AFRL-RI-RS-TR-2021-033

DECENTRALIZED CLASSIFICATION AND COORDINATION WITH NON- PERMISSIVE COMMUNICATIONS

NORTHEASTERN UNIVERSITY

FEBRUARY 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-033 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
CHRISTOPHER A. OKONKWO
Work Unit Manager

/ S /
SCOTT D. PATRICK
Chief, Intelligence Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
FEBRUARY 2021	FINAL TECHNICAL REPORT	SEP 2018 – SEP 2020

4. TITLE AND SUBTITLE

DECENTRALIZED CLASSIFICATION AND COORDINATION WITH NON- PERMISSIVE COMMUNICATIONS

5a. CONTRACT NUMBER
FA8750-18-2-0043

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

Jose Martinez Lorenzo

5d. PROJECT NUMBER
E2SC

5e. TASK NUMBER
NE

5f. WORK UNIT NUMBER
UN

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northeastern University
360 Huntington Av, 418 ISEC Engineering
Boston MA 02115

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIED
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-033

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report presents the results achieved in terms of three main topics:

- The experimental validation of a distributed team of small Unmanned Aerial Systems (SUAS) for performing a complex behavior in coordination.

- The development of a realistic multi-drone simulator to apply Reinforcement Learning Techniques to coordinate a group of SUAS for a particular purpose.

- The design and validation of a fused optical camera with an active Multiple-Input-Multi- ple-Output (MIMO) mm-wave radar sensor mounted on a downward-looking UAV.

15. SUBJECT TERMS
Decentralized Classification, Decentralized Coordination, Dynamic Tasking, Swarm Architecture, Decentralized Fusion, Auction-based coordination, variable communication

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**CHRISTOPHER A. OKONKWO**
U	U	U	UU	39	19b. TELEPHONE NUMBER (*Include area code*) **N/A**

TABLE OF CONTENTS

List of Figures ii
List of Tables iii
1.0 SUMMARY 1
2.0 INTRODUCTION 3
2.1 Validation of a Distributed Team of SUAS for complex behavior 3
2.2 Reinforcement Learning Applied to a Multi-drone Simulator 3
2.3 Conventional UAV-based SAR imaging and the novelty of our work 4
3.0 METHODS, ASSUMPTIONS, AND PROCEDURES 6
3.1 Distributed Team of SUAS 6
3.1.1 System Architecture 6
3.1.2 Multi-layer Perception 7
3.1.3 Multi-layer Policy 8
3.1.4 Multi-layer decisions 8
3.2 Reinforcement Learning Applied to a Multi-drone Simulator 8
3.2.1 Simulator of People Detection by a Team of Explorer Drones 8
3.2.2 Description of the Domain 9
3.2.3 Space of States, Observations, and Actions 10
3.2.4 Modeling of uncertainties 12
3.2.5 Problem Formulation 12
3.2.6 Multi-Target Search and Detection 13
3.2.7 Decentralized Advantage Actor-Critic (DA2C) 14
3.2.8 Network Architecture 16
3.3 SAR imaging experiment 16
4.0 RESULTS AND DISCUSSION 18
4.1 Results of the Team of Drones 18
4.2 Discussion of the drone swarming 21
4.3 Reinforcement Learning results 22
4.4 SAR imaging results 25
5.0 CONCLUSIONS 26
6.0 References 27
APPENDIX A – Publications and Presentations 32
LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS 33

LIST OF FIGURES

Figure 1. Schematic of synthetic aperture imaging using a downward-looking UAV equipped with mm-wave and optical sensors. 5

Figure 2. Decentralized autonomous systems: (left) team operator perspective; (right) agent in the team perspective..... 6

Figure 3. Architecture of multiple sensors fusion with Convolutional Neural Network. Inference results of the fusion module feed into our drone policy, which controls the motion of the drone or performs complex behaviors and output environmental prediction. 7

Figure 4. Flow chart for each drone in the team 9

Figure 5. Top view of the drone cage at Kostas Research Institute, where the simulated drones and people are represented..... 10

Figure 6. Left: Current state of a drone in the simulator. Right: Legend of symbols, (a) Flying drone, (b) Non-operative drone, (c) Non-detected person, and (d) Detected person. 10

Figure 7. Inference for range estimation. Left: Bounding box of person detected from the RealSense camera. Center: Raw depth information, from 0 to 6.55m. (The pixels interpreted farther than the maximum distance are set to 0). Right: Image combination of the raw depth information with the bounding box detection. 11

Figure 8. Representation of the uncertainties affecting the flight of the drones. 12

Figure 9. Overview of our multi-agent decentralized actor, critic approach. 15

Figure 10. SAR imaging experiment setup to detect a metallic box and bar. Both mm-wave and optical sensors are inte-grated in the UAV. 17

Figure 11. Multi-Swarm mission description..... 18

Figure 12. Autonomous Multi-swarm demonstration. 1) Sub-swarm #1 searches for a person, who is followed until it enters a car; 2) Sub-swarm #2 tracks the car around the “simulated road”; 3) Sub-swarm #3 tracks the person after it exits the car. 19

Figure 13. Overall sequence of the whole mission recorded by a stationary ground camera and a drone camera. 20

Figure 14. Multi-swarm performance. For each swarm, the right bar is desired performance for each task, while the left bar is the actual performance. 21

Figure 15. Average reward and standard deviation per episode in an environment with three drones and three targets. 23

Figure 16. The total reward and standard deviation achieved by our learned policy vs. a random policy and a collision-free policy, averaged over 500 episodes. 24

Figure 17. The average reward achieved by different team sizes, ranging from two drones to six drones. 24

Figure 18. The average reward achieved by a team of 3 drones for various number of targets... 24
Figure 19. (a.) and (b.) are the real optical image co-registered with mm-wave reconstructed reflectivity and depth information, respectively. 25

LIST OF TABLES

Table 1 Parameters of DA2C 22

1.0 SUMMARY

This report presents the results achieved in terms of three main topics:

- The experimental validation of a distributed team of small Unmanned Aerial Systems (SUAS) for performing a complex behavior in coordination.
- The development of a realistic multi-drone simulator to apply Reinforcement Learning Techniques to coordinate a group of SUAS for a particular purpose.
- The design and validation of a fused optical camera with an active Multiple-Input-Multiple-Output (MIMO) mm-wave radar sensor mounted on a downward-looking UAV.

The work related to the validation of the team of SUAS presents and experimentally test the framework used by our context-aware, distributed team of SUAS capable of operating in real-time, in an autonomous fashion, and under constrained communications. Our framework relies on three layered approach: (1) Operational layer, where fast temporal and narrow spatial decisions are made; (2) Tactical Layer, where temporal and spatial decisions are made for a team of agents; and (3) Strategical Layer, where slow temporal and wide spatial decisions are made for the team of agents. These three layers are coordinated by an ad-hoc, software-defined communications network, which ensures sparse, but timely delivery of messages amongst groups and teams of agents at each layer even under constrained communications. Experimental results are presented for a team of 10 small unmanned aerial systems tasked with searching and monitoring a person in an open area. At the operational layer, our use case presents an agent autonomously performing searching, detection, localization, classification, identification, tracking, and following of the person, while avoiding malicious collisions. At the tactical layer, our experimental use case presents the cooperative interaction of a group of multiple agents that enable the monitoring of the targeted person over a wider spatial and temporal region. At the strategic layer, our use case involves the detection of complex behaviors--i.e. the person being followed enters a car and runs away, or the person being followed exits the car and runs away--that requires strategic responses to successfully accomplish the mission.

Targets search and detection encompasses a variety of decision problems such as coverage, surveillance, search, observing and pursuit-evasion along with others. We develop a multi-agent deep reinforcement learning (MADRL) method to coordinate a group of aerial vehicles (drones) for the purpose of locating a set of static targets in an unknown area. To that end, we have designed a realistic drone simulator that replicates the dynamics and perturbations of a real experiment, including statistical inferences taken from experimental data for its modeling. Our reinforcement learning method, which utilized this simulator for training, was able to find near-optimal policies for the drones. In contrast toother state-of-the-art MADRL methods, our method is fully decentralized during both learning and execution, can handle high-dimensional and continuous observation spaces, and does not require tuning of additional hyperparameters.

In order to develop a decentralized classification and coordination framework for SUAS operating under constrained communications, our first goal is to build a multi-sensor system on an Unmanned Aerial Vehicles (UAV) for high detection performance. It is known that optical and thermal sensors mounted on UAVs have been successfully used to image difficult-to-access regions. Nevertheless, none of these sensors provide range information about the scene; and, therefore, their fusion with high-resolution mm-wave radars has the potential to improve the performance of the imaging system. We propose our preliminary experimental results of a downward-looking UAV system equipped with a passive optical video camera and an active Multiple-Input-Multiple-Output (MIMO) mm-wave radar sensor. The 3D imaging of the mm-wave radar is enabled by collecting data through the line of motion, thus producing a synthetic aperture, and by using a co-linear MIMO array perpendicular to the motion trajectory. Our preliminary results show that the fused optical and mm-wave image provides shape and range information, that ultimately results in an enhanced imaging capability of the UAV system.

2.0 INTRODUCTION

2.1 Validation of a Distributed Team of SUAS for complex behavior

Recent advancements in the fields of Artificial Intelligence [1]-[7], Machine Learning [8]-[11], Robotics [12]-[17], and Signal Processing [18]-[20] have provided humankind with unique set of tools that, for the first time in history, have the potential to address some of the most important problems existing in the field of group autonomy of unmanned systems [21]. Nowadays, group autonomous systems require either direct human control of many systems [22], [23], contract and auction techniques [24], [25], and or coalition methods [26]-[28]. The latter are heavily dependent on the communications channel, which is often constrained in many realistic scenarios. Other approaches based on Markov Decision Processes do not scale linearly with the number of agents and states, and they often result in a slow reaction to unexpected events, [29]-[35].

This work describes and experimentally validates the framework---hardware, software, and system of systems architecture---used by our context-aware, distributed team of Small Unmanned Aerial Systems (SUAS) to be able to operate in real-time, in an autonomous fashion, and under constrained communications. Our framework relies on three layered approach: (1) Operational layer (fast temporal and narrow spatial scale; partially mimicking functionality of human's peripheral nervous system) - here a single agent performs on-board detection, localization, classification, identification, tracking, following while avoiding malicious collisions; this layer relies on hardware and software that enable to fuse and sparsify in real-time 4D full motion video, 4D millimeter wave radars, 4D infrared cameras using Deep Learning and 4D (space + time) Compressive Sensing (CS); (2) Tactical Layer (intermediate temporal and spatial scale; partially mimicking functionality of human's muscular system): here a group multiple autonomous agents collaborate to jointly perform a complex task that cannot be executed by a single agent due to their spatial (navigation) and temporal (perception) limitations; and (3) Strategical Layer (slow temporal and wide spatial scale; partially mimicking functionality of the endocrine system): here teams of multiple autonomous agents cooperate to jointly perform a multi-step complex task that cannot be executed by a group of autonomous agents due to their spatial (navigation), temporal (perception), and energy (endurance) limitations. These three layers are coordinated by an *ad-hoc*, software-defined communications network, which ensures sparse, but timely delivery of messages amongst groups and teams of agent even under constrained communications.

