

A Dynamic Weighted Area Assignment Based on a Particle Filter for Active Cooperative Perception

José J. Acevedo , João Messias , Jesús Capitán , Rodrigo Ventura , *Member, IEEE*, Luis Merino , *Member, IEEE*, and Pedro U. Lima , *Senior Member, IEEE*

Abstract—This article addresses an Active cooperative perception problem for networked robots systems. Given a team of networked robots, the goal is finding a target using their inherent uncertain sensor data. The article proposes a particle filter to model the probability distribution of the position of the target, which is updated using detection measurements from all robots. Then, an information-theoretic approach based on the RRT* algorithm is used to determine the optimal robots trajectories that maximize the information gain while surveying the map. Finally, a dynamic area weighted allocation approach based on particle distribution and coordination variables is proposed to coordinate the networked robots in order to cooperate efficiently in this active perception problem. Simulated and real experimental results are provided to analyze, evaluate and validate the proposed approach.

Index Terms—Active cooperative perception, coordination variables, networked robots system, particle filter.

I. INTRODUCTION

THE focus of the MONarCH (Multi-Robot Cognitive Systems Operating in Hospitals) project is on social assistive robotics using networked heterogeneous robots and sensors to interact with children, staff, and visitors, engaging in edutainment activities in the pediatric infirmary at the Portuguese Oncology Institute at Lisbon (IPOL), Portugal. The task on *active perception* in Networked Robots Systems (NRS) is an specially relevant task in the project, because social robots in NRS require as much information as possible about the environment and the people around to decide the most suitable behaviors to execute (for instance, to find a lost or hidden child). The idea is to control a set of Perception-Oriented robots (PO Mbots, see Fig. 1) to



Fig. 1. The PO Mbot designed for the MONarCH project.

gain information (based on their on-board sensors) that may be required by other agents in the system (the medical staff or other robots).

In particular, this paper addresses the problem of searching a target (for instance, a child) along the different rooms of a hospital. The proposed method tries to solve the multi-robot, active perception problem until the child is first detected and found. The target is assumed to be dynamic but with slower dynamics than the robots. This means that the paper does not solve a pursuit-evasion problem, where the target is hiding. Instead of moving continuously, the child is considered to be staying at a certain room, which is unknown for the robots. Nonetheless, that child may still change to another room throughout the search, and the NRS should adapt to that situation in order to find her/him eventually.

This problem is challenging, as multiple robots have to coordinate in an efficient and coherent manner to find the target. Also, they only count on limited sensor capabilities (e.g., RFID readers). Negative measurements will be obtained most of the time (robots explore until they find the target), which precludes the NRS from using parametric probabilistic distributions. Therefore, this paper proposes a Particle Filter to model the belief on the target location. Then, an information-gain metric based on the entropy associated with the particle set is used to generate paths for the robots, using a fixed-depth RRT* planner. Last, it is proposed a method based on coordination variables to

Manuscript received September 10, 2019; accepted December 29, 2019. Date of publication January 13, 2020; date of current version January 22, 2020. This letter was recommended for publication by Associate Editor G. Hollinger and Editor N. Y. Chong upon evaluation of the reviewers' comments. This work was supported by the MONarCH European under Project FP7-ICT-2011-9-601033. The work of R. Ventura and P. U. Lima was supported by the strategic funding LARSyS under Project FCT [UID/EEA/50009/2013]. The work of L. Merino was supported by the Spanish Ministry of Science, Innovation and Universities under Projects COMCISE RTI2018-100847-B-C22, MCIU/AEI/FEDER, UE. (*Corresponding author: José J. Acevedo.*)

J. J. Acevedo and J. Capitán are with the Robotics, Vision and Control Group, Universidad de Sevilla, 41092 Sevilla, Spain (e-mail: jacevedo@us.us.es; jcapitan@us.es).

J. Messias is with Latent Logic, OX1 1JD Oxford, U.K. (e-mail: joaovmessias@gmail.com).

R. Ventura and P. U. Lima are with the Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal (e-mail: rodrigo.ventura@isr.tecnico.ulisboa.pt; pedro.lima@tecnico.ulisboa.pt).

L. Merino is with the Service Robotics Lab, Universidad Pablo de Olavide, 41013 Sevilla, Spain (e-mail: lmercab@upo.es).

Digital Object Identifier 10.1109/LRA.2020.2965876

coordinate the NRS while searching and increase the efficiency in the task. This method is distributed and scalable, allowing the multiple robots to converge to a common solution (coherent particle assignment), but at the same time in an efficient manner (i.e., with low computational load).

The authors presented results on active perception for the same application with a single robot in [1]. However, this paper addresses *Active Cooperative Perception* (ACP). That is, the problem in which several robots actively decide their motions, taking into account their effects on their sensors, in order to cooperate to maximize the information related to some features perceived from the environment.

First formulations of active perception problem [2] were looking to improve robot localization by controlling its own motion. In [3], a receding-horizon based algorithm is proposed to solve the active perception problem applied to range sensing tasks. More recently, the active perception problem has been addressed for exploration tasks based on utility functions [4].

