Autonomous Aerial Search and Revisit Behavior for Communication Limited Environments

Wei Cui*, Animesh Shastry†, Derek A. Paley‡
University of Maryland (UMD), College Park, MD, 20742

Stephen M. Nogar§
DEVCOM Army Research Laboratory (ARL), Adelphi, MD 20783

This paper presents a autonomous search and revisit strategy for multi-agent unmanned aerial systems (UAS), specifically tailored for long-range operations where communication may be limited or denied. Multiple UAS are directed to do a wide-area search of an area, then revisit any detected objects of interest, confirming their identity. The strategy leverages onboard object detection and localization to prioritize data communication with a groundstation operator, with the aim of a future fully autonomous solution. Fast prototyping and integration was performed using an existing autonomy software package with software-in-the-loop (SITL) testing in a digital twin of the flight locations. The performance of the proposed search strategy is experimentally evaluated using custom-built quadrotor platforms equipped with onboard sensing and compute. The revisit strategy was found to be effective, particularly due to its ability to gather more data on identified objects, helping mitigate detection models that may not have been optimized for a specific scenario.

I. Introduction

In recent years, the field of unmanned aerial systems (UAS) has offered numerous opportunities and capabilities, ranging from search and rescue, precision agriculture, and disaster relief efforts [1]. One particular advancement within this field is the concept of a multi-UAS team, where multiple unmanned aircraft collaborate and coordinate their actions to achieve complex objectives [2]. An illustrative scenario is a long-distance search operation where several UAS are deployed to search a specific area. However, conducting long-distance operations with UAS presents notable challenges and considerations within the domain of aerial autonomy. The ability to establish and maintain communication links is paramount for the safe and effective operation of UAS, particularly when operating beyond the visual line of sight [3]. While UAS may be able to maintain short-range links between nearby vehicles, long distances can introduce signal degradation, latency, and vulnerability to interference [4]. In the absence of reliable long-distance communication, UAS may encounter difficulties in real-time decision making by a groundstation operator, compromising their overall flight safety and efficiency [5].

Several approaches to object detection have been previously reported, such as probabilistic approaches[6] and reinforcement learning [7]. Recently, there has been considerable interest in utilizing deep neural networks for object detection on UAS [8, 9]. There are two primary categories of object detection networks based on the number of neural networks involved: single-stage networks, where object detection and classification are performed in a single stage, and two-stage networks, which first predict candidate bounding box locations in the initial stage and subsequently conduct object classification in the second stage. Two-stage networks have been demonstrated to achieve higher accuracy, albeit at the cost of increased resource requirements [10, 11]. Single-stage networks like Single-shot Detector (SSD) [12] and You Only Look Once (YOLO) [13] typically offer faster inference speed and require fewer computational resources. Training a convolutional neural network becomes significantly slower as its depth increases[14]. In contrast, single-stage neural networks, which integrate object detection and classification, can be trained more quickly[15, 16].

Existing object geolocation methods leveraging RGB-D or LiDAR are suitable for vehicles that have a large payload capacity[17]. Monocular object localization has been widely studied in computer vision, but less explored on UAS [18].

*Graduate candidate, Department of Aerospace Engineering, University of Maryland, College Park, MD, 20742

†Graduate candidate, Department of Aerospace Engineering, University of Maryland, College Park, MD, 20742, AIAA Member

‡Willis H. Young Jr. Professor of Aerospace Engineering Education, Department of Aerospace Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742. AIAA Senior Member

§Autonomy Team Lead, U.S. Army DEVCOM Army Research Laboratory, Adelphi, MD 20783

To geolocate a ground object of interest, one simple approach is to intersect the ray starting from the camera and passing through the object pixel location in the image plane with the ground [18]. One of the challenges for object geolocation with onboard cameras is scale drift due to fast camera movement and vibrations [19]. This paper extends [19] and improves the geolocation accuracy by incorporating a clustering algorithm. The clustering algorithm is capable of analyzing raw localization data, associating new detections with prior ones, and precisely pinpointing object locations with the highest likelihood. These enhancements collectively contribute to refining the overall precision of geolocation.

This paper presents the following contributions: (1) the development and experimentation of a novel search and revisit capability utilizing MAVericks, an aerial autonomy stack developed by Army Research Laboratory (ARL), suitable for limited or denied communication environments; (2) the development of an enhanced object localization method for UAS deployment; (3) the development of a workflow to quantify performance of localization; and (4) the creation of specialized ground control software (GCS) to enhance situational awareness in communication-constrained environments. To facilitate simulation-based testing in Unity, a high-fidelity digital twin of the flight testing facility was constructed using photogrammetry. Field experiments have been conducted experimentally at the UMD UAS Research and Operations Center flight testing facility located in southern Maryland as well as the ARL Graces Quarters robotic research facility in northern Maryland.

The paper is organized as follows. Section II presents the problem statement and the proposed method. Section III presents the simulation results in Unity and *Fiona*, a custom-built GCS. Section IV presents the experimental results and visualization in *Fiona*. Section V summarizes the conclusions along with an overview of ongoing and future work.

II. Proposed Method

This section presents an overview of the search and revisit strategy tailored specifically for a swarm of UAS to detect and geolocate ground objects of interest in a communication denied environment. Moreover, we describe MAVericks, the ARL aerial autonomy stack operating on small UAS and provide a comprehensive explanation of techniques related to object detection and object localization. Lastly, we introduce *Fiona*, a custom-built user interface, which provides enhanced safety measures and improved mission situational awareness.

A. Overview

The search and revisit strategy proposed in this paper consists of two phases. In the initial phase, the UAS perform a broad area survey by dividing the search area into independent areas of similar sizes using K-means clustering [20]. This decentralized approach is effective in communication-denied environments where real-time centralized coordination may be challenging. Each area is then assigned an auto-generated lawnmower pattern for the aircraft to traverse systematically. Utilizing an onboard high-resolution RGB camera, the UAS detect and geolocate objects of interest within each area. Gaussian Mixture Models (GMM) are employed to determine the centroids of the clusters, which represent the most probable object-containing locations. In the second phase, the UAS optimize their flight paths using a multi-agent traveling salesman problem (TSP), also known as a vehicle-routing problem (VRP) to revisit the identified centroids. This optimization is essential for efficient coverage without the need for continuous communication with the ground station. Object locations, images, and aircraft positions are transmitted back to the ground station using ROS2 for visualization in a custom-built user interface called *Fiona*. To cope with limited or denied communication, the UAS store object locations and detection images onboard. The data is transmitted back to the ground station when communication is restored, ensuring that valuable information is not lost and can be analyzed later. The UAS return to the takeoff location automatically at the end of each phase, if no new mission is assigned. The overall strategy maximizes search coverage within a specified time constraint when continuously establishing and maintaining communication links with the ground station may not be feasible.

