Decentralized Multi-Robot Planning to Explore and Perceive

Laetitia Matignon
Université Lyon 1
LIRIS UMR5205
F-69622, France
Email: laetitia.matignon@univ-lyon1.fr

Laurent Jeanpierre
Université Caen Basse-Normandie
GREYC UMR6072
F-14032, France
Email: laurent.jeanpierre@unicaen.fr

Abdel-Illah Mouaddib
Université Caen Basse-Normandie
GREYC UMR6072
F-14032, France
Email: abdel-illah.mouaddib@unicaen.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 in the environment. The scientific issues of this project deal with SLAM (Simultaneous Localization and Mapping), object recognition and multi-robot collaboration for the exploration. 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 will present an adaptation of this method to improve also object recognition. It consists in integrating in the DVF the interest to cover explored areas with photos. Robots will then act to maximize the explored space and the photo coverage ensuring a better perception and object recognition. After a brief description of the robotic contest and our system, we will present our Dec-MDP resolution based on DVF and its application to multi-robot exploration. The next part of the paper will be dedicated to the use of DVF to explore and perceive with experimental results from simulated scenarios.**

I. Introduction

Some key challenges of robotics reported in the recent roadmap for U.S. robotics [1], such as planetary missions or service robotics, require mobile robots to autonomously travel around unknown environments and to augment metric maps with higher-order semantic information such as the location and identity of objects in the environment. The ability of mobile robots gathering the necessary information to obtain a useful map for navigation is called autonomous exploration. This has been the central topic of a DGA[1]/NRA[2] robotic challenge where 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 deal with SLAM[3], object recognition and multi-robot collaboration for the exploration. As a participant in this challenge, we mainly focused on the last topic. We were particularly interested in multi-robot exploration strategies. We proposed a new Dec-MDP (decentralized markov decision process) resolution technique based on distributed value function (DVF) to consider sparse interactions. Our Dec-MDP model for exploration and its resolution based on DVF have been applied during the contest. It allowed robots to cooperatively explore an unknown area by reducing the overlap between the explored area of each robot. In a second phase, we focused on improving object recognition by integrating in the planification the interest to cover 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 improve the perception.

In the following, we first present the context of this work with details concerning the robotic challenge and the system we developed to participate. Second, related works regarding 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 concerning DVF extension to improve photo coverage of the space before concluding.

[1]Defense Procurement Agency.
[2]French National Research Agency.
[3]Simultaneous Localization and Mapping.

II. Context

A. CaRotte Challenge

This work has been done in the framework of a French robotic contest called Defi CaRotte[4] (Cartography by a Robot of a territory) that consisted in exploring and mapping an unknown indoor area (closed arena made up of a set of rooms) with one or several autonomous robots. The competition took place in an arena of approximately $120m^2$ where objects were laid. The arena contained several rooms, typically 10 or more, with variable grounds and various difficulties (fitted carpet, grid, sand, ...). Several kinds of objects were present, either isolated or gathered, like chairs, books, fans, furnitures, ...

The CaRotte challenge proceeded over three years with an increase in the difficulty over the years. The expected result was to produce 2D, 3D and topological maps of the arena and to detect, localize and classify objects present in the arena. The trials of the competition consisted in several missions with a limited time. During each mission, robots navigate autonomously to map, detect and recognize objects. Robots must return to their starting position before the end of the mission. 5 teams entered this challenge [2], [3] where the goal was to maximize the explored space, the precision of the map and the quality of objects detection.

[4]http://www.defi-carotte.fr/

B. The Robots_Malins System

We developed the Robots_Malins[5] (Robots for Mapping And Localization, using Intelligent Navigation and Search) system that took part in the CaRotte challenge. Our system uses Wifibot[6] μ-trooper M robots. These are six wheel robots characterized by a great flexibility that allows the robots to be over rough terrain. Each μ-trooper embeds an Intel Core 2 Duo processor, 2GB RAM and 4GB flash and is equipped with an Hokuyo LIDAR 30m [7] laser range scanner for the localization and mapping and an Hokuyo LIDAR 4m tilted toward the ground plus an ultrasonic rangefinder for the detection of close obstacles and glass walls. An AVT Marlin firewire camera is used for the object detection. The software running on-board these robots is based on a Data Distribution System (DDS) implementation from OpenSplice[8]. This middle-ware allows for several programs to run concurrently, even on different computers. In our architecture, that 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 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 the specified distance, while correcting 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 beforehand 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 Templates models [5]. The *decision module* runs asynchronously, computing a new strategy every second in 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 V-VI.

