Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 3314–3319

# Distributed Control for Human-Swarm Interaction In Non-Convex Environments using Gaussian Mixture Models

**Mattia Catellani** * **Eloisa Mazzocco** * **Filippo Bertoncelli** * **Lorenzo Sabattini** *

* *Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy.*
{*mattia.catellani, filippo.bertoncelli, lorenzo.sabattini*}*@unimore.it*

**Abstract:** This paper presents a novel implementation of a human-swarm interface that allows humans to define an area with desired shape to be reached by a multi-robot system. Human-swarm interaction can be useful in order to exploit human intelligence and knowledge for the operation of swarm robots. The proposed work deals with limitations usually met when dealing with real-world implementation, e.g. limited sensing capabilities of the agents and hard conditions where communication is difficult or even completely denied. Gaussian Mixture Models are exploited in order to define an appropriate probability density function of the environment based on the area selected by a human operator. Then, velocity input for each robot is calculated in a distributed manner using Voronoi tessellation and Lloyd's algorithm. Finally, results of both virtual and real-world tests are presented, showing the final configuration reached by the multi-robot system in comparison with the desired region defined on the graphical interface.



*Keywords:* Decentralized control and systems, Human operator support, Autonomous robotic systems, Networked robotic system modeling and control, Multi-vehicle systems

## 1. INTRODUCTION

Multi-robot systems are increasing in popularity and research interest during last years as described by Dorigo et al. (2020), thanks to several advantages they offer in terms of robustness, flexibility and reduction of time needed to accomplish a specific task. Those systems employ multiple robots, often with limited capabilities like in Chand and Carnegie (2013), controlled in order to reach complex behaviours, allowing to achieve better performances with lower economic effort with respect to their single-robot counterpart. Among many applications, robot swarms are nowadays successfully employed in agriculture (see Zhang and Noguchi (2017)), collaborative transportation (see Li et al. (2021)) and space monitoring and exploration, as we already described in our previous work (see Catellani et al. (2022)). An interesting topic in the research field of multi-robot systems is the interaction between them and humans. Hussein and Abbass (2018) show that combining the abilities of humans and robots can lead to higher success rates for trivial operations. This benefit can be gained assigning the role of supervisor to the human, in order to exploit their superior intelligence and external point of view, while robots can concentrate on retrieving data from the environment and accomplish the mission. An interesting approach is proposed by Cheah et al. (2009), where a moving region of specific shape is defined for all the robots to stay inside. Potential energy functions are exploited to calculate the control input for each robot, and the results show great performances in shape control while maintaining a minimum distance between the agents. However, this solution requires a central computer with global knowledge, and only simple regular-shaped regions can be defined with a mathematical equation. Another solution is presented by Li and Liu (2019), where the human is equipped with a haptic device acting as a controller robot, while swarm robots are placed into the environment performing a coverage task. A non-uniform density function represents areas with higher sensing interest, and a goal is defined by the operator manipulating the device. This approach exploits Voronoi tessellation and Lloyd's algorithm (see Cortes et al. (2004)) to obtain the control input and deals with the limited capabilities of the agents, but only allows to define a goal instead of a region with a desired shape, and the implementation is strictly related to the presence of a haptic device as controller robot. Swarm shaping is studied in one of the two approaches proposed by Diaz-Mercado et al. (2015), where Gaussian Mixture Models are exploited to define the region of interest and robots are controlled in a distributed manner following the already mentioned Lloyd's algorithm. However, the agents are supposed to operate in a convex and obstacles-free environment, making this solution unlikely to be applied in real-world operations.

*Contribution* Our solution evolves from the work of Diaz-Mercado et al. (2015) with the aim of overcoming the limitation of assuming the environment to be convex and without obstacles inside. We make use of Gaussian Mixture Models to define a non-uniform density, in order to describe the desired region of interest to be reached by the agents. Then, a limited Voronoi diagram is calculated following the methodology presented by Pratissoli et al. (2022), which makes each robot capable of dealing with unknown and non-convex environments. In addition, it is worth noting that communication among robots is not necessary, while communication with a central unit is only

★ This work was supported by the COLLABORATION Project through the Italian Ministry of Foreign Affairs and International Cooperation.

