Expert Systems with Applications 40 (2013) 7168–7183

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A DDDAMS-based planning and control framework for surveillance and crowd control via UAVs and UGVs

Amirreza M. Khaleghi, Dong Xu, Zhenrui Wang, Mingyang Li, Alfonso Lobos, Jian Liu, Young-Jun Son *

Systems and Industrial Engineering, The University of Arizona, Tucson, AZ 85721-0020, USA

ARTICLE INFO

Keywords:
DDDAMS
UAV
UGV
Fidelity
Agent-based simulation
Surveillance

ABSTRACT

A dynamic data driven adaptive multi-scale simulation (DDDAMS) based planning and control framework is proposed for effective and efficient surveillance and crowd control via UAVs and UGVs. The framework is mainly composed of integrated planner, integrated controller, and decision module for DDDAMS. The integrated planner, which is designed in an agent-based simulation (ABS) environment, devises best control strategies for each function of (1) crowd detection (vision algorithm), (2) crowd tracking (filtering), and (3) UAV/UGV motion planning (graph search algorithm). The integrated controller then controls real UAVs/UGVs for surveillance tasks via (1) sensory data collection and processing, (2) control command generation based on strategies provided by the decision planner for crowd detection, tracking, and motion planning, and (3) control command transmission via radio to the real system. The decision module for DDDAMS enhances computational efficiency of the proposed framework via dynamic switching of fidelity of simulation and information gathering based on the proposed fidelity selection and assignment algorithms. In the experiment, the proposed framework (involving fast-running simulation as well as real-time simulation) is illustrated and demonstrated for a real system represented by hardware-in-the-loop (HIL) real-time simulation integrating real UAVs, simulated UGVs and crowd, and simulated environment (e.g. terrain). Finally, the preliminary results successfully demonstrate the benefit of the proposed dynamic fidelity switching concerning the crowd coverage percentage and computational resource usage (i.e. CPU usage) under cases with two different simulation fidelities.

1. Introduction

Multi-vehicle systems consisting of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) have shown promises in recent years for surveillance and information gathering in a wide variety of applications such as border patrol, wildfire fighting and mapping (Jones, 2009). Fast emergence and wide adoption of UAV and UGV was mainly due to the fact that some of the missions are too dull, dirty or dangerous for manned vehicles (Unmanned aerial vehicle, 2013). Table 1 summarizes a hierarchical structure of UAV/UGV behaviors at three different levels (high, middle, and low), where behavioral examples corresponding to each level (i.e. missions, tasks, and actions) are provided. The classical and traditional missions of unmanned vehicles are mainly intelligence, surveillance and reconnaissance (ISR), while in recent years other missions (e.g. suppression, strike, combat search and rescue) also have been started to emerge. Depending on different missions (surveillance vs. suppression), the associated core tasks and actions can be completely different (collect information vs. attack enemy; record video vs. fire weapon). The focus of this work is mainly on surveillance mission via UAVs and UGVs.

Surveillance is defined as a monitoring process (usually for people) of activities and various changing information with the aim to influence, manage, direct or protect (Surveillance, 2013). Surveillance, especially with advanced techniques, is revealing its importance in the current age. Considering the border patrol scenario as an instance, Fig. 1 shows the US border patrol budgets and southwest border apprehensions over the last two decades, which reveal that (1) the budgets that are associated with surveillance functionality had a dramatic increase over the past 20 years, and (2) the decreasing trend of apprehensions after 2005, which was partially due to the upgrade of old monitoring tools (e.g. fixed cameras, ground sensors) and the adoption of new surveillance techniques (e.g. UAVs, UGVs).

Surveillance and crowd control is a complex task as it involves various topics including surveillance region selection and vehicle assignment, vehicle searching and path generation, sensory data collection and processing, target detection and classification, object movement tracking, vehicle formation control and coordination, among others. Successful mission of surveillance and crowd control requires corresponding modules/algorithms under those topics to work both effectively and computationally efficiently

* Corresponding author. Tel.: +1 520 626 9530; fax: +1 520 6266555.
E-mail address: son@sie.arizona.edu (Y.-J. Son).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.039

Table 1
Hierarchy of UAV/UGV behaviors (Marsh, 2007).

Hierarchy in UAV/UGV behaviors	Behavior examples
High level (missions)	Surveillance, reconnaissance, suppression, strike, combat search and rescue
Middle level (tasks)	Avoid obstacle, collect information, react to disturbance, attack enemy, assess damage
Low level (actions)	Change altitude, change speed, change moving angle, take photograph, record video, fire weapon, leave pheromone, call for reinforcement

under an integrated and coordinated manner. While significant research efforts were pursued to study and improve some of the key modules such as searching, detection, tracking, and motion planning, only few works in the literature have addressed the overall system framework to support these modules in a coherent manner. In addition, performance and computation are always issues for cost and quality improvement when operating the UAV/UGV system, which requires more research works in addressing trade-off relationships between them for a specific application/scenario in a more formal and systematic manner.

To overcome these challenges and issues, this paper proposes a dynamic data driven adaptive multi-scale simulation (DDDAMS) based planning and control framework for surveillance and crowd control via UAVs and UGVs. The proposed framework aims to achieve the following major features: (1) comprehensive, in terms of wide control functionality and various algorithms (e.g. detection, tracking, motion planning), (2) multi-level, which can accommodate system uncertainties and problem complexities, and (3) adaptive, based on dynamic real time data collection to adjust the modeling scale (i.e. simulation fidelity) accordingly over time.

To the best of the authors' knowledge, this paper is one of the few works in discussing a comprehensive and coherent framework involving planning/control functionality, hardware/software components, as well as multi-level decisions. In this paper, the planning and control functions of the proposed framework are discussed in details first, with the focus on the role of each major component such as crowd detection, crowd tracking and vehicle motion planning. Different modules can be invoked either on a temporal basis or an event basis in the case of deteriorated performance from the real system. For the case of temporal interaction, the planning module is invoked periodically from the control module; while under the event based interaction, the DDDAMS module is triggered when the system performance goes down. The DDDAMS module collects and filters data from the real system, and then adjusts modeling details for steering the measurement process. As the simulation fidelity can be dynamically altered during the system run, the limited computational resource can be adapted to achieve a better system of interest. The developed simulation system in turn evaluates different control strategies for generating corresponding control commands in the real system. In the experiment section, a surveillance and crowd control scenario via UAVs and UGVs is considered to demonstrate the system performance in terms of crowd coverage percentage at each time period, which belongs to the generic measurement of effectiveness (MOE) for UAV/UGV missions. The goal of the experiment is to demonstrate (1) hardware-in-the-loop test-bed, agent-based simulation system, and the developed communication methods and formats, and (2) the DDDAMS system performance and computation under different system configurations.

The rest of this paper is organized as follows. Section 2 provides a literature survey of tracking and motion planning algorithms in surveillance and crowd control applications followed by a summarized background of general DDDAS applications and methodologies, Section 3 discusses the proposed DDDAMS-based framework for surveillance and crowd control, with details of each component and algorithm involved. Section 4 provides details on our hardware, software and simulation system demonstration, and experimental settings and the results based on different simulation fidelities. Section 5 concludes the paper and provides potential future research directions.

2. Background and literature survey

In the past, surveillance has found its applications across a broad range of areas covering agriculture (e.g. livestock monitoring), transportation (e.g. traffic monitoring, road patrol), wildfire mapping, homeland security (e.g. border patrol), and pipeline security, among others. Various surveillance tools were also reported including fixed camera, radar, ground-based sensor, and UAV/UGV. The scope of this research work is centered on discussing the surveillance and crowd control in rural (unpopulated, varying environmental terrain) area via UAVs and UGVs. The increasing trend of UAV and UGV in surveillance has been observed, which was mainly due to the development of automated object detection tools including face recognition systems, license plate scanners, thermal imaging cameras and open Wi-Fi sniffers (Unmanned aerial vehicle, 2013).

Only few research papers are available in the literature on the development of the overall surveillance system framework via UAVs and UGVs. Girard, Howell, and Hedrick (2004) proposed a hierarchical layered control architecture for border patrol via UAVs, in which the hierarchy formalization and their interfaces were discussed. There are totally 5 layers introduced in the paper: autopilot (layer 1), maneuver coordination (layers 2 and 3), vehicle

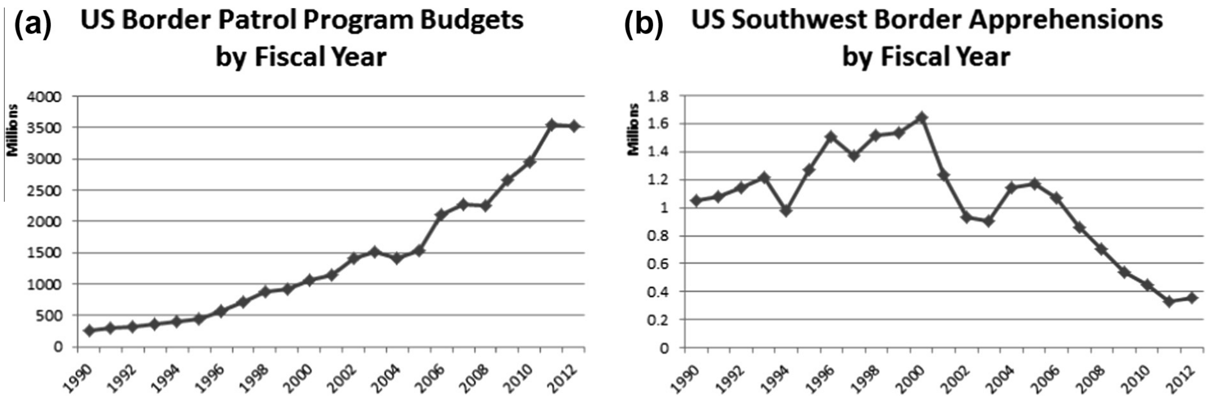

Fig. 1. US border patrol data (US Customs and Border Protection, http://www.cbp.gov).

supervision (layer 4), and team supervision (layer 5); followed by the discussion on controller design in details. Loyall, Schantz, Corman, Paunicka, and Fernandez (2005) presented a distributed real-time embedded (DRE) application system consisting of several UAVs, combat UGVs, and control centers to perform surveillance of time critical targets. Various DRE characteristics and challenges associated with the development complexity were discussed as well. Quigley, Goodrich, Griffiths, Eldredge, and Beard (2005) discussed a fixed wing mini-UAV control system on target acquisition, localization and surveillance involving flight-path generation, low-level UAV-centric control, high-level target-centric control and camera-centric control modules.

As discussed in Section 1, surveillance and crowd control involves various topics. Instead of reviewing the huge literature of all the topics, this section focuses on the research works that are highly relevant with this paper. For this purpose, crowd tracking and vehicle motion planning algorithms are surveyed first for the rest of this section, followed by the DDDAS methods and applications.

2.1. Crowd tracking

Upon the detection of individuals in a crowd by UAVs or UGVs, movement tracking of these individuals can be initiated. Individual movement tracking predicts the location of individuals at next time based on current and historical observations. Based on these predicted locations, UAVs and UGVs can decide their optimal control strategy of the crowd quantified by certain performance measure. Various tracking algorithms have been proposed in the literature, such as Kalman filter (Welch & Bishop, 1995), extended Kalman filter (Julier & Uhlmann, 2004), switching Kalman filter (Pavlovic, Rehg, & MacCormick, 2001), grid-based tracking (Arulampalam, Maskell, Gordon, & Clapp, 2002; Silbert, Mazzuchi, & Sarkani, 2011), and particle filter (Ristic, Arulampalm, & Gordon, 2004). Kalman filter is an algorithm that allows estimation of unknown variables through observation of a series of noisy measurements. The method produces a statistical optimal estimate of unknown variables, e.g., underlying state of individual person, in tracking. The estimation optimality is guaranteed for linear dynamic systems where the state transition and observation models are linear function of the state with Gaussian noises. In real data, the assumption of linear dynamic system may not hold, therefore approaches such as extended Kalman filters and switching Kalman filters have been proposed. The extended Kalman filter approximates the non-linear dynamic function with first-order Taylor series expansion. The switching Kalman filter, models the non-linear dynamics of the state by having a bank of several different Kalman filters and switching between them. Both approaches are developed under Gaussian noise assumption. Grid-based tracking differs from Kalman filter and its extensions by adopting discretized state space. In Kalman filter tracking, the true state of the object will take continuous values in the state space. In contrast, the grid-based tracking divides the state space into small cells and represents object state with these cells. One advantage of grid-based tracking is that it relaxes the normal distribution assumption of continuous state in Kalman filter and presents arbitrary distributions over discrete state space. The disadvantage is the computational complexity. This complexity grows exponentially with the number of state dimensions. Therefore, grid-based tracking is mainly applicable to estimation problems with a low-dimensional state. Particle filter is another approach for handling non-linear dynamic system state estimation. The particle filter models the latent system state with a first order Markov process and estimates these states by generating a set of samples approximating the conditional distribution of the latent state given observations. The expectation of this distribution can serve as estimates for the states. The advantage of particle filter is its capability to represents arbitrary probability distributions. The method converges to true posterior under non-linear dynamic systems with non-Gaussian noise. The major drawback is that the worst-case computational complexity grows exponentially with state dimensions.