A multi-robot exploration task is addressed as an ACP in [5] and solved based on Gaussian Processes to model a spatially distributed static process, RRT algorithms to plan the paths and max-sum algorithms for the distributed cooperation. However, this model is not suited to track dynamic targets. In [6], authors model the ACP problem as auctioned Partially Observable Markov Decision Processes (POMDPs) for cooperative tracking applications. Cooperative target-tracking applications under uncertainty have also been recently addressed as an ACP and solved using an algorithm based on a decentralized Model Predictive Controller (MPC) [7]. Some other target-tracking applications have been addressed as an ACP problem [8]–[10]. They formulate the problem from different approaches, but the most accurate ACP algorithms depend on a thoroughly studied and complex model, only applicable to concrete small-scale scenarios.

Some other problems related to cooperative perception, such as cooperative surveillance or cooperative patrolling have been solved based on area allocation [11], [12]. In any case, the coordination of multiple robots becomes more challenging as the number of robots increases. The algorithms based on *coordination variables* have been successfully applied to similar allocation problems obtaining distributed and very scalable solutions [12], [13]. Moreover, in [14], authors prove that the algorithms based on coordination variables can achieve the desired solution for distributed coordination problems.

The remainder of the paper is as follows: Section II defines formally the ACP problem addressed in this paper; the proposed approach is developed in Section III; the architecture of the NRS for ACP is defined in Section IV; Section V provides simulations and real experimental results to validate the proposed approach; and Section VI concludes the paper.

II. PROBLEM STATEMENT

Given a team of m networked mobile robots equipped with the appropriate sensors, it is required to plan their motion to look for a target moving into a defined space $S \in \mathbb{R}^2$. Therefore, the objective is planning cooperatively the paths of the robots to

maximize the certainty on the position target in the future, given the received measurements. The position of the target at time t is defined by p_t .

Each robot r_i receives the following inputs:

- 1) Its current pose at time t , $q_{i,t} = [x_{i,t} \ y_{i,t} \ \theta_{i,t}]^T$.
- 2) The most recent reading of its sensor at time t , $z_{i,t} \in \{False, True\}$, which defines the positive (*True*) or negative (*False*) detection of the searched target.
- 3) The probabilistic model of the sensor able to detect the target, referred to the robot pose. This description may be non-parametric.
- 4) Optionally, the initial probability distribution over the position of the target. It may be non-parametric. If this distribution is not provided to the robot, it can assume a uniform distribution through the whole map.

From the information provided by the robots (both their poses and their sensor's readings), the task of finding the position of the target in an arbitrary environment may be formally described as the problem of maximizing a defined metric of confidence over the probability of the target's position, i.e.,

$$\Pr(p_t | \mathbf{q}_{1,0:t}, \mathbf{z}_{1,0:t}, \mathbf{q}_{2,0:t}, \mathbf{z}_{2,0:t} \dots, \mathbf{q}_{m,0:t}, \mathbf{z}_{m,0:t}),$$

where $\mathbf{q}_{i,0:t} = \langle q_{i,0}, q_{i,1}, \dots, q_{i,t} \rangle$ and $\mathbf{z}_{i,0:t} = \langle z_{i,0}, z_{i,1}, \dots, z_{i,t} \rangle$ represent the history of poses and the sensor readings of the i -th robot from time 0 to t . Unlike other typical estimation problems, a key issue here is that most of the readings provided by the robots will be negative readings (target not detected) during the exploration. Moreover, for this application of service robots working in a hospital, it is reasonable to believe that the robots will have access to a reliable map of the scenario and will be able to localize themselves with a degree of uncertainty that is negligible with respect to the uncertainty in the target (child) position.

III. PROPOSED APPROACH

This paper proposes decoupling the ACP problem into two different, but inter-connected, tasks: estimating the target position based on the previous motion and sensor readings from the robots; and planning the motion of the robots in order to improve the estimation of the target position. Each of these tasks will be addressed in the following sub-sections.

A. A Particle Filter to Model the Target Position

As most of the readings during the exploration process will be negative, the target position p_t may be difficult to approximate using a parametric model. In this paper, a Particle Filter (PF) is proposed to model the uncertain position of the target, where the particle set is the set of possible solutions (particle positions p_j) with an associated importance (particle weights w_j). The advantages of a PF to model the target position in a similar problem were described in [1].

A PF algorithm is implemented taking into account that the particle set B has to be updated based on the readings of all the robots. See [15] for more details and background on PFs. Some details about the specific implementation in this paper are the following:

1) *Initialization Stage*: If the initial set of particles with their respective weights is not provided as input to the algorithm, an initialization stage is required. It creates an uniform distributed set B of N particles, sampling N valid positions $p_j \in S, \forall j = 1, \dots, N$ from the environment and assigning the same normalized weight to each of the particles, $w_j \leftarrow 1/N, \forall j = 1, \dots, N$.

2) *Prediction Stage*: The current position for each particle is predicted, at each iteration, based on its previous position and the motion model of the target. In this paper, a random walk is assumed as motion model based on a normal distribution centered in the previous particle with a standard deviation σ . However, nothing precludes the method from using more complex prediction models for the particles.

3) *Update Stage*: Based on the readings from the robots, the particle weights are updated applying the Bayes' rule based on the sensor model and the sensor readings. Note that this process considers readings from all the robots.