2405-8963 
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2023.10.1475

needed at the beginning of the mission, where the parameters defining the Gaussian Mixture Model have to be transmitted to the agents. In this way, the human operator can be seen as an external supervisor, possibly with a global view of the environment, while on-field operations are demanded to swarm robots, which can operate even in hard or dangerous conditions. We extensively tested our control strategy in simulations and also with few real-world experiments, where the final configuration of the multi-robot system is compared with the desired shape of the defined region of interest.

## 2. PROBLEM DESCRIPTION

Consider a multi-robot system composed by $n$ robots moving in two dimensions, controlled in order to reach a specific area of the environment. 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, $\forall i = 1, \dots, n$.
We consider the following assumptions:

- *Localization*: each robot is able to localize itself with respect to a global reference frame, which is common for every robot within the team.
- *Limited sensing capabilities*: each robot is able to detect and measure the position of every object (including other robots and environmental boundaries) inside its limited sensing range, defined as a circle with radius $r_{sens} \in \mathbb{R}_{\geq 0}$.

Based on these assumptions, we can formalize the problem addressed in this paper as follows:

*Problem 1.* Implement a human-swarm interface that allows a user to define a specific area of the environment with a desired shape to be reached by a multi-robot system with limited sensing capabilities. The solution must deal with non-convex environments, possibly with obstacles inside.

### 2.1 Proposed Architecture of the Human-Swarm Interface

To solve the mentioned problem, we propose a human-friendly methodology, whose aim is to autonomously interact with swarm robots, while the human operator is only required to draw the desired region to be reached. This solution can be briefly described as a 5-steps procedure:

(1) Human operator draws the region of interest on a graphical interface.
(2) A suitable Gaussian Mixture Model is calculated fitting the desired shape.
(3) Parameters defining the Gaussian Mixture Model are communicated to the agents.
(4) Probability density of the environment is calculated by each robot.
(5) Control action is calculated in a distributed manner.

### 2.2 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 can denote an element of $\mathbb{F}(\mathbb{R}^2)$ as $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, where $\{p_1, \dots, p_n\}$ are points in $\mathbb{R}^2$. We denote, for $p \in \mathbb{R}^2$ and $r \in \mathbb{R}_{>0}$, the closed and open ball in $\mathbb{R}^2$ centered at $p$ with radius $r$ with $\overline{B}(p, r) = \{q \in \mathbb{R}^2 | \|q - p\| \leq r\}$ and $B(p, r) = \{q \in \mathbb{R}^2 | \|q - p\| < r\}$, respectively. 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 $q \in Q$.

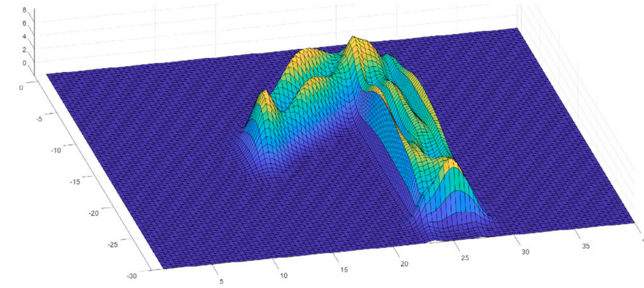

Fig. 1. Probability density defined from the GMM

## 3. GMM-BASED PROBABILITY DENSITY FUNCTION

In this section we analyze how the geometrical shape drawn by the operator is converted into a probability density function of the environment, and how this function is exploited to highlight the desired region of interest. Before going into details, it is necessary to introduce Gaussian Mixture Models, as they will play a key role in our implementation. A Gaussian Mixture Model (GMM) is a multivariate distribution that consists of multivariate Gaussian distribution components, each one defined by a mean point $\mu_i \in \mathbb{R}^2$ and a covariance matrix $\Sigma_i$ (see Kotz et al. (2004)). Given $k$ components, with $k \in \mathbb{R}_{>0}$, the overall model is obtained as the result of their combination following a specific mixture proportion, defined by a vector of weighting factors $\boldsymbol{\omega} = [\omega_1, \dots, \omega_k]^T \in \mathbb{R}^k_{>0}$ related to each component, with $\sum_{i=1}^{k} w_i = 1$.
Given a polygon $S \subset Q$ drawn by the user on a graphical interface, representing the area of interest in the environment, a GMM fitting the desired shape is estimated with a *Maximum Likelihood* method as described by McLachlan et al. (2019). This method uses an *Expectation-Maximization* algorithm to iteratively find the optimal set of parameters $(\mu, \Sigma, \boldsymbol{\omega})$.
Once the Mixture Model has been defined, we can define the probability density function as the sum of the contributions brought by the single components. According to the definition provided by Kotz et al. (2004), the contribution of a single $d$-dimensional component (in our case, $d = 2$), is calculated as

