Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots

Robert Johansson and Alessandro Saffiotti
AASS Mobile Robotics Lab
Örebro University, S-70182 Örebro, Sweden
{robert.johansson,alessandro.safffiotti}@aass.oru.se

***Abstract*— Stigmergy is a mechanism that allows the coordination of actions within the same agent or across different agents by means of traces left in the environment. We propose a stigmergetic approach to robot navigation in which a robot sets values in a hexagonal grid of RFID tags buried under the floor. This approach only requires minimal resources on the robot. The RFID floor will eventually contain a distance map that can guide the robot to a given goal (or set of goals) without the use of any localization system. The same map can be used or improved by other robots or by the same robot at later times. We define algorithms for building the RFID-floor map and for navigating on this map, we prove the convergence of the map building algorithm, and we show an empirical validation of our results using a small robot in a domestic environment.**

I. Introduction

Point to point navigation is arguably the most fundamental ability needed by any autonomous mobile platform. Typically this is achieved by a combination of path planning and path following, and requires that the robot has a map of the environment and is able to reliably assess its own position within this map [1]. The problems of map building and of self-localization have been the subject of intensive study, and many effective solutions are now available [2], [3]. These solutions typically rely on the use of data-rich sensors, like laser range scanners or cameras, and on algorithms which are often very demanding both in terms of memory and computation. Despite the drop in price of sensors and processing resources, considerations of cost, space, energy and reliability still make the use of these solutions prohibitive in mass-produced consumer robots.

A partial solution to this problem is to engineer the environment by adding some active infrastructure. For instance, the Roomba vacuum cleaning robots use beacons in the base station to autonomously return to the charger, and some systems use external cameras to provide self-localization to a sensor-poor robot [4], [5]. These solutions, however, increase the installation and maintenance costs of the system.

In nature, a mechanism is often seen by which relatively simple animals can achieve complex behavior by exploiting the environment as a resource for storage or communication of information. This mechanism is called *stigmergy*.

> Stigmergy is a mechanism of spontaneous, indirect coordination between agents or actions, where the trace left in the environment by an action stimulates the performance of a subsequent action, by the same or a different agent. [6]

The concept of stigmergy has been exploited in several cooperative robotic systems, or "swarm" systems [7]. In these, many simple robotic units apply simple individual, local rules to achieve complex collective, global behavior. The robots cooperate indirectly in a stigmergetic way by leaving and reading information in the environment [8].

In this paper, we present a stigmergetic approach in which the maps used for navigation are stored in the environment itself—namely, in the information contained in a regular grid of RFID tags on the floor. The navigation maps that we store are distance maps to a given goal: each tag stores the distance to the goal, in terms of the number of grid cells on the shortest collision-free path from the tag to the goal.

We show the potential of this idea in two steps. In the first step, we show that a robot equipped with an RFID tag reader can exploit this map to navigate to the goal from any position in the environment by simply following the gradient of the information stored in the tags. Interestingly, the robot itself does not need any internal storage or any self-localization ability to do this; in particular, it does not need to possess any additional sensing ability (except, if desired, for collision detection). Even an extremely sensor-poor and processing-poor robot can perform goal directed navigation in this way.

In the second step, we show that the map itself can be autonomously built by the robot with a very simple algorithm, without the need for any global or local position estimate, and with minimalistic processing and memory requirements. Our map building process is an anytime process: the map is continuously updated over time, and building can be suspended and resumed without saving any state in the robot. It can also resumed by another robot, or performed by several robots simultaneously, without any change to the algorithm.

In this paper, we describe the algorithms for exploration (map building) and exploitation (target reaching), we give a proof of the eventual convergence of the map building algorithm, and show an empirical evaluation of both algorithms.

II. Related Work

Many approaches have been proposed in the multi-agent and multi-robot system literature based on the idea of stigmergetic communication. These usually focus on the achievement of emergent collective behavior, like foraging or flocking [9], [7], [10]. Stigmergy has also been extensively exploited to create swarms of artificial agents (ants) able to solve optimization problems [11], but in that case a software

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

Fig. 1. Layout of the environment used in our development, showing the position of the RFID tags buried under the floor. The circles indicate the two goal tags used in the experiments described in this paper.

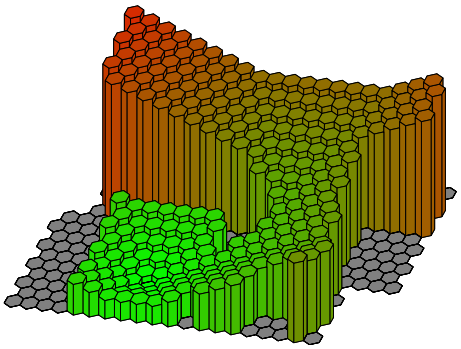

Fig. 2. Distance map computed offline by wavefront propagation.

rather than a physical medium is used. In our work, we focus on the physical implementation of stigmergy for robot navigation, using RFID tags as a medium.

The use of RFID tags in robot navigation is becoming increasingly popular. In most cases, tags must be initialized by writing in them some *a-priori* information, usually about their global position [12], [13], [14], [15]. Our approach does not require the storage or *a-priori* information in the tags, nor does it rely on any global position information.

