

Cooperative Visual Mapping in a Heterogeneous Team of Mobile Robots

Marius Hofmeister, Marcel Kronfeld and Andreas Zell

Abstract—Mapping is regarded as one of the most fundamental tasks for mobile robots. In this work, we present an approach that enables multiple resource-limited mobile robots to cooperatively build an image-based map of the environment and to afterwards localize in it. To achieve this, we deploy a hierarchical team of mobile robots. A *parent robot* possesses state-of-the-art sensors, computation power and acts as a leader. It teleoperates small *child robots* within its line-of-sight. In contrast to other approaches and due to the cooperation among the robots, we can relax the requirement that every robot must be able to self-localize to take part in multi-robot mapping. Additionally, our algorithm ensures the mapping of the entire area in an efficient way, i.e., it fulfills the requirements of area coverage. To test our approach, extensive experiments have been performed both in simulation and real-world. In the latter case, a team of four heterogeneous mobile robots was deployed. Besides the successful cooperation in the robot team, localization results are presented to validate the applicability of the proposed mapping procedure.

I. INTRODUCTION

Small mobile robots in the size of approx. 10 cm have shown to be useful in many cases. In swarm robotics, simulation of biological principles as well as in education, minirobots have been successfully deployed. Yet, this type of robots often provides limited computation power and restricted sensing capabilities. This is where a heterogeneous team of mobile robots becomes attractive: the individual strengths of the robots can lead to a better overall performance of robot teams. In this paper, we propose an algorithm that enables a number of small, resource-limited *child robots* to visually map an environment entirely in an efficient way. To this end, the child robots are assisted by a *parent robot* that possesses state-of-the-art computation power and accurate sensors. While the parent robot is able to robustly localize and navigate in an environment, the child robots can only act with limitations. Another advantage of this configuration is that small mobile robots are inexpensive and thus can be employed in larger numbers. To enable cooperation within the team, the robots have communication abilities and the parent robot is able to teleoperate the child robots within its line-of-sight. Similar team configurations were presented in [1].

For many robotic tasks, a model of the environment is a necessary prerequisite. Since cameras have become flexible, low-priced and lightweight devices, image-based mapping is an attractive alternative to mapping with range sensors,

M. Hofmeister and M. Kronfeld are Ph.D. students with the Chair of Computer Architecture, headed by Professor A. Zell, in the Department of Computer Science, University of Tübingen, Sand 1, 72076 Tübingen, Germany, {marius.hofmeister, marcel.kronfeld, andreas.zell}@uni-tuebingen.de

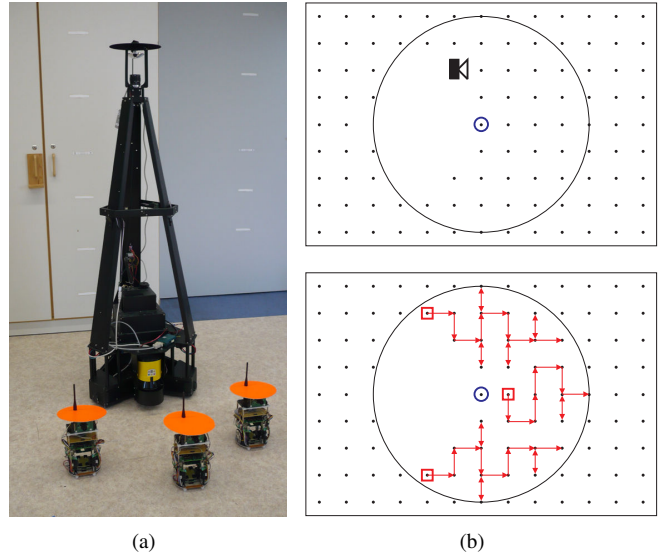


Fig. 1. (a) Heterogeneous team of mobile robots. (b) Mapping example within line-of-sight of the parent robot where the direction of the image grabbing is set east (right). In the upper image, vertices to be covered are depicted. Vertices that lie within line-of-sight of the parent robot but whose view is obstructed by the parent robot can not be covered in this step. The lower image depicts the navigation graphs of three child robots moving stepwise towards east. Rectangles depict the starting positions of the child robots and arrows their roadmap graphs. The position of the parent robot, i.e., the parent vertex, is depicted by a circle.

