window ($N = 10$, $\ell = 10$) the RMS error of the position states between the two tracks is 0.364 m. On the other hand, the right plot shows the tracks with track fusion applied over the entire window. Here, it can be seen that the fusion caused the two tracks to be more aligned. As a result, the RMS error over the window decreased to 0.037 m.

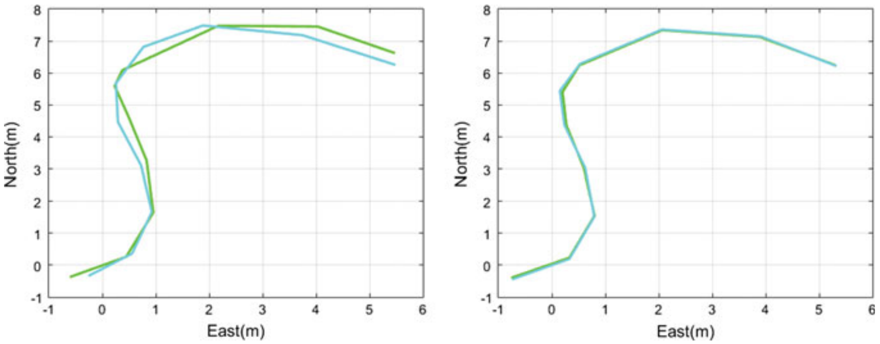


Fig. 9 Results of the track fusion. On the *left* two associated track are aligned, however, no track fusion is performed. It is clear to see that there are areas in which the two tracks did not fully align. On the *right* are the same tracks, however, with track fusion applied over the entire window. The track fusion reduced the differences in the tracks

3.1 Test with Small UAS Platforms

A test was performed with data collected from actual UAS platforms (3DR Y6 multirotor). Again, each UAS viewed the tracking area from a different angle (see Fig. 10). However, unlike the previous test the UAS platforms were not stationary. The vehicle states were provided by the 3DR Pixhawk autopilot. Moreover, each vehicle was equipped with a 3-axis brushless gimbal that was controlled using the BaseCam SimpleBGC 32-bit gimbal controller. Note that the video sequence contained a single object, thus, the data was used to validate the method under the assumption of H_0 only.

The results from the test are summarized in Fig. 11. Overall the algorithm was effective in associating the two tracks from the different vehicles and had a probability of detection $P_D = 1.0$. These results affirm the effectiveness of the method in the presence of actual UAS sensor biases and noise.

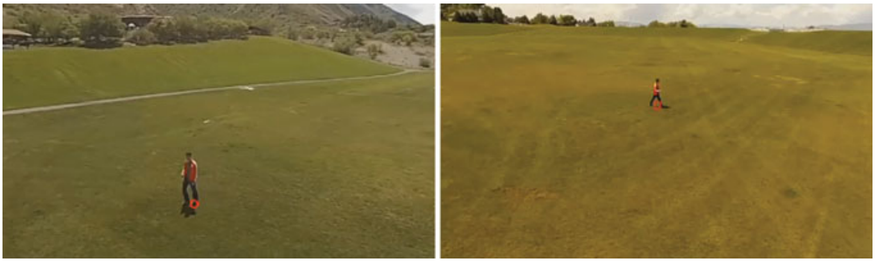
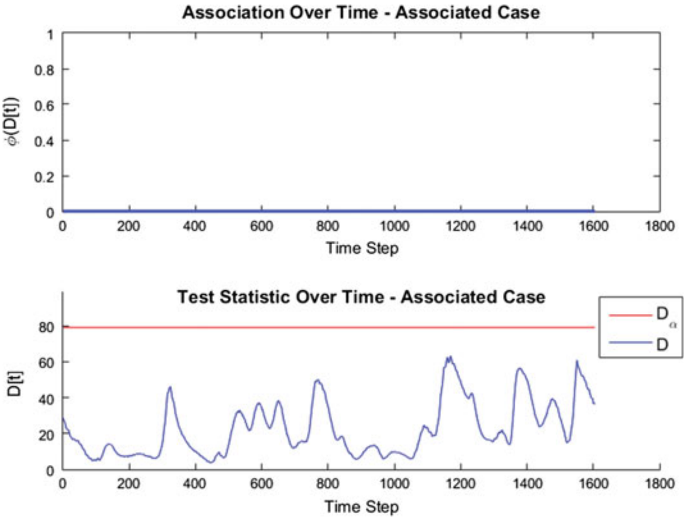


