

DECENTRALIZED MULTI-ROBOT PLANNING TO EXPLORE AND PERCEIVE

LAETITIA MATIGNON^{a,*}, LAURENT JEANPIERRE^b, ABDEL-ILLAH MOUADDIB^b

^a Université Lyon 1, LIRIS UMR5205, F-69622, France

^b Université Caen Basse-Normandie, GREYC UMR6072, F-14032, France

* corresponding author: laetitia.matignon@univ-lyon1.fr

ABSTRACT. In a recent French robotic contest, the objective was to develop a multi-robot system able to autonomously map and explore an unknown area while also detecting and localizing objects. As a participant in this challenge, we proposed a new decentralized Markov decision process (Dec-MDP) resolution based on distributed value functions (DVF) to compute multi-robot exploration strategies. The idea is to take advantage of sparse interactions by allowing each robot to calculate locally a strategy that maximizes the explored space while minimizing robots interactions. In this paper, we propose an adaptation of this method to improve also object recognition by integrating into the DVF the interest in covering explored areas with photos. The robots will then act to maximize the explored space and the photo coverage, ensuring better perception and object recognition.

KEYWORDS: Cooperative multi-robot systems, robot coordination, robot planning, multi-robot exploration, active perception.

1. INTRODUCTION

Some key challenges of robotics reported in the recent roadmap for U.S. robotics [1], *e.g.*, planetary missions and service robotics, require mobile robots to travel autonomously around unknown environments and to augment metric maps with higher-order semantic information such as the location and the identity of objects in the environment. The ability of the mobile robots that gather the necessary information to obtain a useful map for navigation is called autonomous exploration. This was the central topic of a DGA¹/NRA² robotic challenge, in which multiple robots have to explore and map some unknown indoor area while recognizing and localizing objects in this area. The scientific issues of this project involve SLAM³, object recognition and multi-robot collaboration for exploration. As a participant in this challenge, we mainly focused on multi-robot collaboration for exploration. We were particularly interested in multi-robot exploration strategies. We proposed a new Dec-MDP (decentralized Markov decision process) resolution technique based on the distributed value function (DVF) to consider sparse interactions. Our Dec-MDP model for exploration and its resolution based on DVF were applied for the contest. It allowed robots to explore an unknown area cooperatively by reducing the overlap between the explored areas of each robot. In a second phase, we focused on improving object recognition by integrating into the planning stage the interest in covering explored areas with photos. Thus each robot will act to explore and

to perceive. The objective of this paper is to present our interaction-sparse Dec-MDP resolution adapted to achieve improved perception.

In the following, we first present the context of this work, with details concerning the robotic challenge and the system that we developed to participate in the competition. Second, related works on multi-robot exploration, active perception and interaction-oriented models are introduced. Then we present our Dec-MDP resolution, based on DVF, and its application to multi-robot exploration. Finally, we introduce and show experiments aimed at extending DVF to improve photo coverage of the space. Finally, some concluding remarks are made.

2. CONTEXT

2.1. THE CAROTTE CHALLENGE

This work was carried out within the framework of a French robotic contest called Defi CaRotte⁴ (Cartography by a Robot of a territory), which involved exploring and mapping an unknown indoor area (an enclosed arena made up of a set of rooms) with one or several autonomous robots. The competition took place in an arena of approximately $120m^2$, in which objects had been laid out. The arena contained several rooms, typically 10 or more, with variable grounds and various difficulties to be dealt with (fitted carpet, grid, sand, etc.). Several kinds of objects were present, isolated or gathered together, *e.g.*, chairs, books, fans, furniture, etc.

The CaRotte challenge proceeded over a period of three years, with an increase in the level of difficulty over the years. The required outcome was to produce

¹Defense Procurement Agency.

²French National Research Agency.

³Simultaneous Localization and Mapping.

⁴<http://www.defi-carotte.fr/>

2D, 3D and topological maps of the arena and to detect, localize and classify objects present in the arena. The trials within the framework of the competition consisted of several missions with a time limit. During each mission, robots navigated autonomously to map, detect and recognize objects. The robots were required to return to their starting position before the end of the mission. Five teams entered for this challenge [2, 3], in which the goal was to maximize the explored space, the precision of the map and the quality of object detection.