Some authors use RFID tags to realize stigmergetic communication. Herianto *et al* [16] study the use of pheromone-based potential fields for robot navigation. Mamei and Zambonelli [17] use RFID tags distributed in the environments to allow humans and robots to mark objects and places, and to leave trails that lead to them. Ziparo *et al* [18] propose an approach in which robots deploy RFID tags during exploration and later use them for localization. These works share the stigmergetic medium considered here, but they address problems different from shortest path navigation.

O'Hara *et al* [19] use a pervasive network of devices in the environment to compute a navigation graph that can be exploited by a resource-poor robot. Interestingly, paths can quickly adapt to changes in the environment in their approach. The price to pay is the use of active devices, which must be deployed in such a way that communication between them is blocked by the same obstacles that block navigation. By contrast, the RFID tags used in our approach are small, cheap, do not need a internal power source, and are easily embedded in the environment.

III. Navigating on an RFID Distance Map

This section addresses the first step in our development: *if* the RFID tags in the floor contain a distance map to a goal, *then* we can use a simple gradient descent algorithm to navigate to that goal.

A. Distance Maps

In this paper, we assume that a set of RFID tags are layed down on the floor (or buried under it) in a regular hexagonal grid. For example, Fig. 1 shows the tags in the environment used in our experiments. The grid implicitly defines a tessellation of the floor in hexagonal cells, corresponding to the Voronoi decomposition generated by the tag locations. This grid can be seen as a graph (X, E) where the vertices X are the tags, and the arcs E are defined by the 6-neighbor relation between tags. The choice of a hexagonal layout is motivated by its isotropic property: neighbors of any node in a hexagon graph are located at an equal (geometric) distance, which makes it easier to build a distance map.

We define our distance map to be a *shortest path tree* (SPT) constructed out of this hexagonal graph. A SPT solves the *single-source shortest path problem*, which is the problem of finding the shortest path between a predefined root node $x_0 \in X$ to all other nodes in X [20]. A SPT indirectly solves this problem by annotating all nodes in X with the minimum distance to x_0. We denote by $d(x)$ the annotated distance for an arbitrary $x \in X$. Nodes that are unreachable from the root node are annotated by ∞. In our setting, unreachable nodes include tags which are underneath obstacles.

The SPT for a given hexagonal graph can be easily built by standard graph search algorithms [21]. In particular, since the weights in our case are all uniform, simple breadth-first search can be used to produce the correct SPT: a search wavefront is expanded from the goal node outwards, and each node is marked with its distance to the goal node. Similar path planning methods are well known in the mobile robotics field [1], and they go under the names of wavefront propagation [22] or distance transforms [23]. Although these methods are usually applied to square grids, their extension to hexagonal grids is straightforward. Fig. 2 shows the distance map obtained by applying wavefront propagation to the map in Fig 1, starting from a goal node in the bedroom.

Once a distance map has been generated, it can be stored in the RFID tags in the floor by storing the value $d(x)$ in each tag x. It should be noted that this requires that a metric map of the environment is available, and that the position of each tag in this map is known. In Sec. IV we will show how the distance map can be built directly on the RFID tags while the robot navigates, with no prior metric information.

B. Gradient Descent Navigation

If a distance map is stored in the RFID tags on the floor, a robot equipped with an RFID tag reader can reach the goal tag from any location in the environment by simply following the negative gradient of the values in the tags. Intuitively, the robot would perform a "physical" gradient descent on

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

Algorithm 1 GradientDescent()

Require: Robot with a single RFID tag reader in front
Ensure: Robot reaches the goal tag

```
1: OrientToGradient()
2: last_dist ← ReadCurrentTag()
3: while last_dist > 0 do
4:    move forward one step
5:    new_dist ← ReadCurrentTag()
6:    if (new_dist > last_dist) or facing an obstacle then
7:       OrientToGradient()
8:    end if
9:    last_dist ← ReadCurrentTag()
10: end while
```

the $d(x)$ function until it finds the global minimum. By construction of the distance map, the $d(x)$ function has exactly one local minimum, which is also the global one.

In practice, gradient descent can be realized in several ways depending on the available hardware. If the robot has a ring of RFID tag readers, it can read the values in the tags around it at every step and continuously adapt its heading to the direction of the steepest negative gradient. In our experiments, however, we decided to rely on a simpler hardware: a robot that has a single tag reader mounted at its front, as shown in Fig. 3. In this case, the robot has to stop and perform a full rotation in place in order to read the tags around it and estimate the gradient. Gradient descent, then, can be realized by first estimating the direction of the steepest negative gradient, and then moving in that direction until a raise in gradient is detected, at which point the gradient is estimated again. Algorithm 1 formalizes this strategy.

The `OrientToGradient` step in the algorithm is performed as follows. The robot makes a 360° rotation in place, and it records its own heading when each tag enters and leaves the reader's range during the rotation. From this, we can estimate the direction to each detected tag around the robot. The direction of the gradient, then, is set to the direction to the tag holding the lowest distance value. If multiple tags all have the lowest value, we take their average direction by summing the corresponding direction vectors. The robot finally turns to the computed direction.