4) *Re-Sampling Stage*: At each iteration, this stage creates a new particle set B from the old one. Basically, this process removes particles with low weight and replaces them duplicating the ones with high weights. The final particle set will consist of N particles with the same weight $1/N$.

B. An Entropy-Based Path Planning for Information Gain

Given the aforementioned PF to model the position of the target, the objective of maximizing the confidence over the position of the target p_t is equivalent to minimizing the entropy of the particle set B . So, the task of planning the motion of the i -th robot to improve the estimation of the target position may be formally described as the problem of finding a sequence of its K following poses $\hat{\mathbf{q}}_{t+1:t+1+K}^*$, for some $K > 0$, that minimizes the entropy H of the predicted particle set:

$$H(\Pr(p_{t+1+K} | \mathbf{q}_{1,0:t}, \mathbf{z}_{1,0:t}, \dots, \mathbf{q}_{m,0:t}, \mathbf{z}_{m,0:t}, \hat{\mathbf{q}}_{t+1:t+1+K})) \quad (1)$$

Rapidly-exploring Random Trees (*RRT*) [16] are well-suited strategies to this problem. They are computationally efficient *anytime* algorithms. Moreover, some variants like *RRT** [17] are optimal and can be adapted to minimize the entropy of a particle set. However, instead of looking for a defined goal position, the objective here is to minimize the predicted entropy (1). In [10], authors propose a similar approach and formulate information-gathering *RRT* variants, but assuming some energy and time constraints not necessary for ACP.

In this work, a fixed-depth *RRT** is proposed. The process is as follows: generating a fixed-depth tree, evaluating the cost (entropy of the “predicted” particle set) at each node of the generated tree and generating a path with the minimum accumulated cost from the robot pose $q_{i,t}$. Then, as the PF provides a equally-weighted particle set, this method will tend to generate paths that cross zones with high density of particles and avoid zones without particles.

Calculating the entropy of a probability density function (in this case, the particle set) may be computationally too expensive. However, there are approximations [18], [19] that are computationally efficient for this type of distributions. Concretely, this

method uses the approximation based on the entropy H of a Gaussian Mixture defined in [19]:

$$H \leq \sum_{j=1}^N w_j (-\log(w_j) + 0.5 \log(2\pi e)^2 \sigma^4) \quad (2)$$

where w_j is the weight of the j -th particle of the PF, and σ is the standard deviation for the target position.

Even with this approximation (2), the method should store a version of the particle set at each possible future pose (node) while expanding the tree, in order to calculate correctly the predicted entropy. These sets are computed from the particle set in the parent node, applying the different stages of the PF. However, assuming that the robots move much faster than the target, the motion of the particles may be ignored. Thus, it is not required to store the particle positions (only their weights) nor compute the prediction and re-sampling stages. Note that motion models could still be used and prediction and re-sampling steps be included when building the planning RRT, but at the expense of more memory. Moreover, in theory, each node should consider two possible readings from the sensor (negative and positive) when branching, so the amount of particle sets to store would increase exponentially. However, since the problem formulated in this paper focuses on the search part, and not on tracking the target actively, it is reasonable to consider that most observations will be negative, as positive observations would imply the completion of the task. Therefore, only negative measurements are considered when computing the planning tree. Nonetheless, no assumptions are made about the sensor model, which may include false positive and false negative rates.

C. A Dynamic Weighted Area Allocation to Coordinate Multiple Robots

The previous subsection proposes a method to plan the motion of a robot individually, without taking into account the plan of the rest of the robots. If all the robots receive the same target estimation (defined by a PF which takes into account readings from all robots), as they are not coordinated, they may plan to move towards the same positions. It is easy to deduce from (2) that this would be a non-efficient solution (because it does not minimize the entropy).

A straightforward manner to solve this problem is to generate the robot motion plans in a centralized and ordered manner. The first plan would be generated in an independent manner, but the following plans would be based on the predicted particle sets left by the previous plans. However, this solution would imply the synchronization of the robots' motion, as well as delays and high computational costs that increase with the number of robots.

Therefore, this paper proposes an area division approach to coordinate the robots from in a distributed manner with a minimum information interchange among them. The idea is to divide the whole environment into as many areas as robots, such as each robot can generate its motion plans to search the target in its assigned area. It ensures that two robots do not move through the same zone, “cleaning” the same particles. Concretely, the proposed approach considers the particle set B received from the PF to generate dynamically equally-weighted

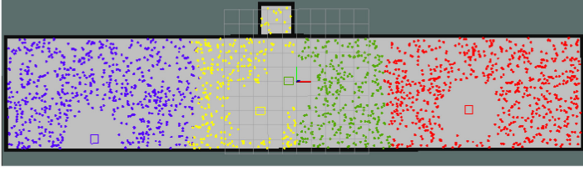


Fig. 2. Snapshot from a simulation with 4 robots, where the Dynamic Weighted Area Allocation has been used. It shows the particle subsets (defined with different colors) assigned to each robot. A video from this simulation can be viewed in <https://youtu.be/qrt5xPvjVwg>

areas to be allocated among the robots (*Dynamic Weighted Area Allocation*). As the particle set B is updated by the PF, this method updates the set of particles assigned to each robot.

