IROS 2015 Workshop on On-line decision-making in multi-robot coordination (DEMUR'15)
http://robotics.fel.cvut.cz/demur15/

CONCENTRIC AND INCREMENTAL MULTI-ROBOT MAPPING TO OBSERVE COMPLEX SCENES

Jonathan Cohen[a,b], Laetitia Matignon[a,*], Olivier Simonin[b]

[a] *Université de Lyon, Université Lyon 1, LIRIS, CNRS, UMR5205, F-69622, France*
[b] *INSA Lyon, INRIA Chroma, CITI Lab., Université de Lyon, Villeurbanne F-69621, France*
[*] corresponding author: laetitia.matignon@univ-lyon1.fr

Abstract. The observation and recognition of complex scenes can benefit from the use of multiple mobile cameras. In this paper we study a fleet of mobile robots where each robot controls the point of view of its embedded camera. The objective is to manage the cooperation between the robots to find a joint position that maximizes the joint observation of a scene, defined as the activity of one person. It is assumed that the robots can communicate but have no map of the environment and no external localisation. This paper presents a spatial concentric modeling of the environment well adapted to the navigation of two-wheeled non-holonomic robots. To reduce the complexity of finding the best solution in the space search, we propose an incremental mapping based on this model and some heuristics to search for the optimal observation in an online context. Experimental results in simulation are presented that show in particular the anytime aspects of the proposed algorithms.

Keywords: Multi-robot coordination, Mapping, Scene observation, Multi-robot exploration.

1. Introduction

Many robotic applications where it is required to observe or recognize activities of one or more humans must be robust to complex perturbations as occlusions or cluttered environments. Providing multiple points of view on a complex scene enables wider coverage area and can reduce the issue of occlusions. However it could be not possible or difficult to cover a large zone with a set of static cameras, and even impossible to deal with dynamic occlusions or lighting changes.

We propose in this article to address the problem of active distributed recognition with mobile cameras, *i.e.* cameras embedded on mobile robots. The context of this work is the CROME[1] project that concerns human activity recognition with a fleet of mobile robots. Each robot is autonomous and embeds a camera whose viewpoint is controlled by the robot. Possible applications of such systems are human monitoring , emergency assistance, cobotics, . . .

In the CROME project, the objective is to exploit the mobility of the robots so that they adapt their positions to find the spatial configuration that maximizes the joint recognition of a human activity. Active distributed recognition provides the advantages to be robust to individual robot failures and to be able to adapt to dynamic changes in the environment. But the use of multiple mobile robots also requires the robots to communicate and coordinate their movements. The main challenge is providing cooperation between robots when each individual point of view does not allow a satisfactory recognition, *e.g.* because of the presence of occlusions. The joint recognition of a human activity must be done by integrating information collected by the fleet. The robots must coordinate to obtain the most informative and complementary observations. The objective is then for the robots to find a joint position that maximizes the observation and joint recognition of a complex scene *e.g.* human activity.

In this paper, we consider realistic and challenging assumptions about knowledge and perception of the robots in such a task of observation. First, we consider that robots have no map of the environment, they can only know in which direction the scene could be observed. The environment holds obstacles that can prevent some displacements and observations of the scene (*i.e.* occlusions). Second, robots have no external localisation, as a GPS, they can only know their relative position to the observed scene, that is distance and orientation towards the scene. Their perception is limited to a local camera view. As we want to explore cooperation between robots, we assume the robots can communicate between them. The last main assumption concerns the scene to observe. We consider, in this paper, that the scene is defined as the activity of one person, performing a sequence of tasks in a same place. We limit our objective to observe the totality of the body (skeleton identification) and to be able to detect task changing.

Observing under such assumptions, *i.e.* in an unknown environment, requires to explore and/or build a map of the environment. Moreover, optimizing the observation requires to move and coordinate the robots while the task of the person could change. In this paper we focus on dealing with the complexity involved by the combination of these different dimensions: the environmental constraints, the number of robots and

[1] Coordination of a mobile robots fleet for multi-view analysis of complex scenes.

the changes of the scene. In order to deal with such a challenge, we propose to explore an incremental mapping of the environment, in parallel of dealing with the observation task. As we want to avoid long processes of exploration or learning, we propose to study anytime algorithms, as they can always give a solution which can be improved by the time.

In the following, we first discuss related works on tracking and recognition with mobile cameras. Section 3 presents the problem modeling and its complexity. Then we detail the incremental mapping we propose to represent the environment and the heuristic approaches we used to search for the optimal joint positions of the robots. Section 6 presents experiments and results. Finally, section 7 concludes this paper and proposes some perspectives.

