Using Target Detection Probability to Evaluate Area Coverage by a UAV

Zachary Bortoff[1], Alec Luterman[2], Derek A. Paley[3], Stephen M. Nogar[4]

***Abstract*— A common task for unmanned aerial vehicles (UAVs) is wide area search using an onboard camera with an object detection model. However, constraints of flight time, camera optics, and onboard compute, particularly in time sensitive applications like search and rescue, requires trade-offs in strategies that balance precision and speed. To address these needs, we propose a novel method for evaluating coverage path plans by estimating the probabilities of detection and false alarm for ground targets for a set of poses that the UAV can reach in the search domain. To demonstrate our method, we evaluate trajectories for various coverage path plans flown by a UAV in a high-fidelity simulation.**

I. INTRODUCTION

Accurately locating injured people following mass casualty incidents, such as building collapses, is critical for informing first responders who must be triaged first. Unmanned aerial vehicles (UAVs) are effective in such search-and-rescue (SAR) operations due to their ability to rapidly survey hazardous or inaccessible terrain [12] [13] [5]. However, effective automated UAV-based SAR operations rely on coverage path planning (CPP) that must account for camera-based sensing and modern object detection models.

We consider the problem of coverage path planning (CPP) for a single UAV equipped with a calibrated pinhole camera that captures images at a fixed rate and an object detection model that identifies ground targets. Classically, CPP problems consist of planning a robot's path such that its sensor footprint covers every point within a defined search domain [7] [4] [14] [2]. In this paper, the sensor footprint is taken to be the area on the ground visible from the UAV's camera. The CPP problem we are interested in is to create a path plan for the UAV such that the entire search domain is not only visible from at least one image, but also that the likelihood of detecting a target with a given object detection model is greater than a prescribed threshold.

This work describes a method for transforming the pose of a mobile camera platform into probabilities of detection and false alarm for every cell in the approximate decomposition of a search domain. After performing approximate cellular decomposition of the search domain into cells equal in size to the expected target size, we present a statistical model relating the probabilities of detection and false alarm of ground targets to their apparent size in pixels. Assuming independence of consecutive object detection observations, we show how to recursively calculate the cumulative probabilities of detection and false alarm for each cell in the search domain. These probabilities are then used to evaluate coverage path plans by generating a heat map of probabilities of detection.

The contributions of this paper are (1) a method for transforming the trajectory of a camera-equipped aerial vehicle into the probabilities of target detection and false alarm for each cell of a grid-based decomposition of a search domain; (2) a quantitative framework for evaluating a vehicle's coverage path plan based on perception performance; and (3) the use of our evaluation framework to evaluate coverage path plans flown by a UAV in a high-fidelity simulation. This work provides a new method to evaluate coverage path plans in the context of objection detection.

Section 2 introduces domain decomposition methods, the concept of pixels-on-target, and a UAV autonomy stack called MAVericks. Section 3 outlines the approach we take to evaluate coverage path plans. Section 4 provides simulation results showing the evaluation procedure in a high-fidelity simulation environment within the MAVericks ecosystem. Section 5 concludes the paper.

II. BACKGROUND

A. *Pixels-on-Target*

Object detection is the computer vision task of localizing each instance of certain object classes in an image using bounding boxes [8]. Objects that appear small are empirically more difficult to detect than larger ones [11] [6]. Insufficient feature representation and background confusion are some reasons given to explain this observation [16]. A camera's intrinsic parameters, which encode information about image resolution, distortion, and focal length have significant implications for the number of pixels-on-target one should expect given a target's distance from the camera, size, and position in the image. These intrinsic parameters are mathematically described using a pinhole camera model.

The pinhole camera model describes the relationship between points in three-dimensional space and the projections thereof onto the two-dimensional image plane of the camera. The pinhole camera model is parameterized by a 3×3 intrinsic camera matrix K and a distortion model. Let $\mathcal{C} = (O, \hat{c}_1, \hat{c}_2, \hat{c}_3)$ denote the camera frame. Let $[\vec{r}_{P/O}]_{\mathcal{C}} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \in \mathbb{R}^3$ be a vector expressed in camera frame

[1]Zachary Bortoff is with the Department of Aerospace Engineering, University of Maryland, College Park, MD, USA. zbortoff@umd.edu
[2]Alec Luterman is with the Department of Computer Science, University of Maryland, College Park, MD, USA. amluter@umd.edu
[3]Derek A. Paley is with the Department of Aerospace Engineering and the Institute for Systems Research, University of Maryland, College Park, MD, USA. dpaley@umd.edu
[4]Stephen M. Nogar is with the DEVCOM Army Research Laboratory stephen.m.nogar.civ@army.mil
This research was supported by Army Cooperative Agreement W911NF2120076