Algorithm 1 can be run on minimalistic robots. It requires one memory cell to store `last_dist`, plus a few cells to record the directions and values of the neighboring cells in

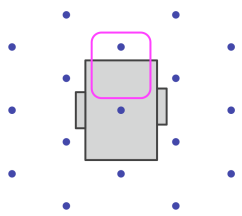

Fig. 3. The minimalistic robot configuration assumed in our gradient descent algorithm. A single RFID tag reader is mounted in the front, whose range is shown in red. A full rotation is needed to compute the gradient.

`OrientToGradient`. The algorithm only assumes that the robot is equipped with an RFID tag reader, a collision sensor (e.g., bumpers) and an orientation sensor (e.g., odometry). No ability to sense environmental features or to estimate absolute location is assumed. If the robot had multiple RFID readers, even the orientation sensor would not be needed.

C. Multiple Goal Points

Typical RFID tags have many writable fields, which can be individually addressed. This means that one can use the above method to store several distance maps in the RFID grid, corresponding to several predefined goal points. A robot placed on the floor can then navigate to any one of these goal points by simply using the corresponding field index when reading the contents of the RFID tags. The tags used in our installation, for instance, provide 64 individually addressable fields, but future tags are expected to have larger memories.

IV. Autonomous Building of Distance Maps

Having assessed the potentiality of a distance map stored in an RFID floor, we now address the second goal of this paper: to study how this distance map can be built on-line and autonomously by a minimalistic robot. Our most notable requirement is that map building should not rely on *any prior information*; in particular, no information about the location of the tags should be given, except that they are placed on a regular hexagonal grid. We also require that it is not necessary for the robot to maintain any position estimation, neither to sense any information from the environment — except for the contents of tags and collision detection.

On-line building of a distance map consists of two parts: an exploration strategy, and an update strategy.

A. Exploration Strategy

For a tag to be assigned a distance value, it must eventually get in the range of the tag reader. In fact, we show below that for the map building to converge, the exploration strategy must be *complete* in the sense that each tag-to-tag arc must be traversed infinitely often as the building time goes to infinity.

In our experimental system, we implemented a simple random walk strategy: (1) when no obstacle is detected, move forward; (2) when an obstacle is detected, perform a random rotation. This strategy is intuitively complete for all physically accessible parts of the grid, and it has produced complete coverage in all our experiments.

B. Update Strategy

As the environment is explored, the data of tags are to be read as they enter the reader's range. Depending on fetched data, an update of tag data might be performed. The update strategy should be such that tag data will eventually constitute a distance map.

Our update strategy is described in Algorithm 2, which is performed continuously during exploration (step 3). It can be seen as a process where a stigmergetic trail is accumulated in the environment. To determine what to store in each tag as they enter the reader's range, the algorithm maintains a

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

Algorithm 2 BuildDistanceMap()

Require: All tags have value ∞ except for one that has value 0 (origin)

Ensure: All tags eventually contain the distance to the origin

1: $X_{curr} \leftarrow \emptyset$
2: distance_counter $\leftarrow \infty$
3: **while** *Explore* **do**
4: $X_{prev} \leftarrow X_{curr}$
5: $X_{curr} \leftarrow$ TagsInRange() {*from tag reader*}
6: $X_{new} \leftarrow X_{curr} - X_{prev}$
7: **with random** $x \in X_{new}$ **do**
8: distance_counter $\leftarrow$ distance_counter + 1
9: **if** distance_counter $> v(x)$ **then**
10: distance_counter $\leftarrow v(x)$
11: **else**
12: WriteTag(x, distance_counter)
13: **end if**
14: **end with**
15: **end while**

distance counter. This is an upper bound on the minimum number of tags that must be traversed from the current position in order to reach the goal tag. The counter is incremented by one when a new tag is detected. Then the counter is compared to the value stored in the current tag x, denoted by $v(x)$: if the counter is greater than $v(x)$, then the distance to goal is lower than the current upper bound in the counter and the counter is updated; if the counter is lower than $v(x)$, then the value in the tag overestimates the distance and the tag is updated. This is basically an adaptation of the standard Bellman-Ford algorithm [20] to our specific case. Eventually, the value $v(x)$ in each tag x should converge to the true distance value $d(x)$.

A necessary condition for Algorithm 2 to converge is that any two consecutively accessed tags are adjacent tags. If this is not the case, it can be seen as a missed increment of the distance counter, which will lead to divergence. This translates into two practical necessary assumptions: (1) the while loop (steps 4–14) must be run frequently enough compared to the speed of motion of the robot; and (2) the number of $TagsInRange$ must be at least one (so the robot never passes in between tags) and at most three (so all tags read in one cycle are adjacent). In our formal proof below, we show that the algorithm converges under the assumption $|TagsInRange| = 1$. In the experiments reported in the next section, we will empirically show that it also converges when $1 \leq |TagsInRange| \leq 3$.

Algorithm 2 is a life-long learning algorithm: it can be run indefinitely during the lifetime of the robot — in fact, since the permanent storage is not in the robot but in the floor, it can be run during the lifetime of the floor. The robot can still update the tags in the background while doing other navigating tasks, possibly by exploiting the current RFID floor map. This has two interesting consequences. First, a robot can be placed on a partially computed floor map for a given goal, and it will simply continue to refine that map. Second, the map can adapt to expansions of the free space: if an area that was previously inaccessible becomes accessible (e.g., a piece of furniture is removed) then the values in the floor tags will be decreased as needed to account for the new topology. However, since the update strategy only allows decreases of tag values, the algorithm cannot handle the inclusion of new obstacles in the environment.