2. Related Work

There has been an increasing amount of research over the last decade on using a network of fixed cameras to detect, track and recognize objects or persons [1, 2]. For instance, merging the information provided by several cameras based on 3D occupancy grid can be used to track and detect falls of elderly people at home [3]. Communication to obtain a consensus in a distributed camera networks can also be used to perform human activity recognition [4]. Although a set of fixed cameras can obtain global views of the scene, as cameras are static, they cannot deal with non-covered zones or occlusions.

Recent works are interested in using mobile cameras that can move to adequate places to cover blind spots not observed by any fixed camera and react to changing conditions as lighting or dynamic obstacles. In the URUS[2] project [5], the objective is to assist and guide people in urban settings by combining the information from on-board cameras on mobile robots and a set of fixed surveillance cameras. Sensors embedded in the environment can complement the narrow perception of the robot's camera, and the mobile robot can cover places occluded from the camera network. In this project, the authors focus on the fusion of sensory information from the different sensors to overcome tracking failures [6] and to manage active cooperative perception [7]. Active perception means that the robot selects actions taking into account their effects on its sensors, in particular to improve their performance. Cooperative active perception involves the fusion of information from multiple sensors and multiple cooperating decision makers. However these works only focus on the cooperation between one robot and the fixed cameras, and do not consider the issue of coordinating multiple mobile robots to improve the perceptual information available to the system.

In Giusti et al. work [8], a group of mobile robots is used to cooperatively sense and classify an entity of interest. The proposed scenario is the distributed visual recognition of hand gestures. The swarm positions itself along a semi-circle centered on the target scene. Robots, equipped with cameras, process hand images from multiple points of view. Then each robot classifies individually the hand shape and a distributed consensus protocol allows the swarm to reach as a whole a final decision about the issued gesture. This work focuses on the distributed recognition and information sharing between robots. The navigation and coordination of the robots is very simple and is not used to improve the recognition. Indeed, although the robots are mobile, once they are positioned uniformly in a semi-circular arc, they maintain this formation. Moreover, the environment is assumed to be without obstacles, which facilitates the navigation.

[2]Ubiquitous Networking Robotics in Urban Settings.

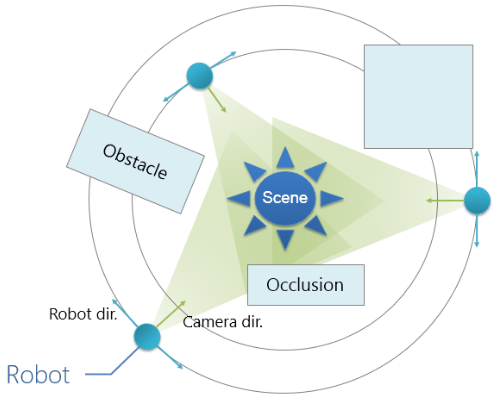


Figure 1. Navigation model based on circles

It has been shown, in many works, that using efficient coordination processes in a fleet of mobile robots equipped with sensors can improve the exploration and the mapping of an unknown environment, see *e.g.* [9–11]. However, this work does not aim to track or recognize the activity of a human person but to create a map of the environment by using the information of multiple mobile sensors.

3. Problem modeling

We first present a spatial concentric modeling of the environment, which is well adapted to the navigation of 2-wheeled robots and reduces the spatial complexity of our problem. We then define robots observation and the quality of the (joint) observation.

3.1. Navigation around the scene

The robots need to navigate through the space around the scene to find an optimal joint position for the observation. The navigation of the fleet requires to consider several navigation constraints. Each robot has to avoid obstacles while moving, and to coordinate with others to avoid collisions. Moreover, robots must track the scene and keep it constantly in their field of views, while remaining some distance away from the scene to observe it as a whole.

In order to reduce the complexity of dealing with these different constraints, we propose to consider a

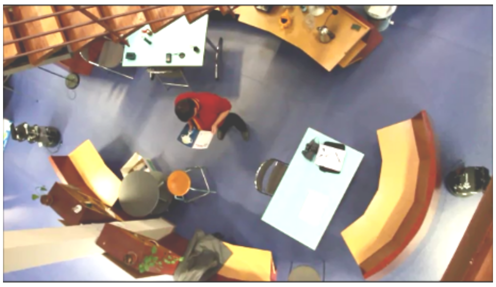


Figure 2. Turtlebot navigation on circle around a scene, see the video link `http://liris.cnrs.fr/lmatigno/videoDemoCROME1.html`.