2.2. Motion planning of UAV/UGV

The goal of motion planning is to determine an optimal/semi-optimal trajectory between any two locations using a sequence of waypoints under some known/unknown constraints (Gilmore, 1991). It plays a key role in autonomous operation of UAV/UGV due to their sensitivity to energy consumption and range of operation. Numerous algorithms have been proposed in literature to achieve optimal trajectories of vehicles. For example, Goerzen, Kong, and Mettler (2010) categorized UAV/UGV motion planning algorithms into seven major classes: (1) roadmap methods, (2) exact cell decomposition, (3) approximate cell decomposition, (4) approximation and decomposition, (5) potential field methods, (6) probabilistic approaches, and (7) weighted region problem.

Motion planning for unmanned vehicles was first approached via studying individual vehicle search with full (or partial) information about the environment. Under the full information situation about the environment, graph search methods such as Dijkstra (Dijkstra, 1959) and A* (Hart, Nilsson, & Raphael, 1968), which belong to road map category, were widely used in the literature. These methods reduce the problem complexity by fitting a graph to the space. In addition, in recent years, motion planning algorithms for partially known or stochastic environment have been developed. Under this situation, various graph search methods proposed in the literature are mainly variants of A* such as D* (Stentz, 1994), which is a dynamic version of A* where cost can vary, and Theta* (Nash, Daniel, Koenig, & Felner, 2007) that finds path with no constraint on angles. Additionally, for motion of robot devices in a real environment, several angles of freedom and complexity should be studied. In this topic, the methods of probabilistic roadmap planner are used, which were developed for fully known static environments in Kavraki and Latombe (1998). A comprehensive comparative study of different motion planning approaches can be found in Goerzen et al. (2010).

2.3. DDDAMS applications

Dynamic Data Driven Adaptive Multi-scale Simulation (DDDAMS) system belongs to the category of Dynamic Data Driven Application Systems (DDDAS), which was firstly proposed by Darema (2004) and has attracted lots of research attentions since then. Pingali and Stodghill (2004) proposed a web service framework (O'SOAP) for DDDAS applications, in which applications were deployed as distributed components and their interactions were achieved via SOAP protocol. The paper also discussed the performance of the proposed system in terms of test problem sizes and runtimes. Carnahan and Reynolds (2006) discussed accomplishing DDDAS adaptation capability via coercion, which is a semi-automated adaptation method for choosing the best abstraction at each flexible point that DDDAS needs. The results were demonstrated in a supply chain system. Douglas et al. (2006) reported DDDAS approaches to fire modeling and contaminant tracking. In addition, Rodríguez, Cortés, and Margalef (2009) discussed using DDDAS for forest fire behavior prediction. Experiment results demonstrated prediction improvement under different wind speeds and directions. Celik, Lee, Vasudevan, and Son (2010) proposed a DDDAS-based multi-fidelity simulation framework for analyzing supply chain system, in which various components (e.g. grid computing module, web service communication server, databases, sensors) and DDDAS enabling algorithms (e.g. abnormality detection, fidelity selection, fidelity assignment) were discussed. Madey et al. (2012) identified several key challenges for swarms of sensor and surveillance UAVs and discussed proof-of-concept DDDAS based simulation test-bed for UAV swarm control. In their framework, two simulators were employed with a synthetic UAV swarm simulator (i.e. MultiUAV2) and another agent-based DDDAS simulation toolkit (i.e. Java/MASON). Various workflows (e.g. agent task transition) and sensor re-tasking and pre-processing were also described. Wei, Blake, and Madey (2013) discussed dynamic mission planning for efficient UAV swarm control, in which a DDDAS-driven hybrid control framework was proposed for UAV mission assignment and scheduling considering different scheduling policies.

3. Proposed DDDAMS-based framework for surveillance and crowd control

Fig. 2 shows an overview of the proposed DDDAMS-based planning and control framework for surveillance and crowd control via UAVs and UGVs, where the major components include (1) real system (UAVs, UGVs, human crowd, and environment), (2) integrated planner, (3) integrated controller, and (4) decision module for DDDAMS. The proposed framework is aimed to enhance the surveillance and crowd control capability of UAVs and UGVs in terms of their performance on crowd detection, tracking and motion planning. In particular, the crowd coverage percentage is considered as the MOE in this work. Note that two preliminary versions of the proposed framework have appeared in two conference proceedings in Wang et al. (2013) and Khaleghi et al. (2013) (accepted for publication, the online version will be available in December, 2013). An overview of different components is provided in the following paragraphs.

The integrated controller is in charge of the effective and efficient control of UAVs and UGVs, where the effectiveness is supported by the integrated planner, and the computationally efficiency is supported by the decision module for DDDAMS. To control UAVs and UGVs, the integrated controller performs four major functions: (1) crowd detection, (2) crowd tracking, (3) motion planning of UAV/UGV, and (4) interaction with the real system. To achieve interactions with the real system, the hardware interface in the integrated controller acts as a medium to collect sensory data (e.g. vision and global positioning system (GPS) data) from the real system, passes them to the command generator, receives control commands from the motion planning module, and sends the corresponding control commands to the real system. Among the received sensory data, vision-based data (e.g. images and video streams) are utilized in the crowd detection module, and the GPS data are used by the motion planning module. Given the vision-based data, the crowd detection module (1) processes them for crowd detection and location identification purposes, (2) invokes the decision module for DDDAMS if actual and predicted system performances (crowd coverage in this work) have a significant deviation, and (3) passes the analyzed crowd data (e.g. locations and crowd coverage) to the database periodically. Based on the crowd location identified at time t, the crowd tracking module predicts the crowd locations at time $t + \Delta t$, where Δt is an interval that users must define depending on the intended frequency of control (10 s are used in this work). The tracking is made at each time unit (so the Δt is an integer), and predicted locations for Δt time periods are then used in the motion planning module as destination locations of the UAVs/UGVs, from which control

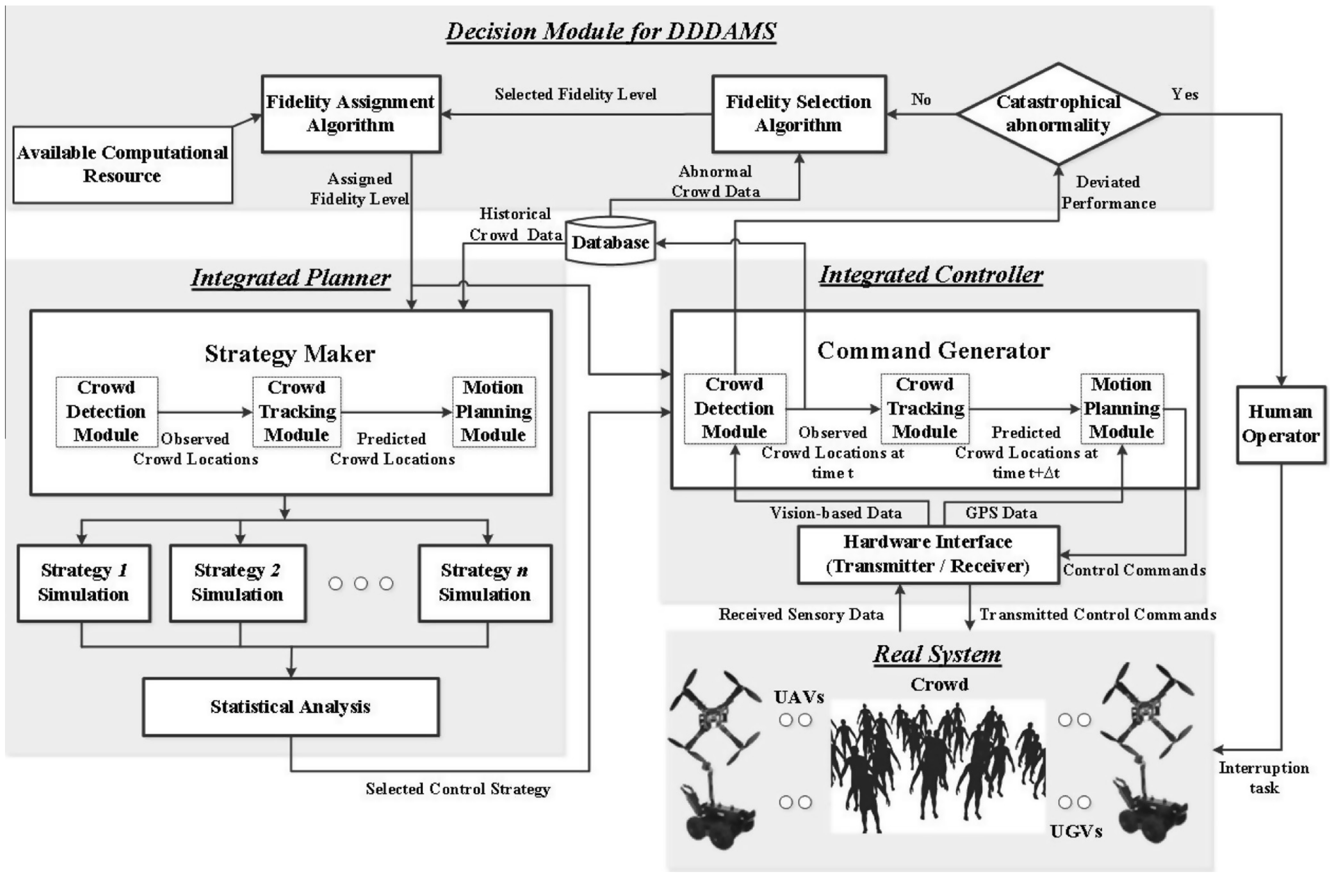

Fig. 2. Proposed DDDAMS-based planning and control framework for surveillance and crowd control via UAVs and UGVs (Khaleghi et al., 2013; Wang et al., 2013).

Table 2
Nomenclature.

