Vision-Based Target Detection and Localization via a Team of Cooperative UAV and UGVs

Sara Minaeian, Jian Liu, and Young-Jun Son

***Abstract*—Unmanned vehicles (UVs) play a key role in autonomous surveillance scenarios. A major task needed by these UVs in undertaking autonomous patrol missions is to detect the targets and find their locations in real-time. In this paper, a new vision-based target detection and localization system is presented to make use of different capabilities of UVs as a cooperative team. The scenario considered in this paper is a team of an unmanned aerial vehicle (UAV) and multiple unmanned ground vehicles (UGVs) tracking and controlling crowds on a border area. A customized motion detection algorithm is applied to follow the crowd from the moving camera mounted on the UAV. Due to UAVs lower resolution and broader detection range, UGVs with higher resolution and fidelity are used as the individual human detectors, as well as moving landmarks to localize the detected crowds with unknown independently moving patterns at each time point. The UAVs localization algorithm, proposed in this paper, then converts the crowds' image locations into their real-world positions, using perspective transformation. A rule-of-thumb localization method by a UGV is also presented, which estimates the geographic locations of the detected individuals. Moreover, an agent-based simulation model is developed for system verification, with different parameters, such as flight altitude, number of landmarks, and landmark assignment method. The performance measure considered in this paper is the average Euclidean distance between the estimated locations and simulated geographic waypoints of the crowd. Experimental results demonstrate the effectiveness of the proposed framework for autonomous surveillance by UVs.**

***Index Terms*—Algorithms, automation cooperative systems, geographic information systems (GISs), position measurement**

I. Introduction

UNMANNED vehicles (UVs) are proper substitutions for the human in some dangerous, dull, dirty, or impossible applications [1]. Nowadays, different types of UVs, including unmanned aerial vehicles (UAVs) as well as unmanned ground vehicles (UGVs), are used for surveillance, crowd control, border patrol, firefighting, agriculture, navigation, and search and rescue purposes. The user may plan the mission remotely, or the control can take place onboard via sensors and actuators, depending on the level of the vehicles' autonomy [2]. In most autonomous cases, the first functional step is to detect the targets through some sensors and to identify their real locations in order to implement further operations. According to [3], visual sensors are the most common sensors for target detection in surveillance applications, due to their low cost and vast variety of analysis methods. For this type of applications, which is the field of focus in this paper, vision-based human detection methods are generally divided into two submodules: 1) motion detection, whose goal is to detect targets in motion and 2) classification, which categorizes the detected target into human classes based on some known features [4].

In this paper, we have further developed our previous optical-flow-based motion detection algorithm in [5], for low fidelity outdoor crowd detection via a UAV, due to the promising performance of optical flow in terms of accuracy and cost. However, considering the low-resolution images from the UAV, the classification would be more challenging. Therefore, we propose to utilize higher resolution visual sensors of a UGV in a closer distance and with a horizontal field of view toward the targets, for high fidelity human classification. In this paper, histogram of oriented gradients (HOG), with a dominant performance at intermediate to higher resolution images, is applied to detect individuals from the UGV camera. Hence, UAVs and UGVs can contribute to enhance the surveillance effectiveness, in terms of crowd detection and localization.

In order to track a detected target, whose location is unknown *a priori*, UVs need the real-time world coordinates of the target (as opposed to its image location, which is the output of the detection algorithm), so that they can predict its future location and plan their routes for the consecutive time stamps, accordingly [6]. The transformation of the image position into real-world location requires either camera position and orientation, or several landmarks locations, where the latter are mainly detected via their unique identifiers. In order to localize the target based on the UAV camera pose (i.e., its position and orientation [7]), the 3-D position (latitude, longitude, and altitude) of the UAV should be considered based on global positioning system (GPS). However, by using this approach to solve the localization problem, both lateral and vertical positioning errors of the GPS receivers will contribute to the sources of error for target's location estimation. Hence, we take advantage of UGVs with known 2-D geographic locations (with lateral positioning error only) as independently moving landmarks. Then, an efficient perspective transformation array is computed for converting the image location to

Manuscript received May 16, 2015; revised August 19, 2015; accepted September 26, 2015. Date of publication November 17, 2015; date of current version June 14, 2016. This work was supported by the Air Force Office of Scientific Research under Grant FA9550-12-1-0238 (A part of Dynamic Data Driven Application Systems Projects). This paper was recommended by Associate Editor Z. Li.

The authors are with the Systems and Industrial Engineering Department, University of Arizona, Tucson, AZ 85721 USA (e-mail: minaeian@email.arizona.edu; jianliu@sie.arizona.edu; son@sie.arizona.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2015.2491878

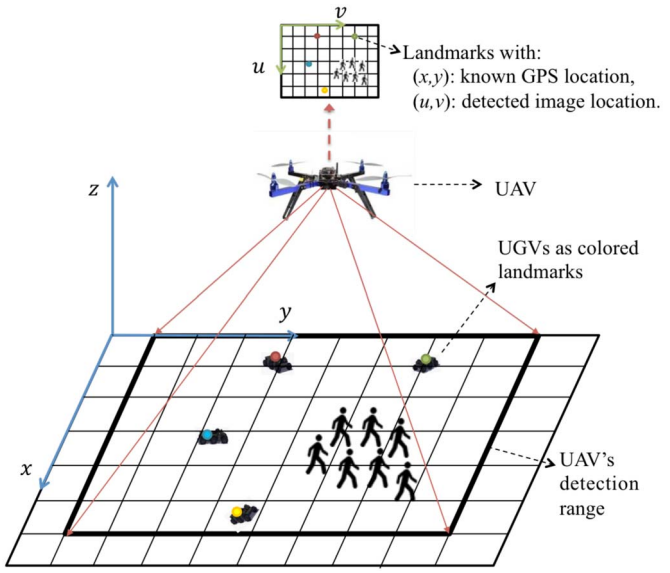


Fig. 1. General framework for detection and localization using a UAV and at least four UGVs, serving as moving landmarks for crowd localization.

the real-world position in real-time, considering the fact that the landmarks (i.e., UGVs) and target (i.e., crowd) are located close to each other on the same plane, which is assumed to be smooth enough for an accurate transformation between the two planes (i.e., the image and the surveillance region). The experiments conducted in this paper reveal that such assumption is fairly realistic.

In this paper, we have considered a collaborative team of a UAV and multiple UGVs, equipped with visual sensors (i.e., cameras), to control crowds of people in a border area of Tucson, Arizona. The authors aim at proposing an effective vision-based surveillance framework by cooperative UVs. Based on UAVs' and UGVs' different capabilities and goals, we have customized and further developed various detection and classification approaches at different fidelities, for two layers of crowd detection in real-time. In particular, for the target's real-world localization, a new approach based on moving landmark detection and plane perspective transformation is presented. This approach is proposed for the first time in this paper to the best of our knowledge. Moreover, we have presented a heuristic method for localization of the detected individuals in outdoor environments, through UGV regular camera. Fig. 1 depicts the general framework and UAV-UGVs team organization that is proposed in this paper.

The rest of this paper is organized as follows. Section II reviews the related literature in two different topics: 1) crowd detection and 2) localization. Section III presents the details of the two crowd-detection algorithms (as localization prerequisites) applied via UAV and UGVs. Section IV discusses the proposed localization algorithm via UAV based on moving landmarks in detail. Moreover, a rule-of-thumb for localization via UGVs is also included in this section. The testbed setting and experimental results are discussed in Section V, which show the effectiveness of the proposed work, and Section VI concludes this paper and suggests future work ideas.

II. Background and Related Work

Applying computer vision methods on the UAV/UGV field have been continuously improved in recent years to process captured image sequences and videos from the environment to produce numerical or symbolic information in forms of decisions. A number of research papers (see [8]–[10]) have studied the problem of autonomous deployment and formation control of cooperative UAVs or UAVs-UGVs teams through visual sensors in surveillance or search and rescue applications. The theme of localization in these studies mostly refers to robot's visual localization, to compensate lack of accurate positioning systems. Furthermore, UAVs GPS data is usually used to localize other cooperative UVs. However, in this paper, computer vision methods are applied to detect, identify, and accurately geo-localize unknown targets of interest (here, crowds) through cooperative UGVs with known geographic positions, in order to reduce the sources of localization error and increase robustness.