B. MAVericks

MAVericks is a ROS2-based autonomy software focused on agile flight that works across both simulation and robot platforms. Developed by ARL, it largely leverages functionality of open-source software. Currently, MAVericks can run on Modal AI VOXL and VOXL2. MAVericks is capable of behavior tree navigation, object detection and localization, precision landing, multi-agent teaming, obstacle avoidance, digital elevation maps, OpenVINS visual inertial odometry, as well as additional simulation environments in Unity verifying the desired behaviors in a software-in-the-loop (SITL) simulation [21].

C. Object Detection

YOLO has been widely used in applications such as autonomous driving, personnel recovery and rescue, and object retrieval and delivery [22]. The architecture unifies object detection and object classification for real-time performance [13]. Our YOLOv5-nano was pretrained on COCO dataset [23] and fine-tuned by Visdrone, which is a dataset captured by a variety of UAS [24]. The inference time on VOXL2 GPU is less than 40ms with MAVericks running.

D. Object Localization

Based on the center of the detection bounding box and the camera intrinsic matrix, the distance of the object from the camera can be inferred from the GPS coordinates and altitude of the UAS and the ground plane [19]. The algorithm is a relative cost-effective alternative to stereo camera and LiDAR to estimate the distance of an object of interest. After the distance from the UAS to an object is found, a series of transformations are used to calculate the object position in the world-fixed frame [19], i.e.,

$$\mathbf{T}_o^w = \mathbf{T}_b^w \mathbf{T}_c^b \mathbf{T}_o^c$$

where $\mathbf{T}_o^w$ is the transformation of the object position relative to the world frame that is used for globally-consistent representations of distances; $\mathbf{T}_o^c$ is the transformation of an object position relative to the camera frame; $\mathbf{T}_c^b$ is the transformation from the camera frame to the vehicle body frame; and $\mathbf{T}_b^w$ is the transformation of the vehicle body relative to the world frame. We operate under the assumption that the localization estimates for a given object follow a Gaussian distribution. We pinpoint the centroids of these Gaussians, which signify the most probable locations of the objects using Gaussian Mixture Models (GMM).

E. Ground Control Software

Fiona is a user interactive application created in React [25] and Spring Boot [26]. This application allows for both real-time and asynchronous visualization of object locations, UAS locations, and flight trajectories. Whereas existing tools such as QGroundControl[27] provide operators with only real-time telemetry and video, *Fiona* is designed to allow a user to collect data of detected objects from the UAS when communications are available. This software is positioned as an experimental aid to establish a robust search and revisit strategy with occasional operator involvement, and will eventually facilitate a fully autonomous solution.

The software is designed with modularity in mind, allowing for easy migration to different UAS platforms. The development process follows a test-driven and agile approach, ensuring the reliability and efficiency of the software. By developing automated tests alongside production code, we can introduce new features to meet operational requirements.

III. Simulation results

This section covers the selection of the vehicle routing solver and presents simulation results in Unity environment. Additionally, we showcase object detection in the simulated environment and *Fiona*.

A. VRP Solver Selection

The utilization of VRP solving for obtaining an optimized flight path has been implemented in multi-phase search and rescue missions [2]. The performance of various solvers on a 20-node problem for two agents is compared in Table 1. Probabilistic techniques are often preferred over deterministic algorithms for applications where computation resources are limited [28]. We use the VRP solver from Google OR-tools in MAVericks for its capabilities of solving VRP problems with additional constraints such as different start and stop points, waiting time, and vehicle capacity [29]. The additional constraints are not formulated in the original VRP, but are often required in a real-world application. Figure 1 shows the results obtained by the probability-based solver for a VRP. The problem consists of two agents starting from two different locations, traversing through the same 20 sites, and eventually meeting at the rendezvous point X. The solver is set with a 1-second time limit and is able to achieve satisfactory results.

B. Unity

Before conducting field experiments, we create and evaluate our algorithms through simulation. The primary flight test field for the UMD UAS Research and Operations Center is located at Raley Farm in southern Maryland. To represent the farm accurately, we constructed a 3D Unity scene, which serves as a digital twin, using photogrammetry techniques. Figure 2 illustrates the resulting digital twin of the farm. The UAS employs broad area surveying using lawnmower

Algorithm	Optimum Guaranteed	Complexity	Shortest path found	Run-time (sec)
Brute Force	Yes	O(n!)	395	> 30
Held Karp[30]	Yes	O(2^n)	395	28
Christofides[31]	No	O(n^3)	422	1.01
Kruskals[32]	No	O(n log(n))	677	0.02
Probabilistic[29]	No	-	395	1 (time limit)

Table 1 Performance comparison of VRP solvers for two agents on a 20-node complete graph with a shortest path of 395.

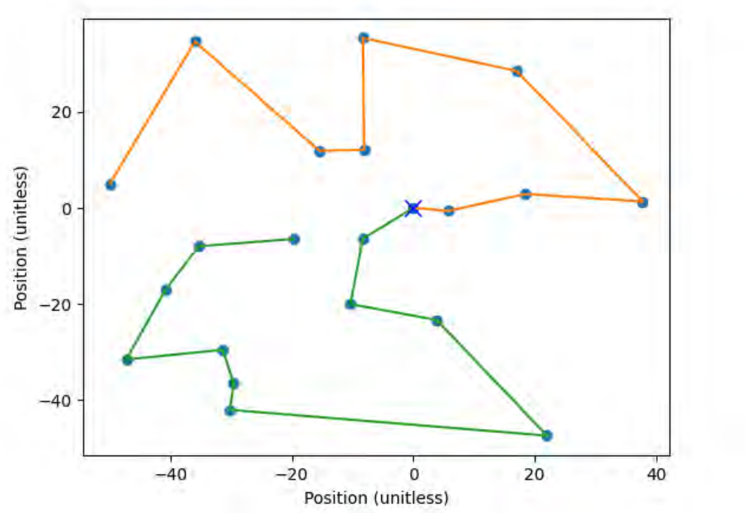


Fig. 1 The optimal paths start from two different locations and traverse 20 locations before converging at the X

patterns (Figure 3a) to detect and localize objects of interest. Subsequently, it revisits these objects at a lower altitude, following the optimal path obtained by the VRP solver, for detailed examination.

