

Exploration of Unknown Environments with Humanoid Robots

R.Gelin, E.Wirbel, A.Mazel, N.Garcia, L.Souchet,
A-Lab
Aldebaran Robotics
Paris, France.
rgelin@aldebaran-robotics.com

H.Le Borgne, P.Morignot¹, C.Panah
CEA, LIST,
Gif sur Yvette, France.
herve.le-borgne@cea.fr

¹ Now INRIA, Rocquencourt, France

Abstract — This paper presents the work with a team of three humanoid robots exploring unknown domestic indoor environments. This work takes place in the context of a French robotic challenge, CAROTTE. Each robot autonomously uses its on-board sensors to explore the environment and to avoid encountered obstacles, while the team spreads in different directions to optimize the exploration. The gathered information is sent to a centralized server which merges them in a global map and performs object detection. The presented framework is designed to answer the constraints of the platform and the environment, resulting in a robust exploration and mapping system for a team of humanoid robots.

Keywords—humanoid robotics, exploration, mapping, localization, vision, integration

I. INTRODUCTION

Exploration of an unknown indoor environment with autonomous robots remains a major goal in Robotics. Having a robotic help for mapping an apartment, recognizing and localizing dedicated objects in it would lead to many civil and military applications. Towards this goal, the French National Research Agency (ANR) and the French Defense Procurement Agency (DGA) have together organized and sponsored the challenge CAROTTE (the French acronym means Mapping of a Territory with Robots). The goal of this challenge is to have one or several mobile robots mapping a 120m² indoor environment, recognizing objects in it and coming back to their initial entry point, all in 30 minutes. The challenge arena consists in a simplified domestic environment, containing rooms and everyday life objects. Five teams were competing in the challenge, with three evaluation sessions over three years. In this paper, we describe the YOJI team (the French acronym stands for Eyes, Ears and Legs for Inspection), composed of Aldebaran Robotics (manufacturer of humanoid robots), CEA LIST (French Atomic Commission) and Voxler (company specialized in sound processing).

Our exploration platform was the humanoid robot NAO, developed by Aldebaran Robotics, a flexible and affordable robotic research platform. The objective of Aldebaran Robotics is to provide robot companions for domestic applications. In that context, the ability to navigate in his new home will be

crucial for the domestic robot. That is why Aldebaran participated to the CAROTTE Challenge.



Fig. 1. The NAO robot with the laser head

Originally the NAO robot is a 58cm high humanoid robot equipped with 25 joints, 4 microphones, 2 cameras, 2 ultrasonic sensors and tactile and pressure sensors on the feet. Because localization and mapping based on these sensors only would have been too complicated, especially for the embedded CPU (ATOM 1,6 GHz) of NAO, we decided to add a laser range finder (Hokuyo, 5m range) on the top of his head. The use of 3 NAO for the exploration was a way to compensate the slowness of his bipedal walking (0,2m/s) compared to speed of all the other wheeled competitors. A remote PC, used as a server, is connected to the three NAO with a Wi-Fi link.

In this paper, after a literature survey (section II), we present an overview of the system used for the exploration in section III. In section IV, the principles of the navigation of each robot are explained. Section V introduces the innovative algorithms used for object recognition. Finally (section VI), we sum up our contributions and highlight directions of future work. The work of Voxler about sound recognition has been left out of this paper. Even if some good results have been obtained and if it appears that audio interaction is a very important feature for companion robots, we have preferred to stick to classical features of exploration in this paper.

II. RELATED WORK

The generic problem of exploring an unknown environment is called Simultaneous Localization And Mapping (SLAM). There are two kinds of approaches : metric and topological. Metric SLAM uses some quantitative metric sensors such as sonar arrays, laser sensors or more recently 3D cameras. The most common approaches, as described for example in [1], are occupancy grids [2], Kalman filter based approaches and particle filters [3]. Topological SLAM does not provide metric information, but constructs a graph where nodes correspond to places and links define possible paths. For example, [4] presents a topological navigation system where places are learned and recognized using camera information. The hybrid approach combines the advantages of both: scalability and easy semantic labeling combined local metric information, as explained for example in [5]. This is particularly adapted to the case of this challenge, because the robots are equipped with a metric sensor and they are navigating in a large environment, where rooms can be associated to a node.

