

A DDDAMS-BASED UAV AND UGV TEAM FORMATION APPROACH FOR SURVEILLANCE AND CROWD CONTROL

Amirreza M. Khaleghi
Dong Xu
Sara Minaeian
Mingyang Li
Yifei Yuan
Jian Liu
Young-Jun Son

Christopher Vo
Jyh-Ming Lien

Systems and Industrial Engineering
University of Arizona
1127 E. James E. Rogers Way
Tucson, AZ 85721, USA

Computer Science
George Mason University
Nguyen Engineering Building
Fairfax, VA 22030, USA

ABSTRACT

The goal of this paper is to study the team formation of multiple UAVs and UGVs for collaborative surveillance and crowd control under uncertain scenarios (e.g. crowd splitting). A comprehensive and coherent dynamic data driven adaptive multi-scale simulation (DDDAMS) framework is adopted, with the focus on simulation-based planning and control strategies related to the surveillance problem considered in this paper. To enable the team formation of multiple UAVs and UGVs, a two stage approach involving 1) crowd clustering and 2) UAV/UGV team assignment is proposed during the system operations by considering the geometry of the crowd clusters and solving a multi-objective optimization problem. For the experiment, an integrated testbed has been developed based on agent-based hardware-in-the-loop simulation involving seamless communications among simulated and real vehicles. Preliminary results indicate the effectiveness and efficiency of the proposed approach for the team formation of multiple UAVs and UGVs.

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) have been widely used for surveillance and border patrol due to their capability in autonomous searching and detection and fitness for dull or dangerous tasks. For example, according to Haddal and Gertler (2010), a significant number of illegal immigrants were detected by the usage of UAVs and UGVs by the US border patrol.

Surveillance and crowd control using UAVs and UGVs is a complex issue involving various problems such as surveillance region assignment, crowd searching, detection, tracking, and motion planning. To resolve such a complex problem, a stream of research efforts has been pursued by the authors of this paper, such as system design, algorithm integration, and testbed implementation. For example, Figure 1 is the system overview of dynamic data driven adaptive multi-scale simulation (DDDAMS)-based planning and control framework for surveillance and crowd control, which have been studied by Khaleghi et al. (2013a, 2013b, 2014). The considered framework consists of 4 different modules including decision module for DDDAMS, integrated planner, integrated controller, and real system. The integrated controller collects vision-based data, processes it via crowd detection, crowd

tracking and motion planning modules, and generates commands to control the real system. This process is iterative until significant system performance deviations are detected, where the decision module for DDDAMS and the integrated planner are invoked to select the best control strategies in use for the future. The hardware and software testbed is constructed involving real and virtual UAVs/UGVs modeled via agent-based simulation. More details on the workflow, component interactions and integration efforts can be found in Khaleghi et al. (2013a, 2013b, 2014).

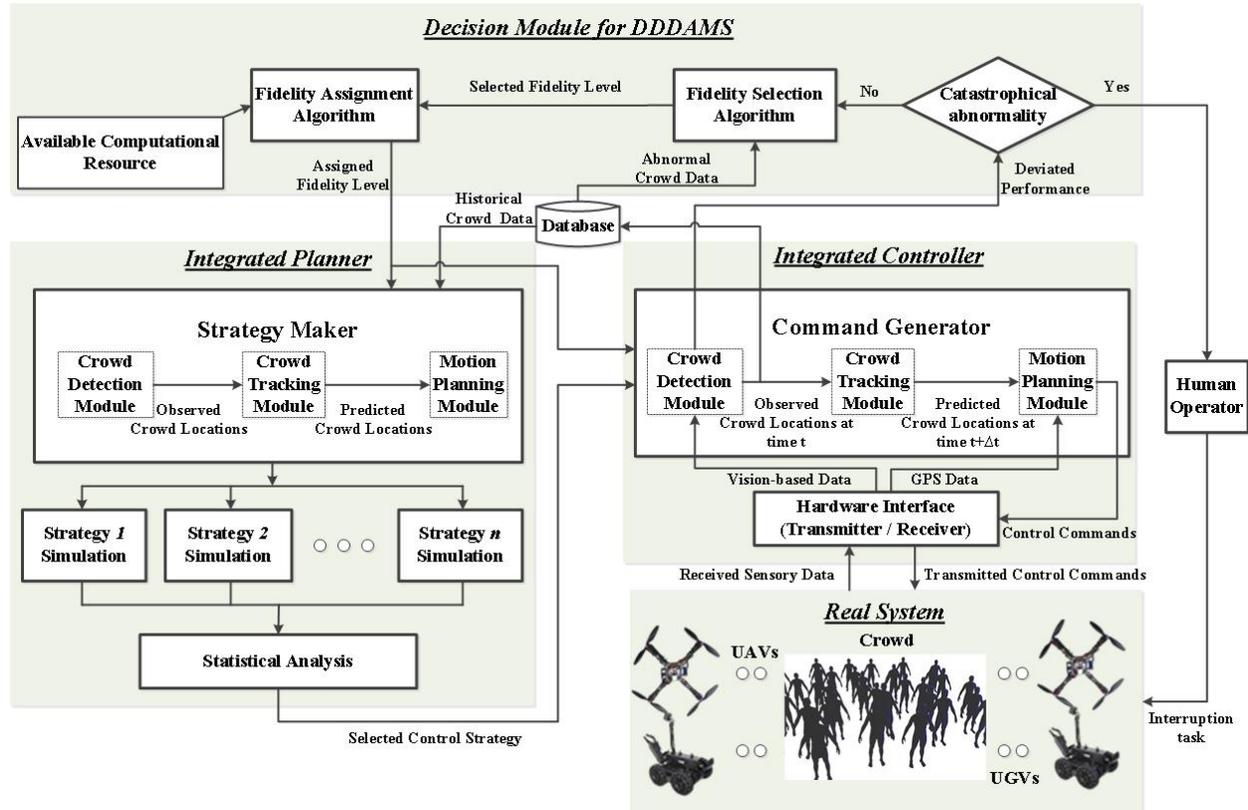


Figure 1: DDDAMS-based planning and control framework for UAV/UGV surveillance and crowd control (Khaleghi et al. 2013a, 2013b, 2014).

