Semi-Autonomous Learning of an RFID Sensor Model for Mobile Robot Self-Localization

Philipp Vorst and Andreas Zell

Department of Computer Science, University of Tübingen, Tübingen, Germany *
{philipp.vorst,andreas.zell}@uni-tuebingen.de

Summary. In this paper, we present a method of learning a probabilistic RFID reader model with a mobile robot in a semi-automatic fashion. RFID and position data, recorded during an exploration phase, are used to learn the probability of detecting an RFID tag, for which we investigate two non-parametric probability density estimation techniques. The trained model is finally used to localize the robot via a particle filter-based approach and optimized with respect to the resulting localization error. Experiments have shown that the learned models perform comparably well as a grid-based model learned from measurements in a stationary setup, but can be obtained easier.

1 Introduction

Radio frequency identification (RFID) is nowadays not only used for identification purposes in the industry, but also for navigation tasks in mobile robotics. The technology allows for the contactless identification of objects and landmarks which are marked with RFID tags (also called labels or transponders) by a reader device and its antennas via radio waves. Passive tags obtain the energy for operation and response from the radio field of the RFID reader, which makes them inexpensive and easily maintainable. In case of passive UHF technology as in this work, however, factors such as the relative position of a tag and nearby materials affect the readability of a tag. Hence, in practice detection rates can be poor and noisy, and whatever application is regarded, it will benefit from an accurate model of tag detection probabilities. For example, the modeled detection field may lead to an improvement in the placement of RFID readers in a plant. Moreover, such a model is the basis of probabilistic localization algorithms. If it is easy to derive, it can be adapted or rebuilt quickly if the setup of the RFID system changes.

* This work has been funded by the Landesstiftung Baden-Württemberg within the scope of the support program BW-FIT and the research cooperation AmbiSense.

In this paper, we present a method of learning a probabilistic RFID reader model with a mobile robot in a semi-automatic fashion. We have chosen a non-parametric approach, which means that we do not claim any specific functional form of the tag detection probability density. The approach should thus be applicable to other tasks, RFID standards, and hardware. A question which arises is how the quality of the learned model can be measured. We decided to plug it into the target application – the localization of the mobile robot in our case – and measure the resulting error there. Using RFID for self-localization is motivated by the idea that, as more and more goods are being labeled with RFID tags and RFID hardware is getting cheaper, mobile robots operated e.g. in trade scenarios will be able to exploit RFID as a lower-cost sensor for self-localization and navigation in general.

We proceed as follows: First, the robot explores its environment and records RFID and position measurements. Then, a probabilistic model of tag detection frequencies is built from the logged data by estimating the probability density for detecting a transponder, conditioned on its relative position to an RFID antenna. For this, we compare two techniques: a simple grid-based method and a k-nearest neighbors search. The estimation step is repeated with different parameter values in order to search systematically for the model which minimizes the localization error of the mobile robot. Each learning technique only depends on one parameter.

This procedure is new insofar as related approaches do not learn the RFID reader model during the navigation of the mobile robot. Moreover, we do not only present a method of learning the model with the robot, but also of validating and optimizing it. The only assumptions that we make are firstly that the robot is equipped with a quite accurate reference localization module (e.g. laser- or vision-based) and secondly that we provide the robot with a list of some tags and their positions in the global frame of reference. The preparation of this list is the reason why our method is only *semi*-automatic.

With regard to related work, Hähnel et al. were one of the first to gain a probabilistic RFID sensor model and use it for Monte Carlo localization [1]. They measured tag detection rates for a single passive UHF tag on a grid of fixed distances and angles. We detail their approach in the subsequent sections, since we follow their method of particle filter-based self-localization. Bajcsy et al. obtained an RFID reader model in a similar fashion, but for several tags spread on the floor and with the possibility to tilt the RFID antenna [2]. Kloos et al. learned a sensor model for RF-based localization via a parametric approach modeling signal strength and distances for active (battery-powered) sensor nodes [3]. In [4], Kantor and Singh presented localization and mapping with RF beacons which provided distance information. They determined probability densities over actual ranges for a discrete number of measured ranges. Djugash et al. also used RF beacons for self-localization [5]. Their likelihood function was explicitly represented by a 2D Gaussian with standard deviation estimated from the variance in range measurements. By contrast, a sensor model-free approach to RFID-based self-localization is described in [6], but its mapping phase can be time-consuming.