2.2 Reinforcement Learning Applied to a Multi-drone Simulator

Recent advancements in unmanned aerial vehicle (UAV) technology have made it possible to use them in place of piloted planes in complex tasks, such as search and rescue operations, map building, deliveries of packages, and environmental monitoring (see [36] for a recent survey).

This work handles the problem of coordinating a team of autonomous drones searching formultiple ground targets in a large-scale environment. The problem of searching and detecting targets in outdoor environments is relevant to many real-world scenarios, e.g., military and first response teams often need to locate lost team members or survivors in disaster scenarios.

Previous methods for target search by UAVs consisted of a division of the surveillance region into cells (e.g., Voronoi cells), and designing a path planning algorithm for each cell [37] - [39]. These

methods require direct communication among the drones, often handle poorly online UAV failures, and have no guarantee on the optimality of the final solution. In contrast, we propose a method based on deep reinforcement learning (DRL), which offers an end-to-end solution to the problem. Our method is fully decentralized (does not require any communication between the drones) and guaranteed to converge to a (local) optimum solution.

While DRL methods have recently been applied to solve challenging single-agent problems [40] - [42], learning in multi-agent settings is fundamentally more difficult than the single-agent case due to non-stationarity [43], curse of dimensionality [44], and multi-agent credit assignment [45].

Despite this complexity, recent multi-agent deep reinforcement learning (MADRL) methods have shown some success, mostly in simple grid-like environments and in game playing [46] - [48]. Most of existing MADRL methods employ the centralized training with decentralized execution approach, where the agents' policies are allowed to use extra information to ease training, as long as this information is not used at test time. This approach has several limitations, as it assumes noise-free communication between the robots during training, and also it does not allow the agents to adapt their policies to changing environmental conditions during execution (when global information is not available). Moreover, the discrepancy between the information available to the agents during training and execution often leads to instability of the learned policies in runtime.

In this work we propose a policy gradient MADRL method, which is fully decentralized during both learning and execution. Our method, called Decentralized Advantage Actor-Critic (DA2C), is based on extending the A2C algorithm [49] to the multi-agent case. To that end, we have developed our own simulator, that is, on one hand, simple and fast enough to generate a large number of sample trajectories; and, on the other hand, realistic enough, accounting for all the dynamics and uncertainties that can affect the deployment of the learned policies on a real team of drones.

We empirically show the success of our method in finding near-optimal solutions to the multi-target search and detection task. To the best of our knowledge, this is the first time that a fully decentralized multi-agent reinforcement learning method has been successfully applied to a large scale, real-world problem.

2.3 Conventional UAV-based SAR imaging and the novelty of our work

Commercial unmanned aerial vehicles (UAVs) equipped with integrated optical and thermal sensors are often used in aerial photography applications [50]. Specifically, optical cameras can be used to produce high resolution images and videos in two-dimensions (2D) [51], and thermal cameras can be used to enable wide-angle imaging. Unfortunately, the former do not provide range information and the latter usually present a degraded imaging performance when the difference between target and background temperature is small [52]. UAVs may also be used in many other industrial and civil applications in which neither thermal nor optical sensors are effective, such as finding metallic power lines, structural steel in reinforced concrete, or concealed security threats under clothing [53].

Millimeter wave (mm-wave) radar possesses the intrinsic ability to penetrate certain optically opaque layers, and it enables the reconstruction of target profiles in 3D [53]. These features serve as the rationale for developing enhanced UAVs inspection systems that combine optical, thermal, and millimeter wave (mm-wave) sensors [54], [55]. The concept of operation of such an UAV-

based multi sensor inspection system is displayed in Figure 1. Specifically, the multi sensor data is collected along the trajectory of motion; and the coherent post-processing of the mm-wave data across this path, the so-called synthetic aperture, enables 3D imaging of the scene. The fusion of co-registered data from different sensors results in enhanced imaging capabilities. It is worth noting that although optical and mm-wave sensors were individually tested in [54]-[56], no fused imaging results have been reported yet.

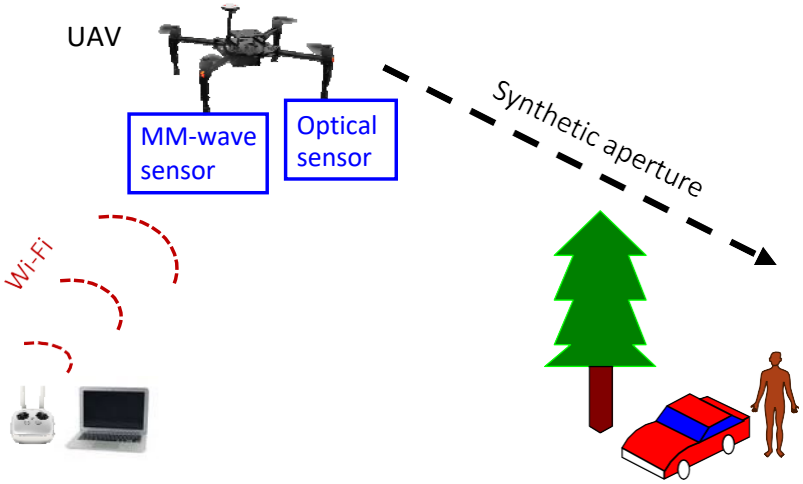

Figure 1 Schematic of synthetic aperture imaging using a downward-looking UAV equipped with mm- wave and optical sensors.

This work presents preliminary experimental results of our UAV system that is equipped with both an optical camera and a Multiple-Input-Multiple-Output (MIMO) mm-wave radar. The metallic objects in the 2D optical image are highlighted and ranged in 3D by its fusion with the mm-wave synthetic aperture radar (SAR) image; while the optical image contributes to reduce the unwanted background noise in the SAR image, thus resulting in an enhanced imaging performance. Our experimental results show a great potential for such a downward-looking UAV system in future inspection applications.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Distributed Team of SUAS

3.1.1 System Architecture

The hierarchical architecture adopted by our autonomous multi-agent system is the one presented in Figure 2. From an operator perspective, see the left side of the figure, a general mission is specified for a team of agents that must organize themselves to accomplish a particular mission. In this use-case work, the general mission is defined as *search and monitor people in a given region*. Based on this mission, an off-line planner parses a multi-layer policy (controller) to each agent in the network using a top-to-bottom approach. The latter leverages on the use of a set of *memory banks*, which resemble the different types of memories used by the human body, including: *(i)* long term strategic, which covers spatial priming memory and temporal procedural memory; *(ii)* long term tactical, which covers spatial semantic memory and temporal episodic memory; and *(iii)* short term memory and sensory memory. The strategic memory is used to load the initial strategic policy, as well as the type of decisions and observations available for the team of agents at this level. A similar functionality is provided to the tactical and operational memory, regarding decisions, observations and policies at its corresponding layer. From an agent perspective, our architecture enables each unmanned system to reason about its own operation, its tactical relationships with a subgroup of agents with whom it is cooperating in a joint task, and its strategical contribution to the overall mission. This perspective is shown on the right part of Figure 2, and a thorough description is described in the next section.

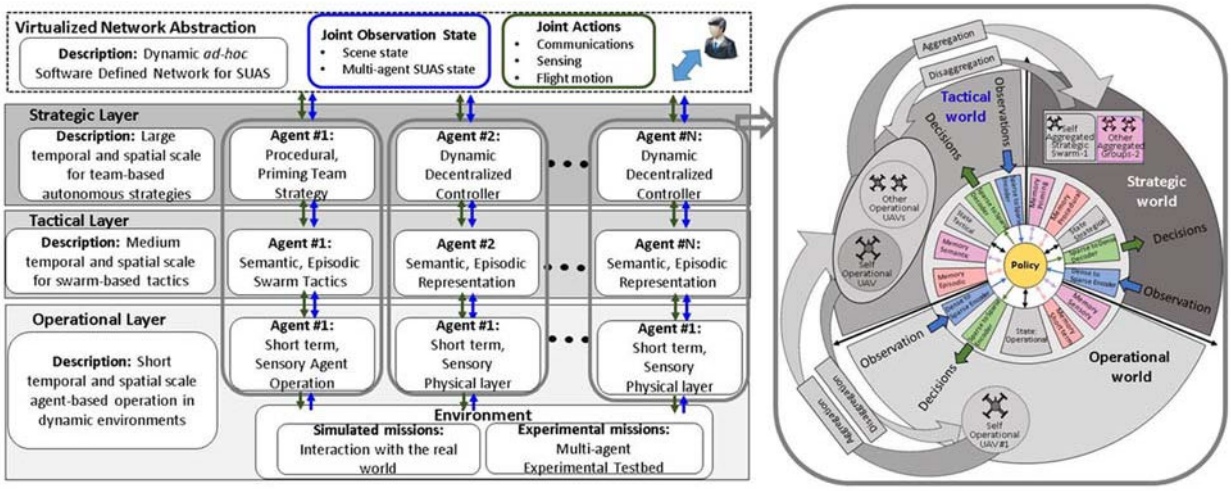

Figure 2. Decentralized autonomous systems: (left) team operator perspective; (right) agent in the team perspective.

3.1.2 Multi-layer Perception

As shown in Figure 3, the agents (SUAVs) in our network can be equipped with three different type of perception sensors: 4D RGB camera, 4D Infrared camera, and a 4D mmWave radar. At the operational level, the raw data of the three sensors is parsed into a Vector Processing Unit, which runs a fine-tuned Convolutional Neural Network to perform the sensor fusion and to provide a sparse representation of the scene. As it can be seen on the top area of Figure 3, our dense to sparse perception module is capable of outputting sparse information about the scene at a 5 Hz rate-- an enhanced frame rate of 100 to 1000 Hz should be achieved with our current architecture. This output contains a list of targets in the scene (e.g., person, car, etc.), classification confidence level for each target, targets' bounding boxes in 2D, targets' ranges from the agent, targets' angular location relative to the agent's orientation, as well as 4D GPS Geo-location of both targets and agent. At the tactical level, medium priority observations involving other agents within the same group, jointly performing a particular activity, is sparsely parsed through the *ad-hoc* network in an asynchronous fashion at a reduced average rate (~ 0.01 Hz per mission). Similarly, at the strategical level, top priority observations of events that require an update on the team strategy are parsed through the *ad-hoc* network in an asynchronous fashion at very low average rate (~ 0.005 Hz per mission).

