The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Cooperative Target Localization Using Heterogeneous Unmanned Ground and Aerial Vehicles

Chad Hager, Dimitri Zarzhitsky, Hyukseong Kwon, and Daniel Pack

Abstract— **This paper describes our on-going efforts toward developing heterogeneous, cooperative systems technologies. In particular, we present the role of unmanned mobile ground systems (robots) in a heterogeneous sensor network, consisting of two unmanned aircraft, a mobile ground robot, and a set of four stationary ground sensors, performing an intelligence, surveillance, and reconnaissance (ISR) mission. The unmanned mobile ground robot is equipped with an infrared (IR) sensor, the aircraft and the stationary ground sensors use optical cameras and radio frequency (RF) detectors, respectively. The primary responsibility of the mobile ground robot is to verify the identity of a target based on its IR signature. In addition, the mobile ground robot also assists with the sensor network's overall target localization estimation efforts by sharing its IR sensor-based target location measurements with members of the sensor network. Our analysis and field experiments demonstrated scalability and effectiveness of our approach.**

I. INTRODUCTION

Ever since the military and civilian use of unmanned aircraft with on-board sensing capabilities became an integral part of surveillance, reconnaissance, and sometimes combat missions, it has become apparent that the next horizon of the technology push requires the use of a network of multiple unmanned platforms, including aerial, ground, surface, and underwater vehicles. This paper describes our efforts to advance the cooperative, heterogeneous systems technologies as we solve some of the critical mobile sensor network problems. Here, we focus on the role of ground mobile sensors as they contribute to sensor network capabilities. The motivation for our work comes from the current operational needs identified by deployed soldiers in theaters: an increasing number of military and humanitarian missions rely on the capabilities rendered by a team of autonomous, cooperating robots. The challenges introduced by such systems require distributed algorithms for coordinated sensing and control of heterogeneous mobile systems, since it is impractical to have a centralized control architecture as the team size increases. We present an asynchronous method for sensor exploitation and fusion of data collected by multiple systems, and a means to reliably share pertinent information among team members.

A number of methodologies discussed in the literature address cooperative target localization using aircraft and ground vehicles. Vidal *et al.* (2002) developed an evading target tracking system using collaboration of one unmanned aerial vehicle (UAV) and several unmanned ground vehicles (UGVs) based on greedy pursuit policies. The potential of joint forces using unmanned systems for collaborative military engagement was explored by Mullens *et al.* (2006). Grocholsky *et al.* (2006) proposed a scalable target detection and localization algorithm for decentralized, heterogeneous sensor networks. A method for topological reconfiguration of control architectures for heterogeneous, distributed UAV-UGV sensor networks with small-scale experiments is discussed by Ippoolito *et al.* (2008).

Very few of the proposed solutions utilize on-board processing, opting to relay the sensor data to a centralized processing node, which limits scalability, and restricts the sensors' operational distance. Previously, we reported on the development of an ISR system using multiple aerial vehicles under fully distributed, cooperative control, where multiple UAVs cooperatively search, detect, and track a ground target [4]. In a related work, we demonstrated that the use of multiple autonomous sensing vehicles renders several key benefits, such as increased robustness and fault tolerance, reduction in time required to achieve mission objectives, and a decrease in the overall cost of ISR activities [10].

In this paper, we extend the sensor network capabilities of our system by adding stationary radio frequency (RF) sensors, which we call ground sensor pods (GSPs), and a mobile infrared (IR) sensor. The mobility of the IR camera is provided by an unmanned ground robot, which is controlled by our cooperative autonomous system (CAS). This mobile ground sensing platform (MGSP) cooperates with the stationary and airborne surveillance assets by first verifying the presence of the target at a predicted location, and then by helping to improve the accuracy of the target localization estimate through collection of additional measurements. In the next section, we outline the motivation behind our work, and then provide a short overview of key sensor network technologies in Sec. III. A detailed description of the mobile robot appears in Sec. IV, followed by experimental results in Sec. V. We conclude this paper with a few remarks and give a brief summary in Sec. VI.