C. Object Detection

In order to enhance situational awareness, object locations and images are transmitted to the GCS via ROS2 in real time. In environments with limited bandwidth capabilities, the images are stored onboard and transmitted to the ground station once the bandwidth recovers. The visualization of object images in *Fiona* during the simulation is shown in Figure 4.

IV. Experiment Results

This section provides an overview of the hardware setup, software diagrams, and flight mission. Furthermore, we quantify the performance of localization using the rosbags recorded in the field. Subsequently, we present some end-to-end field experimental results.

A. Hardware Setup

The major electronic components on the UAS platform (Figure 5a) include the Modal AI VOXL2, mRo Control Zero H7 flight controller and the 2.4 GHz Doodle Labs mini radio. The theoretical communication range of the radio exceeds 10 km and it can achieve a maximum throughput of 80 Mbps. The high-resolution camera is configured to face the ground at an angle of 15 to 60 degrees. The system diagram in Figure 5b illustrates the wiring connections between the VOXL2 onboard computer, the flight controller, and the Doodle Labs mini radio, enabling long-range

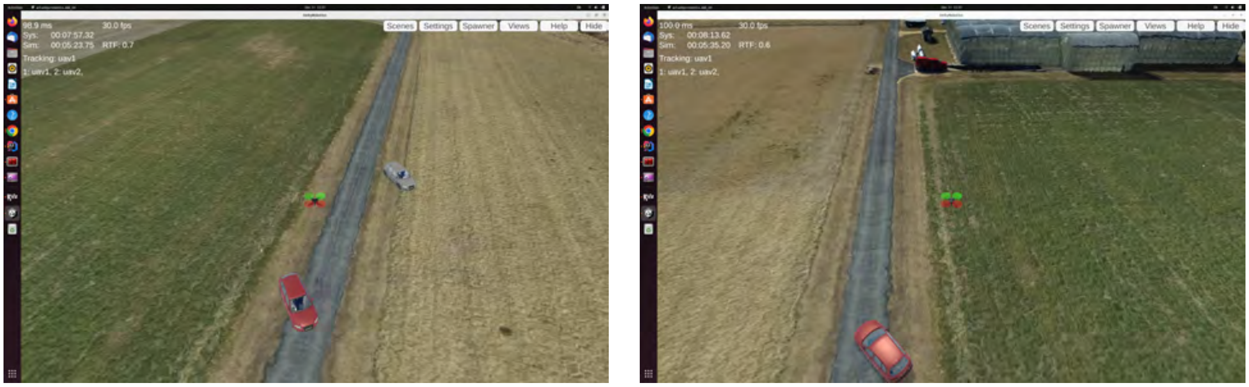
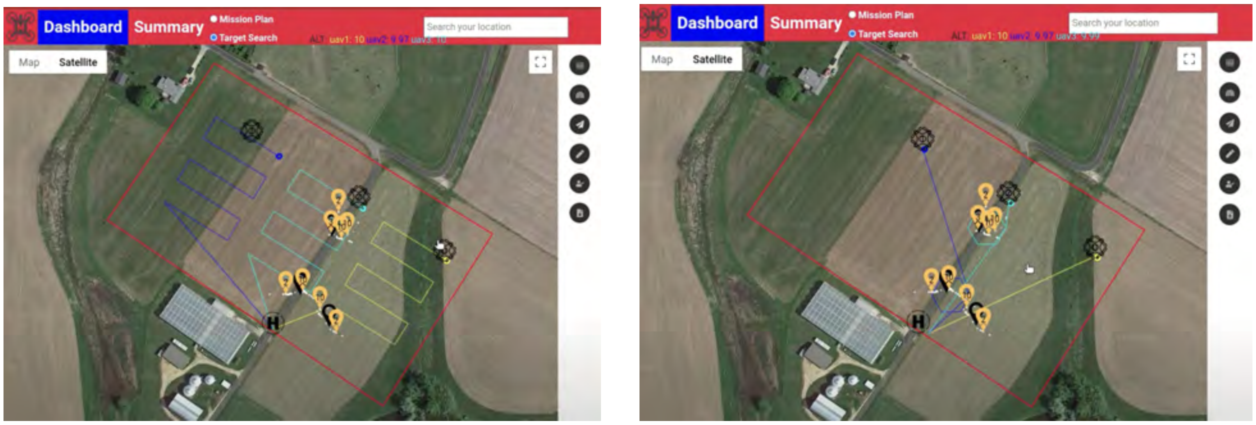


Fig. 2 Digital twin of flight test facility in Southern Maryland built in Unity for high-fidelity simulation



(a) Initial survey paths

(b) Revisiting the objects of interest with hexagonal orbits

Fig. 3 Visualization in *Fiona*: Object pins (yellow) are dropped in real time during the broad area survey; the black pins depict the ground truth target locations

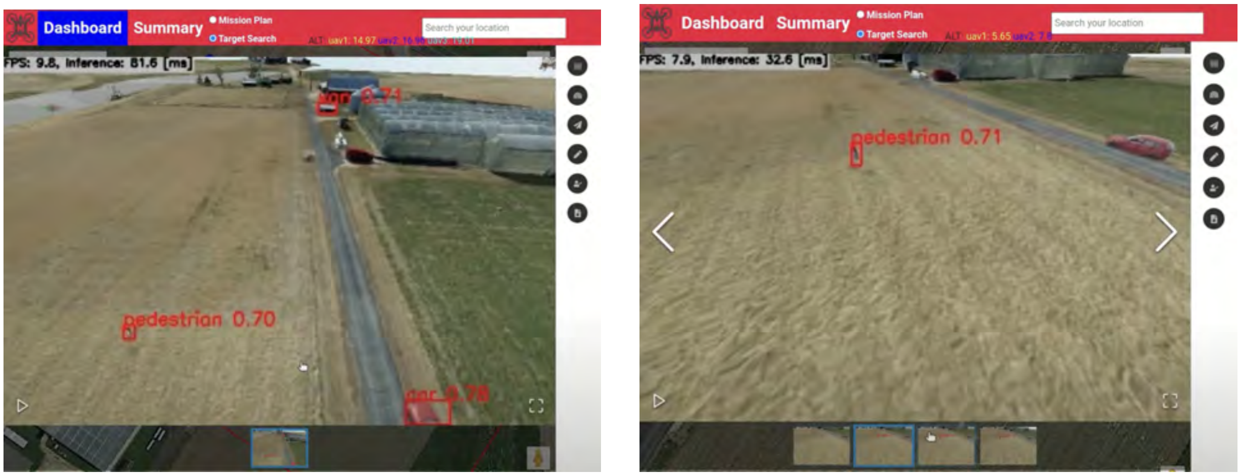
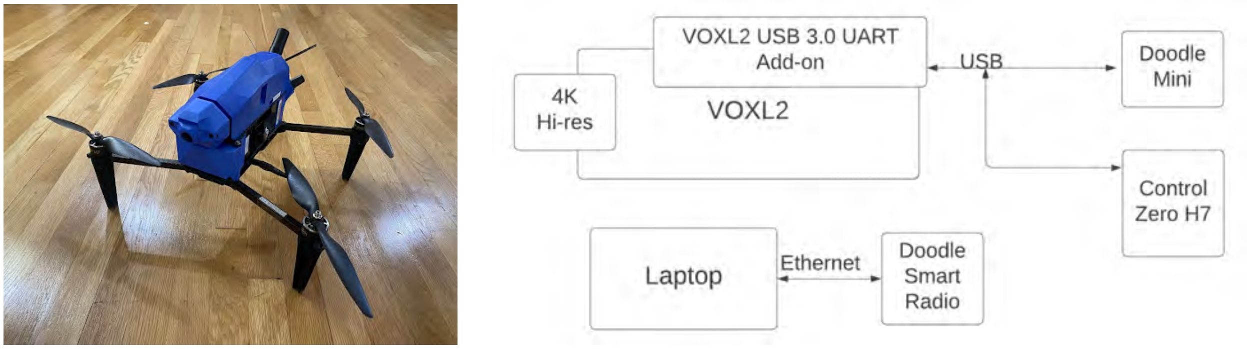