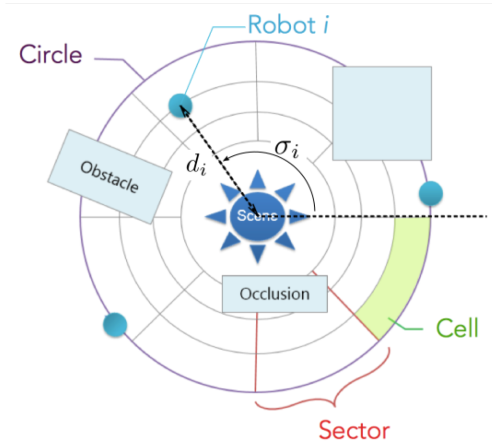


Figure 3. Spatial discretization in cells with $C = 3$ circles and $K = 8$ sectors.

limited navigation space around the scene composed of concentric circles centered on the scene. So we define several circles, at spaced radius, that robots follow to move in two directions (forward/backward), as illustrated in fig. 1.

The primary interest of this approach is to maintain the orientation of each robot's camera towards the scene. This requires to fix the camera orientation perpendicular to the forward direction of the robot. The second interest is that moving along a circle trajectory is easy for a mobile robot, in particular for a two-wheeled non-holonomic robot. Finally, considering circles simplifies the interaction and coordination of the robots and reduces the risks of collisions. Indeed, no collision can arise between robots navigating on different circles, and collisions on a circle are easy to predict and avoid. Concerning obstacles, robots have only to consider obstacles that are on circle trajectories. It is then possible either to bypass the obstacle and continue the circular navigation, or to limit the navigation on circular arcs delimited by obstacles. Such a navigation model has been experimented with Turtlebot 2 robots, as illustrated in fig. 2.

Given the navigation model based on circles that we have described previously, we can define the position of a robot.

Definition 1. *The* ***position*** *of a robot i is defined by (d_i, σ_i) where d_i is the distance of the robot i to the scene, and σ_i is the angle between the horizontal line passing through the scene and the line connecting the scene to the robot.*

This is illustrated in fig. 3. In order to reduce the spatial complexity due to the navigation in a continuous environment, we propose a discrete representation of the positions of the robots. To this end, we define a set of C concentric **circles**, at spaced radius and with diameters within the range $[D_1, D_2]$; and a set of K **sectors**. Sectors allow to divide the circular space centered on the scene, into slices with identical central angle of $\frac{2\pi}{K}$. With these elements we can determine a set of contiguous cells where the robots are moving to explore and observe the scene (cf. fig. 3). At any position (d_i, σ_i) of a robot i is associated a unique **cell** $c_i = \langle [d_a, d_b]; [\sigma_a, \sigma_b] \rangle$ such that $d_i \in [d_a, d_b[$ and $\sigma_i \in [\sigma_a, \sigma_b[$. As well, at any cell $c = \langle [d_a, d_b]; [\sigma_a, \sigma_b] \rangle$ is associated a unique position $(d_a, \frac{\sigma_a+\sigma_b}{2})$. The number of circles and sectors will affect directly the complexity of the discretization in terms of number of cells to store and to explore, that is discussed later.

3.2. Quality of the observation

The observation of a robot i is the information about the scene that is perceived by the robot. We define $o_i(c)$ the **observation** of the robot i when it is in the cell c, *i.e.* at the position associated to the cell c.

In this work, we are interested in the observation of human activity. In a first stage, we limit our objective to observe the totality of the body, that can be characterized by skeleton data obtained from sensors like Kinect camera. Skeleton data defines a set of body joints, the positions and orientations of each joint in the referenced frame of the camera, and a confidence value for each joint equal to 1 when the tracking of the joint seems to work and 0 if the tracking fails and the output is uncertain. So we choose to represent the robot's observation of the scene by a binary vector of size equal to the number of body joints tracked by the camera.

Definition 2. *The* ***quality of an observation*** *o_i made by a robot i is defined as:*

$$q(o_i) = \sum_{j=1}^{|o_i|} o_i^j \tag{1}$$

where o_i^j is the jth element of the vector o_i.

The quality of an observation is the number of body joints accurately tracked by the robot. The following example illustrates this computation:

$$o_1 = [1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0], \ q(o_1) = 7$$

The quality varies depending on the pose of the human body (*i.e.* from side-on), the occlusions or lighting changes. By extension we define the **quality of a cell** as the quality of the observation made from the position associated to that cell. This determines if the view angle from that cell enables to observe accurately the scene.

The quality of an observation is a local information. To quantify the quality of the observation made by the group of robots, we define the quality of the joint observations.

