A new tool to initialize global localization for a mobile robot

Olivier Aycard, Christophe Brouard

A new tool to initialize global localization for a mobile robot

Olivier Aycard, Christophe Brouard
University of Grenoble (UGA) - Grenoble Informatics Laboratory (LIG), France
{olivier.aycard, christophe.brouard}@univ-grenoble-alpes.fr

***Abstract*—Localization is the ability for a mobile robot to know its position at all times. When the initial position is unknown, the localization process has to manage several possible positions that could correspond to the real one. The main drawback of this technique is the cost of computational complexity that could be high. In this paper, we present a new way to determine the set of initial possible positions that is fast (less than 3s) and enables to start the localization process with a small number of possible positions. The consequence is that our localization process determines the real position in a fast and robust way. Experimental results show the benefits of the method.**

***Index Terms*—mobile robots, localization, perception, pattern recognition, information retrieval**

I. Introduction

For the various applications considered by a mobile robot, the capacity for the robot to locate itself, that is to say to know at all times its position is essential. For instance, when a mobile robot moves in its environment to perform a task (ie, to reach a goal), it has to know that it is getting closer to its goal and moreover, it should be able to know that it has reached its final position.

Initial research on localization considered that the initial position of the mobile robot in its environment was known. In this case, the localization process has to track this position while the robot is moving and perceiving its environment. This approach is known as local localization or position tracking. The most common choice to represent the position of the mobile robot is an unimodal gaussian [11] that gives fast and robust implementations for localization of a mobile robot in its environment [5].

But in some cases, the initial position is unknown. Then, the localization process has to track several positions while the robot is moving in its environment. This approach is known as global localization or multi hypothesis tracking[1]. In this case, unimodal gaussian is not appropriate and [9] use a multi gaussian hypothesis representation at the cost of increased computational complexity. The most general representation to solve global localization is to approximate the probability distribution of the possible positions by a set of particles, where each particle has two components: (i) a position and (ii) a probability/score of this position [3]. [15] have obtained very good results for global localization of a mobile robot in indoor environment using particles. Moreover, [7] has shown that global localization is a generalization of local localization starting with a huge set of particles, uniformly distributed over the space of possible positions, at the beginning of the global localization and reducing the size of this set as the robot gains more and more confidence in its real position. For instance, [7] started with a set of 40 000 particles to decrease this set to less than 100 particles when the position has been found. The main drawback of using a set of particles to approximate the probabilty distribution over the possible positions is that the cost of computational complexity is increased especially at the beginning of the process where we have to deal with a huge set of particles.

To deal with this problem of global localization, a simple approach is to compare what the robot has perceived with what it should perceive in each position in its environment, and select the positions with the greatest number of similarities as we did in [16]. But for global localization, this approach is not real time either and could take several minutes because we have several million positions to explore in a map. So our idea is to design a method integrating some concepts of information retrieval [13] to get the positions with the highest similarity with the current observation in a few seconds.

In the next section, we present the mobile robot used in this work. In section III, we formalize the localization process and detail its implementation. Section IV is dedicated to our implementation. We show some experimental results in section V. We conclude and give some perspectives in section VI.

II. Experimental platform

Our mobile robot named robair is a differential drive mobile robot (see figure 1). This is a home made robot designed and built in our fablab. It is made of flower pots: a large pot for the bottom and a smaller one for the top. Its head is equipped with LED strips. Its base is circular with 2 drive wheels and 2 idler wheels for stability (see figure 2). To control it, we can send translation and rotation speed to its base. Each wheel is equipped with encoders and we have designed an odometer to estimate its motion. It is equipped with a laser scanner to perceive its environment at a frequency of 25 Hz. The range of the laser is of 5.5 meters and its field of view is of 240 degrees with an angular resolution of 1/3 degree (see figure 3). So a scan of the laser is composed of 724 values corresponding to the 724 hits of the laser.

*This work has been partially supported by MIAI @ Grenoble Alpes, (ANR-19-P3IA-0003)

[1]A very complete review of localization techniques could be found in [12].

Fig. 1. Robair. The red circle at the bottom shows the position of the laser scanner.

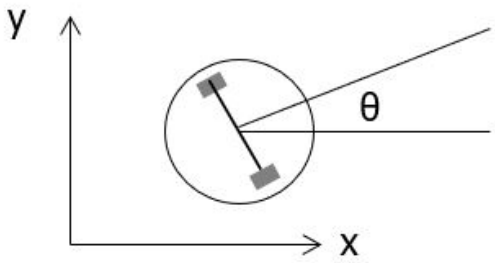

Fig. 2. Circular base of robair showing the 2 driven wheels and its position (x, y, θ) in its environment.