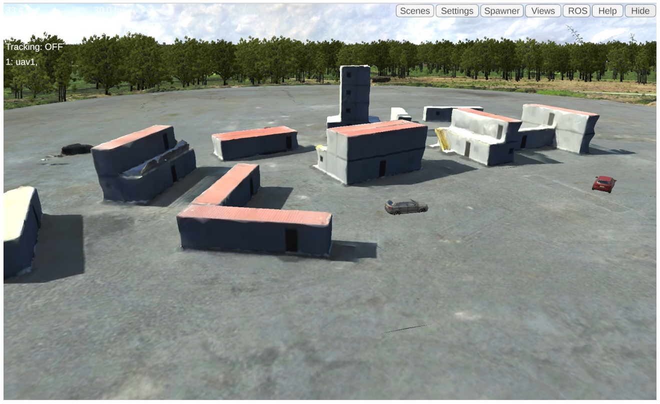


Fig. 1: Unity simulation for MAVericks of the ARL Graces Quarters robotic research facility in northern Maryland

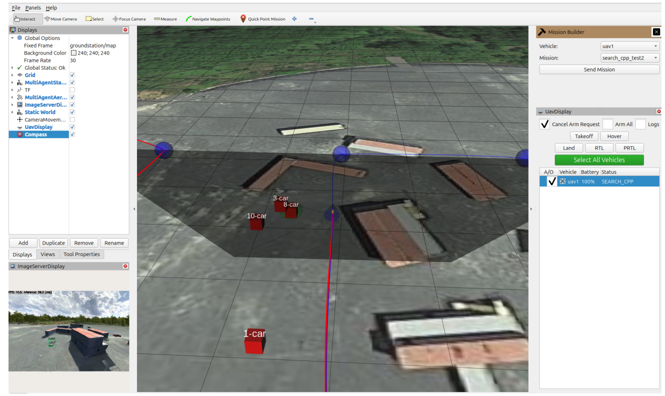


Fig. 2: Screenshot of MAVericks RViz ground station. The grey trapezoid is the sensor footprint of the UAV camera and red cubes are detected and localized targets, labeled with classification predictions. Blue dots are waypoints that the UAV will travel to.

Cartesian coordinates going from the origin of the camera frame O to some point P. The projection of P onto the image plane is

$$[\vec{r}_{P'/O}]_{\mathcal{C}} = \begin{bmatrix} u & v & 1 \end{bmatrix}_{\mathcal{C}}^T = \begin{bmatrix} X/Z & Y/Z & 1 \end{bmatrix}_{\mathcal{C}}^T ,$$

where u and v are so-called normalized, rectified image-plane coordinates. The distortion model is a function $d : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that maps normalized, rectified image-plane coordinates $[u, v, 1]^T$ to normalized, distorted image-plane coordinates $[u_d, v_d, 1]^T$. To express $[u_d, v_d, 1]$ in pixel coordinates, we apply the intrinsic matrix K, i.e.,

$$\begin{bmatrix} x & y & 1 \end{bmatrix}^T = K \begin{bmatrix} u_d & v_d & 1 \end{bmatrix}^T$$

The UAV camera in this paper is modeled as a calibrated pinhole camera. This framework enables us to have a mathematical relation between pixels in an image and the size of targets on the ground. We note that, while this feature is not invoked in this paper, it is possible to account for controlled changes to the camera parameters, for example, by zooming, as long as the camera's parameters are known at the time of capturing the image.

B. MAVericks

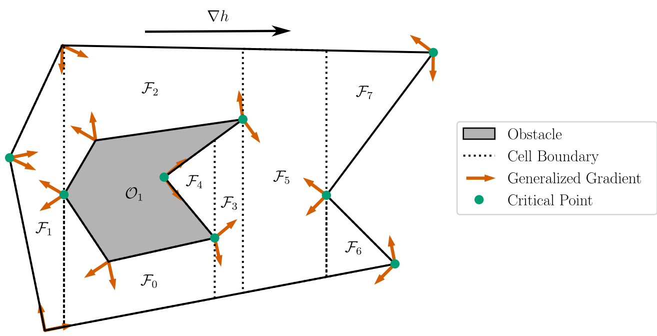
We use an aerial autonomy stack developed by the DEVCOM Army Research Laboratory (ARL), MAVericks, to demonstrate our coverage evaluation framework. The key features of MAVericks we invoke in this paper are: (1) its detection and localization pipeline, which localizes ground targets that are detected by Ultralytics YOLOv5 [9], fine-tuned on the VisDrone dataset [17]; (2) its path planning and trajectory optimization pipeline, which generates the coverage path plans that our method will evaluate; and (3) its ability to simulate its own computer vision and path planning algorithms in a high-fidelity Unity simulation, which includes virtual ground targets as seen in Figure 1. A screenshot of the MAVericks ground station is in Figure 2. This same aerial autonomy stack is also used onboard physical UAVs in the real world.

