Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-18 (2025) 175–180

Distributed Multi-Robot Control for Streets Surveillance from Aerial Images with Neural Networks ⋆

Mattia Catellani, Andrea Alboni, Lorenzo Sabattini

University of Modena and Reggio Emilia, Italy (e-mail: {mattia.catellani, lorenzo.sabattini}@unimore.it).

Abstract: We propose a distributed strategy for aerial robot teams to maximize street coverage in monitoring and search-and-rescue missions. A neural network extracts roads from aerial images, generating a probability density via Gaussian Mixture Models to guide robots toward detected streets. Each robot performs a Voronoi-based coverage mission, focusing on areas of interest. The neural network's performance was validated using Google Maps images, while the control architecture was tested in realistic simulations, demonstrating effectiveness.

Keywords: Multi cooperative robot control, Networked robots, Mobile robots and vehicles.

1. INTRODUCTION

The deployment of multi-robot systems has significantly increased recently, demonstrating their expanding potential across a wide range of scientific and practical domains (see Dorigo et al. (2021)). The reason for their growing adoption lies in the many advantages they offer with respect to the employment of a single robot, such as robustness to single points of failure, reduced time required to accomplish missions, and the capability to deal with large-sized environments. Examples of tasks where swarm robots are successfully employed and show their full potential are coverage control, as proposed in Cortes et al. (2004), whose aim is maximizing the monitored area, agriculture as shown in Ju et al. (2022), search and rescue in Queralta et al. (2020), and exploration of the environment as in our previous work Catellani et al. (2022). In addition, the use of Unmanned Aerial Vehicles (UAVs) offers further advantages compared to ground robots, e.g. an optimal birds-eye view and the capability to easily avoid obstacles on the ground, making them particularly suitable for operations on disaster areas where no humans or other vehicles can access to.

In this paper we employ a team of UAVs to address the problem of streets surveillance. A similar problem was addressed in Bai et al. (2020), where the authors propose a method to solve a formation control problem for monitoring an expanding flood area, splitting the team into two subgroups, one external and the other one internal, with the aim of tracking the boundary of the area of interest and monitoring the region itself, respectively. Monitoring tasks are often faced exploiting neural networks, whose employment is nowadays widely spread in nearly all fields of research, not just in robotics. An example is the work proposed in Thoduka et al. (2021), where a U-Net, that is a convolutional neural network specially developed for image segmentation Ronneberger et al. (2015), is trained with the aim of detecting anomalies while a task is being executed by a robot. In Vallejo et al. (2009) instead a multi-robot system exploits neural networks in order to gain information from the environment and coordinate with each other in order to accomplish a surveillance mission.

Fig. 1. Realistic simulation of the area to be monitored.

In this work we propose a distributed control strategy for a team of UAVs to reach a desired region of the environment and maximize its coverage. More in details, we consider the task of monitoring the streets in a selected area, moving robots above them and spreading in order to maximize the target surface. We make use of a convolutional neural network, the widely known U-Net, trained to extract roads from satellite images. Then, a proper probability density is defined fitting a Gaussian Mixture Model to the mask generated by the neural network, in order to define a higher probability in areas over the streets where UAVs are required to move. A similar strategy was adopted in Yao et al. (2020), where neural networks are exploited in combination with Gaussian Mixture Models to accomplish a search and rescue task with a team of UAVs. However, the authors only consider curve-shaped 1-D target areas such as coastlines or streets modelling a 1-D Gaussian Mixture Model, forcing the team to assume a linear configuration, which may not be ideal for a surveillance task. Gaussian Mixture Models are also employed in Lin and Goodrich

⋆ This work was supported by Gruppo Tecnoferrari, Italy.

10.1016/j.ifacol.2025.10.216

(2014) to model the probability distribution of the target's location and optimize path planning for a single UAV.
In our work, robots are controlled in a decentralized manner to perform Voronoi-based coverage, being attracted where the probability density is higher, thus achieving the goal of the mission. The proposed solution is finally tested selecting an area from Google Maps and generating high fidelity large-sized virtual environments (see Fig. 1), evaluating performances in terms of covered surface.