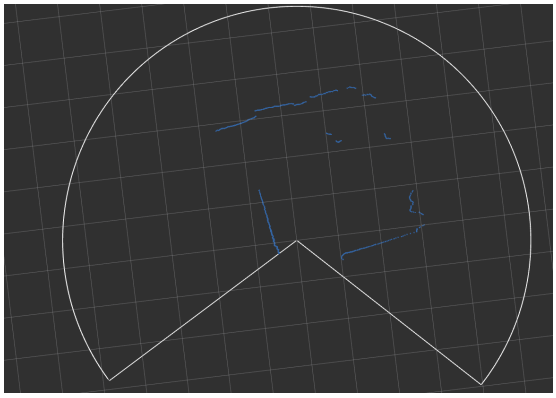

Fig. 3. Range, field of view of the laser in white color and an acquisition in blue color. The background is composed of a grid where each square has a size of 1m x 1m.

Its position is defined by its pose (x, y) in a plane and its orientation θ (see figure 2).

III. Localization process

In this section, we detail the different aspects of the localization process: notations, formalization and resolution of the problem and initialization of localization.

A. Notations

The localization problem can be treated as a process of taking inputs from measurements from motion sensors such as odometry or inertial measurement which is denoted by U and from sensor measurements including measurements from perception sensors such as laser scanners or cameras which is denoted by Z. It includes a static map M as well. The process output is the estimated position of the robot X.

For positions which tend to change over time, we use specific variables to indicate values of each state at a certain time. For instance, x_t indicates the position of the mobile robot at time t. This allows to define the trajectory of the robot over time:

$$X = x_{0:t} = \{x_0, x_1, ..., x_t\} \tag{1}$$

As the robot moves, its state x_t evolves, the motion sensors allow to measure the control u_t of its displacement and the perception sensors allow to collect measurements of the environment z_t. In addition, we define the following set to refer to data leading up to time t:

$$U = u_{1:t} = \{u_1, u_2, ..., u_t\} \tag{2}$$

$$Z = z_{0:t} = \{z_0, z_1, ..., z_t\} \tag{3}$$

B. Formalization and resolution of the localization problem

In the probabilistic form, the localization problem involves estimating the probability distribution $P(X|Z, U, M)$. Instead of determining the whole trajectory of the robot X, in practice we are usually only interested in determining the current position of the robot x_t: $P(x_t|z_{0:t}, u_{1:t}, M)$.

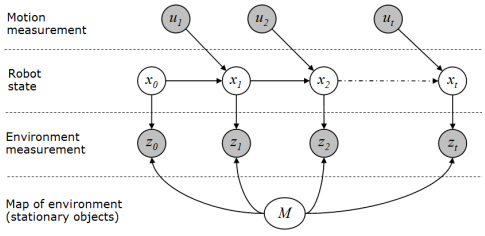

Fig. 4. Graphical model representation of a simple bayesian filtering. Gray circles denotes observed states, clear circles denotes hidden states to be estimated.

This probability distribution describes the posterior density of the robot position at time t given the measurements up to time t, the control inputs up to time t and the map. This posterior density is also called Belief: $Bel(x_t)$. In general, since data arrives over time, a recursive solution is desirable. The

graphical model in Fig. 4 explains the dependency structure of variables in the localization problem.

Bayesian filtering (sometimes known as bayesian sequential estimation) [4] is a widely accepted probabilistic framework to solve the problem of estimating dynamic states of a system evolving in time given sequential observations or measurements about that system. The idea behind the bayesian filtering is that it allows to use past and present measurements in sequence to enhance the estimation of the actual system state. Starting with an estimate for the distribution $P(x_{t-1}|z_{0:t-1}, u_{1:t-1}, M)$ at time $(t-1)$, the joint posterior, following a control u_t and measurement z_t, is estimated in 2 steps using a bayes filtering process:

$$\underbrace{P(x_t|z_{0:t}, u_{1:t}, M)}_{\text{posterior at } t} \propto \underbrace{P(z_t|x_t)}_{\text{update}} \underbrace{\int_{x_{t-1}} P(x_t|x_{t-1}, u_t) \underbrace{P(x_{t-1}|z_{0:t-1}, u_{1:t-1}, M)}_{\text{posterior at } t-1}}_{\text{prediction}} \tag{4}$$

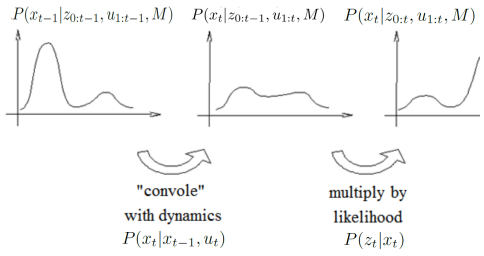

Fig. 5. Sequential bayesian filtering in 2 phases: prediction with motion measurement u_t + update with observation measurement z_t