A. Related Work on the Crowd Detection

According to [11], detection techniques mainly aim at tracking features, appearance, or motion, among which, we use motion-based techniques for detecting the moving crowd. In [4], motion detection methods are conventionally categorized as: 1) background subtraction; 2) spatio-temporal filtering; and 3) optical flow. Background subtraction tries to segment the moving foreground by considering the difference of the current frame compared to a reference. However, this technique is more effective when using a static or pan-tilt-zoom camera (e.g., in [12]). Spatio-temporal filtering technique characterizes the motion pattern of the moving object throughout the frame sequence; but it is more sensitive to noise and variations in the movement pattern. Optical-flow-based techniques, however, consider the relative movements between an observer and the scene, and hence are more robust to simultaneous motions of both camera and target. Optical flow is also suggested as the most popular analysis technique for motion detection using the camera mounted on UAVs [13]. This inspired us to consider optical-flow-based motion detection by the UAV in this paper.

After detecting the moving targets by the UAV, one further step for classification of human versus nonhuman crowds is needed, for which, we use UGVs with closer distances to the targets. Paul *et al.* [4] discussed that a moving object could be classified based on its shape, motion characteristics, or texture. Shape-based methods use pattern recognition approach which is not robust enough, due to various body positions [4]. Motion-based techniques, on the other hand, are based on a key assumption that the target's motion features are unique enough to be recognized. However, these methods rely on the learning of predefined movements, and are restricted to recognize a moving human. Texture-based methods overcame this limitation and provide an improved detection quality as well as a better accuracy in human classification. As a well-known texture-based technique, HOG applies a high-dimensional vector of features on edges of the image.

Then it uses the support vector machine (SVM) as a collection of supervised learning models library for data analysis, to classify the object. According to [14], HOG/linear-SVM has a significant advantage over other state-of-the-art systems for human detection onboard a vehicle, when the image resolution is high enough. Since the UGVs have a higher resolution upright view of the crowd, we customized and applied the HOG algorithm for human detection with a relatively high fidelity, which also provides us with the image positions of detected individuals.

B. Related Work on the Localization

The accurate location of a target (crowd, in this paper) is needed for tracking, restoring, and future processing. According to [15], there are two main approaches toward localization: 1) stereovision, in which the target location is extracted from its image coordinates in multiple cameras and 2) monocular vision, in which the target position is computed from its image location in a single camera, mostly through the camera calibration. The stereovision is not applicable to our problem, since we consider one UAV in each team of UVs to provide the low fidelity big picture of the crowd movements and hence, we do not have access to multiple views of the same target from multiple cameras (UAVs in this paper). Note that we can neither combine the image coordinates from a UAV and a UGV in a stereovision setup, since their fidelities for crowd and individuals detection are different. Therefore, we only consider the case of monocular-vision-based localization in this paper, using the UVs onboard camera.

A number of papers (see [16], [17]) have studied target localization, from a stationary aerial vehicle, with a low altitude and low velocity. However, those methods are not applicable in our case, due to high complexity and lower stability while flying, associated with UAVs. Based on our assumptions, the UAV can only provide low fidelity detection of the moving crowds. Moreover, the onboard computational resources of the UAV are limited due to its payload restrictions.

In the case of nonstationary aerial vehicles, Redding *et al.* [18] discussed a method to find the geolocation of a ground-based stationary target, using a recursive least square filter. However, the 3-D geographic location of the UAV, as well as the camera orientation should be determined accurately in advance, and hence, add to the problem complexity. In general, the geometric camera calibration requires estimation of 11 unknown intrinsic and extrinsic camera parameters and hence, we have considered landmark-based localization as a less complex alternative to this process for a real-time application.

In the monocular vision literature, landmarks have been widely used in applications such as navigation or robot self-localization. In [19], a twofold pose estimation algorithm is proposed in which, the authors compute the pose of the robot (i.e., UAV) via detecting and tracking a planar pattern through the onboard camera, as well as identifying landmarks located on the UAV through an external camera. However, their transformation lacks flexibility due to the assumption of preserved ratio of distance between fixed landmarks. In the well-known pose from orthography and scaling with iterations (POSIT) algorithm, Dementhon and Davis [7] used at least four noncoplanar points on any object of interest with their relative geometry assumed known, in order to find the pose of the object. They also made use of camera's intrinsic parameters to find the perspective scaling of the known object. However, if the relative geometry of noncoplanar landmarks on the object is not known, the algorithm will either converge to a bad pose or not converge at all. In the scenario of crowd surveillance via UAVs, though, we face an unknown environment, in which it is not always feasible to use fixed landmarks with preserved relative geometry. We need to use landmarks with known real-world positions, which can relocate along with the UAV detection range. Thus, in this paper we propose to transform the location of the detected target through localizing independently moving landmarks with unknown moving patterns.

TABLE I
COMPARISON OF LANDMARK-BASED LOCALIZATION IN THIS PAPER VERSUS THE POSIT METHOD

Properties	POSIT	This Work
Target is:	Known	Unknown
Landmarks are:	Fixed	Moving
Landmarks are:	Non-coplanar	Coplanar
Relative geometry is:	Preserved	Changing
Camera location is:	Known	Unknown

Another challenge we faced in this paper was to assign appropriate moving landmarks with aforementioned characteristics. In our border patrol scenario, the individuals in the crowd are not detected via a UAV, due to its low fidelity setup and thus, we cannot consider their body parts (with known relative geometry) as landmarks. Therefore, we opted to consider UGVs as moving landmarks in this paper, for a more robust transformation and to use their location based on geographic information system (GIS), instead of the UAV's (i.e., camera's) geographic location. The reason is that in the latter case, we need the exact altitude of the UAV as well as its latitude and longitude, which will lead to greater estimation error. Furthermore, in our proposed method there is no need to find the intrinsic parameters of the camera in advance, which will omit the camera calibration process (with eleven unknown parameters) as another source of uncertainty. Table I summarizes the main characteristics and assumptions of this paper, compared to the POSIT method.

In order to estimate the target's position with respect to a ground vehicle (e.g., UGV) and tracking it, a number of research works (see [20], [21]) use RGB-D sensors such as Microsoft Kinect, which allow capturing per-pixel depth information along with the RGB images. In spite of their ease of application and popularity for indoor 3-D mapping (because of GPS signal loss), these cameras are not appropriate for outdoor applications, due to their limited detection range. For estimating the target's distance from the regular camera, Mora *et al.* [22] used simple trigonometry in a robot's automatic gaze control application for indoor environments. In this paper, we propose a heuristic method to estimate the 2-D

geographic locations of detected individuals, based on UGVs camera pose in our outdoor application.

III. Target Detection: The Prerequisites

In order to detect the moving target (i.e., crowd), we have applied two different methods for UAVs and UGVs. UAVs usually have a wide detection range, fast coverage of the search area, and low resolution, whereas UGVs have better resolution for detection purpose, though narrower and obscured detection range, and a slower coverage rate [23]. This motivates us to use cooperative teams of UAVs and UGVs in a synergetic manner. For this purpose, we use a motion detection algorithm based on the optical flow for the UAV with slower background motion and a human detection algorithm based on the HOG for UGVs with a higher resolution and a portrait image of targets. In addition, we have used OpenCV as an open-source computer vision library in order to efficiently develop our detection algorithms to operate in real-time. This section discusses these two algorithms in detail.

A. Optical-Flow-Based Motion Detection for UAV

In this paper, we have further developed our motion detection algorithm presented in [5] to detect crowds of different sizes and movement speeds in real-time. Every detection interval (which can be set to every frame, depending on the computational resources), the algorithm first extracts some keypoints, which have good features to be tracked over subsequent image series. In this paper, the keypoints are assigned using the GoodFeaturesToTrack (GFTT) method [24], which is invariant to rotation and spatial movement. GFTT sorts the pixels with two large eigenvalues of the autocorrelation matrix, and then chooses a number of pixels as keypoints, for which, their smaller eigenvalue is higher than a threshold. Hence, these pixels are more likely to represent a corner. The aforementioned threshold is determined based on the image resolution and the illumination (i.e., the contrast), so that the smaller eigenvalues are sufficiently large to compensate for noise. A method to set upper bound and lower bound on this threshold is discussed in detail in [24].