The contribution of this paper is then the definition of an integrated control strategy that, starting from aerial images, is able to automatically detect the areas of interest (i.e., the roads) and subsequently deploy a fleet of UAVs in a way that optimizes the overall capability of observing the environment. The proposed methodology combines a convolutional neural network for image analysis, Gaussian Mixtures Models for describing the relative importance of different areas, and Voronoi-based coverage control for UAV deployment.

2. PRELIMINARIES

This section provides a brief introduction to the notation that will be used in the rest of the paper, and a formulation of the problem addressed in this work together with the assumptions made to face it.

2.1 Notation and Definitions

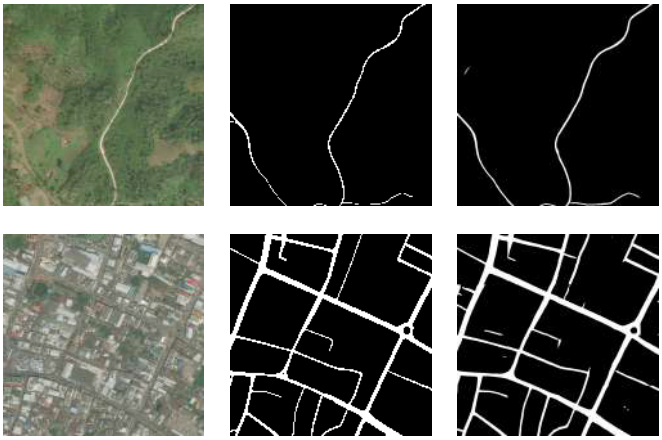
In the rest of the paper, we denote by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ the set of natural, real, real non-negative, and real positive numbers, respectively. Given $x \in \mathbb{R}^n$, let $\|x\|$ be the Euclidean norm. Instead, given the matrix $\Sigma \in \mathbb{R}^{n \times m}$, we define $|\Sigma|$ as its determinant.
Let $\mathbb{F}(\mathbb{R}^2)$ be the collection of finite point sets in $\mathbb{R}^2$. We denote an element of $\mathbb{F}(\mathbb{R}^2)$ as $\mathcal{P} = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$, where $\{p_1, \ldots, p_n\}$ are points in $\mathbb{R}^2$. We denote, for $p \in \mathbb{R}^2$ and $r \in \mathbb{R}_{>0}$, the closed ball in $\mathbb{R}^2$ centered at p with radius r with $B(p, r) = \{q \in \mathbb{R}^2 | \|q - p\| \leq r\}$. In the paper, $Q \subset \mathbb{R}^2$ denotes a generic polygon: it will be used, in particular, to denote the environment where the robots are supposed to operate. An arbitrary point in Q is denoted by $q \in Q$.

2.2 Problem Description

Consider a team of $n \in \mathbb{N}$ aerial robots flying at the same constant altitude, thus moving in a 2D environment. The altitude is assumed to be sufficient to avoid obstacles on the ground. We assume each robot to be modeled as a single integrator system, whose position $p_i \in \mathbb{R}^2$ evolves according to $\dot{p}_i = u_i$, where $u_i \in \mathbb{R}^2$ is the control input, $\forall i = 1, \ldots, n$. We consider the following assumptions:

- *Localization*: each robot is able to localize itself with respect to a global reference frame, shared among robots within the team.
- *Limited Sensing Capabilities*: each robot is able to detect and localize neighbors inside its limited sensing range, defined as a circle $B(p_i, r)$.

Based on these assumptions, the problem addressed in this paper can be described as the implementation of a control architecture for street surveillance employing a team of UAVs with limited sensing capabilities. The goal is to provide a straightforward workflow that enables a human operator to choose a location where the task must be carried out, and then operate the robotic team to maximize the monitored portion of the regions of interest.