C. Morse Decomposition

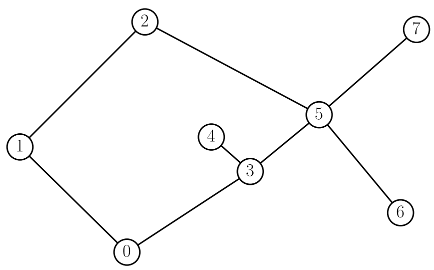
This section outlines an exact cellular decomposition approach to coverage path planning, Morse decomposition, that informs our construction of path plans in MAVericks. However, we note that our coverage path planning evaluation tool is agnostic to the coverage path planner used. Let the search domain $\mathcal{S}$ denote a compact subset of the plane $\mathbb{R}^2$. The boundary of $\mathcal{S}$ may be smooth or non-smooth and may enclose a finite number of compact subsets $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_N$, called obstacles, whose boundaries may also be smooth or non-smooth. Let the free-space $\mathcal{F}$ denote the complement of the union of all obstacles $\mathcal{F} = \left(\cup_{i=1}^N \mathcal{O}_i\right)^{\complement} \subset \mathcal{S}$. Exact cell decomposition methods partition $\mathcal{F}$ into non-empty, disjoint subsets $\mathcal{F}_i$, called cells, such that $\mathcal{F} = \cup_{i=1}^M \mathcal{F}_i$. Two cells are said to be adjacent if and only if they share a boundary. An adjacency graph $G = (V, E)$ encodes the cell decomposition, with the nodes of the graph representing cells and the edges representing the adjacency relation between cells. Complete coverage of any single cell is achieved by sufficiently spacing out simple back-and-forth motions, resembling a lawnmower pattern. Complete coverage of the entire free-space consists of finding a finite walk that visits each node of the adjacency graph at least once—called a Hamiltonian walk—and surveying any node along the walk that has not already been surveyed.

[1] introduced an exact cellular decomposition method based on Morse functions called Morse decomposition. A Morse function $h : \mathcal{F} \rightarrow \mathbb{R}$ is a smooth scalar function mapping elements of the free-space to the reals. The level-sets of a Morse function implicitly define non-overlapping curves called slices. For example, the slices of $h(x, y) = x$ are vertical lines and the slices of $h(x, y) = \sqrt{x^2 + y^2}$ are concentric circles around the origin.

The choice of Morse function has implications for efficient area coverage. We consider two families of Morse functions: linear and spiral. Linear Morse functions have the form $h(x, y) = ax + by$, $\forall a, b \in \mathbb{R}$ and produce

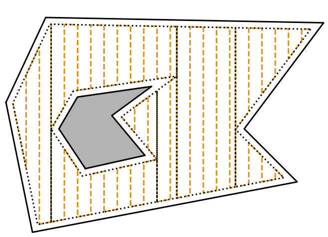


(a) Morse decomposition of search domain.

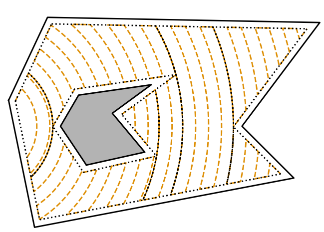


(b) Hamiltonian Walk: 0, 3, 4, 3, 5, 7, 5, 6, 5, 2, 1, 0

Fig. 3: (a) A sample non-smooth search domain with a single non-smooth obstacle decomposed using Morse function $h(x, y) = x$. (b) The adjacency graph corresponding to the decomposition in (a).