Distributed particles allocator: An allocator method is proposed here to assign the particles from a provided set, among the robots. The proposed method does not assign directly an area to each robot r_i , but a subset of particles B_i . However, as the entropy-based motion planning presented in Section III-B generates paths tending to avoid zones without particles, the robot will move only through the area where its assigned particles are. Therefore, implicitly, this method assigns an area to each robot.

The method is distributed and based on coordination variables [20]. The variables are the particle set B and the amount of robots in the system m . If all robots apply the method using the same coordination variables as input, non-overlapping (disjoint) particle subsets are obtained in a distributed manner, see Fig. 2. Note that all robots reach a common solution without explicit negotiation among them.

This method makes two assumptions. First, it assumes that the robots are ordered by an index i . Second, two positions, p_{ini} and p_{end} , are previously defined in the map to be used as references to group and divide the particles among the robots. They may be defined arbitrarily, but they have to be common for all the robots.

Algorithm 1 describes the method. It runs as follows. Given the robot index i and the amount of robots m , each robot knows how many robots has on its “left” m_{left} and on its “right” m_{right} (being “left” and “right” theoretic references depending on the robot indexes and not, necessarily, physical locations). Then, the `divide` method is applied twice. The first time, it divides the particle set B provided by the PF into two subsets: one for the robots on the right of the i -th robot and another for the robot itself and the robots on its left. This last subset is noted as B_{aux} , being w_j^{aux} its particle weights. The second time, it divides the particle set B_{aux} into two subsets: one for the robots on the left of the i -th robot and another for the robot itself, which will be the particle set returned as output.

Algorithm 2 implements the `divide` method. It receives a particle set B' , two positions (p_1 and p_2) and two desired weights (w^1 and w^2). Two empty particle subsets B_1 and B_2 are created and associated with the received positions and desired weights. Each particle in B' is initially assigned to the subset whose position is nearer. If, after assigning a particle, the sum of weights of the particles assigned to that subset is higher than its desired weight, the farthest particle from this set is taken out and assigned to the other subset. It returns the first subset B_1 as output.

Algorithm 1: Distributed Particles Allocator.

Require: B : Set of N particles defined by particle positions $\{p_j\}$ and weights $\{w_j\}$

Require: m : Number of robots

Require: i : Robot’s index

Require: p_{ini} and p_{end} : initial and end positions.

$$m_{left} \leftarrow i - 1$$

$$m_{right} \leftarrow m - i$$

$$w^1 \leftarrow \frac{m_{left} + 1}{m}$$

$$w^2 \leftarrow \frac{m_{right}}{m}$$

$$B_{aux} \leftarrow \text{divide}(B, w^1, w^2, p_{ini}, p_{end})$$

$$w^1 \leftarrow \frac{m_{left}}{m} \sum_{j=1}^N w_j^{aux}$$

$$w^2 \leftarrow \frac{1}{m} \sum_{j=1}^N w_j^{aux}$$

$$B_i \leftarrow \text{divide}(B_{aux}, w^2, w^1, p_{end}, p_{ini})$$

return B_i : Set of particles assigned to the i -th robot

Algorithm 2: Method to Divide a Particle Set Into Two Weighted Subsets, Being $i, j = 1, \dots, \text{size}(B')$ and $k, l = 1, 2$ With $k \neq l$.

Require: B' : Set of particles $\{part_j\}$ to be divided. It is defined by particle positions $\{p'_j\}$ and weights $\{w'_j\}$

Require: w^1 and w^2 : Desired sum of weights that should have the generated subsets B_1 and B_2 , respectively.

Require: p_1 and p_2 : positions around which the particles from B_1 and B_2 have to be grouped, respectively.

- Create the empty sets B_1 and B_2

- Initialize the weights assigned to each side w_c^1 and w_c^2 to zero

for all $part_i \in B'$ **do**

- Assign $part_i$ to the nearest subset B_k , depending on

p_k

- Update the weight of the subset w_c^k

if $w_c^k > w^k$ **then**

- Take out the farthest particle $part_j$ to p_k from B_k

- Assign $part_j$ to the other set B_l

- Update the weights w_c^k and w_c^l

end if

end for

return B_1

This distributed algorithm is fast convergent. Also, its complexity cost does not depends on the number of robots m , but on the number of particles N (which could be configured depending on the requirements). The `divide` method runs at most $N + \frac{N}{2} = \frac{3N}{2}$ particles assignation or re-assignation (in the worst case scenario, when all the particles are nearer to the same set position); and Algorithm 1 executes at most twice the `divide` method. Then, the algorithm runs at most $3N$ iterations, even less because the second method execution does not use the whole set of N particles but a subset. Furthermore, although the PF updates the set of particles continuously, the particles allocation process has not to be made continuously but only when a new plan has to be generated by the entropy-based motion planning.