This algorithm can be interpreted as transformations over distributions of probability. Using the state transition function $P(x_t|x_{t-1}, u_t)$ and the previously estimated probability $P(x_{t-1}|z_{0:t-1}, u_{1:t-1}, M)$, we obtain a distribution $P(x_t|z_{0:t-1}, u_{1:t}, M)$ which is commonly called the prediction step. Then introducing the new measure we update the distribution with the likelihood $P(z_t|x_t)$ to obtain the desired result $P(x_t|z_{0:t}, u_{1:t}, M)$. The process is illustrated in Fig. 5. It is clear that in order to compute the sequential bayesian filter, definitions of the state transition function (or dynamic model) $P(x_t|x_{t-1}, u_t)$ and the likelihood function (or sensor/measurement model) $P(z_t|x_t)$ are required.

C. Initialization of localization process: local localization versus global localization

As mentioned in section I, when the initial position x_0 is known, the belief $Bel(x_0)$ is then usually initialized by a narrow Gaussian distribution centered around the initial position. When it is unknown, $Bel(x_0)$ is initialized by a uniform distribution over the space of all legal positions in the map. In this particular case, it makes sense to perform an initial observation z_0, before starting the localization process, to have a better distribution than a uniform distribution:

$$Bel(x_0) = P(x_0|z_0) \propto P(z_0|x_0) \tag{5}$$

It is obvious that determining a model for $P(z_0|x_0)$ that is able to find the most probable positions is of key importance for global localization. In next section, we detail how we determine this model and what the benefits are for global localization in terms of speed and robustness.

IV. Implementation of global localization process

In this section, we detail our implementation. First of all, we explain how we built a map of the environment and how we pre-processed this map. In section IV-B, we describe the initialization of global localization. The dynamic and observation models are presented in section IV-C. Finally, we detail the implementation of global localization with particles.

A. Building of map and preprocessing

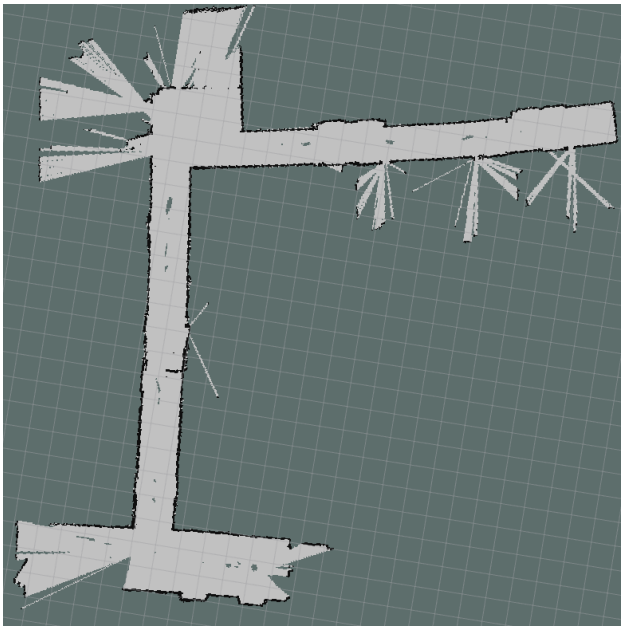

Fig. 6. Occupancy grid map of a part of our lab. This map has a size of about 25m x 20m.

1) Representation of the environment: The most common representation for a map are occupancy grids or grid based representations, that were first introduced by [6]. In this representation, the environment is subdivided into a regular array or a grid of rectangular cells. The resolution of the environment representation directly depends on the size of the cells. In addition to this discretization of space, a probabilistic measure of occupancy is estimated for each cell of the grid which indicates whether that cell is occupied by an obstacle or not. For instance, in figure 6, cells with a low probability of occupancy are white. The cells having a high probability of occupancy are black. In light grey, we have the cells where

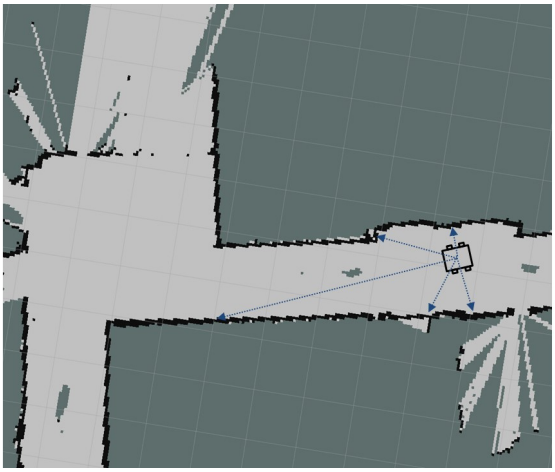

Fig. 7. 5 ray castings of the laser

the laser hasn't taken any measurements, ie. cells for which we have no information about their occupancy. The main advantage of this representation is that this is a very low level representation and as close as possible to the perception of the laser, including the noise and drawbacks of this kind of sensor.