Another interesting property of Algorithm 2 is that it allows for cooperative map building, i.e., multiple robots can participate in a map build. There is no need for explicit coordination or communication between the robots; the stigmergetic communication resulting from reading and writing to tags is sufficient. Furthermore, there is no need for the robots to be aware of each other, their number, or locations.

Finally, multiple maps can be built for multiple goals by using different fields in the RFID tags. Interestingly, all the maps can be built simultaneously by the same robot, by just keeping a separate distance counter for each goal.

C. Convergence

We now prove that our map building algorithm eventually converges to the true distance map. That is, given enough time, the value $v(x)$ stored in each RFID tag will be the distance value $d(v)$ defined in Sec. III above.

We prove convergence under the following assumptions:

a) The reader never "skips over" a tag; and
b) At each cycle, there is exactly one tag in range; (1)
c) The exploration strategy is complete.

The first assumption means that the cycle time τ of the algorithm and the maximum speed V_{max} of the robot are such that $\tau V_{max} < \rho$, where ρ is the minimum distance between adjacent tags. This condition is easily satisfied. The assumption also means that all tags are responding to the reader. The second assumption means that the layout of the RFID tags and the receptive field of the RFID tag reader are such that the tags fields form a tessellation of the space; if this is not the case, there might be positions at which the robot does not read any tag ($|TagsInRange| = 0$), or it reads several tags ($|TagsInRange| > 1$). The third assumption means that the exploration strategy will eventually visit each arc between each pair of accessible tags, and that it will do so infinitely often. The only idealistic assumption is the second one: nonetheless, the experiments reported in the next section empirically show that even if this assumption is relaxed the algorithm still converges.

We define the *error* at time t of the built distance map by

$$Err(t) = \sqrt{\frac{1}{|X_t|} \sum_{x \in X_t} \left(v_t(x) - d(x)\right)^2} \qquad (2)$$

where X_t denotes the set of tags explored up to time t, $v_t(x)$ denotes the value stored in tag x at time t, and $d(x)$ denotes the distance of tag x from the goal cell defined in Sec. III.

The asymptotic convergence of the algorithm is stated by the following theorem, proven in the Appendix.

Theorem 1: Under the assumptions in (1), there is a time T such that, for any $t > T$, $Err(t) = 0$.

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

Fig. 4. The PEIS Home physical test-bed. Left: the kitchen. Right: the construction of the floor, showing the grid of RFID tags.

V. Experimental Validation

We have performed a number experiments to empirically validate our approach. We describe three suites of experiments aimed at evaluating three different aspects.

A. Experimental Setup

In all experiments we used the same experimental setup. The physical platform was the PEIS-Home, an apartment-like test-bed facility that incorporates a living room, kitchen, and a bedroom [24], [5]. The whole space is approximately 7 by 4 meters — see Fig. 1 above. In addition to common household appliances, the PEIS-Home contains several less usual objects, including 350 R/W RFID tags embedded underneath the parquet floor. The tags are of type *Tag-it HF-I Plus* and operate passively with a reader. The tags have been placed out according to a hexagon grid, where adjacent tags are spaced 26 cm apart — see Fig. 4.

The gradient descent and on-line map building algorithms have been implemented on a Centibot platform [25], extended by mounting an RFID reader on the top of it, with its antenna underneath the robot base and extending slightly from the front — see Fig. 5. The size of the antenna was such that between one and three tags were in range from any position on the floor, given the tag density in our environment. During all runs, translation speed was limited to 80 mm/s, and rotational speed to 0.4 rad/s. The control loop, including Algorithm 1 or 2, was run at 10 Hz.

B. Navigation on an Ideal Distance Map

The first suite of experiments was aimed at evaluating the performance of the gradient descent navigation on a single ideal distance map. The map used is the one shown in Fig. 2 above, i.e., the goal is located in the middle of the bedroom.

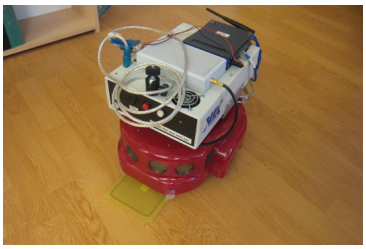

Fig. 5. The robot used throughout the experiments; a Centibot equipped with a RFID tag reader. The antenna is visible, extending from the base.

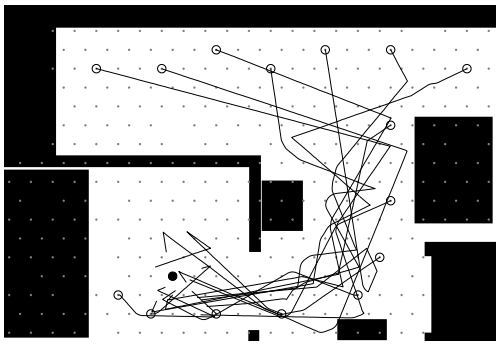

Fig. 6. Multiple navigation runs on the same distance map from different start locations. Open circles represent the starts; the filled circle is the goal.

