

Applied Mathematical Sciences, Vol. 5, 2011, no. 42, 2137 - 2157

Metrical Mapping and Self Localization with RFID Technology and Visual Landmarks

Younes RAOUI

Laboratory of Computer Sciences, Artificial Intelligence and Form Recognition (LIMIARF)
Faculty of Sciences, Mohamed V University, 4 Street Ibn Battouta B.P. 1014 RP, Rabat, Morocco

El Houssine BOUYAKHF

Faculty of Sciences, Mohamed V University, 4 Street Ibn Battouta B.P. 1014 RP, Rabat, Morocco

Michel DEVY

Laboratory of Analysis and Architecture of Systems(LAAS), 7 Street Colonel Roche, Toulouse, France

Fakhita REGRAGUI

Faculty of Sciences, Mohamed V University, 4 Street Ibn Battouta B.P. 1014 RP, Rabat, Morocco

Abstract

This article deals with robot self localization and mapping. The robot equipped with camera and RFID antennas navigates in a commercial center with motion and interaction with a user. It must know its spatial position and its orientation as it should guide correctly the user. We apply for that RFID and vision based self localization. We deploy RFID tags and visual landmarks in our environment. When the robot navigates, the predicted positions are affected by the noise. To correct them, we apply first a sequential Monte-Carlo localization based on the particle filter. We use an active localization method built on the theoretical basis of information entropy to improve the positioning accuracy. Second, we use visual landmarks and a Pinhole camera model to have the coordinates of the feature points in the camera. Then we describe two methods of robot mapping which use the positions pre-computed based on a probabilistic measurement model of RFID readers which allows us to accurately localize RFID tags in our environment.

Keywords: Robots, Localization, Mapping, Extended Kalman Filter, Particle Filter, Monte-carlo localization, RFID, Visual landmarks, Entropy, Pinhole Camera Model

1 Introduction

The roboticists work for making the navigation in cluttered areas more efficient and safe. The robot has to execute tasks while communicating with people that are in its environment. Such environment may be complex and dynamic like shopping centers or airports. Also, the robot has to avoid dynamic obstacle and perform efficient self localization. This work aims to learn to a trolley advanced behaviors so as to assist a user when doing shopping in a commercial center. To interact with the environment, the trolley is equipped with many sensors for detecting, tracking and identifying its user either through vision, Radio Frequency Identification, audio, as well as with haptic, stereo. To navigate safely in the store and to locate itself, it has to use RFID and vision. Commonly, we consider 4 scenarios for interaction between the trolley and the user ([1]) :

- the trolley passive mode, using a haptic interface, for a person who may need a walking aid;
- the trolley active mode, using a haptic interface, for a person who wants more services or who may need a walking aid;
- the trolley passive mode, using a multimodal interface, for an "agile" person used to the shopping center.
- the trolley active mode, using a multimodal interface, for an "agile" person who requires more services.

This work focuses on developing functions for self localization and mapping. Missions are defined in terms of localization and spatial consistency and robust control are ensured by it. In fact, the robot is instantly localized with its position x , y and its orientation θ . At each time instant, it requires to know these coordinates so that it can reach its objective with a high accuracy. Two applications are sought. In one application, we use first RFIDs as landmarks. So we equip our robot with RFID antennas and at the same time we place RFID tags in the extremities of corridors. In order to do self localization of the robot, we apply two approaches, one is deterministic based on the Kalman filtering and the other is stochastic based on the particle filter. This latter is more realistic because it takes into account the uncertainties. Then, we propose a technique to enhance the location information with data of antennas which don't have observations and apply an active method to determine the robot's actions by planning based on localization uncertainties. Uncertainties, which can be quantified as information entropy depends on the route that the robot used to explore the environment ([2]). Secondly, we use visual landmarks for localization. To perceive these landmarks, we use the simplest model which is the Pinhole camera model. Besides, in each part, we show the results of robot navigation on a predefined map. The other application is concerned with the

mapping with RFID tags. By applying the model perception of the RFID antennas, we estimate the position of tags. This is done with a known trajectory of navigation. We propose two algorithms, deterministic and probabilistic, which construct a map with RFID tags. This paper is organized as follows. After discussing related works, we will present the metrical method for self localization in section 3. Then we describe stochastic localization with particle filter. In section 5, we describe how we can localize our robot with visual landmarks using the Pinhol camera model. Finally, we present a deterministic and bayesian methods for mapping with RFID tags.