Definition 3. *The quality of the joint observation, or* ***joint quality*** *noted Q, is defined as the quality of the component-wise maximum of the observation vectors of the robots:*

$$Q = q(\bigvee_{i \in [1,N]} o_i) \tag{2}$$

where N is the number of robots and $\bigvee$ is a logical OR between the elements of observation vectors of the robots.

The objective is then for the robots to find the joint positions that maximize the joint quality. A key point is that maximizing the joint quality is not decomposable into maximizing the individual quality, which could lead to redundant information, but it requires to find the best complementarity of information. For instance two robots which maximize their individual qualities can have a low joint quality because they observe identical joints; and two robots with non optimal individual qualities can have a high joint quality if the individual observed joints complement one another.

3.3. Complexity

The state space to explore is the set of joint positions of the robots to find those that maximize the joint quality. This set is bounded[3] by $(C \times K)^N$ where C, K and N are respectively the number of circles, sectors and robots. Some examples of the size of the state space are given in section 6. It is clear that the state space to explore is exponentially large with the number of robots. A complete exploration can take a very long time in simulation (cf. §6) and is barely impossible to consider with real robots.

To handle this space complexity and avoid long processes of exploration, we define a strategy combining two approaches. We consider initially few cells that will be divided incrementally; and we define heuristics to guide the exploration in the state space.

4. Incremental mapping

One assumption of our work is that the environment is not known in advance by the robots, so they have to explore it and to build a map to use data gathered during their exploration. One common technique for map representation is to use **occupancy grid maps**. Introduced by Elfes [12], an occupancy grid represents the environment with a set of cells (usually squares or hexagons), each with an occupancy probability that determines the probability that the cell will be occupied by an obstacle. The occupancy probability is initialized to 0.5 for each cell and updated by robot's sensor readings that indicate whether the robot observes the cell as occupied or unoccupied. In our work, the occupancy grid is based on the cells derived from our discretization in circles and sectors (cf. §3.1). To each cell of the map is associated :

- an occupancy probability
- a quality of the observation made from that cell.

The robots have to explore the cells to update occupancy and quality information of each cell.

We propose in this work to build a map by an **incremental division** of the cells. The idea is to have at the beginning a coarse representation of the environment, with few circles and sectors that define initial cells. Thus each initial cell covers a large area of the space around the scene, as illustrated on the image at the top left of the fig. 4 where three initial cells are defined at the beginning by one circle and three sectors. Then the robots can divide each cell into sub-cells, and can recursively split the obtained sub-cells (cf. fig. 4). As each cell is associated with a position, this will increase the number of accessible positions. The objective is to refine the discretization only in interesting areas of the environment. The robots will

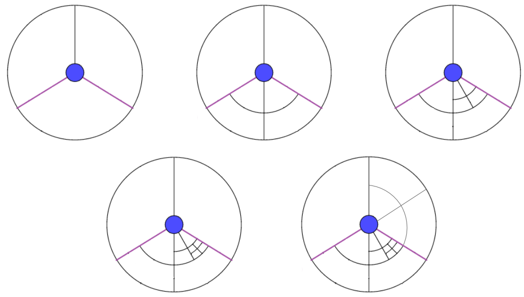


Figure 4. Example of a representation with three initial cells and their successive divisions. The blue circle at the center represents the scene.

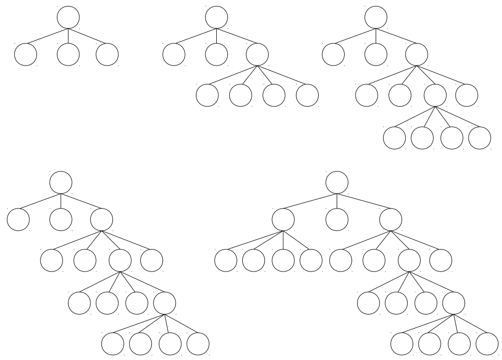


Figure 5. Example of the incremental construction of our tree structure given the divisions of the fig. 4. Each of the three initial cells is a root of a quadtree.

[3]It can be reduced if some cells are inaccessible because of obstacles and if only one robot per cell is allowed.

split cells where an accurate exploration could improve the joint observation quality. Conversely, some cells must not be splitted too finely, *e.g.* because they are behind obstacles. The observation of the scene from any positions in such a cell is occluded.

There are several interests in building such an incremental representation. The major one is to handle the space complexity and the time to explore the environment, by limiting the number of cells to explore. The quality of the joint observation of the scene is also refined over time as the robots explore more and more points of view.

