RFID-Based Hybrid Metric-Topological SLAM for GPS-denied Environments

Christian Forster[1], Deon Sabatta[2,3], Roland Siegwart[2], Davide Scaramuzza[1]

***Abstract*— In this work, we propose a novel RFID-based hybrid metric-topological Simultaneous Localization and Mapping (SLAM) algorithm which enables autonomous navigation in GPS-denied environments. A method based on the normalized-cut is proposed for online clustering of strongly connected Radio Frequency Identification (RFID) tags to form topological nodes. A particle filter together with a sensor model which characterizes the received signal strength (RSS) as well as the tag detection probability is used to create metric submaps for each topological node.**

The hybrid framework is highly scalable, simplifies path planning and promises precision and robustness. The algorithm requires only odometry and RFID measurements to localize the RFID tags with a relative accuracy of approximately 0.3 meters. The ideas presented here are supported by experimental results.

I. Introduction

Underground mining remains difficult, hazardous work. In South African hard-rock mines, trained personnel execute safety inspections of the hanging wall after each blast at great personal risk. In this aspect, an autonomous robot could assist in improving safety. The Mobile Intelligent Autonomous Systems Group (MIAS) at the Center for Scientific and Industrial Research (CSIR) in South Africa therefore took the initiative to develop the necessary algorithms to allow a platform to autonomously explore and map hazardous areas in a mine. This project is called the *Mining Safety Platform*.

Due to the absence of GPS, reliable magnetic readings and unique landmarks, autonomous mapping and localization is challenging in underground environments. To solve this problem, we propose the use of passive RFID tags to provide labelled landmarks that can be artificially introduced into the target environment. The advantages of RFID technology include robustness, the low cost of single tags (contrary to the cost of the antenna) and the fact that RFID tags return a unique ID upon interrogation which resolves the correspondence problem. Additionally, many systems supply a received *signal strength* (RSS) measurement which is highly observation-angle dependent but may still be used as a rough distance metric embedded in a filtering framework.

In brief, our technique works as follows: We use the RSS value of simultaneously observed RFID tags to estimate their relative proximity. The tags are then successively clustered into groups using a novel method based on the normalized cut. Each cluster then forms a topological node of the map. By generating disjoint metric submaps for each topological node, we obtain a hybrid metric-topological representation of the environment (see Fig. 1). The relative tag positions in the metric submaps are estimated by a particle filter using the RSS information combined with a sensor model. During mapping, the robot's position within a submap is estimated using odometry only. We allow this assumption since the size of a submap is bounded and we can trust odometry over short distances. We perform SLAM on the topological level, while on the metric level, we only perform mapping.

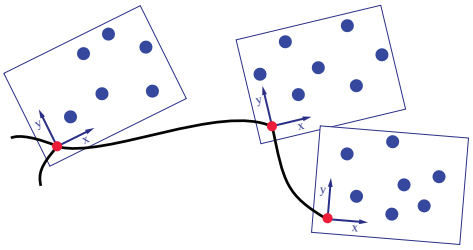


Fig. 1: A topological graph (red nodes and black edges) connects metric submaps of clustered RFID tags (blue) and thereby forms a hybrid metric-topological map.

The remainder of this paper is organized as follows: First, we give an overview of related work in Sect. II. Sect. III and Sect.IV describe the two main building blocks of the algorithm: the topological and metric mapping. These two parts are combined in Sect. V to form the hybrid algorithm. Finally, in Sect. VI, we show experimental results of the algorithm.

II. Related Work

The literature on RFID localization and mapping can be divided into three main groups depending on how the tags are used. The first group creates a sensor model to estimate the tag positions. Hähnel et al. [1] use the FastSLAM algorithm with laser data to create a map of the environment and to derive the position of the robot. A particle filter is then used to estimate the observed RFID tag positions. A sensor model which incorporates the tag-detection probability relative to the robot is used to reweight the particles. It is shown that given the RFID tag positions, the necessary particles for global localization are reduced from 10'000 to 50. Joho et al. [2] extended the idea by incorporating the RSS in the sensor model. Furthermore, the sensor model is learnt by a bootstrapping procedure. Using this method, a mean tag location error of approximately 29 cm is achieved. Using

[1]C. Forster and D. Scaramuzza are with the Artificial Intelligence Lab—Robotics and Perception Group, University of Zurich, Switzerland—http://rpg.ifi.uzh.ch

[2]D. Sabatta and R. Siegwart are with the Autonomous Systems Lab, ETH Zurich, Switzerland

[3]D. Sabatta is also with the Mobile Intelligent Autonomous Systems Group, Council for Scientific and Industrial Research, Pretoria, South Africa

This work was supported by funding from the Council for Scientific and Industrial Research (CSIR), South Africa

the same sensor model the robot can also locate itself within previously generated maps with a mean error of approximately 37 cm.