2.2. THE ROBOTS_MALINS SYSTEM

We developed the Robots_Malins⁵ (Robots for Mapping And Localization, using Intelligent Navigation and Search) system for the CaRotte challenge. Our system uses Wifibot⁶ μ -trooper M robots. These are six-wheel robots characterized by great flexibility, which allows the robots to cover rough terrain. Each μ -trooper embeds an Intel Core 2 Duo processor, a 2GB RAM and a 4GB flash and is equipped with a Hokuyo LIDAR 30m⁷ laser range scanner for localization and mapping and a Hokuyo LIDAR 4m tilted toward the ground, plus an ultrasonic rangefinder for detecting nearby obstacles and glass walls. An AVT Marlin firewire camera is used for object detection. The software running on-board these robots is based on a Data Distribution System (DDS) implementation from OpenSplice⁸. This middle-ware allows several programs to run concurrently, even on different computers. In our architecture, this implies that various modules can run asynchronously: Laser acquisition, SLAM, Decision, Mobility and Object recognition. The architecture allows the robots to exchange their laser scans and their poses. Thus each robot knows the areas explored by the others, and updates its local map with local and distant scans.

In particular, the *SLAM module*, based on [4], receives laser readings and provides the other modules and the other robots with the robot pose (location and heading). So each robot knows the relative positions of all the robots and the map of the zones explored by the team. The *mobility module* implements an advanced point and shoot algorithm, along with a backtrack feature that prevents the robot from being stuck, reverting back on its own trajectory. The point and shoot algorithm consists of turning to the desired heading and then going forward for a specified distance, while correcting the heading drift. The *object recognition module* uses the different pictures taken by the camera to recognize predefined classes of objects. Pictures are taken every 4 seconds. Objects to be detected are known in advance and a database has been built containing each object over different

points of view. Object detection was performed using a shape/template matching technique based on Dominant Orientation Template models [5]. The *decision module* runs asynchronously, computing a new strategy every second on an average. In this paper, we focus on the decision module. Details on the algorithm used to compute a joint policy of the robots to efficiently explore the arena and cover it with pictures are given in Sections 5 and 6.

3. STATE OF THE ART

3.1. MULTI-ROBOT EXPLORATION

Multi-robot exploration has received considerable attention in recent years due to its obvious advantages over single-robot systems: it is faster, robust, fault-tolerant, etc. Most multi-robot approaches assume that robots share the information that they have gathered to build a shared map and know their locations in this map. The robots cooperate through the shared map, but coordination techniques are required to exploit the parallelism inherent to the team. In [6], each robot uses the greedy approach in the shared map, *i.e.*, each robot chooses the nearest exploration frontiers, defined as regions on the boundary between open space and unexplored space. Therefore, there is no coordination and multiple robots can be assigned to the same frontier. In [7–9], the coordination is centralized. A cost-utility model is used, where the gain expected at a target is the expected area discovered when the target is reached, taking into account the possible overlap between robot sensors. Coordination is accomplished by assigning different targets to the robots, thus maximizing the coverage and reducing the overlap between the explored areas of each robot. Coordination can also be decentralized. In [10] robots bid on targets to negotiate their assignments. Classically, frontiers are rated using the distance to the robot, but Bautin *et al.* [11] chose to favor a well-balanced spatial distribution of robots in the environment: for each robot-frontier pair, the cost function is the number of robots closer than it to the considered frontier. Accordingly, each robot is allocated to the frontier for which it has the lowest rank. All these strategies have been devised, but few efforts have been made to compare them. An exception is a recent article [12] that compares some methods for autonomous exploration and mapping using various criteria, *e.g.*, exploration time and map quality.

3.2. ACTIVE PERCEPTION COMBINED WITH EXPLORATION

Early works on object recognition were based on passive approaches. During the last decade, some approaches have investigated the field of active object recognition: the viewpoint of the camera can be controlled to improve the recognition rate. For example, looking at an object from different poses decreases ambiguities in object recognition, thanks to the choice of

⁵https://robots_malins4carotte.greyc.fr/

⁶www.wifibot.com

⁷www.hokuyo-aut.jp

⁸<http://www.opensplice.com>

different viewpoints. Several active approaches have been proposed for planning optimal sequences of views for a camera. An approach for viewpoint selection based on reinforcement learning is proposed in [13]. In [14], the objective is to plan the minimum number of actions and observations required to achieve recognition. Active perception planning is formulated as POMDP. However, the issue of active planning combined with exploration is not addressed in most of these works, as the location of the object to be recognized is known.