To manage the incremental mapping of the environment while building an occupancy grid we define an appropriate data structure which is based on a **quadtree**. A quadtree is a kind of tree in which each non-leaf node has four children [13]. In our structure each initial cell is a root of a quadtree that will be built recursively. Each cell is stored as a node of the quadtree and each cell can be again divided into four cells, and so on, as illustrated in fig. 5. To store and update the occupancy probability of each cell, we use a **probabilistic quadtree** as in [14] where each node (or cell) of the quadtree has an occupancy value. The occupancy value of leaf nodes is initialized from the values of the parents' nodes and updated by robot's sensor readings. The occupancy value of non-leaf nodes is the mean of the values of the four children nodes.

5. Heuristic approaches to search for optimal observation

While the robots can build a representation of the environment, we need an algorithm for the robots to explore and find a joint position that maximizes the observation joint quality.

As the scene to observe is dynamic we propose to move only one robot at a time. This allows the group to better qualify the scene dynamic. If several robots move together, it will be difficult for them to know if the changes observed about the scene are caused by the scene or by the modification of their point of views. In contrast, if one or more robots remain stationary while another is moving around the scene, then it will be easier for the group to detect potential changes in the scene as a change in the activity.

In the following, we will explain the two major steps of our approach to find the optimal observation. The first one is to choose which robot must move by computing a heuristic based on the marginal contribution of each robot. The second step is to determine which action must be done by this robot, moving or spliting a cell, following an exploration heuristic.

5.1. Marginal contribution of a robot

To choose the robot that has to move, we introduce the notion of *marginal contribution* of each robot, in reference to the marginal contribution of a player to a coalition in the Shapley value [15].

Definition 4. *The **marginal contribution** of a robot i in the joint observation of N robots is noted w_i and is defined as:*

$$w_i = q(o_i) - q(o_i \cap \bigvee_{j\in[1;N],j\neq i} o_j). \tag{3}$$

This corresponds to the part of the observation that robot i is the only one to see.

Consider the following example, where three robots have the respective observations, qualities and marginal contributions:

$$\begin{aligned} o_1 &= [1,1,1,1,1,1,0,1,0,1,1,1,1,1,0],\ q(o_1) = 12 \\ o_2 &= [1,0,1,1,1,1,0,1,0,1,1,1,1,1,0],\ q(o_2) = 11 \\ o_3 &= [0,0,0,0,0,0,1,1,1,0,0,0,0,1,1],\ q(o_3) = 5 \\ & w_1 = 1,\ w_2 = 0,\ w_3 = 3 \end{aligned}$$

The robot 3 is situated in a cell with a low individual quality (5), but it is the only one robot in the fleet that observes some parts of the scene, so its contribution is high ($w_3 = 3$). Even if the robots 1 and 2 have high qualities and a good visibility of the scene, their contributions are low because they observe identical parts of the scene. So their observations are redundant and the robot 2, which have the lowest contribution, should move to find a more complementary point of view to improve the joint observation of the team.

Indeed, a key point is that maximizing the joint quality is not decomposable into maximizing the individual quality, which could lead to redundant information, but it requires to find the best complementary information.

This analysis motivated the definition of an heuristic to select the robot that must move. We simply **select the robot with the lowest marginal contribution**. Moving only this robot offers the advantages of minimizing the decay in the current joint quality and maintaining the group configuration stable.

5.2. Exploration heuristics

Given the space complexity of our problem (cf. §3.3), we propose to use some metaheuristics to efficiently explore the state space while escaping local optimum solutions.

Once the robot with the lowest contribution has been chosen, we use a metaheuristic to make a trade-off between exploration and exploitation. Exploration consists in moving to unknown cells to advance the mapping while exploitation consists in moving to known cells to optimize the joint observation. An **action of exploration** is for a robot to visit an adjacent and unknown cell to gain new information and to avoid possibly the team remaining in a local optimum. An **action of exploitation** is for a robot to move to the best adjacent cell[4], or to split its current cell if it is already the best one. Thus the cells that are

[4] The cell that maximizes the joint quality.

potentially interesting for the observation of the scene are divided and explored more accurately.

In this paper, we consider two standard and well-known metaheuristics, which are Simulated Annealing (SA) [16] and Tabu Search (TS) [17]. At each step of SA, the exploitation/exploration trade-off is calculated with a probability that depends on a temperature parameter, which is gradually reduced during the process. TS performs a pure exploitation but uses the history of the search to navigate through the search space. It keep tracks of a short-term set of the last visited cells in a tabu list, which is used to forbid the visit of the cells in the tabu list.

Algorithm 1 presents the main steps of our approach, where the metaheuristic is one of the input data. This version of the algorithm assumes that communications between robots are perfect and unlimited.