The second group does not try to locate the RFID tag positions but characterizes the sensed tag information at visited places in the environment. Vorst et al. [3] describe an approach where RFID fingerprints are compared to data obtained during a training phase. The robot can locate itself by finding the most similar fingerprint from the training data to the currently observed one. Seco [4] extends the idea of descriptors for particular locations and uses Gaussian processes to estimate the received signal strength between calibration points.

The third group uses RFID tags for robust detection of loop closures. Vorst and Zell [5] presented an offline graph SLAM algorithm to optimize the robot trajectory. RFID fingerprints are thereby used to improve loop closure detection. Lavigne et al. [6] use sparsely distributed RFID tags for reliably mapping large-scale underground tunnel networks with an occupancy grid. Kleiner et al. [7] investigated SLAM with sparsely distributed short range RFID tags to map disaster areas jointly with humans and robots. A graph of RFID tags is generated with the edge value set to the estimated distance between the corresponding RFID tags. The graph constructed by this distance information is globally optimized to form a consistent map using a maximum likelihood principle.

Topological and metric maps are two fundamentally different map representations. Metric maps use geometric information to describe the environment as precisely as possible whereas topological maps [8] concentrate on characteristics which are most relevant to the robot for localization. A topological graph specifies two things: *nodes* and the *connectivity* between nodes. Nodes specify areas of interest to the robot, e.g. a room, whereas an edge between two nodes in the graph indicates that the robot can traverse from one room to the other without visiting an intermediate room, e.g. a hallway. Hybrid methods [9], [10], [11] have been proposed to deal with large and complex robot environments. Hybrid maps consist of a global topological map with local metric submaps attached to each node. In this case, the anticipation of creating large scale consistent metric maps is dropped since the topological relationship between neighbouring maps is enough to successfully navigate between the "rooms" in the environment. Accurate path planning within each room however is still possible due to the local metric submaps. Raoui et. al [12] have previously demonstrated the usability of RFID-based hybrid metric and topological maps for self-localization of a trolley robot in a shopping center. However, both the topological areas and the tag positions were assumed to be known a priori.

III. Topological SLAM

Our topological nodes are specified by groups of RFID tags where each RFID tag can only be part of a single topological node. As the robot explores the environment and detects RFID tags, newly observed tags are either assigned to an existing topological node or form a new topological node

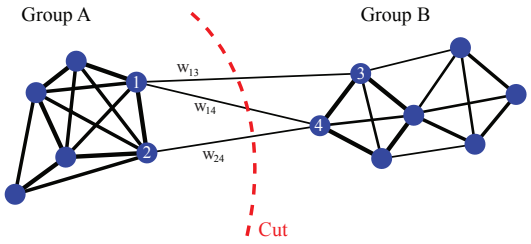


Fig. 2: Example cut of a graph

themselves. Possible approaches described in the literature to assign observed features (e.g. RFID tags) to a topological node either add a fixed number of features to each node [11] or add features to a node until localization of the robot performs poorly [9]. Our approach, however, follows Blanco et al. [13] and uses the *normalized cut* criterion to group observations that are strongly connected. Integrated with the hybrid metric-topological algorithm, this will increase the statistical independence between metric submaps which we will generate in a later stage for each topological node. We extend the offline algorithm by Blanco et al. [13] to RFID measurements and propose an online version of this approach.

A. Normalized Cuts

The Normalized Cut was first proposed by Shi and Malik [14] as a solution to the image segmentation problem in computer vision. The algorithm examines similarities between nearby pixels and separates groups of pixels that are connected by weak similarities. The pixels are described by a weighted undirected graph $G = (V, E)$ where the weight w_{ij} on each edge E_{ij} is a function of the similarity between two pixels, i.e. nodes V_i and V_j. The goal is to separate the graph into a disjoint set of nodes with high similarity measure among the nodes inside a set and low similarity measure across different sets.

In graph theory, a *cut* is defined as the total weight of the edges that connect two disjoint sets A, B where $A \cup B = V$ and $A \cap B = \varnothing$:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{1}$$

Figure 2 illustrates a weighted undirected graph with a possible cut that separates the nodes into two groups. Using the *minimum cut* as a segmentation criterion, however, does not result in reasonable clusters because the criteria favours small sets of isolated nodes in the graph. Therefore, Shi and Malik proposed a new measure of disassociation between two groups: The *normalized cut* (Ncut) is defined as

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{2}$$

where the total connection weight from nodes in A to all nodes in the graph V is defined as,

$$assoc(A, V) = \sum_{i \in A, j \in V} w_{ij}, \tag{3}$$

and analogous for $assoc(B, V)$. The fitness of a segmentation is better reflected by the normalized cut, because it looks for connections which are weak compared to all of the connections both inside and emanating from a segment.