Notation	Explanation	Notation	Explanation
FOV	Field of view	*SW*	UAV sweep width
AGL	Above ground level	P_D	Vehicle detection probability
R_A	UAV detection radius	r_A	Distance between target and ground mapping point of UAV
$d_{\max}$	Maximum distance of detection by UAV	$\mathbf{x}_i(t)$	State vector of the *i*th individual at time *t*
$\mathrm{N_I}$	Number of individuals in crowd	$\mathbf{h}(t)$	Environmental factors at time *t*
$\mathbf{A}(\mathrm{t})$	Dynamics of individuals in crowd & their intra-agent interactions at time *t*	$\mathbf{u}(t)$	States of the controlling team at time *t*
$\mathbf{B_f}(t)$	Impacts of the environmental factors at time t	$\mathbf{w}(t)$	Process noise at time t
$\mathbf{B_u}(t)$	Individual's perceptions of the controlling activities at time *t*	$\mathbf{x}(\mathrm{t})$	State vector of crowd at time t
$\mathbf{y}(t)$	Observation at time *t*	$\mathbf{v}(t)$	Measurement noise at time *t*
$\mathbf{C}(t)$	Maps the true state space into observed space at time *t*	$\mathbf{\Sigma}_{\mathrm{Measure}}$	Variance–covariance matrix of measurement noise
$\mathbf{\Sigma}_{\mathrm{state}}$	Variance–covariance matrix of state noise	β	Elevation angle between two waypoints
$\mathbf{D}_V(t)$	Vehicle coordinates in the 3D environment at time t: $[\mathrm{x_V(t), y_V(t), z_V(t)}]$;	N_G	Total number of individual agents detected by UGV
elevPenalty(β)	Elevation penalty function with respect to β	$p_j(t+\Delta t)$	Proportion of individual agents belong to the *j*th cluster at time $t+\Delta t$
$f_1^{(A)}$	The 1st objective function (minimize Euclidean distance) for UAV	$f_2^{(G)}$	The 2nd objective function (minimize fuel consumption) for UGV
$\boldsymbol{\mu}_{j,G}(t)$	Mean vector of *j*thcrowd cluster at time *t* from UGVs	$\mathbf{\Sigma}_{j,G}(t)$	Covariance matrix of *j*thcrowd cluster at time *t* from UGVs
$\hat{\boldsymbol{\mu}}_{k_G}(t+\Delta t)$	Estimated mean vector of the k_Gth UGV destination at time $t+\Delta t$	$\hat{\mathbf{\Sigma}}_{k_G}(t+\Delta t)$	Estimated covariance matrix of the k_Gth UGV destination at time $t+\Delta t$
$fd_l(t)$	Selected simulation fidelity level for *l*th cell at time *t*	$q_l(t)$	0–1 indicator of whether required computational resource is assigned to *l*th cell at time *t*
$g(fd_l(t), l)$	Utility function of system performance when $fd_l(t)$ is assigned to the *l*th cell at time t	$R(fd_l(t), l)$	System resource usage function when $fd_l(t)$ is assigned to the *l*th cell at time *t*
L	Total number of visibility cells by UAV/UGV	*TR*	Total available system computational resource

commands (e.g. waypoints) for next period $t+\Delta t$ are generated for the UAVs/UGVs. This process (i.e. data collection – crowd detection – crowd tracking – motion planning – control command generation) continues in an iterative manner until the real system performance differs enough from the predicted performance (event basis) or the predefined planning horizon is elapsed (temporal basis), which then invokes the integrated planner.

The integrated planner, when invoked, devises an optimal control strategy for UAVs/UGVs based on predicted system performance and passes the updated control strategy to the integrated controller. The integrated planner in the proposed work has been implemented in an agent-based simulation (ABS) environment, where the strategy maker selects optimal strategies for each of the same components in the command generator (i.e. crowd detection, crowd tracking and motion planning) based on simulation-based evaluation of alternative strategies against different scenarios. This work mainly focuses on (1) evaluation of alternative estimation methods of UAV/UGV destinations in $t+\Delta t$, and (2) evaluation of multi-objective weights in UAV/UGV motion planning. For estimation of UAV/UGV location, the crowd shape and boundary are characterized first via clustering technique, followed by the simulation-based evaluation on UAV/UGV locations contingent to different control strategies (see Section 3.1.4). For the evaluation of multi-objective weights in UAV/UGV motion planning, agent-based simulation is used similarly for evaluating different weights on the objectives (e.g. shortest travel distance, least elevation change) over different scenarios. Under both cases, each control strategy only corresponds to one simulation instance, and the best strategy can be obtained via statistical analysis after all simulation instances are completed. At last, the selected best control strategy is stored and then used by the integrated controller for adjusting the corresponding vehicle parameters.

The overview of DDDAMS components and their interactions are provided in this paragraph. At a given time point *t*, when the decision module for DDDAMS is invoked, the checking condition (catastrophic abnormality block) is processed first. The checking condition determines whether the current control system has a severe problem or performance deviations (predicted vs. real) are too extreme to recover. Under these circumstances, the human operator should interrupt the system. These fatal abnormalities are due to system malfunctions or human errors which are beyond the scope of our analysis in this work. Our interests are on the abnormalities, where the actual and predicted system performances deviate significantly where every component still works under normal condition. For the case of ordinary abnormality, the fidelity selection algorithm is invoked next. The outputs of the fidelity selection algorithm are a combination of different fidelity levels for all the considered crowd regions/cells in terms of information details (collected via UAV or UGV) to be incorporated into simulation. In general, simulating group level behaviors involves a coarse scale and requires less information and computational resource (and time), while simulation of detailed individual behaviors needs finer scale of modeling, more detailed information and more computations. Given the results of fidelity selection algorithm, the fidelity assignment algorithm decides the optimal assignment of modeling scale by adjusting optimum system performance against computational resources available in the system. The fidelity assignment algorithm is formulated as a 0–1 knapsack problem, and the assigned fidelity levels are provided to the integrated planner and integrated controller for the simulation fidelity update, control strategy evaluation and control command generation.

The rest of this section is organized as follows. Section 3.1 discusses the integrated planner and the integrated controller, which include crowd detection, tracking, motion planning, and control strategy. Section 3.2 discusses the decision module for DDDAMS, which covers the defined fidelity and levels, fidelity selection and assignment algorithms. Table 2 summarizes the notations used throughout the paper.

3.1. Integrated planning and control modules

3.1.1. Crowd detection module

To address crowd detection by UGV and UAV, this work considers both real image/stream data from the camera in the real UAV as well as models of them, where key characteristics considered in the model include (1) field of view (FOV), (2) sweep width (SW), (3) target detection probability, (4) reactivity, and (5) endurance.

While there are additional UAV characteristics for other missions, these five characteristics have been selected as they are highly relevant with the detection and surveillance mission. Each of them is explained below:

- FOV is the maximum angle that a UAV can view. This factor is related with the sensor configuration itself and UAV mounting structure (e.g. top–down view, side view; rotation feature). In general, FOV has a negative relationship with the sensor resolution; for a particular sensor, its resolution increases when FOV decreases (Raffetto, 2004).
- SW is defined as the width of the searching area for a single glimpse by the sensor (Raffetto, 2004). In Washburn and Kress (2009), it is defined in another way as the area underneath the lateral range curve, which is a graph capturing the relationship between detection probability and lateral range (minimal distance between the searching vehicle and target object over a path). Theoretically, SW can be represented as a function of Above Ground Level (AGL) altitude and FOV in Eq. (1), and Fig. 3 shows a graphic illustration of these parameters.

$$SW = 2 * AGL * \tan(FOV/2) \tag{1}$$

- Target detection probability (P_D) is the probability that a target can be detected. It's known that not all the targets within the sweep width range can be detected. In the real world, targets that are directly under (or very close to) the UAV have a higher probability being detected than those towards the edge of the sensor range. For simplicity, the detection probability is modeled as a binary variable as shown in Eq. (2), where r_A is the distance between the ground mapping point of UAV and target, and R_A represents the radius of detection range as shown in Eq. (3). Here, d_{max} is the maximum distance that UAV can detect a target, and is assumed to be a fixed number according to the particular vision sensor used. Depending on the AGL altitude, three types of relationship are formulated in Eq. (3). Fig. 3 shows a graphic illustration when the second condition in Eq. (3) is satisfied. In the figure, the green-shaded area under the green cone represents the UAV detection area (i.e. $P_D = 1$).

$$P_D = \begin{cases} 1, & \text{if} \quad r_A \leqslant R_A \\ 0, & \text{if} \quad r_A > R_A \end{cases} \tag{2}$$

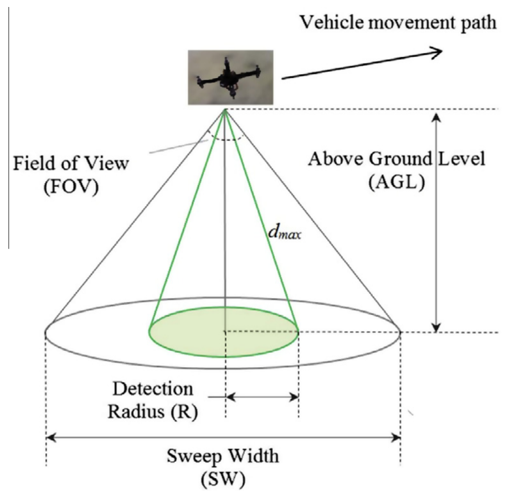

Fig. 3. Illustration of UAV detection and associated parameters.

$$R_A = \begin{cases} SW/2, & \text{if} \quad 0 < AGL \leqslant d_{\max}\cos(FOV/2) \\ \sqrt{d_{\max}^2 - AGL^2}, & \text{if} \quad d_{\max}\cos(FOV/2) < AGL < d_{\max} \\ 0 & \text{if} \quad d_{\max} < AGL \end{cases} \tag{3}$$

- UAV/UGV reactivity defines UAV's or UGV's behavior law when dealing with unknown objects and unclassified agents. In the considered surveillance scenario in this work, people can be of different types (e.g. friends, enemies, or neutral type), but they are not differentiated in this paper. In our future work, when there will be a need to differentiate people from animals, the concept of object classification probability will be used. For those objects not being detected (within SW but not within detection range) or agents not being classified, specific rule is required to be established on how UAV/UGV should react to them. Examples of basic rule can be either (1) staying on the current task, or (2) changing route to explore the unknown objects or agents.
- UAV/UGV endurance is defined as the vehicle operation duration given a particular set of starting point, destination, routes, environmental conditions and the vehicle conditions themselves. Endurance issue needs to be handled carefully, because unexpected UAV/UGV crushes can possibly shut down detection capability and consume additional resources (e.g. working personnel, UAV/UGV re-launch cost). While our work does not explicitly address the relationships of endurance with all other impact factors, the shortest travel distance and fuel/energy consumption issues are considered in Section 3.1.3 (motion planning module).

For the real hardware concern, vision sensors (e.g. cameras) are considered, which are the most common sensors in surveillance due to their low cost and large varieties of data processing approaches (e.g. computer vision). In this work, low cost off-the-shelf cameras are used as onboard vision sensors for collecting crowd observation data. For this purpose, downward looking cameras are mounted on UAVs, and UGVs are equipped with horizontal cameras. During the mission, sensory observation data are transmitted to the ground control station, and further processing and analysis (e.g. human detection, motion detection) can be conducted for extracting useful information from the data. As discussed in Hu, Tan, Wang, and Maybank (2004) motion detection plays a key role in most surveillance systems by detecting reigns of moving objects in motion segmentation and object classification due to the motion varieties involved with moving real world objects. Background subtraction, temporal differencing and optical flow are three popular conventional methods for detection of moving reigns. Moving objects classification is critical for further localization, tracking of objects and analyzing their behavior, where shape-based, motion-based and their combinations are the main approaches found in detection literature.

In terms of models of detection capabilities of UAV/UGV (having cameras) during their surveillance, their complementary role has been considered in this research. UAVs with less accurate detection capability are used to localize crowd as groups by flying over obstacles to keep the entire group in their FOV, and UGVs are used for detection of individuals in crowd due to their accurate localization. This assumption considered in our modeling is based on the visual comparison in Grocholsky, Keller, Kumar, and Pappas (2006) that has demonstrated uncertainties in localization of detected objects between air and ground vehicles. Section 3.2 provides detailed discussion on how information collected by UAV and UGV are associated with modeling scale (simulation fidelity).

The outputs from the detection module are used as input to the crowd track module, which is discussed in the next section.

3.1.2. Crowd tracking module

This paper adopts the Kalman filter for crowd tracking, which has been extensively studied and been successfully implemented for various applications such as video surveillance. The Kalman filter is formulated with a state space model and a measurement model. The state space model quantifies the dynamics of the crowd motion and the measurement model quantifies the relationship between observations and the latent state of each individual in the crowd.

For state space model, denote the state vector $\mathbf{x}_i(t)$, $i = 1,2,\dots,N_I$, for individuals in a crowd at time t, where N_I is the total number of individuals in the crowd. The state vector contains state information of an individual, including its preference state, such as moving speed and direction, and geographic state, such as locations. Thus the preference state vectors are used as elements to model the complex spatial/temporal problem. The state vector of a crowd at time t can be represented as a composition of all the N_I state vectors as shown in Eq. (4).

$$\mathbf{x}(t) = \left[(\mathbf{x}_1(t))^{\mathrm{T}} \quad (\mathbf{x}_2(t))^{\mathrm{T}} \quad \cdots \quad (\mathbf{x}_{N_I}(t))^{\mathrm{T}} \right]^{\mathrm{T}} \tag{4}$$

As an individual person in a crowd, its preference and geographic states at time t are mainly affected by three types of factors: (1) the preference of crowd individuals, including itself, at time $t-1$; (2) the environmental factors that affect crowd preferences; and (3) the impacts of the controlling units, such as UAVs and UGVs. These complex relationships can be modeled by Eq. (5).

$$\mathbf{x}(t) = \mathbf{A}(t)\mathbf{x}(t-1) + \mathbf{B_f}(t)\mathbf{h}(t) + \mathbf{B_u}(t)\mathbf{u}(t) + \mathbf{w}(t),$$
$$t = 1,2,\dots,T \tag{5}$$

where T is the total number of motion stages; $\mathbf{A}(t)$ represents the dynamics of crowd individuals, their *intra-agent interactions*, e.g., among preference options of different individuals, and *inter-agent interactions*; $\mathbf{h}(t)$ denotes the environmental factors, $\mathbf{B_f}(t)$ represents the impacts of the environmental factors on individuals' state; $\mathbf{u}(t)$ denotes the states of the controlling team, e.g. the formation, orientation and movement of UAVs and UGVs; $\mathbf{B_u}(t)$ represents the individual's perceptions of the controlling activities; $\mathbf{w}(t)$ represents the process noise which is assumed to follow a multivariate normal distribution with zero means and variance–covariance matrix $\sum_{\text{state}}$. The measurement model has the form as shown in Eq. (6).