This paper is structured as follows: In Sect. 2 we clarify the background of our work and show how a mobile robot is able self-localize with RFID. Thereafter, we present our approach to learning an RFID reader model in Sect. 3. Numerical evidence for the quality of the different techniques is provided in Sect. 4, before we finally summarize and discuss our work in Sect. 5.

2 Monte Carlo Localization with RFID Sensors

For localizing the mobile robot via RFID, we pursue the particle filter-based approach by Hähnel et al. [1], which is Monte Carlo localization [7] with an observation model adapted to RFID sensors. In Monte Carlo localization, the robot pose $\mathbf{r}_t$ is represented by an arbitrary probability density function (pdf) over the space of locations. This pdf is approximated by a set of n particles. Each particle consists of a pose hypothesis (x^i, y^i, θ^i) and a weighting factor w^i, where (x^i, y^i) are the coordinates in a global frame of reference, θ^i is the global heading of the robot, and w^i states the importance of the i-th particle. The filtering algorithm itself iteratively performs three steps:

1. *Prediction*: The robot pose at time t is predicted by propagating all particle positions according to the latest odometry readings $\mathbf{o}_{t-1}$ and a motion model. Formally, one samples from the distribution $p(\mathbf{r}_t|\mathbf{o}_{t-1}, \mathbf{r}_{t-1})$.
2. *Correction*: Sensor data $\mathbf{z}_t$ are incorporated into the set of particles by correcting the particle weights according to some likelihood function $p(\mathbf{z}_t|\mathbf{r}_t)$:
$$w_t^i = \eta \cdot p(\mathbf{z}_t|\mathbf{r}_t)$$
Here, η is a normalizing constant which ensures that $\sum_{i=1}^{n} w_t^i = 1$.
3. *Resampling*: A new set of n particles with equal weights $1/n$ is obtained from the old one by drawing n samples from the old set of particles, where the probability of choosing particle i corresponds to its weight w_t^i. An option is to resample only if the estimate $\hat{n}_{\mathit{eff}} \approx 1/\left(\sum_{i=1}^{n}(w_t^i)^2\right)$ of the so-called effective sample size falls below some threshold, e.g. $n/2$.

Particle filtering has turned out to be a robust and versatile method for self-localization, even in presence of non-Gaussian noise and highly imprecise measurements, as it is the case also for self-localization with RFID.

To learn a sensor model of the RFID reader means in this paper to learn the likelihood function $p(\mathbf{z}_t|\mathbf{r}_t)$ required by step 2 of the algorithm. This function should represent the likelihood that the observed RFID measurements provide evidence for the current robot pose. More specifically, the measurements $\mathbf{z}_t$ at time t consist of two lists of detected RFID tags, $\mathbf{d}_t^l$ and $\mathbf{d}_t^r$, one for the left and one for the right RFID antenna of our robot (see Fig. 1 (a)). Assuming that both measurements are independent, we set $p(\mathbf{z}_t|\mathbf{r}_t) := p(\mathbf{d}_t^l|\mathbf{r}_t)p(\mathbf{d}_t^r|\mathbf{r}_t)$. The two $\mathbf{d}_t$ can be regarded as sequences $\mathbf{d}_t = (d_t^1, d_t^2, \ldots)$, where $d_t^i \in \{0, 1\}$ states whether $(d_t^i = 1)$ or not $(d_t^i = 0)$ transponder i was detected.

3 Learning an RFID Sensor Model

Gaining a model of an RFID reader means in our case to estimate the probability at which RFID tags can be detected from the perspective of a single antenna, conditioned on a number of parameters of the system. Hence, we first have to choose the types of parameters of the model and then decide on how the probabilities can be computed from the raw binary data. Recall that the information given by an RFID reader is only which transponders have been detected, not their direction or distance.