It is easy to verify that the minimum possible value of the normalized cut (2) is zero when there is no connection between the two segments. The worst possible cut has the value 2 in the case that the nodes in a segment are only connected to nodes in the other segment. Unfortunately, the problem of finding the exact *min-Ncut* segmentation is NP-complete [14]. Therefore, Shi and Malik propose an approximate approach, which produces near-optimal cuts and is based on solving a generalized eigenvalue system. This approach is summarized below.

We are looking for a partition of the graph V into two disjoint sets A and B. Let x be an $N = |V|$ dimensional indicator vector with $x_i = 1$ if node i is part of group A and $x_i = -1$ otherwise. W is an $N \times N$ symmetrical weight matrix with $W(i, j) = w_{ij}$ and D is an $N \times N$ diagonal matrix with $d(i) = \sum_j w_{ij}$, the total connection from node i to all other nodes, on its diagonal. Shi and Malik [14] show that the eigenvector belonging to the second smallest eigenvalue of the generalized eigensystem

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}\mathbf{z} = \lambda\mathbf{z} \tag{4}$$

is the real valued solution of the normalized cut problem. Since the eigenvector can take on continuous values, a splitting point to partition it into two parts must be chosen. One can take for instance zero or the median value as the splitting point. Another approach is to find the lowest normalized cut value by checking all possible splitting points. For our implementation we chose the last option. In order to split the graph into multiple groups we apply the algorithm recursively as proposed by Blanco [13].

B. Topological SLAM Using the Normalized Cut Criterion

Our algorithm requires two weighted undirected graphs; the nodes of the first graph $G = (T, E)$ are the observed RFID tags t_i and the edge weight between two tags in the graph reflect our estimate on how close the two tags are located together. This *co-occurrence graph* G is extended in an online fashion as new tags are observed. Simultaneously, we split the co-occurrence graph into partitions using the min-Ncut criterion. The group of tags in each partition form a node in the second, overlaying graph (see Fig. 3). The overlaying graph is our topological map.

If two tags are observed simultaneously, the edge in the co-occurrence graph between the tags t_i and t_j takes the value

$$w_{ij} = \frac{RSS(t_i) \cdot RSS(t_j)}{RSS(t_i) + RSS(t_j)}. \tag{5}$$

The weight needs to be high if both tags are observed with a high RSS value and a low if one or both tags have a low RSS value. Note that the RSS value is a positive scalar value which is a measure of the radio frequency signal power as received at the receiver.

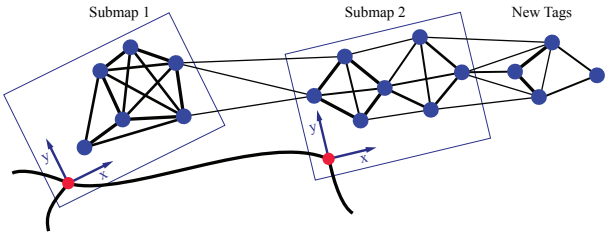


Fig. 3: All tags (blue circles) within a submap are assigned to a topological node (red circle). The min-Ncut criterion tells whether newly observed tags are assigned to an existing submap or form a new submap themselves.

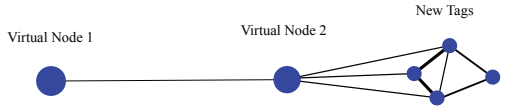


Fig. 4: Before applying the cut, existing submaps are reduced to single virtual nodes, thereby it is guaranteed that existing submaps are not partioned.

In order to implement an online version of the normalized cut algorithm, we make the following design choices

1) As soon as a tag is assigned to a group, it remains in a group with its group members.
2) Two groups can be merged but never partitioned.

Our online normalized cut algorithm works as follows: The robot moves around and registers the $RSS(t_i)$ of each detected tag t_i. Every Δt_{cooc} (2 sec in our implementation), the robot computes the weights according to (5) between all tags that it discovered during the last period. A weight is only stored in the co-occurrence graph if it is larger than the existing weight. Every Δt_{cut} (12 sec in our implementation), the recursive min-Ncut algorithm is applied to test if the graph can be partitioned in two or more groups of tags. Multiple groups are created if the normalized cut value of their connecting weights, computed with (2), is lower than a threshold θ.