In this paper, the following scenarios are considered: a crowd of individuals are passing from a border region and a team of unmanned vehicles (UVs) (assuming one UAV and multiple UGVs as a team in this paper) performing surveillance have detected the crowd; from this time point, the crowd may continue to follow its designated path, or start to split into numerous groups to reduce the risk of being further followed and captured by border patrol agents. Under the latter case, the team of UVs have several options such as 1) team dispersion to capture different crowd groups (system performance may not be optimal) and 2) following one of the crowd groups and call for additional team/teams of UVs to follow other crowd groups. Figure 2 illustrates a sample splitting behavior of a crowd into two groups, where at time point $t + 2\Delta t$, the lower crowd group is out of the detection range of the current UV team. It is assumed that the UAVs flying at a high altitude observe a global view of the crowd without detailed information (low fidelity information, shown as the shaded cells in Figure 2), while the UGVs can observe individual dynamics in the crowd (high fidelity information, shown as the dots in Figure 2).

Given the described problem scenario, a DDDAMS-based UAV and UGV team formation approach is proposed in this paper, where different fidelity information is utilized to perform crowd clustering and UAV/UGV team assignment. First, mixture model-based clustering is employed based on the tracking

outputs of UAVs with aggregated information of UGVs. Then, in order to find the required number of vehicles for each cluster, three cases are formulated according to UGV's detection capabilities and different cluster geometries. Last, a resource allocation problem is presented to optimize the crowd coverage percentage and smoothness of team assignment, where the optimal control strategies depend on the weights of the objectives. This work serves as a starting point to study a full coordination of distributed UAVs and UGVs in the surveillance and crowd control mission. As noted, when the crowd splits, a new team of UAVs and UGVs will be used, instead of re-forming the current UAV/UGV team (current assumption). This motivates us to focus on determining the number of vehicles required for each team.

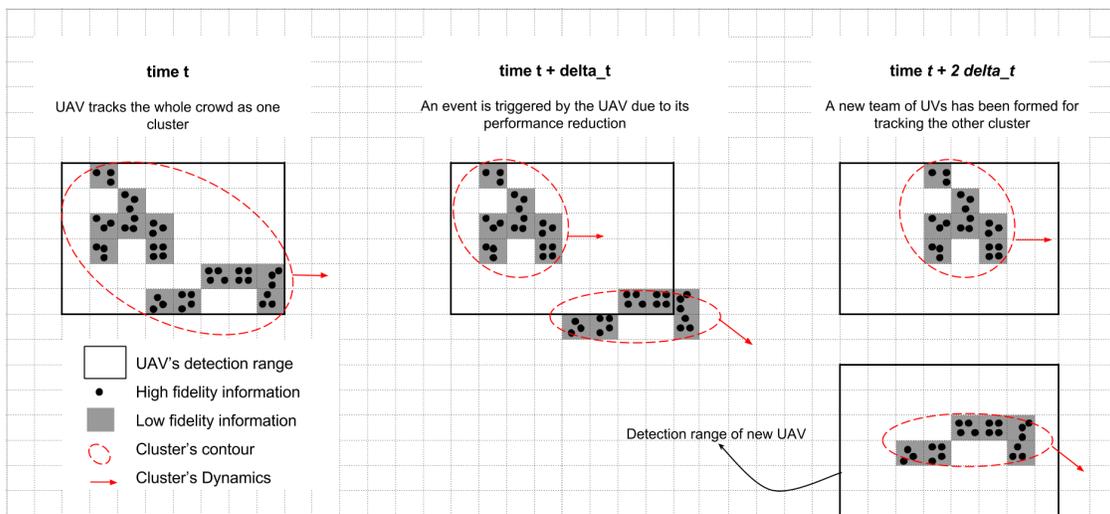


Figure 2: Demonstration of splitting behavior of the crowd and need of forming a new team.

The rest of this paper is organized as follows. Section 2 provides a background and literature review of different clustering methods (focusing on Gaussian Mixture clustering) as well as team formation in multi-robot systems. Section 3 discusses the crowd clustering and team assignment approach for surveillance and crowd control missions. Section 4 provides the details of the integrated agent-based hardware-in-the loop simulation and real UAV/UGV as the testbed and preliminary experiment results are presented. Section 5 concludes the work and discusses future research directions.

2 LITERATURE REVIEW

In the area of crowd tracking and surveillance, cluster analysis is an important issue. Clustering the individuals into several groups, instead of considering all the individuals as one group, helps motion tracking. Andrade, Blunsden, and Fisher (2006) applied unsupervised feature extraction using spectral clustering to identify the optimal number of models in representing normal motion patterns for better crowd event detection. Ge, Collins, and Ruback (2009) developed a bottom-up hierarchical clustering method to group individuals, which improves robustness of tracking individuals.

Many existing clustering methods are available, however in this paper Gaussian Mixture clustering is adopted due to its great flexibility in representing different patterns and its sound statistical model forms. Specifically, Gaussian Mixture clustering considers each observation having a probability of belonging to a certain cluster, making starting points of the algorithm less sensitive to the clustering results. Furthermore, according to the law of large numbers, Gaussian Mixture clustering conforms to reality well. Gaussian Mixture is implemented based on the Expectation–Maximization (E-M) algorithm, which is first introduced by Dempster, Laird and Rubin (1977) and further developed in the context of clustering

problem by Banfield and Raftery (1993). Further developments of Gaussian Mixture clustering are available in the literature, as summarized in Melnykov and Maitra (2010).