III. State of the Art

A. Multi-robot exploration

Multi-robot exploration has received considerable attention in recent years given the obvious advantages over single-robot systems: faster, robust, fault-tolerant, ... Most multi-robot approaches assume that robots share the information they 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 one 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 in between robot sensors. Coordination is accomplished by assigning different targets to the robots, thus maximizing the coverage and reducing the overlap between 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: The cost function is, for each robot-frontier pair, the number of robots closer than it towards the considered frontier. Accordingly, each robot is allocated to the frontier for which it has the lowest rank. Despite all these strategies, few efforts have been made to compare them, except for a recent article [12] that compares some methods for autonomous exploration and mapping using different criteria as exploration time or map quality.

B. Active perception combined with exploration

First 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 different viewpoints. Several active approaches have been proposed to plan optimal sequences of views for a camera. In [13] an approach for viewpoint selection based on reinforcement learning is proposed. In [14] the objective is to plan the minimum number of actions and observations required to achieve recognition. The 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 recognize is known.

Other works are interested in adding in the trajectories planned for the exploration other objectives. For instance, integrated exploration [15], [16] consists in integrating the path planned for exploration with the SLAM in order to plan trajectories that favour the creation of a high quality map. Some actions as 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 with the potential reduction of uncertainty by measuring at selected positions. In these works, the assumption of having an accurate position estimate during exploration is not verified. Integrated exploration considers the problem of acting to better localize but not to better recognize.

IV. Interaction-oriented Models

Decision-theoretic models based on Dec-MDPs provide an expressive mean of modeling cooperative teams of decision makers. We present in this section this formalism and recent advancements in its resolution.

[5]https://robots_malins4carotte.greyc.fr/

[6]www.wifibot.com

[7]www.hokuyo-aut.jp

[8]http://www.opensplice.com

A. 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 $< I, S, A, T, R, \Omega, O >$. I is the number of agents, S the set of joint states and $A = \{A_i\}$ the set of joint actions[9]. $T : S \times A \times S \to [0; 1]$ is a transition function and $T(s, a, s')$ is the probability for the I robots of transitioning from joint state s to s' after doing joint action a. $R : S \to \Re$ is a reward function that represents the global immediate reward for the robots being in s. Ω a set of joint observations agents can receive about the environment and $O : S \times A \times S \times \Omega \to [0; 1]$ is an observation function giving the probability to receive $o \in \Omega$ after doing 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 $< S, A, T, R >$. The goal of MDP planning is to find a sequence of actions maximizing the long-term expected reward. Such a plan is called a policy $\pi : S \to A$. An optimal policy π^* specifies for each state s the optimal action to execute at the current step assuming the agent will also act optimally at 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. Solving a MDP is done by Dynamic Programming, with a polynomial time complexity. Solving a Dec-POMDP is done similarly by computing the optimal joint policy. However its time complexity is NEXP-complete [17], that is incredibly hard.

B. Interaction-oriented Models

When faced with real-world applications such as multi-robot systems, Dec-(PO)MDP models are very difficult to apply given their high complexity. Recent advancements in Dec-(PO)MDPs resolution allowed a notable increase in the size of the problems that can be solved. Especially, one interesting direction that has emerged recently is to take advantage of local or sparse interactions between agents. These methods relax the most restrictive and complex assumption that considers that agents interact permanently with all the others. Based on solving a set of interactive individual decision making problems then reduces the complexity of solving Dec-(PO)MDPs. The ND-POMDP model [19] is a static interaction approach, *i.e.* an agent is always interacting with the same subset of neighbors. But this assumption is not realistic. Thus models have been proposed that use dynamic interactions so that each agent interacts with an evolving set of agents. Dec-SIMDP model assumes full local observability, unlimited and 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. Such approaches are a promising direction concerning real-world applications of decentralized decision makers.

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

V. DVF FOR MULTI-ROBOT EXPLORATION

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

A. Interaction Sparse Dec-MDP with DVF

To improve the complexity or solving Dec-MDPs, we proposed an interaction-oriented resolution based on distributed value functions (DVF). DVFs have been 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 team as a separate layer, which currently seems one of the main tracks to tackle scalability in Dec-(PO)MDPs (cf. IV-B). We represent a Dec-MDP with two classes:

- the global interaction class defined as a collection of augmented MDPs $\{MDP_1^{Aug}, ..., MDP_I^{Aug}\}$. There is one augmented MDP per agent that is defined by $MDP_i^{Aug} =< S_i, A_i, T_i, R_i, \Gamma_i >$ where $< S_i, A_i, T_i, R_i >$ 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 inferred locally. It provides the agent with global information, enabling the 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 leads to a significant reduction of 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 interactions. However when situations of interaction occur, DVF does not handle those situations and the local coordination must be resolved separately with another technique. For instance joint policies could 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 :

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

where $P_r(s'|s_{ij})$ is the probability for agent j of 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 one of agent i.