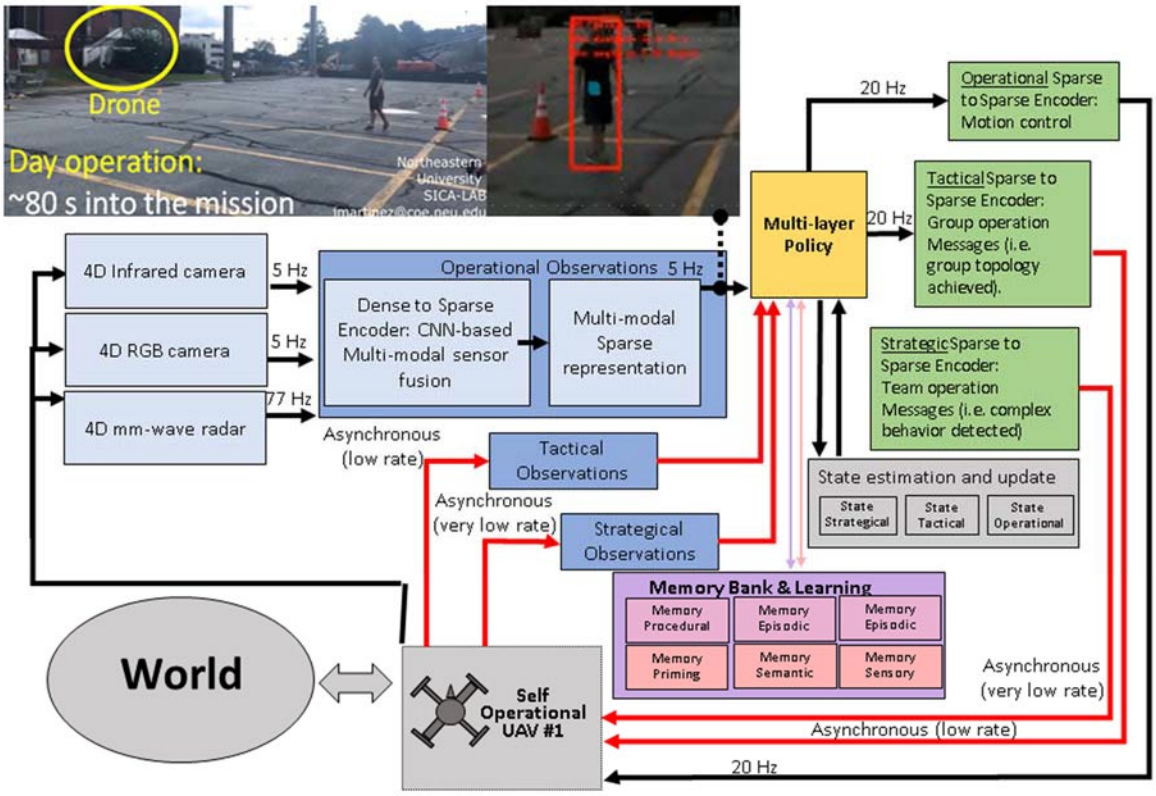

Figure 3. Architecture of multiple sensors fusion with Convolutional Neural Network. Inference results of the fusion module feed into our drone policy, which controls the motion of the drone or performs complex behaviors and output environmental prediction.

3.1.3 Multi-layer Policy

The vector targets at the operational layer are synchronously parsed to our multi-layer policy block (see Figure 3), which uses a sparse to sparse motion controller to generate motion trajectories based on the agent's particular state. At the operational level, each drone can simultaneously be active in one or more of the following operational states: *idle/sleeping*, *takeoff/landing*, *searching*, *following*, *tacking*, *Navigating to a GPS location*, *returning to base*. The latter sparsification at the operational state affords scalability of the Multi-layer policy. At the tactical level, the multi-layer policy handles the information received either by its own operational observations or from another member of its tactical group. The policy enables an asynchronous coordination of the group of agents at the tactical level. When a group of agents are not able to continue a particular group activity, the strategic policy may be able to recruit another group of agents that can finalize the mission in a suitable fashion. The strategic policy observes and controls the strategic perception and actuation channels. The use-case described below will clearly emphasize the type of observations and actions that are provided for each one of the components of the multi-layer policy.

3.1.4 Multi-layer decisions

The multi-layer policy creates a sparse vector that encodes the actions needed at the strategical, tactical, and operational level. In the latter, the *Operational sparse to sparse encoder* shown in Figure 3 generates the signals needed to control the lower-level motion controller. Our system follows a control approach similar to the one presented in [57]. Specifically, once target is recognized and localized, the drone will change its state to follow or track the object and update its position, $\mathbf{p}_t$, based on the change in position, $\Delta\mathbf{p}$, obtained from its own observations. Position

updates can be made by sending the flight controller either local velocity setpoints or local or global position setpoints. By receiving the angle and the distance of the object relative to its current position and orientation, the controller will decide how much to rotate and how to adjust its position. Various uncertainties, U, like external forces, J (e.g., wind), can affect the motion of the drone, which the flight controller needs to be capable of compensating for. Changes to the controller can also come from communication with other drones or other swarms, or from recognizing complex behavior or patterns. When a drone recognizes certain behavior occurring among the objects it is seeing (e.g., a person entering a car), it can communicate to the other drones to change their state (e.g., to return home) and to other swarms to begin or change their mission.

3.2 Reinforcement Learning Applied to a Multi-drone Simulator

3.2.1 Simulator of People Detection by a Team of Explorer Drones

A 2-D simulator has been designed in order to faithfully replicate the dynamics and detection capabilities of the Intel Aero Ready to Fly Drones. The mission of these drones, working as a team, is to detect and locate the position of a given number of people in a given domain in the most efficient way. In order to successfully accomplish the mission, each drone follows the flow chart described in Figure 4, which is based on the two main components: states and observations.

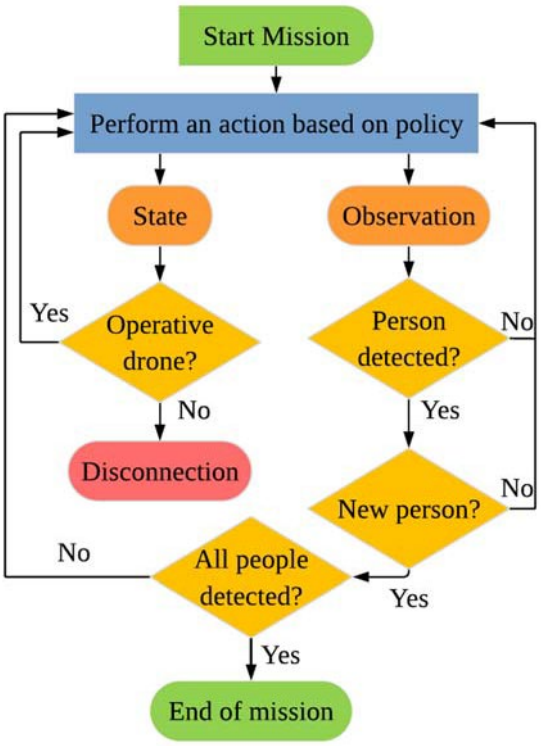

Figure 4. Flow chart for each drone in the team.

These factors determine the actions taken by each drone individually, as well as the global performance of the team.

3.2.2 Description of the Domain

The simulator reproduces the drone cage facility located at Kostas Research Institute (KRI), in Burlington, MA. The dimensions of the cage are 60m x 45m x 15m, as shown in Figure 5. Given that the drones are requested to fly at different but constant altitudes, with enough clearance, a 2-D representation of the scene satisfies a realistic approximation, since an overlap in the simulation does not mean a collision. A team of explorer drones equipped with Intel RealSense cameras R200 and a group of people are represented in the scene.

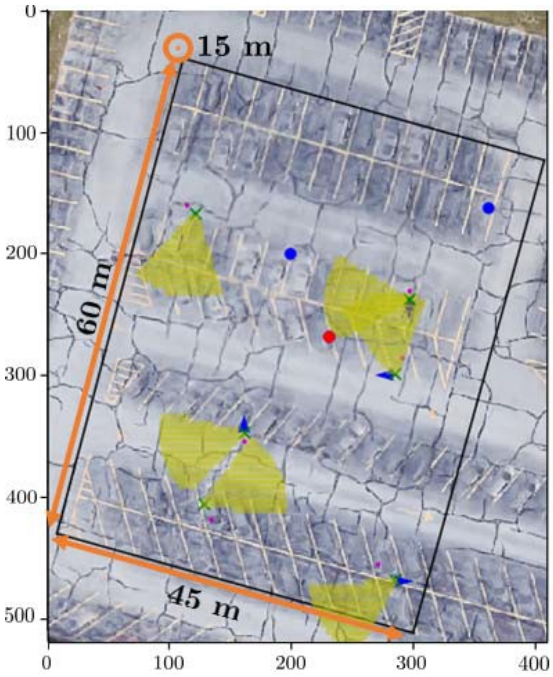

Figure 5. Top view of the drone cage at Kostas Research Institute, where the simulated drones and people are represented.

3.2.3 Space of States, Observations, and Actions

States:

As shown in Figure 6, the state of a drone is represented by several elements: The shape and color illustrate the mode of flying: a green cross represents a flying drone, meanwhile a black square represents a non-operative drone. A yellow circular sector provides the field of view of the camera of the drone, modeled as explained in the Observations part. A blue arrow depicts the direction of movement and speed of the drone. Since the drone has the ability of moving in any direction, the orientation and direction do not need to be the same. Finally, the drones are equipped with a GPS, so its current position is always known.

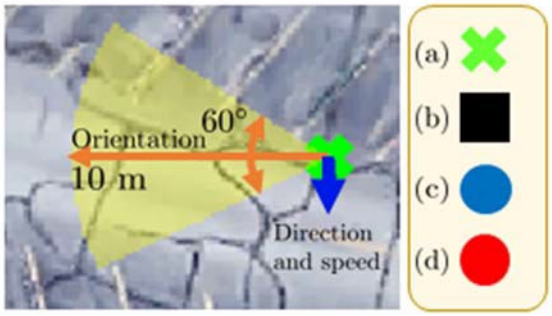

Figure 6. Left: Current state of a drone in the simulator. Right: Legend of symbols, (a) Flying drone, (b) Non-operative drone, (c) Non-detected person, and (d) Detected person.

Observations:

The explorer drones perform a continuous observation of the space trying to identify and locate a given number of people in the scene. Each frame collected by the camera is analyzed in real time by the highly efficient convolutional neural network (CNN) MobileNets [58] to distinguish people among other possible targets, enclosing them into bounding boxes. The horizontal field of view of the camera, as described in the documentation, is 60° [59], and the range of detection of the camera is estimated to be $10m$, based on field experiments. The RealSense cameras are also equipped with depth information, which provide the range from the drone to the elements detected on the field of view, as shown in Figure 7. In order to determine the distance of the person from the drone, the average of the depth values corresponding to the area of the bounding box, discarding the lower and upper 20% percentiles, is computed. The combination of the depth information, together with the GPS location of the drone, allows to determine the position of the detected person. The mission is accomplished when the total number of people is detected; but it will fail when all drones crash against the boundaries or when they run out of battery, whose life is estimated to be 15 min (900 s).

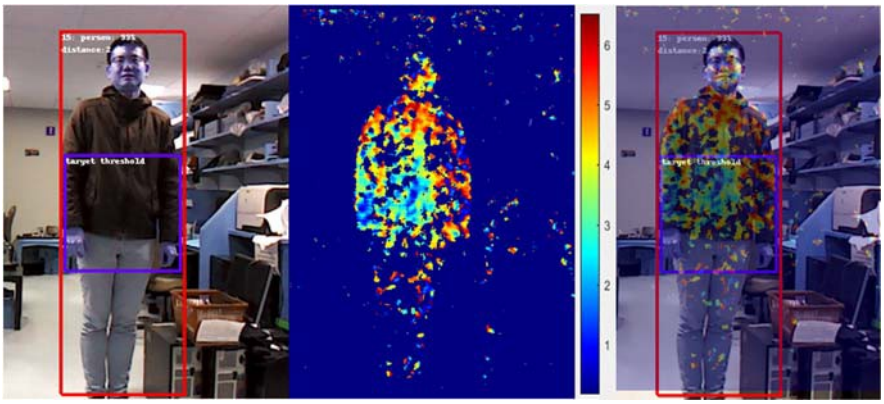

Figure 7. Inference for range estimation. Left: Bounding box of person detected from the RealSense camera. Center: Raw depth information, from 0 to 6.55m. (The pixels interpreted farther than the maximum distance are set to 0). Right: Image combination of the raw depth information with the bounding box detection.