(a) (b) (c)

Fig. 2. Prediction test: (a) input image, (b) desired mask, (c) prediction of the trained U-Net.

3. MODEL TRAINING

In this section, we will present the architecture of the neural network employed to extract roads from satellite images, and we will detail its components and the strategy adopted for training and evaluation of the results.
As already mentioned, the neural network's architecture is modeled after the well-known U-Net Ronneberger et al. (2015), a convolutional neural network specially developed for image segmentation, thus perfectly suiting our use case. In the following, we will briefly present the structure of the network: for additional details, the reader is referred to Ronneberger et al. (2015). First of all, an appropriate dataset has been retrieved from the DeepGlobe Road Extraction Challenge Demir et al. (2018), whose goal was to advance research on roads extraction in order to enhance crisis response in disasters zones, especially in developing countries. This dataset contains 6226 satellite images, each one paired with a mask image for road labels.

3.1 Architecture of the U-Net

The developed network accurately reproduces the standard structure of a U-Net, consisting of a contracting path taking an image with shape $256 \times 256 \times 3$ and an expansive path, returning an image with the same shape as the input. The contracting path follows the typical architecture of a convolutional network, containing encoder layers that progressively reduce the input size while increasing the number of channels, in order to capture high-level features. Each encoder layer consists of the application of two 3×3 convolutions, each one followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation for downsampling. The expansive pathway subsequently transforms the feature map obtained from the contracting path into an image of identical dimensions as the initial input. Every

(a) (b) (c)

Fig. 3. Fitting a Gaussian Mixture Model to a satellite image: (a) input image, (b) predicted mask, (c) heatmap of the probability density generated by the GMM, with darker colors corresponding to higher values.

layer in the expansive path involves an upsampling, which reduces the number of channels while increasing the resolution of the feature map up to the shape of the input image. In addition, skip connections from the contracting path are exploited to allow upsampling layers to find and refine the features in the image. The output image therefore is a binary segmentation map whose shape is the same as the input. Depending on whether it is considered to belong to a street or not, each pixel is either white or black.
The described network was subsequently trained and tested on the aforementioned dataset in order to learn to operate roads extraction even in previously unseen satellite images.

3.2 Training and Evaluation

The dataset of labeled images was split into training, testing and validation sets, with a proportion of 80%, 10%, and 20%, respectively. The optimizer chosen for training was Adam Kingma and Ba (2014), an extension to stochastic gradient descent method adaptively estimating first-order and second-order moments. The main parameters for the optimizer were the learning rate $\alpha = 0.0001$ and the exponential decay rates for the first and second moment estimates, namely $\beta_1 = \beta_2 = 0.99$. The loss function was calculated as the sum of the Dice loss Sudre et al. (2017), widely adopted for pixel segmentation, and the binary focal loss Lin et al. (2017). The Jaccard coefficient, also known as Intersection over Union (IoU), was finally calculated to test the accuracy of predictions on images from the testing set, evaluating how much the binary map generated by the model overlaps the label mask.
After the training phase, labeled images from the validation set were passed as input to the U-Net in order to test its behaviour with previously unseen images. Examples of results are shown in Fig. 2, where predicted masks are compared to the labels from the dataset. In addition to the visual results, the efficiency of the trained U-Net is also proved by the Jaccard coefficient, whose mean value on the validation set was 0.61, indicating good performances in the generation of binary masks from new images.

4. GMM DEFINITION

In this section we present a basic introduction to Gaussian Mixture Models, and we explain how we make use of this tool to define a proper probability density from an input image. A Gaussian Mixture Model (GMM) is a probabilistic model where the sample set is assumed to be generated from a mixture of a finite number $k \in \mathbb{N}$ of Gaussian distributions. Since we are considering a 2D environment, each mixture component is a bivariate Gaussian distribution defined by a mean point $\mu_i \in \mathbb{R}^2$ and a covariance matrix Σ_i (see Kotz et al. (2004)). The overall model is obtained as the combination of single components according to a specific mixture proportion, defined by a weighting factor $w_i \in \mathbb{R}_{>0}$ associated to each component, with