In order to match these keypoints across successive frames based on their displacement, we apply the sparse optical flow concept and solve the tracking problem using the pyramidal Lucas–Kanade (PLK) algorithm. The tracking problem is an over-constrained system of equations on a neighborhood of each detected GFTT keypoint, and needs to be solved for the velocity vector of that keypoint across subsequent frames. The final solution is then formulated as

$$\begin{bmatrix} \mathrm{v}_u \\ \mathrm{v}_v \end{bmatrix} = \begin{bmatrix} \sum_i I_{u_i}^2 & \sum_i I_{u_i} I_{v_i} \\ \sum_i I_{u_i} I_{v_i} & \sum_i I_{v_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i I_{u_i} I_{t_i} \\ \sum_i I_{v_i} I_{t_i} \end{bmatrix} \quad (1)$$

where at each time frame, v_u and v_v are vertical and horizontal pixel's velocity elements for the keypoint, respectively; I_{u_i} and I_{v_i} are the spatial derivatives across the current frame along the image vertical (u) and horizontal (v) axes for pixel i in the keypoint's neighborhood, and I_{t_i} is the time derivative between the two frames for the same pixel i.

After matching the keypoints and finding their displacements in current frame, the algorithm affine-transforms the two successive frames and maps the current frame into the previous one as outlined in [5]. Hence, the camera movements will be compensated and background can be removed. Next, the moving foreground can be segmented using absolute differencing. Due to the assumption of UAVs lower fidelity, detecting the crowd as one target (instead of individuals) is a challenging task. In this paper, a local motion history function tracks general movements of each segmented blob as one unified crowd. The parameters we set for crowd detection are the upper bound and lower bound on target's velocity vector (as the result of PLK) and the blob size (as the result of motion segmentation) as functions of UAV flight altitude. Finally, we determine the size of each detected blob (crowd in this paper) to set a bounding box around it. In this way, each moving target will be visually detected and tracked via the UAV. The target's image location is then extracted and can be reported to the localization module either as the center of blob or as the four corners of its bounding box (representing the crowd size). We opt to use the blob's center in Section V, for convenience and without loss of generality.

B. HOG-Based Human Detection for UGV

The motion detection algorithm, described in Section III-A is not appropriate to be used with the UGVs in this paper. Although the UAV has a fast coverage of the search area, once a target (i.e., crowd) is detected, it hovers above the designated area due to its broader detection range, until the moving crowd leaves the area. However, the UGVs are continuously moving in searching and tracking the individuals in the crowd, though at different speeds depending on the status (searching versus tracking). Therefore, the relative motion of the background is faster in UGVs' video stream, and as a consequence, differentiating between the moving crowd and a fast moving background is computationally more complex. The motion segmentation is hence a challenging task. Moreover, the UGVs resolution is higher for individuals' detection, while its detection range is narrower and can only observe those individuals on the crowd's boundary who are not occluded by others. Therefore, we need to assign multiple UGVs to collaborate with the UAV in classifying human crowd versus nonhuman group of objects.

In this paper, we use an HOG-based detection module for the UGVs as described in [5]. However, we changed a number of parameters (e.g., the window size and the block stride) as outlined below to improve the human detection performance considering our specific environment, and to run the algorithm in real-time onboard UGVs. The reason is that, HOG can provide high accuracy human detection, though at a relatively high computational cost; hence, it cannot operate on the UGV onboard computer with limited computational resources, in real-time.

After retrieving the HOG descriptor, the algorithm applies a linear SVM for human classification, where its coefficients are extracted from an OpenCV trained classifier for detecting people. Depending on the environmental conditions and the

camera's resolution, in this paper we altered the window size, the classifier function, and block stride according to the training data set. The modified values for these parameters are set based on a series of experiments on the UGVs onboard computer with online video stream to find the best combination for a more accurate, real-time human detection. Note that these parameters should be altered depending on application and the level of adequate accuracy. For instance, while reducing the window size, the computational complexity reduces and the algorithm runs faster, though at the cost of missing some targets' detection. So, the block stride should be modified accordingly to compensate the degraded accuracy. Once the targets are detected, the algorithm sets a bounding box around each individual and a scale computation is applied on them to extract the image location of their feet. These coordinates are then passed into a function for computing target's real-world location, as described in Section IV-C.

As mentioned earlier, each UGV detects a number of individuals in the crowd's boundary. After estimating their real-world locations, these coordinates should be averaged in order for the UGV to predict the crowd's next location.

IV. Target Localization: The Proposed Method

After detecting the targets (crowd/individuals in this paper), we need to find the real-world coordinates of them in order to estimate their locations in the next time stamp for UVs' path planning purpose.

As noted in Section II-B, we conduct a landmark-based localization of the detected targets in real-time without the need to calibrate the camera in advance or stereovision of multiple cameras. The main assumption we consider in this paper is the weak perspective approximation, in which the plane containing the crowd and UGVs (i.e., Earth) is far enough away from the UAV camera that can be considered parallel to the image plane and any internal depth differences between the objects on it can be disregarded; so, landmarks and targets are at the same distance from the camera.

In order to transform the image location of the crowd to its real-world location, we need at least four coplanar, noncollinear points with known geographic and image positions, as landmarks. In this paper, we propose a new approach to find transformation matrix between the two planes based on moving landmarks identified, to cope with the localization problem when the camera is moving. An emerging challenge when using moving landmarks over static (i.e., fixed) landmarks for localization is that in the latter case, once the landmarks are identified, there is no need to recompute the transformation matrix; however, in the former case, all the computations (e.g., landmarks assignment and transformation estimation) should be repeated every time the landmarks move and hence, adds more complexity. To solve this problem, we use the cooperation between UVs for sending path-planning data to save some computations. In this setting, the UAV and UGV share their planned waypoints for the next tracking interval (the details can be found in [6]). Therefore, once the assignment problem in (2) is solved at the beginning of each interval (Δt), the real-world and image locations of each landmark are coupled and we use continuity to discard the need to solve assignment problem for the rest of the time interval. This will also reduce the risk of missing GPS data from the UGVs, since the proposed localization algorithm uses the UGVs' planned waypoints as the landmarks' real-world coordinates during the tracking interval.

The main contributions of the proposed work compared to the available literature on the subject (e.g., POSIT) are as follows.

1) We consider independently moving landmarks instead of the static ones with known relative geometry.
2) We take advantage of the known geographic locations of UGVs (i.e., landmarks) as part of our testbed, instead of computing a scale factor for projection.
3) We estimate the real-time location of the targets at each frame from a single camera, compared to stereovision.
4) The camera calibration for finding the camera position and orientation is not needed, since we use the perspective transformation between two planes at every detection interval.

The methods for landmark assignment and location transformation via UAV are discussed in Sections IV-A and IV-B, and then a rule-of-thumb localization method via UGV is presented in Section IV-C.

A. Landmark Assignment by UAV

The landmark identification is an important task in our proposed localization framework. Since the testbed we are using for surveillance purpose contains UGVs for contributory detection and classification of moving targets, we make use of them as moving landmarks. The real-world locations of these landmarks are, then, provided by the GPS sensors mounted on them. To find their image locations, though, computer vision and assignment algorithms are needed as discussed below.

1) Vision-Based Landmark Detection: Considering our testbed, two major properties are required for the landmarks: 1) to be detectable robustly in different illuminations and distances and 2) to be uniquely identifiable. These characteristics suggest that we use color as the landmark identifier, considering its unique properties. We applied labels of unique colors on the UGVs and trained a color detection algorithm to detect those landmarks based on their unique range of hue-saturation-value (HSV) in the image sequence from the UAV camera. To this end, the algorithm applies thresholds on the video frames and segments the blobs of unique colors (those of landmarks) based on the defined HSVs. Once detected, the image locations of the centers of these landmarks are extracted to compute the perspective transformation. Considering the fact that there are eight unknown elements for transformation matrix, we need at least four coplanar noncollinear landmarks, each generating two equations, to estimate the transformation matrix (see Section IV-B for further details). The coplanarity property is assured by considering our major assumption of relatively constant terrain elevation (due to the weak perspective approximation); so that, the UGVs and crowd are all assumed on the same plane. In this paper, though, we run experiments using four to six UGVs to reduce the risk of landmarks collinearity

and study the effect of number of landmarks on localization accuracy.

The movement of the landmarks (i.e., UGVs) pose another challenge for their real-time localization, which we tried to address by discarding camera calibration at each time step and use the UGVs' planned waypoints at each tracking interval, hence, proposing a robust perspective transformation that eliminates the need for iterations in order to compute scaled orthographic projection as in POSIT.

