

Fully Autonomous Trajectory Estimation with Long-Range Passive RFID

Philipp Vorst and Andreas Zell

Abstract—We present a novel approach which enables a mobile robot to estimate its trajectory in an unknown environment with long-range passive radio-frequency identification (RFID). The estimation is based only on odometry and RFID measurements. The technique requires no prior observation model and makes no assumptions on the RFID setup. In particular, it is adaptive to the power level, the way the RFID antennas are mounted on the robot, and environmental characteristics, which have major impact on long-range RFID measurements. Tag positions need not be known in advance, and only the arbitrary, given infrastructure of RFID tags in the environment is utilized. By a series of experiments with a mobile robot, we show that trajectory estimation is achieved accurately and robustly.

I. INTRODUCTION

Radio-frequency identification (RFID) is a technology that allows for the contactless identification of goods. Its growing use in economy makes it attractive also for robotics applications. Whenever a mobile robot is already equipped with an on-board RFID reader, it can cost-efficiently exploit remote RFID transponders (also called tags) as uniquely identifiable landmarks for navigation. Long-range passive RFID (860-915 MHz), the prevailing technology in industry with a read range of up to 10 m, suffers from frequent nondetections of tags in read range. Moreover, passive RFID almost entirely lacks distance and bearing information between RFID reader and transponders, and measurements are influenced by materials such as water and metal surrounding the tags.

In this paper, we present a novel method for trajectory estimation that actually exploits the high degree of location-specificity of those sometimes undesirable characteristics. In densely tagged environments with tags on walls and in shelves (supermarkets, for example), each single RFID measurement can contain enough information to roughly estimate the pose of the robot. This is although tags are only detected, but not localized during trajectory estimation. Here, we even treat the case that the tags are located in unknown positions, making our approach suitable for autonomous mapping without prior map.

In brief, our technique works as follows: We observe RFID measurements and odometry readings while the robot is exploring an unknown environment. The sensor readings are used to first derive an observation model for the given hardware configuration and the explored environment. This

This work was funded by the Landesstiftung Baden-Württemberg in the scope of the research program BW-FIT, research cooperation AmbiSense.

P. Vorst is with the Chair of Computer Architecture headed by Prof. A. Zell, Computer Science Department, University of Tübingen, Sand 1, D-72076 Tübingen, Germany. {philipp.vorst, andreas.zell}@uni-tuebingen.de

model is generated by computing similarity features of combined RFID measurements, followed by training a classifier. Second, the trajectory of the robot is estimated in a batch-processing fashion. This step involves the recognition of places for closing loops and a maximum-likelihood estimation step, based on odometry and derived loop-closure constraints. Although the presented solution operates offline and estimates the trajectory of the robot afterwards, its design potentially allows for on-line operation.

This work is structured as follows: In Sect. II, we give a review of related work. Thereafter, graph-based SLAM methods and RFID fingerprinting are described in Sect. III. In Sect. IV, we present our approach in detail, followed by experimental results in Sect. V. Finally, we summarize and draw conclusions in Sect. VI.

II. RELATED WORK

In order to correct odometric errors in the estimated trajectory of a robot, previously visited places have to be recognized. This issue is known as loop closing. It has been studied intensively for robots with laser range finders (e.g., by 2D or 3D scan matching [1], [2], [3] or closing large loops on a higher level between local metrical maps [4]) and cameras (e.g., [5], [6]).

An RFID-based work close to ours is the one by Kleiner *et al.* [7]: They also performed graph-based SLAM, but with short-range RFID tags on the ground and a sensor model that was given beforehand. Djughash *et al.* [8] used an extended Kalman filter for SLAM with active radio beacons. Such active RFID tags provide signal strength information,

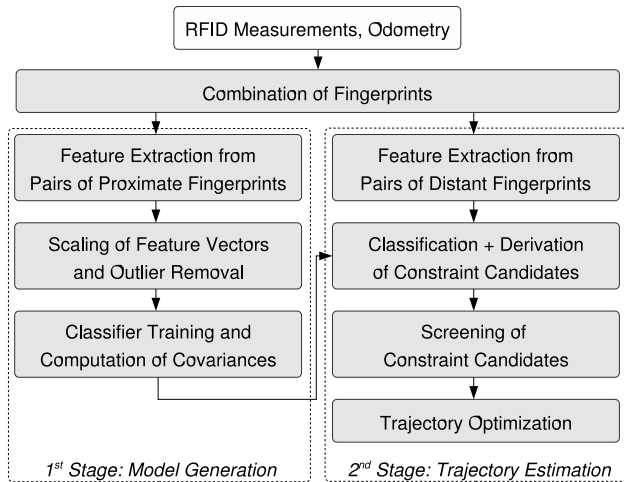


Fig. 1. Overview of the steps to estimate the trajectory of the robot.