Algorithm 1: Search for the optimal observation

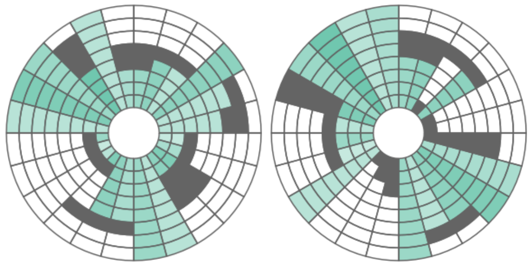
Data: The set of N robots, the number of steps T, a metaheuristic
Result: Q^* the best joint quality found
$Q^* \leftarrow 0$ $t \leftarrow 0$
while $t<T$ **do**
  $t \leftarrow t+1$
  for $i \in [1,..,N]$ **do**
    Recover o_i
  Compute the joint quality Q
  $Q^* \leftarrow max(Q, Q^*)$
  for $i \in [1,..,N]$ **do**
    Compute w_i
  $weakRobot \leftarrow \arg\min_{i\in[1,N]} w_i$
  $weakRobot$ chooses an action (move or split)
  $weakRobot$ executes the selected action

It is is composed of three main steps: (1) determine the observation the robots make of the scene. This allows to compute the joint quality of their joint position; (2) choose, among all robots, which one is the least usefull; that is, the one without whom the joint quality would be the least degraded; (3) move this least usefull robot, according to one of the metaheuristics presented above.

6. Experiments

6.1. Simulator

To conduct experiments, we first designed a simulator. The aim of this simulator is twofold: (i) to allow to run a large quantity of experiments in order to test the validity of our approach, (ii) be realistic enough to properly modelize key features of real mobile robots and real environment. On one hand, assumptions were made to simplify the implementation, *e.g.* we consider that robots' motion around the scene is perfect – that is, robots can move along circles without trajectory errors. We also assume that robots are equipped with sensors allowing them to remotely



(a) Environment A. (b) Environment B.

Figure 6. Two different environments.

	Env. A	Env. B
Best possible joint quality	15	15
# Circles	8	8
# Sectors	24	24
# Cells	192	192
# Obstacles	27	31
With 3 robots :		
# Possible joint positions	1,949,476	1,848,447
# Optimal joint positions	35,457	6,608

Table 1. Configuration of environments A and B.

detect nearby obstacles. Communications between robots are also supposed to be instant and errorless. On the other hand, we simulate noise in sensor (camera) information when performing the perception task of the robots. The observation made by a robot will thus varies from the corresponding real scene.

We generate the observation vectors, *i.e.* skeleton information composed of 15 body joints, by using a technique called *ray tracing* [18]. Virtual rays – one per sector, coming from the center of the scene – carry the observation vectors and assigns to each cell of a same sector a common observation – while considering that obstacles may prevent the light coming from the scene to reach the sensors of a robot. Fig. 6 shows examples of ray tracing in environments containing some obstacles. They also show cells from which the scene is visible (green cells), cells from which the scene is not visible (white cells) and cells containing obstacles (black cells). Different shades of green indicate different local qualities of observation: the greener the cell, the better the local observation. The ray tracing is designed in such way that it's not possible for one single robot to find a cell from which it can see the full joint observation.

Once the ray tracing is generated, it remains fixed during the execution of the exploration algorithms. Note that the robots don't have direct access to the generated environment: it is only used to assign observation vectors and deal with obstacles. Instead of that, they build their own representation of the environment as their exploration of the space goes on.

6.2. Experimental setting

We perform our experiments in two different environments, presented in fig. 6. Experiments are done using 3 robots and 3 initial cells. We neglect noise[5] in observations. Table 1 provides some information concerning the state space of each of those environments, illustrating the difficulty of the problem.

We measured that the exhaustive search of the optimal joint positions takes about 3.40 minutes for each environment[6]. At first glance, it seems that environment B is more difficult than environment A.

The size of the tabu list is 5. The temperature for SA algorithm is initially fixed to 0.6, with a decreasing rate of 1%. We run each algorithm 100 times. We evaluate the quality of our algorithms by measuring the number of times each algorithm reaches the best possible joint quality among those 100 experiments. One experiment consists of either 100, 200 or 300 steps. One step corresponds to an action, that is a robot move or split a cell. The two exploration heuristics are compared to a random algorithm, in which a robot and its action are randomly chosen at each step.

6.3. Results

A video showing the incremental exploration and mapping of the robots can be found at http://liris.cnrs.fr/lmatigno/videoDemoCROME2.html.