2) Moving Landmarks and Targets Differentiation: As pointed out earlier in this section, one major challenge in our proposed scenario is that UGVs may move in the UAV detection range, and this may result in being detected as moving blobs by the UAV detection algorithm. In order to address this challenge, we introduced an assignment problem and solved it at every tracking interval (Δt) for the best assignment of motion-detected blobs to their corresponding color-detected landmarks, if any. Hence, the moving landmarks can be discriminated from the moving targets. The rationale to formulate this problem as an optimization model derives from the need to minimize the total error of substituting the color-detected coordinates with the corresponding motion-detected coordinates for landmarks. This error is a function of distances between moving blobs and colored landmarks. The proposed assignment problem is mathematically modeled as

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} D_{ij} z_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{m} z_{ij} = 1 \qquad \forall i = 1, \ldots, n \\ & \sum_{i=1}^{n} z_{ij} \le 1 \qquad \forall j = 1, \ldots, m \\ & D_{ij} z_{ij} \le \varepsilon \qquad \forall i, j \\ & z_{ij} \in \{0, 1\} \qquad \forall i, j \end{aligned} \tag{2}$$

where D_{ij} is the Euclidean distance between ith color-detected and jth motion-detected blobs' coordinates on image, z_{ij} is a binary decision variable which indicates whether the motion-detected coordinates for blob j corresponds to the color-detected coordinates for landmark i (i.e., $z_{ij} = 1$) or not (i.e., $z_{ij} = 0$), n and m are the numbers of color-detected landmarks and motion-detected moving blobs, respectively, and ε is a threshold on the acceptable Euclidean distance.

In (2), the third constraint ensures that for each i, $z_{ij} = 1$ only if D_{ij} is less than a threshold, which means the ith color-detected landmark (UGV) is also detected as a moving blob by the UAV. This depends on the motion detection performance and the parameters set to reduce the probability of detecting UGVs as moving targets (see Section III-A). The pseudocode of the heuristic method to solve the proposed assignment problem is outlined as below.

1) Initiate matrix $\mathbf{D}_{n\times m}$ at each time stamp (i.e., tracking interval) and set a threshold parameter ε.
2) $\forall i$ $(i = 1, \ldots, n)$ initiate a sufficiently large value *minV* and an index $k(= 0)$.
3) $\forall j$ $(j = 1, \ldots, m)$ compute D_{ij} as the Euclidean distance between ith color-detected and jth motion-detected blobs.
4) Compare D_{ij} with *minV* and ε; update *minV* (with D_{ij}) and k (with j) only if $D_{ij} \le \min\{minV, \varepsilon\}$.
5) If $j < m$, increment j by one and go back to step 3.
6) If $k > 0$ update the image coordinates for color-detected landmark i (i.e., u_i, v_i) with the motion-detected position of blob k; otherwise keep the same color-detected coordinates.
7) Remove k column from matrix $\mathbf{D}$, only if $k > 0$.
8) If $i < n$, increment i by one and go back to step 2.

There are two possible alternatives to report the landmark i's image location after running the assignment algorithm: to use either the output of color-detection algorithm, or the output of motion-detection algorithm after solving the assignment problem presented in (2). We have conducted experiments with both approaches and the results show a better performance when the corresponding motion-detected coordinates are being used. The reason is that using motion-detected coordinates will provide us with the same scale for image locations of both targets and landmarks, and hence eliminate a source of error, which is caused by detection algorithm. This comparison is presented in more detail in Section V.

B. Finding Real-World Location by UAV

Considering the image of the UGVs and the detected crowd from the UAV camera, we can set a perspective transformation matrix to compute the real-world location of the targets. The real-time localization algorithm we proposed in this paper, implements the following generic steps.

1) Extracting landmarks' real-world geographic locations (using GPS).
2) Finding the image locations of landmarks (applying color detection as outlined in Section IV-A1).
3) Extracting the image locations of targets (using the motion detection as outlined in Section III-A and solving the assignment problem as discussed in Section IV-A2).
4) Estimating the elements of transformation matrix between the two planes, using a system of equations [as in (7)].
5) Converting the image locations of the targets to their real-world geographic locations, using the computed transformation matrix [as in (12)].

This part presents the details of the last two steps. To estimate the transformation matrix, assume the unknown 2-D GIS location (longitude and latitude) of landmark i $(i = 1, \ldots, n)$ at time t is noted as $(x_i(t), y_i(t))$ and its reported image location through the UAV camera at the same time is $(u_i(t), v_i(t))$. Then, the equation in (3) applies for each of these landmarks

$$[U_i(t), V_i(t), W_i(t)]^{\mathrm{T}} = \mathbf{M}\big[x_i(t), y_i(t), 1\big]^{\mathrm{T}} \tag{3}$$

where $\mathbf{M}$ is the perspective transformation matrix, $(x_i(t), y_i(t), 1)^{\mathrm{T}}$ is the homogeneous coordinates of the landmark i at time t, and $(U_i(t), V_i(t), W_i(t))^{\mathrm{T}}$ are computed

as the homogeneous image coordinates at frame t through the following equations:

$$u_i(t) = U_i(t)\big/W_i(t), \quad v_i(t) = V_i(t)\big/W_i(t). \tag{4}$$

In this paper, the train elevation change in the UAV detection range is assumed 0, so that $z_i(t)$ (the third dimension of coordinates) for all the landmarks as well as crowd at all time (i.e., frames) is set to a constant (here, 0). The perspective transformation between the two planes (i.e., image and Earth) preserves the cross-ratio of the landmarks, and hence, we can rewrite $u_i(t)$ and $v_i(t)$ as

$$u_i(t) = \frac{ax_i(t) + by_i(t) + r}{px_i(t) + qy_i(t) + 1}, \; v_i(t) = \frac{cx_i(t) + dy_i(t) + s}{px_i(t) + qy_i(t) + 1} \tag{5}$$

where a, b, c, d, p, q, r, and s are elements of the three-by-three perspective transformation matrix $\mathbf{M}$ at time t which can be defined as

$$\mathbf{M} = \begin{bmatrix} a & b & r \\ c & d & s \\ p & q & 1 \end{bmatrix}. \tag{6}$$

Here, r and s mainly serve for translation and the other parameters are applied for the linear transformation part. In order to estimate these eight unknown parameters at each frame t, we need to set up at least eight linearly independent equations. Since each of the landmarks provides us with two equations, at least four noncollinear landmarks on the same plane are required. If the number of landmarks is n, then we can rearrange matrix $\mathbf{M}$ in a vector format $\mathbf{m} = [a, b, r, c, d, s, p, q]^{\mathrm{T}}$, so that (7) is the system of equations with eight unknowns in this problem

$$\begin{bmatrix} x_1(t) & y_1(t) & 1 & 0 & 0 & 0 & -x_1(t)u_1(t) & -y_1(t)u_1(t) \\ 0 & 0 & 0 & x_1(t) & y_1(t) & 1 & -x_1(t)v_1(t) & -y_1(t)v_1(t) \\ \vdots & & & & \ddots & & & \vdots \\ x_n(t) & y_n(t) & 1 & 0 & 0 & 0 & -x_n(t)u_n(t) & -y_n(t)u_n(t) \\ 0 & 0 & 0 & x_n(t) & y_n(t) & 1 & -x_n(t)v_n(t) & -y_n(t)v_n(t) \end{bmatrix} \mathbf{m} = \begin{bmatrix} u_1(t) \\ v_1(t) \\ \vdots \\ u_n(t) \\ v_n(t) \end{bmatrix} \tag{7}$$

The solution to this $\mathbf{Am} = \mathbf{k}$ system is obtainable at each frame as follows, depending on the number of landmarks.

1) If the number of landmarks is less than four ($n < 4$), the system of equations would be underdetermined and hence, it has either no solution or infinitely many solutions.
2) If the number of landmarks is exactly four ($n = 4$), there could be an exact solution to the system of linear equations, which can be easily computed as

$$\mathbf{m} = \mathbf{A}^{-1}\mathbf{k}. \tag{8}$$

3) If the number of landmarks is greater than four ($n > 4$), we can use homogeneous least squares method, where $\mathbf{m}$ is obtained through

$$\mathbf{m} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{k}. \tag{9}$$

However, considering the numerical nature of the proposed problem and the rank of matrix $\mathbf{A}$ (i.e., 8) in this system of equations, using the inverse method would not be appropriate. The reason is that computing the matrix determinant would be very expensive, specifically when the matrix is near singular, and hence by applying (8) or (9) for solving the system, the result will not be very stable numerically. Therefore, we apply the Gaussian elimination with row operations and back substitution as a less expensive and more robust approach in our proposed algorithm to get the best solution. A heuristic applied in this algorithm is that, we first switch the rows of $\mathbf{A}$ (or $\mathbf{A}^{\mathrm{T}}\mathbf{A}$) matrix, so that none of the elements on its main diagonal would be 0. This helps to get a consistent and reliable solution to $\mathbf{Am} = \mathbf{k}$ (or $\mathbf{A}^{\mathrm{T}}\mathbf{Am} = \mathbf{A}^{\mathrm{T}}\mathbf{k}$) system of equations.