2 Related works

Trajectory can be estimated by using low cost passive RFID tags and odometry in unknown environment. This is a prerequisite for mapping RFID with particle filters approach without a reference positioning systems. This method avoids the noisy nature of RFID measurements and the absence of distance and bearing information as it is based on a non-parametric model for spatial relationships between RFID measurements. One of the first surveys of localizing a mobile robot via RFID was developed by Hanel et al. ([3]) They used a probabilistic sensor model for their RFID reader. They associate the probability of tag detection with the relative position of their tag with respect to the antennas. This model was used to map the positions of passive RFID tags shown on a previously computed maps learned through laser based SLAM algorithm. The positions of the tags are represented by a number of particles, and weights are updated at each detection of the tag. Another set of particles is used to represent the robot pose according to the MonteCarlo localization. Montemerlo et al ([4]) work concerned with FastSLAM to generate consistent maps computationally with laser range finders. Their transponders positions are not simultaneously mapped, and the reconstruction of the robot trajectory depends on the sensor reading and loop closure. Hanel et al ([3]) used laser based FastSLAM to structure the trajectory of their robot as well as the positions of passive UHF RFID tags. Thus, given the maps of transponders, the robot was finally able to localize itself with only RFID and odometry. Kleiner et al ([5]) have performed trajectory correction and GraphSLAM with sparsely spread passive transponders. Other works have exploited the active RFID tags. For example, Kantor et al (Kantor et al. (2003)) used EKF for localization, mapping and SLAM. They exploit measured signal strength between the transponders which is not a standardized feature in passive RFID systems. Lu proposed a demonstration of a system for passive UHF tags that exploits the directionality of RFID readers. Beliefs of the positions of tagged objects are formed by varying one time robot poses. Yamano et al ([6]) examine how Support Machine Vector could learn robot localizations. This is applied by generating feature vectors out of the signal strength information gained from ac-

tive RFID tags. Chae et al (Chae and Han. (2005)) computed a weighted sum of the currently detected tags positions. Then the robot was localized at a finer scale by monocular vision involving SIFT features. Tsukiyama et al ([7]) developed a simple navigation mechanism on the basis of vision for free space detection. RFID tags were considered as labels within a topological map of an indoor environment. Recently Zhou et al ([8]) proposed a vision based indoor localization method that used modified active RFID tags. These tags were equipped with LEDs which make the recognition much easier. Ziparo et al ([9]) used RFIDs to coordinate a team of robots for an exploration in unstructured areas. Raoui et al ([10]) presented also two strategies for metrical and topological navigation with tags merged on shelves and on the ground. Concerning navigation with visual landmarks, Hayet et al ([11]) developed a method which extracts planar rectangles from edge segments through a relaxation scheme.

3 Metrical and deterministic method for self-localization

We propose an approach for topological navigation based on sparsely distributed RFID tags. The operator sets RFID tags (with known labels) in dedicated places so that when the robot receives the signal from one tag, it knows that this tag is in the reception field of the antenna. Figure 3 represents a simulated environment, with a simulated trajectory the robot has to execute : the blue dots numbered from 1 to 42 are RFID tags the positions of which are assumed to be known at this step. The robot starts from the position X_1 ; its position after a motion between two successive positions X_i and X_{i+1} is predicted from odometry. The robot model is known so that the odometer delivers motion measurements $(u;Q)$ in the current robot reference frame, with $u = (dx; dy; d\theta)$ and Q the covariance matrix on u . LAAS has evaluated how a virtual robot could cope with self-localization when executing this trajectory, using a stochastic framework that allows to fuse measurements acquired by odometry (in order to predict the robot position from the estimated motions) with other information coming from the reception of a RFID signal or from the detection of a visual landmark. By now, only metrical localization is evaluated, without taking into account the world structuration in different areas. Figure 4 describes the different steps required to cope with robot localization from the observation of RFID tags.

We analyze these steps between the two positions 2 and 3: the true positions are presented in (a). Two tags labelled 5 and 12 will be detected when arriving at position 3. In (b) the estimated X_2^e position is presented with the elliptic uncertainty area in which the true robot position must be with a probability 0.95: this ellipse is computed from the covariance matrix P_2^e on the position vector $(X; Y; \theta)$. In (c), the robot moved from X_2 to X_3 , and predicts its new position from the odometry measurements u , thanks to a function F : $X_3^* = F(X_2^e, u)$ The error P_3^* on X_3^* is



Figure 1: The RACKHAM demonstrator from LAAS(left); The Shopping Trolley demonstrator from FZI (right)



Figure 2: Reception field of one antenna (left). Reception field of eight antennas monted on a circular robot (right)

computed using a linearization of F, using jacobians of F with respect to X and u:

$$P_3^* = \frac{dF}{dX} P_2^e \frac{dF^t}{dX} + \frac{dF}{du} Q \frac{dF^t}{du}$$