Actions:

There are a total of six basic actions to define the possible behavior of the drones, organized in two types: Direction updates, based on the NED commands (North, East, Down). The combination of the N and E determine the direction of the drone. Since they are set to fly at a constant altitude, the D command is kept constant. The four basic actions of this type are the following: move North, East, South, and West, all at $1m/s$. Orientation updates based on the yaw command. The two basic yaw command actions are rotate 30° clockwise and counter-clockwise. Each operating drone is able to perform, at any state, any of these basic actions.

3.2.4 Modeling of uncertainties

A flying drone may be subjected to an enormous amount of uncertainties. In order to perform a realistic simulator, those have to be taken into account. Figure 8 represents a drone with all the uncertainties considered in the simulator. These uncertainties can be categorized into two main groups: the ones related to the states, and the ones related to the observations.

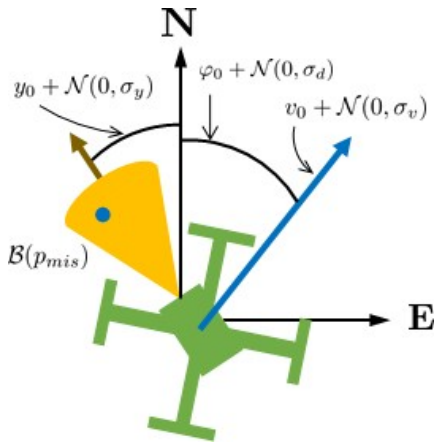

Figure 8. Representation of the uncertainties affecting the flight of the drones.

State uncertainties:

The position, direction, velocity, and orientation of a drone are subject to external perturbations, such as wind, that disturb their desired values. These perturbations will modify the expected behavior of the basic actions requested to the drones, in term of the NED and ya commands. As explained in Figure 8, the actual values of the direction φ_0, velocity v_0, and yaw y_0 , will be the results of adding a perturbation to the desired values. These perturbations are modeled by normal distributions with 0 mean and standard deviations σ_d , σ_v , σ_y, respectively. Since the position of a drone gets determined by its direction and velocity from a previous state, the position uncertainty gets embedded into the ones of the direction and velocity.

Observation uncertainties:

When a person is in the field of view of the onboard camera, there may be a misdetection, not identifying the person in the scene. This false negative is modeled as a Bernoulli random variable with probability p_{mis}.

Since the MobileNets neural network is well trained to identify people, this probability should be very small; however, it will be highly influenced by lighting conditions and partial occlusions.

3.2.5 Problem Formulation

In this section we formalize the multi-target search and detection problem using the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) model [60].

In Dec-POMDP problems, multiple agents operate under uncertainty based on partial views of the world. At each step, every agent chooses an action (in parallel) based on locally observable information, resulting in each agent obtaining an observation and the team obtaining a joint reward.

Formally, the Dec-POMDP model [60] is defined by a tuple $\langle\ , S, \{A_i\}, T, R, \{\Omega_i\}, O, h, \gamma\rangle$, where I is a finite set of agents, S is a finite set of states, A_i is a finite set of actions for each agent i with $A = \times_i A_i$ the set of joint actions, $T: S \times A \times S \rightarrow [0,1]$ is a state transition probability function, that specifies the probability of transitioning from state $s \in S$ to $s^r \in S$ when the actions $\vec{a} \in A$ are taken by the agents, $R: S \times A \rightarrow \mathbb{R}^{|I|}$ is an individual reward function, that defines the agents' rewards for being in state $s \in S$ and taking the actions $\vec{a} \in A$, Ω_i is a finite set of observations for each agent i, with $\Omega = \times_i \Omega_i$ the set of joint observations, $O: \Omega \times A \times S \rightarrow [0,1]$ is an observation probability function, that specifies the probability of seeing observations $\vec{o} \in \Omega$ given actions $\vec{a} \in A$ were taken which results in state $s^r \in S$, h is the number of steps until termination (the horizon), and $\gamma \in [0, 1]$ is the discount factor.

We extended the original Dec-POMDP model by having an individual reward function for each agent, in addition to the global shared reward. This allows the drones to learn the two objectives inherent in the given task: (1) Detect the targets in the shortest time possible, which requires coordination between the drones, and (2) learn to fly within the area boundaries, which is a task that should be learned and thus rewarded by each drone individually. In practice, we combined the shared reward and the individual rewards into a single reward function, that provides the sum of these two rewards for each agent.

A solution to a Dec-POMDP is a joint policy π --- a set of policies, one for each agent. Because one policy is generated for each agent and these policies depend only on local observations, they operate in a decentralized manner. The value of the joint policy from state s is

$$V^{rr} \quad \mathbb{E} \sum_{=0}^{h-1} \gamma\, R\ \vec{a}, s\ \big| s, \pi$$

which represents the expected discounted sum of rewards for the set of agents, given the policy's actions.

An optimal policy beginning at state s is $\pi^* s = \text{argmax}_{rr} V^{rr}$. That is, the optimal joint pol- icy is the set of local policies for each agent that provides the highest value.

3.2.6 Multi-Target Search and Detection

In this work, we address the problem of multi-target search and detection by a team of drones. The objective of the drones is to locate and detect the target objects in the minimum time possible, while keeping flying inside the area boundaries. The observations and actions available for each drone are detailed in Section 3.2.3.

The team gets a high reward (900) for detecting a target, while each drone pays a small cost of -0.1 for every action taken (to encourage efficient exploration) and receives a high penalty (-500) for bumping into the area boundaries.

All the drones start flying from the same region; however, the positions of the targets may change in each episode. In this work, we assume that there is no explicit communication between the drones, and that they cannot observe each other. Since the positions of the targets are unknown a-

prior to the drones, the drones need to find a general strategy for efficiently exploring the environment. Moreover, they need to learn to coordinate their actions, in order not to repeatedly cover areas that have already been explored by other drones.

3.2.7 Decentralized Advantage Actor-Critic (DA2C)

Due to partial observability and local non-stationarity, model-based Dec-POMDP is extremely challenging, and solving for the optimal policy is NEXP-complete [60]. Our approach is model-free and decentralized, learning a policy for each agent independently. Specifically, we extend the Advantage Actor-Critic (A2C) algorithm [49] for the multi-agent case. Our proposed method Decentralized Advantage Actor-Critic (DA2C) is presented in Algorithms 1 and 2.

A2C is a policy gradient method, that targets at modeling and optimizing the policy directly. The policy is modeled with a parameterized function with respect to θ, $\pi_0\ a|$. The objective value of the reward function depends on this policy and can be defined as: $J\ \theta = \sum_{s\in S} d^{rr}\ s\ V^{rr}$, where $d^{rr}\ s$ is the stationary distribution of states.

According to the policy gradient theorem [61]

$$\nabla_0 J\ \theta = \mathbb{E}_{s,a\sim rr}[Q^{rr}\ s, a\ \nabla_0 \log\backslash pi_0\ a|s\ ]$$

A main limitation of policy gradient methods is that they can have high variance [62]. The standard way to reduce the variance of the gradient estimates is to use a baseline function $b\ s$ inside the expectation:

$$\nabla_0 J\ \theta = \mathbb{E}_{s,a\sim rr}[\ Q^{rr}\ s, a - \ b\ s\ \nabla_0 \log\backslash pi_0\ a|s\ ]$$

A natural choice for the baseline is a learned state-value function $b\ s = V^{rr}$, which reduces the variance without introducing bias. When an approximate value function is used as the baseline, the quantity $A\ s, a = Q\ s, a - \ V\ s$ is called the advantage function. The advantage function indicates the relative quality of an action compared to other available actions computed from the baseline.

In actor-critic methods [62], the actor represents the policy, i.e., action-selection mechanism, whereas a critic is used for the value function learning. The critic follows the standard temporal difference (TD) learning [61], and the actor is updated following the gradient of the policy's performance.

Thus, the loss function for A2C is composed of two terms: policy loss (actor), $\mathcal{L}_{rr}$, and value loss (critic), $\mathcal{L}_v$. An entropy loss for the policy, H , is also commonly added, which helps to improve exploration by discouraging premature convergence to suboptimal deterministic policies. Thus, the loss function is given by:

$$\mathcal{L} = \lambda_{rr}\mathcal{L}_{rr} + \lambda_v\mathcal{L}_v - \ \lambda_H\mathbb{E}_{s\sim rr}[H(\pi \cdot |s\ )]$$

with λ_{rr}, λ_v, λ_H being weighting terms on the individual loss components.

The architecture of our decentralized actor-critic algorithm is depicted in **Figure 9**. As described in Algorithm 1, our training process alternates between sampling trajectories by the team of agents (lines 7--14) and optimizing the networks of the agents with the sampled data (lines 17--23).

In the procedure TrainAgent described in Algorithm 2, we accumulate gradients over the mini-batch of samples, and then use them to update the actor and critic networks' parameters. Accumulating updates over several steps provides some ability to trade off computational efficiency for data efficiency.

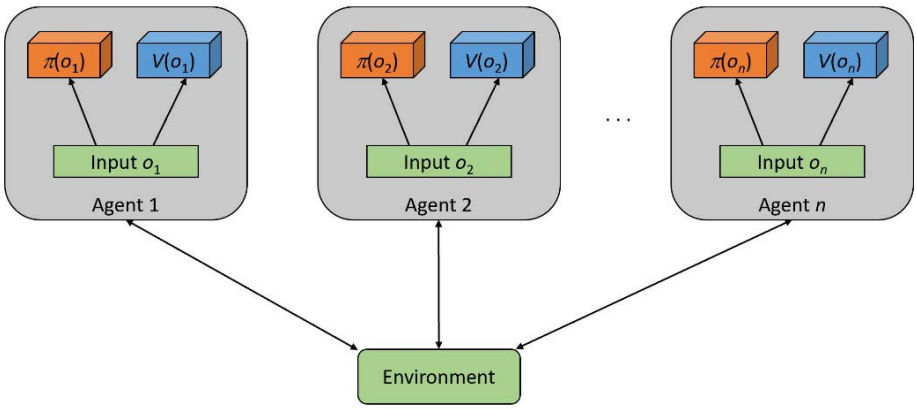

Figure 9. Overview of our multi-agent decentralized actor, critic approach.