especially for robots with limited resources. In this work, we focus on the path planning and cooperation between the robots in the team while using a common mapping and localization method. In former research, we established and investigated visual self-localization for the type of resource-limited mobile robots [2]. In the mapping phase of our previous work, images were grabbed manually at grid points in indoor environments. From these *training images*, global image features were extracted and stored together with the position information in a database of features, similar to the work in [3]. In the retrieval phase, *test images* were grabbed by the robot and features were extracted again. By comparison to the features in the database and employing a particle filter, current poses could be estimated. Since self-localization generally requires that a place has been visited and mapped before, our aim is to exhaustively map the environment by multiple child robots in an efficient way. Since nowadays mapping with laser scanners is usually uncritical, we assume that a range map of the environment was created before, e.g., by the parent robot. Obviously, this range-sensor map can not serve the child robots to self-localize by means of their cameras. Our algorithm defines a grid of points from the map and ensures that at all of

these points an image is grabbed by a child robot and is sent to the parent robot. After mapping, the entire database of image features can be re-sent to the child robots that are subsequently able to perform self-localization with respect to this database. Accordingly, they can act fully autonomously, e.g., as mobile sensor nodes, and do not require the assistance of the parent robot anymore.

II. RELATED WORK

First steps into the direction of heterogeneous robot teams were achieved by Parker [4]. She demonstrated the ability of robots to compensate for heterogeneity in task allocation and execution. Grabowski et al. presented centimeter-scaled robots, called *millibots*, which were composed of various modular components and collaborated to explore and map unknown environments [5]. The design of autonomous behaviors for tightly-coupled cooperation in heterogeneous robot teams was presented in [6] as part of larger experiments by Howard et al. [1]. Their objective was to deploy a large number of robots as sensor nodes by fewer, more capable, robots and to map the interior of buildings to perform detection and tracking of intruders. Their team of robots is similar to ours. Yet, they focused on exploration and deployment instead of coverage and mapping.

Our approach is an instance of mapping with known poses. However, large parts of research focus on simultaneous localization and mapping (SLAM). A common way to solve this task is to jointly estimate the path of the robot and landmark positions. An example using a camera is the work by Karlsson et al. [7]. Alternative visual SLAM approaches determine only the path of the robot, based on the appearance of scenes [8].

In multi-robot SLAM, the goal is to build a global joint map by multiple robots and to localize the robots simultaneously in the map. In [9], Simmons et al. implemented an approach by using maximum likelihood to construct consistent maps with sensor data and odometry. Thrun and Liu solved the problem of unknown relative starting locations by using sparse information filters [10]. Howard et al. used a manifold map structure to provide self-consistency of the map [11] and later on extended a Rao-Blackwellized particle filter to tackle multi-robot SLAM problems [12]. However, all of the proposed algorithms are difficult to realize with resource-limited robots. Building maps, exchanging them between the robots and finding correspondences between the maps become difficult tasks if very limited computation power is available. To simultaneously cover the room in an efficient manner further complicates the problem, particularly due to imprecise movements of the robots. Reasons are noisy odometry and wheel slip that appears more often with lightweight robots. In this work, we overcome these mentioned challenges by supporting the minirobots during the mapping phase in a heterogeneous team of robots.