After the clustering algorithm generates the mean and covariance matrix (as outputs) of each cluster, the UAV/UGV team assignment will be invoked. The UAV/UGV team assignment in this paper is closely related to the multi-robot task allocation in cooperative multi-robot systems (CMRS), which mainly concerns with the assignment of robot groups to different tasks in order to cooperatively achieve the mission. Various multi-robot coordination schemes were proposed in the past several decades. Most of these schemes were domain-specific or architecture-dependent that were difficult to generalize and extend. Given this situation, Gerkey and Mataric (2004) classified the formal model of multi-robot task allocation (MRTA) based on three distinct dimensions: single-task or multi-task robots (ST or MT), single-robot or multi-robot tasks (SR or MR), and instantaneous assignment or time-extended assignment (IA or TA). The team formation in this work, also known as the coalition formation in robotics field, belongs to the ST-MR-IA category of MRTA.

Various questions are to be addressed in the coalition formation problem including how many vehicles (robots) are needed in each team, how to form a team, and how long a team should persist to perform corresponding tasks. Among the literature, Shehory and Kraus's algorithm (1998) was one of most popular algorithms in multi-agent coalition formation, which is based upon set covering and set partitioning heuristics. Specifically, two stages are involved in their algorithm: 1) Preliminary stage, which is responsible for calculating the values of all possible coalitions; 2) Algorithmic stage, which utilizes a greedy algorithm to select the matched coalition iteratively. Various extensions to the Shehory and Kraus's algorithm have been made. Vig and Adams (2006) proposed an innovative multi-robot coalition formation algorithm involving the concept of coalition imbalance, which demonstrated the effectiveness of the proposed algorithm via simulation and robot experiments.

3 SURVEILLANCE AND CROWD CONTROL VIA COOPERATIVE UAVS AND UGVs

3.1 Crowd Clustering Based on UAVs' and UGVs' Observations

Clustering is based on the tracking outputs of UAVs with aggregated information of UGVs. Let $\mathbf{x}_i(t)$ denote the state vector of occupied crowd cell i at time t , which may include the locations, velocities, and other related information. To perform the clustering of crowd agent observations, different clustering methods are available, as summarized in Section 2. In this paper, mixture model-based clustering is considered due to its great flexibility of representing different complex cluster patterns and its unique feature of modeling each cluster as a parametric distribution. Based on estimating the distribution parameters, many statistical quantities (e.g. confidence regions) can be constructed, which could provide meaningful, informative and compact information to UAVs' decision making. A generic formulation of mixture distribution can be represented as

$$g(\mathbf{x}(t)) = \int_{\Omega_0} w(\boldsymbol{\theta})f(\mathbf{x}(t) | \boldsymbol{\theta})d\boldsymbol{\theta}, \quad (1)$$

where $g()$ and $f()$ are the compound probability density function and component probability function; $w()$ is another probability density function governed by a collection of parameters $\boldsymbol{\theta}$ with the parameter space of Ω_0 , satisfying $w(\boldsymbol{\theta}) \geq 0$ and $\int_{\Omega_0} w(\boldsymbol{\theta})d\boldsymbol{\theta} = 1$. It is noted that the number of components in (1) is uncountable. In practice, a finite number of components with positive probabilities are considered and (1) is reduced into

$$g(\mathbf{x}(t)) = \sum_{j=1}^m w_j f_j(\mathbf{x}(t) | \boldsymbol{\theta}_j), \quad (2)$$

where w_j represents the mixing proportion of the j^{th} component and the overall probability density function, $g()$, is a convex combination of m component probability density functions, $f_j()$'s. In the context

of clustering, observations from cluster j is modeled with the parametric distribution of $f_j()$ with parameters θ_j . However, for any specific observation \mathbf{x} , its membership to a certain cluster is unknown and needs to be identified based on the available data. Such unknown membership issue makes the clustering as a classic unsupervised learning problem extensively addressed in the machine learning field.

Given tracking outputs at any time t , denoted as $\mathbf{D}_t = \{\mathbf{x}_i(t)\}_{i=1}^n$, mixture model-based clustering requires to estimate the unknown parameter vector θ and to allocate each individual observation to a specific cluster based on the resulting estimated model. To achieve parameter estimation, the maximum likelihood method can be used to maximize the log-likelihood function represented as

$$l(\theta | \mathbf{D}_t) = \sum_{i=1}^n \log \left(\sum_{j=1}^m w_j f_j(\mathbf{x}_i(t) | \theta_j) \right), \tag{3}$$

where $l()$ is the log-likelihood function. Notice that the first order conditions, i.e. $\partial l(\theta | \mathbf{D}_t) / \partial \theta = 0$, can be expressed as an addition to the n fractional terms, which makes it analytically difficult or even impossible to obtain the maximum likelihood estimate (MLE). Also, directly maximizing (3) based on a nonlinear optimization method is difficult due to its complex functional form. To overcome such difficulties, Expectation-Maximization (E-M) is typically employed, which is formally introduced by Dempster, Laird and Rubin (1977) and widely applied to different statistical problems (for details, see McLachlan and Krishnan (1996) and references therein). Basically, (3) is augmented with additional data, denoted as \mathbf{Z}_t , and a complete joint log-likelihood function of $l(\theta | \mathbf{D}_t, \mathbf{Z}_t)$ can be formulated. A two-step procedure, namely the expectation step and the maximization step, can be performed based on $l(\theta | \mathbf{D}_t, \mathbf{Z}_t)$ in an iterative fashion until convergence. For detailed convergence properties of the E-M algorithm, see Wu (1983). In the context of mixture distribution in (2), let $\mathbf{Z}_t = \{z_{ij}(t) : i = 1, \dots, n; j = 1, \dots, m\}$ and then the E-M algorithm can be specifically expressed as follows:

Initialize $w_j^{(0)}$ and $\theta_j^{(0)}, j=1, \dots, m$. For any iteration step $\phi = 1, \dots, \phi_{\max}$, repeat the following:

E-step: compute $\mathbf{E}[l(\theta | \mathbf{D}_t, \mathbf{Z}_t) | \mathbf{D}_t, \theta^{(\phi-1)}] = \sum_{i=1}^n \sum_{j=1}^m \mathbf{E}(z_{ij}(t) | \mathbf{D}_t, \theta^{(\phi-1)}) [\log w_j + \log f_j(\mathbf{x}_i(t) | \theta_j)]$;

M-step: $\theta^{(\phi)} = \arg \max_{\theta} \mathbf{E}[l(\theta | \mathbf{D}_t, \mathbf{Z}_t) | \mathbf{D}_t, \theta^{(\phi-1)}]$.

It is noted that by the theorem of Bayes,

$$p_{ij} = \mathbf{E}(z_{ij}(t) | \mathbf{D}_t, \theta^{(\phi-1)}) = \frac{w_j^{(\phi-1)} f_j(\mathbf{x}_i(t) | \theta_j^{(\phi-1)})}{\sum_{j=1}^m w_j^{(\phi-1)} f_j(\mathbf{x}_i(t) | \theta_j^{(\phi-1)})}, \tag{4}$$

where $z_{ij}(t)$ is an augmented binary random variable corresponding to $\mathbf{x}_i(t)$ and p_{ij} denotes the associated conditional expectation. If $\mathbf{x}_i(t)$ is in cluster j , $z_{ij}(t)=1$; otherwise, $z_{ij}(t)=0$. As described above, E-step can be evaluated analytically but the M-step may involve numerical maximization. In this paper, Gaussian-mixture is considered due to its great flexibility but more importantly, it yields closed form solution in the M-step. Specifically, assuming normality, $f_j(\mathbf{x}(t) | \theta_j) = N(\mathbf{x}(t) | \mu_j(t), \Sigma_j(t))$, where $N()$ represents the multivariate normal density with mean vector $\mu_j(t)$ and covariance matrix $\Sigma_j(t)$, closed-form solutions in the M-step can be explicitly given by

$$\mu_j^{(\phi)}(t) = \frac{\sum_{i=1}^n q_{ij} \mathbf{x}_i(t)}{\sum_{i=1}^n q_{ij}}, \quad j=1, \dots, m, \tag{5}$$

$$\Sigma_j^{(\phi)}(t) = \frac{\sum_{i=1}^n q_{ij} [\mathbf{x}_i(t) - \mu_j^{(\phi)}(t)][\mathbf{x}_i(t) - \mu_j^{(\phi)}(t)]^T}{\sum_{i=1}^n q_{ij}}, \quad j=1, \dots, m, \tag{6}$$

$$w_j^{(\phi)} = \frac{\sum_{i=1}^n q_{ij}}{n}, \quad j=1, \dots, m. \quad (7)$$

With the aforementioned E-M algorithm, MLE of the Gaussian-mixture model can be approximately obtained as $\boldsymbol{\mu}_j^{(\phi_{\max})}(t)$, $\boldsymbol{\Sigma}_j^{(\phi_{\max})}(t)$ and $w_j^{(\phi_{\max})}$, where ϕ_{\max} is the maximum iteration step when the algorithm is converged. Based on (4), the membership of any observation $\mathbf{x}_i(t)$ can be determined as: $\mathbf{x}_i(t)$ belongs to cluster j^* , where $j^* = \arg \max_j \mathbf{E}(z_{ij}(t) | \mathbf{D}_i, \boldsymbol{\theta}^{(\phi_{\max})})$. The number of clusters can be predetermined based on the total available number of UAVs and UGVs. It can also be determined through the available data, which are addressed in Melnykov and Maitra (2010). Thus, the outputs of clustering will give the number of clusters and model parameters (i.e. mean vector and covariance matrix) of each individual cluster.

3.2 UAV/UGV Team Assignment for Crowd Clusters

In this section, we discuss the team size and team assignment in case of crowd splitting into different clusters. As mentioned earlier, a team in this work consists of one UAV and multiple UGVs working together for performing surveillance mission. In this case, determining the team size responds to assigning different number of UGVs in the team. Thus, the upper and lower bounds of the required number of UGVs for coverage of each cluster are developed first considering the geometry of each cluster. Next, a resource allocation problem is formulated to determine the optimal number of UGVs to be allocated to each cluster.

In this paper, the same cameras (or lenses) are used for UAVs and UGVs, although the performance of the vision modules is not the same. Grocholsky et al. (2006) discussed that UAVs commonly have a wider field of view (FOV), faster coverage of the search area, and lower resolution for detection and localization. However, UGVs have a narrower (or obscured) FOV, slower coverage of the search area, and better resolution. This is a major motivation to employ teams of cooperative UAVs and UGVs in this paper. Along with the vision system characteristics and visual sensor constraints, information sharing in the assigned team of UAVs and UGVs provides a way for estimating the uncertainty in the detection and localization algorithms.