Fig. 4 Simulated object images are available in *Fiona's* gallery

communications. We use 900 MHz Jeti transmitters to avoid interference with the Doodle Labs radios. Additionally, we utilize Modal AI 4K high-resolution cameras for onboard perception algorithms.

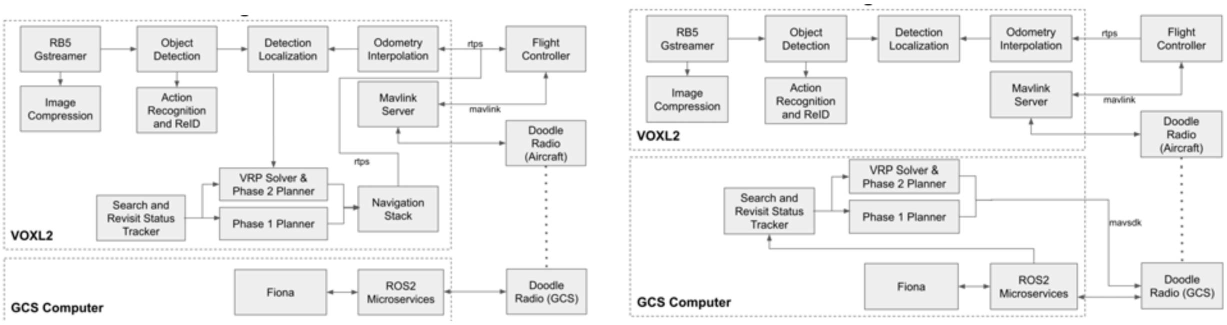


(a) Custom-built UAS **(b) System diagram**

Fig. 5 Custom-built UAS platform and system diagram developed for MAVericks autonomy stack

B. Software Diagrams

Figure 6a illustrates fully autonomous MAVericks configuration conducting search and revisit missions. This configuration involves running a variety of AI/ML algorithms such as object detection, advanced perception algorithms, and object geolocation on the UAS. The UAS also tracks mission status using a behavior tree and performs path planning with a VRP solver. All UAS are connected to the same Doodle network, which ensures that object locations are synchronized across the UAS, guaranteeing that the VRP solver returns the same paths. On the ground station, several ROS2 microservices collect telemetry data for *Fiona*, enabling real-time tracking of UAS positions, speed, and object locations. Because the fully autonomous configuration is resource-intensive and can cause the onboard computer to overheat, we have developed a simplified navigation stack. The less resource-intensive configuration shown in Figure 6b, where mission plans are created in JSON on the ground station and uploaded to the UAS through MAVSDK[33]. The advanced perception algorithms run on the UAS, even in the simplified configuration.



(a) Full-MAVericks configuration **(b) Simplified navigation configuration**

Fig. 6 Software diagrams developed for MAVericks autonomy stack to conduct search and revisit missions

C. Quantifying Localization Errors

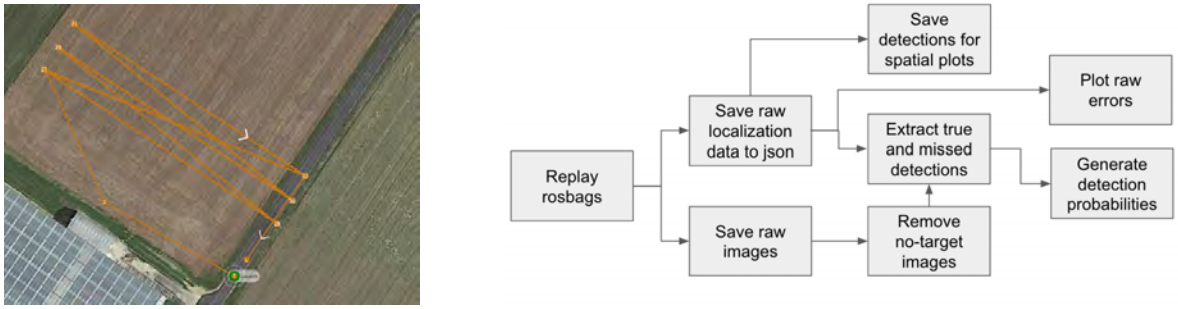
The accuracy of object localization during the broad area survey is essential for the UAS to revisit the objects accurately in the second phase. Localization errors are quantified at various altitudes, ranges, and camera angles by recording rosbags of the UAS flying a predefined mission, as illustrated in Figure 7a. This practice allows us to replicate the same mission with different camera angles.

To quantify localization errors, we placed a human actor in the field for the UAS to detect. The rosbags contain all the topics generated during the mission including raw images, detection images, UAS speed, GPS data, and object location estimations. We manually process the raw images and remove those that do not contain the target of interest. Subsequently, we count the number of false positives, true positives, and missed detections. The workflow is illustrated in Figure 7b, where the raw data is processed into a JSON file for ease of inspection. In the JSON file, each timestamp is associated with the estimated object location, bounding box size, and pixel location, along with the UAS GPS

Altitude (m)	**# True Positives** (TP)	**# False Positives** (FP)	**# False Negatives** (FN)	**# Opportunities** $(TP + FN)$	**Detection Probability** $\left(\frac{TP}{TP+FN}\right)$	**Accuracy** $\left(\frac{TP}{TP+FN+FP}\right)$
5	48	7	251	299	16.05%	15.69%
10	43	1	124	167	25.75%	25.60%
15	5	0	123	128	3.91%	3.91%
20	2	0	117	119	1.68%	1.68%
40	0	0	88	88	0.0%	0.0%

Table 2 Detection probability and accuracy at various altitudes and a camera angle of 15 degrees

coordinates, speed, altitude, range, and orientation.