Algorithm 1 DA2C (I, T_{max}, m)

Input: A group of agents I, maximum episode length T_{max}, and batch size m

for $i \in I$ **do**
  Initialize actor and critic network weights θ_i, ω_i
Get an initial joint observation $\vec{o}$
repeat
  $t \leftarrow 1$
  Initialize buffer $\mathcal{B}$
  // Collect m samples from the environment
  for $j \leftarrow 1, ..., m$ **do**
    for $i \in I$ **do**
      Sample action $a_i \sim \pi_{\theta_i}(\cdot|o_i)$
    Execute the joint action $\vec{a} = (a_1, ..., a_n)$
    Receive a new joint observation $\vec{o'}$ and reward $\vec{r}$
    Store transition $(\vec{o}, \vec{a}, \vec{r}, \vec{o'})$ in $\mathcal{B}$
    $t \leftarrow t + 1$
  // Train the agents using the samples in the buffer
  for $i \in I$ **do**
    Initialize O_i, A_i, R_i, O'_i to empty sets
    for each transition $(\vec{o}, \vec{a}, \vec{r}, \vec{o'}) \in \mathcal{B}$ **do**
      $O_i \leftarrow O_i \cup \{\vec{o}_i\}$
      $A_i \leftarrow A_i \cup \{\vec{a}_i\}$
      $R_i \leftarrow R_i \cup \{\vec{r}_i\}$
      $O'_i \leftarrow O'_i \cup \{\vec{o'}_i\}$
    TRAINAGENT(O_i, A_i, R_i, O'_i)
  $\vec{o} \leftarrow \vec{o'}$
until $t > T_{max}$ or mission accomplished

Algorithm 1 DA2C

Algorithm 2 TRAINAGENT(O, A, R, O', i, m)

Input: A sequence of observations O, actions A, rewards R, and next observations O', agent index i, and batch size m

// Initialize the variable that holds the return estimation

$$G \leftarrow \begin{cases} 0 & \text{if } s_m \text{ is a terminal state} \\ V_{\omega_i}(o'_m) & \text{otherwise} \end{cases}$$

for $j \leftarrow m-1, ..., 1$ **do**

  $G \leftarrow \gamma G + r_j$

  Accumulate gradients w.r.t. θ_i:

  $d\theta_i \leftarrow d\theta_i + \nabla_{\theta_i} \log \pi_{\theta_i}(a_j|o_j)(G - V_{\omega_i}(o_j))$

  Accumulate gradients w.r.t. ω_i:

  $d\omega_i \leftarrow d\omega_i + 2(G - V_{\omega_i}(o_j))\nabla_{\omega_i}(G - V_{\omega_i}(o_j))$

Update θ_i using $d\theta_i$, and ω_i using $d\omega_i$

Algorithm 2 Training Agent

3.2.8 Network Architecture

Each drone has two neural networks: one for the actor and one for the critic. Both networks consist of three fully connected layers with ReLU nonlinearities. The first layer has 200 neurons and the second one has 100 neurons. The output of the actor network is a probability distribution over the actions, thus its output layer has six neurons (one for each possible action), whereas the critic network returns a single number, which represents the approximate state value.

3.3 SAR imaging experiment

The UAV system has an optical camera that is connected to an on-board computer. The downward-looking mm-wave radar module is mounted on the front panel of the UVA, which has an operating bandwidth of 76-81 GHz. The radar applies time division multiplexing (TDM) using 3 transmitters (Txs) and 4 receivers (Rxs), where each frame consists of 3 chirps with each chirp corresponding to the transmission of one TX (Figure 10 left). The SAR imaging experiment is carried out in an indoor environment (Figure 10), where the UAV is ~1.2 m from the ground. The region of interest (RoI) has a size of 0.7 m, 0.24 m, and 0.5 m along x-, y-, and z-axis, respectively. A metallic box and a metallic bar, which are separated by 18 cm, are in located in the RoI. A flying time is 3.5 seconds, covering 0.7 m along x-axis. The total number of transmitted frames is 176.

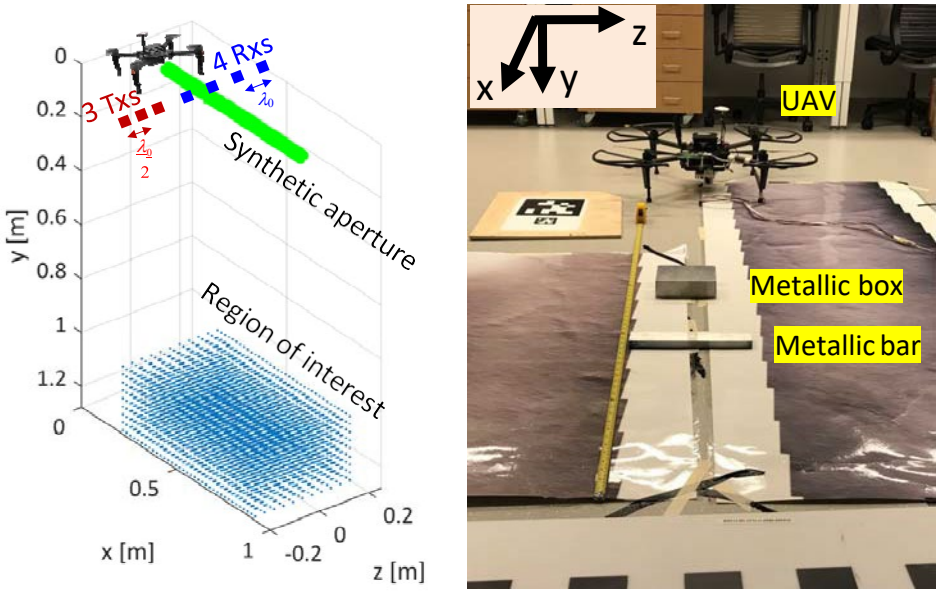

Figure 10 SAR imaging experiment setup to detect a metallic box and bar. Both mm-wave and optical sensors are integrated in the UAV.

The following norm-1 regularized compressive sensing and imaging algorithm, based on a distributed alternating direction method of multipliers (ADMM) [63], is used:

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{H}_i\mathbf{u}_i-\mathbf{g}_i\right\|_2^2+\lambda_r\|\mathbf{v}\|_1$$

$$\text{s.t.} \qquad \mathbf{u}_i=\mathbf{v},\forall i=1,\cdots,N,$$

where the sensing matrix $\mathbf{H}$ and the measurement vector $\mathbf{g}$ are divided into N submatrices $\mathbf{H}_i$ and N subvectors $\mathbf{g}_i$, respectively, $i \in [1, N]$; $\mathbf{u}_i$ is the reflectivity vector for each pair of $\mathbf{H}_i$ and $\mathbf{g}_i$; λ_r is the norm-1 weight factor; and $\mathbf{v}$ is a *consensus* variable that enforces the agreement among all $\mathbf{u}_i$.

The sensing matrix $\mathbf{H}$ can be simply computed using first-order Born approximation with its ($f_{n,m,p}$, k)-th entry calculated as follows:

$$H\left(f_{n,m,p},k\right)=E^{\mathrm{Tx}}E^{\mathrm{Rx}}$$

$$=\frac{e^{-jk_{0n,p,k}\left|\mathbf{r}^{\mathrm{Tx}}_{n,p}-\mathbf{r}^{\mathrm{RoI}}_{k}\right|}}{\left|\mathbf{r}^{\mathrm{Tx}}_{n,p}-\mathbf{r}^{\mathrm{RoI}}_{k}\right|}\frac{e^{-jk_{0m,p,k}\left|\mathbf{r}^{\mathrm{Rx}}_{m,p}-\mathbf{r}^{\mathrm{RoI}}_{k}\right|}}{\left|\mathbf{r}^{\mathrm{Rx}}_{m,p}-\mathbf{r}^{\mathrm{RoI}}_{k}\right|}$$

where n, m, p, and k are the index of the Txs, Rxs, positions in the synthetic aperture, and unknown pixels in the RoI, respectively; k_0 is the wave number in the free space; j is the imaginary unit; and r is the position vector.

4.0 RESULTS AND DISCUSSION

4.1 Results of the Team of Drones

At the top of the system is the mission controller, which controls the subsystems to achieve the swarm's objective. Mission objectives for the swarm of drones is typically defined by an area of exploration and a searching objective, e.g., find survivors in a disaster-struck area. To maximize the ability to search an area and understand the environment, the swarm needs to be divided into a specific number of subswarms depending on the environment and objective. For example, the area of exploration can be partitioned into different sections, each of which is searched by a different subswarm. If needed, subswarms can decide to split up into smaller subswarms depending on what is best for the environment it is in. For example, a subswarm may encounter a building or multiple building, and need to split up to search these newly encountered parts of the environment. On the smallest scale in this system, individual drones make observations and act on them based on a learned policy. Communication with other drones in the subswarm occurs depending on its observations. A mission can be ended when the mission level system sends a signal, either based on time or observations, that the mission is over.

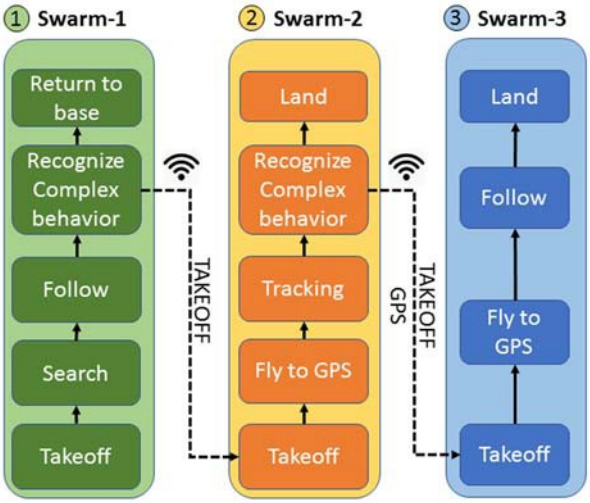

Figure 11. Multi-Swarm mission description

This work brings up the proof of concept of experimenting with drones in navigating, tracking, following, and landing modes with a swarm of ten drones, as represented in Figure 11. In this experiment, swarm-one, swarm-two and swarm-three have three, four, and three drones, respectively, who are participating in the mission, as it is represented in detail in Figure 12. The tasks for swarm-one are detecting, following the person until he/she is entering into the car (which has been defined as a complex behavior), and finally return to base. The tasks for swarm-two are flying to the predefined GPS locations, and start tracking, which means facing to a direction where the car moves. Until the car stopped and the person went out of the car, which also has been defined as a complex behavior, or received the LAND command from both peers or ground control station, the drones keep on tracking. In this stage, if any of the drone detects the complex behavior, i.e., a person and a car is present, it immediately sends ARM command to the swarm-three to fly around the sending drone. The tasks for swarm-three are flying to the GPS position sent from the drone

and start following the person. The swarm-three looks the person and maintain a constant distance with the person before coming back to the base station. The detailed sequence of the mission, along with some frames captured by the cameras of the drones incorporating their perception, is shown in Figure 13.