Since our work is also related to multi-robot area coverage, further classification is necessary. In [13], Choset distinguished between *off-line* algorithms, in which a map of the work area is given *a priori*, and *on-line* algorithms,

in which no map is given. Furthermore, he distinguished between *approximate cellular decomposition* and *exact cellular decomposition*. In the first case, free space is represented as a fine grid, where all cells are of the same size and shape, but the union of the cells only approximates the target region. In the second case, the union of all cells fills the entire target region. In this work, we focus on off-line, approximate cellular decomposition coverage as the basis for the mapping. While our previous work [14] built on the coverage algorithms of Hazon and Kaminka [15] to compute the robots' paths as efficiently as possible, we now also have to consider that we grab images in a specified direction only to limit possible viewpoints and thus facilitate self-localization. This comes with the constraint that no robot must stand in the field of view of another robot at the instance an image is grabbed.

III. HETEROGENEOUS TEAM OF ROBOTS

Our heterogeneous team of mobile robots is presented in Fig. 1. The parent robot is a custom-built service robot with a height of approx. 1.5 m. It is equipped with a laser scanner and an omnidirectional vision system with 780×580 pixels. The child robots are covered with orange hats which are detected and tracked by the blob tracking facilities of the *OpenCV* (<http://opencv.willowgarage.com/>) framework that was extended by a color segmentation module. The tracking frequency is 23 Hz and the measured accuracy detecting the child robot's real position within a line-of-sight radius of 2.0 m is 0.04 ± 0.02 m.

The child robots, named *c't-Bots* (<http://www.ct-bot.de>), have a diameter of 12 cm. They possess only the restricted computation power of an ATmega644 microcontroller with 64 KB flash program memory, a clock frequency of 16 MHz and 4 KB SRAM. The most capable sensor is a *POB-Eye* color camera that includes an image processing module. This module permits to perform image processing directly on it and to send the extracted image features to the robot via I²C. The camera provides a resolution of 120×88 pixels and possesses an ARM7TDMI processor at 60 MHz with 64 KB RAM. To store the image features on the robot, an SD card reader is connected. We additionally use a Devantech CMPS03 compass with a specified accuracy of $3\text{-}4^\circ$. The compass allows building a sparse environmental map where all images are taken looking in the same direction. To send and receive data, the robot is equipped with a WLAN interface. In the mapping step of our work, the child robots are teleoperated by the parent robot that sends out steering commands and receives the current headings of the child robots. When images are grabbed, they are sent directly to the parent robot which stores them.

IV. PATH PLANNING

A. Overview of the Mapping Approach

As mentioned before, the child robots lack reliable sensors and processing power to efficiently map the environment autonomously. By contrast, the parent robot can localize and navigate in a robust way. To take advantage of this

heterogeneity, we propose a teleoperation of the child robots within line-of-sight of the parent robot. Figure 1(b) depicts the basic approach that is structured as follows.

- 1) From the 2D occupancy grid map which was built using the laser scanner, vertices are determined that have to be visited by at least one child robot. At each of the vertices, an image has to be grabbed to establish self-localization capabilities.
- 2) A *parent roadmap graph* for the parent robot is built with *parent vertices* and *parent edges* such that all vertices lie in line-of-sight of at least one parent vertex. Since the images are grabbed in one single direction, we assign only vertices with unobstructed view, i.e., their view is not obstructed by the parent robot standing on the parent vertex. The parent edges determine the order of parent vertices that will be visited.
- 3) For each parent vertex, *child roadmap graphs* for all child robots are determined. All vertices that lie in line-of-sight of a parent vertex will thus be visited, and images will be grabbed by the child robots at these points.
- 4) When the mapping around a parent vertex is finished, the robots move to the next one, until all parent vertices have been visited. This implies that all vertices from step (1) have been covered.

We assume all robots to be able to move in the four directions up, down, left, right and we assume every vertex to be accessible in this way. In the parent and child roadmap graphs, each vertex is connected by n edges with $1 \leq n \leq 4$. Furthermore, the graphs are entirely connected. We now formalize the procedure.

Let there be one parent robot and N child robots. From the given map, a grid of vertices $V = \{v^{(i)} | i = 1, \dots, P\}$ to be visited by the child robots is extracted. The distance d between the vertices is constant. Only vertices are chosen that lie in the center of a free cell of size $d \times d$.