(a) Linear



(b) Spiral

Fig. 4: (a) and (b) Survey patterns generated by $h(x, y) = x$ and $h(x, y) = \sqrt{x^2 + y^2}$, respectively, for the example freespace.

evenly-spaced track lines. This survey pattern uniformly samples each cell. Spiral Morse functions have the form $h(x, y) = \sqrt{(x - a)^2 + (y - b)^2}$, $\forall a, b \in \mathbb{R}$ and produce evenly-spaced concentric track circles around (a, b). This survey pattern also uniformly samples each cell. In the subsequent section, we discuss a novel method of evaluating coverage performance of any coverage path plan.

III. COVERAGE EVALUATION FRAMEWORK

A. Sensor Footprint

This section outlines a procedure for computing a pinhole camera's sensor footprint assuming a flat Earth. The sensor footprint is defined as the set of points on the ground plane in the camera's scene. Let $\mathcal{I} = (O', \hat{e}_1, \hat{e}_2, \hat{e}_3)$ denote the world frame. The ground plane is defined as $G = \{\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}_{\mathcal{I}}^T \,|\, z = 0\}$. Let (x, y) denote arbitrary pixel coordinates. The corresponding normalized, rectified image-plane coordinates are

$$[\vec{r}_{P/O}]_{\mathcal{C}} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathrm{d}^{-1}\left(K^{-1}\begin{bmatrix} x & y & 1 \end{bmatrix}^T\right)$$

Note K^{-1} always exists, because it is positive definite by construction.

Let ${}^{\mathcal{I}}R^{\mathcal{C}}$ denote a rotation matrix representing the orientation of the camera in the world frame. Then the ray passing through pixel (x, y) is

$$[\vec{r}_{P/O'}]_{\mathcal{I}}(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = \left({}^{\mathcal{I}}R^{\mathcal{C}}[r_{P/O}]_{\mathcal{C}}\right) t + [\vec{r}_{O/O'}]_{\mathcal{I}},$$

where $[\vec{r}_{O/O'}]_{\mathcal{I}}$ is the position of the camera in world frame coordinates and $t \in \mathbb{R}$ parameterizes the ray.

If $Z(t_0) = 0$ for some $t_0 > 0$, then $[\vec{r}_{P/O'}]_{\mathcal{I}}(t_0)$ corresponds to a position in world frame coordinates where the ray intersecting pixel (x, y) also intersects the ground plane at a point. The set of all such points is the sensor footprint.

B. Estimating Probabilities of Detection and False Alarm for an Object Detection Model based on Object Size

We propose a simple statistical model for predicting the probabilities of detection and false alarm of an image-based object detection model given the object's size in pixels.

The probability of detection, $P_D(s)$, is the likelihood that an object of size s pixels is detected given its presence in the image. We estimate this probability by computing the relative frequency of correctly detected objects of size s to objects of size s in a test data set. Specifically, we:

1) Define a set of size intervals (bins) $S = \{s_1, ..., s_N\}$, partitioning the range of observed object sizes in pixels.
2) For each bin s_i, estimate the detection probability as $P_D(s_i) := N_{\text{det}}(s_i)/N_{\text{true}}(s_i)$ where $N_{\text{det}}(s_i)$ is the number of correctly detected objects within size range s_i, and $N_{\text{true}}(s_i)$ is the total number of ground-truth objects in that size range. Note this construction corresponds to the maximum likelihood estimate of the detection probability, assuming the detections are Bernoulli distributed.
3) Construct a bar plot mapping object size intervals to their corresponding empirical detection probabilities.

This plot permits efficient lookup of $P_D(s)$ for any object of interest by simply identifying the appropriate bin and retrieving the associated probability.

The probability of false alarm, $P_{FA}(s)$, represents the likelihood that a detection of size s pixels corresponds to a false positive, i.e., an object, is detected when none is present. We estimate this probability by computing the relative frequency of false alarms as a function of detection size in a test data set. Specifically, we:

1) Define the same set of size intervals S as in the detection probability computation.
2) For each bin s_i, compute the false alarm probability as $P_{FA}(s_i) = 1 - N_{\text{det}}(s_i)/N_{\text{total}}(s_i)$ where $N_{\text{total}}(s_i)$ is the total number of detections made within size range s_i, including both correct detections and false alarms.
3) Construct a bar plot mapping detection size intervals to their corresponding false alarm probabilities.

This statistical model of the performance of a given object detection model is parameterized by a confidence threshold, which encodes the confidence over which detections made by the model are interpreted as valid detections, and the intersection-over-union (IoU) threshold, which encodes the degree to which a detection bounding box and a ground-truth bounding box must overlap to be considered a true positive or false positive. These thresholds are used to filter out poor object detections.

