Self-Localization with RFID Snapshots in Densely Tagged Environments

Philipp Vorst, Sebastian Schneegans, Bin Yang, and Andreas Zell

Abstract— In this paper we show that, despite some disadvantageous properties of radio frequency identification (RFID), it is possible to localize a mobile robot quite accurately in environments which are densely tagged. We therefore employ a recently presented probabilistic fingerprinting technique called RFID snapshots. This method interprets short series of RFID measurements as feature vectors and is able to position a mobile robot after a training phase. It requires no explicit sensor model and is capable of exploiting given tag infrastructures, e.g., provided by supermarket shelves containing labeled products.

I. INTRODUCTION

Radio frequency identification (RFID) has found its way into robotics because of advantageous properties such as the unique identification of RFID tags and the resulting unambiguous association of sensor readings with landmarks. Moreover, since RFID reader and tags (also called transponders) communicate via electromagnetic waves, line of sight is not required between them. On the other hand, especially passive ultra-high frequency (UHF, 868/915 MHz) RFID imposes some difficulties: First, RFID readers of this type only report which tags have been detected; they neither know the distance nor the bearing to a detected tag (and do not even report signal strength information). Second, the successful detection of a tag within the read range of up to 7 m depends largely on factors such as the relative position of the tag with respect to the reader antenna as well as on the materials of nearby objects. Hence, today it is widely assumed that the use of RFID for self-localization is disadvantageous due to sensor noise and potentially low positioning accuracy.

In this paper, however, we show that it is possible to localize a mobile robot via RFID and odometry alone at an accuracy of less than 0.3 m. We present empirical results derived from extensive experiments with a mobile robot and off-the-shelf RFID hardware. Our approach is based on a fingerprinting technique called RFID snapshots, which we presented recently [8]. In this method, short series of RFID measurements are interpreted as feature vectors. They can be trained at reference positions during an offline phase and later be used to estimate the pose of the robot corresponding to current tag detections. A particle filter is employed in order to improve accuracy and robustness. The advantages of our approach are that no explicit sensor model is required and that environment characteristics are implicitly learned by the

training, unlike most model-based approaches. Furthermore, we do not constrain the distribution of RFID tags to specific positions, and tag positions need not be known.

This work focuses on densely tagged environments, which can be expected to prevail in supermarkets, warehouses, and logistics scenarios in general on the medium run. For these surroundings, localization via RFID is a cost-efficient solution if a mobile robot is already equipped with an RFID reader for inventory purposes or used as a shopping guide.

The structure of this paper is as follows: In Section II, we investigate related work. After presenting our localization method in Section III, we turn to practical issues of using RFID snapshots in densely tagged environments. The results of our experiments are provided in Section V. We finally conclude our work in Section VI.

II. RELATED WORK

A seminal survey into how to localize a mobile robot via RFID was presented by Hähnel et al. [5]. They first gained a probabilistic sensor model for their RFID reader, which associates the probability of detecting an RFID tag with the relative position of that tag to the antenna. Using this model, the positions of passive RFID tags in an office environment were mapped. Monte Carlo localization was then used to position the robot at an accuracy of approx. 0.5 m from RFID readings and odometry. An extension to this approach which learns the sensor model during the normal navigation of the robot was presented in [10].

Yamano et al. [11] employed active RFID tags. They localized a robot by learning feature vectors from signal strength information via support vector machines. Chae and Han [3] performed two-step indoor localization: First, they positioned coarsely via active RFID, and then the pose estimate was refined by means of monocular vision. Djugash et al. [4] utilized active RFID tags outdoors. They used time-of-flight measurements both for pure self-localization and for simultaneous localization and mapping (SLAM) via Kalman and particle filters. Kleiner et al. [6] performed SLAM with short-range RFID: They improved the dead-reckoning trajectories of humans and robots via synchronizing with sparsely distributed transponders outdoors. At the same time, the tag positions could be mapped, given the corrected paths.