II. MOTIVATION

Our previous work on cooperative multiple unmanned aerial vehicles demonstrated the benefits of using autonomous platforms to efficiently and effectively detect and locate ground targets [5,6,10]. Fig. 1 shows a photo of our UAVs and the two additional sensor platforms, all equipped with on-board autonomous control, sensing, and peer-to-peer communication capabilities. The motivation of the current work is the need to verify targets once they are detected and localized. The focus of this paper is to first show the effective

The authors are with the Department of Electrical and Computer Engineering, U.S. Air Force Academy, CO 80840 {chad.hager, dimitri.zarzhitsky, hyukseong.kwon, daniel.pack}@usafa.edu

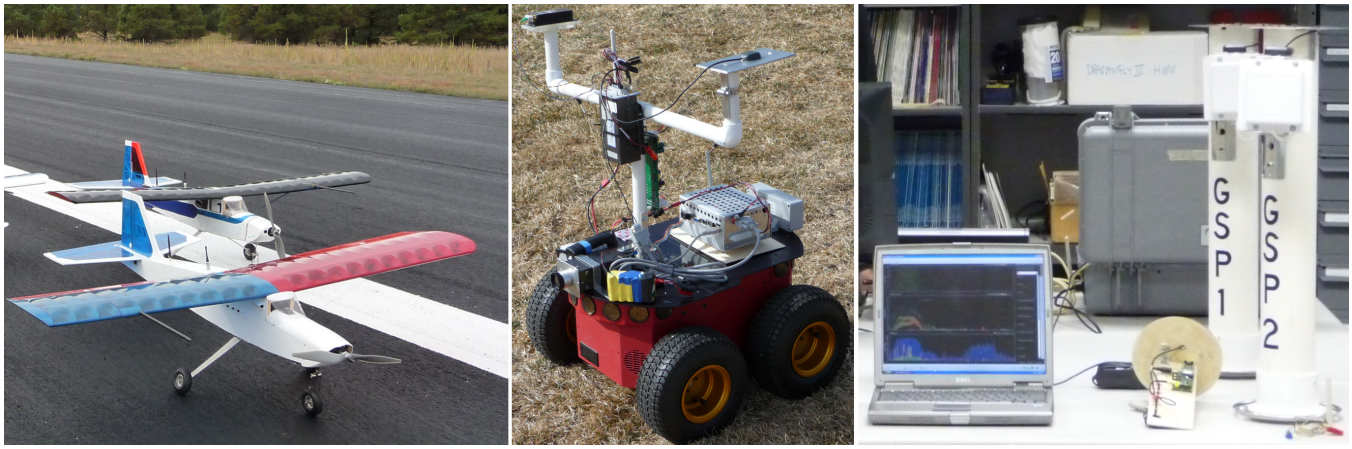

Fig. 1. Aircraft (left), mobile ground sensor platform (center), and the ground RF sensor pods (right) developed by the US Air Force Academy Unmanned Aircraft Systems Research Center

use of UGVs operating in the vicinity of suspected targets as UGVs verify the identities of targets. The UGVs also reduce the overall targets' location uncertainties by providing team members in a sensor network with close-up "views" of targets. The second focus of the paper is to present innovative cooperative technologies we developed, called CAS. The CAS technologies consist of hardware and software modules to facilitate portable control, sensing, and communication. We plan to use CAS in multiple sensing platforms regardless of their mobility functions, removing the need to develop separate capabilities for different systems operating as part of the same sensor network.

III. COOPERATIVE AUTONOMOUS SYSTEMS

This section provides a brief review of the important cooperative autonomous system technologies that we developed. The distinguishing characteristic of our sensor network comes from the fact that it is (1) autonomous, (2) heterogeneous, and (3) distributed/scalable. The autonomy of the system refers to the operational independence of each sensor platform. In particular, the ability to automatically allocate sensing and computational resources, as well as to determine the best allocation of sensing platforms in real-time – all without the need for a human operator to issue control directives. A solid mathematical foundation forms the basis of our sensor fusion method, and a straightforward transformation of each sensor output to a common form allows for a wide variety of sensor outputs to be incorporated. The reactive nature of our control algorithms and the absence of a centralized processing node allow the system to function in a fully distributed fashion (i.e., local sensor information is processed separately on neighboring platforms), with each one making independent decisions. Adding new, or removing old sensor platforms at runtime does not require special handling within the sensor network, which is a desirable property for many combat and emergency deployment scenarios.

A. Event-Driven, Multi-Threaded Software

A highly modular design of our distributed sensor network results in a flexible and practical research platform for developing cooperative sensor fusion, control, and communication technologies. The implementation of each module is sufficiently agile to allow many intelligent behaviors of varying complexity, without unnecessary external dependencies on hardware or software operating environments.