The detection range for the UAV is a function of both FOV and altitude, which can be computed as

$$DR^{(A)} = 2h^{(A)} \tan(FOV/2), \quad (8)$$

where $DR^{(A)}$ represents the 1-D length of the detection range, $h^{(A)}$ is the flying altitude of UAV, and FOV is the field of view. For the UGV, the detection range would be only considered in one dimension for the horizontal FOV. Furthermore, a safety distance between UGV and the target crowd is assumed in this work to ensure the UGV's safety and persistent coverage. The detection range for UGV ($DR^{(G)}$) can be formulated as

$$DR_{\min}^{(G)} = 2 h_{\min}^{(G)} \tan(FOV/2) \leq DR^{(G)} \leq 2 h_{\max}^{(G)} \tan(FOV/2) = DR_{\max}^{(G)}, \quad (9)$$

where $DR_{\min}^{(G)}$ is the minimum detectable range based on the predefined safety distance, $h_{\min}^{(G)}$ is the safety distance, $h_{\max}^{(G)}$ is the longest distance that the camera can detect individuals through the detection algorithm, and $DR_{\max}^{(G)}$ is the maximum detection range. A graphical illustration of such relationship is presented in Figure 3(a).

Based on the safety distance and detection range defined above, the lower and upper bounds of the number of UGVs for complete coverage of each cluster has been developed considering the geometry of each cluster. Since the boundary of the cluster is always projected as an ellipse using GMM clustering algorithm (Section 3.1), we will use the geometry of the multivariate normal distribution to determine the

number of required UGVs for each cluster. In this regard, the longest diameter of the ellipse is in the direction of the eigenvector associated with the larger eigenvalue, and the other eigenvector represents the direction of the shortest diameter of the cluster. The lengths of these diameters are also defined as

$$d_{i,L} = 2\sqrt{\lambda_1 \chi_{2,\alpha}^2} \quad \text{and} \quad d_{i,S} = 2\sqrt{\lambda_2 \chi_{2,\alpha}^2}, \quad (10)$$

where $d_{i,L}$ and $d_{i,S}$ are the i^{th} ellipse's longest and shortest diameter lengths, respectively; λ_1 and λ_2 are its larger and smaller eigenvalues, and α is the significance level of the clustering algorithm. Furthermore, defining $EDD = h_{\max}^{(G)} - h_{\min}^{(G)}$ as the *Effective Detection Depth* of the UGV's camera, three cases can be considered to compute the required number of UGVs for the i^{th} cluster, V_i , by comparing the geometry of the different cluster types and EDD (see Figure 3(b)).

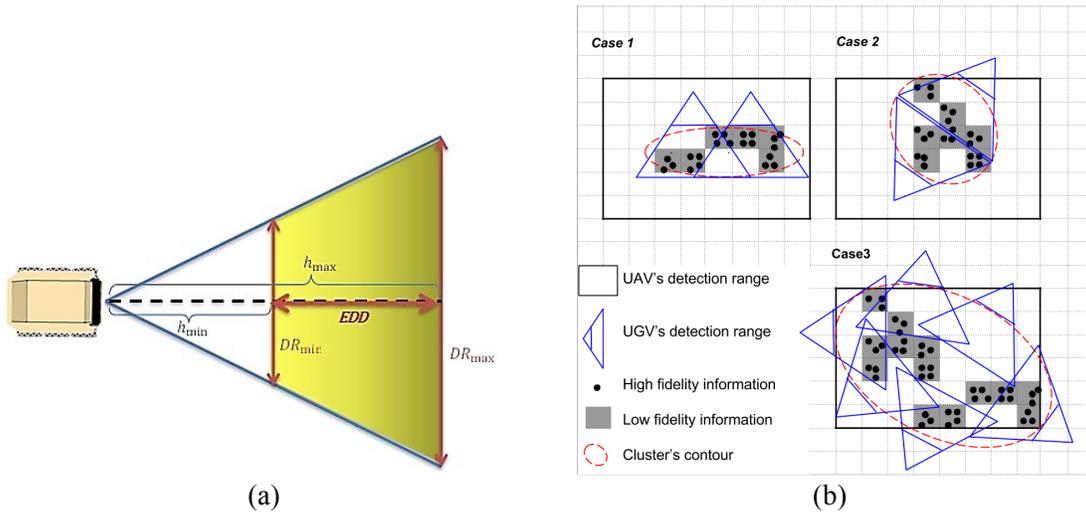


Figure 3: (a) Illustration of UGV's detection range; (b) Three cases of crowd clusters.

Case 1: $d_{i,S} \leq EDD$, which means the i^{th} cluster is narrow enough to be covered by UGVs located along a straight line. Under this case, the lower bound (denoted as $\lfloor V_i \rfloor$) and upper bound (denoted as $\lceil V_i \rceil$) of the required number of UGVs for the i^{th} cluster are computed as

$$\lfloor V_i \rfloor = d_{i,L} / DR_{\max}^{(G)} \leq V_i \leq d_{i,L} / DR_{\min}^{(G)} = \lceil V_i \rceil. \quad (11)$$

Case 2: $EDD < d_{i,S} \leq 2 EDD$, which means the i^{th} cluster is wider than in Case 1, but is smaller than the double of EDD . Thus, the crowd can be covered by two lines of UGVs located along each side of the cluster. The lower and upper bounds for the required number of UGVs for the i^{th} cluster are given as

$$\lfloor V_i \rfloor = 2d_{i,L} / DR_{\max}^{(G)} \leq V_i \leq 2d_{i,L} / DR_{\min}^{(G)} = \lceil V_i \rceil. \quad (12)$$

Case 3: $d_{i,S} > 2 EDD$, which means the i^{th} cluster's shortest diameter is larger than the double of EDD , so it cannot be covered by only one or two lines of UGVs. In this case, the ellipse's perimeter ($P_i = 2\pi\sqrt{(\lambda_1 \chi_{2,\alpha}^2 + \lambda_2 \chi_{2,\alpha}^2)/2}$) is considered to compute the lower and upper bounds for the required number of UGVs for the i^{th} cluster as

$$\lfloor V_i \rfloor = P_i / DR_{\max}^{(G)} \leq V_i \leq P_i / DR_{\min}^{(G)} = \lceil V_i \rceil. \quad (13)$$

It is noted that in all three cases, using the lower bound for the number of UGVs may affect the coverage performance; therefore, there is always a trade-off between assigning smaller group of UGVs to

a cluster and covering the whole boundary, considering the fact that not all the boundary parts include individuals. Also, in the third case, the team of UGVs is designated around the ellipse, which results in some inner parts of the crowd uncovered. However, these uncovered crowds will need to come near to the boundary of cluster in order to escape or split, so the UGVs can still follow them.