C. Probabilities of Detection and False Alarm over Search Domain

We discretize the search domain into cells of uniform size via approximate cellular decomposition. Given a pinhole camera pose, the size of any cell in pixels can be computed using the sensor footprint procedure discussed above. By mapping the cell size in pixels to the probabilities of detection and false alarm histograms, we predict the probabilities of detection and false alarm of a hypothetical target in that cell.

Assuming a sequence of independent observations using the object detection model are made of a particular cell, then we can recursively compute the probability of detecting a target in that cell given a target is present in that cell in any of the observations as

$$P_D^{(k)} = P_D^{(k-1)} + \left(1 - P_D^{(k-1)}\right) P_D(s),$$

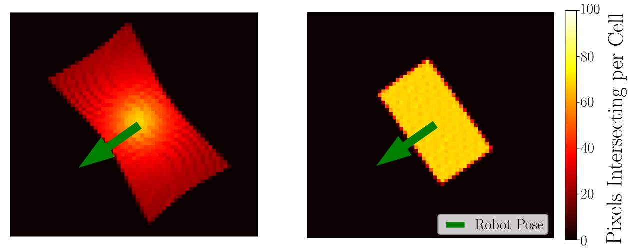
where $P_D(s)$ is the probability of detection of the current observation and $P_D^{(k)}$ is the probability of detection after k independent observations. The same equation applies for $P_{FA}^{(k)}$, which denotes the probability of detecting a target in that cell given a target is absent from that cell in any of the k independent observations.

Despite making several simplifying assumptions, this procedure for computing the probabilities of detection and false alarm for cells within the search domain has several practical benefits. Mission plans generated on the basis of this model will account for their camera's distortion, resolution, and attitude as well as their drone's altitude in establishing the likelihood of detecting targets of particular size on the ground. In the following section, we use this CPP framework to evaluate several UAV missions.

IV. RESULTS

A. Sensor Footprint Analysis

Figure 5 depicts the sensor footprints of two pinhole cameras with different distortion models, illustrating the impact of mounting different lenses on the camera. These figures are meant to illustrate the impact of lens choice on our pixels-on-target evaluation procedure. The simulations in a subsequent section assume a plumb-bob model with minimal distortion as depicted in Figure 5b.



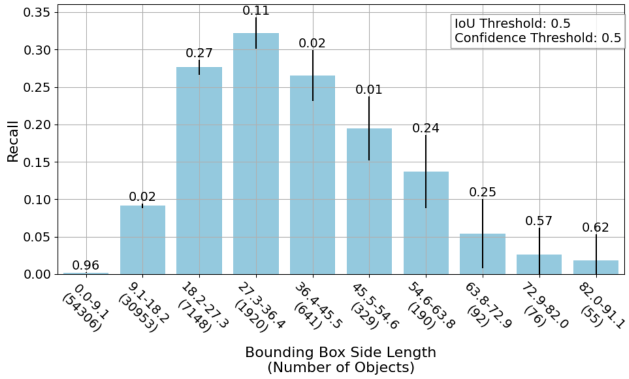
(a) Fisheye distortion (b) Plumb-bob distortion

Fig. 5: Sensor footprints of downward-facing pinhole camera with a low distortion plumb-bob model and a highly distorted fisheye model. Heat map displays the number of rays extending from each pixel of the camera that intersect with the ground plane in each cell.