The on-board software is best characterized as a hierarchical collection of *event-driven* modules that encapsulate and abstract many individual sensor technologies equipped on the vehicles (see Fig. 2). This architecture is implemented using the Qt cross-platform application framework by Nokia, which makes it possible to compile and execute our distributed sensor application on several popular operating systems (i.e., Microsoft Windows and embedded versions of Linux). We were able to implement and evaluate several hybrid configurations of stationary and mobile ground sensors, as well as airborne platforms, with minimal amount of development [10]. Our software solution is multi-threaded, taking full advantage of recent advances in embedded, multi-core

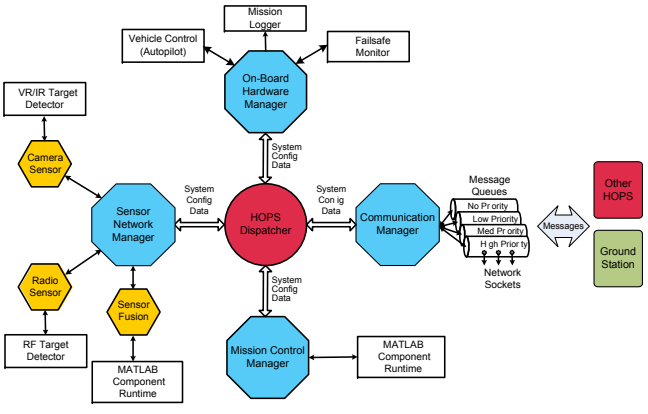

Fig. 2. Software architecture of the heterogeneous on-board processing system (HOPS), configured for the UAV sensor platform. A similar hierarchy, which consists of interacting top-level *managers* overseeing processing and IO operations within their respective module, is in use on all of the nodes participating in the distributed sensor network, including the ground station computer. Each module provides a strict separation between hardware-specific device drivers and general-purpose algorithms and behaviors.