Other works are interested in adding in the trajectories planned for the exploration other objectives. For example, integrated exploration [15, 16] involves integrating the path planned for exploration with SLAM in order to plan trajectories that favour the creation of a high-quality map. Some actions, *e.g.*, closing a loop or returning to precedent positions, may reduce the uncertainty of the robot pose and the uncertainty of the map. A utility function is used which trades off the cost of exploring new terrain against the potential reduction of uncertainty by making measurements at selected positions. In these works, there is no verification of the assumption of having an accurate position estimate during exploration. Integrated exploration considers the problem of acting for better localization, but not for better recognition.

4. INTERACTION-ORIENTED MODELS

Decision-theoretic models based on Dec-MDPs provide an expressive means for modeling cooperative teams of decision makers. In this section, we present this formalism and recent advances in resolving it.

4.1. BACKGROUND ON DEC-MDP

Dec-MDP [17] is an extension of MDP [18] for decentralized control domains. A Dec-MDP is defined with a tuple $\langle I, S, A, T, R, \Omega, O \rangle$. I is the number of agents, S is the set of joint states and $A = \{A_i\}$ is the set of joint actions⁹. $T : S \times A \times S \rightarrow [0; 1]$ is a transition function and $T(s, a, s')$ is the probability of the I robots transitioning from joint state s to s' after performing joint action a . $R : S \rightarrow \mathfrak{R}$ is a reward function that represents the global immediate reward for the robots being in s . Ω is a set of joint observations that agents can receive about the environment and $O : S \times A \times S \times \Omega \rightarrow [0; 1]$ is an observation function giving the probability of receiving $o \in \Omega$ after performing joint action a and transitioning from s to s' . If the global state of the system is collectively totally observable, the Dec-POMDP is reduced to a Dec-MDP.

We can see an MDP as a Dec-MDP where $I = 1$. It is defined with a tuple $\langle S, A, T, R \rangle$. The goal of MDP planning is to find a sequence of actions maximizing the long-term expected reward. Such a plan is called

⁹A state of the problem can be written with a tuple $s = (s_1, \dots, s_I)$ such that s_j is the state of robot j . A_j defines the set of actions a_j of robot j .

a policy $\pi : S \rightarrow A$. An optimal policy π^* specifies for each state s the optimal action to execute in the current step, assuming that the agent will also act optimally in future time steps. The value of π^* is defined by the optimal value function V^* that satisfies the Bellman optimality equation:

$$V^*(s) = R(s) + \gamma \max_{a \in A} \sum_{s' \in S} T(s, a, s') V^*(s'), \quad (1)$$

where γ is the discount factor. An MDP is solved using Dynamic Programming, with polynomial time complexity. A Dec-POMDP is solved similarly by computing the optimal joint policy. However its time complexity is NEXP-complete [17], which is extremely hard.

4.2. INTERACTION-ORIENTED MODELS

When faced with real-world applications such as multi-robot systems, Dec-(PO)MDP models are very difficult to apply, due to their high complexity. Recent advances in Dec-(PO)MDP resolution have allowed a notable increase in the size of the problems that can be solved. An interesting direction that has emerged recently involves taking advantage of local or sparse interactions between agents. These methods relax the most restrictive and complex assumption which considers that the agents interact permanently with all the others. The complexity of solving Dec-(PO)MDPs is then reduced by solving a set of interactive individual decision making problems. The ND-POMDP model [19] is a static interaction approach, *i.e.*, an agent always interacts with the same subset of neighbors. However this assumption is not realistic. Models have therefore been proposed that use dynamic interactions so that each agent interacts with an evolving set of agents. The Dec-SIMDP model assumes full local observability, and unlimited, free communication between agents interacting together in some specific states [20]. DyLIM is similar but applies to partial observation and no communications [21]. It considers Dec-POMDPs as a set of POMDPs, and interactive situations are solved separately by deriving joint policies. For non-interactive situations, each agent has its local policy to behave solely. These are promising approaches for real-world applications of decentralized decision makers.

5. DVF FOR MULTI-ROBOT EXPLORATION

During the robotic contest, we were particularly interested in multi-robot exploration strategies. In this section, we present our fully decentralized approach based on DVF and its application to multi-robot exploration.