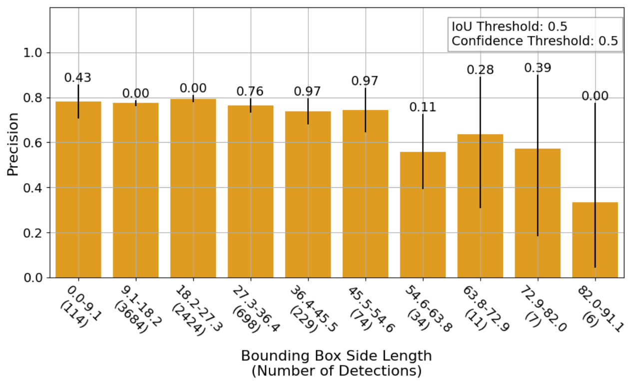
B. Object Detection Analysis

Figure 6 shows the probability of detection (recall) and precision by object size of the medium YOLOv8 model [10] searching for classes "pedestrian" and "people" in the training partition of the VisDrone data set [17] at 0.5 IoU threshold and 0.5 confidence threshold. The black vertical lines at the top of each bin indicate 95% confidence intervals around the recall and precision estimates. To validate this statistical model, the same experiment was carried out against the validation partition of the VisDrone data set, and the observed frequency of detections within each bin was compared to the expected frequency as predicted by our statistical model. p-values computed by performing several chi-squared hypothesis tests on the basis of the observed and expected frequencies are depicted at the top of each bin. Six of the twenty p-values fall below 0.05 confidence threshold, meaning the model does not predict the expected frequency of detections within those bins. This highlights the inherit limitation of using a single feature (target size) in predicting detector performance. Including more features into such a model may improve accuracy, but the model performs sufficiently well across all bin sizes to be used to inform path planning strategies. We emphasize that these results are specific to our particular object detection model and image data; we do not expect these specific results to generalize to another model or data set.

Bear in mind that the images taken by the drone must be resized to account for the input dimensions of the YOLO model. The object sizes reported in the ground truth annotations were normalized to account for this resizing. The figures indicate that the probability of detection is relatively low when objects are both small and large. In contrast, precision seemingly degrades with the size of the detected object. These results motivate the need to fly at an altitude

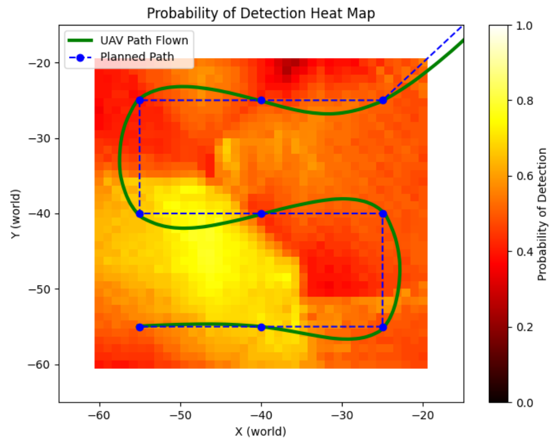


(a) Recall

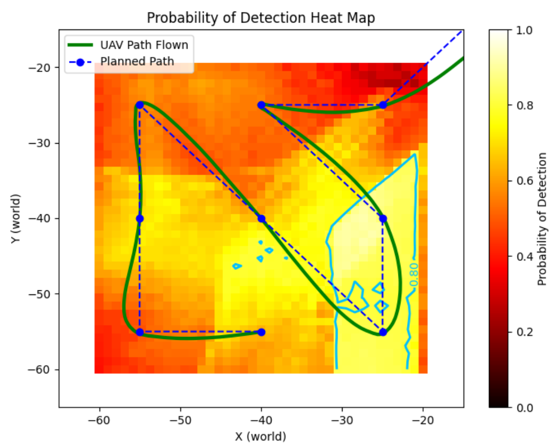


(b) Precision

Fig. 6: Estimated recall and precision by object size. Black lines at the top of each bin indicate 95% confidence intervals. p-values characterizing degree to which this model predictions are consistent against validation data set written over each bin. The units of bounding box side length are pixels.



(a) Linear Search Pattern at an altitude of 10 m



(b) Spiral Search Pattern at an altitude of 10 m

Fig. 7: Planned flight path of the UAV, actual flight trajectory from MAVericks simulation, the search domain, and the probability of detection contours. Regions outlined in light-blue show areas of the search domain that reach a probability of detection threshold of 0.80.

so that ground targets of known size appear in the regime with the highest probability of detection.

C. Coverage Path Planning Performance in High-Fidelity Simulation Environment

This section demonstrates our coverage performance evaluation framework in MAVericks's high-fidelity simulation environment. We configure the simulation environment to take place at a 3D mock village populated with ground targets of interest, i.e., people and cars. We created coverage paths for a simple search domain whose vertices are at the following world frame Cartesian coordinates: $(-60,-60,0)$, $(-60,-20,0)$, $(-20,-20,0)$, $(-20,-60,0)$ in meters. As the drone executes the plan, the camera captures images at a rate of 5 Hz and sends them through the detection and localization pipeline. After finishing the path plan, the UAV's camera poses over time, the camera's intrinsic parameters, and the statistical model of the object detection performance are loaded into a tool that evaluates the coverage path planning performance. This tool decomposes the search domain into a grid of 1x1 m cells and follows the method outlined previously to compute the probabilities of detection for each cell of the search domain.

