Multi-robot Coverage Considering Line-of-sight Conditions

Marius Hofmeister and Marcel Kronfeld

Computer Science Department, University of Tübingen, Sand 1, 72076 Tübingen, Germany
(e-mail: {marius.hofmeister, marcel.kronfeld}@uni-tuebingen.de)

Abstract: In this paper, we present a novel approach to the area coverage problem by using a team of heterogeneous mobile robots. In our method, a *parent robot* is assumed to possess state-of-the-art sensors and sufficient computation power to establish robust localization and navigation. A large number of inexpensive and small *child robots* possess only restricted sensing and computation capabilities. They can only fulfill a certain task, e.g., floor-cleaning, but can be teleoperated in line-of-sight of the parent robot. To exploit the advantages of both types of robots, the team cooperatively covers the area in an efficient way.
In contrast to other approaches and due to the cooperation of the robots, we can relax the requirement that every robot must be able to self-localize and robustly navigate to take part in efficient multi-robot coverage. Simulation results are presented in which our approach was tested intensively.

Keywords: Mobile robots, heterogeneity, path planning.

1. INTRODUCTION

The diversity of mobile robots has increased significantly during the past decades. And since many complex applications have different requirements on sensors and robots, clearly not every robot is suited for every task. This is the main reason that makes teams of heterogeneous mobile robots attractive. The employment of different types of robots has the great advantage to exploit the robots' individual strengths.

In general, multi-robot systems are expected to have several advantages compared to single-robot systems (Cao et al. (1997); Arai et al. (2002)). They often provide improved time performance due to distributed parallel task execution and a higher degree of robustness due to a certain independence between task completion and the number of robots.

Area coverage is an essential task for mobile robots that finds its relevance in various applications, e.g., floor cleaning, de-mining or harvesting. Generally, in coverage, robots visit every point in a target area, at least once.

In this paper, we present a solution to the area coverage problem using a team of heterogeneous mobile robots. Specifically, we assume the employment of two types of robots: a *parent robot* with state-of-the-art computation power and capable sensors, as well as multiple smaller and more flexible *child robots* with restricted sensing and computational capabilities. While the parent robot is able to robustly localize and navigate in an environment, the child robots can only perform a limited task, e.g., cleaning, and do not have the ability to localize themselves. This configuration exploits the fact that small mobile robots are less expensive and thus can be employed in larger numbers. We assume that the parent robot can detect the child robot's positions and is able to teleoperate them. Similar robot configurations were presented by Howard et al. (2006).

2. RELATED WORK

One of the earliest research demonstrations of heterogenity in mobile robot teams was by Parker (1998). She demonstrated the ability of robots to compensate for heterogenity in task allocation and execution. Grabowski et al. (2000) presented centimeter-scaled robots, called *millibots*, which are configured by various modular components and collaborate to explore and map unknown environment. The design of autonomous behaviours for tightly-coupled cooperation in heterogeneous robot teams was presented by Parker et al. (2004) and was part of larger experiments by Howard et al. (2006). The objective of their work was to deploy a large number of robots as sensor nodes into an unexplored building by fewer, more capable, robots and to map the building's interior to perform detection and tracking of intruders. Their robot configuration is similar to ours. However, they focused rather on exploration and deployment than on area coverage.

The multi-robot coverage problem has achieved large interest in recent years due to the enhanced robustness that comes with the use of multiple robots and the higher productivity that the parallelization of sub-tasks provide. A general survey of coverage algorithms was published by Choset (2001). He 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