5.1. INTERACTION SPARSE DEC-MDP WITH DVF

To improve the complexity for solving Dec-MDPs, we proposed an interaction-oriented resolution based

on distributed value functions (DVF). DVFs were introduced by [22] as a way to distribute reinforcement learning knowledge through different agents. Our approach decouples the multi-agent problem into a set of individual agent problems, and considers possible interactions among a team as a separate layer. This currently seems one of the main tracks for tackling scalability in Dec-(PO)MDPs (cf. Section 4.2). We represent a Dec-MDP with two classes:

- The global interaction class, defined as a collection of augmented MDPs $\{MDP_1^{Aug}, \dots, MDP_I^{Aug}\}$. There is one augmented MDP per agent, which is defined by $MDP_i^{Aug} = \langle S_i, A_i, T_i, R_i, \Gamma_i \rangle$ where $\langle S_i, A_i, T_i, R_i \rangle$ individually models agent i in the absence of other agents and Γ_i is some additional information. Γ_i can be communicated by other agents or can be inferred locally. It provides the agent with global information, enabling interaction between MDPs of the global interaction class. Then, each agent resolves solely its augmented MDP so that interactions are minimized. The global Dec-MDP is solved as a set of local MDPs augmented by information from the other agents. This significantly reduces the computational complexity: the NEXP complexity of solving a Dec-MDP is reduced to the complexity of solving one MDP (polynomial) per agent.
- The local interaction class is for close interactions. Indeed, each agent computes strategies with DVF in its augmented MDP to minimize the interactions. However, when situations of interaction occur, DVF does not handle these situations and the local coordination must be resolved separately with another technique. For example, joint policies can be computed off-line for the specific joint states of close interactions, including only interacting agents.

In the exploration context, the additional information of the augmented MDP is limited to the last known state of other agents: Agent i knows at each step the state $s_{ij} \in S_i$ of the other agents j . Then it computes its distributed value function DVF_i according to

$$DVF_i(s_i) = R_i(s_i) + \gamma \max_{a_i \in A_i} \left(\sum_{s' \in S_i} T_i(s_i, a_i, s') \left(DVF_i(s') - \sum_{j \neq i} f_{ij} P_r(s' | s_{ij}) V_j(s') \right) \right) \quad (2)$$

for all $s_i \in S_i$, where $P_r(s' | s_{ij})$ is the probability of agent j transitioning from state s_{ij} to state s' ; $V_j(s')$ is the value function of agent j , computed locally by agent i ; f_{ij} is a weighting factor that determines how strongly the value function of agent j reduces the value function of agent i .

The DVF technique allows each agent to choose a goal which should not be considered by the others, and in this way the interactions are minimized. The value of a goal depends on the expected rewards at

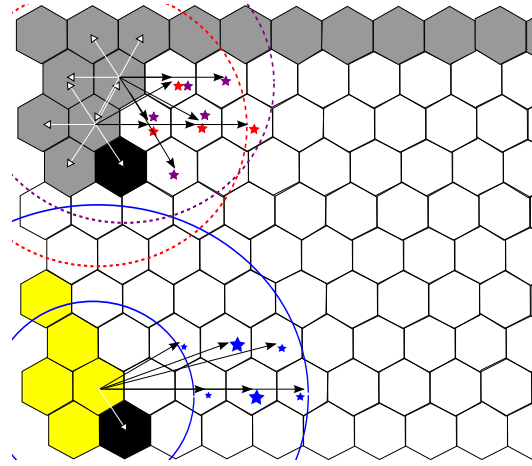


FIGURE 1. Reward propagation mechanisms with stars as resulting rewards. a) From frontier hexagons at the top. An unknown hexagon (grey) propagates its reward over a radius (dotted circle) on free neighborhood hexagons (white). Propagation is stopped if occupied (black) or unknown hexagons are encountered. White arrows show impossible propagations, and black arrows represent active propagations. b) From free and non-covered hexagons at the bottom. A free and non-covered hexagon (yellow) propagates varying rewards at a best view point distance (solid circle).

this goal and on the fact that it is unlikely to be selected by other agents. More details about DVF and its extension under communication constraints can be found in [23].

5.2. DVF APPLIED TO MULTI-ROBOT EXPLORATION

In the second year of the contest, DVF was used in our decision module to compute multi-robot exploration strategies in a decentralized way. Thanks to the SLAM module (cf. Section 2.2), each robot has access to a map updated with all explored areas and to the position of all the robots. We can then assume that the location of the robot and of the other robots is known at decision time.