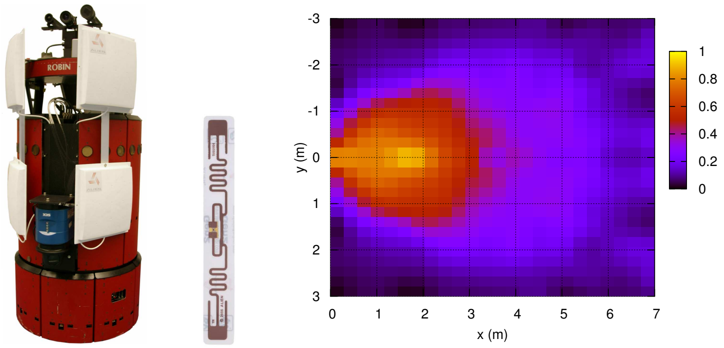
In this paper, we restrict the set of model parameters to the relative position $\mathbf{x} = (x, y)$ of an RFID tag to the antenna. As indicated in the introduction, this is a vast simplification, but works surprisingly well. Besides, parameters such as the materials in the vicinity of RFID tags are difficult to be taken into consideration. So, formally we wish to estimate the detection probability $q_l(\mathbf{x}_l) := p(d_l|\mathbf{x}_l)$ for some tag l, given its relative position $\mathbf{x}^l$ to an antenna. We further assume that q does not depend on the specific tag l, so we will simply write $q(\mathbf{x})$.

3.1 Fixed-Setup Recordings

In order to estimate $q(\mathbf{x})$, the approaches presented in related papers usually choose different fixed positions of the robot, equipped with RFID antennas, and some tag. For each such position $\mathbf{x}$ on a discrete grid, the numbers $n^+(\mathbf{x})$ and $n^-(\mathbf{x})$ of successful and failing detection attempts are counted in order to derive $q(\mathbf{x}) = n^+(\mathbf{x})/(n^+(\mathbf{x})+n^-(\mathbf{x}))$. Hähnel et al., for instance, attached an RFID tag to a cardboard box and rotated the robot in front of it [1]. By repeating these measurements for various distances, they gained a mapping from a discrete number of relative distances and angles to the detection frequencies of a tag. We also recorded the detection frequencies on a grid of relative positions, depicted in Fig. 1 (c). There, the measurements were averaged over different heights and relative orientations of a single tag with respect to the RFID antenna.

The advantage of supervised grid-based measurements is that they are taken at precisely known positions of the grid, and the detection probability on a grid point is simply the rate of successful detection attempts measured there. If only one tag is taken into account, however, recording the RFID inquiries takes a lot of time. Even worse, the ideal case is investigated, claiming that the detection frequencies for one tag are independent of the presence of other tags.

In this paper, we pursue a slightly different approach. The idea is to exploit the fact that our ultimate goal is to localize a *mobile* robot. That is, provided that the robot is equipped with another, fairly accurate self-localization mechanism and that the robot knows at which positions RFID tags are fixed, it can explore its environment and continuously record tag detections. This procedure has a number of advantages: Firstly, much information is retrieved with



(a) (b) (c)

Fig. 1. (a) The RWI B21 robot used in our experiments, with its RFID antennas (white) and the front-mounted laser scanner (blue). (b) The type of tag (”squiggle”/”higgs” tag, by Alien Technology, approx. 10 cm × 1 cm) that we used for our studies. (c) The model obtained from fixed-setup measurements of a single tag. The x axis points to the opening direction of the RFID antenna, the y axis orthogonally, both parallel to the ground. The model was taken on 195 grid points and over 4 different heights and orientations with 100 measurements per configuration. Values between grid points are linearly interpolated. The recording took approx. 50 hours.

every RFID inquiry (each known tag i contributes a response value $d_i = 1$ or $d_i = 0$ at a time), which makes the recording phase fast. Secondly, one does not have to worry about which grid points to choose. Of course, this comes at the expense that computations get slightly more complicated. But the learned models are supposed to be more realistic in presence of more than one tag.

3.2 Overview of the Learning Steps

In order to learn the RFID reader model in a semi-autonomous fashion, our solution comprises three phases: the *exploration phase* (during which RFID and position measurements are recorded), the *offline learning phase* (during which the RFID sensor model is actually computed and optimized), and the *localization and validation* phase (in which the robot is able to localize itself only via RFID and validates the accuracy of the resulting pose estimates).

Exploration phase In the exploration phase, the robot logs RFID measurements and the poses where it takes these RFID measurements while traversing the environment. Note that therefore an additional positioning system (e.g. laser- or vision-based) is required which provides quite accurate position estimates. On the one hand, this seems to be a limitation of our approach. On the other hand, the models learned with our technique can later be used on other robots without the extra localization system. Moreover, the reference pose estimate enables us to assess the quality of the model with respect to self-localization during the last phase. And it is possible to combine the resulting RFID-based and the reference self-localization system, which has proven to be useful [1].