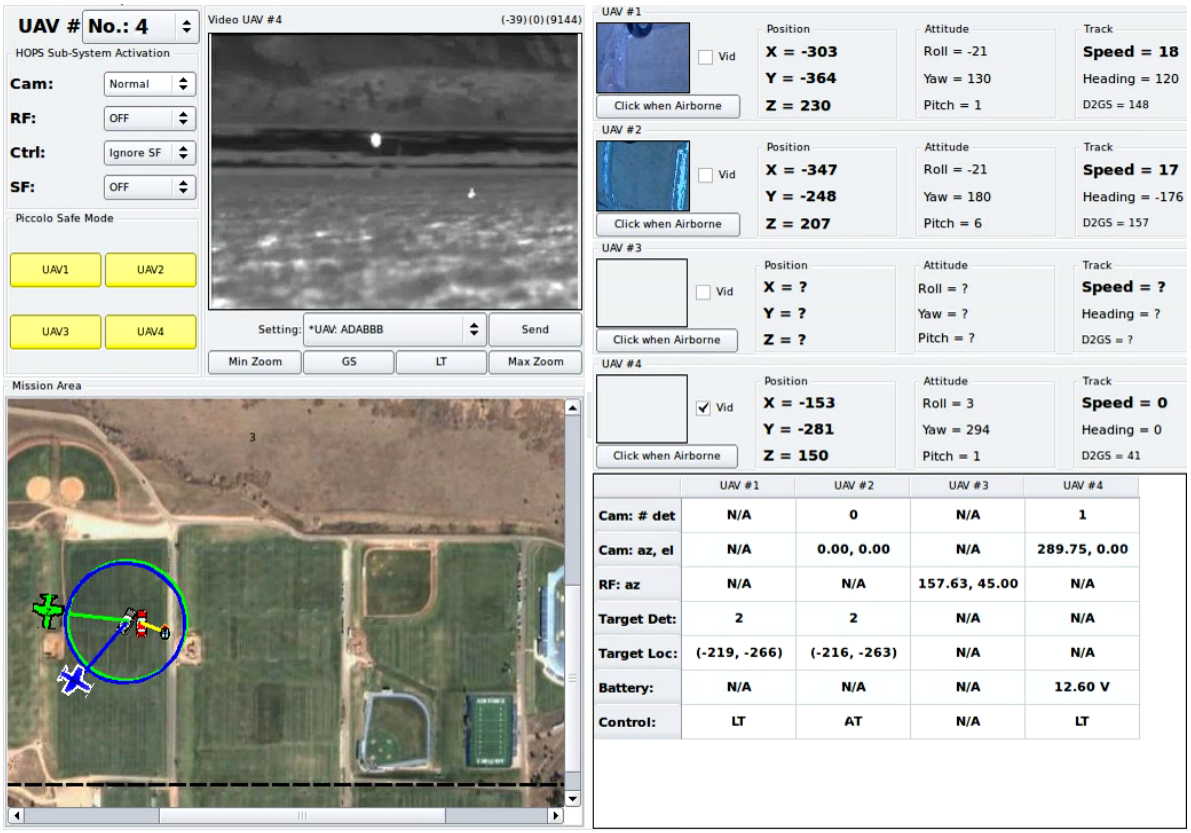

Fig. 3. Interactive graphical user interface in use on the multiple unmanned systems ground station. The ground station provides a real-time view of the telemetry and sensor data collected by the sensor network, along with various control options to modify parameters and behavior modules of each sensor node. The ground station also re-broadcasts the same information over a local network for use by more specialized applications, such as database archival and Google Earth visualization tools.

processing technologies. Support for standard communication methods, such as TCP/IP and UDP network protocols, is also included, enabling straightforward integration with other cooperative, autonomous systems. The software interface of the CAS ground station (see Fig. 3) conveniently allows just one person to monitor real-time progress of several vehicles simultaneously.

B. Heterogeneous Control and Coordination

To coordinate and control multiple mobile platforms in a distributive manner, we developed a state-machine based control architecture that selects an appropriate behavior from a set of collaborative behaviors. Each platform independently makes its control, sensing, and communication decisions in real-time based on the mission objectives, the status of the current mission, and the processed sensor data. For illustration purposes, Fig. 4 shows a sample state machine used by our UAVs. The state-machine shows that a UAV can operate in one of the four states (GS – global search, AT – approach target, LT – locate target, and RT – reacquire target) during a mission. An operating state can change in response to a variety of events (shown as arrows in the figure) which include sensor observations and a request for assistance from neighboring aircraft. The MGSP uses a simplified state-machine at present to meet the objectives

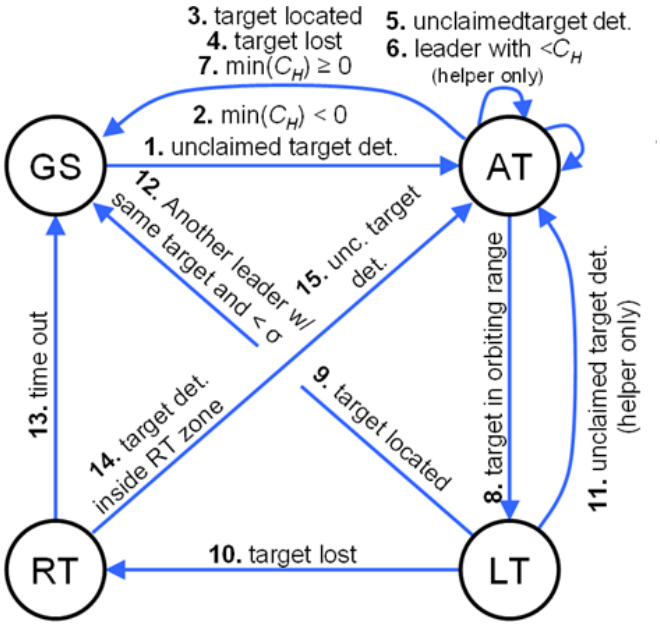

Fig. 4. Different control states (i.e., behaviors) and transitions of the cooperative UAV controller; C_H denotes the cost of helping with target localization. GS (global search) is the initial state of each UAV; target detections activate the AT (approach target) state. The UAVs orbit in the LT (locate target) mode while observing the target. If the aircraft sensors fail to detect a target, the controller switches to the RT (reacquire target) mode, which helps the UAVs to locate nearby targets.

of the target verification task. The stationary sensors do not have any control associated with them.

C. Sensor Fusion Technologies

Distributed and locally processed sensor observations are incorporated with sensor data communicated by cooperating sensor nodes using a modified Sigma-Point Kalman Filter (SPKF). The SPKF is capable of providing higher accuracy than the extended Kalman filter (EKF) because the SPKF incorporates up-to the second-order probability distribution of the estimate information [7]. Given the distributed nature of our sensor network, our foremost concern lies with the synchronization of various measurements, a problem that is exacerbated by the non-deterministic latency of the radio communication. To address this issue, we have developed the *Out-Of-Order Sigma-Point Kalman Filter* (O^{3}SPKF) [6], which uses the following equations for the state update:

$$\begin{aligned}\hat{x}^+(t_x) &= \mathbb{E}\left[x(t_x) \mid \mathbb{Y}^+\right] \\ \hat{x}^-(t_x) &= \mathbb{E}\left[x(t_x) \mid \mathbb{Y}^-\right] \\ \hat{y}(t_m) &= \mathbb{E}\left[y(t_m) \mid \mathbb{Y}^-\right]\end{aligned}$$