D. Discussion

The method for particle assignment ensures a distributed, coherent allocation, since all robots use the same *coordination variables*. This means that no particle is doubly assigned nor remains unassigned. Algorithm 1 is consistent by definition: given m robots and N particles, each robot will assign N/m particles to itself. Regarding the `divide` method, a simple example may illustrate how it runs and how the overlapping and non-assignment issues are avoided. Consider a simple scenario with 3 particles and 2 robots. Each particle will have a weight of 0.333 and each robot should assume weights $w^1 = w^2 = 0.5$. Assume that the three particles are closer to the first position reference p_1 than to the second reference p_2 . The first particle $part_1$, then, is assigned to the first subset of particles B_1 . The second particle $part_2$ is also assigned to B_1 , but as $w_c^1 = 0.66 > w^1 = 0.5$, the farthest particle from p_1 (e.g., $part_1$) is removed from B_1 and added to B_2 . The same happens with the third particle $part_3$. Finally, $B_1 = \{part_2\}$ and $B_2 = \{part_1, part_3\}$. From the other robot point of view, the procedure is similar but assuming opposite inputs. So, the obtained set of particles is $B_1 = \{part_1, part_3\}$ and $B_1 = \{part_2\}$, which is consistent.

Algorithm 1 considers two consecutive divisions to remove particles assigned to the other robots to the right and to the left (being “left” and “right” theoretic references depending on the robot indexes and not, necessarily, physical locations). This is justified because the procedure is performed in a distributed manner but it should still be coherent (without overlapping particles). Each division considers just two sides and two references positions. Performing the particle assignment with a single division procedure between m robots would require of m fixed reference positions to obtain a coherent particle allocation in a distributed manner.

Another aspect for discussion is the criterion to assign particles to each subset. Algorithm 2 does not impose a specific type of distance. The experiments in this paper consider Euclidean distance, as it was enough to achieve a good performance. Nonetheless, alternative distances may be used without problem. In complex scenarios, using Euclidean distances may lead to assignments where the particles are not easily reachable by the robots. However, since our particle assignments are dynamic, this situation, even inefficient, would be temporal and the system would adapt eventually to explore the whole scenario.

IV. MULTI-ROBOT SYSTEM ARCHITECTURE

Figure 3 summarizes the multi-robot system architecture. There is a common PF module. This PF receives poses and sensor readings from all the robots and provides continuously a probability distribution about the target position. The system could also be modified to implement distributed PF modules for each robot, but it should be ensured that they all converge to the same particle set.

Note that, although the motion planning module of each robot receives only a subset of particles from its distributed allocator, the robot reading and pose are used by the PF module to update the whole set of particles. This means that, if a robot crosses a zone with particles assigned to another robot while is moving

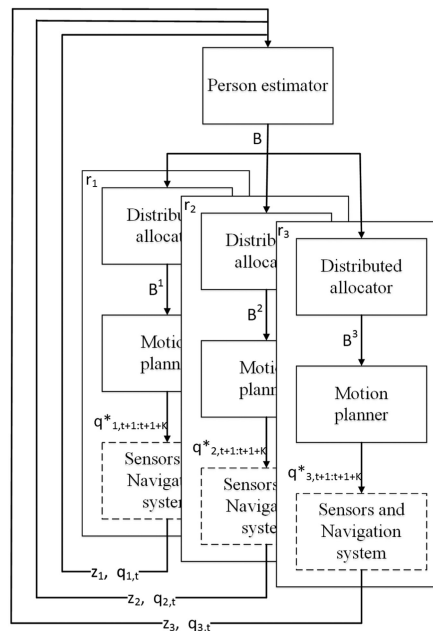


Fig. 3. Multi-robot architecture.

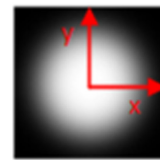


Fig. 4. Sensor model used during the tests. The image scale is 2×2 meters and the degree of white color indicates probability of detection.

to its assigned zone, its readings will be used to update those particles’ weights. Moreover, as the particle set is continuously updated and particles are dynamically re-assigned, the system is robust against unexpected behaviors (e.g., a robot moving too slow or too fast). Finally, as the robots cannot cover all the particles and those are moving (based on the PF prediction stage), the “clean” areas could be filled out again by the nearest particles and, hence, clean areas can not be fully ruled out.

V. VALIDATION RESULTS

The proposed system has been implemented over ROS (Robotics Operating System [21]) and validated in both simulations and real experimental tests. Onwards, it is assumed that the NRS is searching for a child wearing a Radio Frequency Identification (RFID) tag. Robots are assumed to be equipped with RFID readers, which detection model is defined by Fig. 4. This model defines the probability of detecting an RFID tag around the RFID reader. Also, each robot has a map of the scenario and is able to localize itself integrating laser measurements with the Adaptive Monte-Carlo Localization (AMCL) method. The motion model of the child is unknown and, then, a random walk is considered (modeled as a normal distribution centered in its current position with a standard deviation σ).

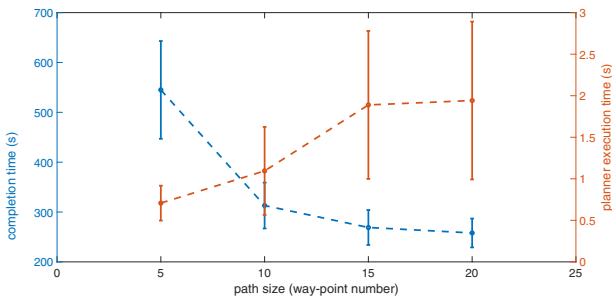


Fig. 5. Average time required by the robots to explore the environment (blue) and to find a valid path (red), with increasing path size for the RRT* planner.