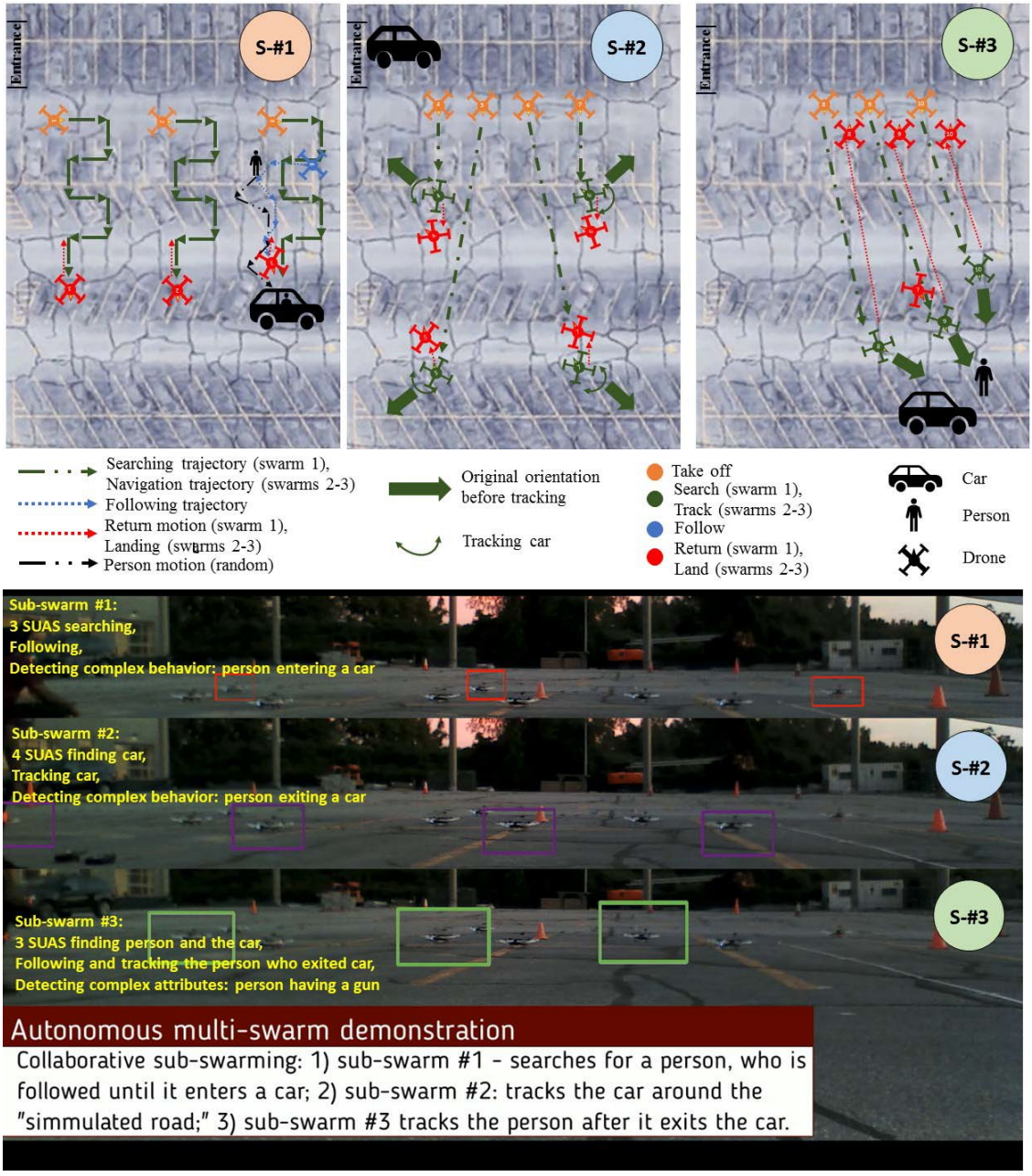

Figure 12. Autonomous Multi-swarm demonstration. 1) Sub-swarm #1 searches for a person, who is fol- lowed until it enters a car; 2) Sub-swarm #2 tracks the car around the "simulated road"; 3) Sub-swarm #3 tracks the person after it exits the car.

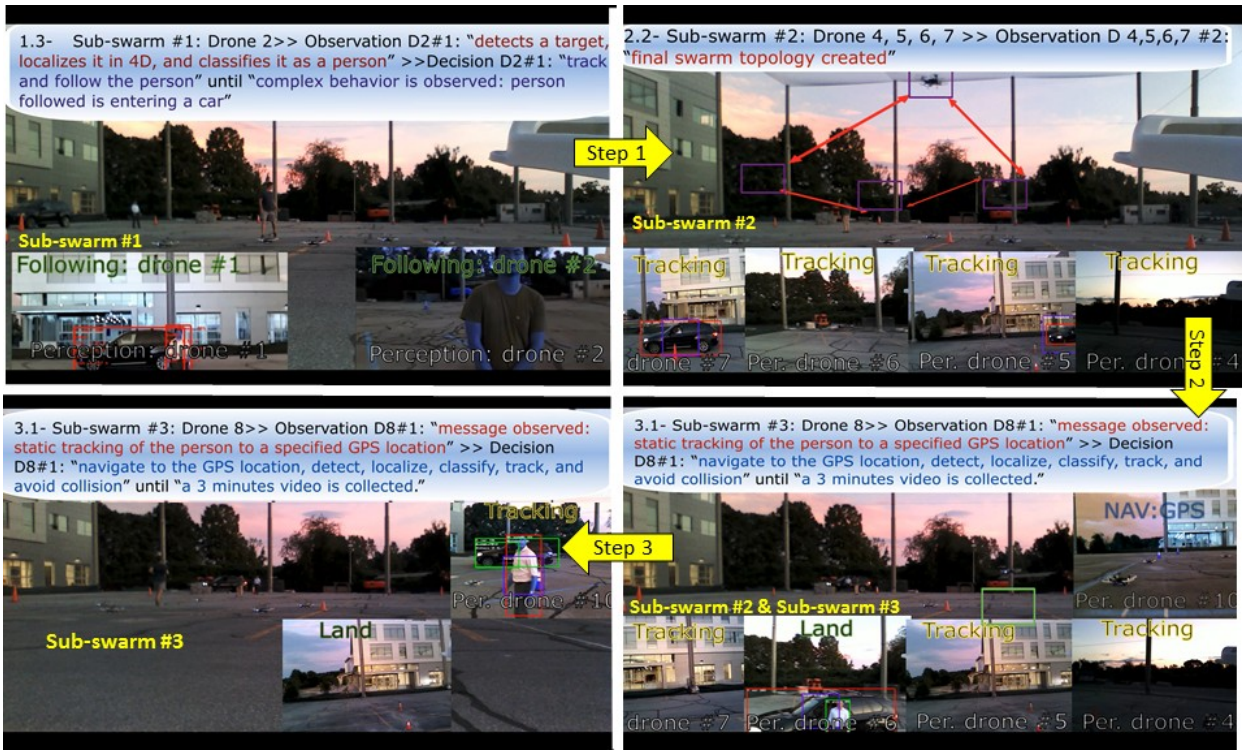

Figure 13. Overall sequence of the whole mission recorded by a stationary ground camera and a drone camera.

The perception in swarms one and two, which leads to the detection and tracking of the person and car, is performed by a computer vision algorithm based on the pre-trained MobileNet-SSD convolutional neural network [64]. The front RGB camera captures the target within a rectangle. When the midpoint of that rectangle appears deviated from the center of the camera, the flying algorithm promotes the drone to rotate until the target center point meets within the threshold of the tracking pattern. Meanwhile, the depth camera measures the average distance to the target. When the distance increases over a given value D_0, the flying algorithm pushes the drone closer to the target and vice-versa.

On the other hand, the perception in swarm three, which leads to the following at a constant distance of the person, is performed by extracting the range distance from the radar 3D point-cloud followed by a negative feedback to the UAV fight controller and moving -(R-R_0) meters in the range direction (the direction of the front-view of the camera), where R_0 is the constant following distance and R is the detected range distance by the radar.

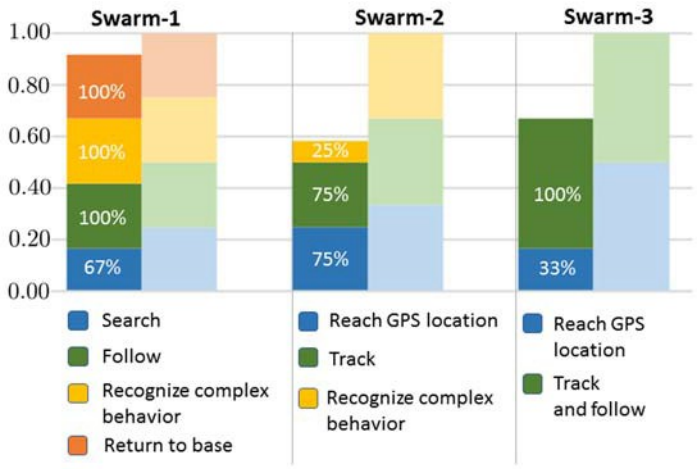

Figure 14. Multi-swarm performance. For each swarm, the right bar is desired performance for each task, while the left bar is the actual performance.

Figure 14 shows the performance of the current experiment. For swarm-one, 67% of drones (two out of three) found and detected the person successfully, made the decision to follow them, and finished the following task successfully; it neither lost the target during following nor hit the obstacle accidentally. The other drone failed to finish its tasks successfully without finding the person; however, at the end, all the drones in swarm-one received the LAND command and returned to base successfully. In the swarm-two, 75% of drones (three out of four drones) reached their predefined GPS points, rotated around their own z-axis and tracked the car as excepted, and one of these three drones finished the complex task, which was defined as detecting the target person entering into the car, and then sending messages to swarm-three to arm and take off. Finally, in the swarm-three, 33% of drones (one out of three drones) received the TAKEOFF message from swarm-two successfully and done the mission to fly over the GPS point, tracking as well as following tasks successfully. However, the other two did not take off as being supposed to.

4.2 Discussion of the drone swarming

While testing, if the drone keeps navigating, tracking, following, and landing, then the tasks are considered as successful, and they are defined as performed the expected mission. It is observed that tracking in negative areas---such as the dark side of the car---, communication antenna orientation, wind speed, sensor calibration, distance between drones resulting packet loss affect the detection, navigating, and tracking performance for the swarms to perform a desired task.

In addition, when multiple targets are captured by the camera of the drone, such as several people or cars in the same frame, some constraints may limit the drone operation, leading to a possible false tracking. In the presented case, the person or car that first appears in the drone's field of view is considered as the main target, tracking it without losing or switching it. However, if two people appear on the scene too close or lap over each other, it is possible that the tracker switches the main target, leading to a failed mission. In future experiments, where the requested mission will be much more complex than current experiment, trying to simulate a scenario in the real world, it will be crucial for the team of SUAS to obtain as much as information as possible from the outside environment. For these cases, a multiple object tracking (MOT) approach is expected to be more

reliable in realistic scenarios. This functionality, which will be vital for a team of SUAS to perceive a large-scale environment, is already available with current online MOT methods, such as deepSORT, MHT_bLSTM, and OneShotDA, benefiting the extensibility of our approach to more complex missions.

Moreover, the current mm-wave radar is employed using time-division multiplexing where only one Tx is transmitting at a time, resulting in a possible low signal-to-noise ratio (SNR) at the receiver end and causing a poor detection accuracy if the object is too far away. Future radar architecture will use spatial multiplexing schemes such as binary-phase-modulation to perform the detection, where all the Txs are transmitting simultaneously to achieve a much higher SNR at the receiver end.

4.3 Reinforcement Learning results

Our first experiment involves an environment with three drones and three targets, where all the drones start flying from the bottom left corner of the area. The parameters of Algorithm 1 and the training process are shown in Table 1.

