Cooperative Vision Based Estimation and Tracking Using Multiple UAVs

Brett Bethke, Mario Valenti, and Jonathan How

Massachusetts Institute of Technology, Cambridge, MA
{bbethke,valenti,jhow}@mit.edu

Abstract. Unmanned aerial vehicles (UAVs) are excellent platforms for detecting and tracking objects of interest on or near the ground due to their vantage point and freedom of movement. This paper presents a cooperative vision-based estimation and tracking system that can be used in such situations. The method is shown to give better results than could be achieved with a single UAV, while being robust to failures. In addition, this method can be used to detect, estimate and track the location and velocity of objects in three dimensions. This real-time, vision-based estimation and tracking algorithm is computationally efficient and can be naturally distributed among multiple UAVs. This chapter includes the derivation of this algorithm and presents flight results from several real-time estimation and tracking experiments conducted on MIT's Real-time indoor Autonomous Vehicle test ENvironment (RAVEN).

Keywords: Cooperative multi UAV vision tracking estimation.

1 Introduction

Unmanned Aerial Vehicles (UAVs) have attracted significant interest in recent years. Due to improvements in embedded computing, communications, and sensing technologies, UAVs have become increasingly capable of carrying out sophisticated tasks. Because UAVs lack a human occupant and are generally simpler and less expensive than their manned counterparts, they are well suited to perform a wide range of "dull, dirty and/or dangerous" missions. Examples of such missions include traffic monitoring in urban areas, search and rescue operations, military surveillance, and border patrol [5].

For many mission scenarios, the deployment of video cameras onboard UAVs is of particular interest due to the richness of information and real-time situational assessment capabilities that can be provided by the video stream. Researcher have used onboard cameras for remote detection of forest fires [3,6,7]. In addition, a number of researchers have used vision-based techniques for object detection, tracking and surveillance [1,2,10]. The measurements of the target location are inherently nonlinear in the single-vehicle case because the observed state variables are measured angles to the target. As such, numerous researchers have investigated using nonlinear estimators, such as Extended Kalman Filters and Unscented Kalman Filters [4,9], to determine the target state. In addition, observations from multiple vantage points are required to provide depth

perception and to obtain a good estimate of the target's position and orientation. Geometrically, using only a single observation (via a single camera), the UAV can only determine a ray along which the target lies. Therefore, the UAV must be maneuvered around the target to provide multiple vantage points to gain a more accurate position estimate of the vehicle. However, if the object of interest is moving, the UAV may not be able to complete the necessary maneuvers to gain a more accurate estimate.

In this chapter, we present a vision-based estimation and tracking algorithm that exploits cooperation between multiple UAVs in order to provide accurate target state estimation and allow good tracking of the target without the need for a single vehicle to execute maneuvers to gain better vantage points. The method uses an optimization technique to combine the instantaneous observations of all UAVs, allowing for very rapid estimation. Furthermore, the algorithm can be naturally distributed among all participating UAVs with very modest communication bandwidth requirements and is computationally efficient, making it well suited to implementation on real-time applications. The vision processing is done in a manner designed to be robust to noise in the video stream, which is often present, especially in applications where the video signal is wirelessly transmitted. Flight results from several estimation and tracking experiments conducted on MIT's Real-time indoor Autonomous Vehicle test ENvironment (RAVEN) are presented [8].

2   Vision Based Tracking and Estimation: A Cooperative Approach

Using multiple UAVs in the real-time, vision-based detection, estimation and tracking problem is advantageous for a number of reasons. First, multiple UAVs provide redundancy, allowing for continued tracking even when individual vehicles experience failures. Second, the presence of obstructions in the environment may temporarily block the field of view of a UAV as it attempts to observe the target. Using multiple UAVs with different lines of sight increases the probability that the target will remain observable to the group of UAVs even when individual vehicles' lines of sight are blocked. Third, because more observations are available at a given time, multiple UAVs working together can estimate the target's state more accurately than a single UAV could.