In real scenarios, due to the limited number of resources (i.e. UGVs), it may not be always possible to assign the required number of UGVs (V_i) to each cluster. Given the decision variable TS_i as the number of UGVs assigned to the i^{th} cluster, the number of occupied cells (i.e. low fidelity information) of the i^{th} cluster, o_i , and the total number of available UGVs, N_G , we assume for each cluster, the inequality $TS_i \leq o_i$ holds, meaning that the number of assigned UGVs to a cluster cannot exceed the number of occupied cells in that particular cluster. Furthermore, B_i is defined as $\min\{o_i, V_i\}$. If $\sum_{i=1}^C B_i \leq N_G$ then $TS_i = B_i$. If $\sum_{i=1}^C B_i > N_G$, where C is the number of clusters, then strategy maker should determine how to distribute UGVs among different clusters. In this case, TS_i / V_i shows the proportion of the required number of UGVs assigned to the i^{th} cluster, and the term $(TS_i / V_i) o_i$ gives the coverage of the entire occupied cells in the i^{th} cluster. So the utility function (f_1) is defined as the normalized overall coverage of all clusters as follows:

$$f_1 = \sum_{i=1}^C \frac{TS_i}{V_i} o_i / \sum_{i=1}^C o_i. \quad (14)$$

However, UGVs might not be uniformly assigned among all the clusters, which can result in very few or no assigned UGVs in some clusters. To smooth the UGV assignment among all the clusters f_2 is defined as *smoothness index*

$$f_2 = M - m, \quad (15)$$

where $m \leq TS_i / V_i \leq M$. In real scenarios the strategy maker may need to adjust the team assignment to maximize the utility function, or minimize the smoothness index, or optimize a composite objective including both. By using the weighting method the multi-objective optimization problem reduces to the following:

$$\begin{aligned} & \max \quad \alpha f_1 - (1 - \alpha) f_2 \quad (16) \\ & \text{subject to} \\ & \quad \sum_{i=1}^C TS_i \leq N_G, \\ & \quad 0 \leq TS_i \leq B_i, \\ & \quad m \leq TS_i / V_i \leq M, \\ & \quad TS_i \text{ integer,} \end{aligned}$$

where $0 \leq \alpha \leq 1$. If $\alpha = 0$, then only f_2 is optimized and if $\alpha = 1$, then only f_1 is considered. Here the first constraint implies that the total number of available UGVs cannot exceed resources and the second constraint determines the lower and upper bound of the decision variable. This optimization problem can be solved using integer linear programming methods, which result in the optimal number of UGVs assigned to different clusters.

Altering the value of α in (16) results in different control strategies for conducting the team assignment. All solutions are Pareto optimal. The control strategies employed here are consistent with what was studied by the authors previously (Khaleghi et al. 2013a). As α gets closer to 1, the overall coverage of all clusters will be increased, but the UGVs may not be assigned uniformly among all the

clusters. On the other hand, by decreasing the value of α closer to 0, UGVs will be assigned uniformly among the clusters, but the overall coverage may not be optimal. It should be noted that the solutions to the optimization problem depend on the selection of V_i (lower and upper bounds are given in previous discussion). Investigating the impact of V_i is left for future research.

4 INTEGRATED TESTBED AND EXPERIMENTS

We developed an autonomous UAV and UGV platform as a testbed that is robust, flexible and capable of carrying a variety of sensors in both indoor and outdoor environments. While commercial UAV and UGV platforms are available, most are either quite expensive; do not offer sufficient payload capacity or robustness for operating different tasks; or use strictly proprietary components and software. To meet our goals, we developed a customized UAV and UGV platform (see Figure 4(a)) using inexpensive off-the-shelf parts and open-source software components.

The computational architecture used in both UAV and UGV platforms consists of a low-level and a high-level computer. The low-level computer is responsible for the control of the vehicle's hardware components such as motors and servos. It is also responsible for vehicle stabilization and simple waypoint-based navigation. For this reason, we use an open-source embedded microcontroller developed by 3dRobotics called the ArduPilotMega (APM). Previously, this autopilot unit was designed for fixed-wing hobby aircraft but has been adapted to use in multicopters. It features an ATmega2560 programmable microcontroller circuit, which processes data from an integrated inertial measurement unit (IMU) measuring 3-axis translational acceleration, rotational velocity, and magnetic field direction. It also accepts peripheral sensor inputs from a global positioning system (GPS) sensor and a sonar sensor, as well as radio control inputs from a 2.4Ghz radio receiver, and serial command inputs using a standardized protocol called MAVLINK. Among the possible serial MAVLINK commands, waypoint commands or direct "joystick" controls are usually used to instruct the vehicle to move to given coordinates (latitude/longitude). These commands come from the high level computer, which is responsible for planning and deliberation. This high level computer is a commercially available microcomputer platform called ODROID, which runs on a powerful quad-core ARM processor and installs a Linux-based operating system.