Table 1. Parameters of DA2C

PARAMETERS OF DA2C

Discount factor	γ	0.99
Learning rate	η	0.0001
Mini-batch size	m	32
Policy loss weight	λ_π	1
Value loss weight	λ_v	1
Entropy loss weight	λ_H	0.001
Maximum episode length	T_{max}	900
Drone's direction std	σ_d	0.1
Drone's orientation std	σ_y	0.1
Drone's speed std	σ_v	0.1
Misdetection probability	p_{mis}	0.05

Figure 15 shows the average reward $\bar{r}$ and standard deviation per episode for 500 training episodes. The average is computed over five independent runs with different random seeds. Each training session took approximately 5 hours to complete on a single Nvidia GPU GeForce GTX 1060.

The maximum possible reward that can be attained in this scenario is $900 \cdot 3 - 0.1 \cdot 3\, n = 2700 - 0.3n$, where n is the number of time steps it takes for the drones to detect all the targets. Since the maximum length of an episode is 900 time steps, the maximum possible reward lies in the range [2430, 2700], depending on the initial locations of the targets. As can be seen in the graph, after a relatively small number of episodes (about 400 episodes), the team was able to reach

an average reward very close to the maximum (2648). The fluctuations in the graph can be attributed to the fact that some of the initial configurations of the targets are significantly harder to solve than others (e.g., when the targets are located in different corners of the environment).

By examining the learned policies of the drones, we can see that the work area is first split between the drones, and then each drone thoroughly explores its own subarea by simultaneously moving and rotating the camera for maximum coverage efficiency.

Next, we compared the performance of our learned joint policy against two baselines. In the first baseline, the drones choose their actions completely randomly. The second baseline is a collision-free policy, where the drones fly randomly most of the time, but change their direction by 180 degrees when they get near the walls. Note that this baseline has an edge over our learned policy, as our drones had to learn not to collide with the walls.

All three policies (the learned one and the two baselines) have been evaluated on 500 episodes with different initial locations of the targets. **Figure 16** shows the results. As can be seen, our learned policy significantly outperforms the two baselines, achieving a mean total reward of 1388.36, while the total mean reward achieved by the random policy and the collision-free policy are -1314.72 and -247.56, respectively.

We have also examined the impact of changing the number of drones in the team on the team's ability to fulfill the task. **Figure 17** shows the average reward achieved by different team sizes, ranging from two drones to six drones. The number of targets remained three in all experiments. Clearly, adding more drones to the team increases the probability of detecting all targets within the time limit. However, increasing the team size for more than five drones does not improve the performance any further, which implies that the team has reached a near-optimal solution (a team with five drones was able to achieve an average reward of 1827 over 500 evaluation runs).

Lastly, we have examined the ability of the drones to detect different numbers of targets. **Figure 18** shows the average reward achieved by a team of three drones, trying to detect between two to six targets. We can observe an almost linear relationship between the number of targets and the average return, which means that the time required to find any additional target is nearly constant.

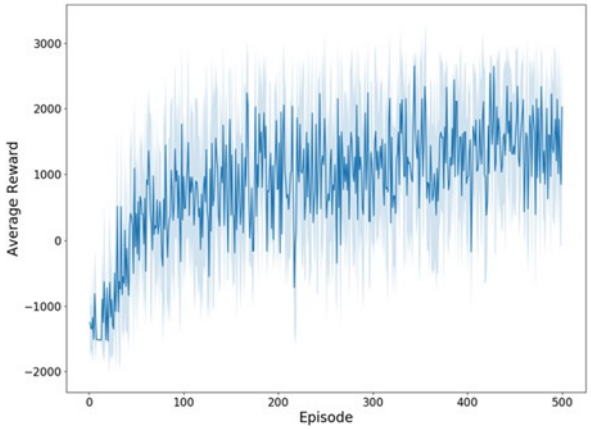

Figure 15. Average reward and standard deviation per episode in an environment with three drones and three targets.

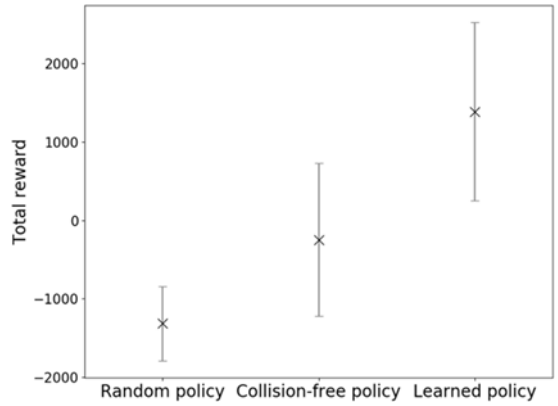

Figure 16. The total reward and standard deviation achieved by our learned policy vs. a random policy and a collision-free policy, averaged over 500 episodes.

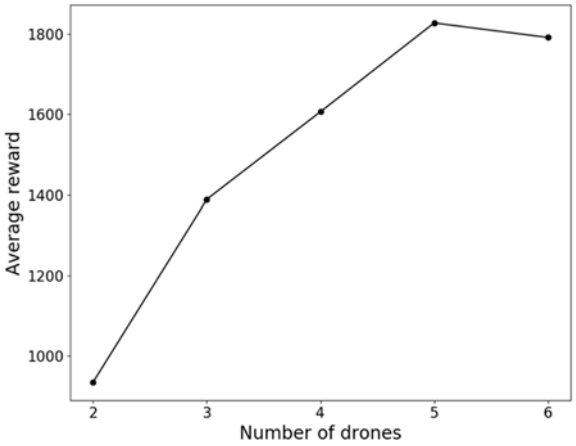

Figure 17. The average reward achieved by different team sizes, ranging from two drones to six drones.

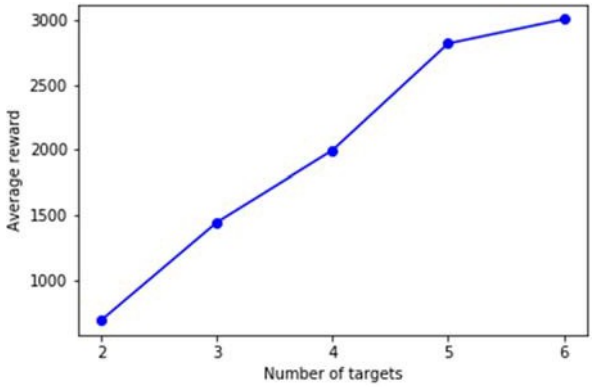

Figure 18. The average reward achieved by a team of 3 drones for various number of targets.

4.4 SAR imaging results

Figure 19a shows the co-registered image where the mm-wave reconstructed reflectivity is fused with the real photo captured by the optical camera, while Figure 19b shows the co-registered depth information in the same photo. Note that a display threshold of 0.2 and a 3-pixel 2D averaging [65] are applied in Figure 19. The white dashed bounding boxes are the ground truth plots of the metallic box and bar, respectively. As it is seen, the high reflectivity of metallic objects is successfully recovered. The reconstruction error is mainly attributed to the unavoidable turbulence of the UAV during its flying, making motion compensation techniques suitable to be considered in the future.

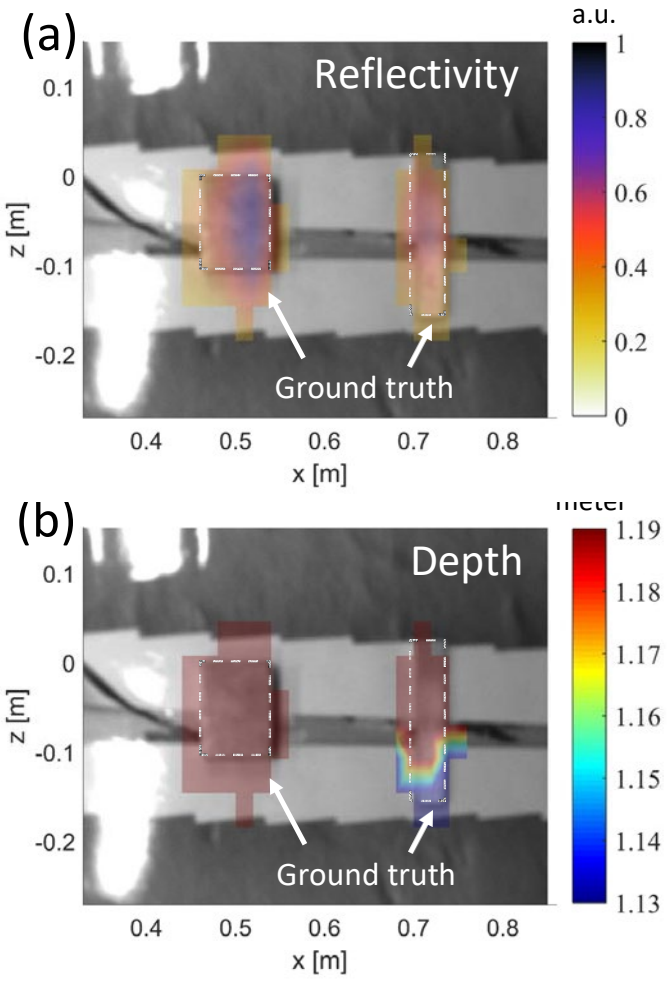

Figure 19 (a) and (b) are the real optical image co-registered with mm-wave reconstructed reflectivity and depth information, respectively.

5.0 CONCLUSIONS

The first work has shown an experimental test of a context-aware distributed team of SUAS coordinately working on a multi-step complex mission, capable of operating in real-time, in an autonomous fashion, and under constrained communications. In this experiment, 10 drones divided into three teams perform the complex three-step task of (i) searching, detecting, and following a person until enters a car, (ii) navigating to a specific GPS position and tracking a car until a person leaves the car, and (iii) navigating to a GPS position given by the previous team, and follow a person at a constant distance for a period of time. The proposed framework relies on a three layers approach: operational, tactical, and strategical, corresponding to single agent actions, group of agents collaboration, and teams of multiple groups of agents join cooperation, respectively. The complex mission is carried out based on the continuous loop perception--policy--decision architecture. The perception is done based on the fusion of 4D RGB and 4D infrared cameras, together with 4D mm-wave radar, and the communication among the agents in the teams is performed by and ad-hoc network. The experimental validation showed that the complex task was propitiously achieved by the cooperation of the three teams. Although some agents in the teams may have not had the expected behavior due to possible packages loss, non-optimal illumination conditions for detection and tracking, and navigation issues due to the uncertainties, the global behavior of the swarm managed to successfully complete the required mission.

In terms of the Reinforcement Learning problem applied to the designed multi-drone simulator, we have proposed a fully decentralized multi-agent policy gradient algorithm to solve a challenging real-world problem of multi-target search and detection. Our method is able to find a near-optimal solution to the problem using a short training time. Despite being completely decentralized, our drones learn to coordinate their actions as to minimize the overlap between the areas they are exploring. In the future we would like to consider dynamic environments, in which the targets may change their locations, as well as adding more sensors to the drones, and testing the results on real drones.

Finally, the experimental results on the SAR imaging using our multi-modal UAV system have been verified. Specifically, the fusion of the mm-wave and optical images enabled accurate detection and ranging of metallic objects in the scene. This fused capability enhances the overall performance the UAV inspection systems.