A fingerprinting approach was pursued by Bahl et al. [1]: In their in-building RF-based system, WLAN signal strength measurements were used for combined localization via signal propagation modeling and fingerprinting. Lim and Zhang [7] presented fingerprinting-based positioning with passive UHF RFID tags, which is similar to our work. Their solution,

P. Vorst, B. Yang, and A. Zell are with the Computer Science Department, University of Tübingen, Sand 1, D-72076 Tübingen, Germany {philipp.vorst,andreas.zell}@uni-tuebingen.de
S. Schneegans is now with the Institute for Neuroinformatics, Ruhr University Bochum, Bochum, Germany sebastian.schneegans@neuroinformatik.rub.de

however, was non-probabilistic, and the transponders were attached to the ceiling at regular distances.

Similarly to our work, Bohn studied location estimation with high-density tag infrastructures [2]. In contrast to our approach, he utilized passive high frequency (HF) tags operating at 13.56 MHz, and the tags were spread on the floor rather than attached to walls and objects.

III. SNAPSHOT-BASED SELF-LOCALIZATION

Our approach compares current RFID measurements with data obtained during a training phase. These comparisons are embedded as observation model in a particle filter. For reasons of self-containedness, the latter is described briefly, before we detail the modeling of RFID snapshots.

A. Particle Filtering

In particle filter-based self-localization, also known as Monte Carlo localization [9], the robot pose $\mathbf{r}_t$ at time step 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 (also called samples). Each particle i consists of a pose hypothesis $\mathbf{r}_t^i = (x_t^i, y_t^i, \theta_t^i)$ and a weight w_t^i. (x_t^i, y_t^i) are the coordinates in a global frame of reference, θ_t^i is the global heading of the robot, and w_t^i states the importance of the i-th particle. A pose estimate $\hat{\mathbf{r}}_t$ can then be computed by $\hat{\mathbf{r}}_t = \sum_{i=1}^n w_t^i \mathbf{r}_t^i$. 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 a motion model which uses odometry readings $\mathbf{o}_t$ since the last iteration of the algorithm.
2) *Correction*: Sensor data $\mathbf{f}_t$ are incorporated into the set of particles by correcting the particle weights according to some likelihood function $p(\mathbf{f}_t|\mathbf{r}_t^i)$:
$$w_t^i = \eta \cdot p(\mathbf{f}_t|\mathbf{r}_t^i)$$
Here, η normalizes the weights such that $\sum_{i=1}^n w_t^i = 1$.
3) *Resampling*: A new set of n particles with equal weights $1/n$ is obtained by drawing n times a sample from the old set of particles. The probability of choosing particle i corresponds to its weight w_t^i. An option is to resample not always, but only if the estimate $\hat{n}_{eff} \approx 1/\left(\sum_{i=1}^n (w_t^i)^2\right)$ of the so-called effective sample size falls below some threshold, e.g., $n/2$.

In this way, particle filtering enables a robot both to localize itself globally and to track its position over time. It has turned out to be robust and versatile, even in case of non-Gaussian noise and highly imprecise measurements. These adverse conditions especially hold for RFID.

The depicted algorithm is generic insofar as the likelihood function $p(\mathbf{f}_t|\mathbf{r}_t)$ must be adapted to the specific application. This is done in the next section, where we tailor a likelihood function to RFID snapshots.

B. RFID Snapshots

In snapshot-based self-localization, so-called RFID inquiries are repeated over short series of cycles. An inquiry represents the attempt to read the identifiers of all transponders within read range. The list of detected tags and their detection counts is regarded as a fingerprint which characterizes the current pose of the robot.

In passive UHF RFID systems, both the power supply of a tag and the communication between the reader and the tags are realized by electromagnetic waves emitted from the reader antennas. The successful detection of a specific tag depends on whether it receives enough energy to reply to the reader. This itself depends on quite a large number of parameters, including the relative position and orientation of the tag from the reader as well as absorbing and reflecting materials in the vicinity of reader and tags. Instead of modeling these parameters, we regard the detection of a tag l as a random event which results in a successful detection with probability q_l. Prior experiments showed that this probability can be assumed fix for a given position of the RFID antenna and the transponder in the environment and to change only little for small changes of the robot pose.

As the result of an inquiry, the reader returns a list of detected tags for each antenna. This list can be interpreted as a vector $\mathbf{f} = (f_1, \ldots, f_k)$, where the value $f_j \in \{0, \ldots, N\}$ counts how often tag j has been reported during the recent measurement series. It is called *snapshot* in our approach, resembling a sensor fingerprint of the current situation. Commonly, N equals 1, which means that the RFID reader reports all tags which have been detected within one read cycle. For our work, it is of importance to consider series of measurements, i.e., we regard cases where $N > 1$.