Knowing elements of transformation matrix $\mathbf{M}$ at time t, we can transform the image location of any detected target at that video frame to its real-world location on the same plane as landmarks. Considering $(x'(t), y'(t))$ as the notation for unknown real-world geographic (GIS-based) location of a detected crowd at frame t, we can convert its image location $(u'(t), v'(t))$ to the homogeneous coordinates. The resulting perspective transformation with the homogeneous coordinates would be as

$$\left[u'(t)W'(t), v'(t)W'(t), W'(t)\right]^{\mathrm{T}} = \mathbf{M}\left[x'(t), y'(t), 1\right]^{\mathrm{T}} \tag{10}$$

where constant $W'(t)$ could be any arbitrary number due to homogeneity. Here we consider $W'(t)$ as in (11) for consistency and simplicity of the solution steps. Again, using homogeneous matrix inverse, the unknown GIS location of the target at time t is obtainable as (12)

$$W'(t) = \frac{1}{\mathrm{M}^{-1}_{(3,1)}u'(t) + \mathrm{M}^{-1}_{(3,2)}v'(t) + 1} \tag{11}$$

$$\left[x'(t), y'(t), 1\right]^{\mathrm{T}} = \mathbf{M}^{-1}\left[u'(t)W'(t), v'(t)W'(t), W'(t)\right]^{\mathrm{T}}. \tag{12}$$

The output of (12) would be estimated longitude $(x'(t))$ and latitude $(y'(t))$ of the detected target (i.e., crowd) as the first and second elements of the solution vector. Since these computations are not very expensive, they can be implemented in real-time on a frame-by-frame basis (except for computations discussed in Section IV-A2), and hence, UAV detects the real-world locations of the targets to plan its path for tracking the crowd. The resulted locations will also be sent to the UGVs, so that they can use it in their path planning for next tracking interval, which is covered in detail in [6]. The performance of the proposed localization algorithm will be discussed in detail in the next section.

C. Finding Real-World Location by UGV

Since both the UGVs and the detected individuals are located on the same plane, and the average size of the targets (here, adult human) can be presumed known, we use triangle similarity for a pinhole camera model to get a good estimation of target's distance to the UGV camera. However, we need to first calibrate the camera for finding its real focal length f in terms of pixels, asking a human of height H stands in front

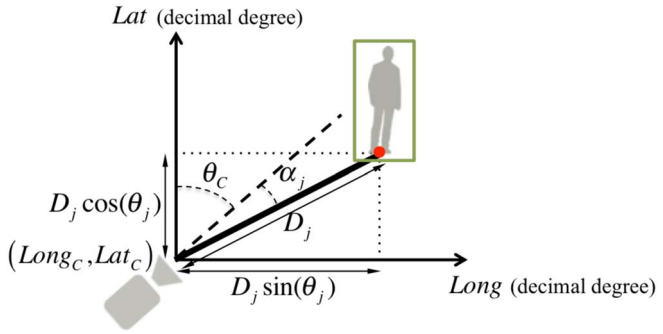


Fig. 2. Illustration of the relationships between key parameters of localization method for UGV.

of the camera at a known distance D and measuring its pixel-wise size in the image, h. The camera's calibrated focal length is then computed as

$$f = D\left(h/H\right). \tag{13}$$

Now, knowing the camera's calibrated focal length (f) and assuming the average height of adult human $\bar{H}$, estimation of target j's distance to the UGV camera (D_j) would be trivial, considering its image size (h_j) as an output of the HOG-based detection, in pixels.

In this paper, we have equipped the UGV with a GPS sensor, which embraces an electronic compass; hence, estimating the target j's geographical location based on D_j is straightforward. Assume the pose of the UGV camera as $(\text{Long}_C, \text{Lat}_C, \theta_C)$, where the first two arguments are the UGV longitude and latitude, and the third element is the camera orientation (i.e., its lens angle with respect to the North Pole), reported by the GPS. Then, the geographic location of the jth target $(\text{Long}_j, \text{Lat}_j)$ would be estimated as

$$\left(\text{Long}_j, \text{Lat}_j\right) = \left(\text{Long}_C + D_j \sin\left(\theta_j\right), \text{Lat}_C + D_j \cos\left(\theta_j\right)\right) \tag{14}$$

where θ_j is the modified orientation of target j and is computed as $\theta_C + \alpha_j$. For the jth target, α_j is estimated as (15), based on the horizontal deviation of the target j's detected center from the image center (denoted as dev_j). Depending on the position of jth detected target with respect to the image center, dev_j may take positive values (to the right of the image center) or negative values (to the left of the image center)

$$\alpha_j = \tan^{-1}\left(\text{dev}_j / f\right). \tag{15}$$

Fig. 2 illustrates the relationships between the key-parameters mentioned in this section. Note that D_j in (14) should be converted to decimal degrees beforehand, in order to conform to the longitude and latitude measures. Hence, it is divided by a factor K, which is a constant value (111 320 m) for the latitude argument; however, this value decreases accordingly for the longitude argument, as we move from the Equator to the North Pole.

V. Testbed and Experiments

The real testbed to run experiments in this paper includes a custom-build quadcopter as the UAV, and a 1/16-scale remote control car as a UGV, which are presented in [6] in more detail. These UVs are equipped with onboard sensors (e.g., GPS and camera) as well as powerful computers for real-time processing of the proposed vision-based detection and localization algorithms. The UAV onboard camera is also equipped with a Gimbal stabilization system for a smoother video stream and to minimize the motion detection errors.

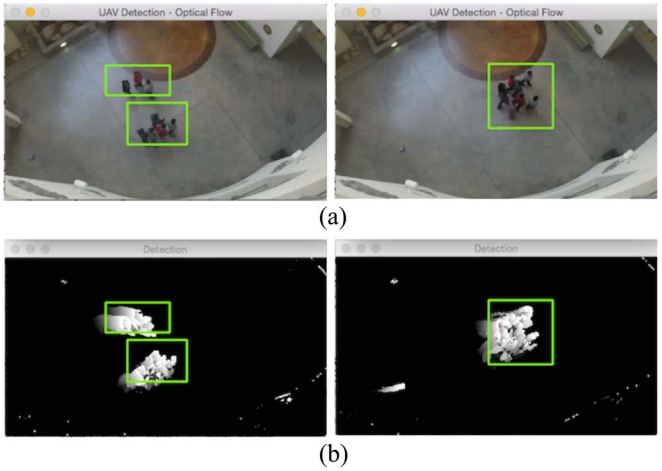


Fig. 3. Snapshots of UAV detection results when two groups of people are joining together from t_1(left) to t_2(right). (a) Final detection results with different bounding boxes considering different crowd's dynamics. (b) Silhouette images after segmentation for the same time stamps, showing the UGV movement.

Moreover, we have developed an agent-based simulation model for designing experiment scenarios, considering the high cost and technical limitations in building numerous UGVs for a sole-hardware-based real testbed. The simulation platform used in this paper is Repast Simphony, an open-source Java-based platform (http://repast.sourceforge.net/repast_simphony.php), which obtains the GIS data (longitude, latitude, and altitude) from NASA World Wind package. This feature helps us verify the targets' estimated locations as results of localization algorithm, knowing their simulated geographic coordinates.

In order to test the detection algorithms, we set a series of experiments, including two crowds moving under different scenarios (e.g., joining, splitting, and randomly moving) with a UAV and a UGV following them at the same time. The captured videos are then processed using the two computer vision algorithms discussed in Section III (i.e., motion detection for UAV and human detection for UGV), in which the moving crowd/individuals are detected and represented by a bounding box around them. Fig. 3 shows two snapshots of applying UAV detection algorithm on an indoor scenario where two crowds of different sizes and with different movement velocities join together. As shown in the figure, the dimensions of the bounding boxes change over time with respect to the crowds' dynamics. Note that the detection parameters discussed in Section III-A are set so that the UGV is not detected as a moving target, despite its moving status. Moreover, to evaluate the performance of the proposed motion detection algorithm in the daylight conditions, we performed a low-altitude flight test with the UAV, for which the results are shown in Fig. 4.