We use t_m to denote the time when a measurement was made, and t_x marks the time of the filter's last state estimate. Symbol $x(t)$ represents the true target state (i.e., its position and velocity) at time t, and $\hat{x}^-(t)$ and $\hat{x}^+(t)$ are the state estimates just prior and after a measurement is made at time t, respectively. Symbols $y(t)$ and $\hat{y}(t)$ represent the actual and expected measurement values at time t, while $\mathbb{Y}^-$ and $\mathbb{Y}^+$ denote the history of observations just before and after the new measurement data is incorporated.

In general, we are interested in the value of $x(t)$; however, the filter only "knows" $\hat{x}(t_x)$ corresponding to the newest measurement. To propagate $\hat{x}(t_x)$ forward in time to predict $x(t)$, we use a target's motion-model state equation. If $t_m < t_x$ then the sensor data is "old", but may still contain valuable information about the target. In this case, the O^{3}SPKF propagates the current state back in time in one step, incorporates sensor measurement at time t_m, and updates the current estimate of the target's position at time t_x using the delayed measurement in conjunction with covariance calculations [6]. In relation to the above state variables and observation, we can determine uncertainties via covariance, as shown below.

$$\begin{aligned}\Sigma^-_{\tilde{x}(t_x)} &= \mathbb{E}\left[(x(t_x)-\hat{x}^-(t_x))(x(t_x)-\hat{x}^-(t_x))^T\right] \\ &= \mathbb{E}\left[\tilde{x}^-(t_x)\tilde{x}^-(t_x)^T\right] \\ \Sigma^+_{\tilde{x}(t_x)} &= \mathbb{E}\left[(x(t_x)-\hat{x}^+(t_x))(x(t_x)-\hat{x}^+(t_x))^T\right] \\ &= \mathbb{E}\left[\tilde{x}^+(t_x)\tilde{x}^+(t_x)^T\right] \\ \Sigma^-_{\tilde{y}(t_m)} &= \mathbb{E}\left[(y(t_m)-\hat{y}(t_m))(y(t_m)-\hat{y}(t_m))^T\right] \\ &= \mathbb{E}\left[\tilde{y}(t_m)\tilde{y}(t_m)^T\right]\end{aligned}$$

Symbol Σ represents covariance of the errors between actual and estimated states, and $\tilde{y}$ denotes the error between true and expected measurement values. Finally, L is the Kalman gain used to adjust the estimated state and the error covariance.

$$\begin{aligned}L(t_x,t_m) &= \frac{\mathbb{E}\left[(x(t_x)-\hat{x}^-(t_x))(y(t_m)-\hat{y}(t_m))^T\right]}{\Sigma^-_{\tilde{y}(t_m)}} \\ &= \frac{\Sigma^-_{\tilde{x}(t_x)\tilde{y}(t_m)}}{\Sigma^-_{\tilde{y}(t_m)}}. \\ \hat{x}^+(t_x) &= \hat{x}^-(t_x)+L(t_x,t_m)\big(y(t_m)-\hat{y}(t_m)\big) \\ \Sigma^+_{\tilde{x}(t_x)} &= \Sigma^-_{\tilde{x}(t_x)}-L(t_x,t_m)\Sigma^-_{\tilde{y}(t_m)}L(t_x,t_m)^T,\end{aligned}$$

The resulting filtering technique allows the distributed system to optimally use even the stale sensor data as a part of the estimation process. The benefits of using O^{3}SPKF instead of buffered SPKF are explored in [10].

IV. MOBILE GROUND SENSOR PLATFORM

Due to its widespread use in robotics research applications and the availability of open-source software API, we selected the Pioneer P3-AT ground robot, manufactured by the Mobile Robots Inc., as the base platform for our UGV implementation. The unit comes equipped with on-board sonar sensors, providing basic obstacle avoidance functionality that is sufficient for outdoor testing in a semi-structured environment.

We use the CAS single-board computer system (SBC) to control the robot. Global Positioning System (GPS) updates and magnetic heading (from the Honeywell HMR2300 compass) are obtained from an autopilot unit, mounted on a non-magnetic scaffold above the robot's chassis (see the center photo in Fig. 1). The on-board CAS software provides IEEE 802.11b ad-hoc WiFi connectivity to other sensor units in the network, as well as a ground station. The CAS software module, as shown in Fig. 2, listens for target detection and localization events, and steers the robot toward an estimated target location. The IR sensor provides thermal gradient information via gray scale images. This camera system was selected because of its low cost and lightweight form factor, ideally suited for our unmanned ground robots and UAVs.