In addition, the cooperative UAV vision tracking problem can be reformulated as a linear estimation problem. Using the observed bearings of the target from each UAV, an estimate of the absolute target position can be obtained by minimizing the errors in distance from the estimate to each measurement. This estimate is then used as a measurement input to a simple linear Kalman filter that uses the target location as the state variables {x,y,z}.

The statement of the multi-UAV vision-based detection, estimation and tracking problem is as follows. Assume that there are n UAVs, each equipped with a camera. The location of each UAV is given by

$$\mathbf{x}_i = \hat{\mathbf{x}}_i + \delta\mathbf{x}_i, \ \ \mathbf{x}_i \in \mathbb{R}^3 \tag{1}$$

Fig. 1. Five vehicle coordinated flight test on MIT's RAVEN [8]

where $\hat{\mathbf{x}}_i$ is the estimated position of the UAV (as given by the UAV's onboard navigation sensors), and $\delta\mathbf{x}_i$ is a random variable that captures the uncertainty in the UAV's position. The distribution of $\delta\mathbf{x}_i$ is assumed to be known.

From each UAV, there is a director vector to the target

$$\mathbf{d}_i = \hat{\mathbf{d}}_i + \delta\mathbf{d}_i, \quad \mathbf{d}_i \in \mathbb{R}^3 \tag{2}$$

where again $\hat{\mathbf{d}}_i$ is the estimated (unit-length) direction vector generated by the vision system, described below, and $\delta\mathbf{d}_i$ represents uncertainty in the direction vector (i.e., uncertainty in the precise direction in which the camera is pointing). Assume that

$$\delta\mathbf{d}_i^T \hat{\mathbf{d}}_i = 0 \tag{3}$$

This assumption is reasonable given that $\hat{\mathbf{d}}_i$ is most naturally characterized by uncertainty in the angles from the camera to the target, so that $\mathbf{d}_i^T \mathbf{d}_i \approx 1$. Again, assume that the distribution of $\delta\mathbf{d}_i$ is known. Finally, a weight w_i is associated with each UAV's estimate. This weight may be used to account for differences in the quality of each UAV's estimate (i.e, differences in video quality). Note that given $\mathbf{x}_i$ and $\mathbf{d}_i$, the target must lie along the ray

$$\mathbf{l}_i(\lambda_i) = \mathbf{x}_i + \lambda_i\mathbf{d}_i, \quad \lambda_i \geq 0 \tag{4}$$

In order to solve this problem, the true position of the object, $\mathbf{q}$, must be estimated given the set of all measurements $\{\hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{d}}_\mathbf{i} : i = 1, \ldots, n\}$. This estimate $\mathbf{q}$ should minimize the error

$$E(\mathbf{q}) = \sum_{i=1}^{n} w_i h_i(\mathbf{q}) \tag{5}$$

where $h_i(\mathbf{q})$ is the square of the minimum distance from $\mathbf{q}$ to the ray $\mathbf{l}_i(\lambda_i)$:

$$h_i(\mathbf{q}) = \min_{\lambda_i} ||\mathbf{q} - \mathbf{l}_i(\lambda_i)||^2 = \min_{\lambda_i} ||\mathbf{q} - (\mathbf{x}_i + \lambda_i\mathbf{d}_i)||^2 \tag{6}$$

Minimizing $h_i(\mathbf{q})$ with respect to λ_i yields the result

$$h_i(\mathbf{q}) = \mathbf{q}^T\mathbf{q} - 2\mathbf{q}^T\mathbf{x}_i + \mathbf{x}_i^T\mathbf{x}_i - (\mathbf{d}_i^T\mathbf{q} - \mathbf{d}_i^T\mathbf{x}_i)^2 \tag{7}$$

Substituting this result into Eq. 5 and minimizing $E(\mathbf{q})$ with respect to $\mathbf{q}$ yields the equation that the optimal estimate must satisfy:

$$\mathcal{A}\mathbf{q}^\star = \mathbf{b} \tag{8}$$

where

$$\mathcal{A} = \sum_{i=1}^n w_i \left(I - \mathbf{d}_i\mathbf{d}_i^T\right) \tag{9}$$

$$\mathbf{b} = \sum_{i=1}^n w_i \left(\mathbf{x}_i - (\mathbf{x}_i^T\mathbf{d}_i)\mathbf{d}_i\right) \tag{10}$$