A parent roadmap graph $G_p = (V_p, E_p)$ for the parent robot is defined, which consists of Q parent vertices $V_p = \{v_p^{(t)} | t = 1, \dots, Q\}$ and $Q - 1$ directed edges $E_p = \{(v_p^{(t)}, v_p^{(t+1)}) | t = 1, \dots, Q - 1\}$ with $V_p \subseteq V$.

For a vertex v we further define the line-of-sight of the parent robot:

$$LOS(v) := \{w | w \in V \text{ is in line-of-sight from } v\} \quad (1)$$

and

$$LOS_{free}(v) := \{w | w \in LOS(v) \text{ and } w \text{ has unobstructed view towards the direction of mapping.}\} \quad (2)$$

Note that $v \notin LOS(v)$. For every parent vertex, a subset $C^{(t)}$ consisting of *child vertices* v_c can be computed, where

$$C^{(t)} = \left\{ v_c \in V | v_c \in LOS_{free}(v_p^{(t)}), v_c \notin \bigcup_{k=1}^{t-1} C^{(k)} \right\}. \quad (3)$$

Note that the child robots do not visit V_p for reasons of simplicity, since our localization method is robust enough to handle these few missing reference positions.

Assuming that $\forall v \in V \exists v_p : v \in LOS_{free}(v_p)$, this guarantees that

$$\bigcup_t C^{(t)} = V \setminus V_p. \quad (4)$$

For every $v_p^{(t)} \in V_p$ and for each child robot $j = 1, \dots, N$ the child roadmap graphs $G_c^{(j)} = (V_c^{(j)}, E_c^{(j)})$ are constructed such that $C^{(t)}$ is covered.

After all vertices in $C^{(t)}$ have been visited, the parent robot moves to $v_p^{(t+1)}$, where the new child roadmap graphs are computed as above.

Recalling our assumption that all $v \in V$ lie in line-of-sight of at least one $v_p \in V_p$ and that all $v \in C^{(t)}$ for all subsets $C^{(t)}$ are visited by the child roadmap graphs, it follows that $V \setminus V_p$ will be covered entirely.

B. Construction of a Parent Roadmap Graph

To construct G_p , we first need to find the parent vertices V_p to fulfill Eq. 4. Therefore, we pursue a greedy algorithm on the set V that chooses vertices for V_p according to a weighting method. For every vertex, a weight is computed that takes into account the number of vertices in line-of-sight and its corresponding number of neighbor vertices (the eight vertices lying in a square around it). Vertices with few neighbors receive larger weights to ensure the visit of vertices at borders of the area. The objective is to minimize $Q = |V_p|$. The algorithm subsequently covers the neighborhood of already chosen parent vertices without creating gaps.

For all $v \in V$, the following steps are performed:

- 1) The set $LOS_{free}(v)$ is computed.
- 2) The number of neighbor vertices $n(v)$ is computed, where $1 \leq n(v) \leq 8$.
- 3) A specific weight $w_s(v) = (9 - n(v))^2$ is assigned to v .

In a second iteration, a total score

$$w(v) = \sum_{l \in LOS_{free}(v)} w_s(l) \quad (5)$$

is computed for all $v \in V$.

The set V_p is then selected as follows. First, let $V_p := \emptyset$ and $M := \emptyset$. While $M \cup V_p \neq V$, the following steps are performed:

- 1) We first define a set of potential next vertices

$$N(V_p) := \{v \in V \setminus V_p | \forall v_p \in V_p : \|v - v_p\| \leq f\}, \quad (6)$$

where f is approx. twice the maximal line-of-sight radius of the parent robot to cover the area continuously. The vertex $v_p \in N(V_p)$ is then picked ensuring

$$\forall v' \in N(V_p) : w(v_p) \geq w(v'). \quad (7)$$

- 2) $V_p := V_p \cup \{v_p\}$
- 3) $M := M \cup LOS_{free}(v)$.
- 4) For all $v \in V$, $w(v)$ is updated.