Multi-robot exploration is an interesting case of SLAM. This setup makes it possible to combine the gathered information, making the whole system more robust, and to optimize the exploration strategy. For example [6] builds a global map which copes with high odometry errors and reduces the exploration time of each robot. Since the odometry errors on the NAO robots are quite high and the walk is much slower than a typical wheeled robot, using this sort of approach is extremely interesting in our case.

The NAO robot is not a classical platform for performing exploration. It has specific constraints due to its humanoid walk, which makes the exploration slower and the integration of new sensors harder (because of the added weight). Localization systems have already been developed on NAO. [7] and [8] use metric data built from a laser or a depth camera to localize the robot relatively to a known 3D model of the environment. [9] performs SLAM for the constrained environment of the RoboCup to make NAO robots cooperate and play soccer. However, both these approaches have strong preliminary hypotheses about the environment (3D model available, beacons), which is not the case in this challenge. Thus our framework does not make any assumption on the environment.

III. OVERVIEW OF THE SYSTEM

The principle of the architecture is to have each NAO as autonomous as possible to prevent the loss of the Wi-Fi connection with the remote PC, while still allowing to merge the information from the different robots.

Each NAO is autonomously able to walk, to avoid obstacles detected by ultra-sonic and tactile sensors and by the laser and to recover from possible falls. Using the laser, NAO is able to build a map of his environment and localize himself while walking. The exploration software, based on the mapping, is embedded as well on each NAO. This local architecture allows each robot to explore safely his environment and to come back to his starting point with a map of the explored zone without any assistance from the remote PC. If the Wi-Fi connection with the PC is available, each

NAO sends the current version of the environment map, the pictures and the sound he obtained from cameras and microphones to the PC. If the PC recognizes objects or sounds, it sends the identifier back to the NAO who takes it into account.

The role of the remote PC is then to get information from the three NAO: pictures, sounds and maps. Based on a database of pictures and sounds, the identification of objects in the pictures and of audio events in the sounds is realized. The identified items are sent back to the robot who detected it. Collecting the maps from the three robots, the PC merges them and processes them to build a complete map. This complete map may then be sent to each robot for the optimization of the exploration task (see section IV.D).

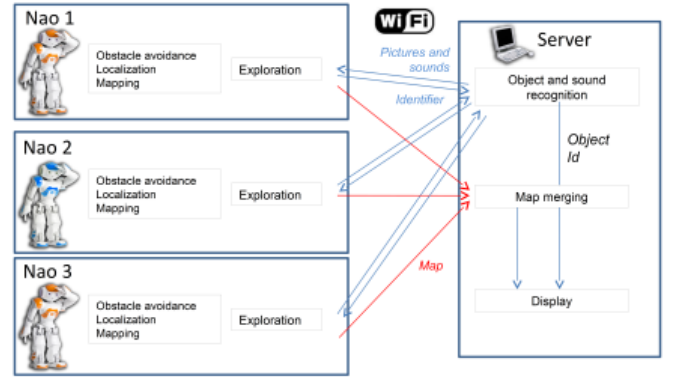


Fig. 2. Client-Server architecture for a team of 3 robots

To perform an exploration, each robot scans the environment with the laser and the cameras before moving. The pictures are then sent to the remote PC. Using the laser scan, the map of the environment is consolidated and new points to explore are computed. The most relevant point to explore is selected and NAO walks towards it. While walking, NAO uses the laser information to localize himself in his map. The obstacle avoidance uses the information coming from all the available sensors as described in section IV. For most of the sensors, the information is processed on board. But the pictures taken by the camera are processed remotely on the PC because of the limitation of the embedded CPU.

IV. NAVIGATION

A. Exploration

The principle of the exploration is to make the robot explore detected points of interest. A point of interest is characterized by a discontinuity in the laser scan performed by the robot when he stops. Such a discontinuity generally corresponds to an open door or a dead angle in the field of view. The robot selects a point of interest among the list he has collected so far. He moves 50cm towards this point of interest, then stops. A new scan is completed to update a tree of points of interest to explore.