In order to not split up a previously-created group, we reduce every existing group to a single virtual tag before we perform the recursive min-Ncut algorithm (Fig. 4). The weight between a virtual tag and every newly observed tag is set to the average weight between the new tag and all tags in the existing group. This choice allows us to add every newly observed tag to the group that it is most connected to. By choosing the mean connectivity the expanse of each group is bounded; since the distance between newly observed tags and the center of the group exceeds the range of the RFID reader, the mean connectivity drops fast. The same principle is applied to weights between virtual tags; we also choose the average of all weights that connect the two groups. Two groups are merged only if their average inter-connection is above threshold θ. Given the virtual tags, we must check before merging if in each group one or more virtual tags exist. If there is no virtual tag, a new group is created; if there is a single virtual tag, all newly observed tags are added to this group and if there are multiple virtual tags, all groups and newly observed tags are merged with the group that already contains the most tags. In case the algorithm returns a non-valid separation with a normalized cut value above θ, we merge all tags together.

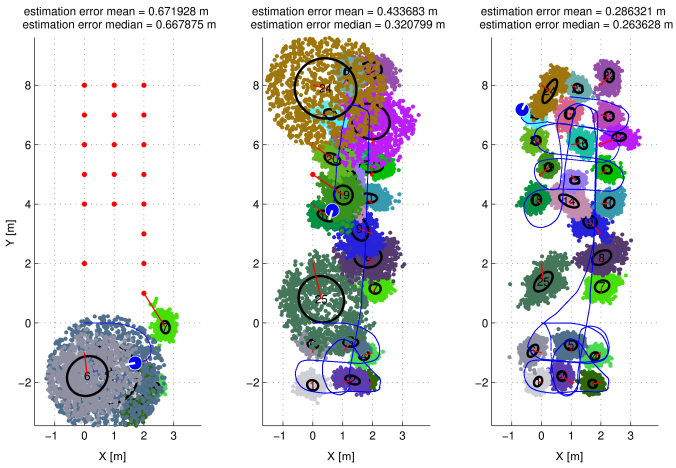


Fig. 5: The metric mapping process using a particle filter. The final average error in the tag position estimate is 0.29 m.

In order to keep the computational complexity from increasing with the size of the map, we maintain a list of *active tags* which were observed during the last period Δt_{cut}. Additionally, an active tag activates all its member tags in its group. By using the list of active tags and the reduction to virtual tags, the complexity of the eigenvalue problem in the recursive min-Ncut algorithm can be kept low, especially since the connectivity matrix is symmetric and only the second largest eigenvector is required. The reduced connectivity matrix W in our experiments did not exceed the size of 15×15. This however depends on the density of tags and the value for Δt_{cut}. The connectivity between nodes in the topological graph is addressed by the same ordered list of active tags. When the robot moves from one topological node to another, the two nodes are connected in the topological graph.

IV. Metric Mapping

The second building block of the hybrid algorithm is a method to produce a metric submap of RFID tag locations. We use a particle filter to locate the RFID tags. This approach to locate RFID-tags was first proposed by Hähnel et al. [1] and is summarized as follows.

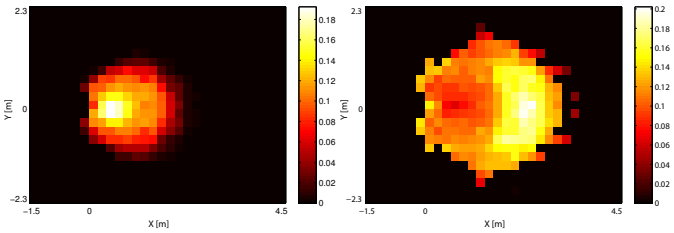
A. Particle Filter for RFID Tag Localization

We represent our belief of the tag position x_{tag} by a set of random samples which are drawn from the posterior distribution $p(x_{\text{tag}}|z_{1:t})$. Where $z_{1:t}$ are the collected measurements from timestep 1 until t. Each sample x_{tag} from the posterior distribution is a hypothesis of the true tag position at time t. This representation is approximate, but it is nonparametric, and therefore we can represent a broad range of distributions. We call the samples *particles* and represent them by

$$\mathcal{X}_{\text{tag}} := x_{\text{tag}}^{[1]}, x_{\text{tag}}^{[2]}, \ldots, x_{\text{tag}}^{[N]}, \tag{6}$$

where N is the number of particles. We assume that consecutive measurements are independent, given the pose of the tag, and formulate the recursive Bayesian filtering scheme:

$$p(\mathbf{x}|\mathbf{z}_{1:t}) = \alpha p(\mathbf{z}_t|\mathbf{x}) p(\mathbf{x}|\mathbf{z}_{1:t-1}), \tag{7}$$



(a) RSS = 3000 (b) RSS = 500

Fig. 6: Likelihood of tag position relative to the robot for given RSS.

where the key term in this equation is the likelihood $p(\mathbf{z}_t|\mathbf{x})$, it specifies the probability of making observation $\mathbf{z}_t$ given the position $\mathbf{x}$ of the tag relative to the robot. The likelihood function is specified by the sensor model which is discussed in the next section. The second term $p(\mathbf{x}|\mathbf{z}_{1:t-1})$ is the posterior from the previous timestep and α is a time-dependent normalizing factor. Note that the relative position of the tag to the robot is a function of the robot position and orientation as well as the tag position: $\mathbf{x} = \mathbf{x}(x_{\text{tag}}, x_{\text{robot}})$. Given a prior guess of the particle positions $p(\mathbf{x})$, we can use the recursive formula (7) to iteratively incorporate new measurements and update our estimate on the tag position. More details on the implementation of a particle filter are given in [15].