6.0 REFERENCES

[1] A. A., G. M., P. A., S. K. and J. P., Plan-based Object Search and Exploration Using Semantic Spatial Knowledge in the Real World, Proceedings of the 5th European Conference on Mobile Robots (ECMR), 2011.

[2] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[3] R. Girshick, "Fast {R-CNN}," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[5] J. Fu, S. Levine and P. Abbeel, "One-shot learning of manipulation skills with online dynamics adaptation and neural network priors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous control with deep reinforcement learning," in *arXiv preprint arXiv:1509.02971*, 2015.

[7] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramar, R. Hadsell and N. de Freitas, "Reinforcement and imitation learning for diverse visuomotor skills," in *arXiv preprint arXiv:1802.09564*, 2018.

[8] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *nature,* vol. 521, p. 436, 2015.

[9] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011.

[10] X. Yang, P. Ramesh, R. Chitta, S. Madhavanath, E. A. Bernal and J. Luo, "Deep multimodal representation learning from temporal data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012.

[12] D. E. Whitney, "Historical perspective and state of the art in robot force control," *The International Journal of Robotics Research,* vol. 6, no. 1, pp. 3-14, 1987.

[13] X. B. Peng, M. Andrychowicz, W. Zaremba and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[14] R. Diankov and J. Kuffner, "OpenRAVE: A Planning Architecture for Autonomous Robotics," Pittsburgh, PA, 2008.

[15] K. Sjo, D. G. Lopez, C. Paul, P. Jensfelt and D. Kragic, "Object Search and Localization for an Indoor Mobile Robot," *Journal of Computing and Information Technology,* vol. 17, no. 1, pp. 67 - 80, 2009.

[16] T. de Bruin, J. Kober, K. Tuyls and R. Babuska, "Integrating state representation learning into deep reinforcement learning," *IEEE Robotics and Automation Letters,* vol. 3, no. 3, pp. 1394--1401, 2018.

[17] .. Cifuentes, J. Issac, M. Wuthrich, S. Schaal and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," *IEEE Robotics and Automation Letters,* vol. 2, no. 2, pp. 577--584, 2016.

[18] W. Zhang, J. Heredia-Juesas, M. Middiy, L. Tirado, H. Singhy and J. A. Martinez-Liorenzo, "Experimental Imaging Results of a UAV-mounted Downward-Looking mm-wave Radar," in *IEEE International Symposium on Antennas and Propagation \& USNC/URSI National Radio Science Meeting*, 2019.

[19] L. E. Tirado, W. Zhang, A. Bisulco, H. Gomez-Sousa and J. A. Martinez-Lorenzo, "Towards three-dimensional millimeter-wave radar imaging of on-the-move targets," in *2018 IEEE International Symposium on Antennas and Propagation \& USNC/URSI National Radio Science Meeting*, 2018.

[20] W. Zhang and J. A. Martinez-Lorenzo, "Single-frequency material characterization using a microwave adaptive reflect-array," in *2018 IEEE International Symposium on Antennas and Propagation \& USNC/URSI National Radio Science Meeting*, 2018.

[21] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," vol. 3, pp. 2149--2154, 2004.

[22] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[23] J. a. S. C. a. S. A. Sinapov, "Learning relational object categories using behavioral exploration and multimodal perception," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[24] J. Kensler and A. Agah, "Neural networks-based adaptive bidding with the contract net protocol in multi-robot systems," *Applied Intelligence,* vol. 31, no. 3, p. 347, 2009.

[25] A. Liekna, E. Lavendelis and A. Grabovsk, "Experimental analysis of contract net protocol in multi-robot task allocation," *Applied Computer Systems,* vol. 13, no. 1, pp. 6--14, 2012.

[26] O. Shehory and S. Kraus, "Task allocation via coalition formation among autonomous agents," in *IJCAI (1)*, 1995.

[27] C. Li and K. Sycara, A stable and efficient scheme for task allocation via agent coalition formation, World Scientific, 2004, pp. 193--212.

[28] O. Shehory and S. Kraus, "Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents," in *Proc. of ICMAS-96*, 1996.

[29] L. P. Kaelbling, M. L. Littman and A. R. Cassandra, "Artificial Intelligence," *Planning and Acting in Partially Observable Stochastic Domains,* vol. 101, no. 1-2, pp. 99--134, 1998.

[30] S. Katt, F. A. Oliehoek and C. Amato, "Learning in {POMDPs} with {Monte Carlo} tree search}," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017.

[31] S. Ross, J. Pineau, B. Chaib-draa and P. Kreitmann, "A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes.," *Journal of Machine Learning Research,* vol. 12, no. 5, 2011.

[32] L. K. Li, D. Hsu and W. S. Lee, "Act to See and See to Act: {POMDP} planning for objects search in clutter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[33] M. L. Littman, A. R. Cassandra and L. P. Kaelbling, "Learning Policies for Partially Observable Environments: Scaling Up," in *International Conference on Machine Learning (ICML)*, Morgan Kaufmann, 1995.

[34] S. Ross, J. Pineau, S. Paquet and B. Chaib-draa, "Online Planning Algorithms for POMDPs," *J. Artif. Int. Res.,* vol. 32, no. 1, pp. 663--704, 2008.

[35] D. a. V. J. Silver, "Monte-Carlo Planning in Large {POMDPs}," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, Vancouver, British Columbia, Canada, 2010.

[36] A. Otto, N. Agatz, J. Campbell, B. Golden and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks,* vol. 72, no. 4, pp. 411-458, 2018.

[37] J. Hu, L. Xie, J. Xu and Z. Xu, "Multi-agent cooperative target search," *Sensors,* vol. 14, no. 6, p. 9408–9428, 2014.

[38] Y. Yang, A. A. Minai and M. M. Polycarpou, "Decentralized coop-erative search by networked UAVs in an uncertain environment," in *in Proceedings of the 2004 American Control Conference*, 2004.

[39] L. F. Bertuccelli and J. How, "Robust UAV search for environments with imprecise probability maps," in *in Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. B. Bellemare, A. Graves and et al., "Human-level control through deep reinforcement learning," *Nature,* vol. 2015, no. 518, p. 529, 2015.

[41] S. Gu, E. Holly, T. Lillicrap and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *in IEEE international conference on robotics and automation (ICRA)*, 2017.

[42] C. -J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," in *arXiv preprintarXiv:1905.02680*, 2019.

[43] P. Hernandez-Leal, M. Kaisers, T. Baarslag and E. M. de Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," in *arXiv preprint arXiv:1707.09183*, 2017.

[44] L. Bu, R. Babu and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 38, no. 2, p. 156–172, 2008.

[45] A. K. Agogino and K. Tumer, "Unifying temporal and structural credit assignment problems," in *in International Conference on Autonomous Agents and Multiagent Systems*, 2004.

[46] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel and I. Mordatch, "“Multi-agent actor-critic for mixed cooperative-competitive environments," in *in Advances in Neural Information Processing Systems*, 2017.

[47] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardel and S. Whiteson, "Counterfactual multi-agent policy gradients," in *in Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[48] G. Palmer, K. Tuyls, D. Bloembergen and R. Savani, "Lenient multi-agent deep reinforcement learning," in *in International Conference on Autonomous Agents and Multi Agent Systems. International Foundation for Autonomous Agents and Multiagent Systems*, 2018.

[49] V. Mnih, A. P. Badia, M. Mirza, A. Graves,, T. Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *in International conference on machine learning*, 2016.

[50] S. Jordan and et al, "State-of-the-art technologies for uav inspections," *IET Radar, Sonar Navigation,* vol. 12, no. 2, p. 151–164, 2018.

[51] D. Takhar and et al., "A new compressive imaging camera architecture using optical-domain compression," *in Computational Imaging IV,* vol. 6065, p. 606509, 2006.

[52] G. G. Shepherd and et al., "Wamdii: wide-angle michelson doppler imaging interferometer for spacelab," *Applied optics,* vol. 24, no. 11, p. 1571–1584, 1985.

[53] D. M. Sheen, D. L. McMakin and T. E. Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Trans. Micro. Theory Tech.,* vol. 49, no. 9, p. 1581–1592, 2001.

[54] C. Li and H. Ling, "Wide-angle, ultra-wideband isar imaging of vehicles and drones," *Sensors,* vol. 18, no. 10, p. 3311, 2018.

[55] P. Huegler and et al., "Radar taking off: New capabilities for uavs," *IEEE Micro. Mag.,* vol. 19, no. 7, p. 43–53, 2018.

[56] C. J. Li and . H. Ling, "Synthetic aperture radar imaging using a small consumer drone," in *in 2015 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, 2015.

[57] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," *arXiv preprint arXiv:1810.10191,* 2018.

[58] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in *arXiv preprint arXiv:1704.04861*, 2017.

[59] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[60] D. S. Bernstein, R. Givan, N. Immerman and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of operations research,* vol. 27, no. 4, p. 819–840, 2002.

[61] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[62] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *in Advances in neural information processing systems*, 2000.

[63] J. Heredia-Juesas and et al., "Norm-1 regularized consensus-based admm for imaging with a compressive antenna," *IEEE Antennas Wireless Propag. Lett.,* vol. 16, p. 2362–2365, 2017.

[64] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.

[65] W. Zhang and et al., "Experimental results of a 3-d millimeter-wave compressive reflector antenna imaging system," *IEEE Antennas Wireless Propag. Lett.,* vol. 18, no. 1, p. 24–28, 2019.

APPENDIX A – PUBLICATIONS AND PRESENTATIONS

1. J. A. Martinez-Lorenzo, J. Hudack, Y. Jing, M. Shaham, Z. Liang, A. A. Bashit, Y. Wu, W. Zhang, M. Skopin, J. Heredia-Juesas, Y. Ma, T. Sweeney, N. Ares and A. Fox, "Preliminary Experimental Results of Context-Aware Teams of Multiple Autonomous Agents Operating under Constrained Communications," submitted to ICRA 2021.

2. R. Yehoshua, J. Heredia-Juesas, Y. Wu, C. Amato, J.A. Martinez-Lorenzo, "Decentralized Reinforcement Learning for Multi-Target Search and Detection by a Team of Drones," submitted to ICRA 2021.

3. W. Zhang, J. Heredia-Juesas, M. Diddi, L. Tirado, H. Singh and J. A. Martinez-Lorenzo, "Experimental Imaging Results of a UAV-mounted Downward-Looking mm-wave Radar," *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeing*, Atlanta, GA, USA, 2019, pp. 1639-1640, doi:10.1109/APUSNCURSINRSM.2019.8889290.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

SUAS	Small Unmanned Aerial Systems
MOT	Multiple Objects Tracking
CDS	Cross Domain System or Solution
UAS	Unmanned Aerial Systems
CS	Compressive Sensing
RL	Reinforcement Learning
SDN	Software Defined Network
UAV	Unmanned Aerial Vehicle
MIMO	Multiple-Input-Multiple-Output
2D	Two-dimensions
MM-Wave	Millimeter Wave
AR	Synthetic Aperture Radar
TDM	Time Division Multiplexing
Tx	Transmitter
Rx	Receiver
RoI	Region of Interest
ADMM	Alternating Direction Method of Multipliers