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

From the above equation, the contribution of a single Gaussian component to the global probability density is obtained from the covariance matrix $\Sigma$, defining the spatial distribution around the mean point $\mu$, specifically calculated to fit the drawn polygon. Finally, the overall probability function is obtained as the sum of each component weighted according to the mixture proportion

$$\Phi(q, \mu, \Sigma) = \sum_{i=1}^{k} \omega_i \phi_i (q, \mu, \Sigma). \tag{2}$$

This probability function defines a non-uniform density of the environment, assigning to each point $q$ high probability values if placed inside the region of interest drawn on the graphical interface. An example can be seen in Figure 1, where the region

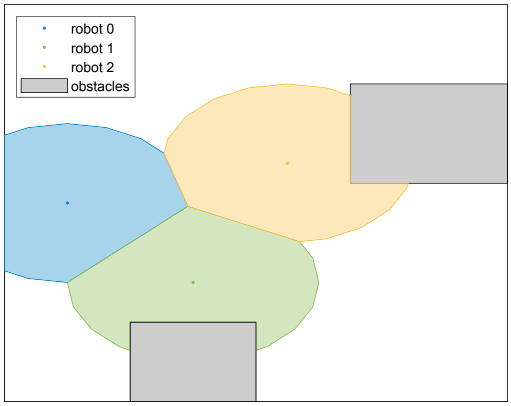


Fig. 2. Limited Voronoi partitioning in the presence of obstacles.

of interest is assigned with much higher probability values than the rest of the environment. In the next section we describe how to take advantage of this probability density function to calculate a proper control action to fulfill the goal of the mission.

## 4. DISTRIBUTED CONTROL ALGORITHM

As previously stated in Section 1, the aim of the proposed work is to tackle limitations usually met in real-world applications. More in details, the proposed solution is expected to work as a distributed methodology to operate in unknown non-convex environments where communication among the agents is denied. Based on the assumptions in Section 2, every robot is capable of detecting and localizing obstacles, environmental borders and other robots falling inside its sensing range $r_{sens}$. For the sake of simplicity, we can denote the surface occupied by obstacles as

$$O = \bigcup_{i=1}^{m} O_i \tag{3}$$

where $m \in \mathbb{R}_{\geq 0}$ is the number of obstacles and $\{O_1, \dots, O_m\} \subset \mathbb{R}^2$ is the set of areas defining the surface occupied by each of them. Thus, we can define a new region $\tilde{\overline{B}}(p_i, r_{sens})$ as the difference between the area $\overline{B}(p_i, r_{sens})$ covered by the agent's sensing range and the surface $O$ occupied by obstacles:

$$\tilde{\overline{B}}(p_i, r_{sens}) = \overline{B}(p_i, r_{sens}) - O. \tag{4}$$

Coordination among the agents is performed with a Voronoi partitioning, consisting in optimally allocating a region of the environment to each robot. Since robots do not have a global knowledge, but are only aware of what is placed inside their sensing range, a *limited* Voronoi partitioning is carried out, according to the definition in Pratissoli et al. (2022):

$$V_i(\mathcal{P}) = \{q \in \tilde{\overline{B}}(p_i, r_{sens}) | \|q - p_i\| \leq \|q - p_j\|, \forall p_j \in \mathcal{P}\}. \tag{5}$$

An example of the obtained partition is depicted in Fig. 2.