B. Sensor Model

We adopt the approach by Joho et al. [2] for the sensor model $p(\mathbf{z}|\mathbf{x})$ which specifies the likelihood of observation $\mathbf{z}$, given the position $\mathbf{x}$ of a particle relative to the robot. The source of information is modeled explicitly by two variables; the logarithmic received signal strength $s = \log(RSS)$ and the binary variable d which models the detection of a specific tag. We can write and expand the sensor model as follows:

$$\begin{aligned} p(z|x) &= p(s,d|\mathbf{x}) = p(s|d,\mathbf{x}) \cdot p(d|\mathbf{x}) \\ &\propto \frac{1}{\sigma_{x,y}\sqrt{2\pi}} \exp\left(-\frac{(s-\mu_{x,y})^2}{2\sigma_{x,y}^2}\right) \cdot p_{x,y} \end{aligned} \tag{8}$$

The sensor model is a product of the probability that we observe a tag at position $\mathbf{x}$ relative to the robot and the probability that we observe the logarithmic signal strength s of a tag at that very relative position. Further, Joho et al. model the received signal strength at a relative position $\mathbf{x} = (x, y)^T$ with a Gaussian distribution with mean $\mu_{x,y}$ and variance $\sigma_{x,y}$ and the detection probability at a relative position is abbreviated with $p_{x,y}$. The sensor model is implemented by discretizing the space relative to the robot with a grid. In each grid cell we store the mean $\mu_{x,y}$, the variance $\sigma_{x,y}$ and the detection probability $p_{x,y}$. In order to learn the sensor model, we placed the RFID tags in a regular arrangement around the robot. For three different orientations of the tags we made 6000 measurements each. We chose different orientations in order to increase the variation of the received signal strength and thereby simulate the signals which the robot will encounter in practice. Figure 6 illustrates the probability of tag position given an observed signal strength.The low precision exhibited in these models

is one of the main challenges in RFID-based localization and mapping.

V. Hybrid Metric-Topological SLAM

Given the two building blocks that allow us to cluster observed RFID tags into topological nodes and to create metric submaps for each node, we can now formulate the full hybrid metric-topological SLAM algorithm.

A. The Hybrid Algorithm

We maintain an *active metric map* which is continuously extended and improved as RFID tags are observed (see Sect. IV). A fixed number of the latest observations (10 in our implementation) define in which topological nodes the robot is currently located, these are the *active nodes*. The number of active nodes can be more than one if the robot is traversing submaps. In parallel, we run the topological SLAM algorithm (see Sect. III). If the topological algorithm returns a feasible min-Ncut or if the robot has moved to another topological node, the active metric map is partitioned. We remove tags which do not belong to the currently active nodes and save those parts as submaps of the corresponding nodes. If a topological node already possesses a submap, the new submap is merged with the old map in order to improve it. The process of merging submaps is described in the next section.

Every timestep
- Update active metric map

Every Δt_{cooc}
- Update tag co-occurrence graph

Every Δt_{cut}
- Check if cut of tag co-occurrence graph is possible $\Rightarrow$ If YES: Perform cut and update topological map. If two nodes are merged, the metric maps must also be merged
- Partition active metric map
- Compute active nodes
- Store metric map of non-active nodes and merge with old map if necessary

We perform SLAM only on the topological level where the last observed RFID tags define the current topological node. On the metric level however, we are always in "mapping mode" due to the low precision of the RSS as a distance metric (see Fig. 6). We propose to improve the existing submaps by remapping and merging them with the existing submaps.

A side result of our algorithm is the ability to mark the location of neighbouring tags in each submap (Fig. 10). We store in each submap the whole active map but assign each tag an indicator if it belongs to the submap or if it is from a neighbouring node. Neighbouring tags in submaps can be used as direction indicators to find neighbouring submaps.

In an extended framework, the robot can leave the "mapping mode" and use the same sensor model from Sect. IV to locate itself in the submap (see [1], [2]). The robot can reach its goal by first searching for the shortest path in the topological map (e.g. with A^* search) and then by local path planning in the metric submaps.