Figure 5 shows the predicted robot position X_i for the simulated trajectory, executed without observation. The odometry errors are cumulative, so that at the end, the robot position prediction has a high uncertainty. This uncertainty can be maintained constant when observing RFID tags. In (d) the robot receives RFID signals. If the robot is only equipped with one omnidirectional antenna, when it receives the signal from an RFID located in a position $(X_t; Y_t)$, we can apply a constraint on its position $(X_r; Y_r)$:

$$(X_t + X_r)^2 + (Y_t + Y_r)^2 < R^2$$

where R is the maximal distance between the tag and the antenna. So without considering the orientation, when the robot receives in X_3 , the RFID signals from tags 5 and 12, it means that its true position is located inside the two discs drawn in figure (d). Using the classical EKF-based framework for robot localization, it is not possible to express such a constraint: so we apply here a particle filtering approach. Hypothesis on the robot position are randomly selected from the gaussian distribution (X_3^*, P_3^*) : then the likelihood of each particle is estimated with respect to the observation constraints. In figure (e) only the particles in the two discs intersection are kept, and finally in (f), the estimated position X_3^e is computed using the barycenter of the acceptable particles, and the uncertainty P_3^e is evaluated from the eigen vectors and eigen values of the cloud of the acceptable particles. Figure 4 shows the estimated robot positions X_i^e when executing all the trajectory taking into account RFID observations. Using only an omnidirectional antenna, it is not possible to update the robot orientation. But if the robot is equipped with several directional antennas, other constraints can be applied on the robot position and orientation from the observation of one RFID tag from one known antenna. The Rackham demonstrator (figure 1(right)) is equipped with 8 directional antennas. Figure 2(left)) presents the calibrated reception fields: antennas receives signals emitted in a 120 deg cone, from less than 4,5 meters. A tag can be received from one, two or three antennas, depending on its position with respect to the robot in the red, blue and green regions. When a tag located in $(X_t; Y_t)$ is received by an antenna located in $(X_a; Y_a)$ with an orientation θ_a with respect to the world frame (see figure 2(right)), it gives two new constraints: the tag must be in the reception field, i.e. in the disk, but also between two straight lines:

Similar constraints can be applied also if a tag is not received. So these constraints are applied in order to estimate the likelihood of a robot position estimate from observations of tags with our RFID reader connected to eight antennas. Figure 6 shows the estimated robot positions and orientations X_i^e with the simulated

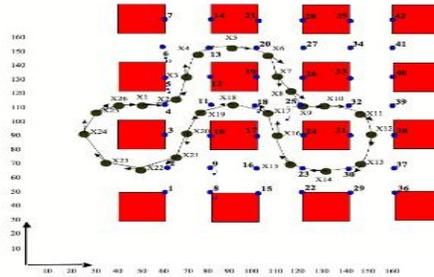


Figure 3: Simulated trajectory

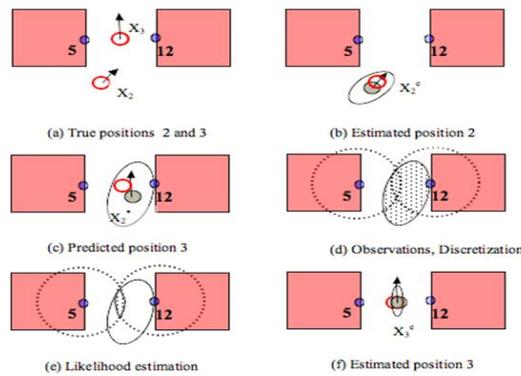


Figure 4: Mixed approach: EKF used to fuse odometry measurements, MLE used to evaluate consistent hypothesis with the RFID observations

environment and trajectory

3.1 Filtering non-observations

In the previous section, we have presented our approach to do deterministic localization using the Kalman filter. However, this method doesn't use the information about the antennas of the robot which does not have observations. These information can be integrated to have more precision in the localization. In order to increase such accuracy, we apply the following algorithm on each step of the robot path :

- Computing the robot observation in the current step based on the model of the antennas.
- Finding the particles around each predicted position with a covariance of P_y computed in the Kalman update step, that receive the same RFID tags as the

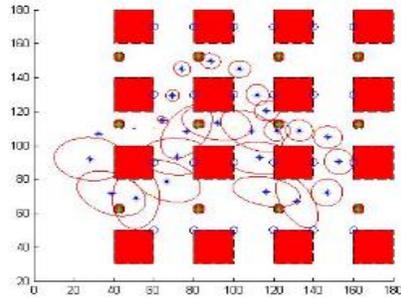


Figure 5: Predicted positions from the robot model (odometry, EKF update of the last estimated position)

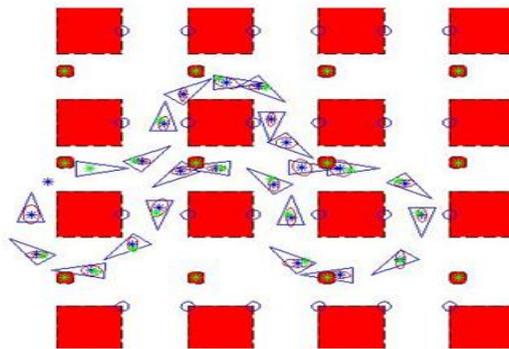


Figure 6: Robot localization at different positions with the computation of standard deviation for x, y, θ