During the training phase, a large number of these snapshots is stored together with the antenna positions $\mathbf{x}$ they were taken at. During self-localization, the snapshots are the basis for the computation of the particle weights.

In our application, the sensor data for every time step of the particle filter algorithm consist of two RFID snapshots – one from each of the two antenna pairs installed on the robot (see Fig. 1(a)). Assuming the independence of the measurements, we can write the likelihood function required for the correction step as
$$p(\mathbf{f}_l, \mathbf{f}_r|\mathbf{r}) = p(\mathbf{f}_l \,|\, \mathbf{a}_l(\mathbf{r})) \, p(\mathbf{f}_r \,|\, \mathbf{a}_r(\mathbf{r})), \tag{1}$$
where $\mathbf{a}_l(\mathbf{r})$ and $\mathbf{a}_r(\mathbf{r})$ denote the poses of the left and right antenna pairs if the robot is at pose $\mathbf{r}$, and $\mathbf{f}_l$ and $\mathbf{f}_r$ are the corresponding snapshots. We will show how the likelihoods of the form $p(\mathbf{f}|\mathbf{a})$ can be determined in two steps: First, an estimate $\hat{\mathbf{q}}$ of the detection probabilities of all tags for the antenna pose $\mathbf{a}$ is computed from the training snapshots taken in the vicinity. Then the probability of the current RFID measurement is determined based on these estimates.

Each training snapshot $\mathbf{f}$ allows for the estimation of the tag detection probabilities at the position where it was taken. Since the detection probabilities cannot be observed directly, we compute the Bayes estimate $\hat{q}_l(f_l)$ for a single tag l:

$$\hat{q}_l(f_l) = \int_0^1 q_l p(q_l|f_l) dq_l \tag{2}$$

$$\text{with} \quad p(q_l|f_l) = \frac{p(f_l|q_l)p(q_l)}{\int_0^1 p(f_l|q_l')p(q_l')dq_l'}. \tag{3}$$

The conditional probability $p(f_l|q_l)$ follows the binomial distribution, stating how likely a tag will be detected f_l (among N) times, given a detection probability q_l:

$$p(f_l|q_l) = \binom{N}{f_l} q_l^{f_l} (1 - q_l)^{N-f_l}. \qquad (4)$$

The term $p(q_l)$ in Eq. 3 is the prior detection probability q_l. It is modeled by a step function with a large value $p(q_l)$ if q_l is smaller than some ϑ close to zero and a constant low value in the rest of the interval $(\vartheta, 1]$. This is because in the major part of the environment, a specific tag will not be detected most of the time.

Now the idea is to select single training snapshots near an antenna position $\mathbf{a}$ as queried by the particle filter and to combine them to a reference snapshot. A reliable estimate is the weighted mean of the estimates obtained from training snapshots $\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(r)}$ recorded close to $\mathbf{a}$:

$$\hat{\mathbf{q}}(\mathbf{a}) = \alpha_1 \hat{\mathbf{q}}(\mathbf{f}^{(1)}) + \cdots + \alpha_r \hat{\mathbf{q}}(\mathbf{f}^{(r)}) + \beta \hat{\mathbf{q}}_0 \qquad (5)$$

with $\hat{\mathbf{q}} = (\hat{q}_1, \ldots, \hat{q}_k)^T$. k is the total number of tags known from training. The weights α_j and β are computed by

$$\alpha_j = \nu \cdot \exp\left(-\tfrac{1}{2} \cdot (||\mathbf{x}_i - \mathbf{a}||/\sigma)^2\right) \quad \text{and} \quad \beta = \nu \cdot \beta_0.$$

The $\mathbf{x}_i$ are the positions where the training snapshots $\mathbf{f}^{(i)}$ were taken. ν is a normalization factor which ensures that $\sum_{j=1}^{r} \alpha_j + \beta = 1$. Note that the metric $||\cdot||$ must also respect the orientations of the antennas. $\hat{\mathbf{q}}_0$ in Eq. 5 is an estimate of the detection probabilities in the absence of training scans. Its equal entries $\hat{q}_0$ can be obtained from the prior probability $p(q)$ via

$$\hat{q}_0 = \int_0^1 p(q)\, q\, dq. \qquad (6)$$

The weight β_0 is a constant small value such that the influence of $\hat{\mathbf{q}}_0$ decreases if many training snapshots are available close to $\mathbf{a}$ and correspondingly their α_i are large.