Multiple Starts, Single Goal: In this experiment, we tested the ability of the robot to reach the goal from different starting locations throughout the environment. Since our navigation solution does not require that the robot knows its initial pose, we simply placed the robot at a starting location and let it go. When the robot reached the goal, we manually replaced it to the next starting position and released it again. Fig. 6 shows the path of the robot in fifteen navigation instances. At each run, the robot successfully reached the goal tag. The slight misplacement of the ending point of the trajectories is due to the fact that trajectories were recorded from uncorrected odometric data.

Single Start, Single Goal: In this experiment, we tested the performance of the robot by repeatedly performing the same gradient descent navigation. We made ten runs using the same distance map and obstacle set-up. The goal of this experiment was to investigate repeatability and optimality. For each run, we measured traveled distance, estimated by odometry, and time. In all runs the robot successfully reached the goal tag. The results reported in Table I show a good degree of repeatability, since the minimum and maximum distances only differ approximately one meter.

The optimal navigation distance was determined to be 6.5 meters. This figure was calculated by counting (by hand) the minimum amount of tags between the start and goal tags, taking the size of the robot into account. As it can be seen in Table I, the mean navigation distance is 33% longer than optimal. We conjecture that this is due to the use of an orient-and-go strategy rather than a strategy where orientation is corrected continuously. Our strategy was dictated by the decision to use a robot with a single RFID tag reader.

C. On-line Building of a Distance Map

The second suite of experiments was aimed at evaluating the map building algorithm under real execution conditions. Two on-line builds were performed; one with the goal located

Property	Min	Mean	Max	Optimal
Distance (m)	8.04	8.63	9.20	6.5
Time (s)	231	249	274	–

TABLE I

Results of ten gradient descent navigation runs.

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

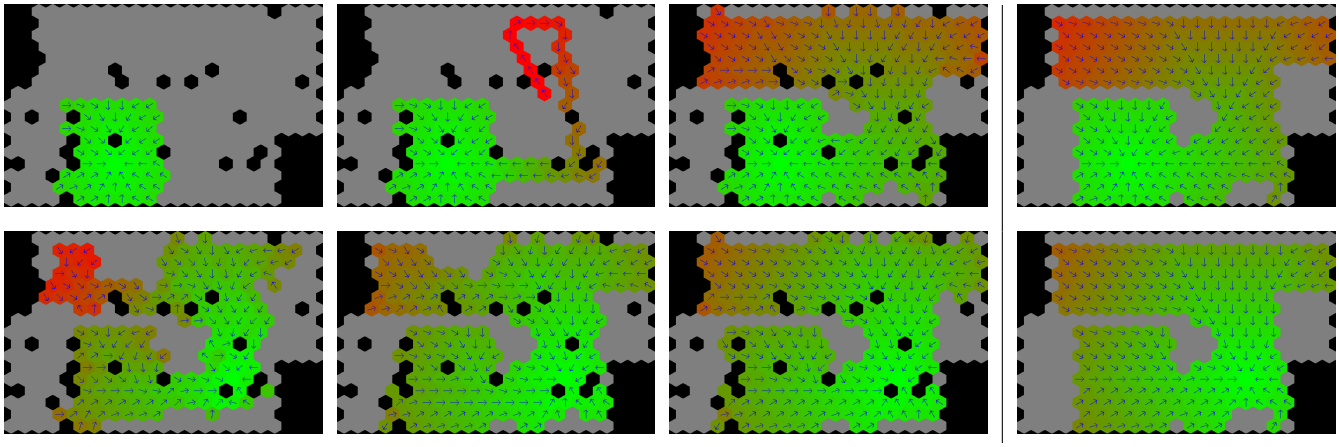

Fig. 7. Two on-line builds of distance maps. Colors indicate distance values: pure green is 0, pure red is 30. Arrows indicate the negative gradient direction for visualization purposes. Unexplored tags are grey, broken or missing tags are black. The top row shows the building from a goal in the bedroom: before the robot was released from the bedroom (left), right after the robot has left the bedroom (middle left), and when building ended after 20 hours (middle right). The bottom row shows the building from a goal in the living room: after 1 hour (left), before the kitchen chairs were removed (middle left), and when building ended after 20 hours (middle right). The rightmost column shows the ideal maps for the two cases, for comparison.

in the middle of the bedroom, and one with the goal in the lower regions of the living room (see Fig. 1 above). The rightmost column of Fig. 7 shows the ideal distance maps for these goals, built offline.

The result of map building for the goal in the bedroom are shown in Fig. 7 (top). Gradient arrows are only included for visualization purposes, they are not stored in the tags. The final map after $t = 20$ hours is rather close to the ideal distance map. Some residual differences remain due to the presence of broken tags (the black holes). Fig. 8 plots the error function $Err(t)$ (see equation (2) above) over t during the building process. The figure also shows the average difference between the direction of the gradient in the built map and in the ideal map, computed cell by cell. The trend of both curves suggest that convergence has occurred.

The bedroom experiment also tested the ability of our algorithm to *extend* an existing map when the environment gets larger. In our run, the robot was initially enclosed in the bedroom until the small map built there converged, at $t = 1.8$ hours, as shown in the top-left map in Fig. 7. At that moment, the door was open and the robot was allowed to exit the bedroom. The top-middle map shows the distance map right after (at $t = 1.9$ h): the stigmergetic trail left by the robot is clearly visible. This event is reflected in the error plot; there is a sudden rise at $t = 2$ h, due to the fact that the X_t set in equation (2) quickly extended from the bedroom to the full environment.