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{v}(t) \tag{6}$$

where $\mathbf{y}(t)$ is the observation at time t; $\mathbf{C}(t)$ maps the true state space into observed space and $\mathbf{v}(t)$ is the measurement noise assumed as zero mean Gaussian with variance–covariance matrix $\sum_{\text{measure}}$.

If the values of the matrices used in Kalman filter, such as $\mathbf{A}(t)$, $\mathbf{B_f}(t)$, $\mathbf{B_u}(t)$, and $\mathbf{C}(t)$ have been specified, the Kalman filter performs tracking with two recursive steps. The first step, i.e. prediction step, estimates the value of the latent state variables in future time using the state space model, along with an uncertainty measure of the estimate. The second step, i.e. updating step, obtains new measurement and updates the estimates using weighted average approach, with more weights on the estimates with a higher certainty. In cases where these matrices are unknown, system identification must be performed to determine them and then the tracking of crowd can be implemented with Kalman filter.

Different formats of $\mathbf{A}(t)$, $\mathbf{B_f}(t)$, $\mathbf{B_u}(t)$, $\mathbf{h}(t)$ and $\mathbf{u}(t)$ can be specified to model the motion dynamics of the crowd under different scenarios. The common scenarios include crowd moving towards a direction; shrinking or spreading of the crowd, change of movement direction and speed, crowd split, or merge. For example, the scenario of crowd moving towards a direction with stable shape can be modeled by adding interaction term in matrix $\mathbf{A}(t)$. The interaction between the individual persons in one crowd can be modeled (Gning, Mihaylova, Maskell, Pang, & Godsill, 2011) by using the average speed of individuals in the crowd plus a noise as the speed of each individual. In this case, the matrix $\mathbf{A}(t)$ become the following equation (Eq. (7)):

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{11} & \dots & \mathbf{a}_{1N_I} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \dots & \mathbf{a}_{2N_I} \\ \dots & & & \\ \mathbf{a}_{N_I1} & \mathbf{a}_{N_I2} & \dots & \mathbf{a}_{N_IN_I} \end{bmatrix}, \quad \mathbf{a}_{ii} = \begin{bmatrix} 1 & 1/N_I & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1/N_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$i = 1,2,\dots,N_I, \quad \mathbf{a_{ij}} = \begin{bmatrix} 0 & 1/N_I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/N_I \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{i} \neq \mathbf{j} \tag{7}$$

The non-zero values $1/N_I$ in matrix $\mathbf{A}(t)$ allows the modeling of the interaction. The effect of $1/N_I$ term in matrix $\mathbf{A}(t)$ is depicted in Fig. 4. The figure is plotted based on a simulation of a crowd with 200 individuals in 50 time units. The locations of individuals at time 1, 21, and 41 are plotted, using square, circle and plus sign, respectively. The initial individual locations at time 1 is drawn from a bivariate normal distribution with mean $(3,2)$ and variance–covariance matrix $(1,1.5;1.5,3)$. This distribution setting results in ellipsoidal shape of the crowd. Each agent in the crowd has moving speed of $(1,0.5)$ in the two-dimension space with i.i.d. process noise following $N(0,0.1)$. Two settings of matrix $\mathbf{A}(t)$ are used. One is from Eq. (7) and the other is a diagonal matrix. These two settings correspond to Fig. 4(a) and (b), respectively. The figure shows that with the modeled interaction all the individuals form a stable shape and move towards a direction, while the crowd modeled without interaction spreads out and loses its shape. Thus, when a scenario suggests that crowd moves in a stable shape, $\mathbf{A}(t)$ of Eq. (5) needs to be used in the state space model of Kalman filter.

3.1.3. UAV/UGV motion planning module

In this work, A* algorithm has been used for motion planning of UAVs/UGVs to find their optimal trajectory from the starting location to the destination location with given Δt control interval, where the destination location is defined based on the output of crowd tracking module. A* is a well-known graph search algorithm, where the optimality and efficiency of the chosen path depend on the chosen heuristic (Dechter & Pearl, 1985). It combines concepts of Dijkstra (least cost from the initial location) and Best-First Search on a heuristic basis (least cost to the destination) to find the optimal path by searching a graph efficiently (Choset, 2005). Considering the environment (e.g. terrain elevation, obstacles), it is assumed that the world is static and prior geographical information is known. In the case of insufficient prior knowledge about the environment and existing obstacles, vision-based sensory data (e.g. image, video streams) can be used to provide such dynamic data.

In A* algorithm, similar to other graph search algorithms, grids are generated by discretizing the environment and a graph is defined where each node of the graph represents one grid. In this work, two different grids have been used. For UGV motion planning, the grid is placed on the ground surface, and for UAV, the grid is 30 m above the ground surface. Fig. 5 shows two possible paths in a map with one starting location, one destination location, and several waypoints. At each current location that the vehicle belongs to, it chooses 1 out of the 8 possible waypoints to decide the next movement.

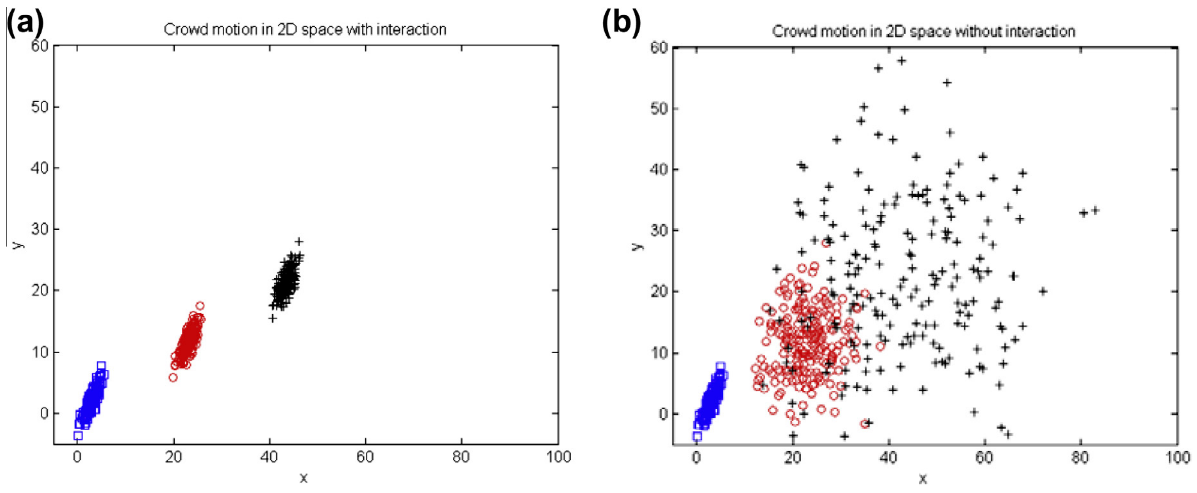

Fig. 4. Crowd motion simulations: (a) with interactions and (b) without interactions.

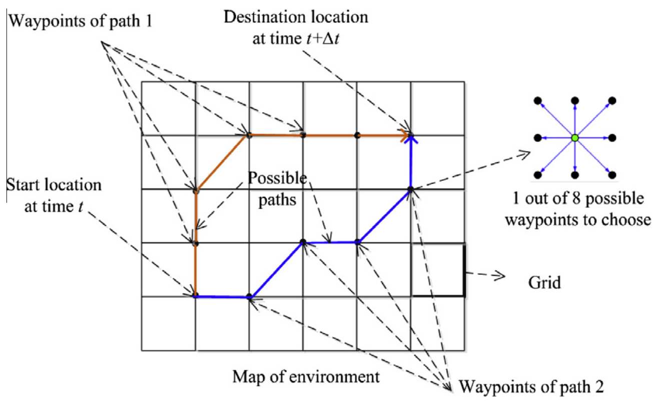
Waypoints of path 1
Destination location at time $t+\Delta t$
Start location at time t
Possible paths
1 out of 8 possible waypoints to choose
Grid
Map of environment
Waypoints of path 2

Fig. 5. Two possible paths in a discretized map generated by A* algorithm.

Now, let $\mathbf{D}_V(t) = [x_V(t), y_V(t), z_V(t)]$ represent the vehicle coordinates (location) in the 3D environment and the terrain elevation of the environment at vehicle's location is obtained by accessing GIS data from NASA World Wind® (http://worldwind.arc.nasa.gov/java/) which is represented by $elev(x_V(t), y_V(t))$. The altitude of UAVs from Mean Sea Level (referred to as MSL altitude) is shown by $z_V^{(A)}(t)$, and it can be written in Eq. (8). For UGVs, $z_V^{(G)}(t)$ is equal to the terrain elevation of the environment (see Eq. (9)) at the UGVs' location on the ground (i.e. latitude, longitude).

$$z_V^{(A)}(t) = elev(x_V^{(A)}(t), y_V^{(A)}(t)) + AGL \tag{8}$$

$$z_V^{(G)}(t) = elev(x_V^{(G)}(t), y_V^{(G)}(t)) \tag{9}$$

Assume $t = t_0$ is the initial time of a vehicle's path and $T = t_0 + \Delta t$ is the arrival time to the destination location. Two objective functions are considered in this work: the first objective is to minimize the vehicle traveling distance (Euclidian distance) from the starting location to the destination location during each control interval (Δt); the second objective is to minimize a composite function involving vehicle altitude/elevation change. The two objective functions are shown in Eqs. (10) and (11), respectively.

$$f_1 = \sum_{t=t_0}^{T-1} \sqrt{[x_V(t+1) - x_V(t)]^2 + [y_V(t+1) - y_V(t)]^2 + [z_V(t+1) - z_V(t)]^2} \tag{10}$$

$$f_2 = \sum_{t=t_0}^{T-1} elevPenalty(\beta(t)) * \sqrt{[x_V(t+1) - x_V(t)]^2 + [y_V(t+1) - y_V(t)]^2 + [z_V(t+1) - z_V(t)]^2} \tag{11}$$

Where f_1 stands for $f_1^{(A)}$ or $f_1^{(G)}$, and f_2 stands for either $f_2^{(A)}$ or $f_2^{(G)}$, furthermore $[x_V(t), y_V(t), z_V(t)]$ stands for $[x_V^{(A)}(t), y_V^{(A)}(t), z_V^{(A)}(t)]$ or $[x_V^{(G)}(t), y_V^{(G)}(t), z_V^{(G)}(t)]$ representing either UAV or UGV coordinates in the 3D environment. In Eq. (11), *elevPenalty* is the penalty function for elevation change based on different angles formed by two-waypoint connection line and horizontal X–Y surface (Wichmann & Wuensche, 2004). The angle can be defined in Eq. (12), and a sample penalty function for elevation change is shown in Fig. 6.

$$\beta(t) = \arctan\left(\frac{|z_V(t+1) - z_V(t)|}{\sqrt{[x_V(t+1) - x_V(t)]^2 + [y_V(t+1) - y_V(t)]^2}}\right) \tag{12}$$

To illustrate the trade-offs between the two objectives, this paragraph discusses a special case of an unmanned vehicle traveling from a starting location to the destination location involving significant elevation increase (e.g. climb a mountain). On one hand, if the vehicle selects to travel the minimum Euclidian distance between the two points (i.e. minimize f_1), it will result in a steep lift for $z_V(t)$ (MSL attitude for UAV, elevation for UGV) in short time duration (Δt). This may cause a very slow movement speed of the

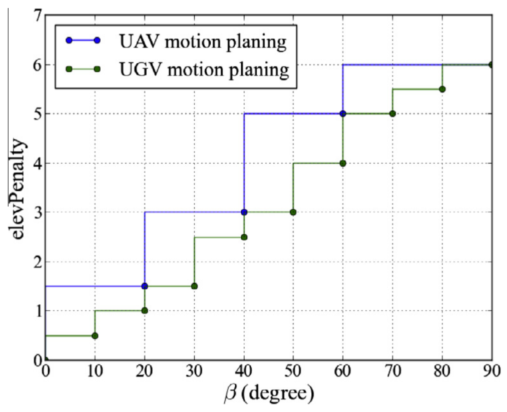

Fig. 6. Penalty function for elevation change in UAV/UGV motion planning.

vehicle for lifting its own weight under other disturbances (e.g. wind, wet road), which in turn cause inefficient usage of fuel (e.g. drain fast) in addition to stabilization issues. It has been found that vehicles consume more energy when changing their MSL altitude during the flight of UAVs (Sauter et al., 2008), or moving downhill/uphill for UGVs. On the other hand, if the vehicle chooses the path with the least elevation change (i.e. minimize f_2), it results in spiral traveling path (e.g. spiral road) to the destination location. It will cause (1) long traveling distance in turn consumes more fuel, and (2) high uncertainty along the path that may delay the vehicle's arrival time for surveillance mission. So, to increase the endurance (i.e. reduce the fuel consumption) for UAVs and UGVs, next section considers a multi-objective formulation based on weights representing the relative importance of the objectives over different scenarios.