Fig. 5(a) shows the image of the outdoor propane heater (see Fig. 6, right) as observed by the IR sensor installed on the MGSP. The Thermal-Eye camera firmware uses automatic gain control when converting thermal data into pixels, which means that IR sources in the sensor's field of view can be identified without *a priori* knowledge of the target's total thermal output. This turns out to be a very useful feature when target detection must be performed without knowing

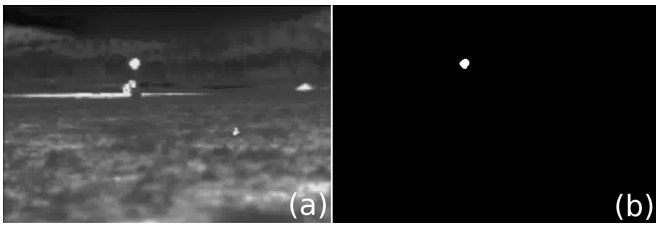

Fig. 5. An image captured by our IR camera sensor (left) and the filtered image through image processing (right)

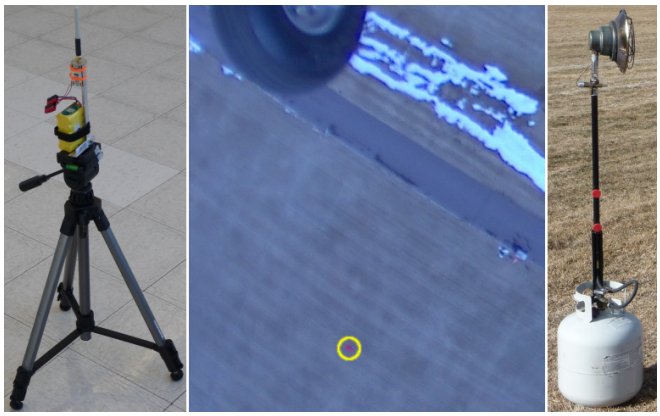

Fig. 6. Radio frequency (left), visual spectrum (center), and thermal (right) target emitters used during field evaluation of the distributed sensor network

the distance to the target, but it also affects the size of the target in the image as seen by the camera, requiring a typical camera calibration before each mission. In our experiments, we assume that the target emits a roughly circular thermal signature, and the presence of this IR feature is used to verify the target's position. During execution, the MGSP target detection algorithm searches for a circular compact "blob" in IR images. As seen in Fig. 5(a), blobs caused by reflections and other artifacts are characterized by a low compactness measure, allowing for a successful detection of the IR target, as visualized in Fig. 5(b).

Recall the earlier discussion of our sensor fusion algorithm, in which we described how measurements from different sensors are combined to produce an estimate of the target's location. Considering the type of information and feature detection abilities of each sensor, we found that in the general case, the optical camera can contribute more to the localization effort than the IR sensor [9]. However, in the environment that contains visually complicated backgrounds, extracting the target information from the scene can be challenging. Therefore, if the target has an active heat signature, e.g., an engine of a running vehicle, then the IR camera sensor can provide key discriminating information to enable successful target detection. In the experimental configuration that we consider in the next section, the GSPs' long-range RF-detection capability is first used to cue the orbiting UAVs on the approximate location of the target. The UAVs then use their on-board optical sensors to refine the position estimate. Finally, the MGSP is called-in to verify target identity with its IR camera.

V. Experimental Results

The main goal of the target localization experiment was to illustrate the value of cooperation by heterogeneous sensing platforms on a real-world ISR problem. As a secondary objective, we were interested in how each sensor contributes toward the overall target localization of our distributed sensor network. To allow for such an in-depth analysis, the CAS software collects and records detailed run-time execution traces, including timestamps for each sensor measurement and the corresponding output of the sensor fusion module. Thus, we can reconstruct each mission in detail using off-line, software-in-the-loop tools.