Subsequently, the centroid of each Voronoi cell $C_{V_i}$ is calculated taking into account the probability density function $\Phi(q, \mu, \Sigma)$ defined in (2):

$$C_{V_i} = \frac{\int_{V_i} q\Phi(q, \mu, \Sigma)dq}{\int_{V_i} \Phi(q, \mu, \Sigma)dq}. \tag{6}$$

Finally, the desired control input for the $i$-th robot is calculated proportionally to the distance from the centroid of its cell, making it moving towards it according to the law

**Algorithm 1** Best Fitting GMM Calculation Algorithm

**input :** $S, k, \delta$
**output:** $(\mu, \Sigma, \omega)$
**begin**
  Discretize area of $S$.
  $i = 1$.
  **while** $i \leq k$ **do**
    Get GMM with $i$ components using *Expectation-Maximization* algorithm.
    Calculate $BIC(i)$.
    **if** $BIC(i) - BIC(i-1) \geq -\delta$ **then**
      break
    **else**
      $i++$

$$u_i = -k_{prop}\left(p_i - C_{V_i}\right) \tag{7}$$

where $k_{prop} \in \mathbb{R}_{\geq 0}$ is a proportional gain. Such a control law, together with the probability density function defined in (2), leads agents towards the region of interest, which is assigned with a higher probability density. It is worth noting that, according to (5) and (6), no global knowledge is required by the agents to calculate their control action, since both the limited Voronoi partitioning and the centroid can be computed in a distributed manner. Moreover, communication among swarm robots is also not required. As a matter of fact, the only data needed by each robot to calculate its control input are its own position, the neighbors' location and the probability density of the environment, as was extensively tested by Bertoncelli et al. (2022).

However, it must be taken into account that, in certain circumstances, the limited Voronoi region could result in a non-convex region, thus its centroid could be placed outside. This means that the robot could be driven towards an obstacle and crash into it. Therefore, the proposed control strategy does *not* guarantee obstacle avoidance, so a further implementation could be needed to safely operate in real-world applications.

In the following section, an experimental evaluation of the developed solution is presented, describing how this has been implemented and showing the final results obtained in different scenarios.

## 5. EXPERIMENTAL EVALUATION

In this section we will describe how the proposed solution was implemented to be tested on both virtual and real mobile platforms. First of all, the implementation of the graphical interface is presented, in order to explain the transition from a drawn polygon to a specific GMM. The definition of the region of interest through the graphical interface is equally applied to both simulations and real-world tests. Subsequently, experimental evaluation is presented both in simulations and real-world environments, showing the behaviour of the controlled system to reach the region of interest defined through the graphical interface.

### *5.1 Implementation of the Graphical Interface*

The implementation is carried out using MATLAB, and a blank canva is presented to the operator, who is expected to draw a polygon $S \subset \mathbb{R}^2$ moving the mouse cursor on it and to decide the maximum number of components $k \in \mathbb{R}_{>0}$ to be generated. Subsequently, the best fitting Gaussian Mixture

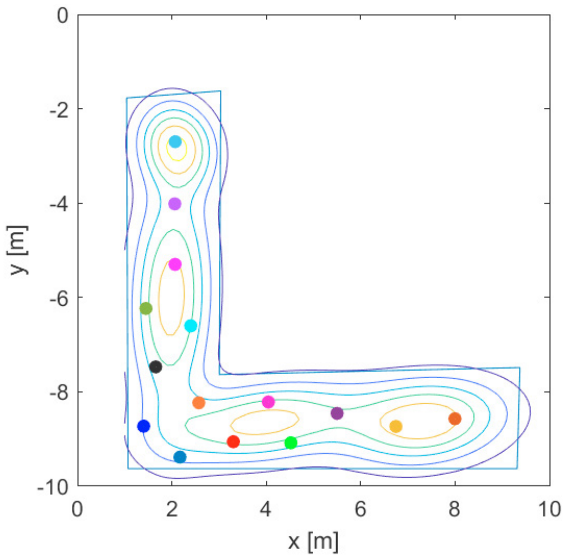


Fig. 3. Simulation with 15 networking nodes acting as robots

Model is calculated following Algorithm 1: a discretization of the area is performed and a suitable Mixture Model is iteratively calculated with an increasing number of components according to the *Expectation-Maximization* algorithm proposed by McLachlan et al. (2019). Then, the Bayesian Information Criterion (BIC) is calculated according to Schwarz (1978): the lower is the BIC, the better the model is fitting the region. The process continues until convergence is achieved, or the maximum number of components $k$ is reached. It is interesting to note that the proposed algorithm iteratively searches for the optimal number of components fitting the desired shape, in order to have a low number of parameters to be stored and computationally efficient calculations to be carried out by the swarm robots during the mission. The optimal set of parameters $(\mu, \Sigma, \omega)$ defining the GMM calculated with Algorithm 1 is stored to be communicated to the robotic agents.