Distributed communication is essential for the testbed. Each agent, whether it is a UAV or UGV, must be able to communicate information with the other agents to perform coordinated tasks, and a ground station should be able to monitor most of the activities among agents in the system for safety and analysis purposes. The transmitted information will include the agent's estimated GPS location, task-specific status messages, and commands from the ground station. For the application in this paper, we have selected the Digi Xbee Pro 900 radio to be embedded in each UAV and UGV platform.

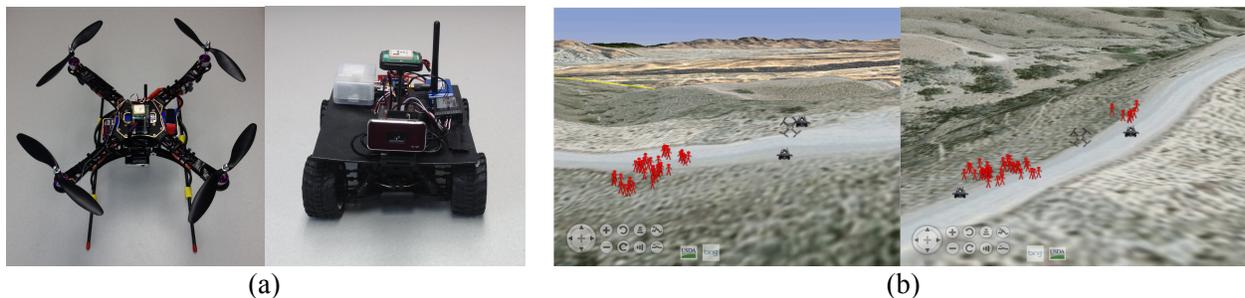


Figure 4: (a) Assembled UAV and UGV platform for the integrated testbed; (b) Demonstration of crowd splitting behavior in agent-based simulation.

For the experiment of this work, a similar situation demonstrated in Figure 2 has been considered. An agent-based hardware-in-the-loop simulation has been implemented in Repast Symphony utilizing the GIS data from NASA WORLD WIND (Khaleghi et al. 2013b) while communicating with simulated/real UAVs and UGVs. Figure 4(b) illustrates two snapshots of the agent-based hardware-in-the-loop simulation, where a team of UVs consists of one UAV and two UGVs tracking a crowd of 40 individuals moving together as one cluster. Then after 40 seconds, the crowd starts to split into two clusters with different number of individuals in each cluster. The performance (i.e. coverage percentage) of the UAV for 100 seconds has been illustrated in Figure 5(a). As the figure shows, the coverage percentage (i.e. ratio of number of the occupied cells to total number of cells in UAV’s detection range) starts to decrease after 40 seconds as the two crowds getting far from each other. During this time, the UAV attempts to track both crowds by selecting the mean of both clusters until 70 seconds. After this time, due to the distance between two clusters, UAV starts to track the cluster with more occupied cells. The model parameters of the two clusters (see Figure 5(b)) have been sent to the ground station for assigning UGVs required for each cluster.

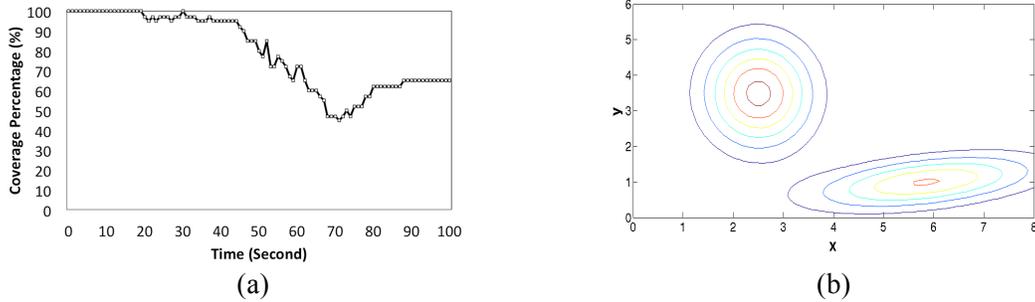


Figure 5: (a) Coverage percentage reduction due to crowd splitting; (b) Results of clustering algorithm.

A numerical experiment involving four different test scenarios has also been performed to investigate the effects of different cluster patterns and weight values on the number of UGVs assigned. To this end, each test scenario has been solved with two settings of weight values (i.e. $\alpha = 0.5$, $\alpha = 1$). As shown in Table 1, by setting α to 0.5, UGVs assigned to each cluster were more uniformly distributed comparing with $\alpha = 1$, where the UGVs were assigned in an imbalanced manner. Furthermore, the overall coverage of two clusters by setting α to 0.5 is less in comparison with the situation where α equals 1.

Table 1: Experimental results for UAV/UGV team assignment ($N_G = 8$).

Test scenarios	Cluster index (i)	o_i	V_i	$\alpha = 0.5$		$\alpha = 1$	
				TS_i	$\frac{TS_i}{V_i} o_i$	TS_i	$\frac{TS_i}{V_i} o_i$
Case 1	1	100	12	6	50	8	66.6
	2	10	2	1	5	0	0
Case 2	1	100	12	6	50	8	66.6
	2	30	5	2	12	0	0
Case 3	1	100	12	5	41.6	1	8.3
	2	60	7	3	25.7	7	60
Case 4	1	100	12	4	33.3	0	0
	2	90	10	4	36	8	72

5 CONCLUSIONS AND FUTURE WORKS

In this paper, a DDDAMS-based UAV and UGV team formation approach was presented involving crowd clustering and UV team assignment. The motivation comes from a crowd splitting scenario, where different teams of UVs are used to follow crowds. The proposed approach first captured the crowd dynamics via Gaussian Mixture clustering technique, followed by the decision of assigning UGVs to teams. An integrated testbed involving real and virtual UAVs/UGVs in simulation was developed, and experimental results demonstrated the effectiveness of the proposed approach.