In this experiment, we used four stationary RF ground sensor pods positioned at randomly-selected locations within a rectangular, flat, one square kilometer mission area. Two UAVs, with the cruising speed of 22 m/s, were each equipped with a $100^\circ \times 140^\circ$ field-of-view, digital pan, tilt, and zoom optical camera, providing 640×480 color JPEG image output at a rate of two frames per second. The MGSP Thermal-Eye IR camera, with a 17° FOV, utilized a fixed, non-gimbaled mount on the Pioneer P3-AT robot, and provided a 720×480 gray scale JPEG output at three frames per second. The stationary GSPs first estimated an approximate position of a 2.4 GHz RF target by integrating the distance information obtained through analysis of received RF power attenuation. The coarse GSP estimation was then shared with all of the CAS vehicles via the on-board WiFi radio, and served as the initial target location for the airborne UAVs. The UAVs' on-board cameras, programmed to look for a red, car-sized target (see Fig. 6, middle), refined the estimate, and once the sensor network's computed uncertainty of the target's position fell below a predefined threshold, the MGSP approached the target for verification. It is important to emphasize that all of these decisions, including asynchronous cooperation between the sensors, as well as distributed navigation planning are completely autonomous, requiring no human involvement [4,10].

In this experimental study, the detection range of the RF-based GSPs exceeded the size of the search area, which eliminated the search phase of the problem from the experiment (evaluation of our system's ability to find targets is given in [10]). Instead, here we look at the localization accuracy and the rate of the error convergence as a function of the number and type of CAS units participating in the mission, during the last 100 seconds of the localization effort. As explained in the previous section, we utilize the off-line, software-in-the-loop capability of our sensor network implementation to obtain a post-mission reconstruction for most of the localization data presented in Table I and Fig. 7 (note that the solid-black line in the figure corresponds to the "GSP/UAVx2/MGSP" entry in the table, which was the actual tested configuration).

From the convergence data in Table I, we confirm the hypothesis that the localization error is decreased as additional sensors are introduced into the network. In particular, note that due to the inherent ambiguity involved in estimating distance from an RF source based on the received power measurement, the GSP-only network can resolve the target's position to within a 40 m radius (which gives a spatial resolution error of about 4% over the 1×1 km^2 mission area). The estimate error is reduced by an order of magnitude once the target is observed by one of the UAVs (which occurs approximately 60 seconds into the mission). Making use of the sensor data collected by the second UAV results in additional 54% to 65% improvement in the accuracy of the position estimate (compared to the UAVx1 scenario) during

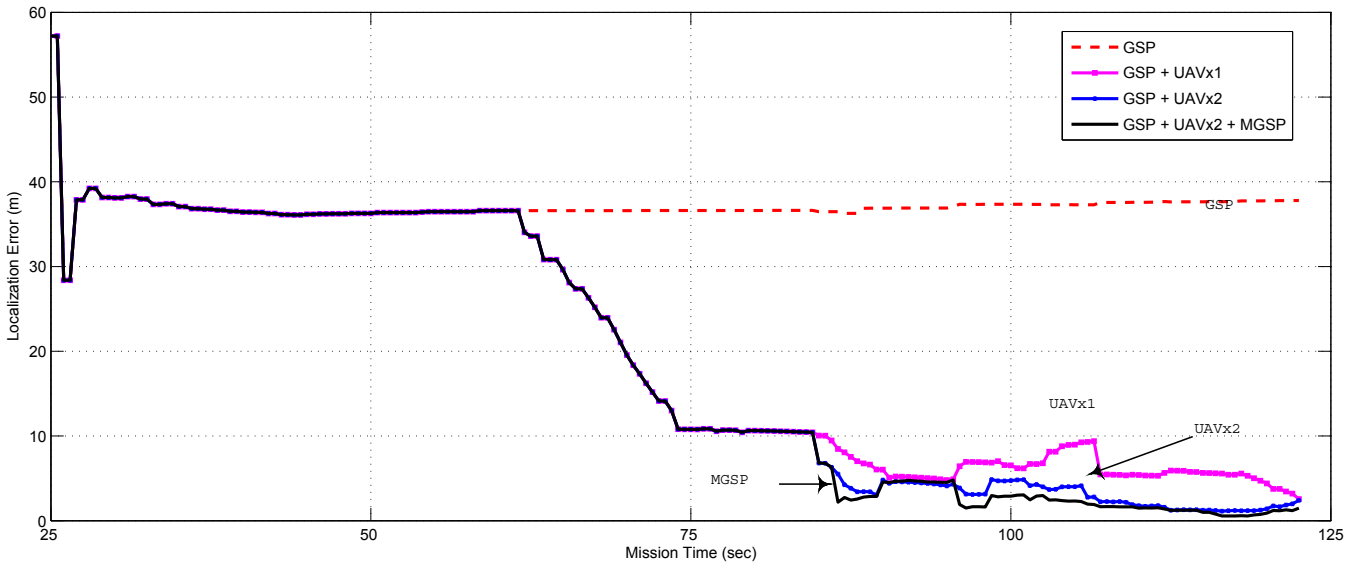

Fig. 7. Time-indexed convergence of the target localization error for each configuration of the distributed sensor network