### 5.2 Virtual Tests

MATLAB has also been used for a first implementation of the algorithm, where swarm robots were approximated to networking nodes moving in the environment as single integrator systems. This first set of tests has shown that the multi-robot system behaved as expected, as we can see from Fig. 3 where the final configuration of the network perfectly fitted the desired shape of the region of interest. Moreover, the low computational effort required for those simulations allowed to employ a large number of robots, thanks to the absence of a physical engine modeling interactions of robots with each other and with the environment. Subsequently, trials were performed employing mobile platforms and using ROS2 as a control architecture. The behaviour of the controlled multi-robot system was extensively tested with simulations carried out within the Gazebo physical engine employing virtual models of the TurtleBot3 Burger robot. Robots were controlled in a distributed manner, and the set of parameters defining the Gaussian Mixture Model was communicated to each one at the beginning of the mission using a custom ROS message, directly sent over the ROS network from the MATLAB implementation of the graphical user interface. An example of final configuration reached by the multi-robot system can be seen in Fig. 4, where the probability density shown in Fig. 1 was exploited to define the region of interest.

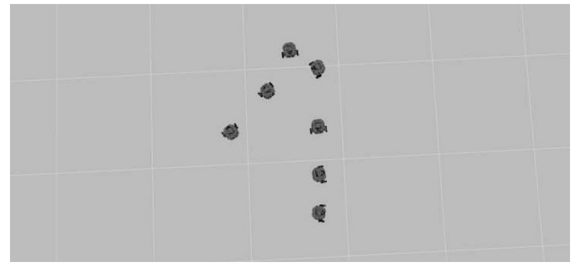

Fig. 4. Simulation within a virtual environment

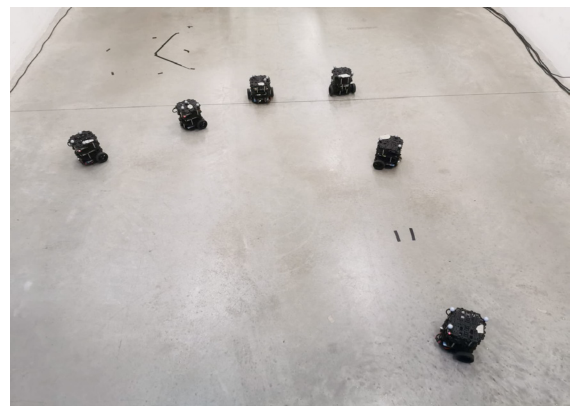

Fig. 5. Real-world implementation

### 5.3 Real-world Experiments

A further step in testing the developed solution has been made with real-world trials, carried out employing the real counterpart of the TurtleBot3 Burger platforms used for simulations. As for virtual trials, the set of parameters defining the GMM was communicated to the agents only at the beginning of the mission over a ROS network. This kind of tests were performed with randomly chosen starting positions of the agents within a $4.5 \times 3.5\ m^2$ environment, and the Optitrack motion capture system was exploited to obtain localization of the robots with respect to a global reference frame. It is important to note that communication with a central unit has been exploited by each robot only to gain information on its global position and the relative position of its neighbors, in order to emulate localization capabilities as described in the assumptions in Section 2. Initially, results were investigated with the same probability density that was shown in Fig. 1, in order to have a comparison with the behaviour obtained in simulations where the final configuration in Fig. 4 was reached. Swarm robots ended up reaching the desired region of interest as shown in Fig. 5, demonstrating that the presented approach ensures good performances in swarm shaping. Because of the limited size of the area at our disposal for the execution of on-field tests, only simple scenarios were set up and no obstacles were placed in the environment. Several heterogeneous regions of interest were tried out, and the multi-robot system always displayed a performing behaviour in reaching a final configuration fitting the desired shape. A further example is shown in Fig. 6, where the entire workflow of the presented methodology is displayed, from the definition of the region of interest, to the generation of a Gaussian Mixture Model fitting the desired shape, and finally to the actuation of robotic agents. As one would expect, the higher is the number of robots employed in the mission, the better they will fit the region of interest.