Fig. 7 and 8 present our results for experiments in environments A and B. In both of those, our approach with TS and SA algorithms outperforms the random algorithm. In environment A, all three algorithms tested behave the same way: the more steps they are given, the more likely they are to find the best possible quality. For instance, TS finds the best possible quality 47 times out of 100 experiments (47% efficiency) with only 100 steps. But with 300 steps, its efficiency goes up to 81%; while random goes from 28% to 53%. SA performs similarly as TS, although a little less efficiently.

As intuited, environment B is found to be slightly more difficult, as a random exploration hardly finds good joint positions, even with a large number of steps. Actually, unlike environment A, randomly wander in the space of environment B does not insure that, with more steps, we will get a better solution. This can be explained by the presence of structures looking like walls, preventing from going from one area to an other without having a good exploration strategy. Yet, SA and TS reach efficiencies close to the ones obtained in environment A. More, in just 100 steps, SA finds every time a best joint quality equal to 14 or 15, while it is the case in only about 50% of the experiments with the random algorithm. This illustrates the anytime aspect of our approach, where TS and SA, unlike random, find a good joint quality even in few steps and improve their results with more steps.

Finally, Table 2 presents the average length (steps), for each algorithm, to find for the first time the best possible joint quality. It clearly appears that TS is the most efficient heuristic – among those tested – as it is the fastest to find the optimal joint quality in both environments. Note that performing 196 actions is very few compared to the state space size, and it represents also a reasonable time for experiments with real robots.

[5] As noise can randomly improve the quality of a joint observation, taking it into account would skew our results.

[6] Using a 2.4GHz Intel Xeon E5 Quad-Core - 2GB RAM.

	Env. A	Env. B
SA	478.18	703.16
TS	**258.6**	**196.03**
Random	720.24	1034.91

Table 2. Mean steps before finding the best possible joint quality for the first time.

7. Conclusion

We presented an original spatial concentric modeling of the environment, serving as a base for an incremental mapping. This incremental mapping allows to explore promising areas of the environment while keeping complexity reasonable. We also introduced the notion of *marginal contribution* of a robot, representing how useful a robot is for the team. This notion can easily be extended from the frame of the observation task to a much larger multi-agent framework.

This early work and results reveal many perspectives. First we would like to test how our method cope with noise and errors in robot movements, localization and perception. We intend also to test the ability of the system to detect activity changing. We proposed to move only the robot with the lowest contribution to better qualify the dynamic of the scene. The robots that remain stationary will have to find a consensus about whether or not the activity has changed. Once they agreed on a change, exploration of the cells must be thrown again *e.g.* by increasing the temperature parameter in the simulated annealing method. Another perspective is to implement our method on real robots. We have already experimented an exhaustive search of the best joint quality with a navigation model based on circles (cf. fig. 2). We intend to extend this work by adding an incremental mapping of the environment and the use of our heuristics to search for the optimal observation in an effective and online way.

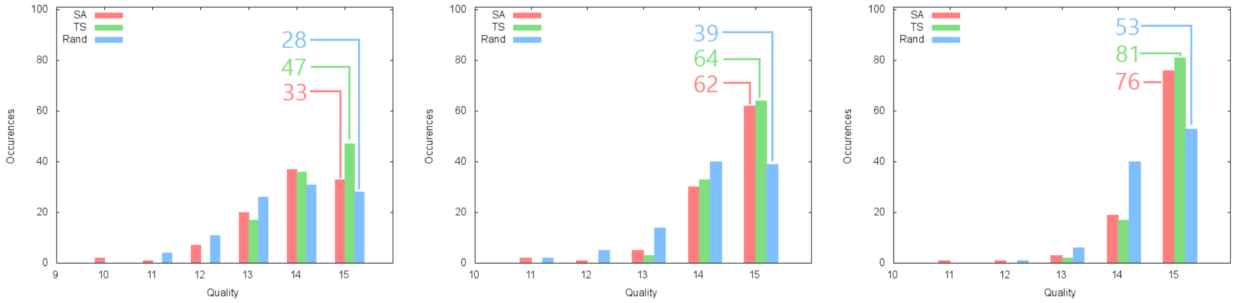
Acknowledgements

This work has been supported by the INSA Lyon Foundation CROME project.