We built an occupancy grid of the LIG. In this grid, the discretization is of 5 centimers $\times$ 5 centimeters $\times$ 5 degres. This discretization is in fact the discretization used to perform global localization. This offline process has been done using the gmapping module of ROS [1] and took about 1 hour. This map will be used to perform global localization.

2) Preprocessing of the map: As mentioned in section I, our idea is based on the comparison of what the robot has perceived with what it should perceive in each position in its environment. Then, with the grid map of the environment, we need to compute for each position, what the robot should perceive, ie. the value of the 724 hits of the laser scanner. To perform this task, for each position and for each hit, we use a simple ray casting technique [2] to compute the intersection between a theoretical infinite hit in a given direction and the first occupied cell in the map in this direction (see figure 7). This pre-processing is done offline and took about 1 hour on the grid map of our lab. The output of this process is a list of about 3.5 million positions and for each position, a vector of 724 hits corresponding to what the laser should perceive in this particular position.

B. Initialization of global localization

In this section, we propose an algorithm, integrating some concepts from information retrieval [13], making it possible to obtain in a few seconds a list of the positions yielding the best match between what the laser should perceive, and what it is currently perceiving. This algorithm proceeds in 2 steps: the first step is to find the poses (x, y) that best correspond to the patterns extracted from the list of positions in the map and the second step is to specify the orientation of the robot. Before detailing these 2 steps, we explain how we extract some patterns from the observation of the laser and how we represent the list of 3.5 million positions as a two level index.

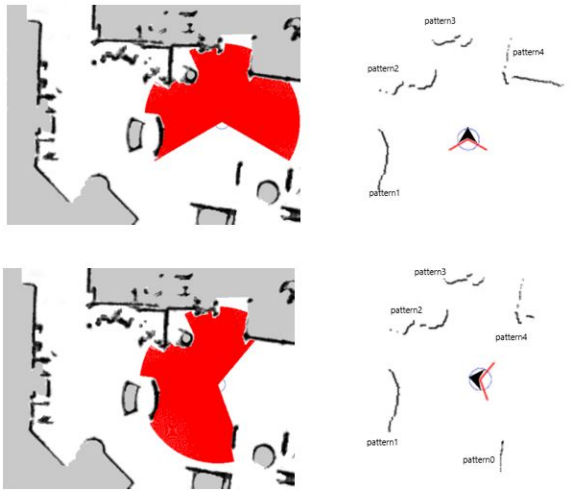

Fig. 8. Extracted patterns: the red circle shows the field of view of the laser scanner. As we can see, the same patterns are globally extracted for both orientations. In general, this fact allows us to predict x,y from patterns.

1) Extraction of patterns from a laser scan: In order to abstract from the orientation, we will consider the sequences delimited by the hits corresponding to an empty area (see figure 8). As the field of view of the laser is limited to 240 degres, we consider two different positions for each pose (x, y) one with $\theta = 0$ and one with $\theta = 180$ so as to cover all the visible patterns in a pose (x, y).

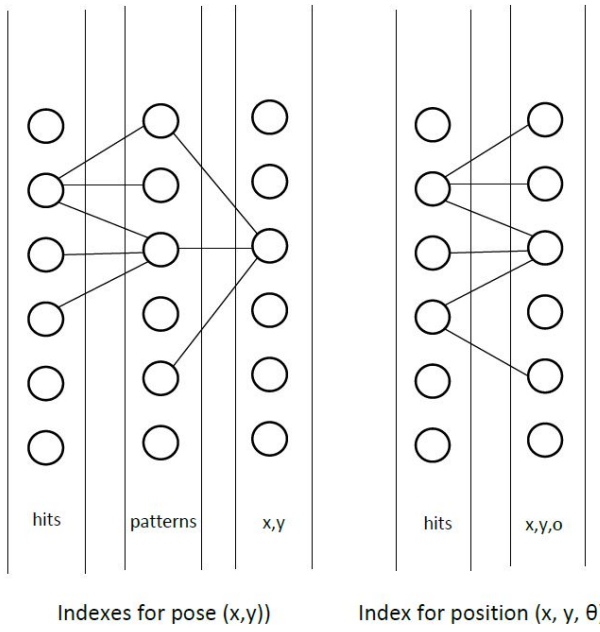

Fig. 9. creation of indexes: on the left, the two level indexes to find the pose(x, y) and on the right, the index to find the position(x, y, θ)

2) Creation of a two level index: To create a two level index, for each possible position of the robot:

1) we extract all the patterns;
2) to create the index between hits and patterns, for each pattern:
   - if it has not been seen previously, we create some links between each hit of the pattern and a new node at the pattern level;
   - else we increase the frequency of the corresponding node at the pattern level;
3) when the first level of the index is completed, we proceed in the same way to create the index between the pattern level and the pose level.