Offline learning phase In this phase, the sensor model function is actually learned from the logged data of the previous phase. As additional input, it requires a map of RFID tags and their positions in the global coordinate frame. Note that we found that establishing a short list of RFID tag numbers and their positions can be done faster than the supervision of RFID measurements for a fixed-setup model. In the following, we detail the steps of the offline training phase.

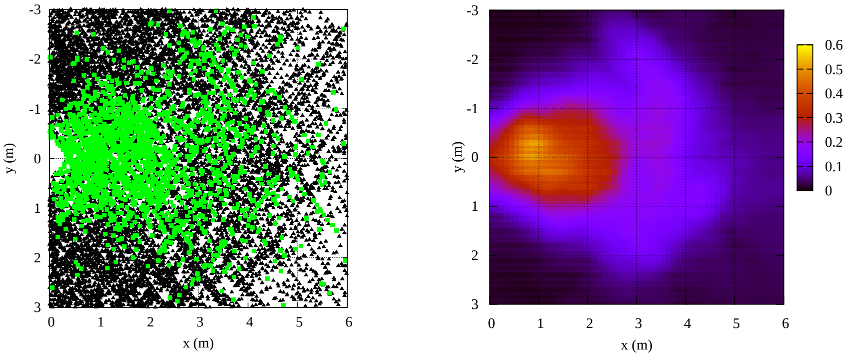
1. Given a map of transponder positions in the environment, the robot transforms the global tag positions into coordinates relative to the RFID antennas for each time step t when an RFID inquiry was performed. In this, each tag l contributes a sample position $\mathbf{x}_t^l$ in relative coordinates and a response value $d_t^l \in \{0, 1\}$, which states whether or not tag l was detected. From now on, we ignore the identity l of the tag and the time index t of the measurement. The result of this transformation is a set of samples $S := \{(\mathbf{x}_i, d_i)\}$ as depicted in Fig. 2 (a).
2. The raw samples are used to compute the conditional probability density $p(d|\mathbf{x}) = q(\mathbf{x})$ of detecting one tag, given its relative position to the antenna. Note that this function fully represents the target sensor model and is used as likelihood function in the correction step of the particle filter. For this step, we investigated two different techniques which are elaborated below.
3. The function $q(\mathbf{x})$ is evaluated at fixed positions of a fine-grained grid (with a resolution of 0.1 m) and stored as a look-up table. By this, $q(\mathbf{x})$ is only approximated, but the efficiency of evaluating the likelihood function for a potentially large number of particles is increased.

Online localization and validation phase The robot is finally able to localize itself, based on the method described in Sect. 2 and supplied with a map of RFID tags and the stored sensor model from the previous phase. Given the reference poses, it can validate and optimize the learned model.

3.3 Detection Probability Estimation

Grid-based Estimation One of the simplest approaches to estimating the tag detection probability $q(\mathbf{x})$ is to divide the detection field (see Fig. 2 (a)) into a uniform grid with square bins of length λ and count the observed tag detections for the resulting bins. Optimizing a model of this kind then means to find the λ which minimizes the localization error. Note that this method resembles the manual recordings with a fixed setup, but samples are not forced to lie on discrete grid points only.

The advantage of the grid-based approach is that $q(\mathbf{x})$ can be stored as a look-up table, since it only depends on discrete bin indices. This makes it very efficient. A problem, however, is that the grid introduces discontinuities in the density function. Moreover, λ is fixed for the entire detection



(a) (b)

Fig. 2. (a) The aligned positive (green/gray) and negative (black) samples as obtained from RFID measurements while the robot was moving around. (b) The sensor model learned via the k-nearest neighbors approach ($k = 2000$) from 11,429 positive and 102,950 negative samples which were recorded in about 50 minutes.

field. In parts where there are many samples available, a large value of λ will lead to over-smoothing, whereas a small value of λ will not be able to provide good estimates of $q(\mathbf{x})$ in areas for which only few (typically noisy) RFID measurements have been recorded. So, the optimal choice of λ may actually depend on the query position $\mathbf{x}$ in the detection field.

k-Nearest Neighbors (k-NN) Approach The k-NN approach overcomes the limitation of the grid-based approach in that it takes the density of training samples around $\mathbf{x}$ into account. In this method we consider the k samples which are closest to $\mathbf{x}$ and compute the probability of detecting a tag as $q(\mathbf{x}) = n^+(\mathbf{x}, k)/k$, where $n^+(\mathbf{x}, k)$ denotes the number of positive samples around $\mathbf{x}$ among the entire set of the k nearest samples in the vicinity of $\mathbf{x}$. In a sense, this approach is dual to the grid-based estimation technique: Instead of fixing the size of the bin inside which we count the response values d_i, we allow for arbitrary bin sizes and fix the number of samples from which we estimate the tag detection probability.