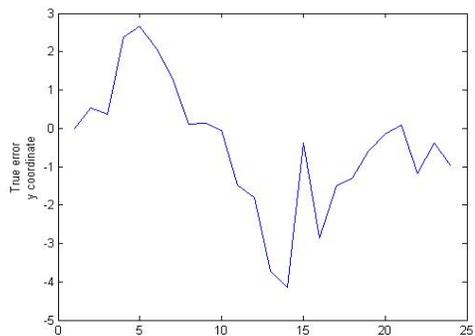


Figure 7: True error, Y coordinates

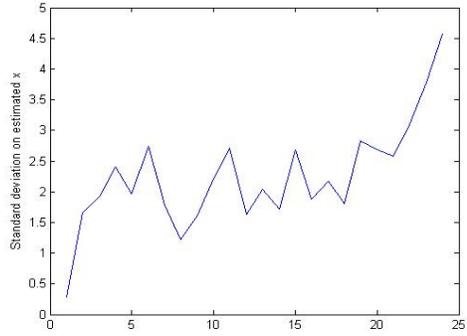


Figure 8: Standard deviation on estimated x

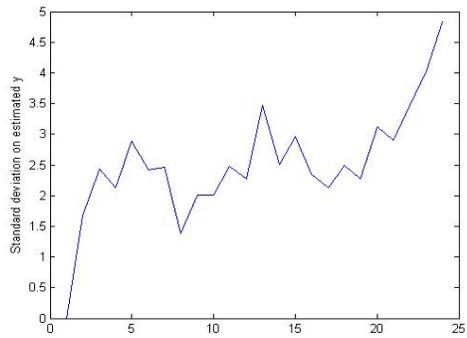


Figure 9: Standard deviation on estimated y

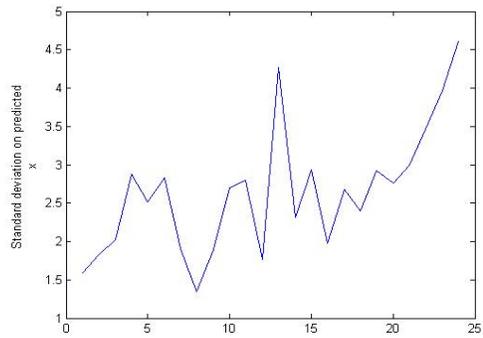


Figure 10: Standard deviation on predicted x

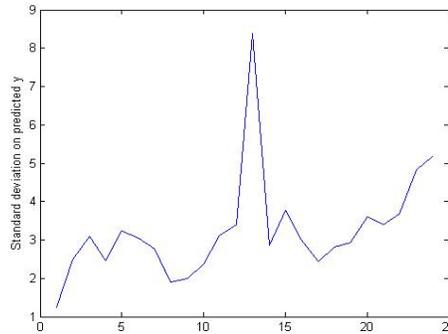


Figure 11: Standard deviation on predicted y

observation.

- Considering only the particles that receive the same tags as observation.;
- Rejecting particles that receive other tags

We do statistical measures in order to show how the performances of localization are improved with the non observation operation. For that, we move the robot with different error noises. At each cycle, we compute the standard deviation of measurements ($x_{Est}-x_{True}$) for the case of using non observations or not using non observations. The results are presented in figure 13.

4 Stochastic localization using particle filter

The method is based on the particle filter and includes some modifications that improve the localization performance. We consider then the approach based on the modeling of physical properties of the sensor and the observation process. We explain the principal steps of the algorithm and the improvement that we do. First, we initialize the algorithm with a uniform distribution of the positions of our environment if we don't know the first position of the robot; or with a distribution centred on the first position if we know it. Then at each iteration, we apply the following steps([12]) :

- Prediction of the movement :in this step we use the displacement estimated by the odometer and the displacement model for taking the next position in the probability distribution of the next positions. We modify this behavior by taking N_t positions instead of 1 position. We obtain the set M_k and we associate for each particle $Mk[i]$ the probability of $\frac{F_{k-1}[i]}{N_t}$. We obtain then $N_t \cdot N_{sample}$ particles.

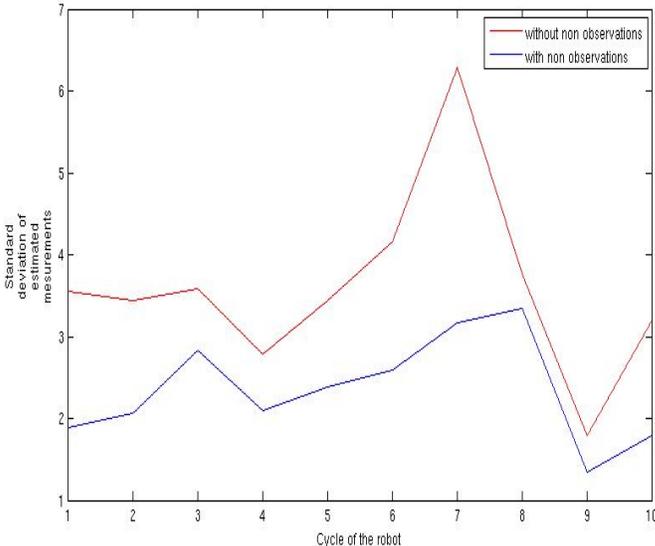


Figure 12: Standard deviation of $x_{Est}-x_{True}$ for 10 robot cycles, the application of non observation reduces it. red line is without non observation, blue line is with non observation