B. Merging of Submaps

In order to improve a submap, we propose to map it again and merge it with the existing one. We are given the set of n tags in both maps $\{x_i, x_i'\}$ with known correspondences and tag position distributions: $x_i \sim p_i(x)$ and $x_i' \sim p_i'(x)$. The desired rotation and translation $\{R, t\}$, which optimally aligns the second map with the first one is the one that maximizes the overlap between the two maps. Therefore, we can write:

$$\{R,t\} = \arg\max_{R,t} \sum_{i=1}^{n} \iint p_i(x)\ p_i'(x|R,t)\ dx \qquad (9)$$

If we model the particle distribution for each tag with a Gaussian, i.e. $p_i(x) \simeq \mathcal{N}(\mu_i, \Sigma_i)$, a closed-form solution can be found for the integral. The product of two Gaussians gives an un-normalized Gaussian [16]:

$$\mathcal{N}(\mu_i, \Sigma_i)\ \mathcal{N}(\mu_i', \Sigma_i') = Z \cdot \mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i). \qquad (10)$$

The value of the integral in the optimization function amounts exactly to the normalization factor Z:

$$\begin{aligned} Z =& \frac{1}{2\pi}\ |\Sigma_i + \Sigma_i'|^{-\frac{1}{2}} \\ & \cdot\ \exp\big(-\frac{1}{2}(\mu_i - \mu_i')^T(\Sigma_i + \Sigma_i')^{-1}(\mu_i - \mu_i')\big). \end{aligned} \qquad (11)$$

We initialize the alignment using a least squares method and then solve (9) using the Nelder-Mead simplex algorithm [17] to obtain $\{R, t\}$.

The distribution of the merged tag positions is directly derived from Equation (10). The fused Gaussians have mean and variance:

$$\begin{aligned} \hat{\mu}_i &= \hat{\Sigma}_i(\Sigma_i^{-1}\mu_i + \Sigma_i'^{-1}\mu_i') \\ \hat{\Sigma}_i &= (\Sigma_i^{-1} + \Sigma_i'^{-1})^{-1}. \end{aligned} \qquad (12)$$

This alignment procedure gives a higher weight on tags with high position certainty both in the alignment and the fusion step.

VI. Experimental Evaluation

To evaluate our approach, we mounted an ALIEN ALR-9900+ RFID antenna at the front of a Pioneer platform from Mobile Robots and distributed 25 tags in the environment in 3 different arrangements (see Figures 7, 8 and 5). The robot was steered by a human operator and collected RFID as well as odometry measurements. We used the presented algorithm in an offline stage to reconstruct the robot's path given this information. It was necessary to drive the robot very slowly at approximately 10 cm/s in order to receive enough sensor measurements to produce good results. The detection frequency is however dependent on the total number of observed tags.

Figures 7 and 8 illustrate the performance of the online min-Ncut algorithm. The images in the left show the true

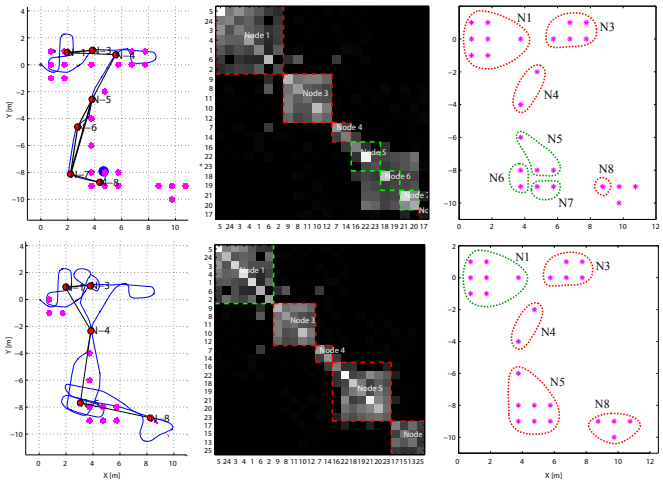


Fig. 7: The left image illustrates the robot path estimated by odometry only, the middle image shows the matrix representation of the partitioned co-occurrence graph (green square is active node) and the right image shows the partitioning by the online min-Ncut algorithm. In the upper row the robot observes tags from node 5, 6 and 7 simultaneously; therefore, these nodes will be merged in the next step.