Finally, we conducted a further set of virtual trials, focusing on

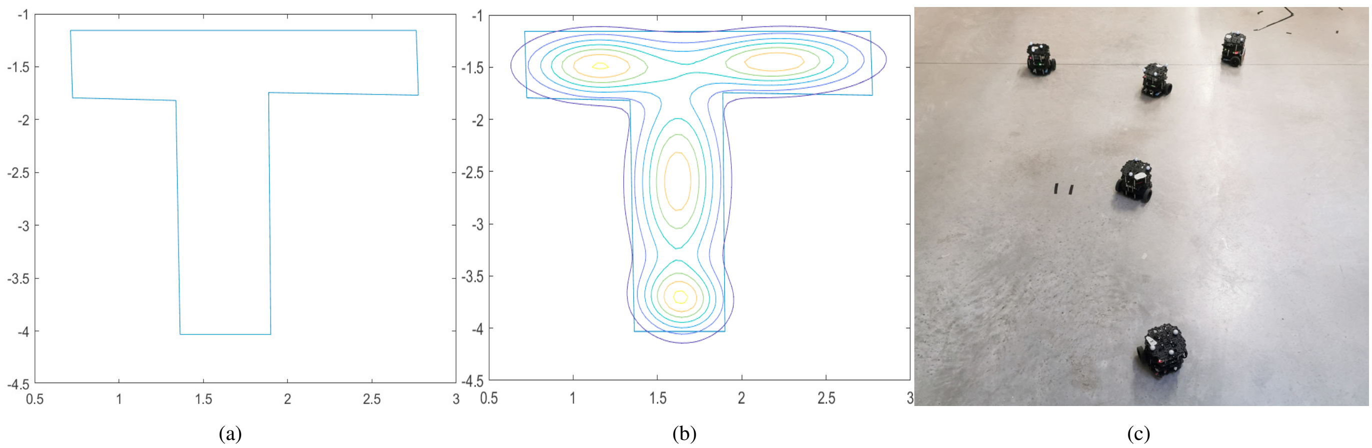


Fig. 6. Complete workflow of the proposed methodology: (a) the region of interest is drawn on a graphical interface, (b) a GMM is calculated fitting the desired shape, (c) the multi-robot system is actuated and reaches the desired configuration