However, $\mathcal{A}$ and $\mathbf{b}$ cannot be calculated by the algorithm directly because only the the noisy measurements $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{d}}_i$ are known. To compensate for these errors, $\mathcal{A}$ and $\mathbf{b}$ are expanded by substituting Eqs. 1 and 2 into Eqs. 9 and 10. After dropping second-order terms and grouping the known and unknown terms, the equations become

$$\mathcal{A} = \hat{\mathcal{A}} + \delta\mathcal{A} \tag{11}$$
$$\mathbf{b} = \hat{\mathbf{b}} + \delta\mathbf{b} \tag{12}$$

where

$$\hat{\mathcal{A}} = \sum_{i=1}^n w_i \left(I - \hat{\mathbf{d}}_i\hat{\mathbf{d}}_i^T\right) \tag{13}$$

$$\delta\mathcal{A} = -\sum_{i=1}^n w_i(\delta\mathbf{d}_i\hat{\mathbf{d}}_i^T + \hat{\mathbf{d}}_i^T\delta\mathbf{d}_i) \tag{14}$$

$$\hat{\mathbf{b}} = \sum_{i=1}^n w_i \left(\hat{\mathbf{x}}_i - (\hat{\mathbf{x}}_i^T\hat{\mathbf{d}}_i)\hat{\mathbf{d}}_i\right) \tag{15}$$

$$\delta\mathbf{b} = \sum_{i=1}^n w_i(\delta\mathbf{x}_i - (\hat{\mathbf{x}}_i^T\delta\mathbf{d}_i)\hat{\mathbf{d}}_i - (\delta\mathbf{x}_i^T\hat{\mathbf{d}}_i)\hat{\mathbf{d}}_i - (\hat{\mathbf{x}}_i^T\hat{\mathbf{d}}_i)\delta\mathbf{d}_i) \tag{16}$$

Note that $\hat{\mathcal{A}}$ and $\hat{\mathbf{b}}$ are known terms, because they involve only quantities that are measured directly. $\delta\mathcal{A}$ and $\delta\mathbf{b}$ are random variables because they involve the uncertain quantities $\delta\mathbf{x}_i$ and $\delta\mathbf{d}_i$. The optimal estimate can now be written as

$$\mathbf{q}^\star = \mathcal{A}^{-1}\mathbf{b} = (\hat{\mathcal{A}} + \delta\mathcal{A})^{-1}(\hat{\mathbf{b}} + \delta\mathbf{b}) \tag{17}$$

We assume that the error terms are small ($\delta\mathcal{A} \ll \hat{\mathcal{A}}$). Expanding the matrix inverse function in a Taylor series around $\hat{\mathcal{A}}$ gives

$$(\hat{\mathcal{A}} + \delta\mathcal{A})^{-1} \approx \hat{\mathcal{A}}^{-1} - \hat{\mathcal{A}}^{-1}\delta\mathcal{A}\hat{\mathcal{A}}^{-1} \tag{18}$$

Thus, Eq. 17 becomes

$$\mathbf{q}^\star = \hat{\mathcal{A}}^{-1}\hat{\mathbf{b}} + \hat{\mathcal{A}}^{-1}\delta\mathbf{b} - \hat{\mathcal{A}}^{-1}\delta\mathcal{A}\hat{\mathcal{A}}^{-1}\hat{\mathbf{b}} - \hat{\mathcal{A}}^{-1}\delta\mathcal{A}\hat{\mathcal{A}}^{-1}\delta\mathbf{b} \tag{19}$$

$$\approx \hat{\mathbf{q}}^\star + \hat{\mathcal{A}}^{-1}\delta\mathbf{b} - \hat{\mathcal{A}}^{-1}\delta\mathcal{A}\hat{\mathcal{A}}^{-1}\hat{\mathbf{b}} \tag{20}$$

where

$$\hat{\mathbf{q}}^\star = \hat{\mathcal{A}}^{-1}\hat{\mathbf{b}} \tag{21}$$

is the optimal estimate which can be calculated from the measurements. The error $\delta\mathbf{q}^\star$ in the estimate is

$$\delta\mathbf{q}^\star = \hat{\mathcal{A}}^{-1}\delta\mathbf{b} - \hat{\mathcal{A}}^{-1}\delta\mathcal{A}\hat{\mathcal{A}}^{-1}\hat{\mathbf{b}} \tag{22}$$