Save detections for spatial plots
Plot raw errors
Save raw localization data to json
Extract true and missed detections
Generate detection probabilities
Replay rosbags
Save raw images
Remove no-target images

(a) Mission used to collect rosbags

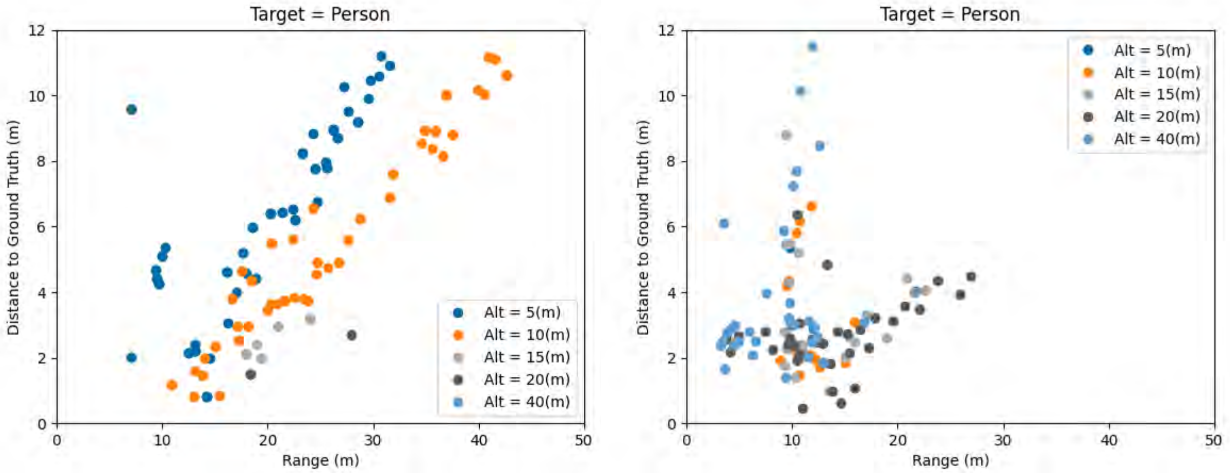
(b) Workflow for qualifying localization errors

Fig. 7 The quantification of object localization errors involves a series of postprocessing steps applied to rosbags collected in the field

Tables 2 and 3 list the detection probability and accuracy at various altitudes and camera angles. For a 15-degree angle relative to the horizon, the UAS achieves the highest detection probability and accuracy at an altitude of 10 meters. The performance deteriorates as the UAS ascends, and the number of detection opportunities also decreases with increasing altitude. When using a 60-degree camera angle, the UAS demonstrates a higher detection probability and accuracy at altitudes greater than 5 meters. However, it also encounters more false positives at lower altitudes as compared to a 15-degree angle. These false positives occur due to the misclassification of UAS shadows as the object of interest.

By following the workflow outlined in Figure 7, localization errors can be extracted from the data collected during flight. Figure 8 shows the impact of the camera on these errors. When the camera is set at a 15-degree angle from the ground level, the UAS can observe a range of up to 50 meters; however, the errors are larger when the UAS is far away from the object of interest. In addition to factors such as fast camera movement, GPS errors, and time-sync problems reported in [19], we observe that large errors can result from inaccuracies in placing the bounding box by the detection model. This is significant because the localization algorithm utilizes the center of the bounding box in the detection images to determine the object's real-world location. Figure 9 illustrates that the bounding box only encompasses a portion of the body when the human actor is close to the UAS and it encloses the entire body when the object is small. At a 15-degree camera angle, a one-meter deviation in the bounding box leads to an approximately 5-meter localization error. Conversely, with a 60-degree camera angle, a one-meter bounding box error results in a mere 0.6-meter localization discrepancy.

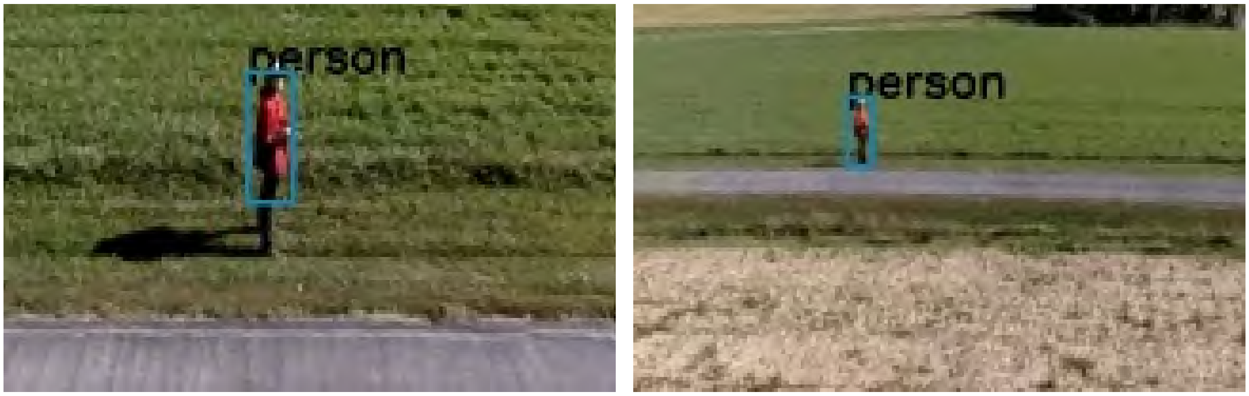
Therefore, when the camera angle is 60 degrees, the localization accuracy is less affected by inaccuracies of the bounding box. Figures 10a and 10b depict the relationships between UAS poses (blue chevrons) and estimated object locations (white circles), with black lines connecting the UAS poses to their respective object locations. These lines show the UAS's pose at the time of localization. The figures confirm that the localization errors are larger at the 15-degree camera angle than at the 60-degree angle. To enhance the overall accuracy of localization, we posit that