Figure 7 shows example search path plans, actual flight trajectories, and heat maps of the cumulative probability of detection for each cell in the search domain for linear and spiral search patterns. The differences between the planned and actual paths are due to MAVericks's trajectory optimization. The cumulative probability of detection, as shown in Figure 7a, of the linear search pattern flown at an altitude of 10 m is patchy and entirely below the required threshold of 0.8. By contrast, the cumulative probability of detection of the spiral search pattern flown at the same altitude, as shown in Figure 7b, is mostly below the desired threshold of 0.8, but it does a better job than the linear search plan;

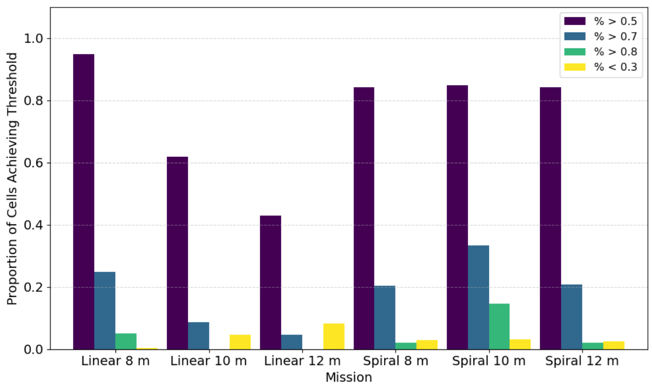


Fig. 8: Proportion of cells in search domain that achieve probability of detection threshold at altitudes of 8, 10, and 12 meters.

the proportion of cells in the search area with a probability of detection greater than 0.8 is 0.145.

We also investigated the effect of the altitude of the path plan on the probability of detection. Figure 8 shows the proportion of cells in the search domain that achieve the probability of detection threshold by altitude and search pattern. We flew three identical missions in simulation but varied the altitude between 8 m, 10 m, and 12 m for both linear and spiral patterns. In the linear case, the figure shows a negative relationship between altitude and the proportion of cells in the search domain achieving a given probability of detection. The proportion of cells achieving the threshold of 0.8 at altitudes of 10 m and 12 m is 0.0, but 0.051 at an altitude of 8 m. The proportion of cells achieving a threshold of 0.5 is 0.948 at an altitude of 8 m, 0.619 at 10 m, and 0.429 at 12 m. In the spiral case, the figure does not show a negative relationship between altitude and the proportion of cells in the search domain achieving a given probability of detection. According to Figure 8, for thresholds of probability of detection greater than 0.5, 0.7, and 0.8, the mission with an altitude of 10 m had the highest proportion of cells achieving each of these three thresholds, and the proportions for 8 m and 12 m were within 4 thousandths of each other for each threshold.

These results validate the utility of the coverage evaluation framework and demonstrate that the interplay between camera intrinsics, flight path, and object detection model performance make predicting area coverage difficult. These results motivate developing more sophisticated coverage path planning algorithms that can account for these system parameters using our framework. Even with using the same nine (x,y) waypoints for all six flights, the difference in routing and altitude resulted in significant differences in coverage quality based on the proportion of cells that achieved a target probability of detection. In ongoing work, we are using our insights from this evaluation to develop a new coverage path planner that is focused on guaranteeing a minimum probability of detection throughout the entire search domain.

V. Conclusion

We propose an evaluation method for offline coverage path planning algorithms for a UAV equipped with a calibrated pinhole camera using an object detection model. We show how to transform the robot's trajectory into probabilities of detection and false alarm for every cell of a decomposed search domain, and how to evaluate coverage path plans based on probabilities of detection. We also showed the need for more advanced coverage path planning algorithms through our results from a high-fidelity simulation.

In ongoing and future work, we are exploring vantage point coverage path planning algorithms in order to achieve a minimum probability of detection throughout the entire search domain. By solving a variant of the set cover problem [3], a set of vantage points is generated such that the union of the sensor footprint of the UAV at each vantage point covers the entire search domain. A vehicle routing problem is then solved to calculate the path the UAV will fly, where the UAV stops for a short duration at each vantage point to obtain better object detection performance due to issues with target detection performance while the drone is moving [15].

Another approach we are exploring for coverage path planning is generating a set of possible path plans by varying the parameters for search patterns such as track spacing, altitude, and speed. We then will use our evaluation framework to choose the best path plan based on which set of parameters gives the highest proportion of cells achieving a minimum threshold probability of detection throughout the entire search domain.