Since the probability distributions of the random variables $\delta\mathbf{x}_i$ and $\delta\mathbf{d}_i$ are known, the covariance of $\delta\mathbf{q}^\star$ can be calculated. This covariance is needed in order to implement the Kalman filter, discussed below.

Eq. 21 demonstrates that the optimal estimate $\hat{\mathbf{q}}^\star$ can be computed in time that is *linear* in the number of measurements to the object, n. $\hat{\mathcal{A}}$ and $\hat{\mathbf{b}}$ can be constructed in linear time since they are sums over all rays. Once $\hat{\mathcal{A}}$ and $\hat{\mathbf{b}}$ are known, Eq. 21 can be solved in constant time by inverting the 3 x 3 matrix $\hat{\mathcal{A}}$. Since the entire process runs in linear time with respect to n, this method is very computationally efficient. Note that if there is only a single vehicle, $n = 1$, the matrix $\hat{\mathcal{A}}$ is singular and Eq. 21 cannot be solved. In this case, a single vehicle would have to make an additional assumption about the location of the vehicle, such that it was located on the ground ($z = 0$), in order to calculate a solution. In all other cases, however, $\hat{\mathcal{A}}$ is invertible as long as the observed direction vectors $\hat{\mathbf{d}}_i$ are not all parallel to each other. As long as the observation points $\hat{\mathbf{x}}_i$ are not the same, which cannot happen since the UAVs cannot occupy the same physical point in space, a solution can always be found.

Once the estimate $\hat{\mathbf{q}}^\star$ is known, it can be used as the measurement into a simple linear Kalman filter based on the assumed dynamics of the target vehicle [1]. This paper uses a system model with state vector

$$\mathbf{X} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \tag{23}$$

The discrete time system dynamics are then given by

$$\mathbf{X}_{k+1} = A\mathbf{X}_k + \mathbf{v}_k \tag{24}$$

$$\mathbf{Y}_k = \hat{\mathbf{q}}^\star = C\mathbf{X}_k + \delta\mathbf{q}^\star \tag{25}$$

where $\mathbf{v}_k$ is the process noise and $\delta\mathbf{q}^\star$ is the measurement noise. The process noise covariance is assumed to be known, and the covariance of $\delta\mathbf{q}^\star$ is found as discussed above. A and C are given by

$$A = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{26}$$

where Δt is the sampling rate of the filter. Using these dynamics, a linear Kalman filter can be easily designed and implemented. This filter can be run on each UAV; the only requirement is that the UAVs communicate their locations $\mathbf{x}_i$ and estimation directions $\mathbf{d}_i$ to each other. Since each of these quantities is a three dimensional vector, this method requires only six numbers to be transmitted by each UAV, making it well suited for environments where communication bandwidth is limited.

3   Hardware and Software Setup

A real-time, vision-based tracking system was implemented on the quadrotor platform. The motivation behind implementing the vision system was to allow the quadrotors to carry their own sensor payloads and make decisions based on their own sensor data instead of relying upon artificially synthesized sensor data, allowing for a more realistic overall hardware platform.

The hardware setup consists of two Draganfly quadrotors outfitted with a Draganfly SAVS wireless camera. The SAVS camera broadcasts to a diversity receiver on the ground. The receiver is connected to a PC with a LifeView FlyVideo 3000FM video capture card. The Intel OpenCV video capture and processing library is used to interface with the video card and provide a software API for accessing and processing the images.