3.1.4. Control strategy

The control strategy belongs to the integrated planner (left-hand side in Fig. 2), which aims to evaluate different strategy simulations against various scenarios of crowd movements. In this paper, we consider the control strategy associated with the following two aspects: (1) evaluation of alternative estimation methods of UAV/UGV destinations in $t + \Delta t$, (2) evaluation of multi-objective weights in UAV/UGV motion planning. It is noted that the UAV/UGV control strategy can involve more detailed control of kinematic/dynamic issues (e.g. vehicle stabilization to wind) that are usually handled via vehicle autonomy. In addition, group control strategies (e.g. swarm coordination under different control architectures such as centralized vs. decentralized) can be also considered. However, these topics are beyond the scope in this work, and are left for future research.

To accurately estimate the UAV/UGV location for the next destination, the shape of the predicted crowds and justifiable probabilistic boundary need to be characterized first. Since the crowd region often exhibits cluster patterns, clustering is adopted for crowd pattern extraction and quantification. Clustering is a method of categorizing unlabeled data into homogenous groups based on certain proximity measures, with the goal to identify the hidden pattern or structure of data. Due to its indispensable role in data exploration and knowledge discovery, numerous works have been studied, summarized in Berkhin (2006) and Xu and Wunsch (2005) as well as its many applications in different fields (Everitt, Landau, & Leese, 2001; Hartigan, 1975) including economics, computer science, bioinformatics, and social science. Generally, clustering methods can be classified into five categories (Han, Kamber, & Pei, 2006), namely partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. In this paper, Gaussian mixture clustering, one type of model-based clustering methods is employed to characterize the accurate shape of the crowd region by considering: (1) the great capability and flexibility of Gaussian mixture in representing different complex cluster patterns, (2) its parametric form in generating probabilistic assurance region.

Considering the crowd observation and tracking from UGV as an example: given the predicted locations $\hat{x}_i(t+\Delta t)$, $i = 1,\ldots,N_G$, by UGVs, where N_G is the total number of individuals that the UGVs could detect and track, the Gaussian Mixture Clustering can be performed to estimate the model parameters, $\Theta_G(t+\Delta t) = \{p_{j,G}(t+\Delta t), \boldsymbol{\mu}_{j,G}(t+\Delta t), \boldsymbol{\Sigma}_{j,G}(t+\Delta t)\}$, $j = 1,\ldots,m$, where $p_{j,G}(t+\Delta t)$ is the proportion of individuals belong to the jth cluster at time $t + \Delta t$, $\boldsymbol{\mu}_{j,G}(t+\Delta t)$ and $\boldsymbol{\Sigma}_{j,G}(t+\Delta t)$ are the mean vector and the covariance matrix for the location of the jth cluster at time $t + \Delta t$, all based on information collected by UGVs. Expectation–Maximization algorithm (McLachlan & Krishnan, 2007) is typically used for parameter estimation and the algorithm can be described as follows:

Step1 Initialize $p^{(0)}_{j,G}(t+\Delta t)$, $\boldsymbol{\mu}^{(0)}_{j,G}(t+\Delta t)$ and $\boldsymbol{\Sigma}^{(0)}_{j,G}(t+\Delta t)$;
Step2 Given data $\hat{x}_i(t+\Delta t)$, $i = 1,\ldots,N_G$, and for the iteration $\psi = 1, 2, \ldots, \psi_{Max}$, repeat the following:
Step 2.1 (Expectation Step): compute

$$\gamma_{ij} = \frac{p^{(\psi-1)}_{j,G}(t+\Delta t)\mathcal{N}(\hat{x}_i(t+\Delta t)|\boldsymbol{\mu}^{(\psi-1)}_{j,G}(t+\Delta t), \boldsymbol{\Sigma}^{(\psi-1)}_{j,G}(t+\Delta t))}{\sum_{k=1}^{m}\mathcal{N}(\hat{x}_i(t+\Delta t)|\boldsymbol{\mu}^{(\psi-1)}_{k,G}(t+\Delta t), \boldsymbol{\Sigma}^{(\psi-1)}_{k,G}(t+\Delta t))},$$

$i = 1,\ldots,N_G$, $j = 1,\ldots,m$, where $\mathcal{N}(\cdot)$ is the multivariate normal density;
Step 2.2 (Maximization Step): update

$$\boldsymbol{\mu}^{(\psi)}_{j,G}(t+\Delta t) = \sum_{i=1}^{N_G}\gamma_{ij}\hat{x}_i(t+\Delta t) \Big/ \sum_{i=1}^{N_G}\gamma_{ij},$$

$$p^{(\psi)}_{j,G}(t+\Delta t) = \sum_{i=1}^{N_G}\gamma_{ij} \Big/ N_G \text{ and,}$$

$$\boldsymbol{\Sigma}^{(\psi)}_{j,G}(t+\Delta t) = \sum_{i=1}^{N_G}\gamma_{ij}\left(\hat{x}_i(t+\Delta t) - \boldsymbol{\mu}^{(\psi)}_{j,G}(t+\Delta t)\right) \times \left(\hat{x}_i(t+\Delta t) - \boldsymbol{\mu}^{(\psi)}_{j,G}(t+\Delta t)\right)^{\mathrm{T}} \Big/ \sum_{i=1}^{N_G}\gamma_{ij}, \quad j = 1,\ldots,m.$$

So far, $\hat{\boldsymbol{\mu}}_{j,G}(t+\Delta t)$ and $\hat{\boldsymbol{\Sigma}}_{j,G}(t+\Delta t)$ of each cluster are found via EM algorithm. Then, the next step is to estimate the UGV location for the next destination, which can be calculated via the weighted linear combination of multiple cluster centers under the coverage of the considered UGV as shown in Eqs (13) and (14).

$$\hat{\boldsymbol{\mu}}_{k_G}(t+\Delta t) = \left(p_{j,G}(t+\Delta t) \Big/ \sum_{j\in S_{k_G}} p_{j,G}(t+\Delta t)\right)\hat{\boldsymbol{\mu}}_{j,G}(t+\Delta t) \tag{13}$$

$$\hat{\boldsymbol{\Sigma}}_{k_G}(t+\Delta t) = \left(p_{j,G}(t+\Delta t) \Big/ \sum_{j\in S_{k_G}} p_{j,G}(t+\Delta t)\right)\hat{\boldsymbol{\Sigma}}_{j,G}(t+\Delta t) \tag{14}$$

where $\hat{\boldsymbol{\mu}}_{k_G}(t+\Delta t)$ and $\hat{\boldsymbol{\Sigma}}_{k_G}(t+\Delta t)$ are the mean vector and covariance matrix for the k_Gth UGV destination at time $t + \Delta t$, k_G (=1,…,$K^{(G)}$) is the UGV index, and $j \in S_{k_G}$ indicates that the jth cluster belongs to the coverage of the k_Gth UGV. Fig. 7 shows an example with 3 crowds under the coverage of 1 UGV, as well as the location relationship between the crowds and the UGV. In the figure, the UGV center location and boundary for all possible destinations are calculated by Eqs. (13) and (14). As shown, all possible UGV destinations are filtered out within the considered boundary (red dot).[1]

To decide the best-fitted vehicle destination among all possible ones, agent-based simulation as the strategy maker can be used to evaluate different control strategies. Some sample strategies for determining the vehicle destination are as follows:

- Select the destination location that is the weighted crowd center.
- Select the destination location with minimum distance variance to all the crowds (i.e. non-weighted crowd center).
- Select the destination location with the nearest distance to the crowd that has the largest proportion of individuals.
- Select the destination location with the nearest distance to the crowd that has the most frequent changes in crowd pattern (e.g. crowd shape, speed).

[1] For interpretation of color in Fig. 7, the reader is referred to the web version of this article.

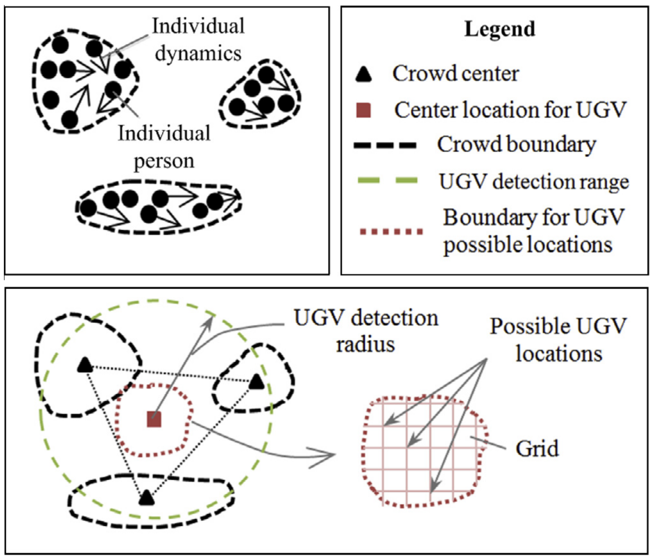

Fig. 7. Estimation of UGV locations based on 3 crowd observations.

- Select the destination location with the least number of blocks in the field of view.

The planning horizon can involve multiple control intervals over the path, which means multiple destinations need to be estimated. Under this circumstance, the above-mentioned sample strategies can be combined together and switched over time to optimize our MOE (i.e. crowd coverage) in the surveillance mission.

Upon the determination of the destinations of UAV/UGV, the corresponding path can be planned via motion planning algorithm. As discussed in Section 3.1.3, when different objective functions are used, the resulting optimal path can be different. Fig. 8 depicts three different paths based on the two objectives discussed in motion planning module, and a weighted combination of both objectives.

To balance the trade-offs between the two objectives discussed in motion planning module and increase the endurance (i.e. reduce the fuel consumption) for UAVs and UGVs, weighting method with normalized objectives is used in our work. The generic normalization formula can be written in Eq. (15).

$$\overline{f_a}(\mathbf{D}_V^{(s)}) = \frac{f_a(\mathbf{D}_V^{(s)}) - m_a}{M_a - m_a} \tag{15}$$

where $f_a(\mathbf{D}_V^{(s)})$ (a = 1,2) are the ath objective associated with decision vector $\mathbf{D}_V^{(s)}$ ($\mathbf{D}_V^{(s)} = \{[x_V(t+1), y_V(t+1), z_V(t+1)], \ldots, [x_V(t+\Delta t), y_V(t+\Delta t), z_V(t+\Delta t)]\}$ during the sth control interval), and $m_a = \min f_a(\mathbf{D}_V^{(s)})$ and $M_a = \max f_a(\mathbf{D}_V^{(s)})$. Notice that $\overline{f_a}(\mathbf{D}_V^{(s)}) \in [0,1]$, and the lower and upper bound function values are transformed to 0 and 1, respectively, which eliminates the scaling issue among different objectives. Then the weighted average of the multiple objectives is expressed by a composite objective defined in Eq. (16).

$$\alpha_1 \overline{f_1}(\mathbf{D}_V^{(s)}) + \alpha_2 \overline{f_2}(\mathbf{D}_V^{(s)}) \tag{16}$$

where $\alpha_1, \alpha_2 > 0$ and $\alpha_1 + \alpha_2 = 1$. In our work, by adjusting the two weights in the objective functions (e.g. (0.9,0.1); (0.4,0.6)), different potential paths can be established, where these weights are evaluated by agent-based simulation in the integrated planner. When there are multiple feasible paths, some sample rules for determining the best UAV/UGV path can be applied and they are listed below:

- Select the path with the best field of view (i.e. least number of obstacles).
- Select the path with the least UAV/UGV speed changes (i.e. variance).
- Select the path with the least possibility of collision with other vehicles.
- Select the path with the least potential disturbance (e.g. wind, wet road).
- Select the path optimizing the combination of above-mentioned measures.