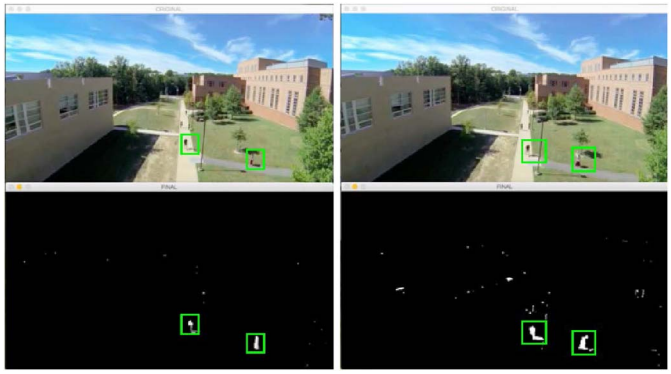


Fig. 4. Two snapshots of UAV detection results on a test flight outdoor. Since the flight altitude is low, the moving individuals are detected as targets.

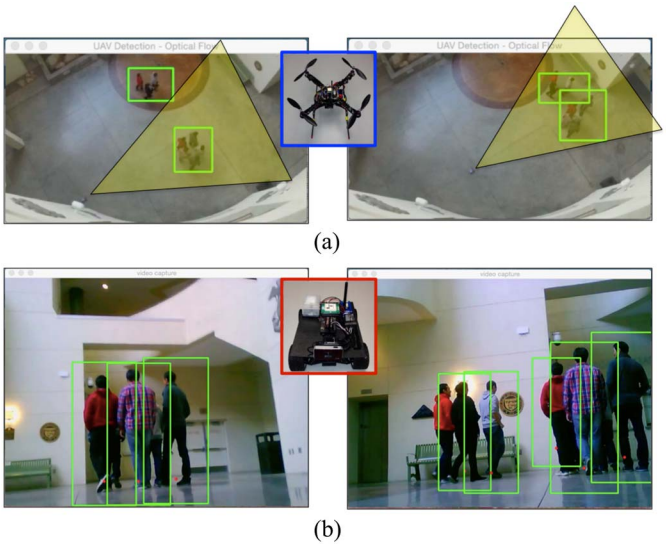


Fig. 5. Detection results for a scenario of two moving crowds. (a) Detected crowds by UAV at two different time stamps including UGV projected detection range. (b) Detected individuals by UGV at two different time stamps.

Finally, to illustrate the cooperation between the UVs, two snapshots of the results in two different time stamps from UAV camera and the corresponding snapshots from the UGV viewpoint are displayed in Fig. 5. As expected, the UAV detects those parts of the crowd moving together at the same speed in Fig. 5(a), as one unified target. However, in Fig. 5(b) the UGV tends to detect every nonoccluded individual, as a separate target. It is notable that the UGV camera only captures one of the crowds for the first time stamp [see Fig. 5(b), left image], due to its limited detection range [which is depicted in Fig. 5(a) for illustrative purposes]. Although the HOG algorithm is more sensitive to obstacles and its detection performance in covering the individuals in crowd's boundary is not always 100% (due to occlusion), it does not influence the system-level localization performance. The reason is that the UGV only needs the rough location of the crowd's center for path planning. The more exact crowd location is the output of the UAV global view localization.

In this paper, an agent-based simulation model is used for testing the localization algorithm, as mentioned earlier. It is because that: 1) less time and cost are needed for running simulation-based experiments compared to the real testbed in a border area and 2) it is convenient to verify estimated locations and run experiments with changing parameters. In the proposed simulation model, the UGVs and individuals

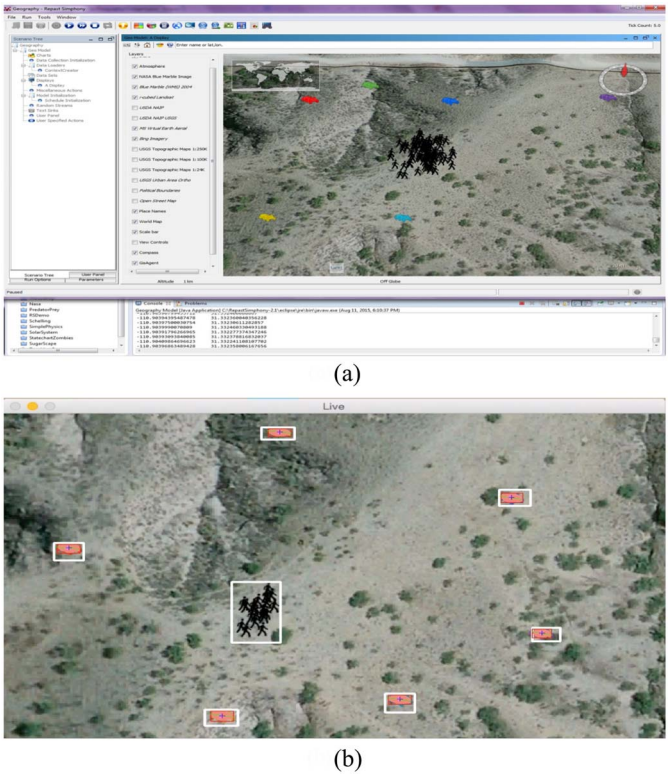


Fig. 6. Snapshots of (a) agent-based simulation for localizing the detected targets on the border area of Tucson, Arizona and (b) detection/localization algorithm runtime, using the agent-based simulation data as an input. The bounding boxes are the results of motion detection algorithm, while the blobs with a "+" on their center, represent the color detection results for landmarks.

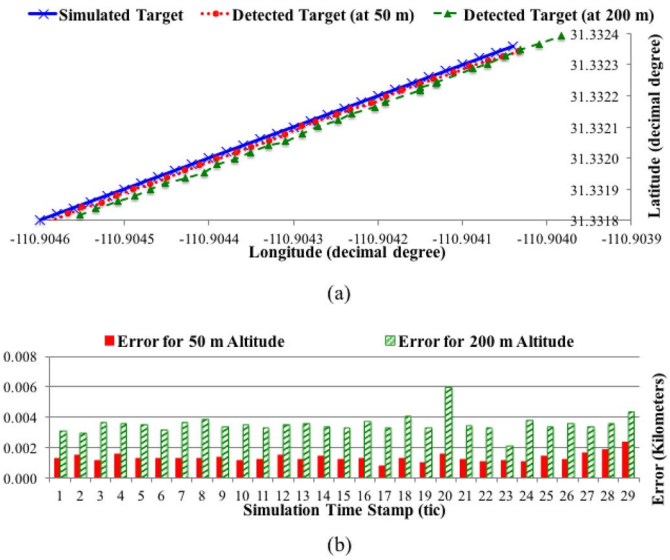


Fig. 7. Comparison of localization results using 200 and 50 m as UAV flight altitude, with respect to the simulation output at each time point. Based on (a) crowd's movement path over a frame sequence and (b) Euclidean distance error with the simulated locations.

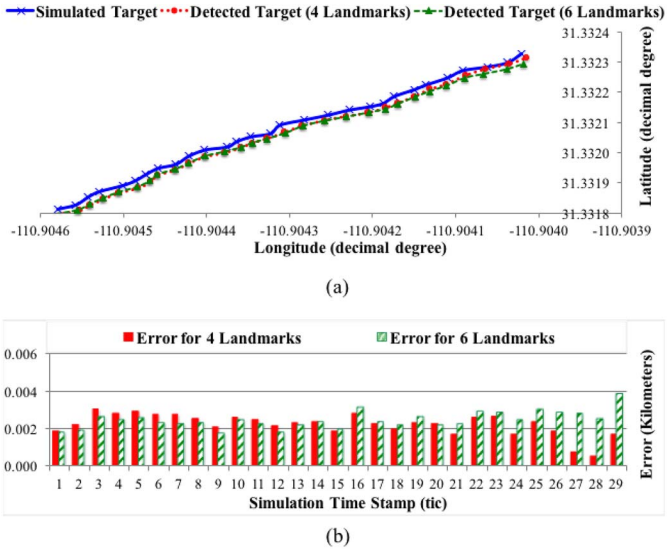


Fig. 8. Comparison of localization results using four landmarks versus six landmarks, with respect to the simulation output at each time point. Based on (a) crowd's movement path over a frame sequence and (b) Euclidean distance error with the simulated locations.