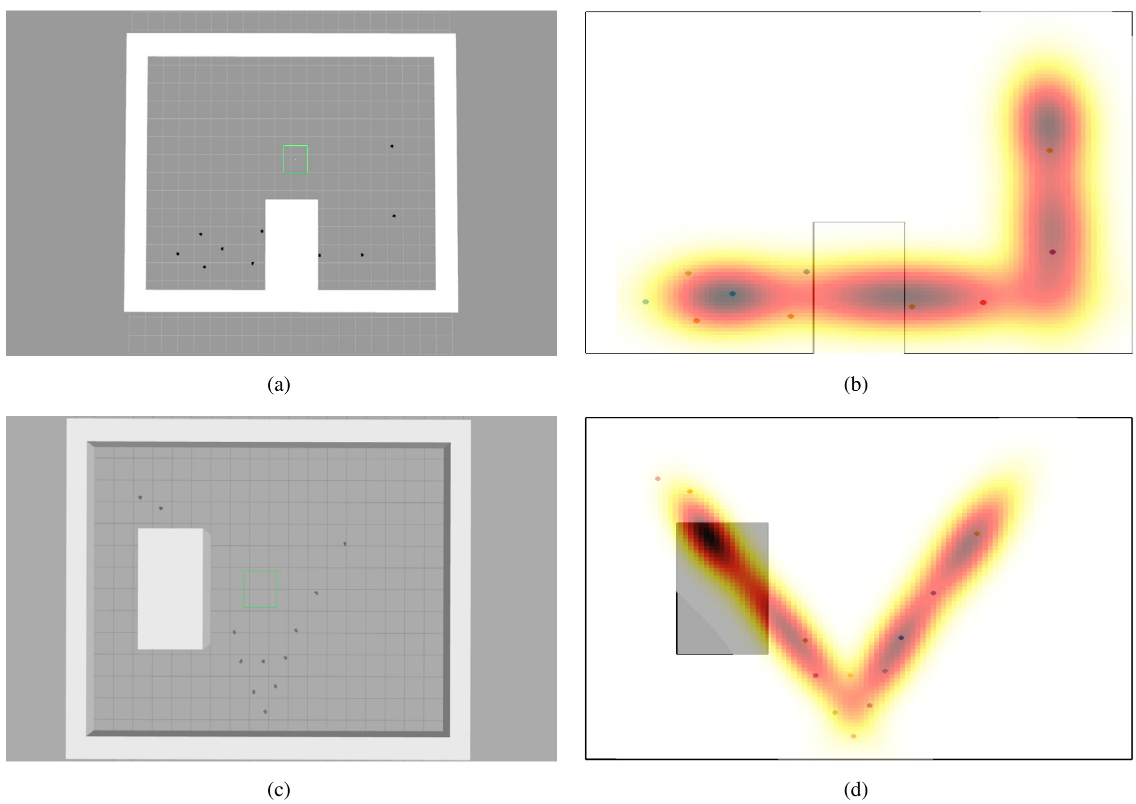


Fig. 7. Multi-robot system dealing with obstacles in the environment. (a), (c): The final configuration is reached and the obstacle is avoided. (b), (d): Comparison between the final configuration and the probability density representing the region of interest.

the behaviour of the controlled multi-robot system within a non-convex environment. As we mentioned in the above sections, the ability to deal with obstacles and cluttered environments is a fundamental feature for robot swarms to be employed in real-world operations. Several virtual environments were prepared with randomly positioned obstacles of different regular shapes, in order to lower the chance of generating non-convex limited Voronoi regions that could make robots crash. Results have shown that the multi-robot system behaves as expected, avoiding obstacles while navigating and reaching the final desired shape only covering areas within the region of interest that do not contain obstacles. Examples are shown in Fig. 7, where swarm robots sense the presence of obstacles preventing them to reach certain areas, so they rearrange themselves to cover obstacles-free portions of the region of interest. The area covered by the multi-robot system has been evaluated and compared with the overall area enclosed by the region of interest as shown in Fig. 8, resulting in a complete sensing of the desired region.

In conclusion, we can say that those trials validated the proposed methodology, so it can be stated that the presented control architecture ensures great performances in covering a specific area with desired shape using a multi-robot system, even when dealing with non-convex environments or obstacles.

## 6. CONCLUSION

In this paper, a human-swarm interaction methodology was presented, whose aim is to allow a human operator to define

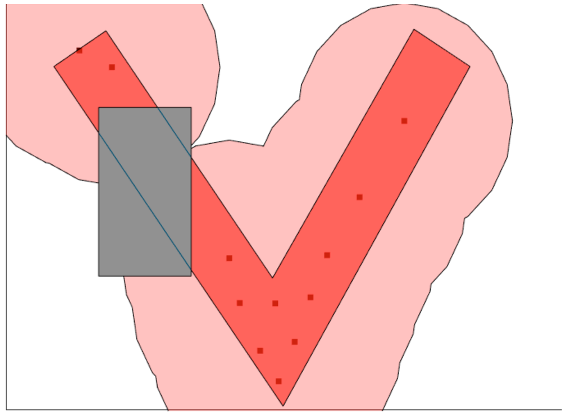

Fig. 8. Comparison between total area of the desired region and area covered by swarm robots with 3.5 $m$ sensing range