Optimizing a model of this kind then again means to find the value of k which minimizes the localization error. Analogously to the grid-based estimation technique, a too small value of k will not be robust to noisy data, but a too large value will lead to over-smoothing.

4 Experiments and Results

In order to evaluate and compare the different techniques of learning an RFID sensor model, we conducted a series of experiments with a B21 robot depicted in Fig. 1 (a). The robot is equipped with an Alien ALR-8780 RFID reader, two pairs of RFID antennas spanning an angle of 90°, and a laser scanner.

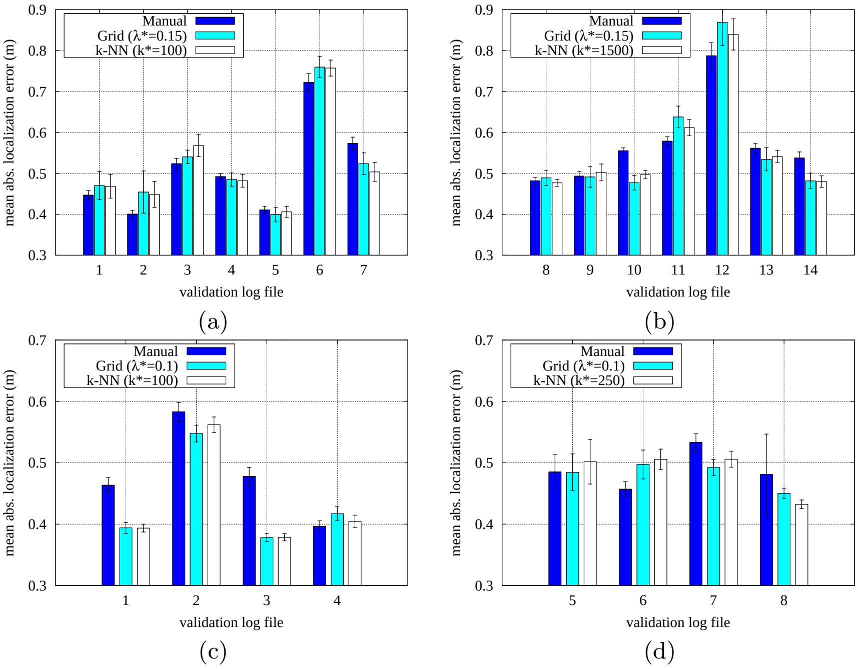


Fig. 3. Localization errors and standard deviations for the lab (a,b) and corridor (c,d) experiments. Each column represents one outcome of the 2-fold cross-validation: We first learned and optimized the models (the best choice of λ^* and k^*) on the one half of the log files. Thereafter, we measured the localization accuracy when using the optimized models on the respective other half of the log files.

All experiments were conducted in a laboratory and an adjacent corridor. We installed 39 transponders in the corridor and 23 in the laboratory, spread over $75\,\mathrm{m}^2$ and $50\,\mathrm{m}^2$, respectively. The tags were attached to walls and desks at intervals of 1-2 m, roughly at the height of the upper antennas. We recorded 14 log files in the lab and 8 log files in the corridor, containing RFID data and the corresponding poses at which the robot performed the RFID inquiries. Ground truth was provided by laser-based Monte Carlo localization, for which we used the CARMEN toolkit [8]. The localization error can be assumed below 0.1 m on average. Within an average duration of 7 minutes the robot traveled distances of 32-155 m at varying speeds. RFID data arrived at approx. 2 Hz, resulting in 712 RFID inquiries on average per log file. The offline optimizations took less than 45 hours on a 3 GHz PC in total. They did not require human intervention, in opposition to recording the hand-crafted model.

Each of the two sets of log files was randomly split into two halves for two-fold cross-validation. One half was once used to extract the raw RFID measurements and to optimize the models learned with the two techniques presented in Sect. 3, the other half was used to validate those mod-

Table 1. Localization errors of the fixed-setup (manual) model and the two learned models, averaged over the 700 validation runs in the laboratory and 400 validation runs in the corridor, respectively.