(a) Camera angle: 15°

(b) Camera angle: 60°

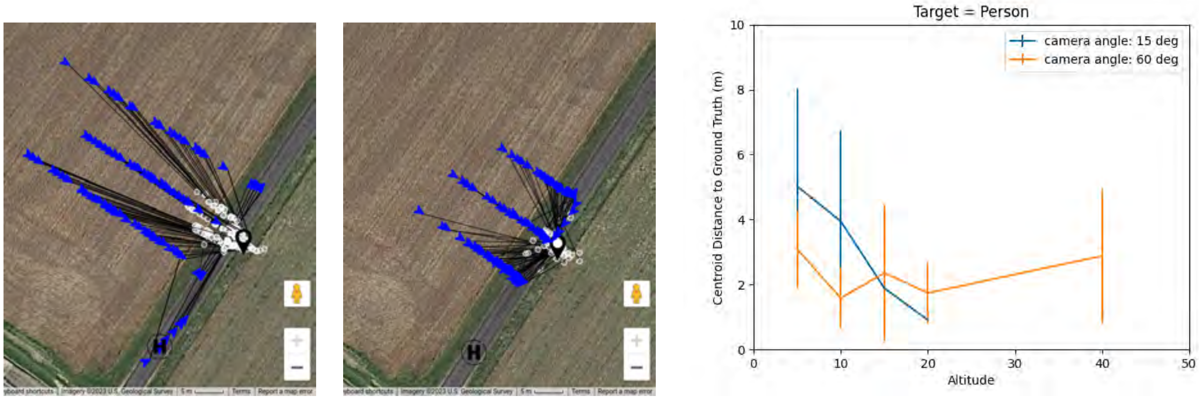
Fig. 8 Impact of camera angles on object localization errors



(a) The bounding box only encloses a portion of the body when the object is large.

(b) The bounding box encloses the entire body when the object is small.

Fig. 9 Inaccuracies of placing bound boxes occur when the object is large and close to the UAS



(a) Camera angle: 15°

(b) Camera angle: 60°

(c) Centroid errors and standard deviations

Fig. 10 UAS poses (blue chevrons) and estimated object locations (white circles)

Altitude (m)	**True Positives** (TP)	**False Positives** (FP)	**False Negatives** (FN)	**Number of Opportunities** ($TP+FN$)	**Detection Probability** $\left(\frac{TP}{TP+FN}\right)$	**Accuracy** $\left(\frac{TP}{TP+FN+FP}\right)$
5	3	40	55	58	5.17%	3.06%
10	15	3	45	60	25.00%	23.81%
15	24	0	61	85	28.24%	28.24%
20	35	0	59	94	37.23%	37.23%
40	33	0	123	156	21.15%	21.15%

Table 3 Detection probability and accuracy at various altitudes and a camera angle of 60 degrees

localization errors for the same object adhere to a Gaussian distribution. The center of this distribution signifies the most probable object location. Employing a Gaussian Mixture model allows us to pinpoint the centers or centroids, representing object locations with the highest likelihood. The centroid error is defined by Eqn (1), which is the distance from the centroid to the ground truth location. Figure 10c shows the impact of the camera angles and altitudes on the centroid errors. Centroid errors e_c and the standard deviation tend to be lower with a 60-degree camera angle compared to a 15-degree angle. Let

$$e_c = ||\bar{x} - x_t|| \tag{1}$$

where $\bar{x} = \frac{\sum x_i}{N}$, x_i is each detection location, x_t is the ground truth object location, and N is the number of detections.

D. End-to-end Field Experiment

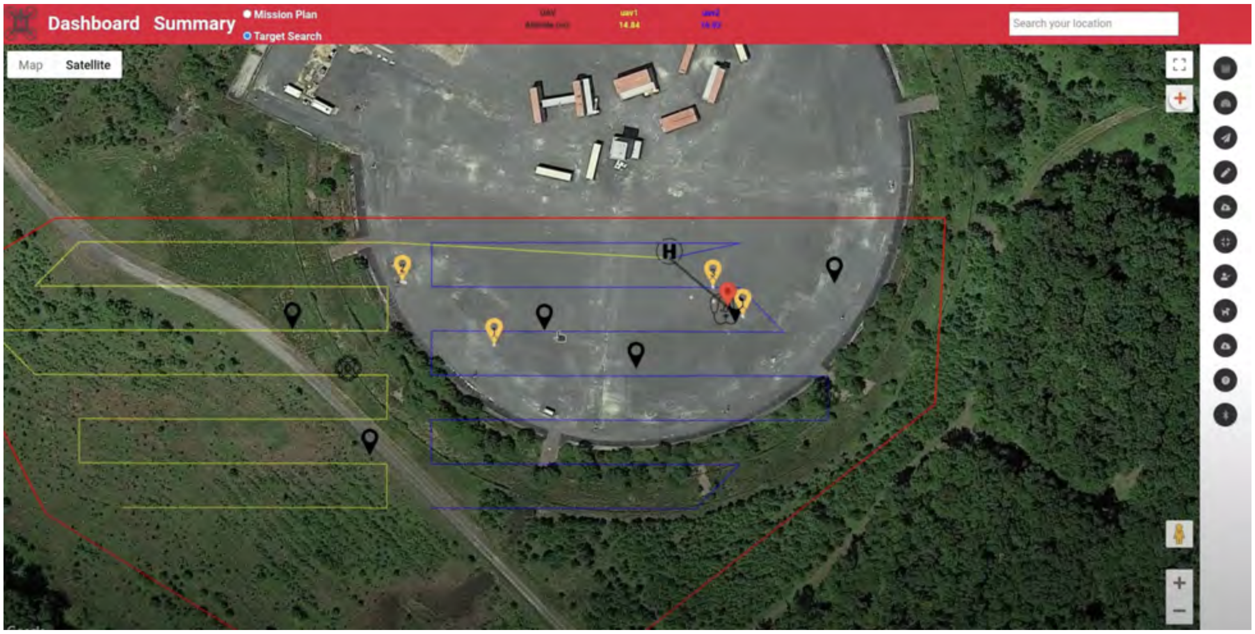


Fig. 11 Flight paths for the initial broad area survey with six objects of interest (black pins).

Here we describe a field evaluation that took place at Graces Quarters in northern Maryland, covering a test field of approximately 100 acres. For the experiment, we utilized five manikins dressed in Army uniforms and one human actor. Two of the manikins were positioned in the meadow field, while the other four were placed on the gravel pad, marked by black pins in Figure 11. This figure also illustrates the flight paths for the initial broad area survey and the initial estimated object locations indicated by yellow pins.