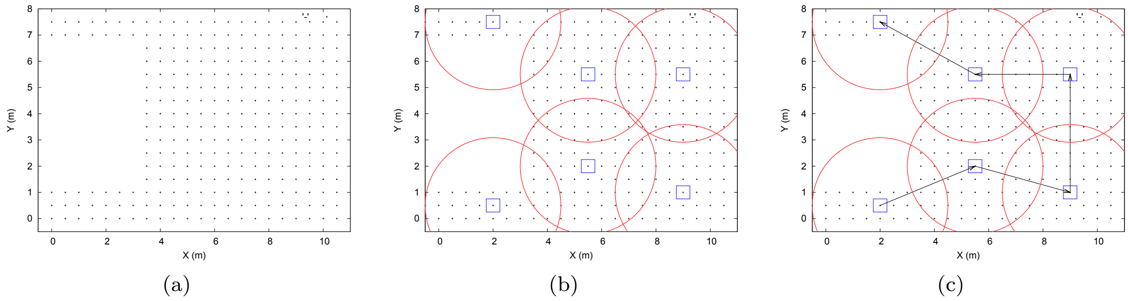

Fig. 1. Basic coverage approach. (a) depicts the vertices to be covered. In (b), parent vertices, depicted by small rectangles, are chosen. The order in which the parent vertices are visited is determined by parent edges (c).

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.

Our approach partly builds on the work of Hazon and Kaminka (2008). They extended the single-robot off-line spanning-tree coverage algorithm by Gabriely and Rimon (2001) to the use of multiple robots and addressed robustness and efficiency. Beyond the assumption that elimination of redundancy leads to improved efficiency and thus shorter coverage time, they analytically showed the robustness of their methods, as long as one robot is still able to move.

Various other approaches to the coverage problem were proposed. Rekleitis et al. (2004) built on the Boustrophedon decomposition, that is, an exact cellular decomposition where each cell can be covered with back and forth motions. Their algorithm operates under the limitation that communication between robots is available only when they are within line-of-sight of each other. Futhermore, biological approaches were proposed, e.g., by Wagner et al. (1999), who considered robots to leave chemical odor traces inspired by the pheromones of ants. In contrast to our approach, all of the presented coverage algorithms require that the employed robots are able to fully self-localize in the environment.

3. COVERAGE APPROACH

3.1 Overview

In the following, a general overview of the covering approach is given. It was mentioned above that the child robots are assumed to lack reliable sensors and processing power to autonomously localize themselves in the environment. In contrast, the parent robot can localize and navigate in a robust way. To support the child robots in this respect, they have communication abilities. Additionally, the parent robot is able to determine the global positions of the child robots and to teleoperate them within line-of-sight.

Our approach is structured as follows.

(1) From the given 2D-occupancy grid map, vertices to be covered by the child robots are determined.
(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. 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. In this way, vertices that lie in line-of-sight of a parent vertex will be covered.
(4) When the coverage around a parent vertex is finished, the robots cooperatively move to the next one, until all parent vertices have been visited. This implicates 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 $1 \leq n \leq 2$. Furthermore, the graphs are entirely connected. Figure 1 depicts the basic approach. We now formalize the procedure.

Let there be one parent robot and N child robots. From the given map, a grid of vertices $V = \left\{v^{(i)} | i = 1, ..., P\right\}$ to be covered by the child robots is extracted as described in Sect. 3.2.

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

We further define for a vertex v:

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

For every parent vertex, a subset $C^{(t)}$ consisting of *child vertices* v_c can be computed, where

$$C^{(t)} = \left\{ v_c \in V | v_c \in LOS(v_p^{(t)}), v_c \notin \bigcup_{k=1}^{t-1} C^{(k)} \right\}. \quad (2)$$

Assuming that $\forall v \in V \, \exists v_p : v \in LOS(v_p)$, this guarantees that

$$\dot{\bigcup_t} C^{(t)} = V. \quad (3)$$

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

If $t < Q$, the robots move together 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 will be covered entirely.

3.2 Extraction of the Grid of Vertices

Given the map of the environment, we extract a grid of vertices V that have to be covered. The distance d between the vertices is constant. Only vertices are chosen that lie in the center of a free cell of size $d \times d$. This procedure is equal to the construction of cells within free space of the room that is described in related work. Because of the construction of the roadmap graphs, we considered our formulation to be more practical in this context.