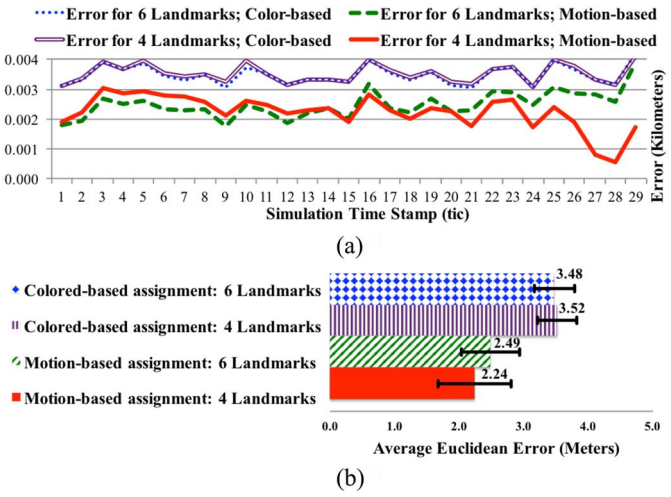


Fig. 9. Comparison of localization Euclidean distance error, considering four sets of alternatives: color-based assignment of four landmarks, color-based assignment of six landmarks, motion-based assignment of four landmarks, and motion-based assignment of six landmarks. Based on (a) frame sequence error and (b) error mean and variance.

in crowd act as independent types of agents to move along a simulated border area of Tucson, Arizona [see Fig. 6 (a)].

In this model, the randomly moving UGVs report their longitude and latitude at each time point (tracking interval) to the detection and localization system embedded in the UAV. Then, based on these data, the UAV controller estimates the perspective transformation matrix. Finally, using the image positions of the detected targets (i.e., crowds) as an output of the motion detection algorithm [see Fig. 6(b)], the crowds' real-world geographic locations in longitude and latitude decimal degrees are estimated. As shown in Fig. 6(b), although the moving UGVs are detected by the UAV motion detection algorithm, they are not reported as moving targets after applying the landmark differentiation method, described in Section IV-A2.

A series of experiments have been conducted using the simulation model. In these experiments, the values for the following parameters have been altered in order to evaluate their effects on the system performance.

1) The flight altitude.
2) The crowd's movement velocity.
3) The number of landmarks (i.e., UGVs).
4) The landmark assignment method (image coordinates).

We also changed the UGVs' initial locations and movement path to generate randomness in the experiments. The main performance measure in this paper is the average Euclidean distance error between the crowd's real GIS locations based on the simulation, and its estimated GIS coordinates as the output of detection and localization system for a series of simulated video frames. In general, the lower is the average error, the better the system performance will be. In this paper, we consider any localization error in the order of 3.5 m or less, as a desirable precision, since most of the high quality GPS receivers can provide a horizontal positioning accuracy of 3.5 m at a 95% confidence level, according to Federal Aviation Administration (FAA) real-world data [25].

To study the effects of flight altitude on the localization results, a number of experiments are performed with different simulated altitude values. The above mean sea level (AMSL) terrain elevation at our designated border area is about 1300 m on average; thus, we considered a maximum of 1500 m AMSL [approximately 200 m above ground level (AGL)] flight altitude in our experiment setting and decreased it step by step to comply with FAA regulations on public unmanned aircraft systems. Fig. 7 shows the comparison of localization results for 200 m AGL flight altitude (the maximum value allowed) versus 50 m AGL altitude (the minimum value to get a reasonably wide detection range). In this scenario the crowds are moving at the speed of (2e-5, 2e-5) decimal degrees per simulation time unit [equivalent to (2.22, 2.22) m per time-unit, which is rational for human walking speed]. As shown in the error charts, the localization performance is significantly improved as the altitude decreases; hence, the 3.1e-5 decimal degrees average Euclidean distance error (≈3.5 m) for 200 m scale reduces to 1.2e-5 decimal degrees average error (≈1.3 m) for 50 m, showing about 63% improvement in performance. This improvement is mainly due to increased resolution of the video at 50 m, and hence, a better performance of the motion detection algorithm in extracting moving targets' image locations. Moreover in a closer distance, the same region of interest in the image represents a smaller area of the real world, which results in a more compact bounding box around the moving blob (i.e., crowd). Hence, the image location of the moving crowd's center would be more accurate. The achieved results are satisfactory, since in the real testbed, the flight altitude would be lower to comply with the quadcopter design requirements and we can expect even more improvements.

The number of landmarks (i.e., UGVs) is also considered as another effective parameter. To test the effect of this parameter, we run another set of experiments, in which, the speed of crowd's movements is a random variable with uniform distribution of $U(1e-5, 3e-5)$ decimal degrees per

simulation-time-unit (equivalent to U(1.11, 3.34) m per time unit). Fig. 8 shows a graph of localization results comparison for using six landmarks versus four landmarks at 50 m flight altitude. In this set of experiments, the UGV agents start to move toward each other at a higher speed initially (to get closer to the detected crowd, as in a real scenario) and hence, the motion-detection error to extract their image locations is higher at the beginning. After time $t = 15$ when they are in a less distance toward the crowd, their movement speed reduces, subsequently. As mentioned earlier in Section IV-B, estimating the transformation matrix using four landmarks is expected to provide us with an exact solution, while using six landmarks reduces the sources of error. This expectation is met in our conducted experiments: In the first part of the experiment (up to $t = 15$) the six landmarks setting provides a better estimation, because it can compensate the miss-detection of landmarks' image locations; whereas, in the final part, the four landmarks setting is dominant. Using six landmarks, the average error for localization is 2.49 m, while it reduces to 2.24 m (equivalent to 10% improvement) for four landmarks. However, the improvements are not significant enough and using six landmarks can be beneficial to assure noncollinearity. Hence, we set the color detection parameters so that the UAV can switch to use more landmarks whenever the detection performance is low, to account for a portion of detection error. This may include situations where there are more than four UGVs in the UAVs detection range and their average speed pass a threshold.

As another set of experiments, we compared the performance of the localization algorithm considering the landmarks' image coordinates, based on two approaches mentioned in Section IV-A2: 1) the centers of color-detected blobs and 2) the centers of the corresponding motion-detected blobs, after solving the assignment problem. Moreover, we repeated the same experiments with sets of four and six landmarks to compare the algorithm performance for different alternatives. Fig. 9 depicts the Euclidean distance error trend for four possible combinations of these parameters. As shown in Fig. 9(a), the motion-based landmark detection outperforms the color-based landmark detection significantly. Furthermore, the alternative of motion-based detection for four landmarks, tend to perform better in target localization in long-term run; however, the same approach with six landmarks have a better performance at the initial periods, as mentioned before. The main source of performance difference between the four landmarks and six landmarks motion-based approaches is the motion-detection error. Therefore, both the color-based approaches using four and six landmarks show rather similar performance (similar pattern and magnitude for error) when detecting landmarks based on their colors only.

Fig. 9(b) visualizes a quantitative comparison of the system performances in terms of average Euclidean distance errors and data randomness (i.e., error variance) at 50 m flight altitude for the set of four proposed alternatives, which supports our previous argument. As shown in this figure, the motion-based landmarks assignment significantly outperforms the color-based approach and the four landmarks motion-based approach provide the best overall performance (the lowest average Euclidean distance) in the long-run. Moreover, the error variance of the motion-based approach is higher than that of the color-based approach due to influence of the motion-detection variation in localization. Table II provides a comparison of the accuracy of target localization algorithm proposed in this paper, versus a number of related reasearch papers in the literature.

TABLE II
COMPARISON OF THE LOCALIZATION ERROR IN THIS PAPER VERSUS THE LITERATURE

Research Work	Mean Error (m)	Max Error Reported (m)
This Work	2.2 ~ 3.5	<6
Sharma and Pack [26]	12 ~ 21	<32
Farmani *et al.* [27]	10 ~ 18	<25
Redding *et al.* [18]	11	<30
Sohn *et al.* [28]	10	NA
Barber *et al.* [29]	4	<15
Johansen [30]	3.9	<10.3

VI. CONCLUSION

In this paper, a comprehensive vision-based crowd detection and GIS localization algorithm for a cooperative team of one UAV and a number of UGVs is proposed. In our novel localization method, the UGVs (with their real-world geographic positions known via GPS) are considered as moving landmarks for a perspective transformation in order to convert the image locations of the detected targets into their GIS coordinates. A heuristic method for target localization by UGV is also proposed. Furthermore, a testbed consists of real UVs and an agent-based simulation model is developed to conduct experiments. We altered the key parameters of the system (e.g., flight altitude, number of landmarks, and landmark assignment method) and studied their impacts on the system-level performance.