Fig. 10 Tracking scenario with two cameras and one ground object of interest. Each camera is mounted to a UAS platform that is maneuvering and views the tracking area from a different angle. The *red circle* indicates the pixel measurement that is used to geolocate the object of interest


Association Over Time - Associated Case

Test Statistic Over Time - Associated Case

Fig. 11 Track-to-track association between two associated tracks. For each column the *top plots* represent the determined association over time, where a 0 indicates that H_0 was accepted and a 1 indicates that H_0 was rejected. The *bottom plots* show the test statistic over time (*blue*) compared to the threshold (*red*)

4 Conclusions

This paper presents a complete method for cooperative estimation of ground targets using a vision-based object tracking system. The method estimates and accounts for both translational and rotational biases between tracks, and performs a hypothesis test to determine the track-to-track association. The test statistic is calculated using a window of estimates and follows a chi-squared distribution. The correlation between associated tracks, and the correlation in time of the estimation errors, is accounted for in the calculation of the covariance. This paper also presents a track fusion technique, which accounts for the estimated biases.

The complete system is demonstrated in actual tracking scenarios. The results show that the bias estimation is effective in aligning associated tracks from different vehicles. Moreover, the track-to-track association method is able to make the proper assignments with a high probability of detection and rejection. Lastly, the track fusion technique decreases the relative estimation error between associated tracks.

Acknowledgements This work has been funded in part by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation-sponsored industry/university cooperative research center (I/UCRC) under NSF Award No. IIP-1161036 along with significant contributions from C-UAS industry members, and in part by AFRL grant FA8651-13-1-0005.