The result of the build with the goal in the living room is shown in the lower part of Fig. 7. The map when building ended is again very close to the ideal one. The error plots in Fig. 9 suggest that convergence has occurred. The faster convergence compared to the bedroom experiment can be explained by the central location of the goal: the robot often re-visits tags that are near the goal and have a value close to

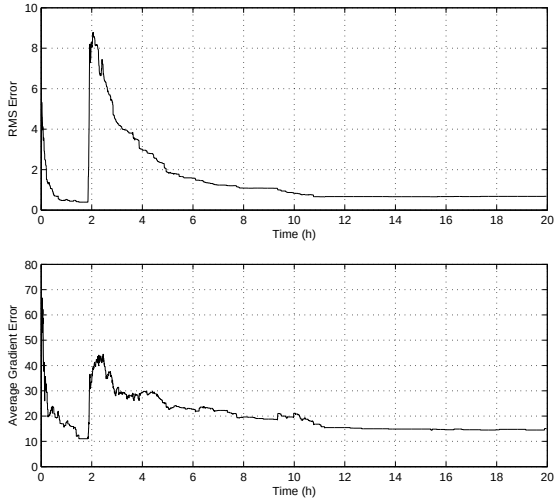

Fig. 8. Error functions during the build for the goal in the bedroom. Top: error in the distance values. Bottom: error in the gradient directions. Around time = 2 h the door of the bedroom was opened.

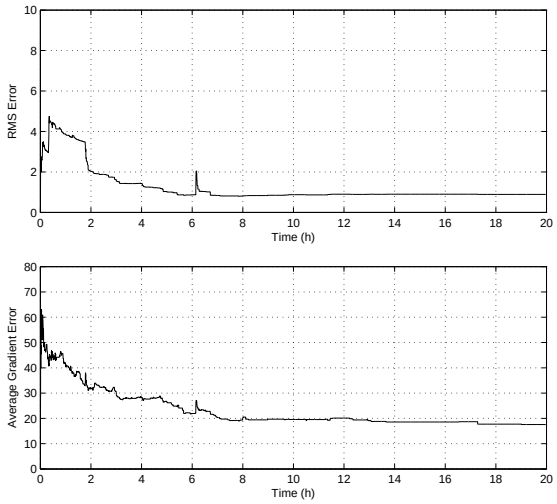

Fig. 9. Error functions during the build for the goal in the living room. Top: error in the distance values. Bottom: error in the gradient directions. Around time = 6 h the chairs in the kitchen were removed.

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

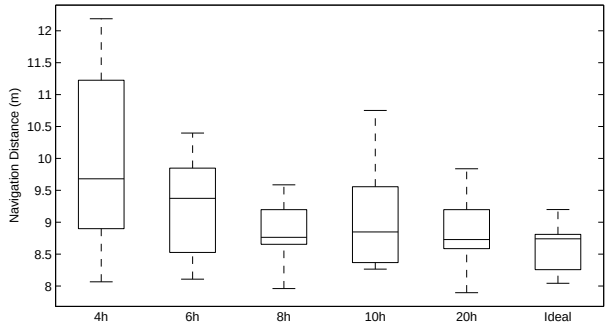

Fig. 10. Path length from ten runs of gradient descent on the bedroom goal at different stages of the build. Gradient descent on the corresponding ideal map (built off-line) has been included for reference.

the correct one. In this experiment, we also tested the ability of our algorithm to *adapt* an existing map to the removal of objects. To show this, we allowed the building to go on for six hours without any change in the environment, at which point we removed the chairs under the kitchen table. The second map in Fig. 7 shows the situation before the removal of the chairs. Notice how the final map adapted to this change. This is also detectable in Fig. 9; notice the sudden rise at $t = 6$ h, and how the curve thereafter stabilizes.

D. Navigation on the Built Distance Map

Our final suite of experiments was aimed at testing the performance of the gradient descent navigation system when navigating on the maps built on-line, as opposed to the ideal one. The goal was to show that navigation on the built maps is possible, and to evaluate how the amount of convergence affects the quality of navigation.

The experiment was conducted as follows: we used the bedroom build described earlier, and made gradient descent runs at different stages of the build. Navigation took place under the same conditions described under *Single Start, Single Goal*, and therefore the results shown in Table I can be used as a baseline to determine the quality of navigation. Navigation was performed on the map built after four, six, eight, ten and twenty hours. For each map, ten navigation runs were performed from the same starting position.

The results of the experiment are shown in Fig. 10 in the form of a box plot for the path length. The bottom line, middle line, and top line of a box corresponds to the lower quartile, median, and the upper quartile, respectively. The most extreme values within 1.5 times the interquartile range are connected to the box by dashes lines. These results show that the robot can successfully reach the goal also on a partially built map, and that the navigation performance increases as the map building time increases, and it eventually approaches the one on the ideal map.

VI. Conclusions

To the best of our knowledge, this paper is the first one to propose a stigmergetic approach to physical robot navigation in which the robot creates and exploits a navigation map which is stored in a passive environment.