3.1   Persistent Object Filter

Given the low-power nature of the wireless camera's transmitter, the presence of RF noise, and other impediments to receiving a clear video signal, it is important that the vision tracking system be able to function even when the received video stream is noisy. Experimental data from several wireless cameras shows that noise is often present in the processed images from the camera (as shown in Figure 2).

An expected but important characteristic of the noise seen in the video stream is that it is highly uncorrelated from one frame to the next (as shown in Figure 2). Based on this observation, a filtering algorithm was designed to extract the features of the images that are persistent across multiple frames while rejecting the transient noise components. The key component of the algorithm is a dynamic list P of objects which have been seen in previous frames. By comparing the objects in P with the objects that are in the current frame, denoted by C, the algorithm can decide which objects in C have appeared before, and these objects are given higher weight in the filter.

A detailed description of the algorithm follows.

```
1    set C = ∅; (The set of objects in the current frame)
2    set P = ∅; (The set of persistent objects)
3    while true do:
4        for p in P do:
5            p.f = false; (Mark p as "not found")
```

Fig. 2. Two images taken within one second of each other showing noise in the video stream

```
6        end for;
7        F = getCurrentFrame(); (Get the current image frame)
8    G=preprocessFrame(F); (Downsample image and convert to binary image)
9        C = findObjectsInFrame(G); (Find objects in the current frame)
10       for c in C do:
11           f = false;
11           for p in P do:
12               if (‖c.x − p.x‖ < ε_x) and (|c.A − p.A| < ε_A) do:
                 (Determine whether c is similar to p in terms of location in the
                 image x and area a)
13                   p.f = true; (Mark p as "found")
14                   p.n = p.n + 1; (Add 1 to the running frame count of p)
15                   f = true;
16                   break; (Since p is determined, there is no need to continue
                     examining the other elements of P)
17           end for; (Ends for p in P)
18           if f == false do:
19               P.push(c); (If c is not found anywhere in P, append c to P)
20           end if;
21       end for; (Ends for c in C)
22       for p in P do:
23           if p.f == false do: (If p was not found in this frame...)
24               p.n=p.n − 1; (...subtract 1 from the running frame count of p)
25           end if;
26           if p.n == 0 do: (If the frame count of p is zero...)
27               P.pop(p); (...remove p from P)
28           end if;
29       end for;
30   end while;
```
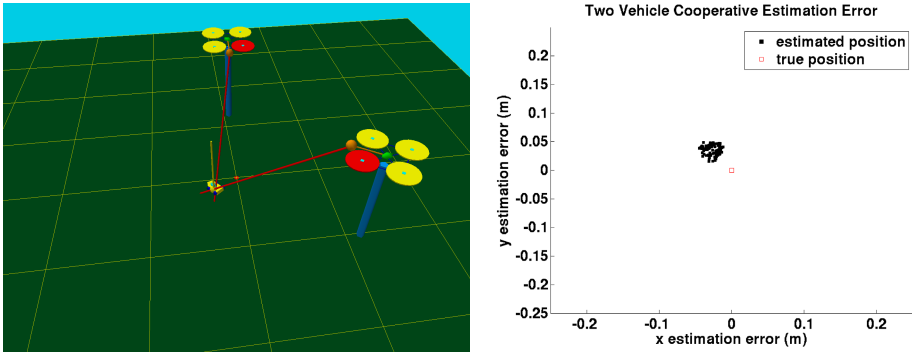
Fig. 3. Left: First experiment configuration. Right: First experiment results. Mean estimation error: x = -0.0265 m, y = 0.0368 m. Standard deviation: σ_x = 0.0082 m, σ_y = 0.0083 m

The current implementation detects objects based on image intensity, although other schemes for object recognition are possible as well. Continuous dark areas ("blobs") in the image are assumed to be objects of interest. The preprocessing step in Line 7 of the algorithm downsamples the image (this is to allow faster processing of the image) and applies a threshold filter in order to convert it to a binary image. This binary image is then passed to a function in Line 8 that detects the centers of each object in the image.

Once the objects are found, the filtering algorithm is applied. The output of the filter is a list of persistent objects in the video stream. The positions of these objects can then be estimated.