The selection of the point of interest in the list is based on a heuristics, impacting the behavior of the robot. For example, if the robot goes systematically to the latest discovered point, this will be inefficient. In section IV.D we explain how the selection can be optimized according the relative position of

each robot if they can share their knowledge. But each robot has to be able to select his destination by himself. The priority of each point is evaluated based on its proximity and the complexity of the trajectory to reach it.

When all points of interest have been explored the robot goes back to his starting point. In practice, there was also a timer making the exploration stop, because each run of the challenge took place in a limited time (30mn).

B. Obstacle avoidance

Obstacle avoidance is a well know topic in the robotics community [10]. In the case of NAO, the real problem is to detect the obstacle more than to avoid them, because of the reduced walking speed. The size and the shape of NAO constrain the choice of embedded sensors. NAO can only carry a 5m range Hokuyo laser. Furthermore, the bipedal locomotion of NAO makes NAO much more sensitive to collision than a wheeled robot. A fall leads to a loss of time, sometimes of the localization and, fortunately rarely, to a possible robot break (because the laser sensor is fragile).

All available information is combined to prevent and, if necessary, detect collisions. While the laser is mainly used for localization and mapping (see subsection C), we also use it to detect obstacles. When the robot walks, the laser plane is horizontal at 65cm above the ground, so small objects cannot be detected. That is why the robot performs a vertical scan with a motion of his head whenever it stops, to cover smaller obstacles (Fig. 3).

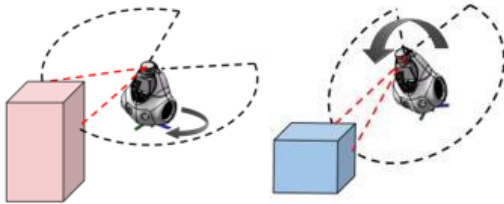


Fig. 3. Horizontal detection (left) and vertical detection (right)

One of the most dangerous obstacles in the CAROTTE challenge is the presence of loose grounds. Some parts of the ground in the arena are made of gravel or loose earth on which it is forbidden to walk. The only way for NAO to detect them is vision. The bottom camera of NAO is used to take pictures of the ground when the robot stops. Using the texture detection of the object recognition feature (see section V), NAO is able to identify a part of the ground as an obstacle to be avoided. Recognized objects from the previously taken pictures are also considered as obstacles. The drawback is that a wrong detection of an object also creates an obstacle: false positive may surround the robot with non-existent obstacles. In practice, this is one of the causes of the failure of the final exploration of the YOJI team.

NAO is equipped as well with 2 ultrasonic sensors with a horizontal field of view of 75° and a small overlap in front of the robot. This very wide angle does not really allow localizing objects around the robot. Thus the ultrasonic data is taken into account only when both of the sensors give the same value smaller than 40cm, which means that an obstacle is exactly in front of the robot. Although this detection is limited, it provides

an alternative to the laser, in particular to detect glass walls, which are invisible to the laser.

Sometimes collisions cannot be avoided: it is then detected with the feet bumpers, arms and inertial unit. By checking the position error of the arms, it is possible to detect whether the arms have touched an obstacle or not. The inertial unit can detect that the robot is no longer strictly vertical. It generally means that the top of the robot has collided with an object invisible for the laser and the ultrasonic sensors, typically the seat of a chair which is in a dead zone for them.

Despite of all these detections, it could happen that the robot falls down anyway. In this case, the robot stands up again and creates an obstacle in the map at his current location. Even if the reason of the fall is not known, the added obstacle will prevent it from coming back to that location.

At last, according to the sensor that detects the obstacle, a life period is given to this obstacle. At the end of this period, the obstacle disappears. This is a way to deal with false detections that can happen with some sensors. The life period is much longer if the obstacle is detected by the laser than if it has been detected by the ultrasound, according to their respective reliability. This period is also a simple way to take into account moving obstacles.

Once obstacles are detected we applied classical methods (A* [11], Configuration Space [12]) to avoid them. Thanks to this set of methods, we improved drastically the robustness of the navigation of NAO along the three years of the challenge.