References

1. Bar-Shalom, Y.: On the track-to-track correlation problem. IEEE Autom. Control **26**(2), 571–572 (1981)
2. Bar-Shalom, Y., Fortmann, T.E.: Tracking and Data Association. Academic Press, New York (1988)
3. Bar-Shalom, Y., Daum, F., Huang, J.: The probabilistic data association filter. IEEE Control Syst. Mag. **26**(9), 82–100 (2009)
4. Barber, D.B., Redding, J.D., McLain, T.W., Beard, R.W., Taylor, C.N.: Vision based target geo-location using a fixed-wing miniature air vehicle. J. Intell. Robot. Syst. **47**(4), 361–382 (2006)
5. Beard, R.W., McLain, T.W.: Small Unmanned Aircraft: Theory and Practice. Princeton University Press, Princeton (2012)
6. Blackman, S.S.: Multiple hypothesis tracking for multiple target tracking. IEEE Aerosp. Electron. Syst. Mag. **19**(1), 5–18 (2004)
7. Campbell, M.E., Wheeler, M.: A vision based geolocation tracking system for UAVs. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit. Keystone, Colorado (2006)
8. Casbeer, D.W., Kingston, D.B., Beard, R.W., McLain, T.W., Li, S.-M., Mehra, R.: Cooperative forest fire surveillance using a team of small unmanned air vehicles. Int. J. Syst. Sci. **37**(6), 351–360 (2006)
9. DeFranco, P.C.: Detecting and Tracking Moving Objects from a Small Unmanned Air Vehicle. Master's thesis, Brigham Young University, Provo, Utah (2015)
10. Dobrokhodov, V.N., Kaminer, I.I., Jones, K.D.: Vision-based tracking and motion estimation for moving targets using small UAVs. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit. Keystone, Colorado (2006)
11. Frew, E.W.: Sensitivity of cooperative target geolocation to orbit coordination. J. Guid. Control Dyn. **31**(4), 1028–1040 (2008)
12. He, Z., Xu, J.-X., Lum, K.-Y.: Targets tracking by UAVs in an urban area. In: IEEE International Conference on Control and Automation (ICCA), pp. 1834–1838. Hangzhou, China (2013)
13. Holt, R.S., Egbert, J.W., Bradley, J.M., Beard, R.W., Taylor, C.N., McLain, T.W.: Forest fire monitoring using multiple unmanned air vehicles. In: Eleventh Biennial USDA Forest Service Remote Sensing Applications Conference. Salt Lake City (2006)
14. Ingersoll, J.K.: Vision Based Multiple Target Tracking Using Recursive RANSAC. Master's thesis, Brigham Young University (2015)
15. Ingersoll, K., Niedfeldt, P.C., Beard, R.W.: Multiple target tracking and stationary object detection using recursive-RANSAC and tracker-sensor feedback. In: Proceedings of the International Conference on Unmanned Air Vehicles. Denver, CO (2015)
16. Kamal, A.T., Bappy, J.H., Farrell, J.A., Roy-Chowdhury, A.K.: Distributed multitarget tracking and data association in vision networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(7), 1397–1410 (2016)
17. Kumar, R., Sawhney, H., Samarasekera, S., Hsu, S., Tao, H., Guo, Y., Hanna, K., Pope, A., Wildes, R., Hirvonen, D., Hansen, M., Burt, P.: Aerial video surveillance and exploitation. Proc. IEEE **89**(10), 1518–1539 (2001)
18. Mutambara, A.G.O.: Decentralized Estimation and Control for Multisensor Systems. CRC Press, Boca Raton (1998)
19. Niedfeldt, P.C.: Recursive-RANSAC: A Novel Algorithm for Tracking Multiple Targets in Clutter. Ph.D. thesis, Brigham Young University, Provo (2014)
20. Niedfeldt, P.C., Beard, R.W.: Recursive RANSAC: multiple signal estimation with outliers. In: Proceedings of the 9th Symposium on Nonlinear Control Systems, pp. 430-435. Toulouse, France (2013)
21. Niedfeldt, P.C., Beard, R.W.: Multiple target tracking using recursive RANSAC. In: Proceedings of the American Control Conference, pp. 3393–3398. Portland, OR (2014)

22. Niedfeldt, P.C., Beard, R.W.: Convergence and complexity analysis of recursive-RANSAC: a new multiple target tracking algorithm. IEEE Trans. Autom. Control **61**(2), 456–461 (2016)
23. Pachter, M., Ceccarelli, N., Chandler, P.R.: Vision-based target geo-location using camera equipped MAVs. In: Proceedings of the IEEE Conference on Decision and Control. New Orleans, LA (2007)
24. Ruggles, S., Clark, J., Franke, K.W., Wolfe, D., Reimschiissel, B., Martin, R.A., Okeson, T.J., Hedengren, J.D.: Comparison of SfM computer vision point clouds of a landslide derived from multiple small UAV platforms and sensors to a TLS based model. J. Unmanned Veh. Syst. (2016). doi:10.1139/juvs-2015-0043
25. Sakamaki, J.Y.: Cooperative Estimation for a Vision Based Target Tracking System. Master's thesis, Brigham Young University, Provo, Utah (2016)
26. Scharf, L.L.: Statistical Signal Processing. Prentice Hall, Englewood Cliffs (1990)
27. Tian, X., Bar-Shalom, Y.: Sliding window test vs. single time test for track-to-track association. In: Proceedings of the 11th International Conference on Information Fusion, FUSION 2008 (2008). ISBN 9783000248832. doi:10.1109/ICIF.2008.4632281
28. Vo, B.N.: Ma, W.K.: A closed-form solution for the probability hypothesis density filter. In: 2005 7th International Conference on Information Fusion, FUSION 2, pp. 856–863 (2005). ISBN 0780392868. doi:10.1109/ICIF.2005.1591948
29. Whang, I.H., Dobrokhodov, V.N., Kaminer, I.I., Jones, K.D.: On vision-based tracking and range estimation for small UAVs. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit. San Francisco, CA (2005)