3.2. Decision module for DDDAMS

Basically, a DDDAMS system can be viewed to involve the following 3 generic steps:

(1) Sensory data collection and abnormality detection,
(2) Fidelity selection, assignment, and update,
(3) Simulation-based planning (evaluation) and control generation.

In our proposed framework (see Fig. 2), Step 2 (i.e. fidelity selection, assignment and update) are done by the Decision Module for DDDAMS, while Steps 1 and 3 are performed by the integrated controller, which interacts with Step 2 if necessary. The main focus of this section is on Step 2: specifically, defining fidelity (Section 3.2.1) and discussing the fidelity selection and assignment algorithms (Section 3.2.2). Before proceeding to them, first Step 1 is briefly discussed, and Step 3 is explained shortly after Step 2.

According to the UAV/UGV configuration, sensory data (e.g. image, video) can be continuously transmitted into the integrated controller through hardware interface, and the system performance is periodically calculated in the intervals of length Δt utilizing the collected data. The image and video transmission will consume different durations of time and communicational

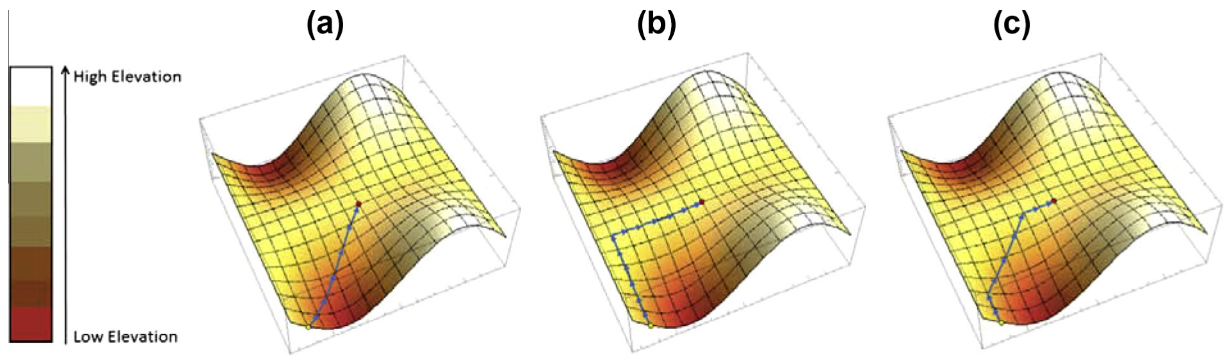

Fig. 8. Three UAV/UGV paths with minimizing (a) travel distance, (b) elevation penalty function, and (c) the composite function in Eq. (16) given the same weights.

resources (e.g. bandwidth) according to the levels of details inside the sensory data. This is different with the computational resources that are utilized by simulation model (discussed later in this section). After the sensory data being received, data processing can be performed to analyze the crowd location, speed, moving direction, and other motion characteristics for deriving the real system performance. The abnormality of the system is characterized by a single performance criterion – the crowd coverage percentage at each time (discussed before), while other criteria can be established under the same framework by building up the relationship of other system performances with the sensory data observation. Upon the real system performance being calculated, the performance deviation between simulated and real systems can be compared. If it exceeds a pre-defined threshold value, there are requirements for fidelity update in simulation. Fig. 9 illustrates the time synchronization between the DDDAMS (in the integrated planner) and real crowd control system based on the control interval Δt.

3.2.1. Defined fidelity levels

In this work, fidelity is defined as the degree of similarity (Hays & Singer, 1989) to which the simulation reproduces the real crowd control system in terms of both crowd dynamics and UAV/UGV behaviors. Under this definition, the fidelity represents how accurate the simulated model is when compared with the real system. Note that the "fidelity" and "simulation fidelity" are used interchangeably throughout the paper.

UAVs and UGVs play different roles in crowd information gathering. On one hand, UAV offers a wide area of searching and a global view of the crowd. Therefore, the size of visibility cell by UAV tends to be larger than that of UGV, and the overall group characteristics of the crowd such as the crowd shape and the number of dense regions can be roughly captured by UAV. However, the information accuracy in capturing the individual dynamics and the resulting crowd dynamics is restricted by other factors such as AGL altitude of UAV, UAV movement speed, onboard sensor capability (e.g. resolution), weather condition, and other disturbances (e.g. wind). Given this information, this work assumes that the UAV can only view the dense crowd regions, which means a cell is visible to the UAV only when the number of individuals in that cell exceeds a pre-defined threshold number (e.g. 5 people).

On the other hand, UGV as moving on the ground near crowds, the visibility cell by UGV is finer than that of the UAV. In addition, UGV exhibits accurate sensing capability of individuals and their various characteristics such as individual speed and moving pattern. However, as the information collected by UGV involves higher image and video resolution than that of UAV, transmitting such data requires a larger bandwidth, and further processing of these data and incorporating them into simulation need more computational resources. Depending on the mission purpose and available computational resources, the UAV and UGV data should be used in an integrated and intelligent way to balance the trade-offs between system performance and computational resource usage.

Based on this discussion, the two basic fidelity levels defined in this work can be stated as follows:

- In the low fidelity model, each dense crowd is considered as an agent that is visible to UAVs with coarse visibility cell.
- In the high fidelity model, each individual is considered as an agent under the view of UGVs with fine visibility cell.

The term of (visibility) cell is different with grid: grid refers to the minimum spatial unit that is considered by the motion planning algorithm as one-step movement, while cell refers to the minimum spatial unit inside which UAV/UGV can detect people. Fig. 10 shows the representations of different simulation fidelities incorporating UAV/UGV collected information.

It is noted that the two extreme simulation fidelity levels are the special cases being considered in this work. The generic extensions of the current fidelity definition can be accomplished in either the spatial or temporal direction or both. Considering an area that can be partitioned into multiple conjoint regions and one crowd region may involve multiple UAVs and UGVs performing the surveillance mission. At a particular time, the simulation may have both UAVs with dense crowd view and UGVs with the view of individuals, whose combinations constitute the current simulation fidelity lying between the lowest and highest fidelity levels. To improve performance, a particular crowd region also needs to change its simulation fidelity over time adapting to the computational resource usage. Time-persistent data can be collected to capture the mean and variance of simulation fidelity used in that region.

3.2.2. Fidelity selection and assignment

This section describes fidelity selection algorithm and fidelity assignment algorithm. Note that we do not claim that the algorithms described below are the best possibilities, as testing these algorithms requires fully automated DDDAMS system under

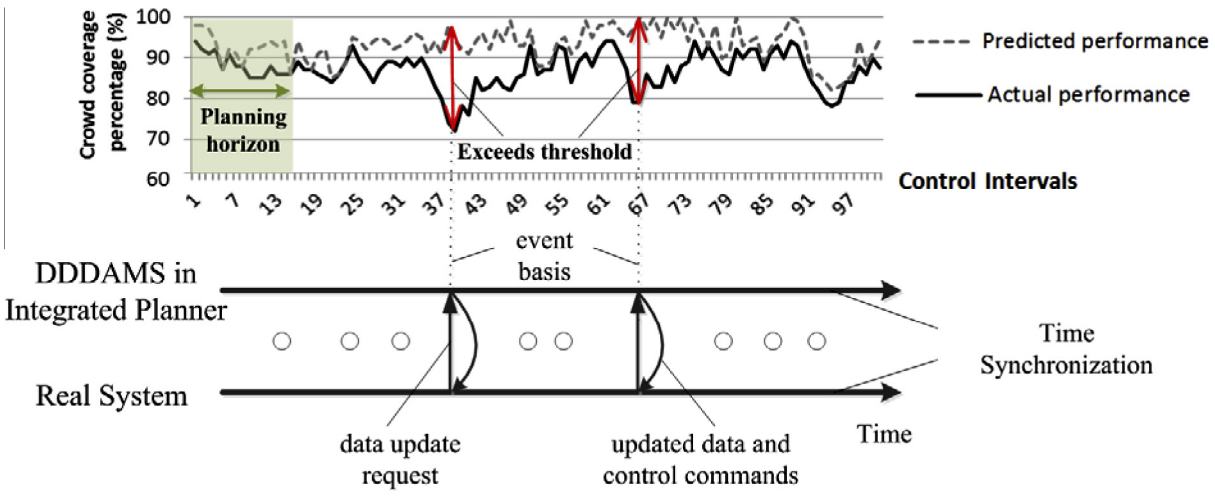

Fig. 9. Time synchronization between DDDAMS in the integrated planner and real system.

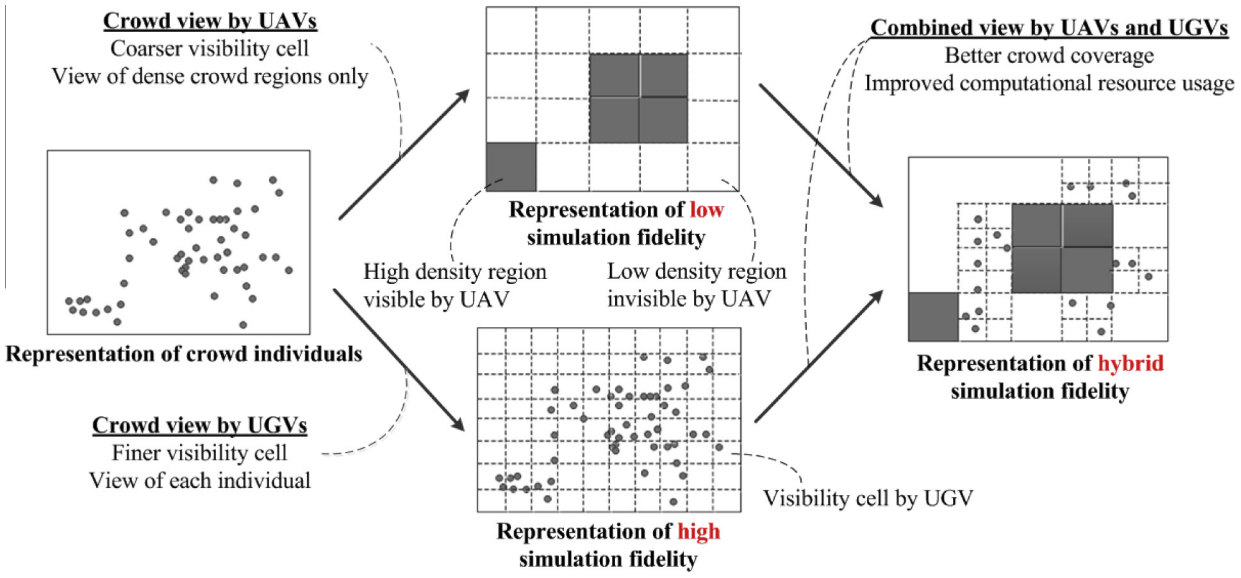
Crowd view by UAVs
Coarser visibility cell
View of dense crowd regions only
Representation of crowd individuals
Crowd view by UGVs
Finer visibility cell
View of each individual
Representation of low simulation fidelity
High density region visible by UAV
Low density region invisible by UAV
Representation of high simulation fidelity
Visibility cell by UGV
Combined view by UAVs and UGVs
Better crowd coverage
Improved computational resource usage
Representation of hybrid simulation fidelity

Fig. 10. Defined fidelity and its representation.

a pool of realistic scenarios, and we leave this topic as future research.