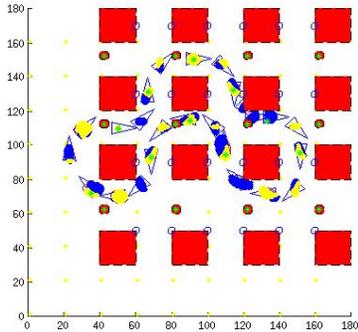


Figure 13: Evolution of the distribution of particles (by consideration of non observations, we keep only yellow particles)

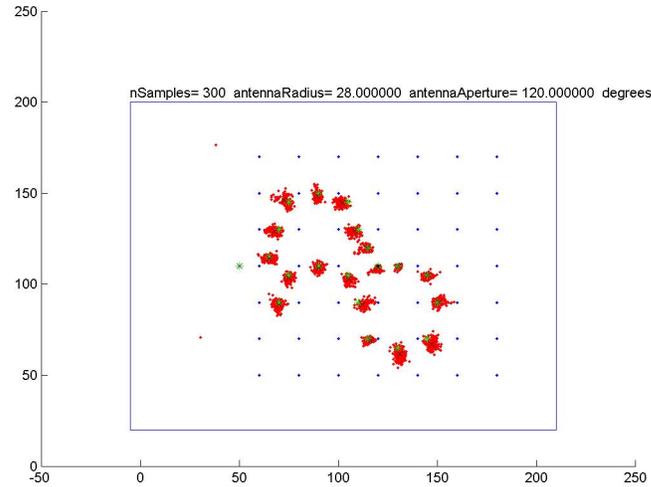


Figure 14: Stochastic simulation, the particles are illustrated with red points

- Insertion of random particles : We insert N_{aux} particles uniformly distributed in the environment with an association of a low probability p_{aux} . This step allows a quicker correction if the robot is lost which influences all the generated particles
- Integration of the observations: We change the probabilities of $N_t \cdot N_{sample} + N_{aux}$ points with the measure of the correspondences with observations.
- The resampling step This step takes in entry the precedent points with their new probabilities and generates the final set with taking uniformly N sample particles among them. The probabilities associated to these new particles will be equal each other and equal to $\frac{1}{N_{sample}}$.

We present below the results of the simulation in which 300 samples are used and with the probability distribution of the odometer set to be 3 and the number of injected particles set to be 30. In figure 12, we present the aggregation of the first 20 displacements. These correspond to displacement in a rich environment with tags.

4.1 Information gain based active localization

In order to maximize localization performances, we use an intelligent control strategy named active localization. It consists of coupling control actions into the estimation process. Then, an information theoretic control is used. In fact, information measures quantify the uncertainty in a probabilistically representation estimation

Algorithm 1 Algorithm ActiveMCL(X_{t-1} , u_t , z_t , m)

```

1:  $X_t \leftarrow 0$ 
2: for  $m \leftarrow 1$  to  $M$  do
3:   for  $i \leftarrow 1$  to  $I$  do
4:      $x_{t,i}^m \leftarrow \text{sampleMotionMmodel}(u_{t,i}, x_{t-1}^m)$ 
5:      $w_{t,i}^m \leftarrow \text{measurementModel}(z_t, x_{t,i}^m, m)$ 
6:      $X_{t,i} \leftarrow X_{t,i} + \langle x_{t,i}^m, w_{t,i}^m \rangle$ 
7:   end for
8: end for
9: for  $i \leftarrow 1$  to  $I$  do
10:   $x_{t,i} = \text{mean}(x_{t,i}^m)$ 
11:   $x_{t-1,i} = \text{mean}(x_{t-1,i}^m)$ 
12:   $L(i) \leftarrow \text{entropy}(x_{t,i}) - \text{entropy}(x_{t-1,i})$ 
13: end for
14:  $i_{max} \leftarrow \text{max}(L)$ 
15:  $xa_t \leftarrow x_{t,i_{max}}$ 
16: for  $m \leftarrow 1$  to  $M$  do
17:  draw  $i$  with probability  $w_t^i$ 
18:  add  $xa_t^i$  to  $Xa_t$ 
19: end for
20: return  $Xa_t$ 

```

and are used as cost functions for potential control actions. Thus, the information metric is defined as a function of probability distribution.

$$I[x] = f[p(x)]$$

where $p(x)$ represents the estimates of the robot positions $[x \ y]$, which is considered as a gaussian with a mean $\hat{hat}(x)$ and a covariance P . The information metric is $h(x) = \frac{1}{2} \log[(2\pi e)^n |P|]$. The information gain is defined as the difference in the information of our estimation before and after a particular action.

$$I[x, a] = h[p(x/a)] - h[p(x)]$$

While the vehicle moves it follows the following procedure :

- choose a trajectory that maximizes $I[x,a]$
- Propose several trajectories
- Estimate the observation based on the antenna reception field that will be made along each trajectory