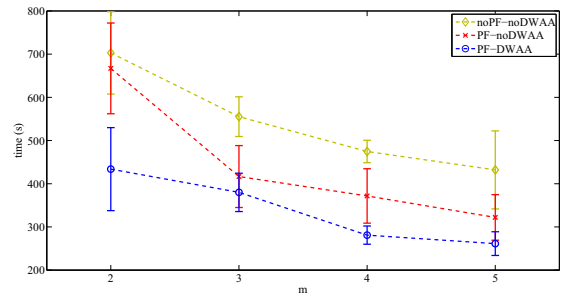


Fig. 6. Average time required by the robots to explore the environment, depending on the number of robots m .

A. Performance Analysis

A set of more than 100 simulations have been run using the implemented system to analyze its overall performance. The scenario is defined by two rectangular rooms (see Fig. 2). The robots are in the large room ($40 \times 8 \text{ m}^2$) and have no access to the small room ($2 \times 2 \text{ m}^2$). The objective is to find a target, who will be in the small room. As the robots cannot go into the small room and detect the target, they have to ensure that the target is not in the large room (when all the particles from the PF are in the small room). For all simulations, the reference positions used by the distributed allocator were $p_{ini} = (-25, 0) \text{ m}$ and $p_{end} = (25, 0) \text{ m}$. As expected, there were no overlapping particle assignments during these simulations.

First, the system is evaluated to analyze the influence of the motion planner. These simulations consider 4 robots with maximum speed of 2 m/s , a $\sigma = 0.15$ for the target motion model and a PF with $N = 1000$ particles. A relevant parameter in the fixed-depth RRT* planner is the horizon or maximum path size K . Increasing this parameter leads to a less greedy solution, but it requires a higher computational cost. Figure 5 shows results depending on the maximum path size for the RRT* planner and confirms these expectations. The results also show that from a certain value (15 for this scenario), the improvement regarding the *completion time* is no longer relevant. This completion time is defined as the time that the team of robots needs to be confident that the target is not in the accessible zone. It means the time required to “clean” all the particles from the accessible zone.

In order to analyze the influence of the PF and the distributed allocator, the proposed algorithm (using a common PF-based estimator and the Dynamic Weighted Area Assignment **PF-DWAA**), has been compared with the following approaches:

- A common PF but without distributed allocation (**PF-noDWAA**). There is a common PF updated with sensor readings from all robots, but the particle set is not divided and allocated to the robots.
- An approach (**noPF-noDWAA**) where each robot uses its own PF (which is not updated by the sensor readings from the rest) and the particle set is not divided either.

The maximum path size is set to 20, the maximum robot speed to 1 m/s , $\sigma = 0.05$ and $N = 2000$. Figure 6 summarizes the obtained results. It shows a better behavior using **PF-DWAA**. Furthermore, the tests showed that the number of robot collisions using **PF-DWAA** was much lower than those using the

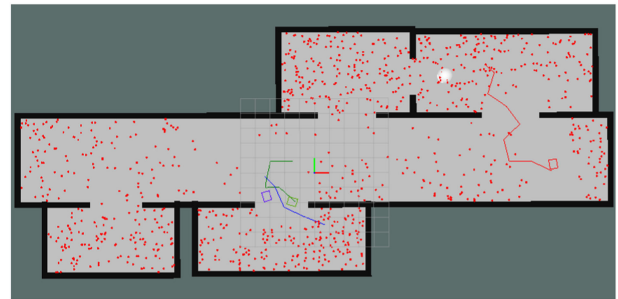


Fig. 7. Snapshot from a simulation with three robots and without allocator module. Green and blue robots generate paths to explore the same area.

other approaches (even more, using **noPF-DWAA**, because the robots use the same particle set and tend to move to the same zones).

B. Use Case in Complex Scenario

A second set of simulations have been run to test the approach under more realistic conditions. A more complex scenario with connected rooms is used so that three robots search for a child moving. In this case, non-ideal RFID readers are used, with a False Negative rate of 5% and a False Positive rate of 0.5%. The reference positions chosen for the distributed allocator were $p_{ini} = (-45.0, 0.0) \text{ m}$ and $p_{end} = (45.0, 0.0) \text{ m}$, $N = 1000$, $\sigma = 0.15$ and maximum path size for the RRT* planner was 10. In this scenario, the proposed method **PF-DWAA** has been compared with **PF-noDWAA** and with another coordinated approach with a fixed area allocation. Without coordination, the robots tend to explore areas reaching an inefficient solution (see Fig. 7). Figure 8 show how the robots behave using the proposed method **PF-DWAA**. A video with the full simulation and comparison with the fixed area allocation method can be viewed in <https://youtu.be/535hFsdmahs>.