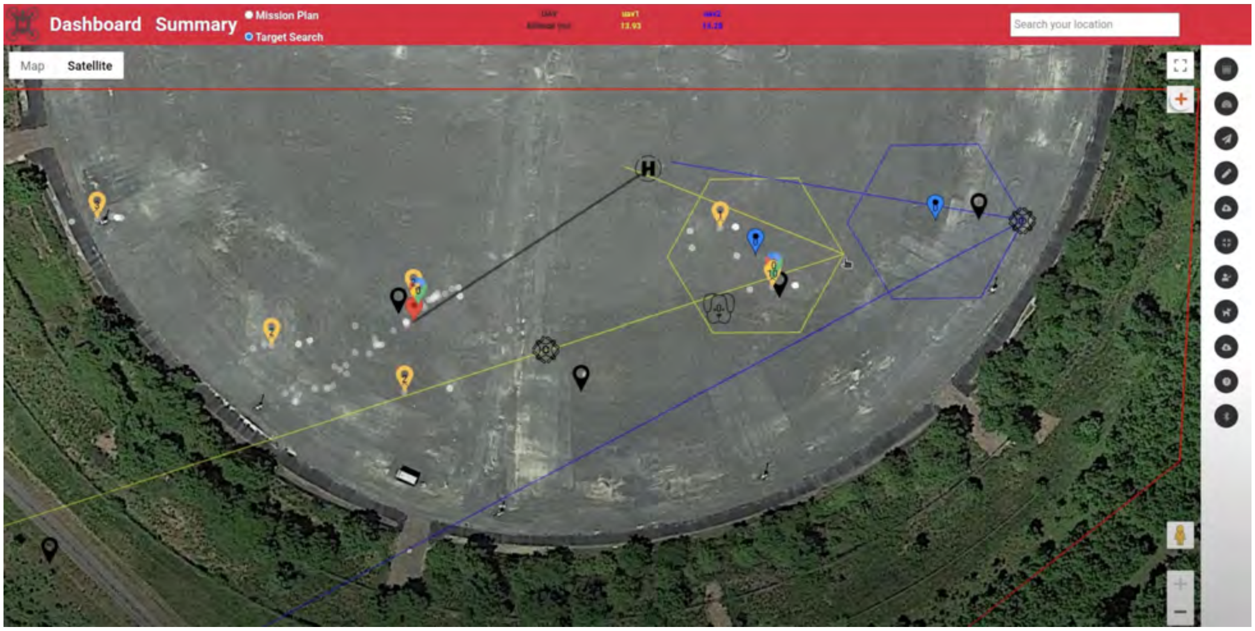
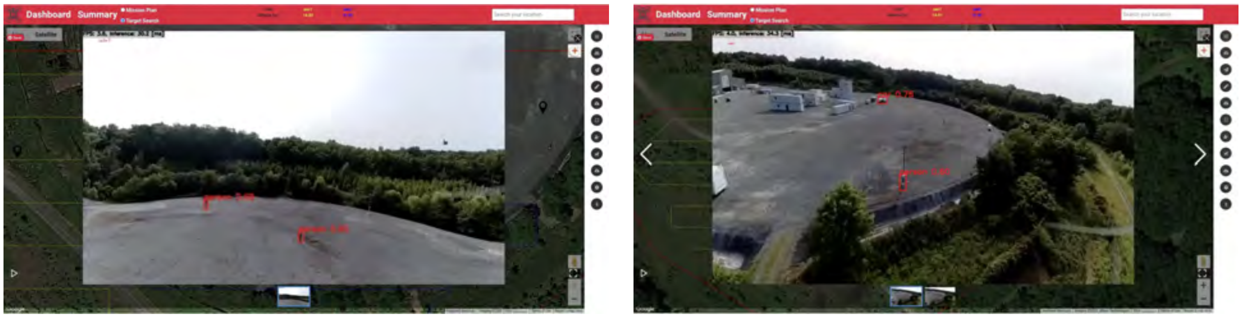


Fig. 12 Modified flight paths executed by the two UAS for revisiting the objects of interest.

After the initial broad area survey, the flight paths for revisiting the objects are auto-generated (not shown). To prevent breaching the geofence (marked by red lines) and to dispatch ground robots, the flight paths were modified using *Fiona*. The modified flight paths are shown in Figure 12. A ground robot is manually dispatched to revisit another object and its estimated walking path is indicated by a black line. The blue pins on the map are the locations to be revisited by the UAS. By right clicking on the colored pins on the map, the detection images that are transmitted back to the ground station are available for operators to inspect as shown in Figure 13.



(a) True positives - two objects of interest are correctly identified by the detection model

(b) False positive - the detection model misclassifies a lamp pole as a person.

Fig. 13 Detection images are transmitted back the ground station for operators to inspect in *Fiona*

Phases	Flight Time (mins)	Speed (m/s)	Altitude (m)	Objects Found/Total	Time on Objects (mins)
Broad Area Survey	5.83	3	15	3/6	0
Revisit	3.67	2	12	3/6	2.50

Table 4 Performance metrics of the end-to-end experiment

Table 4 presents the performance metrics of the end-to-end experiment. The entire mission takes approximately 9.5

minutes. The UAS detected three out of six objects. The two missed objects are located in the meadow, close in color to the Army uniform. Another missed object is lying down. In ongoing and future work, we aim to enhance the detection model's capability to identify objects in diverse backgrounds and poses.

V. Conclusion

This paper introduces a novel search and revisit capability developed for MAVericks. It enables a swarm of UAS to identify and geolocate objects of interest in a long-distance collaborative operation where communication may be denied. The broad area survey is divided into several areas of similar size using K-means. The flight trajectories are optimized by the VRP solver. A custom-trained neural network enables object detection and localization on the UAS. The accuracy of object localization is enhanced by GMM. Prior to experimental evaluation, rigorous testing of the algorithms is conducted within a high-fidelity digital twin of the flight test facility located in southern Maryland. End-to-end flight tests are conducted experimentally to assess the performance of the search and revisit capability. In the ongoing and future work, we seek further integrate the mission planning with ground robots for autonomous air-ground teaming. Ground robots will provide inspections in areas that are not visible to the UAS. Additionally, we will incorporate advanced perception algorithms such as action recognition and person re-identification into our software stack running on VOXL2.

Acknowledgments

This research was supported by Army Cooperative Agreement No. W911NF2120076. We thank Josh Gaus, Grant Williams, Mckenzie Turpin, and Darren Robey from the UMD UAS Research and Operations Center for conducting flight experiments. We extend our sincere gratitude to Benjamin Linne from the Army Research Laboratory and Isaac Carlson, Nikhil Deshmukh, Mike Rawding, Michael Smith, and Joel Witman from Survice Engineering for their expertise, which contributed to the development of both the hardware and software components of this work.