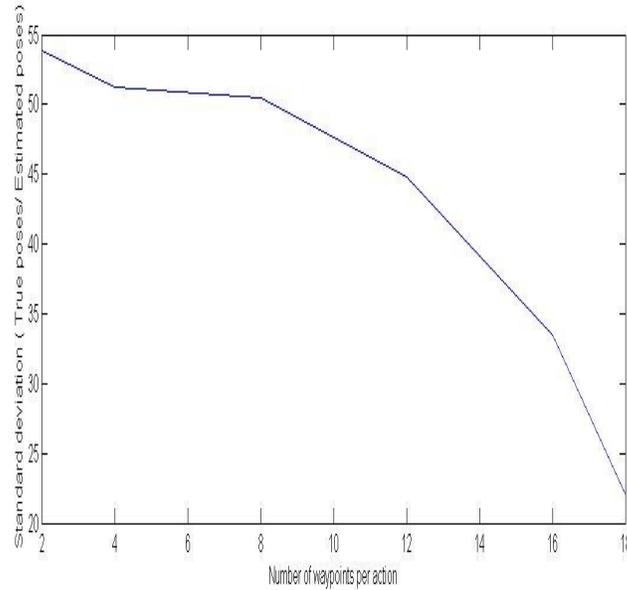


Figure 15: Robot localization at different positions with the computation of standard deviation for x, y, θ

MCL represents the belief of x_t by a set of M particles. Line (4) in our algorithm create samples from present belief as starting point. The measurement model is then applied as indicated in line (5) to determine the importance weight of that particle. The implementation model is done by using the perception model of RFID antennas. In line (3) we make an iteration over all possible actions from t to $t+1$. So we have possible posterior positions $x_{t,i}^m$ and weights $w_{t,i}^m$. The initial belief $bel(x_0)$ is obtained by randomly generating M such particles from the prior distribution $p(x_0)$, and assigning the uniform importance function M^{-1} to each particle. For each action i , we compute the difference in the entropy between the mean of the particles at time t and $t+1$. In line 12, we look for the index of the action that maximizes $L(i)$.

In order to evaluate the method, we calculate an error value according to the number of way points during the motion of the vehicle. That's why we compute :

- The covariance of the particles during the motion of the robot.
- The covariance of a randomly distributed particles around each true pose.
- The difference between these two values.

5 Self localization with visual landmarks

In this section, we choose to use punctual visual landmarks to correct the positions of our robot. The use of these features improve the localization and many researchers prefer to use them.

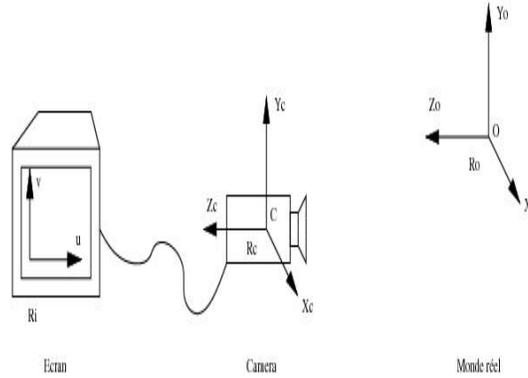


Figure 16: Geometrical referential related to camera calibration

Several methods exist to represent the process of image formation. The simplest one is the pinhole model. The pinhole camera model is based on an ideal pinhole camera. It represents the relationship between the coordinates of a 3D point and its projection onto the image plane. Here, the camera aperture is described as a point. Geometric distortions or blurring of unfocussed objects are not included in this model. Also, it does not integrate the fact that most practical cameras have only discrete image coordinates. This system respects the Gauss conditions. To describe this process, we just represent the relations between the space of the world and those of the image, then express the projection of the camera space on the image plane and apply the affine transformations which give the image coordinates. The relations of a point M, in the space of the world (X,Y,Z,1) which image coordinates are (su,sv,s), are expressed as :

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} k_u & s_{uv} & c_u \\ 0 & k_v & c_v \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} & t_x \\ R_{2,1} & R_{2,2} & R_{2,3} & t_y \\ R_{3,1} & R_{3,2} & R_{3,3} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ z \\ 1 \end{pmatrix}$$

The parameters employed in this model are divided into two categories: extrinsic and intrinsic parameters. Extrinsic parameters are : F: focal distance k_u and k_v : zoom factors in the image c_u and c_v : projection coordinates of optical center in the image plan. $S_{u,v}$: related to the non orthogonality of rows and columns of electronic cells of the camera. This parameter is neglected . Intrinsic parameters : $R_{3 \times 3}$:matrix of rotation that passes from the world of work space to world of the camera. t_x, t_y, t_z : components of the translation vector that passes from workspace to camera. We use the extended Kalman filter ([13])to compute the position of the robot. The innovation is given with : $Innov = z - z_{pred}$, with

$$Z_t = h^t(X_t) = \left[\begin{array}{c} \sqrt{((\delta t)^2)} \\ \arctan\left(\frac{\delta t(2)}{\delta t(1)}\right) - \theta_t \end{array} \right]$$

and

$$\delta(1) = u - x_{veh}$$

$$\delta(2) = v - y_{veh}$$

We consider x_{veh}, y_{veh} and θ_{veh} the coordinates of the robot.

$$[x_{veh} y_{veh}] = [x_{pred} y_{pred}]$$

for z_{pred}

$$[x_{veh} y_{veh}] = [x_{true} y_{true}]$$

for z

Thus : x_{Est} , the estimated position of the robot, is computed with :

$$x_{Est} = x_{Pred} + Innov.K$$

K is the Kalman coefficient.