which allows for distance estimation, contrary to passive RFID tags of the standard used in our work. Hähnel *et al.* used a similar technical setup as ours and mapped the positions of long-range passive RFID tags with a particle filter [9]. Their probabilistic sensor model was learned from a series of empirical measurements. The reference path of the robot was computed with a laser-based FastSLAM approach. Our approach differs in that we neither require a sensor model beforehand nor reference localization. We make use of fingerprints, i.e., measurements characterizing their recording positions. This choice is predicated on studies on successful self-localization with Bluetooth and RFID fingerprints [10], [11], [12], [13]. All these approaches require potentially time-consuming prior training. Bolliger *et al.* suggested to supersede exhaustive prior training with the help of user interactions [14]: In their system, users can add new GSM/Bluetooth/WiFi fingerprints to a central database whenever a location is not yet known. The work at hand also aims at overcoming prior training, but does not require any user interaction.

In [15], we also presented an approach to trajectory estimation. Loop closure was achieved by different similarity measures for RFID fingerprints, but required the acquisition of loop closure models by prior training. The work at hand further extends the previous work significantly insofar as the new approach automatically learns the current RFID configuration in the target environment. Moreover, there are only few parameters to be predefined by the user.

III. GRAPH-BASED SLAM, LOOP CLOSURE, AND RFID FINGERPRINTS

Trajectory estimation is the central stage in recent graph-based approaches to simultaneous localization and mapping (SLAM): The reconstruction of the path of the robot is decoupled from map building: Observed features can be mapped consistently once that the past positions of the robot are known. Graph-based SLAM aims at finding the maximum-likelihood configuration of robot positions by nonlinear sparse optimization. A graph is constructed whose nodes represent robot locations, and edges are given by geometric constraints between the nodes. Constraints result from measured displacements between consecutive robot poses (odometry), or from observed displacements between arbitrary robot locations. These loop-closure constraints are derived from observations which have been associated with each other. Loop closure is important because it enables the robot to correct odometric errors that have accumulated along its path. Since odometry and sensor readings are noisy, each constraint is additionally given an error estimate. Solving graph-based SLAM consequently turns into finding a configuration of node locations which minimizes the overall error of the network. The original solution dates back to Lu and Milios [2]. Recent far more efficient algorithms employ fast gradient descent techniques with advantageous state space parameterizations [16], [17] or explicitly utilize the sparse nature of the constraint graph [18].



Fig. 2. Left: The mobile robot (RWI B21) used for the experiments, with its UHF RFID antennas (white), spanning an angle of approx. 90° . Right: One of the environments in which we conducted the experiments. Approx. 400 RFID tags had been placed in the hatched areas. The second test area is very similar in size and shape.

Formally, let $\mathbf{X}_{0:T} = (\mathbf{x}_0, \dots, \mathbf{x}_T)$ denote the path of the robot to be estimated up to time step T , with $\mathbf{x}_t = (x_t, y_t, \theta_t)$ the global 2D coordinates and heading of the robot at time t . Let \mathbf{X} comprise the poses $\mathbf{X}_{0:T}$ of the robot. In graph-based SLAM, all sensor readings are expressed as constraints in the shape of rigid-body transformations between two nodes i and j with expected values $\delta_{ij} = (\Delta x_{ij}, \Delta y_{ij}, \Delta \theta_{ij})$ (2D translation plus rotation) and associated error covariance matrices Σ_{ij} (cf. [16], [17]). Let $f_{ij}(\mathbf{X})$ generate zero-noise observations according to the current configuration of the nodes i and j in the pose graph. Then, computing the SLAM solution equals the iterative minimization of¹

$$\sum_{(i,j) \in \mathcal{X} \times \mathcal{X}} (f_{ij}(\mathbf{X}) - \delta_{ij})^T \Sigma_{ij}^{-1} (f_{ij}(\mathbf{X}) - \delta_{ij}) \quad (1)$$

With RFID the detection of re-entering the same *area* is easy, since measured tag IDs are unique. The estimation of a metric transformation between two positions, however, is tricky: Due to a read range of up to 10m the position uncertainty of RFID measurements can be even larger than accumulated odometric errors. The solution is to recognize similar *positions* at a fine resolution and to assume that they are actually the same. Now, the chance of detecting a single RFID tag largely depends on its distance and relative orientation to the antenna of the RFID reader. Tag detections are further dependant on a number of physical phenomena such as absorption by nearby objects and multi-path effects. From a practical point of view, it is impossible to model such factors. Models of RFID detection rates usually accept those factors as noise, which can result in less accurate navigation performance. Quite the contrary, in our approach, we make use of all influencing factors, but do not model them explicitly: We regard local RFID measurements as *fingerprints* of their recording position. These fingerprints are highly location-specific [19]. They can be compared to visual features in appearance-based methods used for SLAM or self-localization.