As a future work, a time-based triggering mechanism of crowd clustering and team formation will be considered, where the problem tends to be more complex in the frequency of clustering and team assignments. We also plan to improve the selection of the required number of UGVs for each cluster (lower and upper bounds are given in Section 3.2) to make all stages of the process more efficient. Furthermore, the searching patterns and coordination issue between UAVs and UGVs will be addressed under the DDDAMS framework.

ACKNOWLEDGMENTS

This work is financially supported by the Air Force Office of Scientific Research under FA9550-12-1-0238 (A part of Dynamic Data Driven Application Systems (DDDAS) projects).

REFERENCES

- Andrade, E. L., S. Blunsden, and R. B. Fisher. 2006. "Modelling Crowd Scenes for Event Detection." In *Proceedings of the 18th International Conference on Pattern Recognition*, 1:175-178.
- Banfield, J. D., and A. E. Raftery. 1993. "Model-based Gaussian and Non-Gaussian Clustering." *Biometrics* 49(3): 803-821.
- Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B39* (1):1-38.
- Ge, W., R. T. Collins, and B. Ruback. 2009. "Automatically Detecting the Small Group Structure of a Crowd." In *Proceedings of the 2009 IEEE Workshop on Applications of Computer Vision*.
- Gerkey, B. P., and M. J. Mataric. 2004. "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems." *The International Journal of Robotics Research* 23(9):939-954.
- Grocholsky, B., J. Keller, V. Kumar, and G. Pappas. 2006. "Cooperative Air and Ground Surveillance." *IEEE Robotics & Automation Magazine* 13(3):16-25.
- Haddal, C. C., and J. Gertler. 2010. "Homeland Security: Unmanned Aerial Vehicles and Border Surveillance." Library of Congress, Washington DC, Congressional Research Service.
- Khaleghi, A. M., D. Xu, Z. Wang, M. Li, A. Lobos, J. Liu, and Y. J. Son. 2013a. "A DDDAMS-Based Planning and Control Framework for Surveillance and Crowd Control via UAVs and UGVs." *Expert Systems with Applications* 40(18):7168-7183.
- Khaleghi, A. M., D. Xu, A. Lobos, S. Minaeian, Y. J. Son, and J. Liu. 2013b. "Agent-Based Hardware-in-the-Loop Simulation for UAV/UGV Surveillance and Crowd Control System." In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 1455-1466. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Khaleghi, A. M., D. Xu, S. Minaeian, M. Li, Y. Yuan, C. Vo, A. Mousavian, J.-M. Lien, J. Liu, and Y. J. Son. 2014. "A Comparative Study of Control Architectures in UAV/UGV-Based Surveillance System." In *Proceedings of the 2014 Industrial and Systems Engineering Research Conference*, edited by Y. Guan and H. Liao. Institute of Industrial Engineers.
- McLachlan G., and T. Krishnan. 1996. *The EM Algorithm and Extensions*. New York: John Wiley & Sons.

- Melnykov, V., and R. Maitra. 2010. "Finite Mixture Models and Model-Based Clustering." *Statistics Surveys* 4:80-116.
- Shehory, O., and S. Kraus. 1998. "Methods for Task Allocation via Agent Coalition Formation." *Artificial Intelligence* 101(1):165-200.
- Vig, L., and J. A. Adams. 2006. "Multi-Robot Coalition Formation." *IEEE Transactions on Robotics* 22(4):637-649.
- Wu, C. J. 1983. "On the Convergence Properties of the EM Algorithm." *The Annals of Statistics* 11(1):95-103.

AUTHOR BIOGRAPHIES

AMIRREZA M. KHALEGHI is a Ph.D. student in the Department of Systems and Industrial Engineering at the University of Arizona. His research focuses on simulations, agent-based modeling, and mathematical modeling. He can be reached at amirreza@email.arizona.edu.

DONG XU is a Ph.D. candidate in the Department of Systems and Industrial Engineering at the University of Arizona. His research interests are in the areas of distributed simulation technology, supply chain modeling and game theory. He can be reached at dongxu@email.arizona.edu.

SARA MINAEIAN is a Ph.D. student in the Department of Systems and Industrial Engineering at the University of Arizona. Her research interests include distributed multi-paradigm simulation, computer vision and applications in healthcare systems. She can be reached at minaeian@email.arizona.edu.

MINGYANG LI is a Ph.D. student in the Department of Systems and Industrial Engineering at the University of Arizona. His research interests include quality and reliability engineering, Bayesian statistics, data mining and computational intelligence.

YIFEI YUAN is a Ph.D. student in the Department of Systems and Industrial Engineering at the University of Arizona. His research interests are Bayesian statistics and machine learning.

CHRISTOPHER VO is a Ph.D. student affiliated with the Motion and Shape Computing (MASC) group and the Autonomous Robotics Laboratory at George Mason University. His research is in the area of robustness, scalability, and performance in robotic motion planning.

JYH-MING LIEN is an Associate Professor in the Department of Computer Science and director of Motion and Shape Computing (MASC) at George Mason University. He can be reached at jmlien@cs.gmu.edu.

JIAN LIU is an Assistant Professor in the Department of Systems and Industrial Engineering at the University of Arizona. His research interests focus on the integration of manufacturing engineering knowledge, control theory and advanced statistics for quality, reliability and productivity improvement.

YOUNG-JUN SON is a Professor and Head of Systems and Industrial Engineering at the University of Arizona. His research focuses on the coordination of a multi-scale, networked-federated simulation and decision model needed for design and control in manufacturing enterprise, renewable energy network, homeland security, agricultural supply network, and social network. His email address is son@sie.arizona.edu.