Finally, the estimate $\hat{\mathbf{q}}(\mathbf{a})$ can be used to compute the likelihood of a robot pose. To this end, the probabilities of the observed detection frequencies are determined by inserting the estimated detection probabilities $\hat{q}_l(\mathbf{a})$ of each tag into Eq. 4. Assuming the independence of the measurements of single tags, the likelihood of the entire current snapshot $\mathbf{f}$ is

$$p(\mathbf{f}|\mathbf{a}) = \prod_{l=1}^{k} p(f_l|\hat{q}_l(\mathbf{a})). \qquad (7)$$

For further details and a comparison of the snapshot approach to Hähnel's method, we would like to refer the reader to [8].

C. Adaptation of the Likelihood Function to Densely Tagged Environments

In densely tagged environments, a problem arises which is well-known from Monte Carlo localization, e.g., involving laser scanners: If the likelihood function is computed as the product of a larger number of independent likelihoods, as in Eq. 7, the particle weights will receive very small values. Besides numerical problems of the CPU in representing the particle weights, the single weights among the set of particles may differ by several orders of magnitudes. In these cases, even large numbers of particles do not suffice to represent the posterior distribution, and resampling will cause particle depletion because very few samples are replicated a large number of times. In the worst case, this ends up in the delocalization of the robot.

Since the value of k in Eq. 7 (the number of present/known transponders) is large in densely tagged surroundings (in our case $k \gg 100$), we decided to evaluate the likelihood function only partially. This is based on the abovementioned observation that $\hat{q}_l$ is close to zero for most tags l: At any position, only a relatively small fraction of known tags is likely to be detected. Those tags should be considered in likelihood evaluation, while the others can be ignored. On the other hand, if the snapshot $\mathbf{f}$ measured during localization contains a tag j which is considered negligible by the estimate $\hat{q}_j$, a particle should receive a lower weight because the reference snapshot does not explain the particle pose well enough. For this reason, we adapt the likelihood function of Eq. 7 such that only $k' < k$ tags are assessed:

$$p(\mathbf{f}|\mathbf{a}) = \prod_{l \in L} p(f_l|\hat{q}_l(\mathbf{a})), \qquad (8)$$

$$\text{where} \quad L = \{l_{s_1}, \ldots, l_{s_{n_t}}\} \cup \{l_{\pi_1}, \ldots, l_{\pi_m}\} \qquad (9)$$

The l_{s_i} are the tags of the current scan, $\pi_i : \{1, \ldots, k\} \rightarrow \{1, \ldots, k\}$ is a permutation such that $\pi_i < \pi_j \Rightarrow \hat{q}_i \geq \hat{q}_j$, and m is chosen such that $|L| = k'$. Also, the n_t tags l_i in Eq. 9 of the current RFID scan are sorted and cut off at k' if $n_t > k'$. So, essentially, the tag indices of the currently measured snapshot and the estimated reference snapshot are sorted by descending detection rates. Then, the likelihood function is evaluated on the first k' permuted tags, which are currently the most discriminative ones.

IV. PRACTICAL ISSUES

A. Parameter Choices

A remaining question is the choice of the parameter N, the number of measurements incorporated in one snapshot. Larger values of N are supposed to yield more discriminative snapshots, because the detection count of a specific tag will reflect the detection probability better which characterizes its relative position to the recording position of the snapshot. On the other hand, as can be seen from Tab. I, RFID inquiries take quite a lot of time. $N = 4$ repeated measurements per antenna last 1.5 s, for example. This is a problem because there are simply fewer pose corrections, and the robot may have moved significantly during two inquiries. This trade-off in the choice of N is experimentally investigated in Sect. V.