After the set V_p is completed, the determination of the order in which the vertices are visited can be interpreted as a traveling salesman problem. A standard heuristic such as a nearest neighbour algorithm can be used to compute the paths E_p . Since the Euclidean distance is not feasible in obstructed environments, we use the distance transform algorithm [16] to compute the distance between every pair of $v_p^{(t)}$, which provides the shortest path, ordering V_p into the sequence $v_p^{(1)} \dots v_p^{(Q)}$.

C. Construction of Child Roadmap Graphs

After G_p has been created, for all $v_p^{(t)} \in V_p$ and for each child robot $j = 1, \dots, N$, the roadmap graphs $G_c^{(t,j)} = (V_c^{(t,j)}, E_c^{(t,j)})$ are constructed such that all child vertices $C^{(t)}$ will be visited.

The construction of child roadmap graphs works as follows. Starting points for the construction of child roadmap graphs are chosen reverse to the direction of mapping while it is ensured that no starting point obstructs the view of another one. For each starting point, a goal point is chosen that lies in the direction of mapping, maximizing the distance between the goal points. A path is planned towards this goal point as in [16] (see Fig. 1 (b)).

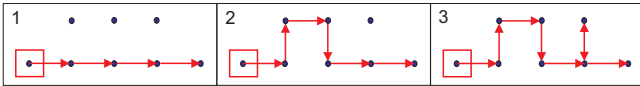


Fig. 2. Extending the original path (1) using the hilling procedure (2) and one branch out (3). Arrows depict the roadmap graph of the child robot. The starting point of the child robot is displayed as a square.

Then, the graphs are extended stepwise in parallel until all child vertices are covered (see Fig. 2):

- If there are two not yet visited neighboring vertices, *hilling* is performed, i.e., the path is winded.
- If hilling cannot further be done, the path will be branched out.

This procedure is similar to the approach in [15]. After all vertices have been visited, the subset H is determined to switch from stage t to $t + 1$:

$$H := LOS(v_p^{(t)}) \cap LOS(v_p^{(t+1)}) \quad (8)$$

If $|H| \geq N$, the child robots will move to a vertex from H and the parent robot afterwards moves to $v_p^{(t+1)}$. Otherwise, a following behavior will be performed as explained in the next section.

D. Following Behavior

The following behavior is performed during transition between the mapping phases t and $t+1$ if the line-of-sight of subsequent parent vertices do not overlap (or only slightly). In this case, the child robots move towards the parent robot to their individual following positions. Then, the parent robot moves to $v_p^{(t+1)}$, where the mapping procedure is restarted. To have the child robots follow the parent robot, different methods were tested. A distributed chaining behavior where each child robot follows its predecessor is not always stable since the robots might cut corners, overtake their leading robot or collide with objects. As in [6], we therefore decided to implement a behavior-based approach similar to [17]. As during the mapping, the parent robot tracks the position of the child robots and teleoperates them. By incorporating the difference between their current positions and their target positions, velocity vectors are computed.

V. EXPERIMENTAL RESULTS

A. Simulation Results

We conducted various simulation experiments to test our approach. 2D grid maps of size 30m×30m were generated randomly, where the distance d between the grid vertices was set 0.5m. Then, we added obstacles of different sizes. To perform the greedy algorithm to construct G_p , we set $f = 2 \cdot r_{los}$, where r_{los} is the line-of-sight radius of the parent robot (see Eq. 7). To determine which vertices in line-of-sight of a parent vertex have unobstructed view towards the arbitrarily chosen direction of mapping (east), the opening angle of the child robots' cameras was set 80°.

Figure 3 depicts the time that was required to map differently obstructed environments with a varying number of child robots. The results are averaged over ten cycles on randomly generated, different maps. r_{los} was set 2.0m. The ratio of the obstructed area was set 0%, 20% and 40% of the map. One timestep comprises one step of a robot in the direction up, down, left or right. We further distinguish between *mapping timesteps* that are conducted by the child robots while grabbing images of the environment, and the remaining *coordination timesteps* that are required to move the robots from one parent vertex to the next, including the following behaviors of the robots. Table I shows the number of extracted parent vertices in the differently obstructed environments. Since at 0% obstruction the map is always equal, there is no standard deviation given for Q , but the number of executed following behaviors differs due to the different number of child robots.