Stigmergy brings a number of advantages. Since the map is not stored in the robot but in the floor, the building algorithm can be suspended and resumed without saving any state in the robot, and it can be continued by a different robot. It can also be carried out by several robots at the same time, which cooperate implicitly through the values in the floor. A robot does not need to be initialized or localized to navigate to the goal associated to the map: as soon as is placed on the floor, it will "descend" to that goal. Finally, the robot only needs to have minimal computation and sensing resources.

A number of interesting challenges remain for future research. Among these, we mention: how to deal with the case of multiple robot that are not homogeneous; how to deal with changes in the environment that reduce traversability; and how to extend the approach to the case in which the tags are not layed down in a regular grid.

A final remark concerns the applicability of the proposed approach to practical problems. RFID tags have very low cost, and deploying an RFID tag grid in an indoor environment is not complex since tags can easily be buried under carpet of wooden floors. Some companies are already selling or planning to sell RFID floors [26] and carpet tiles [27]. An infrastructure of this type would allow reliable navigation of both complex and very simple robots. The anytime property of the map building algorithm can help to make map learning effortless. For instance, a cleaning robot could run the map building algorithm while doing its standard cleaning task. If the robot operates for two hours a day, in a couple of weeks it would have created workable map and could start to use it for goal-directed navigation. The infrastructure would also be easily adapted to new goals or to changes in the environment layout, by storing a new map in a new field of the tags. These elements suggest that our approach has a favorable balance between costs and benefits for real applications.

Acknowledgments

This work was partially supported by the Swedish Knowledge Fundation. Our thanks to Mathias Broxvall, Federico Pecora and Per Sporrong for their help and suggestions.

References

[1] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
[2] S. Thrun, "Robotic mapping: a survey," in *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann, 2003, pp. 1–35.
[3] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II-state of the art," *Robotics and Automation Magazine*, vol. 13, pp. 108–117, 2006.
[4] A. Lilienthal and T. Duckett, "An absolute positioning system for 100 euros," in *Proc of the Int Workshop on Robotic Sensing (ROSE)*, 2003.
[5] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B. Seo, and Y. Cho, "The PEIS-ecology project: vision and results," in *Proc of the Int Conf on Intelligent Robots and Systems*, 2008.
[6] Wikipedia, "Stigmergy," http://en.wikipedia.org/wiki/Stigmergy.
[7] E. Sahin and W. Spears, Eds., *Swarm Robotics*, ser. LNCS. Springer, 2005, no. 3342.
[8] M. Dorigo, E. Bonabeau, G. Theraulaz, *et al.*, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 851–871, 2000.
[9] O. Holland and C. Melhuish, "Stigmergy, Self-Organization, and Sorting in Collective Robotics," *Artificial Life*, vol. 5, no. 2, pp. 173–202, 1999.

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.

[10] J. Werfel and R. Nagpal, "Extended stigmergy in collective construction," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 20–28, 2006.
[11] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Mit Press, 2004.
[12] O. Kubitz, M.O.Berger, M.Perlick, and R.Dumoulin, "Application of radio frequency identification devices to support navigation of autonomous mobile robots," in *Proc of the IEEE Vehicular Technology Conference*, 1997, pp. 126–130.
[13] S. Park, R. Saegusa, and S. Hashimoto, "Autonomous navigation of a mobile robot based on passive RFID," in *Proc of the Int Symp on Robot and Human interactive Communication*, 2007, pp. 218–223.
[14] K. Kodaka, H. Niwa, Y. Sakamoto, M. Otake, Y. Kanemori, and S. Sugano, "Pose estimation of a mobile robot on a lattice of RFID tags," in *Proc of the Int Conf on Intelligent Robots and Systems*, 2008, pp. 1385–1390.
[15] J. Bohn, "Prototypical implementation of location-aware services based on a middleware architecture for super-distributed RFID tag infrastructures," *Personal and Ubiquitous Computing*, vol. 12, no. 2, pp. 155–166, 2008.
[16] T. Herianto, K. Tomoaki, and K. Daisuke, "Realization of a pheromone potential field for autonomous navigation by radio frequency identification," *Advanced Robotics*, vol. 22, pp. 1461–1478, 2008.
[17] M. Mamei and F. Zambonelli, "Pervasive pheromone-based interaction with RFID tags," *ACM Trans on Autonomous and Adaptive Systems*, vol. 2, no. 2, pp. 1556–4665, 2007.
[18] V. Ziparo, A. Kleiner, B. Nebel, and D. Nardi, "RFID-based exploration for large robot teams," in *Proc of the Int Conf on Robotics and Automation*, 2007, pp. 4606–4613.
[19] K. O'Hara, D. Walker, and T. Balch, "Physical path planning using a pervasive embedded network," *IEEE Tran on Robotics*, vol. 24, no. 3, pp. 741–746, 2008.
[20] D. Bertsekas, "A simple and fast label correcting algorithm for shortest paths," *Networks*, vol. 23, pp. 703–709, 1993.
[21] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice Hall, NJ, 1995, 2nd edition.
[22] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in *Proc IEEE Int Conf on Robotics and Automation*, 1989, pp. 2–7.
[23] D. Murray and C. Jennings, "Stereo vision based mapping and navigation for mobile robots," in *Proc of the IEEE Int Conf on Robotics and Automation*, 1997, pp. 1694–1699.
[24] A. Saffiotti and M. Broxvall, "PEIS ecologies: Ambient intelligence meets autonomous robotics," in *Proc of the Int Conf on Smart Objects and Ambient Intelligence*, Grenoble, France, 2005, pp. 275–280.
[25] K. Konolige *et al.*, "Centibots: Very large scale distributed robotic teams," in *Proc of the 9th Int Symp on Experimental Robotics*, 2005.
[26] FutureShape, "NaviFloor," www.future-shape.com/navifloor.html.
[27] C. Zah and S. Fezer, "Embedded sensor scout flooring system by interface flor," in *Proc of the IEEE Int Conf on Technologies for Practical Robot Applications*, 2008, pp. 106–110.