Experimental results revealed the effectiveness of the motion detection algorithm for UAV, the human detection algorithm for UGV, and the real-world localization using the simulated UGVs as moving landmark. Future studies will be conducted to complete the testbed and use real UGVs as numerous moving landmarks and run further experiments to verify the detection/localization performance. As a future work, we also plan to improve the motion detection module and enhance the landmark assignment method for a more robust localization. As another potential research area, a complete framework for data fusion between cooperative UVs will be studied, in which the sensory data required for target localization is obtained from the UV platform that provides the most accurate one, considering its historical data.

REFERENCES

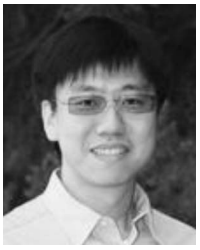
[1] D. Weatherington, "Unmanned aerial vehicles roadmap: 2002-2027," Off. Secr. Def., Washington, DC, USA, Tech. Rep. ADA414908, 2002.
[2] Z. Li *et al.*, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Trans. Syst., Man, Cybern., Syst*, vol. PP, no. 99, pp. 1–10, Aug. 2015.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.
[4] M. Paul, S. M. Haque, and S. Chakraborty, "Human detection in surveillance videos and its applications—A review," *EURASIP J. Adv. Signal Process.*, vol. 176, no. 1, pp. 1–16, 2013.
[5] S. Minaeian, J. Liu, and Y.-J. Son, "Crowd detection and localization using a team of cooperative UAV/UGVs," in *Proc. Ind. Syst. Eng. Res. Conf.*, Nashville, TN, USA, 2015, pp. 1–10.
[6] A. M. Khaleghi *et al.*, "A DDDAMS-based UAV and UGV team formation approach for surveillance and crowd control," in *Proc. Winter Simulat. Conf.*, Savannah, GA, USA, 2014, pp. 2907–2918.
[7] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.*, vol. 15, no. 1, pp. 123–141, 1995.
[8] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, "Coordination and navigation of heterogeneous UAVs-UGVs teams localized by a hawk-eye approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 2166–2171.
[9] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. B. Kosmatopoulos, "Multi-robot three dimensional coverage of unknown areas," *Int. J. Robot. Res.*, vol. 31, no. 6, pp. 738–752, 2012.
[10] M. Saska *et al.*, "Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Orlando, FL, USA, 2014, pp. 584–595.
[11] Y. Liu and Q. Dai, "A survey of computer vision applied in aerial robotic vehicles," in *Proc. Int. Conf. Opt. Photon. Energy Eng. (OPEE)*, vol. 1. Wuhan, China, 2010, pp. 277–280.
[12] A. Ferone and L. Maddalena, "Neural background subtraction for pan-tilt-zoom cameras," *IEEE Trans. Syst., Man, Cybern., Syst*, vol. 44, no. 5, pp. 571–579, May 2014.
[13] G. R. Rodríguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald, "A real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera," *Remote Sens.*, vol. 4, no. 4, pp. 1090–1111, 2012.
[14] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2179–2195, Dec. 2009.
[15] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall, 1998.
[16] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Proc. RSS*, Rome, Italy, 2015, pp. 1–9.
[17] X. Wang, H. Zhu, D. Zhang, D. Zhou, and X. Wang, "Vision-based detection and tracking of a mobile ground target using a fixed-wing UAV," *Int. J. Adv. Robot. Syst.*, vol. 11, no. 156, pp. 1–11, 2014.
[18] J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target localization from a fixed-wing miniature air vehicle," in *Proc. Amer. Control Conf.*, Minneapolis, MN, USA, 2006, pp. 2862–2867.
[19] C. Martínez, I. F. Mondragón, M. A. Olivares-Méndez, and P. Campoy, "On-board and ground visual pose estimation techniques for UAV control," *J. Intell. Robot. Syst.*, vol. 61, no. 1, pp. 301–320, 2011.
[20] H. Xiao, Z. Li, C. Yang, W. Yuan, and L. Wang, "RGB-D sensor-based visual target detection and tracking for an intelligent wheelchair robot in indoors environments," *Int. J. Control Autom. Syst.*, vol. 13, no. 3, pp. 521–529, 2015.
[21] L. Shao, J. Han, D. Xu, and J. Shotton, "Computer vision for RGB-D sensors: Kinect and its applications," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1314–1317, Oct. 2013.
[22] A. Mora, D. F. Glas, T. Kanda, and N. Hagita, "A teleoperation approach for mobile social robots incorporating automatic gaze control and three-dimensional spatial visualization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 3, pp. 630–642, May 2013.
[23] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 16–25, Sep. 2006.
[24] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 1994, pp. 593–600.
[25] J. William, "Global positioning system (GPS) standard positioning service (SPS) performance analysis report," Fed. Aviat. Admin., GPS Prod. Team, Washington, DC, USA, Tech. Rep. 86, 2014.
[26] R. Sharma and D. Pack, "Cooperative sensor resource management for multi target geolocalization using small fixed-wing unmanned aerial vehicles," in *Proc. AIAA Guid. Navig. Control*, Boston, MA, USA, 2013, pp. 1–11.
[27] N. Farmani, L. Sun, and D. Pack, "An optimal sensor management technique for unmanned aerial vehicles tracking multiple mobile ground targets," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Orlando, FL, USA, 2014, pp. 570–576.
[28] S. Sohn, B. Lee, J. Kim, and C. Kee, "Vision-based real-time target localization for single-antenna GPS-guided UAV," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 4, pp. 1391–1401, Oct. 2008.
[29] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo-location using a fixed-wing miniature air vehicle," *J. Intell. Robot. Syst.*, vol. 47, no. 4, pp. 361–382, 2006.
[30] D. L. Johansen, "Video stabilization and target localization using feature tracking with small UAV video," M.S. thesis, Dept. Elect. Comput. Eng., Brigham Young Univ., Provo, UT, USA, 2006.

Sara Minaeian received the B.S. degree in industrial engineering from the Iran University of Science and Technology, Tehran, Iran, in 2004, and the M.S. degree in industrial engineering from the University of Tehran, Tehran, in 2008. She is currently pursuing the Ph.D. degree in systems and industrial engineering with the University of Arizona, Tucson, AZ, USA.

From 2012 to 2014, she was a Teaching Assistant with the University of Arizona, where she has been a Research Assistant with the Computer Integrated Manufacturing and Simulation Laboratory since 2013. Her current research interests include computer vision, distributed multiparadigm simulation/emulation, and dynamic data driven application systems.

Ms. Minaeian is a Student Member of the Institute of Industrial Engineers and INFORMS.



Jian Liu received the B.S. and M.S. degrees in precision instruments and mechanology from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the M.S. degree in statistics and the Ph.D. degree in industrial and operations engineering from the University of Michigan, Ann Arbor, MI, USA, in 2006 and 2008, respectively.

He is an Associate Professor with the Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA. His current research interests include integration of manufacturing engineering knowledge, control theory, and advanced statistics for quality, reliability, and productivity improvement.

Dr. Liu is a member of the Institute of Industrial Engineers and INFORMS.

Young-Jun Son received the B.S. degree in industrial engineering from the Pohang University of Science and Technology, Pohang, Korea, in 1996, and the M.S. and Ph.D. degrees in industrial and manufacturing engineering from Pennsylvania State University, State College, PA, USA, in 1998 and 2000, respectively.

He is a Professor and the Department Head of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA, where he is also the Director of the Advanced Integration of Manufacturing Systems and Technologies Center. His current research interests include coordination of a multiscale, networked-federated simulation and decision model needed for design and control in manufacturing enterprise, renewable energy network, homeland security, agricultural supply networks, and social networks.

Dr. Son was a recipient of several research awards such as the Society of Manufacturing Engineers 2004 Outstanding Young Manufacturing Engineer Award, the Institute of Industrial Engineers (IIE) 2005 Outstanding Young Industrial Engineer Award, the Industrial Engineering Research Conference Best Paper Awards in 2005, 2008, and 2009, and the Best Paper of the Year Award in 2007 from *International Journal of Industrial Engineering*. He is a Fellow of the IIE.