C. Localization and Mapping

The localization algorithm is based on a particle filter [13]. It looks for the localization of the robot that optimizes the matching between the current laser scan and the current version of the map while the robot is walking. The same particle filter, with a greater number of particles, is used for mapping. When the robot stops, he performs a 360° scan by turning his head. The particle filter finds the best matching between the current version of the map and this global scan. Typically, the laser measurements behind the robot are used to localize accurately the robot in the current map and the measurements in front of the robot are used to create new parts of the map.

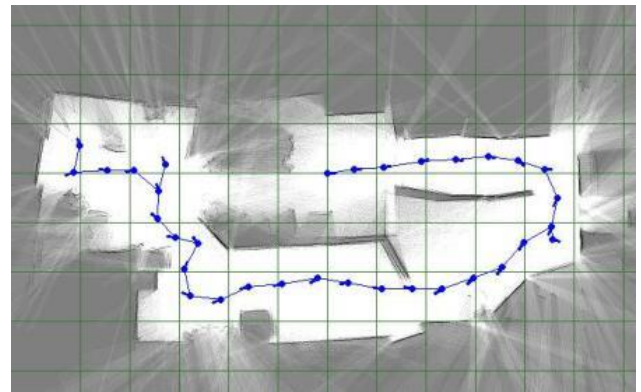


Fig. 4. Raw map created during an exploration

Fig. 4 shows the result of the mapping of one robot in a real environment. The nodes and links of the topological graph are

marked in blue, and placed in the global reconstructed map. White areas correspond to free space, black areas to obstacles and gray areas to unknown environments.

When the robot stops, a topological node is created to store the current horizontal laser scan, the vertical scan, the pictures of the ground and the pictures of the surrounding environment. These pictures are sent to the object recognition server presented in the next section. For the requirements of the CAROTTE challenge, we combine the local metric maps to build a global map of the environment. The robot navigates within this metric map.

The CAROTTE contest rules also require computing the localization of the different rooms. Because the raw map data are noisy, it is necessary to clean the contours of the map. We then build a distance map based on these contours, and erode it. The blobs are then used as seeds for a watershed algorithm (as described for example in [14]). The output gives an estimation of the surface of each room. Fig. 5 presents the contour cleaning step (top) and the obtained seeds for the watershed (bottom).

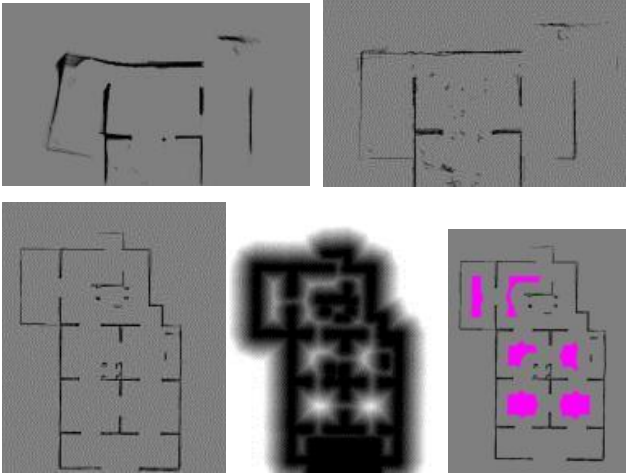


Fig. 5. Adjustment of the contours (top) and detection of the rooms (bottom)

D. Multi-robot exploration

It is interesting to consider the optimization of the exploration by several robots. Each robot has a partial knowledge of the environment and can only explore the interest points he has discovered so far. From a theoretical point of view, it can happen that a robot could reach faster an interest point detected by another robot. Merging the knowledge of the three robots on the remote PC could bring to a global optimization of the environment. One of our competitors at the CAROTTE contest, the team Robots Malins, has made very interesting work about that [15].

We have worked on a supervisor running on the remote PC. Its role would be to send instructions on interest points to be explored by each robot. This appeared to be very complicated because the robots have to share their topological trees, their maps and their localization. Considering the low accuracy available on localization of each robot, it was quite difficult to evaluate the real proximity of a robot from an interest point detected by another robot. Furthermore, the results we got in

simulation, with a better localization, did not really prove that this supervision was very useful on the 2D environments proposed by the CAROTTE contest.