When a performance deviation is detected, the goal of the fidelity selection algorithm is to determine the proper simulation fidelity level (i.e. low or high) at each cell containing targets (i.e. people) utilizing the information UAVs/UGVs collected. Possible reasons for performance deviation are as follows:

- Crowd movement speed increases or decreases unexpectedly.
- Crowd motion pattern changes unexpectedly.
- Environmental condition (e.g. weather) changes unexpectedly.

In practice, historical data is usually available on the triggering threshold condition of fidelity update based on the performance deviation. Also, given different abnormality sensory data (e.g. crowd speed, pattern, and weather condition), the initial relationship between abnormality sensory observations and the corresponding reactions (e.g. which fidelity level should be chosen for simulation for planning and control generation) can be extracted from the historical data. These practices indicate to use a probabilistic approach to figure out the proper fidelity levels based on the newly collected sensory data as well as historical stored data. One such approach is Bayesian Belief Network (BBN) (Jensen, 2001; Pourret, Naïm, & Marcot, 2008), which infers the current occurring probability of various different random variables/events in the network based on the historical data and newly collected information. Celik et al. (2010) utilized this approach for the fidelity selection algorithm when addressing multi-scale supply chain decision making problems.

In general, there are many different fidelity levels considering a particular crowd region; however, we simplify our case by looking only at each cell, in which only one UAV or UGV is considered. Notice that there are only two fidelity levels (i.e. high and low) considered, we utilize the following rules for fidelity selection:

- Update the modeling scale of a cell from the high to low fidelity, under the following conditions: (1) the fidelity level is high at the current time, (2) for the previous fixed (τ) control intervals ($\tau\Delta t$), the performance deviation did not exceed the pre-defined threshold.
- Update the modeling scale of a cell from low to high fidelity, when the performance deviation exceeds a pre-defined threshold.
- For all other conditions, the previous simulation fidelity of the considered cell is unchanged.

Essentially, the fidelity selection algorithm dynamically releases system computational resources (from high to low fidelity level) and improves system performance (by changing simulation fidelity level from low to high, which drives more accurate decision evaluation for the control purpose). Computational issue becomes an important concern when applying the outcomes from the fidelity selection algorithm. There is a high chance that the available computational resource cannot afford to process all the high simulation fidelities decided from the fidelity selection algorithm. Under this circumstance, another algorithm (i.e. fidelity assignment) should be used to filter out the more significant cells/crowds in terms of better enhancing system performance, and prioritize them for utilizing the limited computational resource to achieve a finer modeling scale (higher simulation fidelity).

With this consideration, fidelity assignment is to re-select the proper fidelity levels found by the fidelity selection algorithm given the available computational resource as a constraint. The computational resource usage for the selected simulation fidelity of a cell is measured with the amount of minimal computational units consumed. The simulation of a particular cell involves computational resource for modeling crowds, environment conditions, and their interactions. It is noted that the total computational resource is a fixed number given a particular system configuration. The utility optimization problem can be formulated as 0–1 knapsack problem as shown in Eqs. (17) and (18).

$$Max\sum_{l=1}^{L} g(fd_l(t), l)q_l(t) \tag{17}$$

Subject to

$$\sum_{l=1}^{L} R(fd_l(t), l)q_l(t) \leqslant TR \tag{18}$$

where L is total number of cells considered in the system; $g(fd_l(t), l)$ is the utility value if the computational resource requirement for simulating the data associated with lth cell is satisfied, which is a function of selected fidelity level at that particular cell at time t; $R(fd_l(t), l)$ represents the computational resource requirement for simulating the data associated with the lth cell which is a function of selected fidelity level at that particular cell at time t; TR

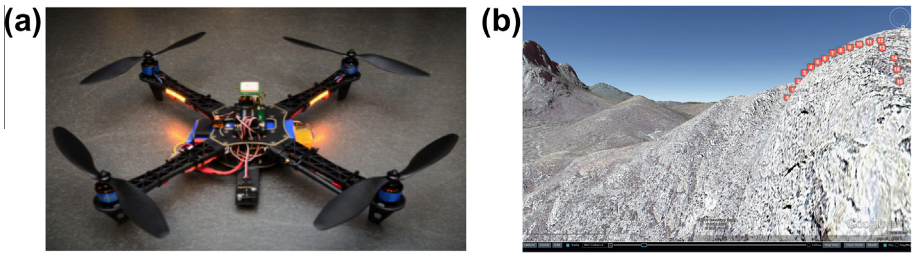

Fig. 11. Assembled UAV system and snapshot of QGroundControl® tool acting as hardware interface.

Table 3
Sample waypoint file format generated by agent-based simulation.

Index	Current waypoint	Coordinate frame	Command	Param1	Param2	Param3	Param4	Param 5 (longitude)	Param 6 (latitude)	Param 7 (AGL altitude)
0	1	0	16	0	5	0	0	31.40029	−110.4256	30
1	0	0	16	0	5	0	0	31.40033	−110.4256	30
2	0	0	16	0	5	0	0	31.40036	−110.4256	30
3	0	0	16	0	5	0	0	31.40039	−110.4256	30
4	0	0	16	0	5	0	0	31.40043	−110.4256	30

represents the total computational resources available in the system; and $q_l(t)$ is the 0–1 decision variable, where 1 represents the case when the required computational resource is assigned to the lth cell at time t, and 0 otherwise.

In our current system, the utility value given a selected fidelity level was pre-defined by offline measurements; however, various other methods can be also employed to assign the most effective utility value during the simulation run. In general, given the corresponding computational resource requirement for modeling a particular cell is satisfied, the utility value of assigning the cell with corresponding fidelity relates to either system performance improvement (if the cell is to be modeled in high fidelity) or computational resource saving (if the cell is to be modeled in low fidelity). To achieve effective fidelity assignment, one strategy is to rank the candidate cells based on their importance index under these criteria (from significant improvement or saving to insignificant), and select from the ranking pool when assigning the fidelity levels. In this way, the entire system can dynamically adjust its computational resource to achieve the best system performance. As a future work, more sophisticated computational resource allocation approaches (e.g. gradient approach, ranking and selection approach) will be employed for the fidelity assignment purpose. Effects of increment usage in computing budget will also be analyzed.

After the corresponding fidelity levels of all considered cells are assigned, the fidelity update request can be sent to the strategy maker and command generator (see Fig. 2). The simulation system with an updated fidelity level performs decision planning for the best control strategy selection (see Section 3.1.4). Then the command generator generates control commands (e.g. speed, waypoints) with the updated fidelity and incorporated control strategies. Then the simulated and real system will run to the next time (time synchronization point) and the entire process is repeated.

4. System implementation and experiments

4.1. System implementation (planning/control framework and real system)

The proposed planning and control framework (see Fig. 2) has been implemented employing a number of state-of-the-art software tools: (1) Repast Simphony® (ABS tool) for command generator and strategy maker, (2) QGroundControl® for hardware interface. It is noted that the same simulator used in the strategy maker (running in fast-mode) is used as the command generator (running in real-time mode). To represent a real system to demonstrate the proposed planning and control framework, both real hardware components as well as the simulated components are used under the real-time hardware-in-the-loop (HIL) simulation environment, (3) manually assembled quad-copter for UAVs, (4) social force model implemented in Repast Simphony® running in real-time for simulated crowd behaviors, (5) GIS 3D data from NASA World Wind® for simulated environment (i.e. elevation data) in Repast Simphony®, and (6) radio communications in 915 MHz with air data rates up to 250 kbps and approximate range of 1 mile.

In addition, HIL simulation environment has been employed to represent a real system as conducting full scale tests with a mission of UAVs and UGVs at the physical region in an early stage of the planning and control system development (which is the case for our work) involves high costs and potential risks. On the other hand, while the testing can be performed based on computer simulation only without involving hardware components, the results may suffer due to the misleading assumptions and lack of real world environments. In the proposed HIL setting, hardware components provide realistic parameter settings to the simulated agents (for mimicking the hardware behavior) that are used in the simulation as the integrated planner. After significant testing of the proposed planning and control framework involving the HIL setting, a full scale testing in an open field with all real system is left as a future work.

As the hardware component, our assembled UAV (see Fig. 11(a)) has capability to remain stable while carrying an additional payload of up to 1200 g, and fly for up to 30 min with one 5000 mAh 11.1 V lithium polymer battery. This vehicle is equipped with the GPS sensor and the autopilot unit (APM 2.5) for autonomous navigation to given waypoints outdoors with an accuracy of about +/−5 meters despite the environmental influences (e.g. wind). High-accuracy sonar unit has been used for holding the altitude and position with centimeter accuracy.

The connection between Repasst Simphony® and real UAV is achieved via a hardware interface known as QGroundControl® (http://qgroundcontrol.org/), through which Repast Simphony® sends waypoints and control commands to the real UAV and

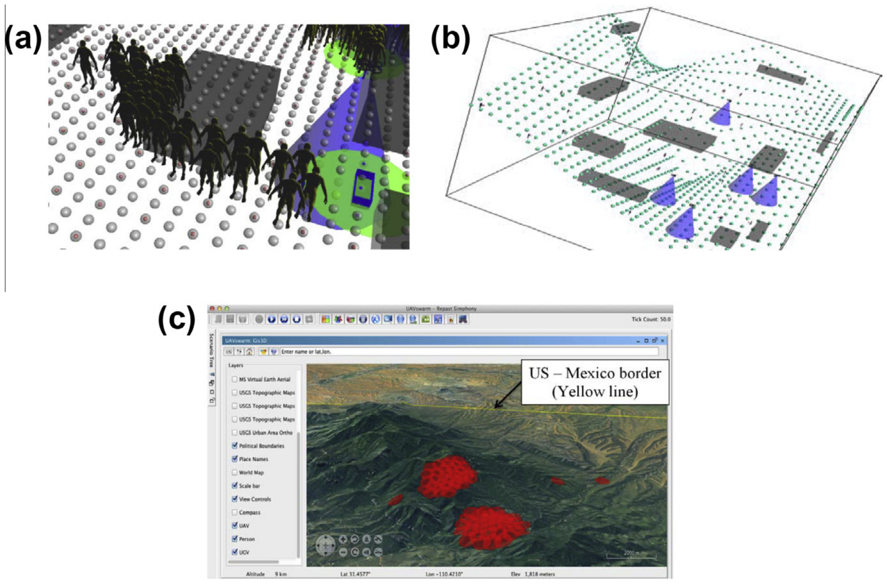

Fig. 12. ABS in 3D contionous and 3D GIS environment in Repast Simphony®.

receives sensory data (e.g. image, video streams) and telemetry data (e.g. GPS position) back. Fig. 11(b) shows an example for a sequence of waypoints in QGroundControl®, in which each red dot represents a waypoint along the vehicle path. The generated waypoints and control comanands for UAVs/UGVs are in MAVlink format as shown in Table 3, in which various parameters are included (e.g. latitude, longitude and AGL altitude based on motion planning results). Through this integration, the simulated UAVs in Repast Simphony® behave similarly with the real UAV in terms of moving speed, velocity, vision range, detection capability, tracking and so on.

Fig. 12 demonstrates ABS implementation in Repast Simphony® under two different environment conditions: (1) simulation in a relative flat area (no elevation change) (Fig. 12(a)), (2) simulation in a mountain area with lots of elevation changes (Fig. 12(b)). Three types of agents are implemented in Repast Simphony®: (1) UAV, (2) UGV and (3) individual person. For UAV and UGV, they can be in the state of searching (nothing detected) or tracking (following detected targets), and the blue and green cones represent the detection range of UAV and UGV, respectively. For crowd, social force model is employed to simulate each individual's behavior considering the interactions with others and obstacles. In 3D continuous environment (see Fig. 12(a) and (b)), the dark rectangles are the obstacles and the dots are the terrain elevation data extracted from 3D GIS environment (Fig. 12(c)). The simulation in the relative flat area is used to calculate system performance and computational resource usage in the next sub-section.

4.2. Preliminary results

To illustrate and demonstrate the benefit of the proposed framework, a hypothetical case study has been conducted for surveillance and crowd control scenario considering two different fidelities as discussed in Section 3.2. A border region in the state of Arizona in US with Mexico of size of 92120 m^2 (i.e. 329 m $*$ 280 m) was selected and a crowd of 80 individuals in two separate groups was considered in Repast Simphony®, where the comfortable walking speed of each individual was assumed to be 1.5 m/s subject to changes due to the interactions with other individuals using social force model. Crowd locations were observed in each 1 s in simulation. The crowd tracking module utilized these observations and predicted the crowd locations after 10 s in both fidelities. In the motion planning module, the high and low simulation fidelities utilize the predicted locations from UGV and UAV, respectively. Furthermore, different spatial resolutions (grid sizes) have been considered to perform motion planning in high fidelity (i.e. number of grids: 60 $*$ 60) and low fidelity (i.e. number of grids: 30 $*$ 30) due to different specifications of real hardware.