$$\sum_{i=1}^{k} w_i = 1. \tag{1}$$

In our work, we make use of GMMs to define a particular probability density such that robots are driven into the desired regions. More in details, since we are interested in covering streets surface, we aim at fitting a suitable GMM to the areas detected by the trained U-Net. The fitting operation entails selecting the optimal set of parameters $(\mu, \mathbf{\Sigma}, \mathbf{w})$ for a GMM from a set of input data points. The estimation of these parameters is performed with a *Maximum Likelihood* method as described in McLachlan et al. (2019), exploiting an *Expectation-Maximization* algorithm to iteratively find the optimal set of parameters. The number k of Gaussian components can be arbitrarily chosen, but the algorithm is also capable of finding the optimal number. For this purpose, the Bayesian Inference Criterion (BIC) can be calculated, a likelihood-based measure used to compare multiple models fit to the same data. In our case, we use the foreground pixels (i.e., white pixels in Fig. 2c) in the binary mask generated by the U-Net as the input samples for the *Expectation-Maximization* algorithm.
Once the GMM has been defined, we calculate the probability density function as the sum of the contributions brought by the single components. According to the definition in Kotz et al. (2004), the contribution of a single d-dimensional component (in our case, $d = 2$) is calculated as:

$$\phi_i(q, \mu_i, \Sigma_i) = \frac{\exp\left(-\frac{1}{2}(q-\mu_i)\Sigma_i^{-1}(q-\mu_i)^T\right)}{\sqrt{|\Sigma_i|(2\pi)^d}}. \tag{2}$$

From the above equation, the resulting contribution of a single Gaussian component to the global probability density depends on the covariance matrix Σ_i, which defines the spatial distribution around the mean point μ_i and is specifically calculated to fit the sample set. Finally, the overall probability function is calculated as the weighted sum of every single component, according to their weighting factor w_i:

$$\Phi(q, \mu, \Sigma) = \sum_{i=1}^{k} w_i \phi_i\left(q_i, \mu_i, \Sigma_i\right). \tag{3}$$

The outcome of the described method is the definition of a non-uniform density of the environment, where each point $q \in Q$ is assigned with a high probability value if belonging to a street. An example is shown in Fig. 3, where a binary mask is generated by the U-Net from a satellite image, and a GMM is calculated fitting its foreground pixels. With this strategy, areas where a street is detected are assigned with a high density in order to attract the robots inside. In this way, the team of UAVs will be placed above streets, giving them the best possible view for monitoring.

5. DISTRIBUTED UAVS CONTROL

After describing the implementation of the neural network and how the generated binary mask is exploited to highlight areas of interest, we present the distributed control algorithm for the multi-robot system. The Voronoi-based coverage control Cortes et al. (2004) serves as a foundation for our approach, allowing robots to optimally arrange in order to maximize the covered surface according to a specified probability density. Briefly, coverage control exploits Lloyd's algorithm to reach an optimal configuration for the system, which maximizes the covered area. This solution exploits the Voronoi partitioning to divide the total area into cells, optimally allocating a region of the environment to each robot. Since we employ robots with limited sensing capabilities, they do not have a global knowledge of the positions of their teammates, and only directly measurable information can be used to calculate the Voronoi partitioning in a decentralized fashion. For this reason, a *limited* Voronoi partitioning is carried out, according to the definition provided in Pratissoli et al. (2022):

$$V_i(\mathcal{P}) = \{q \in B(p_i, r) | \|q - p_i\| \leq \|q - p_j\|, \forall p_j \in \mathcal{P}\}. \quad (4)$$

According to the above formulation, the i-th Voronoi cell contains all points in the environment within the sensing range of robot i that are closer to i than to any other robot. After the definition of the limited Voronoi partitioning, each robot is able to autonomously calculate the centroid of its cell, taking into account the probability function (3):

$$C_{V_i} = \frac{\int_{V_i} q\Phi(q, \mu, \Sigma)dq}{\int_{V_i} \Phi(q, \mu, \Sigma)dq}. \quad (5)$$