	Laboratory		Corridor	
	Validation 1	Validation 2	Validation 1	Validation 2
Manual model	0.5710 m	0.5101 m	0.4890 m	0.4803 m
Grid approach	0.5687 m	0.5190 m	0.4810 m	0.4342 m
k-NN approach	0.5641 m	0.5191 m	0.4864 m	0.4347 m

els, and vice versa. Validation means that we played back the log files and measured the mean absolute localization error when the particle filter used the learned model. Because the mean accuracy in RFID-based positioning strongly depends on a good initialization, we localized with known initial pose (tracking case). For each probability estimation technique, we systematically searched over several parameter values ($k = 25, 50, 100, 250, 500, 1000, 1500$ and $\lambda = 0.1, 0.15, 0.2, 0.4, 0.6, 0.8$). Then we committed to the best parameter choices λ^* and k^* and investigated the localization error with the optimized model on the validation log files over 100 repeated experiments. The localization accuracy achieved by the hand-crafted sensor model served as a benchmark. The outcomes of the experiments are visualized in Fig. 3.

The results show that the learned models yield very similar results as the hand-crafted model. This is in spite of the small inaccuracies in the reference positions. On some log files, the learned models outperform the fixed-setup model, on others they are inferior. In both cases, the difference is in the range of few centimeters and therefore comparatively low with regard to the typical mean absolute localization error of approx. 0.5 m. Table 1 also shows that on average the learned models perform similarly well as or even better than the fixed-setup model. Moreover, the two employed estimation techniques provide similar accuracy. The grid-based approach performs surprisingly well, despite the arguments mentioned in Sect. 3.3. We observed, however, that the k-NN approach seems to be slightly more robust to changes of its parameter k. Firstly, it is consequently no contradiction that for the cross-validation experiments in the lab we obtained rather different best parameters $k^* = 100$ and $k^* = 1500$ (see Fig. 3). Secondly, we would therefore consider the k-NN approach the first choice if a preliminary reference sensor model is to be created quickly or whenever optimizations play a minor role.

5 Conclusion

In this paper, we have shown how to gain a probabilistic model of an RFID reader semi-automatically with a mobile robot. During an exploration phase RFID and position data are recorded. The logged data are then aligned and used to learn tag detection frequencies by means of simple bin-based averaging and a k-nearest neighbors search. By repeating the learning step with different parameter values, one can systematically search for the model which minimizes the localization error of the mobile robot. Note that the learned reader models were used to localize a mobile robot and benchmark the quality of the models, but in general, their utilization need not be restricted to self-localization.

The employed two model learning methods yield similar results in self-localization as a sensor model which was recorded by hand. None of the two estimation methods should be preferred to the other one, although the k-nearest neighbor approach appears to be slightly more robust to changes of its model parameter. The presented approach eases the creation of an RFID sensor model, because tedious measurements are replaced by automatic offline computations and optimization. This enables one to change the setup of an RFID system and re-learn the model quickly.

For the future, we plan to extend the learning to more model parameters, e.g. the height of a tag over ground or its orientation. And we strive to further automate the exploration phase.

References

1. Hähnel, D., Burgard, W., Fox, D., Fishkin, K., Philipose, M.: Mapping and localization with RFID technology. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04). (2004)
2. Bajcsy, P., Kooper, R., Johnson, M., Soe, K.: Toward hazard aware spaces: Localization using passive RFID technology. Techn. Report ISDA06-002, National Center for Supercomputing Applications/Univ. of Illinois Urbana-Champaign (2006)
3. Kloos, G., Guivant, J.E., Nebot, E.M., Masson, F.: Range based localisation using RF and the application to mining safety. In: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006). (2006) 1304–1311
4. Kantor, G., Singh, S.: Preliminary results in range-only localization and mapping. In: Proceedings of the IEEE Conference on Robotics and Automation (ICRA'02). (2002)
5. Djugash, J., Singh, S., Corke, P.: Further results with localization and mapping using range from radio. In: International Conference on Field & Service Robotics (FSR'05). (2005)
6. Schneegans, S., Vorst, P., Zell, A.: Using RFID snapshots for mobile robot self-localization. In: Proceedings of the 3rd European Conference on Mobile Robots (ECMR 2007). (2007) 241–246
7. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99). (1999)
8. Montemerlo, M., Roy, N., Thrun, S.: Perspectives on standardization in mobile robot programming: The Carnegie Mellon Navigation (CARMEN) Toolkit. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003). Volume 3. (2003) 2436–2441