DVF technique allows each agent to choose a goal which should not be considered by the others so as to minimize interactions. The value of a goal depends on the expected rewards at this goal and on the fact that it is unlikely selected by other agents. More details about DVF and its extension under communication constraints can be found in [23].

B. DVF applied to multi-robot exploration

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

1) MDP Model: Each robot generates its local augmented MDP from a four layers grid. The first layer is the real world layer where the robots move. The pixels 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. Hexagons layer is an occupancy grid of hexagons[10]. Hexagons and Voronoi layers are each computed from the pixels layer and used to generate the data structures of the local augmented MDP. States and rewards are based on the hexagonal layer, actions and transitions on hexagonal and Voronoi 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 frontier hexagons, respecting line-of-sight constraints (cf. Fig. 1a). More details about our model are given in [25].

2) DVF: To apply DVF (eq. 2), we consider that robots are homogeneous so the value functions V_j of the other robots j can 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.

3) Decision step: A decision step consists in building the model, computing the policy from the DVF, and producing a smoothed trajectory. The agent plans continuously, executing a decision step as it perceives changes in its environment. We can notice 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, that is the source of the rewards, goes back as the robot moves toward 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 that plans on short-term horizon.

4) Experimentations: Experimental results from simulation and real-world scenarios can be found in [23], [25]. Videos[12] present different exploration tasks with real robots and some interesting situations are underlined as global task repartition or local coordination. These experiments show that this method is able to effectively coordinate a team of robots during exploration. The global interaction class addresses the allocation of exploration goals and also deals with the minimization of close interactions between robots. Each robot computes locally a strategy that minimizes the interactions between the robots and maximizes the explored space.

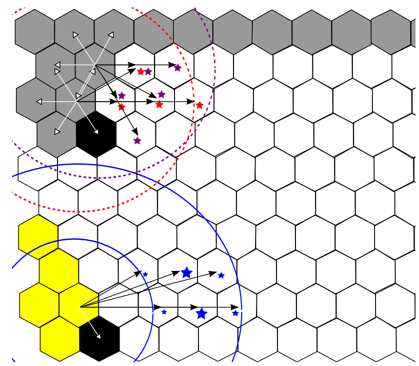


Fig. 1. Reward propagation mechanisms with stars as resulting rewards. a) From frontier hexagons at the top. An unknown hexagon (grey) propagates its reward over a radius (dotted circle) on free neighborhood hexagons (white). Propagation is stopped if occupied (black) or unknown hexagons is encountered.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

VI. Planning to explore and perceive

The first two years of the contest, taking pictures to locate and recognize objects 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.* camera took pictures at a specified rate along the trajectory planned for the exploration. However, this led to poor performance concerning the object recognition results, firstly because some objects in the arena were not photographed. Indeed, pictures taken depend on the trajectory computed for the exploration, so some objects were not covered by photos and in contrast 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 deal with the exploration of the arena but must also take care of the coverage of the explored areas with photos. Indeed, if all the explored space has been photographed, it is ensured that each object is at least on one picture and that the recognition module is going to handle it. In this section, we introduce a DVF extension to improve photo coverage of the space and present some experiments.

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

[11] If the robots are not homogeneous we just need to compute one value function by type of robots

[12] available at `http://liris.cnrs.fr/laetitia.matignon/research.html`

A. DVF extension to improve photo coverage of the space

The decision module must combine at the same time the interest to explore an area and also to cover it with pictures. To consider both the criteria of exploration and picture coverage, we modified the reward function R_i of the augmented MDP. Specifically, we introduced a specific reward for areas explored and covered with photos in addition to the exploration reward. MDPs allow for 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:

$$\forall s_i \in S_i \quad DVF_i(s_i) = (1-\alpha)R_{i,exp}(s_i) + \alpha R_{i,cov}(s_i) + \gamma \max_{a_i \in A_i} \Big(\sum_{s' \in S_i} T(s_i, a_i, s')[DVF_i(s') - \sum_{j \neq i} f_{ij} P_r(s'|s_{ij}) V_j(s')] \Big) \tag{3}$$