The other parameters required in our approach can be determined empirically by measuring the localization performance in preliminary experiments. We set $\beta = 0.25$ and $\sigma = 0.5$. Moreover, for $q_l \in [0, \vartheta]$ with $\vartheta = 10^{-3}$, we set $p(q_l)$ such that $\int_0^{\vartheta} p(q_l) dq_l = 0.8$, and for $q_l \in (\vartheta, 1]$, we set $p(q_l)$ such that $\int_{\vartheta}^1 p(q_l) dq_l = 0.2$.

N	2	4	6	8
Duration	0.77 ± 0.02	1.46 ± 0.02	1.99 ± 0.03	2.66 ± 0.05

B. Accelerated Likelihood Evaluation During Localization

The computation of $\hat{\mathbf{q}}(\mathbf{a})$ is time-consuming if the number of particles is large and if a lot of training snapshots were recorded. The latter is normally advantageous because the localization accuracy is increased [8]. The estimation is even time-consuming if training snapshots with a distance of more than 2σ from $\mathbf{a}$ are pruned and the reference positions of training snapshots are stored in a kd-tree. As a first step of research into more efficient data structures for storing snapshots, we decided to precompute a look-up table (LUT) of reference snapshots.

Since the discretization of reference positions and angles may introduce intolerable localization errors, depending on the bin size along x, y and the orientation θ, we studied the behavior when using LUTs of varying bin sizes in a series of preceding experiments. The precomputation of snapshots by means of a look-up table (LUT) yielded an acceleration factor of approx. 40, i.e., per particle, the computation of its likelihood was forty times faster. We observed that the localization accuracy is highly sensitive to the angular resolution of the look-up table. On the whole, we drew the conclusion that a LUT is a reasonable means of speed-up without too large additional localization error for x/y resolutions of up to $0.2\,\mathrm{m}$ and angular resolutions of up to $10°$. Doubtlessly, precomputing the LUT is memory-demanding, depending on the resolution of the table and the number k' of tags in likelihood evaluations. (The latter is because precomputed snapshots can be pruned to size k'.) But for the finally chosen resolution, the setting $k' = 50$ and the larger one of our experimental environments, the corridor, the size of the LUT in RAM was $90.55\,\mathrm{MB}$, which we regard as unproblematic.

C. Initialization of the Particle Filter

By incorporating different RFID measurements and odometry in the particle filter, the pose estimate usually improves significantly over time. A central question, however, is how fast a decent estimate is achieved from scratch. The naïve filter initialization strategy is to distribute the samples uniformly to then apply the observation model. The success of this strategy depends on whether particles are placed near the true position of the robot by chance. Therefore, one possible extension to this approach is to boost the number of samples in the early stages of pose estimation.

The snapshot-based solution to the initialization issue, by contrast, is to initialize the particles with poses that are directly sampled from the training snapshots as soon as the first RFID measurements $\mathbf{f}$ arrive. Formally, the i-th particle

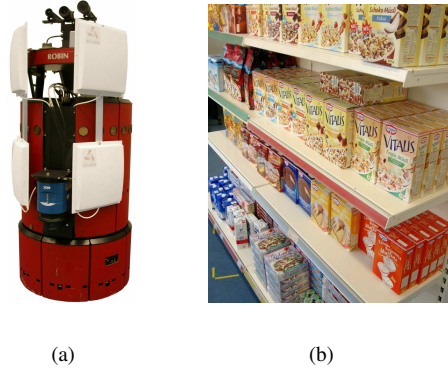(a)   (b)

Fig. 1.   (a) The B21 service robot with which we gathered the experimental data. White: the RFID antennas, connected to the RFID reader inside the robot. Blue: the front-mounted laser scanner, used for reference localization. (b) The supermarket shelf containing tagged product packages.

$(\mathbf{r}_i, w_i)$ is initially drawn randomly according to

$$\mathbf{r}_i \sim p(\mathbf{f} \,|\, \mathbf{a}(\mathbf{r}_i)) =: w_i,$$

where again the $\mathbf{a}(\mathbf{r}_i)$ assume the reference antenna positions of the training snapshots. The particle initialization includes the addition of a small portion of Gaussian noise. Again, we compared the performance of the different strategies and present the results in the next section.