TABLE I
NUMBER OF PARENT VERTICES Q AND NUMBER OF EXECUTED FOLLOWING BEHAVIORS AT DIFFERENTLY OBSTRUCTED MAPS

	0% obstr.	20% obstr.	40% obstr.
$ V_p = Q$	167.0	180.5 ± 4.1	174.9 ± 5.4
follow.	32.5 ± 12.3	54.8 ± 18.1	82.9 ± 24.9

Figure 3 shows that in obstructed environments the number of mapping timesteps decreases due to the fact that fewer

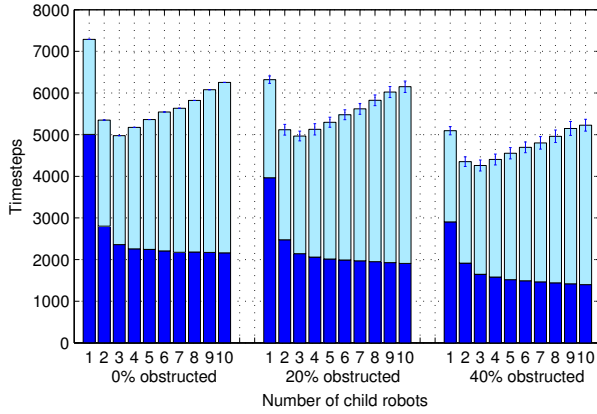


Fig. 3. Distribution of total timesteps for different environments. The lower parts of the boxes represent mapping timesteps, the upper part coordination timesteps.

vertices have to be visited. By deploying more child robots, the mapping timesteps generally decrease due to the parallel task execution. By contrast, the coordination timesteps grow since more robots require more time to be coordinated, e.g., to form the following behavior.

An important parameter for the mapping procedure is the line-of-sight radius of the parent robot r_{los} . Depending on the robot configuration, different line-of-sight radii are plausible. In Tab. II, the number of extracted parent vertices are shown at varying line-of-sight radii. Larger radii result in fewer parent vertices. In Fig. 4, the timesteps required at varying line-of-sight radii are depicted. By enlarging r_{los} , the overall mapping time reduces. A demanding factor corresponding to the overall mapping time is the coordinated following procedure. If child robots can be steered directly from the line-of-sight area of one parent vertex to the line-of-sight area of the next, this saves time. As the size of r_{los} increases, the possible overlap between the radii increases correspondingly. Consequently, the number of following behaviors decreases (see Tab. II) and thus the number of coordination timesteps does.

TABLE II
NUMBER OF PARENT VERTICES $Q = |V_p|$ AND NUMBER OF EXECUTED FOLLOWING BEHAVIORS AT DIFFERENT LINE-OF-SIGHT RADII

	1.0 m	r_{los} 2.0 m	4.0 m	8.0 m
$ V_p $	538.5 ± 5.4	180.5 ± 4.1	71.2 ± 4.1	42.0 ± 2.6
follow.	403.1 ± 154.7	54.8 ± 18.1	18.4 ± 5.6	9.5 ± 3.6

Noticeably, a larger line-of-sight radius also implies a drawback: The fragmentation of vertices gets larger, that is, during the mapping steps, vertices have to be crossed that have been visited in an earlier phase. Therefore, the mapping timesteps grow again at $r_{los} = 8.0$ m. If r_{los} is very small (1.0 m), multiple child robots can only slightly improve the overall mapping time because of high coordination costs. The minimal overall time is achieved by using 2 (3, 5, 7) child robots at a line-of-sight radius of 1.0 m (2.0 m, 4.0 m,