¹with $\Sigma_{ij}^{-1} := \mathbf{0}$ if there is no constraint between the nodes i and j

When comparing two non-consecutive fingerprints, the degree of similarity can be used to detect that a position is being visited again. With regard to Sect. III, a constraint edge will be added to the pose graph in such a case. The next section will treat the issues of how to recognize a position of the robot that has been assumed earlier.

IV. APPROACH

We achieve trajectory estimation by two passes over the recorded sensor data. The organization of the single steps is illustrated in Fig. 1. Initially, RFID measurements are preprocessed by combining sequenced fingerprints over short distances. Then, we derive an observation model which learns to classify closed loops. Since the classifier is trained on the target data, it adapts to the current RFID setup and the traversed environment. In the second pass over the recorded data, the derived model is used to detect closed loops, and finally the trajectory of the robot can be estimated by the described graph optimization.

A. Combination of Fingerprints

Measurements of long-range passive RFID only contain the information how often which tag has been detected. The data are noisy insofar as even in the very same position, inquiries may come to quite different results [19]. That is why we combine successive fingerprints which are sufficiently close to each other. Formally, we represent a fingerprint \mathbf{F}_t at time t by the tuple $\mathbf{F}_t = (\mathbf{f}_t, \mathbf{x}_t, d_t^\Sigma, r_t^\Sigma)$. Let A be the number of antennas connected to the RFID reader on the robot. Then, the measurement $\mathbf{f}_t = (\mathbf{f}_{t,1}, \dots, \mathbf{f}_{t,A})$ consists of A single lists of detected tags, with $\mathbf{f}_{t,a} = (f_{t,a}^{(1)}, \dots, f_{t,a}^{(K)})$, $a = 1, \dots, A$, a vector of integers. In our case, the robot possesses two pairs of antennas (Fig. 2), so $A = 4$. $f_{t,a}^{(k)}$ counts how often antenna a has detected tag k . K is the total number of tags that are observable in the environment. $\mathbf{x}_t = (x_t, y_t, \theta_t)$ is the odometric recording position and heading of the robot. d_t^Σ and r_t^Σ are the accumulated absolute translation and rotation, respectively, from \mathbf{x}_0 up to time t .

Now, a combined fingerprint $\mathbf{G}_t = (\mathbf{g}_t, \mathbf{x}_t, d_t^\Sigma, r_t^\Sigma)$ contains the same robot pose \mathbf{x}_t and accumulated values d_t^Σ and r_t^Σ as \mathbf{F}_t , but a new, real-valued vector $\mathbf{g}_t := \sum_{j=0}^t w_j \mathbf{f}_j$ of averaged detection frequencies with

$$w_j = \begin{cases} \exp(-\frac{1}{2}d(\mathbf{F}_t, \mathbf{F}_j)), & (d_t^\Sigma - d_j^\Sigma \leq \vartheta_d) \\ & \wedge (r_t^\Sigma - r_j^\Sigma \leq \vartheta_r) \\ 0 & \text{else} \end{cases} \quad (2)$$

So, if fingerprint j is close enough to fingerprint t in that the differences in accumulated odometric forward translation and rotation are below the thresholds ϑ_d and ϑ_r , respectively, \mathbf{f}_j will receive non-zero weight in the combined fingerprint. The distance measure $d(\mathbf{F}_t, \mathbf{F}_j)$ must take translations and rotations into account. We set

$$d(\mathbf{F}_t, \mathbf{F}_j) = \frac{(d_t^\Sigma - d_j^\Sigma)^2}{\vartheta_d^2} + \frac{(r_t^\Sigma - r_j^\Sigma)^2}{\vartheta_r^2} \quad (3)$$

with $\vartheta_d = 1.0$ and $\vartheta_r = 30^\circ$.