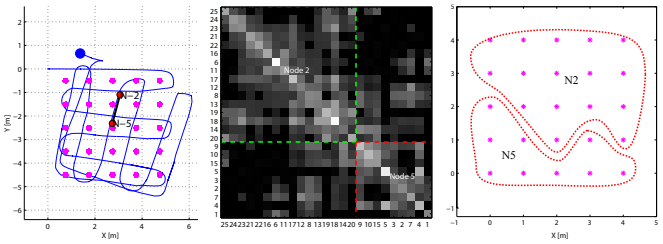


Fig. 8: This example illustrates how the algorithm bounds the size of the submaps. Due to the limited range of the antenna, the average connectivity between the two nodes (mean of all values in the top right corner of the connectivity matrix) will not cross the specified threshold θ with further exploration.

tag positions (magenta), the robot pose (blue circle), the topological graph (red circles and black lines) as well as the robot trajectory, estimated by odometry only (blue line). In the middle we can see the matrix representation of the co-occurrence graph with the active nodes colored green and on the right is the partitioning of the observed tags. In the first row of Figure 7, the robot observes tags from node 5, 6 and 7 simultaneously and therefore the algorithm merges the nodes in the next min-Ncut evaluation. The final result in the lower row illustrates the performance of the online min-Ncut algorithm. Reasonable clusters are generated. Figure 7 shows that the size of the submaps are bounded. Further mapping does not lift the average connectivity between the nodes over the threshold θ since the tags in the top right corner cannot be observed together with the tags in the bottom left corner.

In the purely metric part we achieved a final average tag position error of 0.29 m (see Fig. 5). Thereby, we achieved the same order of accuracy as the results by Joho et al. [2].

Figure 9 illustrates the result of the hybrid algorithm. The average estimation error is 0.31 meters. In the environment of Figure 5 we achieved a localization accuracy of 0.18 m using the hybrid approach with two submaps. The final accuracy of the maps in Figure 8 was 0.20 m. Since the submaps are not bound to a global reference frame and only indicate the

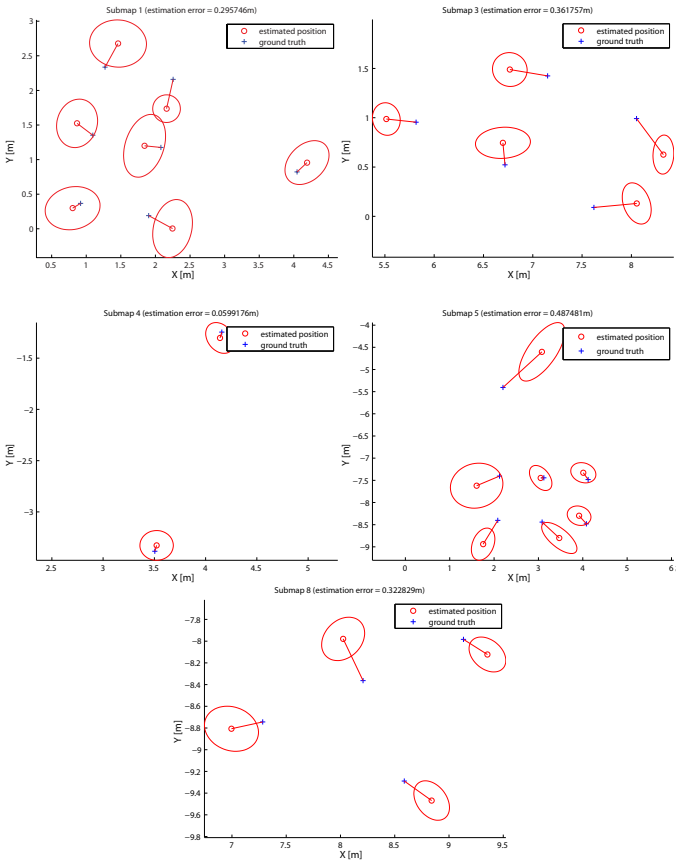


Fig. 9: Final estimation error for the relative tag positions of the submaps in Fig. 7. The average error for all submaps is 0.31 m.

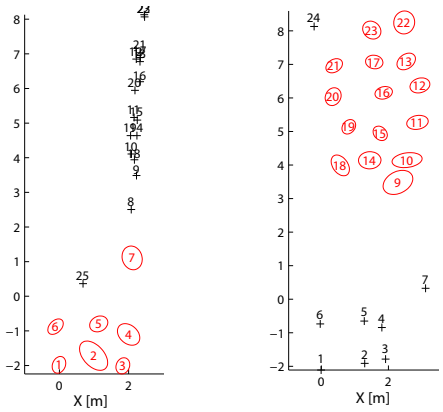


Fig. 10: Final submaps of environment in Figure 5. The mean tag position error is 0.29 m. The black marks illustrate the approximate location of adjacent tags which do not belong to the submap. This information can be used for submap traversal.

relative position of the tags, we measured this error by fitting the true positions as good as possible on the local submaps.