4   Results

A number of multi-vehicle vision tracking experiments were conducted to verify the performance of the vision estimation and tracking system. Two vehicles equipped with Draganfly SAVS camera systems were used as the test platform. A small, radio-controlled truck (shown in Figure 2) was used as a target.

In the first experiment, the goal of the UAVs was to hover with the target in the field of view and cooperatively estimate its position. Figure 3 shows the configuration of the experiment using a screenshot from a real-time 3D data visualization tool that was developed for use with the system. The visualization tool shows the locations of the UAVs and target, as well as the rays from each UAV's camera (red lines) and the cooperatively estimated position of the target (yellow sphere with vertical yellow line). Data is displayed in real-time as the system is running, and can also be logged and played back later for analysis purposes.

Results of the first experiment are shown in Figure 3. The scatter plot shows the estimated X-Y positions of the target over a period of about 60 secs of flight. Note that during this time, the UAVs were subject to disturbances in
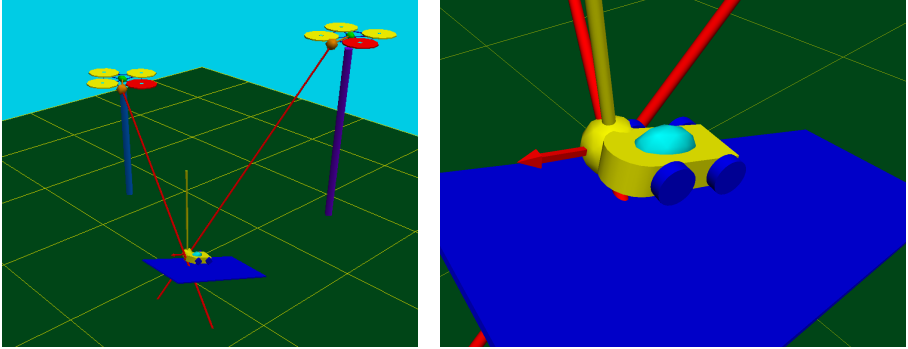
Fig. 4. Second experiment configuration

the form of wind (from ambient air currents as well as propeller wash from the neighboring UAV) which caused them to execute small corrective pitch and roll maneuvers to maintain their assigned hover locations. In the presence of these disturbances, the estimated position of the target remained within 0.04 m of its true location, an excellent result considering that the target itself is over 0.2 m in length. Note that the estimator also gives the Z position of the target (not plotted in Figure 3), and the estimated Z position was also within 0.04m of the true location.

The second experiment shows the advantage of cooperation in the vision estimation problem. In this experiment, two UAVs hovered near a target vehicle which drove up a ramp at constant speed. The goal of the vehicles was to estimate the position and velocity of the target as it moved up the ramp, using both the cooperative estimation method and the noncooperative, single vehicle estimation method. Figure 4 shows the experiment setup, including the ramp and ground vehicle. Note that the arrow protruding from the estimated position of the target shows the estimated velocity, which is clearly aligned with the slope of the ramp.

Figure 5 shows that as the target vehicle moves up the ramp, the estimated error for the noncooperative estimation technique grows due to the fact that the target vehicle is farther from the ground. Without another UAV's perspective, the single UAV is unable to accurately determine the position of the target. Furthermore, the single UAV is unable to determine the velocity of the target well, since it is unable to estimate the ground object's z-axis velocity. Meanwhile, the cooperative estimates for target position and velocity remain very accurate in all three dimensions.