B. Observation Model Generation

The model generation phase comprises three substages: First, features are extracted which describe the similarity of pairs of combined fingerprints. Then, the resulting feature vectors are scaled appropriately, and training data with extremal values are removed. Finally, the classifier is trained.

1) *Feature Extraction from Pairs of Proximate Fingerprints*: One key insight in our approach is that – although odometric errors accumulate in the long term and make the mapping of environments difficult – odometry is sufficiently accurate over short distances. Hence, we compare pairs of fingerprints which were recorded not too far away from each other. In the work at hand, we assume that at distances of less than 10m, odometry is sufficiently accurate. This threshold should be adapted to the employed robot and ground surface, but we think that 10m is a decent value for most indoor platforms. For each such pair $(\mathbf{G}_i, \mathbf{G}_j)$ of fingerprints, we compute a feature vector $\mathbf{m}(\mathbf{g}_i, \mathbf{g}_j)$. Its columns describe the *similarity* of the combined RFID measurements \mathbf{g}_i and \mathbf{g}_j . Similarities are computed for each of the A antennas independently, and the features are concatenated:

$$\mathbf{m}(\mathbf{g}_i, \mathbf{g}_j) = (\mathbf{m}_1(\mathbf{g}_i, \mathbf{g}_j), \dots, \mathbf{m}_A(\mathbf{g}_i, \mathbf{g}_j))$$

Each similarity $\mathbf{m}_a(\mathbf{g}_i, \mathbf{g}_j)$, with $a = 1, \dots, A$, should express how well the lists of detection frequencies of tags $(\mathbf{g}_{i,a}$ and $\mathbf{g}_{j,a})$ match for the two fingerprints at a given antenna a . In our implementation, we employ the *cosine similarity*², which represents the cosine of the angle spanned by the vectors $\mathbf{g}_{i,a}$ and $\mathbf{g}_{j,a}$. In order to enhance the expressiveness of the feature vector, we additionally store the logarithm of the number of tag IDs which appear in $\mathbf{g}_{i,a}$ or $\mathbf{g}_{j,a}$:

$$\mathbf{m}_a(\mathbf{g}_i, \mathbf{g}_j) = (\cos(\mathbf{g}_{i,a}, \mathbf{g}_{j,a}), \log(|\mathbf{g}_{i,a} \cup \mathbf{g}_{j,a}| + 1)) \quad (4)$$

with

$$\cos(\mathbf{g}_{i,a}, \mathbf{g}_{j,a}) = \frac{\sum_{k=1}^K g_{i,a}^{(k)} g_{j,a}^{(k)}}{\sqrt{\sum_{k=1}^K (g_{i,a}^{(k)})^2} \cdot \sqrt{\sum_{k=1}^K (g_{j,a}^{(k)})^2}} \quad (5)$$

If $\mathbf{g}_{i,a} = \mathbf{g}_{j,a} = \emptyset$, we set $\cos(\mathbf{g}_{i,a}, \mathbf{g}_{j,a}) = 0$. The reason for also storing the number of unified tag IDs is that the cosine similarity may be large if in two RFID measurements only few tags were detected and in common. The enclosing logarithm is supposed to prevent that larger values of tag numbers dominate the corresponding feature column.

So, for each compared pair of fingerprints, we store a feature vector with $2 \times A$ dimensions. For the computation of covariances (Sect. IV-B.3), we further remember the relative distances Δd and angles $\Delta \theta$ between the recording positions of the fingerprints. Finally, we assign one of two class labels $y \in \{\text{LC}^+, \text{LC}^-\}$ to each training vector, according to the following simple rule:

$$y = \begin{cases} \text{LC}^+ & |\Delta d| < 0.5 \wedge |\Delta \theta| \leq 30^\circ \\ \text{LC}^- & \text{else} \end{cases} \quad (6)$$

²We tested the normalized histogram intersection and the squared distance to compare two fingerprints, too, but the cosine similarity performed best.

Class LC^+ contains feature vectors from which loop closure can be inferred: The recording positions of the source fingerprints are located in distances and angles of below 0.5 m and 30° , respectively. These small bounds describe the situation that the robot has visited the same place again. In opposition, LC^- contains feature vectors of compared fingerprints with distant recording positions. Hence, samples in LC^- describe situations from which loop closure should *not* be inferred.

2) *Scaling of Feature Vectors and Outlier Removal:* The features computed in the previous section have dimensions whose ranges differ. During classification, some dimensions might thus dominate others. That is why we scale the feature vectors such that the values of each dimension are mapped to the range $[-1; 1]$, as common in many classification tasks.