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

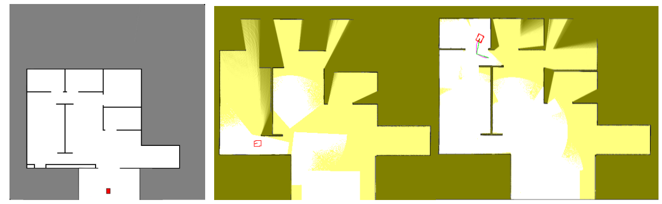
To compute the cover reward function $R_{i,cov}$, a cover grid of pixels counts the number of times each pixel has been photographed. Each time the robot takes a picture, the cover grid is updated by tracing a set of rays covering the optimal recognition area of the camera. Pixels are updated with Bresenham's 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 rewards at the optimal recognition distance (cf. Fig. 1b). This allows the MDP to select the best viewpoint for the next picture. Regarding communications, this new reward implies sending a new data: 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.

We chose not to add an action for taking a picture to keep the computing complexity low. 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. We specify that photos must be taken only when the robot velocity is low, to avoid having blurred pictures where the recognition process will fail.

B. Experimentations

1) Simulated robots: Stage[13] 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 (poorly) simulate the camera and object detection as it can track areas of color in a simulated 2D image, giving the location and size of the color blobs. Thanks to this architecture, the decision module of the real robots can be used with the simulator without modification.

[13]http://playerstage.sourceforge.net/



(a) Simulation environment.   (b) Simulation screenshots.

Fig. 2. In b), explored areas but not yet covered with photos are yellow; explored areas covered with photos are white; non explored areas are grey.

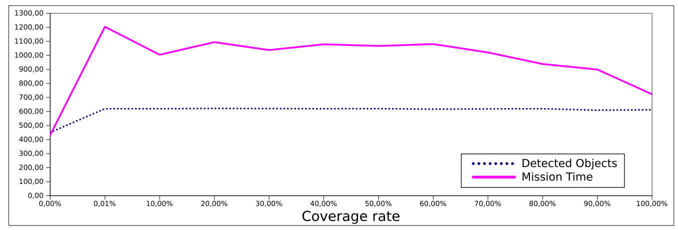


Fig. 3. Number of objects detected and mission time (in seconds) versus coverage rate (produced from 340 simulations).

2) Results: We conducted a set of experiments in autolab environment (cf. Fig. 2a). Fig. 2b shows successive screenshots of one simulation. The robots are initially in the starting zone and objects are regularly positioned in the environment. We compute for each coverage rate, the number of objects detected and the mission time (cf. Fig. 3). With $\alpha = 0$, the mission is the fastest given the robot does not care about taking pictures. The only goal of the robot is to map all the arena, and it only takes few pictures when it stops between two actions. So few objects are 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 remaining time is sufficient. This is illustrated in Fig. 4 with $\alpha = 0.0001$ where the number of detected objects increases while 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 twice the arena and the mission time is high. When α is high, pictures are taken during all the mission so the robot deals with both tasks at the same time. With $\alpha = 1$, we obtain the second fastest mission

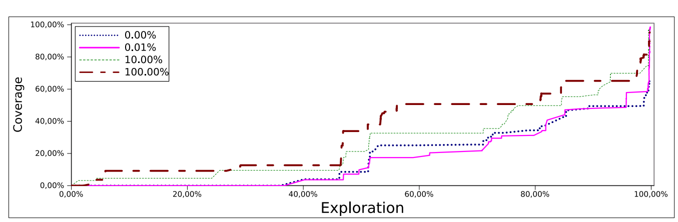


Fig. 4. Percentage of objects detected versus percentage of exploration.

time and all the objects are detected. So this new approach based on exploration and photo coverage of the space allows to photograph all of the objects. α factor must be chosen to balance the priority of exploration versus object detection. For instance if the time is limited and α is high, the robot may not explore the whole map since taking pictures is time consuming but will photograph all objects in the explored area.

VII. Conclusion and Perspectives

We have shown a new algorithm for coordinating multiple robots that explore some unknown area. DVF allow for solving the expensive Dec-MDP 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 chose the right positions to take meaningfull pictures. As a result, we can observe the robots take longer to explore, but also takes much more pictures, efficiently covering the whole space. We can then expect a better detection rate, as any object should be at least on one picture. An immediate perspective is therefore to compare object recognition results with different parameters on real robots to confirm on a real experiment that our method improves the perception.