In practice, the heuristics we used for the CAROTTE challenge were simple: for his first choices, the robot on the right of the start line selects the interest points on its right. The one on the left the points on its left, and the robot in the middle will prefer to go straight forward. After two meters, each robot can select his interest points independently from this initial heuristic. The experiments showed that this simple decision process allows spreading the robots in different parts of the environment. Fig. 6 shows the beginning of an exploration with three robots going in three different directions thanks to these simple heuristic.

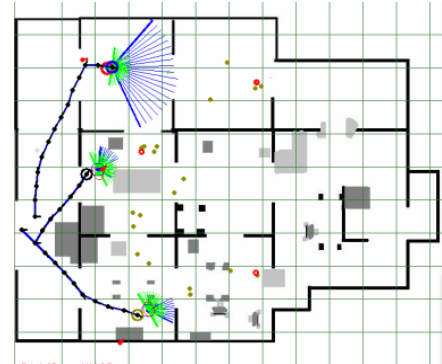


Fig. 6. Paths followed by the 3 robots using the heuristics “to the right”, “to the left” and “to the middle” for exploration (simulation).

The collaboration of the robots was used to merge the map built by each robot off-line, after the exploration phase. In Fig. 7, we present the maps created in the real world by two robots and the way these maps are merged to produce a global map. Merging two maps involves finding a translation and a rotation of the second map with regards to a first one that maximize the number of similar pixels or grid cells. The stochastic search approach of [10] has led to parameter settings (e.g., for parameter clock) which were too difficult and unstable. As opposed to it, the approach of [16], based on the Hough transform, quickly detects horizontal and vertical lines, and therefore their matching between two maps, which is appropriate in structured indoor environments such as the ones of the CAROTTE challenge.

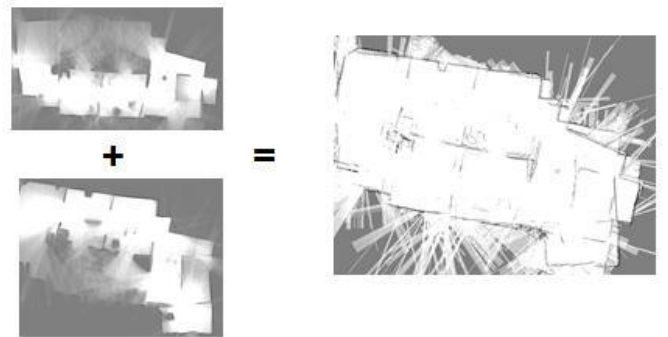


Fig. 7. Merging of the maps created by two robots

V. OBJECT AND GROUND RECOGNITION

The robot is able to recognize both rigid objects within his environment and the nature of the ground around him. Although being based on computer vision mainly, these two tasks rely on different algorithms.