After scaling, feature vectors assigned to class LC^+ are removed if at least one of its entries equals -1 . This either means that an antenna has not detected any tag in any of the two fingerprints, or that the tag lists of two fingerprints are disjoint. Classification is more reliable if such obviously undesirable training samples are filtered out.

3) *Classifier Training and Computation of Covariances:* Given the generated training data, a classifier can be trained whose task it is to recognize a previously visited place. Our approach does not depend on a specific type of classifier. In our implementation, we employ k-nearest neighbor (k-NN) classification. Training is fast, since it only requires to store new training samples (lazy learning). Furthermore, classification can be implemented efficiently. Consequently, the classifier could also be used for *on-line* trajectory estimation.

In k-NN classification, a query feature vector \mathbf{q} is classified by retrieving those k training samples $\mathbf{m}^{(i_1)}, \dots, \mathbf{m}^{(i_k)}$ which yield the smallest distance $\|\mathbf{q} - \mathbf{m}^{(i_j)}\|$ among the training data. In our implementation, we set $k = 16$. Let $y(\mathbf{m}^{(i_j)})$ denote the class labels of the retrieved training vectors. The predicted class $\hat{y}(\mathbf{q})$ will be the one with which the majority of the retrieved training vectors are labeled:

$$\hat{y}(\mathbf{q}) = \underset{c}{\operatorname{argmax}} p(y(\mathbf{q}) = c) \quad (7)$$

$$p(y(\mathbf{q}) = c) = |\{i_j \mid y(\mathbf{m}^{(i_j)}) = c, j = 1, \dots, k\}|/k \quad (8)$$

The posterior probability $p(y(\mathbf{q}) = c)$ also estimates the degree of certainty of the classification result.

We finally have to compute covariance estimates which describe the uncertainty of a constraint in the pose graph. That is why we group training features from the loop-closure class, LC^+ , by their posterior probabilities, obtained by feeding them to the k-NN classifier. For each group, we store in a table the means (which are theoretically and practically close to zero) and covariances of the stored relative positions and relative orientations between the compared fingerprints which produced the feature vectors.

C. Trajectory Estimation

In the first pass over the sensor data, a loop-closure model was trained and parameterized that adapts to the target data. The derived model can now be used to detect if fingerprints taken at distant time steps indicate that the robot has assumed

a pose similar to an earlier one. In this stage, we proceed as follows: We construct the pose graph, whose nodes represent positions for which RFID measurements are available. Their odometric positions serve as initial estimates of the poses along the trajectory. Consecutive nodes are connected by edges whose rigid-body transformations $(\Delta x, \Delta y, \Delta \theta)$ correspond to odometry. The associated uncertainties are estimated by the motion model of the employed robot. Then, loop-closure constraints are added to the pose graph.

1) *Feature Extraction from Pairs of Distant Fingerprints:* In order to test for closed loops, feature vectors are computed for pairs $(\mathbf{G}_i, \mathbf{G}_j)$ of fingerprints with a large difference in absolute distance traveled, that is, $|d_i^\Sigma - d_j^\Sigma| \geq 20.0 m$. For pairs which satisfy this criterion, feature vectors are computed and scaled in the same fashion as above.

2) *Classification and Derivation of Constraint Candidates:* The feature vectors from the previous step are classified, using the trained classifier of Sect. IV-B.3. If $\hat{y}(\mathbf{m}(\mathbf{g}_i, \mathbf{g}_j)) = LC^+$ for a feature vector $\mathbf{m}(\mathbf{g}_i, \mathbf{g}_j)$, the robot may have been in similar positions in time steps i and j . Let $p_{ij} = p(y(\mathbf{m}(\mathbf{g}_i, \mathbf{g}_j)) = LC^+)$ denote the posterior probability that the poses at times i and j were the same. We will add the tuple (i, j, p_{ij}) to a set of preliminary constraint *candidates* if p_{ij} exceeds some threshold ϑ_p . ϑ_p is one of the few user-supplied parameters in our approach. As our experimental results show, the choice of ϑ_p is not too crucial. $\vartheta_p = 0.8$ should be a good value in most scenarios. Compared fingerprints with posterior probabilities below ϑ_p will be rejected, although they were classified as being similar, in order to avoid false-positive loop closures.