The control action can be finally calculated in order to move each robot towards the centroid of its cell:

$$u_i = -k_p (p_i - C_{V_i}) \quad (6)$$

where $k_p \in \mathbb{R}_{>0}$ is a proportional gain.
It is interesting to note that the adoption of the probability function (3) calculated fitting a GMM to the binary mask generates an attractive effect into the region of interest, moving robots over streets, while the Voronoi partitioning guarantees an optimal distribution of the agents in order to maximize the covered street surface. In addition, the whole process can be performed by each UAV only exploiting locally available information, with no need for communication among the agents nor with a central computing unit.

6. EXPERIMENTAL EVALUATION

After presenting the methodologies used in our work, we show how the entire workflow has been tested with simulations. In particular, we show how a human operator is allowed to select an area from Google Maps taking a snapshot of a desired region and feeding it to the U-Net. Then, we present how a realistic environment is created in order to reproduce the selected area, and how a GMM is calculated fitting the output binary mask from the neural network, whose probability density is exploited for the decentralized control of UAVs. Finally, we present the results of our tests, analysing the evolution of the team's configuration and performance metrics.

(a) (b)

Fig. 4. Test region: (a) Snapshot of the area from Google Maps, (b) heatmap of the calculated GMM, with darker colors corresponding to higher values.

6.1 Simulation Setup

Virtual tests were conducted employing aerial robots within the RotorS Gazebo simulator framework developed by Furrer et al. (2016), controlled exploiting the ROS middleware. Due to the need for a visual feedback on the accuracy of our methodology, we created realistic virtual environments with the aim of faithfully reproducing the region to be monitored. In this way, it has been possible to visually check whether the developed control strategy is successful in moving the UAVs over streets. The Gazebo world was generated searching for a specific area on Google Maps and extracting the 3D features with the RenderDoc software (see Karlsson (2019)), a graphics debugger capable of capturing rendered frames. The saved 3D model was then imported into Blender, a 3D open-source graphics software, and converted into a suitable format for the Gazebo simulator. The result was a realistic 3D environment reproducing the desired area to be monitored, where it is possible to control robots and evaluate how they interact with the environment. An example is shown in Fig. 1, where the area surrounding the Coliseum in Rome is reproduced.
Different tests were performed in different environments, evaluating the effect of both the number of robots and the sensing range on the coverage capability of the team. In every simulation, robots were controlled in a decentralized manner, and no communication was allowed among them. Communication was only allowed with the simulation engine in order to emulate sensing capabilities, sending information about the position of neighbors only when they were found inside the sensing range.

6.2 Roads Extraction and GMM Fitting

As previously mentioned, the first step of the workflow can be identified with the choice of the region to be monitored. After identifying the desired location and generating the Gazebo world using the previously discussed method, we took a snapshot of the satellite image from Google Maps to be taken as the input for the U-Net. Fig. 4a shows one example among several satellite images used to test performances in different scenarios. In particular, this image shows an area of 300×200 m^2 with a simple scenario of a roadway in a flat environment. Then, this aerial image has been resized in order to be consistent with the neural network, which requires the input images to have a size of 256×256 pixels. The resized images was subsequently elaborated by the trained model of the U-Net in order to recognize streets and generate a binary mask with white pixels for streets and black pixels for the background. Finally, white pixels were taken as the dataset for the GMM

Fig. 5. Final positions of a team of 16 UAVs.

Fig. 6. The team of UAVs is not capable of monitoring such a large area.

fitting. For this specific environment, we employed $k = 20$ Gaussian components, determined evaluating the BIC and comparing it against values calculated for various alternative numbers of components. This choice of 20 components balanced the trade-off between the speed of the fitting algorithm execution and the GMM quality in modeling the samples set. The resulting GMM is represented with a heatmap in Fig. 4b, and clearly show that the resulting probability density has the highest values in areas where streets are detected, thus highlighting their importance for the UAVs team.