Another perspective of our work is to plan to perceive by interlacing detection and decision modules to have a 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 Dominant Orientation Templates method [5]. For each object, a set of views is defined, and the method gives a weight for each view that is equivalent to its precision[14]. A positive detection is reliable when the point of view is of high weight. 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 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 easily done by generating specific high rewards at these viewpoints so that the decision module would adapt the planned trajectory so that the robots take pictures from there.

Acknowledgment

This work has been supported by NRA and DGA (ANR-09-CORD-103) and jointly developed with the members of the Robots Malins consortium. It has also been partially supported by 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, O. Simonin, S. Lagrange, N. Delanoue, and F. Charpillet, "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*, 2013, pp. 210–215.

[3] D. Filliat, E. Battesti, S. Bazeille, G. Duceux, A. Gepperth, L. Harrath, I. Jebari, R. Pereira, A. Tapus, C. Meyer, S. Ieng, R. Benosman, E. Cizeron, J.-C. Mamanna, and B. Pothier, "Rgbd object recognition and visual texture classification for indoor semantic mapping," in *Proc. of TePRA*, 2012.

[4] J. Xie, F. Nashashibi, N. M. Parent, and O. Garcia-Favrot, "A Real-Time Robust SLAM for Large-Scale Outdoor Environments," in *17th ITS World Congress*, Oct. 2010.

[5] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *CVPR*, 2010, pp. 2257–2264.

[6] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*, ser. AGENTS '98, 1998, pp. 47–53.

[7] R. Simmons, D. Apfelbaum, W. Burgard, D. a. M. M. Fox, S. Thrun, and H. Younes, "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, and F. Schneider, "Coordinated Multi-Robot Exploration," *IEEE Transactions on Robotics*, vol. 21, pp. 376–386, 2005.

[9] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *Proc. of IROS*, 2008, pp. 1160–1165.

[10] R. Zlot, A. Stentz, M. Dias, and S. Thayer, "Multi-Robot Exploration Controlled By A Market Economy," in *Proc. of ICRA*, vol. 3, May 2002, pp. 3016–3023.

[11] A. Bautin, O. Simonin, and F. Charpillet, "MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration," in *ICIRA*, ser. Lecture Notes in Computer Science, vol. 7507, 2012, pp. 496–508.

[12] M. Julia, A. Gil, and Ó. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, no. 4, pp. 427–444, 2012.

[13] F. Deinzer, C. Derichs, H. Niemann, and J. Denzler, "A framework for actively selecting viewpoints in object recognition," *IJPRAI*, vol. 23, no. 4, pp. 765–799, 2009.

[14] S. Brandao, M. Veloso, and J. P. Costeira, "Active object recognition by offline solving of pomdps," in *Proc. of the International Conference on Mobile Robots and Competitions*, 2011, pp. 33–38.

[15] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An Experiment in Integrated Exploration," in *Proceedings of IROS*, 2002, pp. 534–539.

[16] C. Stachniss, G. Grisetti, and 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, and S. Zilberstein, "The Complexity of Decentralized Control of Markov Decision Processes," *Math. Oper. Res.*, vol. 27, pp. 819–840, 2002.

[18] M. L. Puterman, *Markov decision processes*, 1994.

[19] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo, "Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs," in *Proc. of AAAI*, 2005, pp. 133–139.

[20] F. S. Melo and M. M. Veloso, "Decentralized MDPs with sparse interactions," *Artificial Intelligence*, vol. 175, no. 11, pp. 1757–1789, 2011.

[21] A. Canu and A.-I. Mouaddib, "Collective Decision- Theoretic Planning for Planet Exploration," in *Proc. of ICTAI*, 2011.

[22] J. Schneider, W.-K. Wong, A. Moore, and M. Riedmiller, "Distributed Value Functions," in *Proc. of ICML*, 1999, pp. 371–378.

[23] L. Matignon, L. Jeanpierre, and 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, and A.-I. Mouaddib, "Unknown Area Exploration with an Autonomous Robot using Markov Decision Processes," in *Proc. of TAROS*, 2010, pp. 119–125.

[25] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib, "Distributed Value Functions for Multi-Robot Exploration," in *Proc. of ICRA*, 2012.

[26] R. Bellman, *Dynamic programming: Markov decision process*, 1957.

[27] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[14] or positive predictive value