The third experiment incorporated active tracking into the detection and estimation problem. Two UAVs were commanded to detect a ground vehicle and estimate its position. The estimated position was then passed to a tasking system that generated waypoints for the UAVs to follow. The waypoints were chosen in a way that was designed to keep the ground vehicle in the field of view of the UAVs, thus enabling them to continue tracking. In this case, the tasking
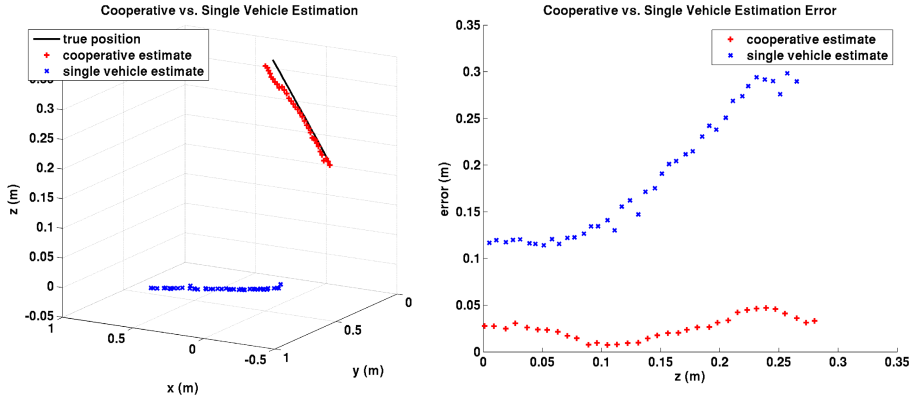
Fig. 5. Second experiment results: cooperative and noncooperative estimated trajectory

system kept one UAV two meters south of the ground vehicle and the other UAV two meters west of the ground vehicle.

Results of the third experiment are shown in Figure 6. The results show that the UAVs were able to estimate the position of the ground vehicle well (within about 5cm) even while they were moving cooperatively in order to keep the vehicle in the field of view of both UAVs. Note that in this experiment, the ground vehicle moved enough that it would have been outside the field of view of both UAVs at times had the UAVs not moved along with it.
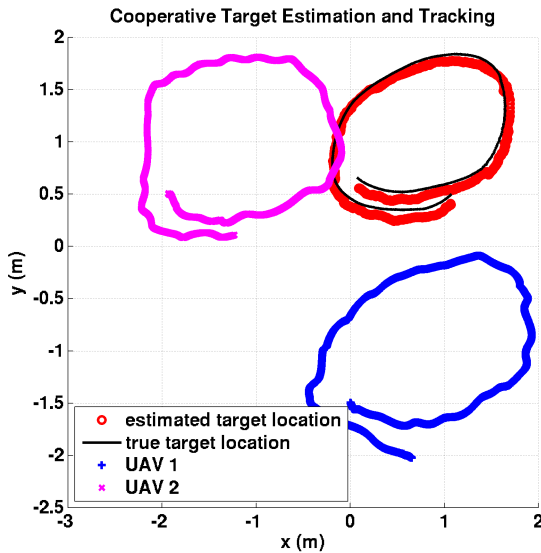

Fig. 6. Results of cooperative estimation and tracking

5   Conclusions

This paper has demonstrated a cooperative target estimation algorithm that is well-suited for real-time implementation and can be distributed among the cooperating vehicles. This approach to target tracking may be advantageous in terms of performance and robustness when multiple vehicles are available to perform the tracking. Flight results indicate that the algorithm performs well in tracking both stationary and maneuvering targets.

References

1. N. Gordon B. Ristic, S. Arulampalam. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA, 2004.
2. C. Sharp, O. Shakernia, and S. Sastry. A Vision System for Landing an Unmanned Aerial Vehicle. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, volume 2, pages 1720–1727, 2001.
3. D. Casbeer, S. Li, R. Beard, R. Mehra, T. McLain. Forest Fire Monitoring With Multiple Small UAVs. Porland, OR, April 2005.
4. E. Wan, R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Alta, Canada, October 2000.
5. G. Goebel. In the Public Domain: Unmanned Aerial Vehicles. http://www.vectorsite.net/twuav.html, April 2006.
6. L. Merino, F. Caballero, J. R. Martinez de Dios, A. Ollero. Cooperative Fire Detection using Unmanned Aerial Vehicles. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.
7. L. Merino, F. Caballero, J. R. Martinez de Dios, J. Ferruz, A. Ollero. A Cooperative Perception System for Multiple UAVs: Application to Automatic Detection of Forest Fires. *Journal of Field Robotics*, 23:165–184, 2006.
8. M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron. Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006.
9. S. Julier, J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.
10. T. McGee, R. Sengupta, K. Hedrick. Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.