3.3 Construction of a Parent Roadmap Graph

To construct G_p, V_p have to be chosen from V to fulfill Eq. 3. 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 their corresponding number of direct neighbor vertices (that are the eight vertices laying in a square around it), while vertices with few neighbors are weighted higher. The objective is to minimize $Q = |V_p|$. The algorithm starts ensuring the visit of vertices at borders of the area and then subsequently covers the neighborhood of already chosen parent vertices.

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

(1) The set $LOS(v)$ is computed.
(2) The number of direct neighbor vertices $n(v)$ is computed, where $1 \leq n(v) \leq 8$.
(3) A specific weight $w_n(v) = (n(v) - 9)^2$ is assigned to v.

After this, for all $v \in V$, a total weight

$$w(v) = \sum_{l \in LOS(v)} w_n(l) \tag{4}$$

is computed.

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

(1) We first define a set of neighboring nodes

$$N(V_p) := \{v \in V \setminus V_p | \forall v_p \in V_p : ||v - v_p|| \leq f\}, \tag{5}$$

where f is approx. twice the maximal line-of-sight radius of the parent robot. The vertex $v \in N(V_p)$ is then picked, ensuring

$$\forall\, v' \in N(V_p) : w(v) \geq w(v'). \tag{6}$$

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

After the set V_p is built, the order in which the vertices are visited can be interpreted as a traveling salesman problem. A standard heuristic can be used to approximate this problem and compute E_p. Since the Euclidean distance is not feasible in obstructed environments, we here use the distance transform algorithm by Jarvis (2006) to compute the distance between every pair of $v_p^{(t)}$, which provides the shortest path. Generally, this approach is related to *sampling-based planning* methods, such as the probabilistic roadmap method (PRM) by Kavraki et al. (1996), where the free space of a room is sampled and a graph search can be performed to plan a path.

3.4 Construction of Child Roadmap Graphs

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

To achieve efficient coverage paths for the child robots, we extend the spanning-tree based coverage method by Hazon and Kaminka (2008). Their algorithm creates spanning trees on a coarse grid of cells in the work-area while considering the initial position of the robots. A *coarse grid cell* contains four child vertices that lie in a square. The challenge is to construct a spanning tree that minimizes the time to complete coverage. After the tree is built, the robots follow it around, creating a Hamiltonian cycle visiting all cells of the coarse grid. Since our aim is to implement this algorithm in real-world experiments, we found it necessary to enhance the algorithm to also work with cells that do not lie in a coarse grid since this restricts the approach a lot. Figure 2 illustrates the original procedure and our extensions.

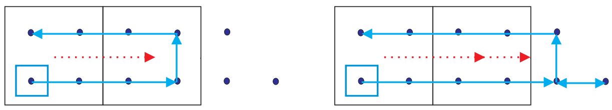

Fig. 2. Creating a spanning tree in the original (left) and the extended version (right). Dotted arrows depict the tree that connects two coarse grid cells. Continuous arrows depict the final roadmap graph of the child robot (displayed as small square). The growing of the tree has started in the coarse grid cell of the child robot. After the coarse grid cells have been connected by the tree, three vertices remain unvisited in the original version. In the extended version, an additional branch out of the tree was done. After that, one vertex on the right is still not nearby the tree. It can be accessed, but this requires the child robot to visit the vertex before twice and thus perform a *redundant timestep* to return.

The creation of N roadmap graphs works as follows. First, a coarse grid is created over $C^{(t)}$. Then, the child robots are moved to the nearest free coarse grid cell that becomes the starting point for N separate subtrees. These subtrees grow independently for every child robot and will be connected later to a single spanning tree.

To create the subtrees, for every child robot the following steps are performed stepwise in parallel:

- For each not visited neighboring coarse grid cell (up, down, right, left), the Manhattan distance is computed from the current location of all other child robots.