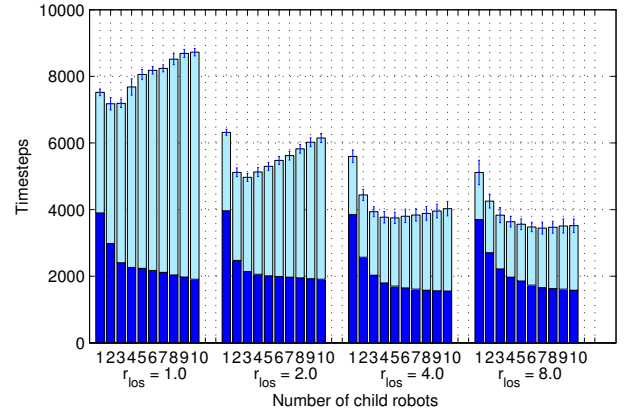


Fig. 4. Distribution of total timesteps at a map with 20% obstruction and different line-of-sight radii. The lower part of the boxes represent mapping timesteps, the upper part coordination timesteps.

8.0 m). A reasonable ratio between r_{los} and the number of child robots has to be chosen to achieve the minimal mapping time.

B. Real-World Results

We conducted real-world experiments in an indoor environment of approx. 32 m^2 (see Fig. 5). Our localization methods, however, were evaluated within a larger area of approx. 75 m^2 [2]. 84 grid vertices were extracted from the occupancy map. The parent robot was able to robustly detect the child robots within $r_{los} = 2.0$ m. The opening angle of the child robots' cameras was measured as 80° , the grid resolution d was set 0.5 m and the direction of mapping was arbitrarily chosen east.

The number of determined parent vertices was seven. Totally, we conducted the experiment ten times. The averaged time to map the entire environment was 38.4 ± 1.7 min. Since the child robots' movements are fairly imprecise, we steered the robots at 0.15 m per second only. The reason for this was the avoidance of child robot collisions and tracking losses which accordingly did not occur. To approach a vertex, we first let the child robots rotate towards the direction of the goal point by means of their compass and then perform a straight-line motion. Since compass data and robot rotations suffer from inaccuracies, we first let the child robots approach the goal point up to a distance of 0.2 m. If necessary, the child robots were then rotated again and steered straight-line directly towards the goal. This ensured reaching the vertices safely without missing the target points.

Figure 5(b) depicts the actual paths of the child robots measured by the tracking system of the parent robot. The movements of the child robots are fairly imprecise: the *position error* between the child robot's desired mapping positions and their measured positions was 0.10 ± 0.05 m. The *orientation error* between the robot's measured orientations and the desired mapping orientations was $8.79 \pm 9.99^\circ$.

To test the localization accuracy of the images, we performed cross-validation on the ten mapping datasets that consisted of 77 images each. For every image, the distance to

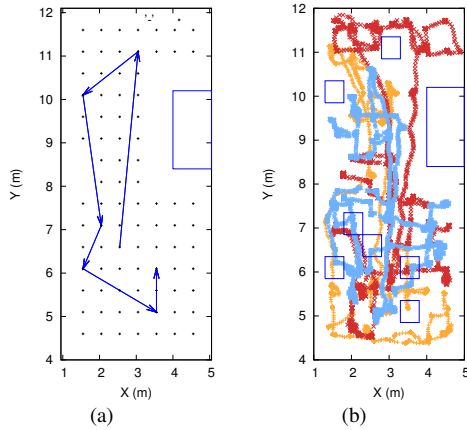


Fig. 5. (a) Grid of vertices and extracted parent vertices in the indoor test environment. Arrows depict parent edges. (b) Paths of the three child robots as determined by the tracking system of the parent robot. Rectangles depict the positions of the parent vertices.