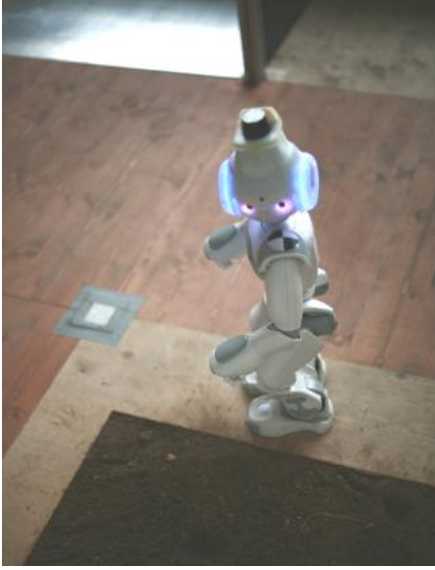


Fig. 8. NAO detects and avoids a dangerous soil

Ground and wall nature recognition is treated as a texture classification problem. Eight types of grounds and nine for walls have to be recognized, but some of them are similar (e.g. metallic grid can be found as a wall or on the ground) and other were considered as too hard to be detected visually as a texture (mirrors, transparent Plexiglas). We finally retain 11 classes and collected 25 to 106 learning images per class, resulting into 559 learning images that were cropped manually in order homogeneously reflect each texture. We then extracted a RGB color histogram (64 bins) and a 512 texture histogram based on local edge pattern [17] from the learning images and used this database to learn a model with a fast shared boosting algorithm [18]. Although designed for larger scale problem, this algorithm has the advantage to be fast to learn thanks to its linear complexity according the number of learning images, the feature size and the number of classes. Its ability to share feature between classes makes him interesting for memory-limited systems. In our case we learn 2000 stumps as weak classifiers to make a robust strong one according to the boosting principle. During the on-line testing phase, each received image is divided according to a 4x4 grid and each one of these regions is tested independently. The classifiers return a confidence score and a threshold was learned for each class (i.e. texture) in order to avoid false positive. Two images are analyzed at each topological node. The upper part of an image taken with a horizontal camera is used to determine the wall texture only. The full image resulting from a camera oriented with a 45° slant is used to identify ground texture only.

Regarding rigid objects, two approaches were used, according to whether the considered object is textured or not (Fig. 9)

For textured objects, we are able to use a classic method that matches keypoints. We first detect points of interest according to the SURF method: it detects interest keypoints with a Hessian matrix-based measure then describes the local area around each point with a 64-dimension vector computed from Haar-wavelets filtering [19]. Then, descriptors extracted from a testing image are matched to those of a learning model using a Flann-based KD tree [20]. Finally, we use RANSAC to determine whether a set of points within the testing image can match with a learned model. For learning, we take several pictures of each object, under all viewpoints of interest. Each image is then compared to the testing image and the final decision is taken from views that match. Each view leads to determine a bounding box thus we simply retain the average of the boxes that do not differ too much (one standard deviation).



Fig. 9. Example of textured object (bottom) and non-textured one (top)

However, such a strategy based on interest keypoints detection would fail with weakly textured object such as balloon of uniform color boxes. In that case, we use a strategy based on efficient subwindow search [21] (ESS) from color descriptors. First, we learn a Support Vector Machine of an object, from color feature. Then, at each pixel of a testing image, we use this model to compute the probability of presence of the object. The ESS algorithm is then based on a branch-and-bound strategy, efficiently implemented with integral images, to rapidly converge toward a bounding box of the object. Since a box is always found, we compare the score to a threshold to possibly reject the detection.

All objects have their own models and are tested separately. Hence, the complexity of the full process may grow linearly with the number of objects (although the process to detect non-textured object is usually lighter than for textured objects). To reduce this global complexity, we sorted the object list to compute only once the features from the testing image. On a core2 duo P8400@2.26GHz, the detection performances are 0.086 second per object and per image, on average, measured with a base of 12 objects (7 textured – 5 non textured) and 118 testing images.

VI. RESULTS AND FUTURE WORK

The presented architecture is designed to deal with the constraints of the platform and the challenge arena, and was successfully used during the challenge. The particle filter mapping combined with the exploration heuristics and the final map merging made it possible to spread the robots and cover a significant part of the arena, while still being able to run directly on the robots. Merging sensors data provided a fairly reliable obstacle avoidance, along with the collision and fall detection. In particular, this solved the problems of chairs and table, which require the whole merging framework to be detected. Several successful explorations were run, with an artificial limit of a maximum 10 meters walked distance. This [video](#) shows an example of successful explorations run on the first year of the challenge. The robot can be seen performing laser scans and avoiding obstacles, along with some example of ground and wall types.

However, some problems occurred in the final run. Some were due to particularly difficult obstacles such as a tricky ground or the robot getting stuck under a bed. Another problem came from the high number of false detections of non-textured objects, which generated virtual obstacles, consequently blocking the robot. However these false detections rather came from a last minute change of the camera, and so a significant difference between the actual images and the training base.

Despite these issues, this challenge was an opportunity to prove that it is possible to explore an unknown environment with a humanoid platform. The issues encountered come mainly from mistakes done in the rush of the challenge, not from the architecture itself. It is also worth noticing that even if the YOJI team was ranked last it competed against wheeled robots, with long ranged lasers, Kinect depth sensors or full computers on board.

These results open interesting perspectives for indoor navigation. In particular, it shows that despite their specific constraints humanoid robots will be able to evolve in domestic environments.