6 Mapping with RFID tags

In our approach, we do mapping separately from localization, which means that we perform first localization, obtain the positions of our demonstrator, then we do mapping. As described before, the conception of maps is a key in the robotic navigation, thus it attracts a great interest of the robotic community. We start in our application by calculating the robot positions using deterministic or probabilistic methods as described in the section dealing with localization. Besides, we present two methods :

6.1 Deterministic method

The robot circulates in the environment. At each time it detects a new tag, it reduces its areas of mapping. Supposing it is firstly in the area A, it draws a zone of existence that corresponds to the model of perception of the antennas (see figure 16), after advancing, the new zone is the intersection of the two zones which is the part B of the figure 16, and so on. We follow these steps until we don't receive any more tag. In the following, we present the algorithm which describes our method.

6.2 Probabilistic method

While the robot moves, it verifies whether it receives some tags. If not, it continues until it receives a tag. It discretizes the zone according to the perception model and then, for each particle our demonstrator verifies if it is received from the past zones.

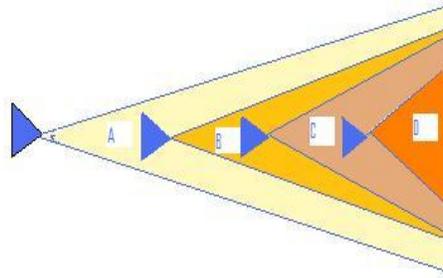


Figure 17: Movement of the robot. At each step, it reduces the zone of the estimated tags

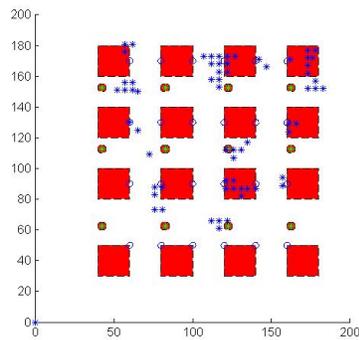


Figure 18: Estimated positions of the tags with blue stars (First algorithm)

Algorithm 2

```

1: for tag  $\leftarrow$  1 to N do
2:   for robot-position  $\leftarrow$  1 to P do
3:     detected-tags = scan(environment)
4:     if detected-tags  $\neq$   $\emptyset$  then
5:       memorize this zone  $z_{robot-position}$ 
6:       intersect with the precedent  $z_{robot-position-1}$ 
7:     end if
8:   end for
9: end for

```

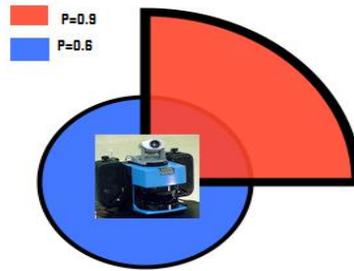


Figure 19: Simplified sensor model for one robot antenna

If not, it is discarded. We need to know the posterior $p(x|z_{1:t})$ for each particle. x is the predicted pose of the tag and $z_{1:t}$ represents the data gathered in the time step 1:t. We use the Bayes rule which considers the assumption of independence of consecutive given measurements. We obtain the following recursive update rule :

$$p(x|z_{1:t}) = \alpha \cdot p(z_t|x)p(x|z_{1:t-1})$$

$p(z_t|x)$ specifies the likelihood of the observation z_t given the pose x of the tag relative to the robot pose.

The model of perception of the antennas consists of 2 components. Figure 19 shows the detection range for each antenna. It consists of an arc with an opening angle of 95 degrees in the direction of the antenna. Besides, RFID tags which are close are always detected. This is modeled by a circle around the center of the receiver. Figure 19 also depicts the corresponding likelihood for two detection ranges.

We apply this method by considering the posterior positions which do not receive any tag that allows to filter more particles. We evaluate our method by computing both for x and y coordinates of the tag, the difference between the average of the predicted positions, and its true position. We show in figure 21 the error on x coordinate (blue), and on y coordinate (green). The accuracy is found to be about

Algorithm 3

```

1: for tag  $\leftarrow$  1 to N do
2:   for robot-position  $\leftarrow$  1 to P do
3:     repeat
4:       R=Memorize the robot position
5:     until received-tags  $\neq \emptyset$   $P \leftarrow$  ellipse(robot-position)
6:     for  $x_i \leftarrow$  1 to size(P) do
7:
8:       
$$p(x_i|z_{1:t}) = \alpha \cdot p(z_t|x_i)p(x_i|z_{1:t-1})$$

9:       if  $R_i$  receives  $p_i$  then
10:         reject  $p_i$ 
11:       end if
12:     end for
13:   end for
14: end for

```

0.2 m on x axis, and 0.4 m on y axis.