It is interesting to note that all the steps described up until this point are carried out offline by a central computer. As a matter of fact, applying the trained U-Net model to the satellite image and fitting a GMM to the resulting mask can be quite time-consuming, taking a few seconds to be carried out. However, this operation is only needed before starting the mission, thus not affecting the control loop of each robot.

6.3 Analysis of Results

After fitting a GMM to the desired environment, the effectiveness of the proposed solution was tested controlling the robotic team and evaluating their behaviour. In order to achieve a statistical significance for our results and understand how different parameters influence the performances of the team, we ran a series of several simulations for every environment. More in details, different tests were conducted with teams employing 12, 16, and 20 UAVs, with a varying sensing range r of 7.5, 15, and 30 m. For the sake of simplicity, the same sensing range values were considered both for the detection of other robots, and the area coverage. Random starting positions were chosen for each run, and drones were assumed to be flying at a constant altitude $H_z = 1.0$ m, which was high enough to consider the environment free of obstacles. The calculated parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{w})$ of the GMM were communicated to each robot only at the beginning of the task, then communication was completely denied except for emulating detection capabilities as already mentioned.

The first analysis of results was conducted visually checking the final configuration assumed by the multi-robot system. An example is shown in Fig. 5, where it can be clearly seen that a team of 16 UAVs with sensing range $r = 30$ m successfully reaches the desired final configuration, being placed exactly above streets and trying to maximize the monitored surface. Another interesting result is the total absence of collisions among the agents, proving the effectiveness of the limited Voronoi partitioning method in intrinsically integrating collision avoidance in its trajectory planning strategy. A different scenario instead is shown in Fig. 6, where a team of 20 UAVs was tasked to monitor a larger area of 440×275 m^2. In this case, the area to be monitored is too large for the team, thus a portion of the environment remains uncovered.

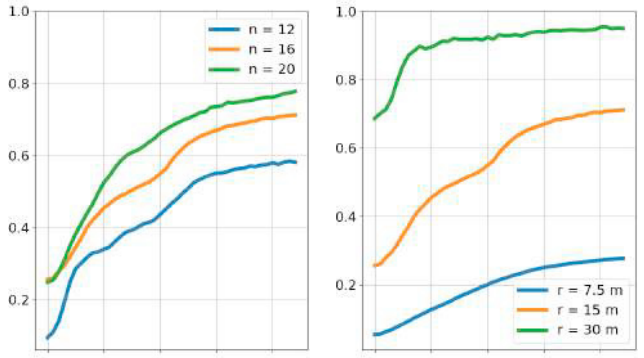
Other than a qualitative analysis of the outcomes, a quantitative examination of results was also carried out, retrieving log data from each simulation run. In details, we evaluated the coverage effectiveness $\eta \in [0, 1]$, calculated as:

$$\eta = \frac{\sum_{i=1}^{n} \int_{V_i} \Phi(q, \mu, \Sigma) dq}{\int_{Q} \Phi(q, \mu, \Sigma) dq}. \tag{7}$$

This metrics evaluates the total probability sensed by the team compared to the sum of the probability over the whole environment (which is always equal to 1). For this reason, the coverage effectiveness is particularly efficient in indicating how the team is performing, quantifying the area of interest the UAVs are able to monitor. The evolution of η over time is shown in Fig. 7 with different number of robots and different sensing range. Results show that the value of η is always increasing, indicating that robots are attracted towards the region of interest as desired. In addition, the final value converges to a maximum for each configuration, depending on the number of robots and their sensing range. As easily predictable, a complete streets surveillance is only possible if a sufficient number of robots is employed, depending on their sensing range and the size and configuration of the considered environment. From the discussed results, we can consider the developed strategy to be successful in addressing the problem of streets surveillance with a team of quadrotors, since robots are driven above streets without any collision with each other.