V. EXPERIMENTS

We conducted several series of experiments on a B21 service robot (see Fig. 1(a)) in order to measure the effectiveness and accuracy of our approach. The robot is equipped with a UHF RFID reader by Alien Technology (type ALR-8780, 866 MHz) with two pairs of antennas. The two antennas of each pair work cooperatively, one antenna transmits energy and data while the other one is listening for potential responses by RFID tags. The antenna pairs span an angle of approx. $90°$, which enables them to scan for RFID tags in front and sideways of the robot. Reference positions are obtained with Monte Carlo localization based on a laser scanner ($240°$ field of view). The localization error of the reference poses can be assumed to be sufficiently small, below $0.1\,\mathrm{m}$.

The environment in which we recorded the experimental data is illustrated in Fig. 2. It is comprised of a laboratory ($50\,\mathrm{m}^2$ of accessible area) and a corridor ($75\,\mathrm{m}^2$). In both areas, we installed passive RFID tags of the standard EPC Class 1 Generation 2. For the lower-density experiments, tags were attached to walls and desks at distances of approx. $1\,\mathrm{m}$ on average and roughly at the height of the upper RFID antennas. For the higher-density experiments, we attached them to walls at a density of four per meter and at about the same height as before. In the center of the laboratory we placed a metal shelf ($2\,\mathrm{m} \times 1.2\,\mathrm{m}$) with almost 400 empty, individually RFID-tagged product packages at different heights in order to imitate a supermarket environment (see Fig. 1(b)). The width of robot-navigable passages (without transponders,
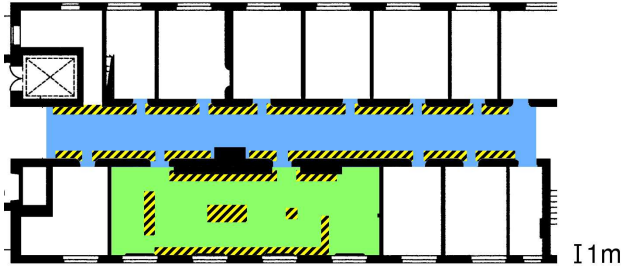
Fig. 2. The environment where we conducted the experiments: a laboratory (green/light gray) and the adjacent corridor (blue/dark gray). The RFID tags were attached to walls, desks and shelves in the hatched areas at a height of approx. 1 m.
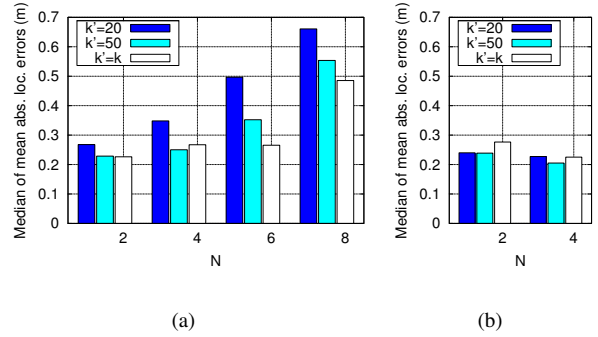

(a)   (b)

Fig. 3. Localization results (a) in the lab and (b) in the corridor environment for different values of k' and the number of measurement cycles, N.

e.g., between the shelf and the walls) was at least 1.2 m, usually 1.8 m or more, and at least 2.2 m on the corridor.

In total, we recorded 8 data sets, varying the density of tags as well as the number of cycles N. All data sets consist of ten log files. Each log file contains a robot trajectory of 89-188 m length and a duration of at least five minutes. For the corridor, we recorded four data sets, with $N \in \{2, 4\}$ and approx. 140/210 tags (where about 100 could be detected from the adjacent lab). In the lab, four data sets with the same high tag density (374 tags) were logged, but we took one data set for each $N \in \{2, 4, 6, 8\}$. In each single localization run of the subsequently depicted setups, the robot was trained with snapshots from a subset of those log files. Then, the robot had to localize on a remaining single validation trajectory which was not part of the training set, of course. The particle filter in the validation runs consisted of 2000 samples in general.