This process is done offline and takes between 2 and 3 hours to be performed. This process is illustrated on the left part of figure 9.

3) Find the best poses (x, y): Using the two level index, we find the best poses (x, y) in several steps:

1) We extract the different patterns from the scan of the laser;
2) Using the hit-patterns index, a search for the N patterns closest to each extracted pattern (simple counting of common hits) is performed. To perform this search, for each hit, we increase the frequency of each connected pattern. This count enables us to quickly find the patterns close to the 724 hits of the laser. The score of each close pattern found is normalized by dividing by the maximum score obtained;
3) Using the patterns-positions index, the search for the N positions most associated with the patterns found by the previous operation (simple summation of normalized scores) is performed.

This process is very fast and takes less than 1 second.

4) Find the best position (x, y, θ): When we have the list of the best poses (x, y), we proceed in several steps to determine the most probable positions (x, y, θ):

1) Using the N best poses (x, y) found in the previous phase, we build the list of all the corresponding positions (x, y, θ);
2) We build "on the fly" a new index linking hits and positions (x, y, theta) and compute the idf weighting[2] of each hit;
3) We search for the best positions (x, y, θ) by summing the idf weightings of each hit. The idea is to give more importance to the hits which are specific to a position. The weight of a hit therefore corresponds to the number of indexed positions divided by the number of occurrences of this hit in all indexed positions. The more frequent the hit, the lower its weight.

This process is very fast as well and takes between 1 and 2 seconds.

C. Dynamic model and observation model

When the first step of the global localization is performed as described in the previous section, we have a set of possible initial positions, robair will start to move in its environment and each time it has traveled a predefined distance, we will predict its position using a dynamic model. In a second step, we will update its position using a sensor model. In this section, we detail these 2 models that we have adapted from our previous work [16].

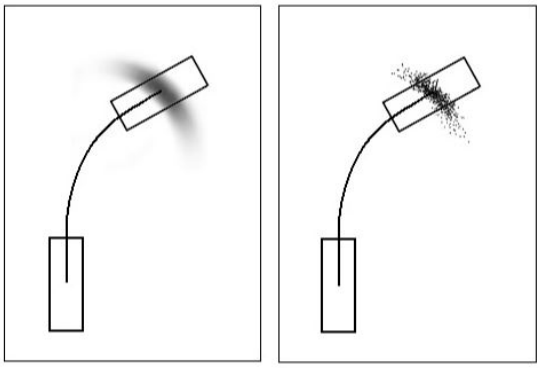

Fig. 10. The probabilistic motion model $P(x_t \mid x_{t-1}, u_t)$ of the robot (left) and its sampling version (right).

1) motion model $P(x_t \mid x_{t-1}, u_t)$: For the motion model, we adopt the probabilistic velocity motion model similar to some of our previous work [16]. The robot motion u_t is comprised of two components, the translational velocity v_t and the yaw rate ω_t. Fig. 10 depicts the probability of being at position x_t given previous position x_{t-1} and control u_t. This distribution is obtained from the kinematic equations, assuming that robot motion is noisy along its rotational and translational components. A sampling version of the motion model (Fig. 10 right) is used to generate all possible poses x_t given the previous pose x_{t-1} and the control u_t.

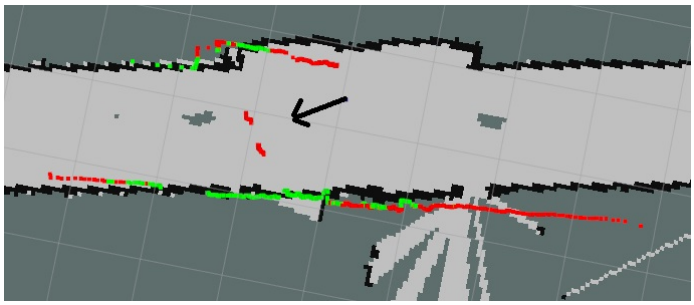

Fig. 11. Illustration of the sensor model $P(z_t \mid x_t, M)$ for a given position of the robot. The robot is located at the origin of the arrow and the arrow shows its orientation. Green points are hits of the laser that match to an occupied cell and red one are points that match to an empty cell.