5.2.1. MDP MODEL

Each robot generates its local augmented MDP from a four-layer grid. The first layer is the real world layer where the robots move. The pixel layer is an occupancy grid of pixels, where each pixel is initialized as unknown and updated as free (no obstacle) or occupied (something there) by the data acquisition process. The hexagon layer is an occupancy grid of hexagons¹⁰. Hexagons and Voronoi layers are each computed from the pixel layer, and are used to generate the data structures of the local augmented MDP. States and rewards are based on the hexagonal layer, while actions and transitions are based on the hexagonal and Voronoi

¹⁰Each hexagon is composed of a set of pixels and is considered as unknown, free or occupied according to the value of its pixels.

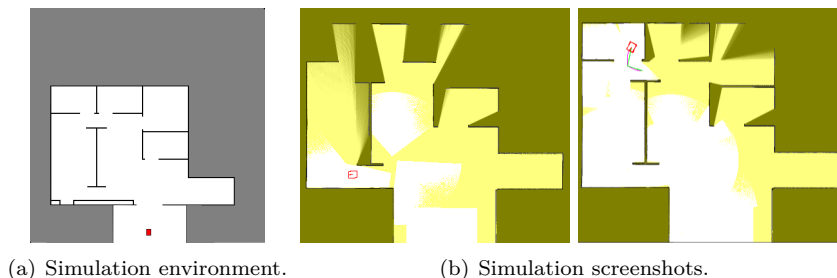


FIGURE 2. In b), areas that have been explored but not yet covered with photos are in yellow; explored areas covered with photos are in white; non explored areas are in grey.

layers. The exploration reward function is computed with a reward propagation mechanism based on the expected information gain in each state as in [24]. We propagate rewards in some radius around the frontier hexagons, taking into account line-of-sight constraints (see Fig. 1a). Further details about our model are given in [25].

5.2.2. DVF

To apply DVF (2), we consider that the robots are homogeneous. The value functions V_j of the other robots j can therefore be computed only once by robot i ¹¹. Robot i computes an *empathic value function* with the standard value iteration algorithm [26] in its local augmented MDP. To evaluate the transition probability of other robot j , i applies a wavefront propagation algorithm from the last known state s_j of robot j .

5.2.3. DECISION STEP

A decision step involves building the model, computing the policy from DVF, and producing a smoothed trajectory. The agent plans continuously, executing a decision step as it perceives changes in its environment. We can observe that exploration rewards will never be gained by the robot. Indeed, as the robot comes close enough to the frontier, it will gather new information with its sensors and unknown cells will become known before they are reached. Therefore, the exploration rewards will disappear before the robot can claim them and the frontier between known and unknown areas, which is the source of the rewards, retreats as the robot approaches it. The action plan must then be updated quickly to react as soon as possible to this kind of information gained *en route*. However, this requires the decision step to be quick enough for on-line use. Given that the model will be updated at each decision step, we use the greedy approach, which plans on a short-term horizon.

5.2.4. EXPERIMENTS

Experimental results from simulation and real-world scenarios can be found in [23, 25]. Videos¹² present

¹¹If the robots are not homogeneous we just need to compute one value function for each type of robot

¹²available at <http://liris.cnrs.fr/laetitia.matignon/research.html>

various exploration tasks with real robots, and some interesting situations are underlined as global task repartition or local coordination. The experiments show that this method is able to coordinate a team of robots effectively during exploration. The global interaction class addresses the allocation of exploration goals, and also minimizes close interactions between robots. Each robot locally computes a strategy that minimizes the interactions between the robots and maximizes the explored space.

6. PLANNING TO EXPLORE AND PERCEIVE

In the first two years of the contest, pictures were taken to locate and recognize objects. This was done separately from the decision module. Pictures taken by the camera and analyzed by the object recognition module were gathered along the way, *i.e.*, the camera took pictures at a specified rate along the trajectory planned for the exploration. However, this led to poor performance in terms of object recognition results, primarily because some objects in the arena were not photographed. Indeed, the pictures that are taken depend on the trajectory computed for the exploration, so some objects were not covered by photos, while some areas without objects were photographed several times.

To improve the coverage of the objects with pictures, we extended the decision module so that it does not only explore the arena but must also ensure that the explored areas are covered by photos. Indeed, if all the explored space has been photographed, each object must be in at least one picture and the recognition module will handle it. In this section, we introduce a DVF extension to improve the photo coverage of the space, and we also present some experiments.