References

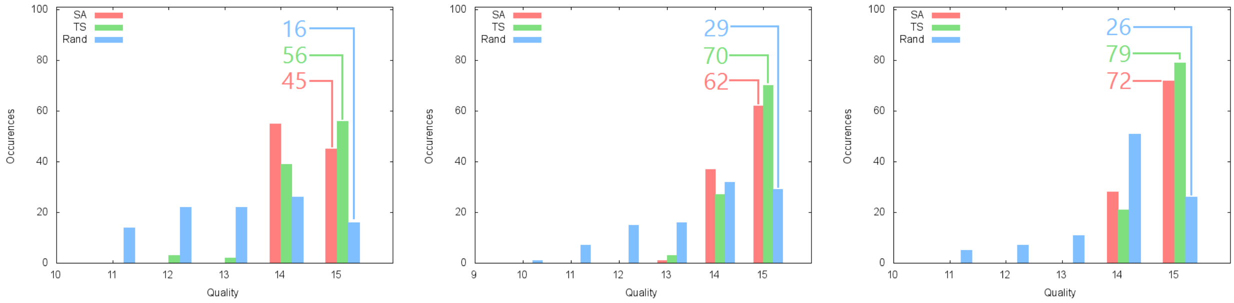
[1] M. M. T. Michael B. Holte, Cuong Tran, T. B. Moeslund. Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal of Selected Topics in Signal Processing* **6**(5):538 – 552, 2012. doi:10.1109/JSTSP.2012.2196975.

[2] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recogn Lett* **34**(1):3–19, 2013. doi:10.1016/j.patrec.2012.07.005.



(a) 100 steps. (b) 200 steps. (c) 300 steps.

FIGURE 7. Environment A. Number of times each quality is the best quality found by the team, over 100 experiments.



(a) 100 steps. (b) 200 steps. (c) 300 steps.

FIGURE 8. Environment B. Number of times each quality is the best quality found, over 100 experiments.

[3] A. Dubois, F. Charpillet. Tracking Mobile Objects with Several Kinects using HMMs and Component Labelling. In *Workshop Assistance and Service Robotics in a human environment, International Conference on Intelligent Robots and Systems*. 2012.

[4] E. Adeli Mosabbeb, K. Raahemifar, M. Fathy. Multi-view human activity recognition in distributed camera sensor networks. *Sensors* **13**(7):8750, 2013. DOI:10.3390/s130708750.

[5] A. Sanfeliu, J. Andrade-Cetto, M. Barbosa, et al. Decentralized sensor fusion for ubiquitous networking robotics in urban areas. *Sensors* **10**(3):2274–2314, 2010. DOI:10.3390/s100302274.

[6] L. Merino, A. Gilbert, J. Capitan, et al. Data Fusion in Ubiquitous Networked Robot Systems for Urban Services. *Annals of Telecommunications, Special Issue Ubiquitous Robots* **67**, 2012. DOI:10.1007/s12243-012-0311-1.

[7] M. T. J. Spaan, T. S. Veiga, P. U. Lima. Active cooperative perception in network robot systems using POMDPs. In *Proc. of IROS*, pp. 4800–4805. 2010. DOI:10.1109/IROS.2010.5648856.

[8] A. Giusti, J. Nagi, L. Gambardella, G. A. Di Caro. Cooperative sensing and recognition by a swarm of mobile robots. In *Proceedings of the 25th IROS*, pp. 551–558. 2012. DOI:10.1109/IROS.2012.6385982.

[9] M. Julia, A. Gil, O. 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.

[10] L. Matignon, L. Jeanpierre, A. Mouaddib. Distributed value functions for multi-robot exploration. In *Proceedings of ICRA*, pp. 1544–1550. 2012. DOI:10.1109/ICRA.2012.6224937.

[11] A. Bautin, P. Lucidarme, R. Guyonneau, et al. Cart-O-matic project : autonomous and collaborative multi-robot localization, exploration and mapping. In *5th IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pp. 210–215. 2013.

[12] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer* **22**(6):46–57, 1989. DOI:10.1109/2.30720.

[13] R. A. Finkel, J. L. Bentley, R. A. Finkel, J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica* **4**(1):1–9, 1974. DOI:10.1007/bf00288933.

[14] G. K. Kraetzschmar, G. P. Gassull, K. Uhl, et al. Probabilistic quadtrees for variable-resolution mapping of large environments. In *Eds., Proceedings of the 5th IFAC/EURON*. 2004.

[15] S. Hart. Shapley value. In S. N. Durlauf, L. E. Blume (eds.), *The New Palgrave Dictionary of Economics*. 2008. DOI:10.1057/9780230226203.1517.

[16] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. Optimization by simulated annealing. *Science* **220**(4598):671–680, 1983. DOI:10.1126/science.220.4598.671.

[17] F. Glover, M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[18] A. Appel. Some techniques for shading machine renderings of solids. In *Proceedings of Spring Joint Computer Conference*, pp. 37–45. ACM, 1968. DOI:10.1145/1468075.1468082.