In the future, the robots should not depend on the laser sensor, for price, security and walk stability reasons. This means that, unless a depth sensor can be integrated better, the navigation framework will be mostly based on camera information, and topological data. The work presented in [22] shows a first step towards using the camera only, by computing the current robot orientation from images. Future work includes implementing a vision-based topological framework

ACKNOWLEDGMENT

We thank all the other members of the team: David Coz, Rémy de la Crouée, Damien Henry, Pedro Cardoso, Nicolas Delatre, A special thanks to Laurent Eck who helped us to navigate in the state of the art of localization and navigation.

REFERENCES

- [1] S. Thrun, « Robotic Mapping: A Survey », in *Exploring Artificial Intelligence in the New Millennium*, 2002.
- [2] A. Elfes, « Using occupancy grids for mobile robot perception and navigation », *Computer*, vol. 22, n° 6, p. 46–57, 1989.
- [3] S. Thrun, « A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots », *Int. J. Robot. Res.*, vol. 20, n° 5, p. 335–363, janv. 2001.
- [4] A. Angeli, S. Doncieux, J.-A. Meyer, et D. Filliat, « Visual topological SLAM and global localization », in *IEEE International Conference on Robotics and Automation*, 2009. *ICRA '09*, 2009, p. 4300–4305.
- [5] K. Konolige, E. Marder-Eppstein, et B. Marthi, « Navigation in hybrid metric-topological maps », in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, p. 3041–3047.
- [6] S. Thrun, « A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots », *Int. J. Robot. Res.*, vol. 20, n° 5, p. 335–363, janv. 2001.
- [7] A. Hornung, K. M. Wurm, et M. Bennewitz, « Humanoid robot localization in complex indoor environments », in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, p. 1690–1695.
- [8] D. Maier, A. Hornung, et M. Bennewitz, « Real-time navigation in 3d environments based on depth camera data », in *Proc. of the IEEE/RSJ Int. Conf. on Humanoid Robots (Humanoids)*, 2012.
- [9] C.-H. Chang, S.-C. Wang, et C.-C. Wang, « Vision-based cooperative simultaneous localization and tracking », in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, p. 5191–5197.
- [10] O. Khatib, « Real-time obstacle avoidance for manipulators and mobile robots », in *1985 IEEE International Conference on Robotics and Automation. Proceedings*, 1985, vol. 2, p. 500–505.
- [11] S. J. Russell et P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.
- [12] T. Lozano-Perez, « Spatial Planning: A Configuration Space Approach », *IEEE Trans. Comput.*, vol. C-32, n° 2, p. 108–120, 1983.
- [13] I. M. Rekleitis, « A particle filter tutorial for mobile robot localization ».
- [14] L. Vincent et P. Soille, « Watersheds in digital spaces: an efficient algorithm based on immersion simulations », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, n° 6, p. 583–598, 1991.
- [15] L. Matignon, L. Jeanpierre, et A.-I. Mouaddib, « Distributed value functions for multi-robot exploration », in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, p. 1544–1550.
- [16] S. Carpin, « Fast and accurate map merging for multi-robot systems », *Auton. Robots*, vol. 25, n° 3, p. 305–316, oct. 2008.
- [17] Y.-C. Cheng et S.-Y. Chen, « Image classification using color, texture and regions », *Image Vis. Comput.*, vol. 21, n° 9, p. 759–776, sept. 2003.
- [18] H. Le Borgne et N. Honnorat, « Fast shared boosting: Application to large-scale visual concept detection », in *2010 International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2010, p. 1–6.
- [19] H. Bay, T. Tuytelaars, et L. Van Gool, « Surf: Speeded up robust features », *Comput. Vision–ECCV 2006*, p. 404–417, 2006.
- [20] M. Muja et D. G. Lowe, « Fast approximate nearest neighbors with automatic algorithm configuration », in *In VISAPP International Conference on Computer Vision Theory and Applications*, 2009, p. 331–340.
- [21] C. H. Lampert, M. B. Blaschko, et T. Hofmann, « Beyond sliding windows: Object localization by efficient subwindow search », in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. *CVPR 2008*, 2008, p. 1–8.
- [22] E. Wirbel, B. Steux, S. Bonnabel, et A. de La Fortelle, « Humanoid robot navigation: From a visual SLAM to a visual compass », in *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2013, p. 678–683.