6.1. DVF EXTENSION TO IMPROVE PHOTO COVERAGE OF THE SPACE

The decision module must simultaneously combine the interest in exploring an area with the interest in covering it with pictures. To take into account both the exploration criterion and the picture coverage criterion, we modified the reward function R_i of the augmented MDP. We introduced a specific reward for

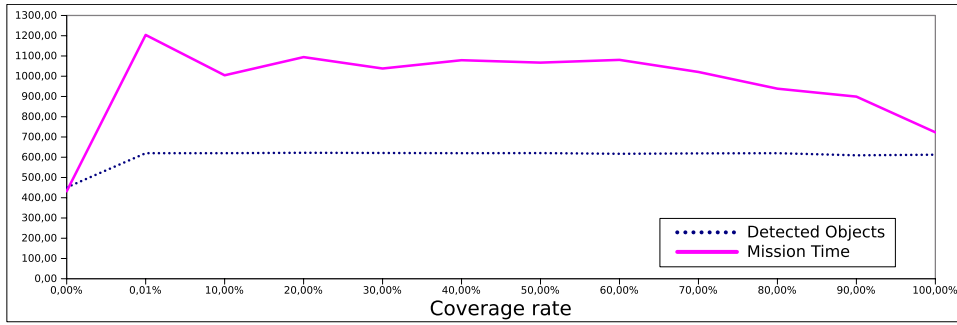


FIGURE 3. Number of objects detected and mission time (in seconds) versus coverage rate (produced from 340 simulations).

areas explored and covered with photos in addition to the exploration reward. MDPs allow planning while optimizing several criteria at once: in our case, the expected information gain, the picture-taking and the cost to reach the chosen location. During the challenge, we also optimized the return-to-base criterion at the end of the mission and the ball-pushing feature when the ball was detected. These two criteria are not considered in this paper. The DVF equation is then adapted to: $\forall s_i \in S_i$

$$\begin{aligned} \text{DVF}_i(s_i) = & (1 - \alpha)R_{i,\text{exp}}(s_i) + \alpha R_{i,\text{cov}}(s_i) \\ & + \gamma \max_{a_i \in A_i} \left(\sum_{s' \in S_i} T(s_i, a_i, s') \left(\text{DVF}_i(s') \right. \right. \\ & \left. \left. - \sum_{j \neq i} f_{ij} P_r(s' | s_{ij}) V_j(s') \right) \right) \quad (3) \end{aligned}$$

where $R_{i,\text{exp}}(s_i)$ is the reward function for exploring a state s_i and $R_{i,\text{cov}}(s_i)$ is the reward function for taking a photo of s_i . $\alpha \in [0, 1]$ is the picture coverage rate for balancing the exploration and picture coverage behaviors of the robots. With $\alpha = 0$, the robots will only optimize the exploration without covering the space with photos. With $\alpha = 1$, the robots will only optimize photo coverage. Exploration then occurs only as a side-effect.

To compute the cover reward function $R_{i,\text{cov}}$, a cover grid of pixels counts the number of times that each pixel has been photographed. Each time the robot takes a picture, the cover grid is updated by tracing a set of rays covering the optimal recognition area of the camera. The pixels are updated with Bresenham’s algorithm [27]. The cover reward is then computed with a hexagonal reward propagation mechanism similar to the exploration rewards, multiplied by a bell-shaped factor to boost the rewards at the optimal recognition distance (see Fig. 1b). This allows the MDP to select the best viewpoint for the next picture. Regarding communications, this new reward implies that new data is sent: each robot needs to know where a picture has been taken. Thus, each robot sends the location of its photos to all the other robots. Then they can update their coverage grid and maintain an accurate reward function.

In order to keep the computing complexity low, we chose not to add an action for taking a picture. When the policy brings the robot to a location where a picture must be taken, the robot will stop there and face the reward. When a picture is taken, the reward disappears and the new policy drives the robot farther. In order to avoid having blurred pictures where the recognition process will fail, we specify that photos must be taken only when the robot velocity is low.

6.2. EXPERIMENTS

6.2.1. SIMULATED ROBOTS

Stage¹³ simulator was used with an architecture that mimics the real robots. DDS is replaced by an Inter Process Communication shared memory segment. Laser acquisition is simulated by a “ranger” virtual sensor. A “position” virtual device simulates both the SLAM module by providing odometric data and the mobility module by executing the point and shoot algorithm. The Stage blobfinder model is used to simulate (poorly) the camera and object detection, as it can track areas of color in a simulated 2D image, giving the location and the size of the color blobs. Thanks to this architecture, the decision module of the real robots can be used with the simulator without modification.