2) sensor model $P(z_t \mid x_t, M)$: For the sensor model $P(z_t \mid x_t, M)$, the mixture beam-based model is widely used in the literature [8], [10]. However, the model comes at the expense of high computation since it requires a ray casting operation for each beam. This can be a limitation for real time application if we want to estimate a large amount of measurements at the same time. To avoid ray casting, we propose an alternative model that only considers end-points

[2]inverse document frequency [13]

of the beams. Because it is likely that a beam hits an obstacle at its end-point, we focus only on occupied cells in the grid map.

First, from the robot position x_t, each individual measurement z_t^k is projected into the coordinate space of the map. If the grid cell that the projected end-point falls into is occupied, we increase the score. The final voted score represents the likelihood of the measurement. Figure 11 shows an example of the score obtained for a given position in the environment.

The proposed method is just an approximation to the measurement model because it does not take into account visibility constraints, but experimental evidence shows that it works well in practice. Furthermore, with a complexity of $O(K)$, the computation can be done rapidly.

D. Particle filter and resampling

According to our initialization phase, we decide to manage only 100 particles. After each prediction and update of our estimation, we proceed to a resampling of the particles as in [3]. Our goal is to progressively shift from a global localization to a local localization. So, performing resampling at each step will focus the localization process only on particles that have a high probability to correspond to the real position of the mobile robot. Moreover, with a very small number of particles, the localization process is able to work in less than 0.5s.

V. Experimental results

In this section, we firstly present some quantitative results about the initialization phase and secondly we describe the complete localization process on some examples.

A. Initial localization

ranking	1	5	10	30	50	100
pose (x, y)	9.35	33.09	40.29	58.27	68.35	79.86
position (x, y, o)	37.41	54.68	58.27	66.91	69.78	72.66

TABLE I

Percentage of cases in which the real pose or position is in the first position, in the 5 first positions, etc. We observe that the second step integrating the orientation improves the localization.

ranking	1	5	10	30	50	100
without idf	25 .90	41.73	46.76	53.96	57.55	63.31
with idf	37.41	54.68	58.27	66.91	69.78	72.66

TABLE II

Percentage of cases in which the real pose or position is in the first position, in the 5 first positions, etc. We observe that the idf weighting improves the results.

A test set containing 139 positions was built by moving the robot in the previously mapped space and recording the sensor data at each position. This test set includes various difficulties linked to the difference between the mapped space and the real space (open/closed doors, presence of mobile obstacles, etc.) and also the relative resemblance of many positions due to the fact that the mapped space mostly corresponds to corridors. Despite these difficulties, the algorithm, with a number of 100 particles, generally succeeds well in the task since in almost 70% of cases the correct answer (within a distance of 50 centimeters and 25 degrees) is in the first 30 answers proposed in the initialization process (see table I). We also note the very positive effect of taking into account the idf weighting (see table II).

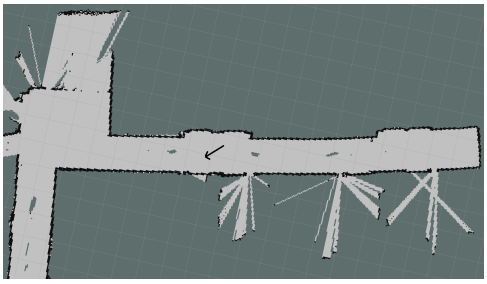

Fig. 12. real position of the robot at the beginning of the localization process: the pose is at the origin of the arrow and the arrow represents the orientation

B. Global localization

In this subsection, we detail an example of the complete localization process. In this example, the robot is localized in front of an office at the beginning of the process (see figure 12). It has to follow a moving person in the corridor of our lab using our follow-me behavior [14]. Before starting to follow a moving person, it initializes its localization which takes around 2-3 seconds. While the robot is following a moving person, we perfom a relocalization each time the robot has traveled about 1 meter or 20 degrees. The process of relocalization takes about 0.5s while the robot continues to move. In the next paragraphs, we explain the initialization phase and after we describe the relocalization process and illustrate the complete localization process with some figures[3].

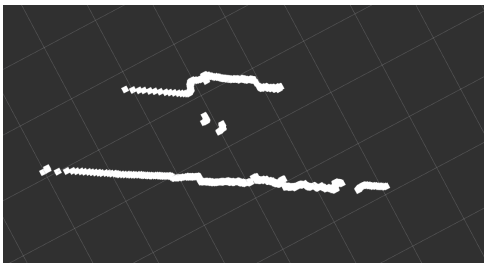

Fig. 13. Initial observation of the robot used to initialize the localization process: we see the two walls of the corridor and the two legs of the followed person

1) Initialization of localization: The initial observation of the robot is shown in figure 13. According to this observation, in figure 14, we illustrate the 3 first positions found by the initialization. We see that the first position found by the initialization process is close to the real position. The second

[3]A video of the complete process is available at lig-membres.imag.fr/aycard/html/Projects/Localization/localization1.mp4.