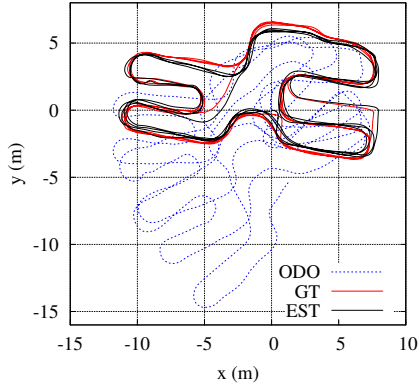
3) *Screening of Constraint Candidates:* False loop closures endanger the consistency of maps and trajectories obtained by SLAM algorithms. Many SLAM solutions therefore employ checks for the joint compatibility of observations (e.g., [6]). In some respects analogously, we thin the set of constraint candidates to yield robustness to outliers: For each constraint candidate (i, j, p_{ij}) we check if there is another candidate (l, m, p_{lm}) in a window of ϑ_w meters before or after i along the robot's path. If yes, the candidate with the lower posterior probability will be removed. In case of equality, the constraint will be removed whose fingerprint of the outgoing node (i or l) contains fewer tag IDs.

After screening, edges will be added to the pose graph for all remaining constraint candidates. The rigid-body transformation between pairs of nodes is parameterized by an expected value of 0 for both translation along x and y , and rotation. The covariance estimate of each constraint, depending on the posterior probability provided by the classifier, can be looked up in the table of model covariances (Sect. IV-B.3).

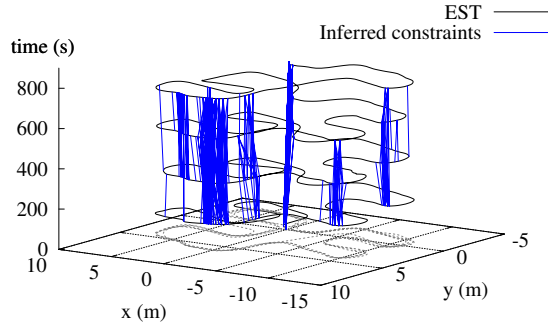
4) *Trajectory Optimization:* The trajectory can finally be optimized, using a graph optimization technique. Any framework that relies on rigid-body transformations with uncertainty estimates can be chosen, e.g., [16], [17], [18].

V. EXPERIMENTAL RESULTS

We conducted several experiments with the RWI B21 robot depicted in Fig. 2. The robot possesses a synchronous



(a)



(b)

Fig. 3. (a) A sample trajectory (labeled “EST”) obtained at full power level and with a posterior threshold of $\vartheta_p = 0.8$. “GT” (ground truth) is the actual path of the robot. “ODO” denotes odometry. In this case, a residual Cartesian error of 0.376 m was achieved, based on 3 selected loop-closure constraints that remained after the screening stage. 34,646 training samples contributed to the observation model. (b) The estimated trajectory over time (vertical axis) with the inferred constraints before screening.

drive and an Alien Technology ALR-8780 UHF RFID reader. The reader is connected to two pairs of antennas, which span an angle of approx. 90° . A 240° laser range finder provides ground truth positions.

The experiments were performed in two different environments, in each of which we distributed about 400 passive UHF RFID tags. Both environments have sizes of approx. 195 m^2 . They contain loops, and tags were spread in corridor-alike shapes (see Fig. 2), similarly to the arrangement of shelves in storehouses and supermarkets. The robot was steered manually through these environments while it was recording RFID measurements and odometry. Laser-based pose estimates were additionally logged as ground truth data. The resulting trajectories have different shapes and lengths of 68–295 m, corresponding to durations of 4–14 minutes and 313–1213 RFID measurements. Every RFID measurement detected 11.5 tags on average and at most 37 tags. Some paths contain several loops and significant overlaps, others only few potential loop-closure positions.

The different trajectories of the robot were estimated as described in Sect. IV. For graph optimization, we pursued the sparse-matrix approach by Takeuchi *et al.* [18]. Table I lists the estimation errors of the resulting trajectories depending on the posterior probability threshold, ϑ_p , for 13 trajectories

TABLE I
RESIDUAL CARTESIAN ERRORS (IN METERS) OF TRAJECTORY ESTIMATION AT FULL RFID TRANSMITTING POWER LEVEL (1 W AT EACH ANTENNA)

| Posterior Threshold (ϑ_p) | Mean \pm Std. Dev. | Median | Maximum |
|---------------------------------------|----------------------|--------|---------|
| 0.5 | 0.403 ± 0.194 | 0.350 | 0.917 |
| 0.6 | 0.419 ± 0.190 | 0.364 | 0.838 |
| 0.7 | 0.408 ± 0.173 | 0.350 | 0.840 |
| 0.8 | 0.397 ± 0.105 | 0.412 | 0.568 |
| 0.9 | 0.417 ± 0.166 | 0.436 | 0.661 |
| 1.0 | 0.436 ± 0.197 | 0.401 | 0.858 |