6.2.2. RESULTS

We conducted a set of experiments in the autolab environment (see Fig. 2a). Figure 2b shows successive screenshots of one simulation. The robots are initially in the starting zone and objects are regularly positioned in the environment. For each coverage rate, we compute the number of objects detected and the mission time (see Fig. 3). With $\alpha = 0$, the mission is fastest, given that the robot does not care about taking pictures. The only goal of the robot is to map the entire arena, and it only takes few pictures when it stops between two actions. Few objects are therefore detected. With $\alpha > 0$, all objects are detected but the mission time varies over α . Indeed α modifies the priority of the two tasks. When α is low, pictures are taken at the end of the mission, if there is sufficient time remaining time. This is illustrated

¹³<http://playerstage.sourceforge.net/>

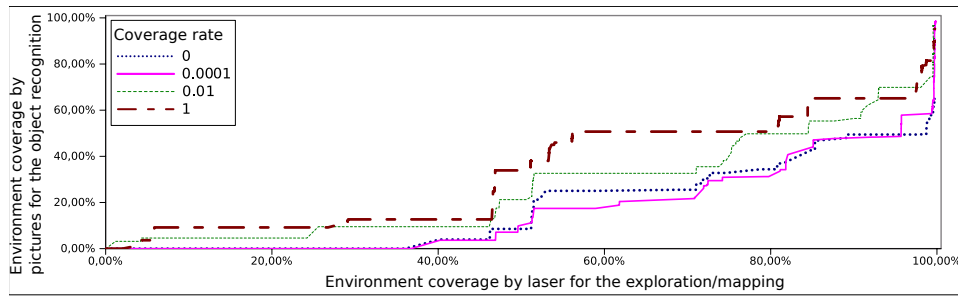


FIGURE 4. Percentage of objects detected versus percentage of exploration.

in Fig. 4, with $\alpha = 0.0001$, where the number of detected objects increases and 100% of the arena has been explored. With such a low α , the robot takes pictures once the exploration has been finished, so it travels the arena twice and the mission time is high. When α is high, pictures are taken throughout the mission, and the robot deals with both tasks at the same time. With $\alpha = 1$, we obtain the second fastest mission time and all the objects are detected. So this new approach, based on exploration and photo coverage of the space, allows all of the objects to be photographed. The α factor must be chosen to balance priority of exploration versus object detection. For example, if the time is limited and α is high, the robot may not explore the whole map, since taking pictures is time consuming, but it will photograph all objects in the explored area.

7. CONCLUSION AND PERSPECTIVES

We have shown a new algorithm for coordinating multiple robots that explore some unknown area. DVF enables the expensive Dec-MDP to be solved as a set of augmented MDPs in a distributed way. The versatile reward function can be adapted so that the robots explore the whole area, but also so that they choose the right positions to take meaningful pictures. As a result, we can observe that the robots take longer to explore, but that they also take many more pictures, efficiently covering the whole space. We can then expect a better detection rate, as any object should be in at least one picture. An immediate perspective is therefore to compare object recognition results with different parameters on real robots to confirm through a real experiment that our method provides improved perception.

Another perspective of our work is to plan to perceive by interlacing detection and decision modules to achieve more robust object recognition. The idea is not to take more pictures, but to take better pictures. The objective would be to plan viewpoints where the recognition process would be more reliable. To achieve this kind of active perception planning, the decision module should use information from the recognition module. In our work, the recognition module is based on the Dominant Orientation Template method [5]. For each object, a set of views is defined, and the

method gives a weighting for each view that is equivalent to its precision¹⁴. A positive detection is reliable. For each object detected, the module also gives the location of the object and scores (matching results) for each view. Thus the decision module could manage a set of hypotheses about the probability of the presence of each object. These hypotheses could then be confirmed or discarded by taking pictures of the object from viewpoints that maximize the precision. This could be done easily by generating specific high rewards at these viewpoints, so that the decision module will adapt the planned trajectory to ensure that the robots take pictures from there.

ACKNOWLEDGEMENTS

This work has been supported by NRA and DGA (ANR-09-CORD-103) and was developed jointly with the members of the Robots Malins consortium. It has also received support from the INSA Foundation CROME project.