the best match of the nine test datasets was determined. The best match was determined by extracting global features from the images and comparing these features. We decided to use *Weighted Gradient Orientation Histograms (WGOH)* at full image resolutions, since this feature led to best localization results in [2], and a minimalistic *pixelwise image comparison* on a reduced image size. For a detailed introduction to the methods, we refer the reader to [2]. In case of WGOH at images of 120×88 resolution, the mean localization error was 0.43 ± 0.59 m. In case of the pixelwise image comparison on 15×11 images, the corresponding error was 0.54 ± 0.79 m.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel cooperative mapping strategy for a heterogeneous team of mobile robots that includes entire coverage. Several contributions have been made in this work. First, we developed a robotic team structure in which a larger number of inexpensive and resource-limited robots are assisted by a more capable robot to conduct a task that they could hardly perform alone. In contrast to other approaches, our setup does not require all robots to possess accurate localization and navigation abilities. Second, we developed a technique which uniformly partitions the work area in order to save mapping time. Third, a path planning method for the small robots was described that ensures that no robot obstructs the view of another one at the instant of grabbing an image. Despite the restrictions of small robots, namely the relatively large rotation error and the fairly imprecise movements, we can conclude that a localization based on the recorded image database is possible at a reasonable accuracy. Our approach is able to handle the challenging restrictions of small mobile robots. However, this comes at the cost of a centralization in form of the parent robot: the system can hardly recover from a parent robot failure. For this reason, future work will include the deployment of multiple robot teams and a dynamic association of the child robots to the parent robots, possibly to further improve overall mapping time. Moreover,

the discretization of the world could be performed differently from a regular grid: vertices may be chosen depending on environmental properties that reveal the self-similarity of the surrounding. Finally, uncertainty concerning the child robots' positions could be taken into account in the future.

VII. ACKNOWLEDGMENT

The first author would like to acknowledge the financial support by the Friedrich-Ebert-Stiftung of his Ph.D. scholarship at the University of Tübingen.

REFERENCES

- [1] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *Int. Journal of Robotics Research*, vol. 25, no. 5, pp. 431–447, May 2006.
- [2] M. Hofmeister, P. Vorst, and A. Zell, "A comparison of efficient global image features for localizing small mobile robots," in *Proceedings of the joint conference of ISR 2010 and ROBOTIK 2010*. VDE Verlag, June 2010, pp. 143–150.
- [3] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image retrieval system with monte carlo localization," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 208–216, April 2005.
- [4] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multi-robot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, April 1998.
- [5] R. Grabowski, L. E. Navarro-Serment, C. Paredis, and P. Khosla, "Heterogeneous teams of modular robots for mapping and exploration," *Autonomous Robots - Special Issue on Heterogeneous Multirobot Systems*, vol. 8, no. 3, pp. 271–298, June 2000.
- [6] L. E. Parker, B. Kannan, F. Tang, and M. Bailey, "Tightly-coupled navigation assistance in heterogeneous multi-robot teams," in *Proceedings of IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, vol. 1, Sendai, Japan, September 2004, pp. 1016–1022.
- [7] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings of IEEE Int. Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005, pp. 24–29.
- [8] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The Int. Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [9] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. An Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI)*, Austin, USA, 2000.
- [10] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filters," in *Proceedings of the 11th Int. Symposium of Robotics Research (ISRR)*, Sienna, Italy, 2003.
- [11] A. Howard, G. S. Sukhatme, and M. J. Matari, "Multi-robot mapping using manifold representations," in *Proceedings of IEEE Int. Conference on Robotics and Automation (ICRA)*, New Orleans, USA, April 2004, pp. 4198–4203.
- [12] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.
- [13] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.
- [14] M. Hofmeister and M. Kronfeld, "Multi-robot coverage considering line-of-sight conditions," in *7th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, Lecce, Italy, 2010.
- [15] N. Hazon and G. A. Kaminka, "On redundancy, efficiency, and robustness in coverage for multiple robots," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1102–1114, December 2008.
- [16] R. Jarvis, "Robot path planning: Complexity, flexibility and application scope," in *Proceedings of the 2006 Int. Symposium on Practical Cognitive Agents and Robots (PCAR)*, Perth, Australia, November 2006, pp. 3–14.
- [17] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, December 1999.