TABLE II
MEAN TRANSLATIONAL AND ROTATIONAL ERRORS AND MEAN NUMBERS OF LOOP-CLOSURE CONSTRAINTS

| Posterior Threshold (ϑ_p) | Distance error (m) Mean \pm Std. Dev. | Angular error (rad) Mean \pm Std. Dev. | Number of Constraints |
|---------------------------------------|---|--|-----------------------|
| 0.5 | 0.700 ± 0.352 | 0.258 ± 0.140 | 37.8 ± 20.2 |
| 0.6 | 0.691 ± 0.365 | 0.253 ± 0.150 | 31.7 ± 18.1 |
| 0.7 | 0.626 ± 0.301 | 0.233 ± 0.152 | 24.2 ± 14.4 |
| 0.8 | 0.597 ± 0.285 | 0.212 ± 0.152 | 20.4 ± 13.0 |
| 0.9 | 0.551 ± 0.349 | 0.203 ± 0.142 | 11.7 ± 8.5 |
| 1.0 | 0.528 ± 0.378 | 0.182 ± 0.164 | 7.8 ± 6.0 |

on which RFID measurements were performed at full power level. We set $\vartheta_w = 1.0$, that is, for the optimization of the graph, the best constraints were selected within a one-meter window, as described in Sect. IV-C.3. Depending on the trajectory, 13,233 to 54,970 training feature vectors were obtained for classification. This shows that, although we trust odometry only in the short term, a lot of training data is available to learn the observation model on the fly.

For each trajectory, we computed the mean residual Cartesian error per node in the pose graph after aligning the estimated trajectory at the ground truth positions. We obtained a mean error of approx. 0.4 m in all cases, independent of the posterior threshold. (By comparison, the residual error for odometry only was $1.27 \text{ m} \pm 0.65 \text{ m}$, maximum 2.40 m.) As can be seen, however, both variance and maximum residual error are lowest for $\vartheta_p = 0.8$. The reason for this is that this value is (also intuitively) a good trade-off between a sufficient *number* of inferred loop-closure constraints on the one hand and the *accuracy* of inferred constraints on the other hand, which is indicated by a high posterior probability of the classification result.

This explanation is underlined by Table II. There, it is shown that the larger ϑ_p , the smaller the error of inferred constraints. The inferred constraints, for which we set both the expected distance and angle between two positions zero, introduce small errors (below 0.6 m and 12° for $\vartheta_p \geq 0.8$). In comparison with the reader’s read range of up to 10 m, the inaccuracies are rather negligible. They still permit to compensate for accumulated odometry errors, which are tackled in all SLAM-related applications. Yet, the trajectory has to be corrected continuously, and thus it is important to ensure a decent number of inferred edges. In this respect, purely RFID-based trajectory estimation does differ from

TABLE III
RESIDUAL CARTESIAN ERRORS (IN METERS) OF ESTIMATED
TRAJECTORIES AT DIFFERENT POWER LEVELS

| Power Level | Mean ± Std. Dev. | Median | Max. | ∅ number of tags |
|------------------|---------------------|--------|-------|---------------------|
| Full | 0.446 ± 0.091 | 0.415 | 0.568 | 11.5 |
| Reduced by 4 dB* | 0.296 ± 0.093 | 0.301 | 0.424 | 9.4 |
| Reduced by 8 dB* | 0.260 ± 0.157 | 0.226 | 0.491 | 4.8 |
| Reduced by 12 dB | 0.259 ± 0.115 | 0.256 | 0.467 | 3.2 |

* Sample results, obtained only on four test trajectories

SLAM based on highly accurate laser scanners.

On a subset of seven trajectories, we also reduced the transmission power level of the RFID reader to examine how well our method adapts to different hardware configurations. Tags can then be detected only at shorter ranges, which decreases the position uncertainty of a single tag detection, but fewer transponders are identified. Table III shows that the accuracy of estimated trajectories improves. On the test dataset, the estimation error was reduced significantly from approx. 0.45 m at full power to approx. 0.26 m at a power level that corresponds to a read range of 1-2 m.

The mean position errors in this work are larger than in other previous studies where we examined self-localization with RFID fingerprints and particle filters [12], [20]. There, we achieved mean absolute localization errors of approx. 0.25 m at full transmission power. The difference is that the density of provided reference fingerprints was larger, which considerably improves the positioning accuracy. The approach in the paper at hand, however, redundatizes both a training phase and a reference positioning system, which were still required in the previous works.