a region of interest with a desired geometrical shape to be explored by the agents in the environment. With this solution, the operator plays the role of a supervisor with a global knowledge of the environment, while on-field execution of the mission is demanded to robotic agents. Hence, safety conditions are guaranteed to the human being, who is not required for an in-person visit in a possibly dangerous environment. This strategy exploits the definition of Gaussian Mixture Models from a polygon drawn on a graphical interface to define a non-uniform density function, where higher importance is given to the region of interest, together with a limited Voronoi partitioning of the environment. This approach sets up a distributed strategy in order to deal with limitations usually met in real-world implementations, e.g. limited sensing capabilities of the agents and communication impossibility. It is important to note that the definition in (4) only considers the difference between free and occupied surfaces, but it does not take into account blind spots generated using a sensor on a real robot. In particular, the real field of view of a ground vehicle will be slightly different from the one calculated with this assumption, while no differences are met when employing aerial vehicles observing from above. However, the mentioned assumption does not prevent from using this methodology in real-world applications, since the absence of blind spots in the sensing range can only positively affect the overall behaviour of the controlled system. Several tests have been made both in simulation and on real mobile platforms, and the results show that the multi-robot system behaves as expected and reaches a final configuration fitting the desired shape of the region of interest. At the moment, only regular environments have been tested, with simple-shaped obstacles placed inside of them only in simulations, therefore robots have been able to avoid collisions.

Future work will extend on-field trials to more complex scenarios, where agents are required to operate in larger areas with obstacles placed into the environment. Furthermore, a necessary step could be the integration of an inner control layer to always guarantee obstacle avoidance, in order to allow for a real-world application of the proposed control strategy. An interesting approach could exploit Control Barrier Functions to define a minimum distance to be maintained between robots and obstacles (see Ferraguti et al. (2022)), calculating the optimal control action compliant with this constraint from the desired one obtained with the methodology presented in this paper. Finally, another interesting enhancement of the proposed architecture could exploit a previously taken picture of the operation area, allowing its integration into the graphical interface. In this way, the operator will be able to interactively draw the region of interest in order to precisely fit a specific area of the environment. The image could also be elaborated with the aim of finding specific features, from which the region of interest can be automatically generated. An example of a scenario where this solution could be exploited is a search and rescue mission taken in a urban environment with a group of UAVs, where buildings must be treated as obstacles and a specific area must be reached by the agents, where a certain target feature is located.

## REFERENCES


Bertoncelli, F., Belal, M., Albani, D., Pratissoli, F., and Sabattini, L. (2022). On limited-range coverage control for large-scale teams of aerial drones: Deployment and study. In *The 16th International Symposium on Distributed Autonomous Robotic Systems 2022*.

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*.

Chand, P. and Carnegie, D.A. (2013). Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots. *Robotics and Autonomous Systems*, 61(6), 565–579.

Cheah, C.C., Hou, S.P., and Slotine, J.J.E. (2009). Region-based shape control for a swarm of robots. *Automatica*, 45(10).

Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2), 243–255.

Diaz-Mercado, Y., Lee, S.G., and Egerstedt, M. (2015). Distributed dynamic density coverage for human-swarm interactions. In *2015 American control conference (ACC)*, 353–358.

Dorigo, M., Theraulaz, G., and Trianni, V. (2020). Reflections on the future of swarm robotics. *Science Robotics*, 5(49).

Ferraguti, F., Landi, C.T., Singletary, A., Lin, H.C., Ames, A., Secchi, C., and Bonfè, M. (2022). Safety and efficiency in robotics: The control barrier functions approach. *IEEE Robotics & Automation Magazine*, 29(3), 139–151.

Hussein, A. and Abbass, H. (2018). Mixed initiative systems for human-swarm interaction: Opportunities and challenges. In *2018 2nd Annual Systems Modelling Conference (SMC)*.

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

Li, G., Ge, R., and Loianno, G. (2021). Cooperative transportation of cable suspended payloads with mavs using monocular vision and inertial sensing. *IEEE Robotics and Automation Letters*, 6(3), 5316–5323.

Li, W.T. and Liu, Y.C. (2019). Human-swarm collaboration with coverage control under nonidentical and limited sensory ranges. *Journal of the Franklin Institute*, 356(16), 9122–9151.

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*.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464.

Zhang, C. and Noguchi, N. (2017). Development of a multi-robot tractor system for agriculture field work. *Computers and Electronics in Agriculture*, 142, 79–90.