A. Initialization (Global Localization)

We compared the three initialization strategies described in Sect. IV-C: (a) the naïve uniform distribution of samples, (b) a uniform distribution with boosted number of particles ($n = 10000$), and (c) the sampling of reference positions from training snapshots. We performed fivefold cross-validation with five repeated experiments per combination of training and validation data, where we chose log files with $N \in \{2, 4\}$ and set $k' = 50$. In each localization run, we measured the mean absolute localization error after 1 time step and after 10 time steps, i.e., 10 iterations of the particle filter.

The results are listed in Table II. The snapshot sampling approach performs best, although not significantly better than the boosting approach. It requires, however, fewer particles (2000) in the first steps to achieve the given accuracy.

B. Localization Performance

In order to measure the accuracy of snapshot-based self-localization, we conducted various experiments in which we varied the number of measurement cycles, N, and the parameter k'. We performed cross-validation by first training a map of snapshots on N log files and then using the remaining log files of the same N for validation. By this, at least 2000 training snapshots were provided. In total, we performed 400 experiments for each combination of k' and N and varying training data. In each experiment, we

measured the mean absolute localization error, averaged over all time steps of the given validation log file. Note that all experiments include global localization, i.e., the initial pose was unknown and the particle filter was initialized via the snapshot sampling approach described above. Based on the insights from Sect. IV-B, we used a LUT with an x/y resolution of 0.1 m and an angular resolution of $10°$.

On the corridor, we tested two different numbers of cycles ($N \in \{2, 4\}$) and two different densities of tags (141/209 tags). $k' = 50$ was fixed. Table III lists the medians over the mean absolute localization errors of the single localization runs in the corridor. For $N = 4$ and the higher tag density, we achieved an accuracy of 0.2051 m.

In another series of experiments, we fixed the higher tag density and measured the influence of the choice of k'. The results are illustrated in Fig. 3(a) and 3(b), and listed in detail in Tab. IV. In both experiment environments, we achieved the best results for $N \in \{2, 4\}$ and $k' = 50$. Unfortunately, we observed a few cases in which the robot was delocalized, which results in the high standard deviations in Tab. IV.

TABLE II

POSITIONING ERRORS (IN METERS) DURING INITIALIZATION

Method	N	After time step 1		After time step 10	
		Mean	Median	Mean	Median
Uniform distribution		0.950	0.882	0.494	0.432
Uniform + boosting	2	0.859	0.746	0.464	0.334
Snapshot sampling		0.838	0.543	0.427	0.334
Uniform distribution		0.524	0.442	0.400	0.381
Uniform + boosting	4	0.477	0.380	0.329	0.298
Snapshot sampling		0.461	0.359	0.323	0.290

TABLE III

LOCALIZATION RESULTS IN THE CORRIDOR ENVIRONMENT: QUARTILES OF THE MEAN ABSOLUTE LOCALIZATION ERRORS (IN METERS)

N	Tag density	Median	25th percentile	75th percentile
2	Higher	0.239	0.215	0.256
2	Lower	0.263	0.244	0.288
4	Higher	0.205	0.189	0.218
4	Lower	0.260	0.232	0.323

A closer look at the log files revealed, however, that large initialization errors had been the cause. In these cases the particle filter was not able to converge later. Despite the (rare) failures, the mean absolute localization errors of approx. 90% of the localization runs were below $0.3\,\text{m}$.

From the outcomes of the experiments, we draw two conclusions: First, the localization accuracy is better for smaller values of N, i.e., fewer measurement cycles incorporated in a single snapshot. The reason for this is that RFID measurements arrive too rarely for larger N. Second, while it seems reasonable to set $k = k'$ for $N \geq 6$, the partial evaluation of the likelihood function by incorporating only $k' < k \approx 370$ tags appears to yield better results for smaller (and the more relevant) $N \in \{2, 4\}$. The outcomes for $k' = 20$ show, however, that k' must not be chosen too small. The abovementioned position losses mainly occurred for $k' = k$, which we regard as evidence of our assumptions made in Sec. III-C. We experimented also with other values of k', but did not optimize it systematically, because our suggestion is to first choose the other parameters and finally fine-tune k', depending on the environment.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper we have shown that it is possible to localize a mobile robot quite accurately via passive UHF RFID. To this end, we employ RFID snapshots, a probabilistic RFID fingerprinting technique embedded in a particle filter. In this work, we focused on densely tagged environments, because one can expect such tag arrangements in industrial, logistics, and warehousing scenarios in the near future.