References

[1] Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, and Douglas Hull. Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4):331–344, 2002. https://arxiv.org/abs/https://doi.org/10.1177/027836402320556359 arXiv:https://doi.org/10.1177/027836402320556359, https://doi.org/10.1177/027836402320556359 doi:10.1177/027836402320556359.

[2] Prasad N. Atkar, Aaron Greenfield, David C. Conner, Howie Choset, and Alfred A. Rizzi. Uniform coverage of automotive surface patches. *The International Journal of Robotics Research*, 24(11):883–898, 2005. https://arxiv.org/abs/https://doi.org/10.1177/0278364905059058 arXiv:https://doi.org/10.1177/0278364905059058, https://doi.org/10.1177/0278364905059058 doi:10.1177/0278364905059058.

[3] Graeme Best and Robert Fitch. Probabilistic maximum set cover with path constraints for informative path planning. 12 2016.

[4] Michael Bosse, Navid Nourani-Vatani, and Jonathan Roberts. Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 698–703, 2007. https://doi.org/10.1109/ROBOT.2007.363068 doi:10.1109/ROBOT.2007.363068.

[5] Milan Erdelj, Enrico Natalizio, Kaushik R. Chowdhury, and Ian F. Akyildiz. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Computing*, 16(1):24–32, 2017. https://doi.org/10.1109/MPRV.2017.11 doi:10.1109/MPRV.2017.11.

[6] Alireza Fathi, Anoop Korattikara, Chen Sun, Ian Fischer, Jonathan Huang, Kevin Murphy, Menglong Zhu, Sergio Guadarrama, Vivek Rathod, Yang Song, and Zbigniew Wojna. Speed and accuracy trade-offs for modern convolutional object detectors. Honolulu, Hawaii, 2017.

[7] Douglas W. Gage. Randomized search strategies with imperfect sensors. In William J. Wolfe and Wendell H. Chun, editors, *Mobile Robots VIII*, volume 2058, pages 270 – 279. International Society for Optics and Photonics, SPIE, 1994. https://doi.org/10.1117/12.167503 doi:10.1117/12.167503.

[8] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderhauf. Probabilistic object detection: Definition and evaluation. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1020–1029, 2020. https://doi.org/10.1109/WACV45572.2020.9093599 doi:10.1109/WACV45572.2020.9093599.

[9] Glenn Jocher. YOLOv5 by Ultralytics, May 2020. URL: https://github.com/ultralytics/yolov5, https://doi.org/10.5281/zenodo.3908559 doi:10.5281/zenodo.3908559.

[10] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL: https://github.com/ultralytics/ultralytics.

[11] Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 680–688, 2016. https://doi.org/10.1109/CVPRW.2016.90 doi:10.1109/CVPRW.2016.90.

[12] Mingyang Lyu, Yibo Zhao, Chao Huang, and Hailong Huang. Unmanned aerial vehicles for search and rescue: A survey. *Remote Sensing*, 15(13), 2023. URL: https://www.mdpi.com/2072-4292/15/13/3266, https://doi.org/10.3390/rs15133266 doi:10.3390/rs15133266.

[13] Sharifah Mastura Syed Mohd Daud, Mohd Yusmiaidil Putera Mohd Yusof, Chong Chin Heo, Lay See Khoo, Mansharan Kaur Chainchel Singh, Mohd Shah Mahmood, and Hapizah Nawawi. Applications of drone in disaster management: A scoping review. *Science Justice*, 62(1):30–42, 2022. URL: https://www.sciencedirect.com/science/article/pii/S1355030621001477, https://doi.org/10.1016/j.scijus.2021.11.002 doi:10.1016/j.scijus.2021.11.002.

[14] F. Yasutomi, M. Yamada, and K. Tsukamoto. Cleaning robot control. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1839–1841 vol.3, 1988. https://doi.org/10.1109/ROBOT.1988.12333 doi:10.1109/ROBOT.1988.12333.

[15] Mehran Yazdi and Thierry Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:157–177, 2018. URL: https://www.sciencedirect.com/science/article/pii/S1574013716301794, https://doi.org/10.1016/j.cosrev.2018.03.001 doi:10.1016/j.cosrev.2018.03.001.

[16] Yin Zhang, Mu Ye, Guiyi Zhu, Yong Liu, Pengyu Guo, and Junhua Yan. FFCA-YOLO for small object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. https://doi.org/10.1109/TGRS.2024.3363057 doi:10.1109/TGRS.2024.3363057.

[17] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. https://doi.org/10.1109/TPAMI.2021.3119563 doi:10.1109/TPAMI.2021.3119563.