Fig. 13(a) demonstrates the performance of the two fidelity scenarios as crowd coverage percentage in intervals of 10 s. As seen in the figure, high fidelity has higher performance due to the utilization of individual observations in tracking module as well as more accurate locations of vehicle motion planning under finer grid. However, more computational resources are needed in the

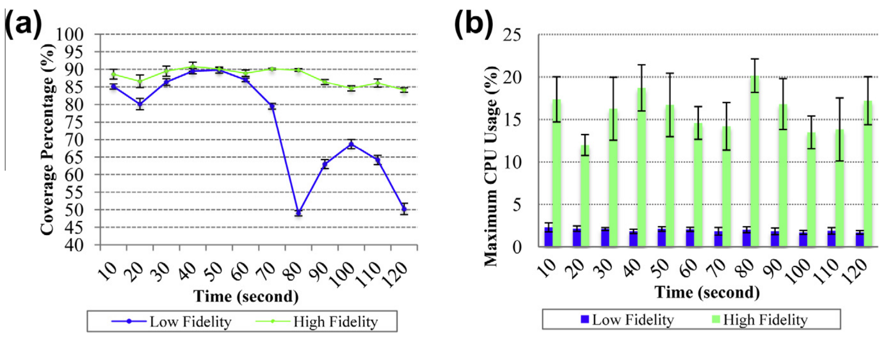

Fig. 13. System performance and CPU usage under two simulation fidelity cases with 95% confidence interval.

high fidelity. Fig. 13(b) illustrates CPU usages of the tracking module under situations with the high and low simulation fidelity, and such difference mainly lies in the dimension of state vectors used in the tracking algorithm. In the case with low fidelity, the center of the crowd dense region was collected and inputted to the tracking algorithm, which leads to a state vector of dimension of 4. In contrast, under the case with high simulation fidelity, under which every individual in the group was tracked in the crowd tracking algorithm, the state vector has a dimension that is a product between 4 and the total number of individual persons in the group. For our experiments with 40 individuals in each group, the state vector has a dimension of 160, which consumed more computations during the tracking. As Fig. 13(a) shows from time 10 s to 70 s, the two fidelities result in similar performance, which justifies the low fidelity case can maintain similar system performance with high fidelity case under certain circumstances.

Then, a hypothetical event (i.e. some people start to lead the entire group) was purposely created at time 70 (seconds), which caused 4 individuals in each group increase their comfortable walking speed from 1.5 m/s to 1.7 m/s. As the results show, due to this event the low fidelity performance has been decreased dramatically after some delay (this delay is the time for the leaders separating from the major group). As the result shows, the performance in high fidelity has also been reduced, but compared with that of the case with low fidelity, the effect of the event is much less due to tracking individuals instead of that of crowd. By adopting high fidelity, the required performance (80% coverage percentage) is maintained, but more computational resources are consumed.

5. Conclusions and future works

In this research, we proposed a comprehensive DDDAMS planning and control framework for effective and efficient surveillance and crowd control via UAVs and UGVs. The proposed framework has been discussed in details with the focus on crowd detection, tracking, motion planning, control strategy, defined fidelity, fidelity selection and assignment algorithms. The planning, control system and real-system have been implemented in an agent-based simulation environment under different settings (i.e. fast running mode, real-time mode, hardware-in-the-loop setting) considering the GIS data from NASA World Wind®. Two different fidelities were defined and investigated for crowd tracking, where in the high fidelity, groups of crowds were observed and tracked, and in the low fidelity, tracking was done using the observation data of individuals in the crowd. Furthermore, the system performance and computational usage for the two fidelities have been examined. The obtained results have revealed the advantage of dynamic fidelity switching by saving computational resources, while keeping the system performance within the desired level. In our experiment, we considered computational usage of tracking module only. For the settings where crowd detection is performed involving computer vision algorithms (which involve high computational expenses), saving with the proposed dynamic switching between different fidelity levels is expected to be even higher.

Future studies can be conducted in the following aspects: (1) investigating the effects of crowd detection algorithm involving real vision algorithms on system performance and computational resource usage under the proposed DDDAMS framework, (2) studying different approaches for crowd tracking considering measurement error and camera resolution, (3) extending the current planning and control framework for the UAV/UGV swarm control and coordination mechanism, (4) considering more fidelity levels and studying effective, efficient and robust algorithms for fidelity selection and assignment, and (5) testing a full scale demonstration of the DDDAMS in an open field with all the real components.

Acknowledgment

This work is financially supported by the Air Force Office of Scientific Research under FA9550-12-1-0238.

References

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Systems Science and Cybernetics, 50*, 174–188.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25–71). Springer.

Carnahan, J. C., & Reynolds, P. (2006). Requirements for DDDAS flexible point support. In *Simulation conference, 2006. WSC 06. Proceedings of the winter* (pp. 2101–2108). IEEE.

Celik, N., Lee, S., Vasudevan, K., & Son, Y.-J. (2010). DDDAS-based multi-fidelity simulation framework for supply chain systems. *IIE Transactions, 42*, 325–341.

Choset, H. M. (2005). *Principles of robot motion: Theory, algorithms, and implementation*. Bradford Books.

Darema, F. (2004). Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *Computational science-ICCS 2004* (pp. 662–669). Springer.

Dechter, R., & Pearl, J. (1985). Generalized best-first search strategies and the optimality of A*. *Journal of the ACM (JACM), 32*, 505–536.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik, 1*(1), 269–271.

Douglas, C. C., Loader, R., Beezley, J., Mandel, J., Ewing, R., Efendiev, Y., et al. (2006). DDDAS approaches to wildland fire modeling and contaminant tracking. In *Simulation conference, 2006. WSC 06. Proceedings of the winter* (pp. 2117–2124). IEEE.

Everitt, B., Landau, S., & Leese, M. (2001). *Cluster analysis*. London: Arnold.

Gilmore, J. F. (1991). Autonomous vehicle planning analysis methodology. *Association for Unmanned Vehicles Systems, 91*(12–16), 199.

Girard, A. R., Howell, A. S., & Hedrick, J. K. (2004). Border patrol and surveillance missions using multiple unmanned air vehicles. *43rd IEEE conference on decision and control, 2004. CDC* (Vol. 1, pp. 620–625). IEEE.

Gning, A., Mihaylova, L., Maskell, S., Pang, S. K., & Godsill, S. (2011). Group object structure and state estimation with evolving networks and Monte Carlo methods. *IEEE Transactions on Systems Science and Cybernetics, 59*, 1383–1396.

Goerzen, C., Kong, Z., & Mettler, B. (2010). A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems, 57*, 65–100.

Grocholsky, B., Keller, J., Kumar, V., & Pappas, G. (2006). Cooperative air and ground surveillance. *Robotics & Automation Magazine, IEEE, 13*, 16–25.

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: Concepts and techniques*. Morgan Kaufmann.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*, 100–107.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons Inc..

Hays, R. T., & Singer, M. J. (1989). *Simulation fidelity in training system design: Bridging the gap between reality and training*. Springer-Verlag. New York, NY.

Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 34*, 334–352.

Jensen, F. V. (2001). *Bayesian networks and decision graphs. Statistics for engineering and information science* (Vol. 32). Springer [p. 34].

Jones, P. J. (2009). Cooperative area surveillance strategies using multiple unmanned systems. ProQuest.

Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE, 92*, 401–422.

Kavraki, L. E., & Latombe, J. -C. (1998). Probabilistic roadmaps for robot path planning. In K. Gupta & P. del Pobil (Eds), Practical Motion Planning in Robotics: Current Approaches and Future Directions (pp. 33–53). John Wiley & Sons LTD.

Khaleghi, A. M., Xu, D., Lobos, A., Minaeian, S., Son, Y. -J., & Liu, J. (2013). Agent-based hardware-in-the-loop simulation for modeling UAV/UGV surveillance and crowd control system. In *Proceedings of the winter simulation conference 2013, Washington, DC, USA*.

Loyall, J., Schantz, R., Corman, D., Paunicka, J., & Fernandez, S. (2005). A distributed real-time embedded application for surveillance, detection, and tracking of time critical targets. In *11th IEEE real time and embedded technology and applications symposium, 2005. RTAS 2005* (pp. 88–97). IEEE.

Madey, G. R., Blake, M. B., Poellabauer, C., Lu, H., McCune, R. R., & Wei, Y. (2012). Applying DDDAS principles to command, control and mission planning for UAV swarms. *Procedia Computer Science, 9*, 1177–1186.

Marsh, W. E. (2007). *An Initial methodology for the definition and implementation of unmanned aerial vehicle agent behaviors*. Wright State University.

McLachlan, G. J., & Krishnan, T. (2007). *The EM algorithm and extensions* (Vol. 382). Wiley-Interscience.

Nash, A., Daniel, K., Koenig, S., & Felner, A. (2007). Theta^∗:Any-angle path planning on grids. *Proceedings of the national conference on artificial intelligence* (Vol. 22, pp. 1177). Menlo Park, CA; Cambridge, MA; London: AAAI Press; MIT Press [1999].

Pavlovic, V., Rehg, J. M., & MacCormick, J. (2001). Learning switching linear models of human motion. *Advances in Neural Information Processing Systems*, 981–987.

Pingali, K., & Stodghill, P. (2004). *O'SOAP–A web services framework for DDDAS applications. Computational science-ICCS2004*. Springer (pp. 797–804). Springer.

Pourret, O., Naïm, P., & Marcot, B. (2008). *Bayesian networks: A practical guide to applications* (Vol. 73). Springer.

Quigley, M., Goodrich, M. A., Griffiths, S., Eldredge, A., & Beard, R. W. (2005). Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera. In *Proceedings of the 2005 IEEE international conference on robotics and automation, 2005. ICRA 2005* (pp. 2600–2605). IEEE.

Raffetto, M. (2004). Unmanned aerial vehicle contributions to intelligence surveillance and reconnaissance missions for expeditionary operations. In DTIC document.

Ristic, B., Arulampalm, S., & Gordon, N. (2004). *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers.

Rodríguez, R., Cortés, A., & Margalef, T. (2009). Injecting dynamic real-time data into a DDDAS for forest fire behavior prediction. In *Computational science-ICCS 2009* (pp. 489–499). Springer.

Sauter, J. A., Mathews, R. S., Yinger, A., Robinson, J. S., Moody, J., & Riddle, S. (2008). Distributed pheromone-based swarming control of unmanned air and ground vehicles for RSTA. In *SPIE defense and security symposium* (pp. 69620C–69620C-69612): International Society for Optics and Photonics.

Silbert, M., Mazzuchi, T., & Sarkani, S. (2011). Comparison of a grid-based filter to a Kalman filter for the state estimation of a maneuvering target. In *SPIE optical engineering+ applications* (pp. 81370J–81370J-81310). International Society for Optics and Photonics.

Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. In *Proceedings IEEE International Conference on Robotics and Automation, 1994* (pp. 3310–3317). IEEE.

Surveillance. (n.d.). In *Wikipedia*. Retrieved May 31, 2013. Available from: https://en.wikipedia.org/wiki/Surveillance.

Unmanned aerial vehicle. (n.d.). In *Wikipedia*. Retrieved May 31, 2013. Available from: http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle.

Wang, Z., Li, M., Khaleghi, A. M., Xu, D., Lobos, A., Vo, C., et al. (2013). DDDAMS-based crowd control via UAVs and UGVs. *Procedia Computer Science, 18*, 2028–2035.

Washburn, A. R., & Kress, M. (2009). *Combat modeling* (Vol. 134). Springer.

Wei, Y., Blake, M. B., & Madey, G. R. (2013). An operation-time simulation framework for UAV swarm configuration and mission planning. *Procedia Computer Science, 18*, 1949–1958.

Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter.

Wichmann, D. R., & Wuensche, B. C. (2004). Automated route finding on digital terrains. In *Proceedings of IVCNZ* (Vol. 4, pp. 107–112). Citeseer.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*, 645–678.