In our indoor experiments with a mobile robot and an off-the-shelf RFID reader, we achieved accuracies of $0.25\,\text{m}$ or better in both environments. To the best of our knowledge, these are currently the most accurate results achieved in such a UHF RFID setup. The advantage of snapshot-based localization is that, despite the adverse properties of RFID, quite accurate localization is achieved without an explicit sensor model, which can be cumbersome to build. While the accuracies of the well-known laser-based approaches are not achieved, RFID represents a cost-neutral technology for positioning if robots already use RFID for inventory purposes in warehousing and logistics. Note that a specific distribution of the tags is not required, e.g., integration into the floor as required by other approaches. Of course, RFID can still also be fused with other sensors to improve positioning accuracy.

The requirement to record training snapshots before the localization phase is the disadvantage of our method. One could argue that any different approach will also require some mapping phase in order to learn the positions of RFID tags. But the problem remains that reference positions are needed when fingerprints are recorded during the training.

B. Future Works

For the future, we plan to address the elimination of the reference localization system required in the training phase. Moreover, we are going to allow for the revision of the learned snapshot distribution, since tagged objects can usually be removed or relocated. Finally, we are going to test our approach in larger-scale environments with comparable densities of RFID tags as in the scenario investigated here.

REFERENCES

[1] P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proc. of the IEEE Infocom 2000*, volume 2, pages 775–784, March 2000.

[2] J. Bohn. Prototypical Implementation of Location-Aware Services based on Super-Distributed RFID Tags. In *Proc. of the 19th International Conference on Architecture of Computing Systems (ARCS 2006)*, number 3894 in LNCS, pages 69–83. Springer-Verlag, 2006.

[3] H. Chae and K. Han. Combination of RFID and Vision for Mobile Robot Localization. In *Proc. of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 75–80, 2005.

[4] J. Djugash, S. Singh, and P. Corke. Further results with localization and mapping using range from radio. In *Field and Service Robotics (FSR '05)*, volume 25 of *Springer Tracts in Advanced Robotics*, pages 231–242. Springer, July 2005.

[5] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and Localization with RFID Technology. In *Proc. of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 1015–1020, 2004.

[6] A. Kleiner, C. Dornhege, and S. Dali. Mapping disaster areas jointly: RFID-Coordinated SLAM by Humans and Robots. In *Proc. of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, Rome, Italy, 2007.

[7] A. Lim and K. Zhang. *Advances in Applied Artificial Intelligence*, volume 4031 of *Lecture Notes in Computer Science*, chapter A Robust RFID-Based Method for Precise Indoor Positioning, pages 1189–1199. Springer Berlin / Heidelberg, 2006.

[8] S. Schneegans, P. Vorst, and A. Zell. Using RFID Snapshots for Mobile Robot Self-Localization. In *Proc. of the 3rd European Conference on Mobile Robots (ECMR 2007)*, pages 241–246, Freiburg, Germany, September 2007.

[9] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[10] P. Vorst and A. Zell. *European Robotics Symposium 2008*, volume 44/2008 of *Springer Tracts in Advanced Robotics*, chapter Semi-Autonomous Learning of an RFID Sensor Model for Mobile Robot Self-localization, pages 273–282. Springer Berlin / Heidelberg, 2008.

[11] 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. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 4, pages 3756–3761, September/October 2004.

TABLE IV

LOCALIZATION RESULTS (IN METERS) IN THE LAB

N	k'	Mean $\pm$ Std.dev.	Median	90th percentile
2	20	0.283 ± 0.075	0.268	0.355
	50	0.238 ± 0.054	0.229	0.304
	k	0.283 ± 0.388	0.226	0.306
4	20	0.380 ± 0.141	0.348	0.511
	50	0.259 ± 0.040	0.250	0.316
	k	0.264 ± 0.047	0.267	0.322
6	20	0.529 ± 0.307	0.497	0.668
	50	0.364 ± 0.065	0.352	0.450
	k	0.282 ± 0.064	0.266	0.365
8	20	0.813 ± 0.588	0.661	1.190
	50	0.602 ± 0.232	0.554	0.824
	k	0.565 ± 0.249	0.485	0.899