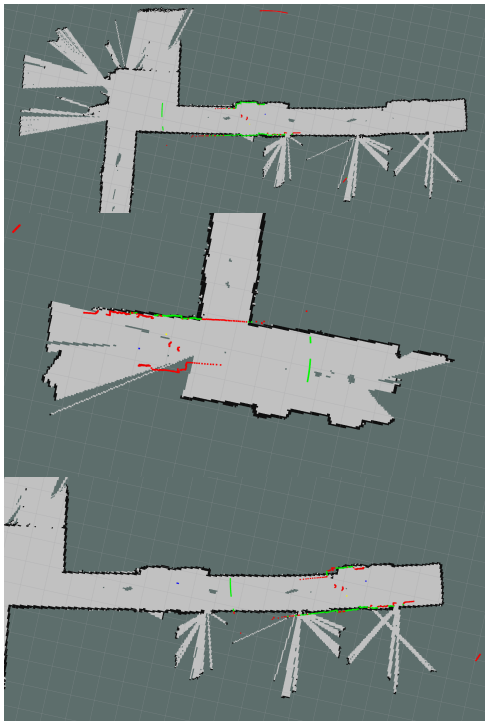

Fig. 14. 3 first positions found by the initialization process: for each position, we show in green and red the matching between the hits of the laser and the grid map using our sensor model

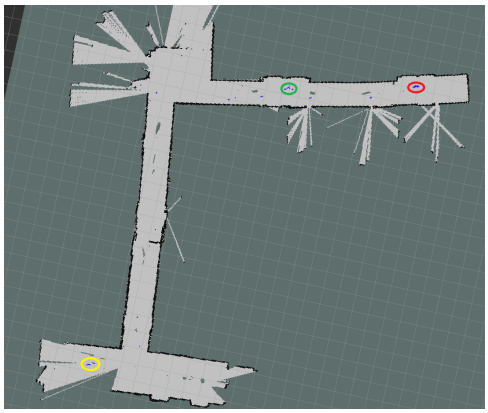

Fig. 15. The 100 positions found by the initialization process: each position is represented by a blue point

position is wrong and with a small similarity from a global point of view and the third one is logical because this position is similar to the real one in terms of observation. At the end of the initialization process, we see in figure 15 the 100 positions found. We distinguish 3 clusters (ie, a set of particles close to one-another) corresponding to the 3 first positions found: the green cluster corresponds to some positions close to the first position found ie, the real one, the yellow cluster corresponds to some positions close to the second position found and the red one corresponds to some positions close to the third position. It means that these 3 positions have been considered as important by the initialization process. This is exactly the goal of our approach because the initialization process has concentrated the particles on positions that could correspond to the real one. Moreover, we see other particles corresponding to some other positions found by the initialization process.

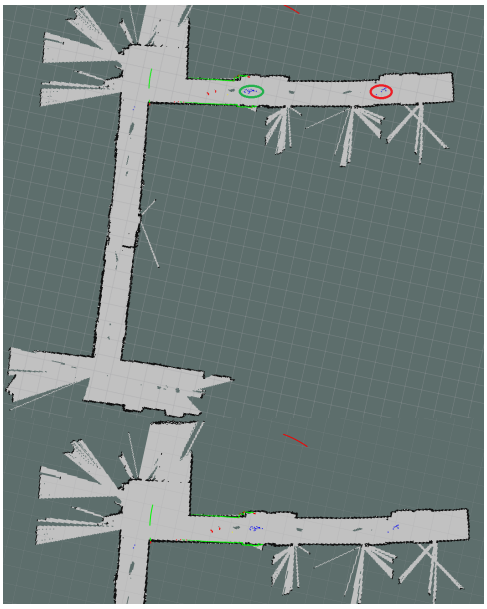

Fig. 16. The 100 positions after a first relocalization and resampling. In the bottom part, we see the best position and its matching with the grid map

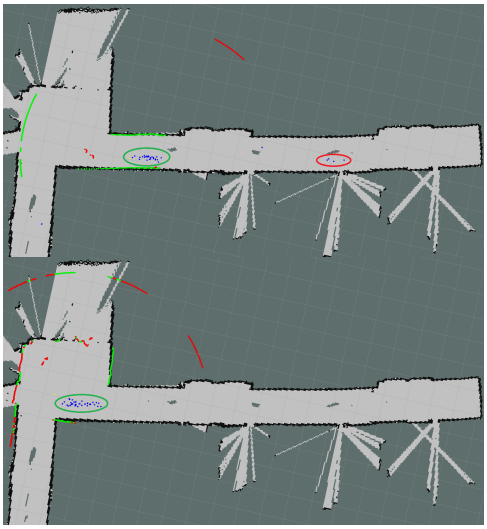

Fig. 17. The 100 positions after a fourth and sixth relocalization and resampling