VI. CONCLUSION

In this paper we have presented a novel approach which interleaves the estimation of the trajectory of the robot with the learning of an RFID observation model. It is purely based on passive RFID and odometry and overcomes the prior semi-automatic calibration stages of previous related works. Our technique does not make any assumptions about the distribution of RFID tags in the environment. It rather learns characteristics of RFID measurements. The approach achieves a satisfactory accuracy of the estimated trajectory, at approx. 0.4 m. Several integrated mechanisms enable the robot to robustly detect revisited places. It is well-suited for environments with higher tag densities, that is, scenarios in which given tag infrastructures allow for several tag detections per inquiry on average. Moreover, surroundings with corridors are beneficial, because we rely on frequent revisiting of places with similar orientations of the robot. Although this appears to be a limitation at first glance, corridors are typical in the target applications involving RFID such as supermarkets.

The derived trajectory can be used to map the positions of RFID transponders in another pass over the sensor data, using RFID mapping approaches [9]. Alternatively, RFID measurements plus the corrected poses could serve as a reference for self-localization with RFID fingerprints [12], [20].

In our experiments, new measurements could be processed in real-time on a 3 GHz PC. Hence, the presented solution could be turned into an on-line approach. Future extensions will also take non-static, relocated tags into account.

REFERENCES

- [1] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999, pp. 318–325.
- [2] F. Lu and E. E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, October 1997.
- [3] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM – 3D mapping outdoor environments," *Journal of Field Robotics (JFR)*, vol. 24, no. 8-9, pp. 699–722, August/September 2007.
- [4] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas framework for scalable mapping," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1899–1906.
- [5] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *9th IEEE International Conf. on Computer Vision (ICCV)*, vol. 2, September 2003, pp. 1403–1410.
- [6] L. Clemente, A. Davison, I. Reid, J. Neira, and J. D. Tardós, "Mapping large loops with a single hand-held camera," in *Proc. of Robotics: Science and Systems III*, Atlanta, GA, USA, June 2007.
- [7] 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.
- [8] J. Djughash, S. Singh, and P. I. Corke, "Further results with localization and mapping using range from radio," in *Int. Conf. on Field and Service Robotics (FSR)*, ser. Springer Tracts in Advanced Robotics, vol. 25. Springer, July 2005, pp. 231–242.
- [9] D. Hähnel, W. Burgard, D. Fox, K. P. Fishkin, and M. Philipose, "Mapping and localization with RFID technology," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004, pp. 1015–1020.
- [10] M. S. Bargh and R. de Groote, "Indoor localization based on response rate of Bluetooth inquiries," in *Proc. of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*. ACM, 2008, pp. 49–54.
- [11] A. Lim and K. Zhang, *Advances in Applied Artificial Intelligence*, ser. LNCS. Springer, 2006, vol. 4031, ch. A Robust RFID-Based Method for Precise Indoor Positioning, pp. 1189–1199.
- [12] S. Schneegans, P. Vorst, and A. Zell, "Using RFID snapshots for mobile robot self-localization," in *Proc. 3rd European Conference on Mobile Robots (ECMR)*, Freiburg, Germany, 2007, pp. 241–246.
- [13] 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. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 4, 2004, pp. 3756–3761.
- [14] P. Bolliger, "Redpin - adaptive, zero-configuration indoor localization through user collaboration," in *Proc. of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*. ACM, 2008, pp. 55–60.
- [15] P. Vorst, B. Yang, and A. Zell, "Loop closure and trajectory estimation with long-range passive RFID in densely tagged environments," in *Proc. 14th Int. Conf. on Advanced Robotics (ICAR)*, Munich, 2009.
- [16] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proc. of Robotics: Science and Systems III*, Atlanta, GA, USA, June 2007.
- [17] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.
- [18] E. Takeuchi and T. Tsubouchi, "Multi sensor map building based on sparse linear equations solver," in *Proc. of the 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 2511–2518.
- [19] M. Buettner and D. Wetherall, "An empirical study of UHF RFID performance," in *Proc. of the 14th Annual Int. Conf. on Mobile Computing and Networking (MobiCom)*. ACM, 2008, pp. 223–234.
- [20] P. Vorst, S. Schneegans, B. Yang, and A. Zell, "Self-localization with RFID snapshots in densely tagged environments," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 1353–1358.