The results show that the proposed solution is able to adapt quickly to the target motion and exploits better the robots capabilities to explore the environment. The robots change dynamically their exploration area to maximize the probability of detection. Apart from the required time to detect the target, which depends on the target motion, the main advantage of **PF-DWAA** is its capability to adapt the motion of the robots toward zones with more probability of detection. As shown in

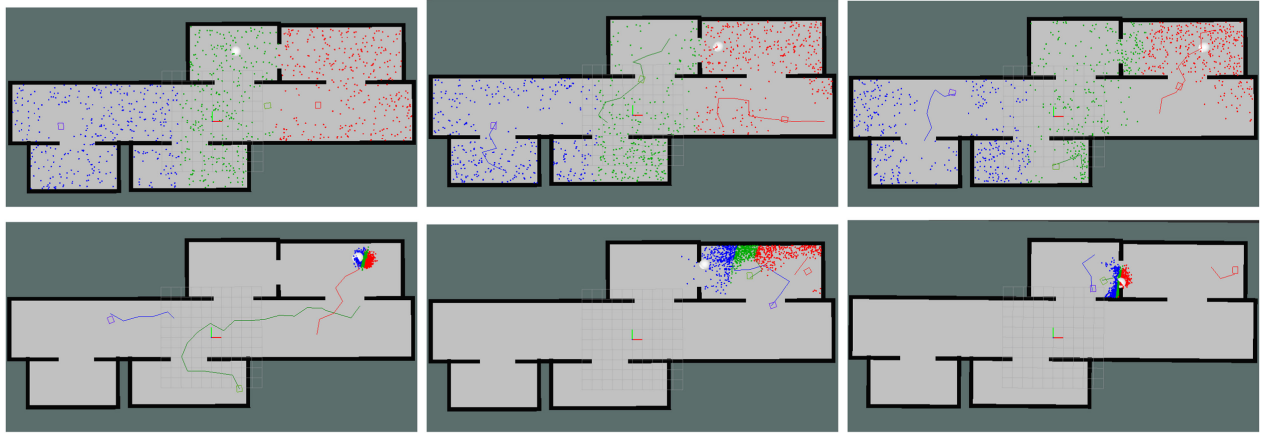


Fig. 8. Sequence of snapshots from simulation using the proposed PF-DWAA approach. It shows the generated path (lines) and the subset of particles assigned to each robot (different colors). White sphere represents the moving target (child).

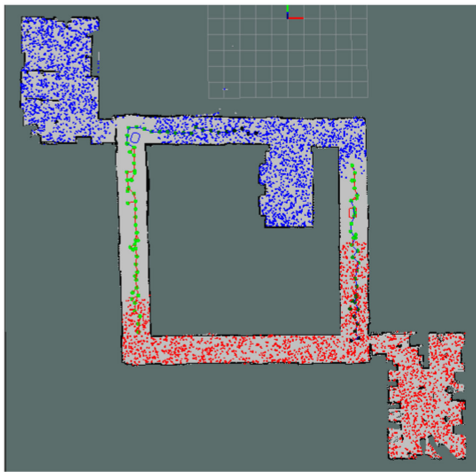


Fig. 9. Snapshot of the real experiment. It shows particles assigned to each robot (red and blue sets) and the connected trees generated by the planning algorithms for both robots. Each node is drawn as a green-black dot. The robots tend to move to darker nodes (less entropy).

Fig. 8, once the child is detected by one of the robots the rest of the robots moves toward her/him, allowing the system to keep a high certainty about its location. However, using the fixed area allocation approach (see simulation video), each robot continues patrolling its assigned area, even once the child has been detected by a robot. When the child moves away from this robot, the certainty about its location is decreasing because the rest of the robots did not get closer.

C. Real Experiments

The system has also been validated experimentally using two actual PO Mbots (see Fig. 1) that, using the proposed algorithms and their on-board RFID readers, have to find a person in a closed environment defined by a corridor and two rooms not accessible for the robots (see the map in Fig. 9). The reference positions used by the distributed allocator were $p_{ini} = (3.85, -14.1)$ m

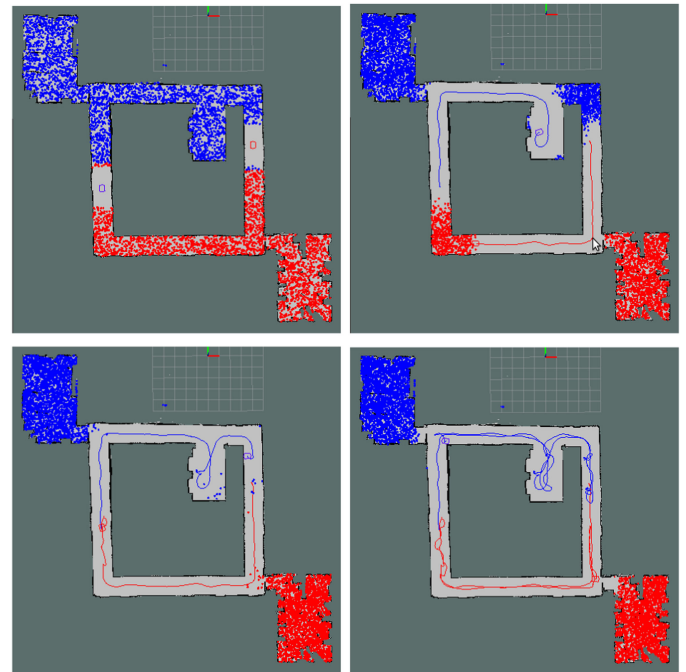


Fig. 10. Sequence of snapshots for the real experiment. It shows the robot trajectories (lines) and the subset of particles assigned (different colors) to each robot. A video can be viewed in [Online]. Available: <https://youtu.be/simBmuGMcOY>.

and $p_{end} = (3.85, -13.9)$ m. Moreover, $N = 5000$, $\sigma = 0.05$ and maximum path size for the RRT* planner was 20.