TABLE I
CONVERGENCE OF THE OFFLINE LOCALIZATION ERROR

	Position Error $\pm\sigma$ *(meters)*		
	last 60 sec	last 30 sec	last 5 sec
GSP	37.04 $\pm$ 0.47	37.45 $\pm$ 0.25	37.76 $\pm$ 0.03
GSP/UAVx1	10.30 $\pm$ 7.37	5.95 $\pm$ 1.47	4.29 $\pm$ 1.01
GSP/UAVx2	8.38 $\pm$ 8.50	2.75 $\pm$ 1.33	1.55 $\pm$ 0.41
GSP/UAVx2/MGSP	7.94 $\pm$ 8.78	2.01 $\pm$ 1.18	0.94 $\pm$ 0.33

the last 30 seconds of the ISR mission. Finally, incorporating the IR measurements from the MGSP provides another 27% to 39% gain. Occasional increases in the localization error are caused by occluded sensor measurements, wind turbulence, GPS drift, and communication lag. These artifacts increase the uncertainty of the target position estimate, which perturbs the localization error as shown in Fig. 7. Along with the improvement of the target location estimate to the sub-meter level, note that the addition of the MGSP sensor observations also decreases the variance of the position estimate (see Table I), indicating smaller uncertainties within the O^{3}SPKF algorithm, which in turn meets our goal for IR-based target verification.

VI. SUMMARY

In this paper we described a decentralized solution for an autonomous, heterogeneous sensor network, and considered the use of a mobile ground robot equipped with an IR sensor for solving a target verification and localization problem. This CAS framework makes use of event-driven, multi-threaded, cross-platform software to achieve optimized data processing and increased robustness. Challenges caused by latency and non-deterministic, asynchronous sensor data processing are addressed with a novel O^{3}SPKF sensor fusion algorithm. Our theoretic developments are evaluated in the context of a real-world ISR mission, in which we use RF, IR, and optical sensors to detect and localize a stationary ground target.

We plan to continue sensor development to include mobile RF sensors that can improve on the localization accuracy of the stationary GSPs. In addition, we are extending the current O^{3}SPKF and control implementations to handle multiple ground targets.

REFERENCES

[1] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 1070-9932:16–26, 2006.

[2] C. Ippolito, S. Joo, K. Al-Ali, and Y. H. Yeh. Flight testing polymorphic control reconfiguration in an autonomous UAV with UGV collaboration. In *IEEE Aerospace Conference*, March 2008.

[3] K. Mullens, B. Troyer, R. Wade, B. Skibba, and M. Dunn. Collaborative engagement experiment. In *SPIE Proc of Unmanned Systems Technology VIII, Defense Security Symposium*, April 2006.

[4] D. Pack, P. DeLima, G. Toussaint, and G. York. Cooperative control of UAVs for localization of intermittently emitting mobile targets. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, 39(4):959–970, 2009.

[5] D. Pack and G. York. Developing a control architecture for multiple unmanned aerial vehicles to search and localize RF time-varying mobile targets: Part I. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3965–3970, April 2005.

[6] G. Plett, D. Zarzhitsky, and D. Pack. Out-of-order sigma-point Kalman filtering for target localization using cooperating unmanned aerial vehicles. *Advances in Cooperative Control and Optimization, Lecture Notes in Control and Information Sciences*, 369:22 – 44, 2007.

[7] R. van der Merwe, E. Wan, and S. Julier. Sigma point Kalman filters for nonlinear estimation and sensor fusion: Applications to integrated navigation. In *AIAA Guidance, Navigation, and Control Conference at Exhibit*, August 2004.

[8] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Trans. Robot. Automat.*, 18(5):662–669, 2002.

[9] Y. Yoon, S. Gruber, L. Krakow, and D. Pack. Autonomous target detection and localization using cooperative UAVs. *Optimization and Cooperative Control Strategies, Lecture Notes in Control and Information Sciences*, 381-1, 2008.

[10] D. Zarzhitsky, P. DeLima, and D. Pack. Localizing stationary targets with cooperative unmanned aerial vehicles. In *Proceedings of the IFAC Workshop on Networked Robotics (NetRob)*, 2009.