References

[1] Alsamhi, S. H., Shvetsov, A. V., Kumar, S., Shvetsova, S. V., Alhartomi, M. A., Hawbani, A., Rajput, N. S., Srivastava, S., Saif, A., and Nyangaresi, V. O., "UAV Computing-Assisted Search and Rescue Mission Framework for Disaster and Harsh Environment Mitigation," *Drones*, Vol. 6, No. 7, 2022. https://doi.org/10.3390/drones6070154.

[2] Zhang, X., and Ali, M., "A Bean Optimization-Based Cooperation Method for Target Searching by Swarm UAVs in Unknown Environments," *IEEE Access*, Vol. 8, 2020, pp. 43850–43862. https://doi.org/10.1109/ACCESS.2020.2977499.

[3] Ivancic, W. D., "Flying Drones Beyond Visual Line of Sight using 4G LTE: Issues and Concerns," *2019 Integrated Communications, Navigation and Surveillance Conference*, 2019. https://doi.org/10.1109/icnsurv.2019.8735278.

[4] Pons, M., Valenzuela, E., Rodríguez, B., Nolazco-Flores, J. A., and Del-Valle-Soto, C., "Utilization of 5G Technologies in IoT Applications: Current Limitations by Interference and Network Optimization Difficultiesmdash; A Review," *Sensors*, Vol. 23, No. 8, 2023. https://doi.org/10.3390/s23083876.

[5] Mohsan, S. A. H., Khan, M. A., Noor, F., Ullah, I., and Alsharif, M. H., "Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review," *Drones*, Vol. 6, No. 6, 2022. https://doi.org/10.3390/drones6060147.

[6] Ha, I.-K., and Cho, Y.-Z., "A Probabilistic Target Search Algorithm Based on Hierarchical Collaboration for Improving Rapidity of Drones," *Sensors*, Vol. 18, No. 8, 2018, p. 2535. https://doi.org/10.3390/s18082535.

[7] Wu, C., Ju, B., Wu, Y., Lin, X., Xiong, N., Xu, G., Li, H., and Liang, X., "UAV Autonomous Target Search Based on Deep Reinforcement Learning in Complex Disaster Scene," *IEEE Access*, Vol. 7, 2019, pp. 117227–117245. https://doi.org/10.1109/ACCESS.2019.2933002.

[8] de Oliveira, D., and Wehrmeister, M., "Using Deep Learning and Low-Cost RGB and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by Multirotor UAV," *Sensors*, Vol. 18, No. 7, 2018, p. 2244. https://doi.org/10.3390/s18072244.

[9] Cheng, Q., Wang, H., Zhu, B., Shi, Y., and Xie, B., "A Real-Time UAV Target Detection Algorithm Based on Edge Computing," *Drones*, Vol. 7, No. 2, 2023. https://doi.org/10.3390/drones7020095.

[10] Wu, X., Hong, D., Ghamisi, P., Li, W., and Tao, R., "MSRI-CCF: Multi-Scale and Rotation-Insensitive Convolutional Channel Features for Geospatial Object Detection," *Remote Sensing*, Vol. 10, No. 12, 2018, p. 1990. https://doi.org/10.3390/rs10121990.

[11] Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., and Zou, H., "Multi-Scale Object Detection in Remote Sensing Imagery With Convolutional Neural Networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 145, 2018, p. 3–22. https://doi.org/10.1016/j.isprsjprs.2018.04.003.

[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., *SSD: Single Shot MultiBox Detector*, Springer International Publishing, 2016, p. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.

[13] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Los Alamitos, CA, USA, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.

[14] He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[15] Online, "Multi-Digit Number Recognition From Street View Imagery Using Deep Convolutional Neural Networks," https://nladuo.github.io, 2018. Online; accessed Nov 18 2019.

[16] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y., "Reading Digits in Natural Images with Unsupervised Feature Learning," *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[17] Zhou, Y., and Tuzel, O., "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4490–4499. URL https://api.semanticscholar.org/CorpusID:42427078.

[18] Lim, H., and Sinha, S. N., "Monocular Localization of a Moving Person Onboard a Quadrotor MAV," *IEEE International Conference on Robotics and Automation*, 2015, pp. 2182–2189.

[19] Kim, J., Lin, K., Nogar, S. M., Larkin, D., and Korpela, C. M., "Detecting and Localizing Objects on an Unmanned Aerial System (UAS) Integrated With a Mobile Device," *International Conference on Unmanned Aircraft Systems*, 2021. https://doi.org/10.1109/icuas51884.2021.9476850.

[20] MacQueen, J. B., "Some Methods for Classification and Analysis of MultiVariate Observations," *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, edited by L. M. L. Cam and J. Neyman, University of California Press, 1967, pp. 281–297.

[21] Linne, Benjamin, "Streamlined Development Pipeline for Mavericks, ARL's Unmanned Autonomous Vehicle (UAV) Software," https://apps.dtic.mil/sti/citations/AD1181447, 2022. Online; accessed Apr 2023.

[22] Redmon, J., and Farhadi, A., "YOLOv3: An Incremental Improvement," *ArXiv*, Vol. abs/1804.02767, 2018.

[23] Community, "COCO - Common Objects in Context," https://cocodataset.org/#home, 2019. Online; accessed Apr 2019.

[24] Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., and Ling, H., "Detection and Tracking Meet Drones Challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, pp. 1–1. https://doi.org/10.1109/TPAMI.2021.3119563.

[25] Meta and Community, "React," https://react.dev/, 2021. Online; accessed Apr 2023.

[26] VMWare, "Spring Boot," https://spring.io/projects/spring-boot, 2021. Online; accessed Apr 2023.

[27] Community, "QGroundControl," http://qgroundcontrol.com/, 2020. Online; accessed Apr 2023.

[28] Mitzenmacher, M., and Upfal, E., *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*, Cambridge University Press, 2020.

[29] Google, "Google OR-tools, Google Developers," https://developers.google.com/optimization, 2021. Online; accessed Apr 2023.

[30] Held, M., and Karp, R. M., "A Dynamic Programming Approach to Sequencing Problems," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 10, No. 1, 1962, pp. 196–210. https://doi.org/10.1137/0110015.

[31] Christofides, N., "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem," *Operations Research Forum*, Vol. 3, 1976.

[32] Kruskal, J. B., "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem (1956)," *Ideas That Created the Future*, 2021, p. 179–182. https://doi.org/10.7551/mitpress/12274.003.0019.

[33] Community, "Introduction to MAVSDK," https://mavsdk.mavlink.io/main/en/index.html, 2023. Online; accessed Apr 2023.