Figure 9 shows the most relevant information generated and used by the proposed methods. A sequence of snapshots of the experiment are shown in Fig. 10, with the target estimation, the particle assignments and the robot trajectories.

During the experiment, the child (target) was not in the corridor but in one of the rooms. As the robots could not access the rooms, they were not able to detect the target, but once they finished exploring the corridor they were sure that it was in one

of the rooms. Finally, each robot warded a different room in front of its door. Note that, in case that the robots were not using the distributed allocator module, both robots could tend to move together and finish warding the same room.

VI. CONCLUSION

The ACP problem posed here considers a NRS searching for a moving target. This paper proposes an approach based on three key issues: a Particle Filter to estimate the target position, a Dynamic Weighted Area Assignment to coordinate the robots and a entropy-based motion planner to generate each robot path.

The approach has shown to be helpful in this exploration application, where most robot readings are negative (target not detected) and the target motion model is unknown but with slow dynamics.

The robots are coordinated based on a particle allocation using a computationally efficient and fast convergent algorithm which generates non-overlapping subsets of particles. As the motion planning algorithms tends to generate paths towards the zones with higher density of particles, the robots do not cross their motion plans. The provided experimental results (both in simulation and physical tests) validate the system. Moreover, they proved how the system was able to compute solutions in a distributed and efficient manner with multiple robots in coordination, improving alternative methods; even making use of limited RFID sensors with negative information mostly. The proposed system is partially distributed. Although each robot self-assigns its own sub-set of particles and plans its own path, the PF based estimator is common for all of them. Therefore, the future work will be directed towards studying different data fusion techniques to generate common particle sets from asynchronous communication among the robots. This would lead to a totally distributed solution for the problem.

REFERENCES

- [1] J. Messias, J. J. Acevedo, J. Capitan, L. Merino, R. Ventura, and P. U. Lima, "A particle filter approach for active perception in networked robot system," in *Proc. 7th Int. Conf. Social Robot.*, Oct. 2015, pp. 451–460.
- [2] D. Fox, W. Burgard, and S. Thrun, "Active Markov localization for mobile robots," *Robot. Auton. Syst.*, vol. 25, no. 3, pp. 195–207, 1998.
- [3] L. M. Miller and T. D. Murphey, "Optimal planning for target localization and coverage using range sensing," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Aug. 2015, pp. 501–508.
- [4] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, "Autonomous robotic exploration using a utility function based on rényi's general theory of entropy," *Auton. Robots*, vol. 42, no. 2, pp. 235–256, Feb. 2018. [Online]. Available: <https://doi.org/10.1007/s10514-017-9662-9>
- [5] A. Viseras, Z. Xu, and L. Merino, "Distributed multi-robot cooperation for information gathering under communication constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 1267–1272.
- [6] J. Capitan, M. T. J. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned pomdps," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 650–671, 2013.
- [7] R. Tallamraju *et al.*, "Active perception based formation control for multiple aerial vehicles," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4491–4498, Oct. 2019.
- [8] A. Ahmad, T. Nascimento, A. G. S. Conceicao, A. P. Moreira, and P. Lima, "Perception-driven multi-robot formation control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1851–1856.
- [9] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least squares minimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5696–5701.
- [10] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [11] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6* (Distributed Autonomous Robotic Systems), vol. 6, Berlin, Germany: Springer-Verlag, 2007, pp. 221–230.
- [12] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, May/Jun. 2014, pp. 4735–4740.
- [13] D. Kingston, R. Beard, and R. Holt, "Decentralized perimeter surveillance using a team of UAVs," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1394–1404, Dec. 2008.
- [14] X. Geng, "Consensus-reaching of multiple robots with fewer interactions," in *Proc. WRI World Congr. Comput. Sci. Inf. Eng.*, Apr. 2009, vol. 5, pp. 249–253.
- [15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
- [16] S. M. LaValle and J. J. Kuffner, Jr., "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*. Boca Raton, FL, USA: CRC Press, 2000.
- [17] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [18] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal, "Particle filter based entropy," in *Proc. 13th IEEE Conf. Inf. Fusion*, 2010, pp. 1–8.
- [19] L. Merino, F. Caballero, and A. Ollero, "Active sensing for range-only mapping using multiple hypothesis," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 37–42.
- [20] T. W. McLain and R. W. Beard, "Coordination variables, coordination functions, and cooperative timing missions," *J. Guidance, Control, Dyn.*, vol. 28, no. 1, pp. 150–161, 2005.
- [21] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. Open-Source Software Workshop Int. Conf. Robot. Autom.*, Kobe, Japan, 2009.