2) The localization process: When the initialization is done, the robot starts to follow a moving person. After about one meter, it performs relocalization and resampling of the particles. The result is presented in figure 16. The yellow cluster has disappeared because the score of the corresponding particles were low (ie, low probability that given the second observation, the robot is localized at one of the corresponding positions). Some other positions have disappeared as well and

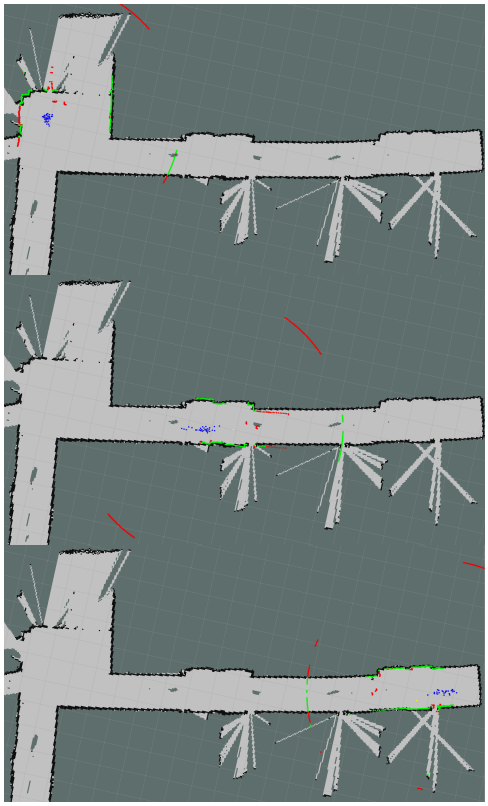

Fig. 18. The 100 positions after different relocalization and resampling

now the particles are focused on the red and green clusters that have a high probability to correspond to the real position.

After a fourth relocalization (figure 17), the two clusters are still present but most of the particles are focused on the green cluster. The red cluster now has very few particles and moreover the corresponding particles are sparse. After a sixth relocalization, there is only the red cluster that is present and the particles are now focused on the real position meaning that the robot is localized.

When the robot is localized (figure 18), it is able to track its position while following a moving person. All the particles are focused in an area which is always close to the real position.

VI. Conclusion and perspectives

In this paper, we presented a method to perform global localization for a mobile robot. The main advantage of the method is that it initializes global localization with interesting positions, reducing the cost of managing multiple hypotheses. Moreover, with this small number of hypotheses, our approach runs in real time. The approach is based on information retrieval and indexes for initialization, and particles and sampling for progressively finding the real position of the robot.

Our approach still has two main drawbacks:

1) the real position belongs to the 100 first positions in only 75% of the cases. Moreover, if there are no positions found that are close to the real one, it's difficult for our approach to find the real position of the robot. But this is a drawback of almost all known approaches to perform global localization;
2) we use two different processes to initialize global localization and to perform global localization. Moreover, these 2 processes use different strategies to solve 2 problems that are very close.

Taking into account the previous drawbacks, our future work is to perform the complete global localization process using an approach only based on information retrieval.

References

[1] gmapping - ros wiki. http://wiki.ros.org/gmapping.

[2] Arthur Appel. Some techniques for shading machine renderings of solids. In *AFIPS*, 1968.

[3] S. Arulampalam, S Maskell, N Gordon, and T Clapp. A tutorial on particle filter for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 2002.

[4] Anderson Brian D. and Moore John B. *Optimal filtering*. Prentic-Hall, Englewood, 1979.

[5] E.D. Dickmanns and V. Graefe. Application of monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[6] A. Elfes. Occupancy grids: A probabilistic framework for robot perception and navigation. *Journal of Robotics and Automation,*, 3:249–265, 1987.

[7] D. Fox. Kld-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, 2002.

[8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[9] X. Zhang H. Gao and Y. Jing. A novel global localization approach based on structural unit encoding and multiple hypothesis tracking. *IEEE Transactions on Instrumentation and Measurement*, 99:1–16, 2019.

[10] D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7):579–598, 2003.

[11] R-E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME—Journal of Basic Engineering*, pages 35–45, 1960.

[12] W. Burgard S. Thrun and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[13] Gerard Salton and Michael McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

[14] Philip Scales, Olivier Aycard, and Véronique Auberge. Studying navigation as a form of interaction: a design approach for social robot navigation methods. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. To appear.

[15] Wolfram Burgard Sebastian Thrun, Dieter Fox and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(99-141), 2001.

[16] TD. Vu, O. Aycard, and N. Appenrodt. Online localization and mapping with moving objects tracking in dynamic outdoor environments. In *IEEE International Conference on Intelligent Vehicles (IV)*, 2007.