APPENDIX: PROOF OF THEOREM 1

Throughout this proof we implicitly rely on assumptions (1) of Section IV-C above. First we state, without proof, the following obvious properties.

Proposition 1: In Algorithm 2:

(a) The values $v_t(x)$ stored in each tag x are monotonically non-increasing with time.
(b) At the end of any cycle, the distance counter has the same value as the tag x under the reader.

The proof is in three steps: first, we show that the algorithm is sound, that it, it never assigns values to tags that underestimate the real distance. Then, we show that the algorithm is complete, that is, if a tag x has distance $d(x)$, the algorithm will eventually assign that value to that tag. Finally, we put these pieces together and show convergence.

Lemma 1 (Soundness): At the end of any cycle of Algorithm 2, the value $v_t(x)$ stored in any tag x is greater or equal to $d(x)$.

Proof: By induction on the number of cycles. At start the thesis is trivially implied by the required condition of the algorithm. Consider now cycle $n+1$. If the tag read is the same as in the previous cycle, then $X_{new} = \emptyset$ and the thesis is true since it was true at cycle n and no tag writing operation is performed at cycle $n+1$. Else, if $X_{new} \neq \emptyset$ and `distance_counter` $> v_t(x)$, then again no tag writing operation is performed and the thesis is true. Consider then the case $T_{new} \neq \emptyset$ and `distance_counter` $< v_t(x)$. This means that the robot has passed from one tag to a different one: we denote these two tags by x' and x, respectively. Because of the assumption that tags are never skipped, x' and x are adjacent. Hence, if $d(x)$ denotes the real distance value of tag x, then

$$d(x) \leq d(x') + 1\,. \tag{3}$$

By Proposition 1 (b), the value of the counter at the end of cycle n was $v_t(x')$, hence, by step 8, its current value is $v_t(x')+1$. By step 12, this value has been stored into tag x, that is, $v_t(x) = v_t(x') + 1$. By the inductive hypothesis,

$$d(x') \leq v_t(x')\,. \tag{4}$$

Putting together (3) and (4), we have $d(x) \leq v_t(x') + 1 = v_t(x)$. Since this holds for any x, we have proved our thesis. □

Definition 1: An exploration strategy is *complete* if it visits any arc infinitely often, that is, for any x_1, x_2 and N, it will eventually visit the arc (x_1, x_2) more than N times.

Lemma 2 (Completeness): If the exploration strategy is complete, then each tag x will eventually be assigned a value $v_t(x)$ which is less or equal to $d(x)$.

Proof: By induction on the distance from the goal tag x_0. The thesis is trivially true for any tag that has zero distance, since tag x_0 is assigned the value 0 at start. Consider then any tag x which is $n+1$ steps from x_0, that is, $d(x) = n+1$. Then, there must be a tag x' which is adjacent to x such that $d(x') = n$. By inductive hypothesis, there is a time t_1 such that $v_{t_1}(x') \leq n$. By completeness of the exploration strategy, there will be a time $t_2 \geq t_1$ at which the arc (x', x) is traversed. By virtue of Proposition 1 (b), the value of the counter before the arc is traversed will be $v_{t_2}(x')$, and by Proposition 1 (a) (monotonicity) $v_{t_2}(x') \leq n$. After the arc is traversed, then, the value of the counter will be $v_{t_2}(x') + 1$, and because of steps 9 and 12, we have $v_{t_2}(x) \leq v_{t_2}(x') + 1 \leq n + 1 = d(x)$. Since this does not depend on the choice of x, we have proved our thesis. □

We are now in a position to prove convergence.

Theorem 1: Under the assumptions in (1), there is a time T such that, for any $t > T$, $Err(t) = 0$.

Proof: By Lemma 2, for each tag x there is a time t_x so that $v_{t_x}(x) \leq d(x)$, and by monotonicity (Proposition 1 (a)) $v_t(x) \leq d(x)$ for any $t \geq t_x$. By Lemma 1, $v_t(x)$ cannot be smaller than $d(x)$, so it must be $v_t(x) = d(x)$ for any $t \geq t_x$. Let $T = \max_{x \in X} t_x$, where X denotes the set of all tags in the grid. Then, for any time $t > T$ we have $v_t(x) = d(x)$ for any $x \in X$. The thesis then follows immediately from the definition of the $Err(t)$ function in (2). □

Cite as: R. Johansson and A. Saffiotti. Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. *Proc of the IEEE Int Conf on Robotics and Automation* (ICRA), Koba, Japan, 2009.