7. CONCLUSIONS

In this paper we presented a distributed approach to control a team of UAVs to perform a surveillance task. Specifically, the goal of this work was to detect roads from a satellite image and drive the UAVs above them for monitoring. Roads extraction has been possible employing a U-Net, a widely known neural network for image segmentation, which we trained on a dataset of labeled satellite images.



(a) (b)

Fig. 7. Coverage effectiveness: (a) with $r = 15$ m and varying n, (b) with $n = 16$ and varying r.

Then, suitable probability density of the environment was defined, fitting a GMM to the output binary mask in order to emphasize the higher importance of roadways. Finally, robots were controlled with a distributed Voronoi-based strategy driving the aerial robots towards areas of interest without requiring an explicit communication among them. Our methodology was tested choosing an area from Google Maps and creating a realistic 3D environment to run a series of simulations. The UAV team was able to arrange itself above streets trying to maximize the covered surface, and achieved great performances in terms of coverage effectiveness.

Future works will focus on integrating the detection of other features, indicating areas to be avoided by robots. Another interesting improvement would be integrating the U-Net on-board to the UAVs, allowing them to detect streets by themselves instead of only relying on the probability density calculated offline.

REFERENCES

Bai, Y., Asami, K., Svinin, M., and Magid, E. (2020). Cooperative multi-robot control for monitoring an expanding flood area. In *2020 17th International Conference on Ubiquitous Robots (UR)*, 500–505. IEEE.

Catellani, M., Pratissoli, F., Bertoncelli, F., and Sabattini, L. (2022). Coverage control for exploration of unknown non-convex environments with limited range multi-robot systems. In *The 16th International Symposium on Distributed Autonomous Robotic Systems 2022*.

Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2).

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., and Raskar, R. (2018). Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Dorigo, M., Theraulaz, G., and Trianni, V. (2021). Swarm robotics: Past, present, and future [point of view]. *Proceedings of the IEEE*, 109(7), 1152–1165.

Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). Rotors—a modular gazebo mav simulator framework. *Robot Operating System (ROS) The Complete Reference (Volume 1)*, 595–625.

Ju, C., Kim, J., Seol, J., and Son, H.I. (2022). A review on multirobot systems in agriculture. *Computers and Electronics in Agriculture*.

Karlsson, B. (2019). Renderdoc. *URL https://renderdoc.org*.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kotz, S., Balakrishnan, N., and Johnson, N.L. (2004). *Continuous multivariate distributions, Volume 1: Models and applications*, volume 1. John Wiley & Sons.

Lin, L. and Goodrich, M.A. (2014). Hierarchical heuristic search using a gaussian mixture model for uav coverage planning. *IEEE transactions on cybernetics*, 44(12).

Lin, T.Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

McLachlan, G.J., Lee, S.X., and Rathnayake, S.I. (2019). Finite mixture models. *Annual review of statistics and its application*, 6, 355–378.

Pratissoli, F., Capelli, B., and Sabattini, L. (2022). On coverage control for limited range multi-robot systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Queralta, J.P., Taipalmaa, J., Pullinen, B.C., Sarker, V.K., Gia, T.N., Tenhunen, H., Gabbouj, M., Raitoharju, J., and Westerlund, T. (2020). Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 8.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, 240–248. Springer.

Thoduka, S., Gall, J., and Plöger, P.G. (2021). Using visual anomaly detection for task execution monitoring. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4604–4610. IEEE.

Vallejo, D., Remagnino, P., Monekosso, D.N., Jiménez, L., and González, C. (2009). A multi-agent architecture for multi-robot surveillance. In *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems: First International Conference, ICCCI 2009, Wrocław, Poland, October 5-7, 2009. Proceedings 1*, 266–278. Springer.

Yao, P., Zhu, Q., and Zhao, R. (2020). Gaussian mixture model and self-organizing map neural-network-based coverage for target search in curve-shape area. *IEEE Transactions on Cybernetics*, 52(5), 3971–3983.