The result of the algorithm is promising given the fact that the final odometry error is 4.5 meters in Figure 7. The experiments were conducted in a large open space on a planar surface and the tags were stiched to the floor. We assume that the accuracy of the mapping is lower if many obstacles are present since they are currently not modeled with the sensor model. The experiments further showed that the algorithm

successfully improved existing maps. The results of the tag localization are similar to the basic metric mapping algorithm of Section IV for small environments. If the basic mapping algorithm was applied to the environment in Figure 7, the map would exhibit large errors due to the large odometry error. By using the hybrid framework, the accuracy of local maps is independent of the globally accumulated odometry error. Even the kidnapped robot problem can be handled.

However, the hybrid framework also introduces a problem. When the robot moves out of a node and into another submap without mapping the previous node thoroughly, an inaccurate submap is stored. We can improve the submap by later remapping but the robot will have to remap it from the beginning. Therefore, the mapping result in the hybrid algorithm may be worse than the basic mapping algorithm, at least in the beginning.

The experiments in three different environments have shown that the hybrid algorithm is indeed able to construct accurate local maps as well as a topological graph that connects the submaps. We consider the proposed algorithm as a valid solution to map vast environments equipped with RFID tags with odometry only since the accuracy does not deteriorate with the total size of the environment.

VII. Conclusion

In this paper a novel RFID SLAM algorithm was demonstrated which is based on a hybrid metric-topological mapping algorithm. The framework allows for the creation of large-scale maps with odometry and RFID sensors only. The average estimation error in the metric submaps is approximately 0.3 meters and the diameter of the submaps is bounded to approximately two to three times the range of the RFID antenna. The hybrid metric-topological framework provides a compact world model which does not seek global metric consistency but provides precision in important regions as well as robustness. Important regions may be defined by areas with higher tag density, where improved localisation accuracy becomes available. This may be exploited in environments where several work areas are connected by passageways. In the passages/corridors, the robot would only need an RFID tag every few meters to find its way between high accuracy regions or work areas. In these work areas, the density of the RFID tags can be used to indicate areas of importance. Furthermore, due to the hybrid nature of the map, if the tags within a work area are changed, the map may be easily updated to reflect these changes without affecting the whole map.

VIII. Acknowledgements

The authors would like to thank Liam Candy for providing his ROS driver for the ALIEN RFID system.

Fig. 11: Platform used for the experiments.

References

[1] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and Localization with RFID Technology," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1015–1020, 2004.

[2] D. Joho, C. Plagemann, and W. Burgard, "Modeling RFID Signal Strength and Tag Detection for Localization and Mapping," *IEEE International Conference on Robotics and Automation*, pp. 3160–3165, May 2009.

[3] P. Vorst, S. Schneegans, B. Yang, and A. Zell, "Self-Localization with RFID Snapshots in Densely Tagged Environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1353–1358, 2008.

[4] F. Seco, C. Plagemann, and A. R. Jim, "Improving RFID-Based Indoor Positioning Accuracy Using Gaussian Processes," *Indoor Positioning and Indoor Navigation*, pp. 15–17, 2010.

[5] P. Vorst and A. Zell, "Fully Autonomous Trajectory Estimation with Long-Range Passive RFID," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1867–1872, 2010.

[6] N. Lavigne, J. Marshall, and U. Artan, "Towards Underground Mine Drift Mapping with RFID ," *Canadian Conference on Electrical and Computer Engineering*, vol. 23, pp. 1–6, 2010.

[7] A. Kleiner, C. Dornhege, and S. Dali, "Mapping Disaster Areas Jointly: RFID-Coordinated SLAM by Humans and Robots," *IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 1–6, 2007.

[8] H. Choset and K. Nagatani, "Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization," *IEEE Transactions on Robotics*, vol. 17, no. 2, pp. 125–137, 2001.

[9] M. Bosse, P. Newman, J. Leonard, and S. Teller, "SLAM in Large-scale Cyclic Environments using the Atlas Framework," *International Journal on Robotics Research*, vol. 12, no. 23, pp. 1113–1139, 2004.

[10] J.-L. Blanco, J.-a. Fernandez-Madrigal, and J. Gonzalez, "Towards a Unified Bayesian Approach to Hybrid Metric-Topological SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 259–270, 2008.

[11] C. Estrada, J. Neira, and J. Tardos, "Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.

[12] Y. Raoui, M. Goller, M. Devy, T. Kerscher, J. Zollner, R. Dillmann, and A. Coustou, "Rfid-based topological and metrical self-localization in a structured environment," *International Conference on Advanced Robotics (ICAR)*, june 2009.

[13] J.-L. Blanco, J. Gonzalez, and J.-a. Fernandez-Madrigal, "Consistent Observation Grouping for Generating Metric-Topological Maps that Improves Robot Localization," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 818–823, 2006.

[14] J. Malik and J. Shi, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.

[16] C. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[17] J. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.