REFERENCES

- [1] From Internet to Robotics: A Roadmap for U.S. Robotics . 2013.
- [2] A. Bautin, P. Lucidarme, R. Guyonneau, et al. Cart-O-matic project : autonomous and collaborative multi-robot localization, exploration and mapping. In *Proc. of 5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pp. 210–215. 2013.
- [3] D. Filliat, E. Battesti, S. Bazeille, et al. Rgb-d object recognition and visual texture classification for indoor semantic mapping. In *Proc. of TePRA*. 2012. DOI:10.1109/TePRA.2012.6215666.
- [4] J. Xie, F. Nashashibi, N. M. Parent, O. Garcia-Favrot. A Real-Time Robust SLAM for Large-Scale Outdoor Environments. In *17th ITS World Congress*. 2010. DOI:10.1028/ITS.SLAM.Nashashibi.
- [5] S. Hinterstoisser, V. Lepetit, S. Ilic, et al. Dominant orientation templates for real-time detection of texture-less objects. In *CVPR*, pp. 2257–2264. 2010.
- [6] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents, AGENTS '98*, pp. 47–53. 1998.

¹⁴or positive predictive value

- [7] R. Simmons, D. Apfelbaum, W. Burgard, et al. Coordination for Multi-Robot Exploration and Mapping. In *Proc. of the AAAI National Conf. on Artificial Intelligence*. 2000.
- [8] W. Burgard, M. Moors, C. Stachniss, F. Schneider. Coordinated Multi-Robot Exploration. *IEEE Transactions on Robotics* **21**:376–386, 2005. doi:10.1109/TRO.2004.839232.
- [9] K. M. Wurm, C. Stachniss, W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. of IROS*, pp. 1160–1165. 2008. doi:10.1109/IROS.2008.4650734.
- [10] R. Zlot, A. Stentz, M. Dias, S. Thayer. Multi-Robot Exploration Controlled By A Market Economy. In *Proc. of ICRA*, vol. 3, pp. 3016–3023. 2002. doi:10.1109/ROBOT.2002.1013690.
- [11] A. Bautin, O. Simonin, F. Charpillat. MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration. In *ICIRA*, vol. 7507 of *Lecture Notes in Computer Science*, pp. 496–508. 2012.
- [12] M. Julia, A. Gil, Ó. Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots* **33**(4):427–444, 2012. doi:10.1007/s10514-012-9298-8.
- [13] F. Deinzer, C. Derichs, H. Niemann, J. Denzler. A framework for actively selecting viewpoints in object recognition. *IJPRAI* **23**(4):765–799, 2009.
- [14] S. Brandao, M. Veloso, J. P. Costeira. Active object recognition by offline solving of pomdps. In *Proc. of the International Conference on Mobile Robots and Competitions*, pp. 33–38. 2011.
- [15] A. A. Makarenko, S. B. Williams, F. Bourgault, H. F. Durrant-Whyte. An Experiment in Integrated Exploration. In *Proceedings of IROS*, pp. 534–539. 2002. doi:10.1109/IRDS.2002.1041445.
- [16] C. Stachniss, G. Grisetti, W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proc. of Robotics: Science and Systems (RSS)*. Cambridge, MA, USA, 2005.
- [17] D. S. Bernstein, R. Givan, N. Immerman, S. Zilberstein. The Complexity of Decentralized Control of Markov Decision Processes. *Math Oper Res* **27**:819–840, 2002.
- [18] M. L. Puterman. *Markov decision processes*. 1994.
- [19] R. Nair, P. Varakantham, M. Tambe, M. Yokoo. Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs. In *Proc. of AAAI*, pp. 133–139. 2005.
- [20] F. S. Melo, M. M. Veloso. Decentralized MDPs with sparse interactions. *Artificial Intelligence* **175**(11):1757–1789, 2011. doi:10.1016/j.artint.2011.05.001.
- [21] A. Canu, A.-I. Mouaddib. Collective Decision-Theoretic Planning for Planet Exploration. In *Proc. of ICTAI*. 2011.
- [22] J. Schneider, W.-K. Wong, A. Moore, M. Riedmiller. Distributed Value Functions. In *Proc. of ICML*, pp. 371–378. 1999.
- [23] L. Matignon, L. Jeanpierre, A.-I. Mouaddib. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Proc. of AAAI*. 2012.
- [24] S. Le Gloannec, L. Jeanpierre, A.-I. Mouaddib. Unknown Area Exploration with an Autonomous Robot using Markov Decision Processes. In *Proc. of TAROS*, pp. 119–125. 2010.
- [25] L. Matignon, L. Jeanpierre, A.-I. Mouaddib. Distributed Value Functions for Multi-Robot Exploration. In *Proc. of ICRA*. 2012. doi:10.1109/ICRA.2012.6224937.
- [26] R. Bellman. *Dynamic programming: Markov decision process*. 1957.
- [27] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal* **4**(1):25–30, 1965.