7 Conclusion

In this paper, Firstly, we have presented two approaches for self localization using deterministic and probabilistic methods in order to localize a robot in a commercial center. These methods use respectively Kalman and particle filters. By now the Monte Carlo localization has been implemented. To improve the performances, we have discarded the predicted positions that receive tags not belonging to the observation. We have used a simplified model to heuristically estimate the entropy of the map which maximizes the accuracy of localization. Besides, we have changed our landmarks from RFIDs to visual features. The pinhole camera model is suitable to perceive these landmarks. Secondly, we have developed two methods which generate maps of RFID tags. Our sensor model allows us to compute the likelihood of tag detection given the relative pose of trajectory.

References

- [1] H. Kaindl, E. Arnautovic, D. Ertl, and J. Falb., "Iterative requirements engineering and architecting in systems engineering," *Proceeding of ICONS*, 2009.
- [2] N. Fairfield and D. Wettergreen, "Active slam and loop prediction with the segmented map using simplified models," *FSR*, 2009.

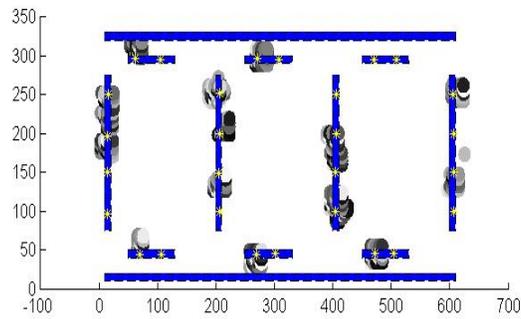


Figure 20: Estimated positions of the RFID tags. The color of the circles represent the posterior probability of the corresponding positions

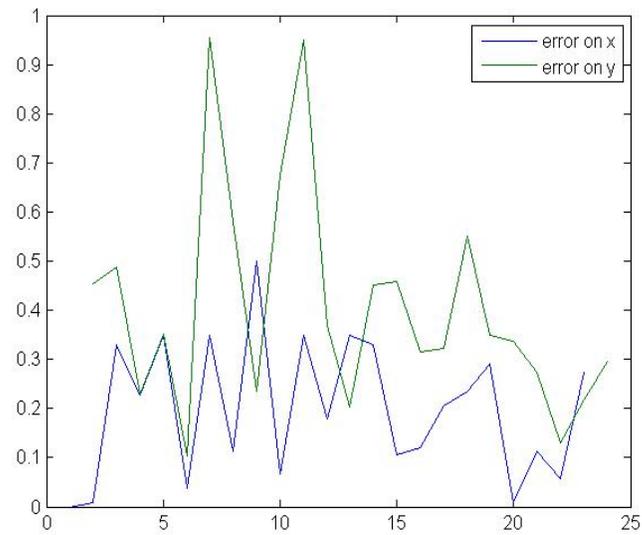


Figure 21: Error positioning of the tags

- [3] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Filipose, “. mapping and localization with rfid technology,” *Proceedings of ICRA*, 2005.
- [4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “Fastslam: A factored solution to the simultaneous localization and mapping problem.” *Proceeding of AAAI*, 2002.
- [5] A. Kleiner, C.Dornhege, and S.Dali, “Mapping disaster areas jointly: Rfid-coordinated slam by humans and robots.” *Proceeding of SSRR*, 2007.
- [6] K. Yamano, K. Tanaka, M. Hirayama, E. Kondo, Y. Kimuro, and M. Matsumoto, “Self-localization of mobile robots with rfid system by using support vector machine,” *Proceeding of IROS*, 2004.
- [7] T. Tsukiyama., “Navigation system for mobile robots using rfid tags.” *ICAR*, 2003.
- [8] Y. Zhou, W. Lu, and P. Huang, “Laser-activated rfid-based indoor localization system for mobile robots,” *ECMR*, 2007.
- [9] V. Ziparo, A. Kleiner, A. Farinelli, and L. Marchetti, “Cooperative exploration for usar robots with indirect communication,” *Proceedings of IAV*, 2007.
- [10] Y. Raoui, M. Goller, M. Devy, T. Kerscher, J. Zollner, R. Dillmann, and A. Coustou, “Rfid-based topological and metrical self-localization in a structured environment,,” *Proceeding of ICAR*, 2009.
- [11] J. Hayet, F. Lerasle, and M. Devy, “A visual landmark framework for mobile robot navigation,” *Proceeding of CVPR*, 2003.
- [12] P. Vorst, S. Schneegans, B. Yang, and A. Zell., “Self-localization with rfid snapshots in densely tagged environments,” *Proceeding of IROS*, 2008.
- [13] I. Ashokaraj, A. Tsourdos, P. Silson, and B.White, “Sensor based robot localisation and navigation: Using interval analysis and extended kalman filter.” *Proceeding of IROS*, 2004.

Received: December, 2010