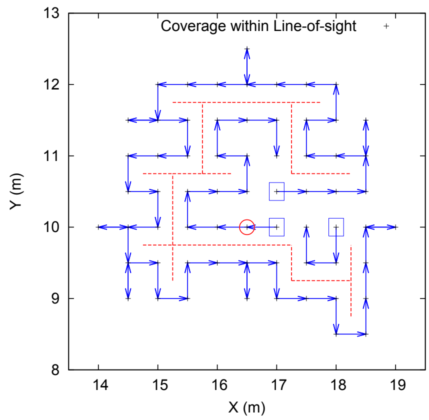

Fig. 3. Coverage example within line-of-sight of the parent robot. Small rectangles depict the starting positions of three child robots and arrows their roadmap graphs. Dotted lines denote the created spanning tree. The position of the parent robot is depicted by a small circle.

- The coarse grid cell with the maximal minimal distance to any other child robot is chosen.
- If there is no neighboring unvisited coarse grid cell left at the end of the tree, the tree is branched out from the main branch to all possible directions.
- If the tree is totally branched out on the coarse grid, it is further branched out on the original grid where possible.

Afterwards, the N subtrees are connected to one entire spanning tree by choosing connections that aim at partitioning the paths for the child robots equally to maximize parallel coverage. For a more detailed description of this, we are referring to the work of Hazon and Kaminka (2008).

By cycling the robots clockwise or counter-clockwise around this single spanning tree, we then create their roadmap graphs $G_c^{(j)}$. Different methods were proposed by Hazon and Kaminka (2008), where robots can follow the tree in one direction or backtrack to the other direction, if this saves coverage time. Here, we use the simple *non-backtracking multi-robot spanning-tree coverage (NB_MSTC)*, in which the robots start at their current positions and cycle the tree until they reach the starting position of the next robot.

Figure 3 depicts example child roadmap graphs that encircle a coverage spanning tree. To cover the parent vertex with a child robot, we assume that the parent robot leaves its vertex for at most one timestep in a random direction (not necessarily on a child vertex) and then returns. This appears reasonable due to narrow environments where we can not expect to have enough child vertices around the parent robot that allow the child robots for circumnavigating the parent robot.

3.5 Coordinated Following

After a parent vertex $v_p^{(t)}$ was visited and $C^{(t)}$ was covered, the team of robots has to move cooperatively to the next parent vertex. This can be formulated as a formation control task that requires the robots to move together while maintaining certain relative positions. Since the coverage algorithm should work in every area, we must also expect narrow passages that allow only one robot at a time to pass through. Therefore, we use a chain of robots and, e.g., a leader-follower formation control. In this way, all $v_p^{(t)}$ in V_p can be visited.

Fig. 4. Team of heterogeneous mobile robots.

4. HETEROGENEOUS ROBOTS

Figure 4 shows our team of heterogeneous mobile robots that served as an example for this paper and that will be used in future work. 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 that allows for detecting and tracking the red hats of the child robots. The 13 child robots of our lab are in the size of approx. 15 cm and were developed by the German computer magazine *c't* (http://www.ct-bot.de). They possess only the restricted computation power of a microcontroller and low-cost sensors.

5. SIMULATION RESULTS

To test the approach, we performed various simulation experiments. We randomly generated a 2D-grid map of size $30\,m \times 30\,m$ and set the distance d between the grid vertices 0.5 m. Then, we added obstacles in 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. In our team of heterogeneous robots, we found the parent robot to be able to detect the child robots within $r_{los} = 2.5\,m$, which we chose as the default value in our experiments. Figure 8 depicts an example map with the extracted parent vertices and edges.

Figure 5 depicts the timesteps to cover the entire environment with a varying number of child robots in different environments. r_{los} was set 2.5 m. The ratio of the obstructed area was set 0 %, 20 % and 35 % of the map. One timestep comprises one step of a robot in the direction up, down, left or right. Figure 5 shows that the minimal coverage time in the obstructed environments can be achieved by employing five child robots. This is because in coordinated following, the child robots need to keep a certain distance to their leader that allows for robustly tracking the robots. Based on our team of heterogeneous robots, we set this following distance d. When reaching a parent vertex and $r_{los} = 2.5\,m$, five child robots stand within line-of-sight from the parent robot. Any more child robots first have to

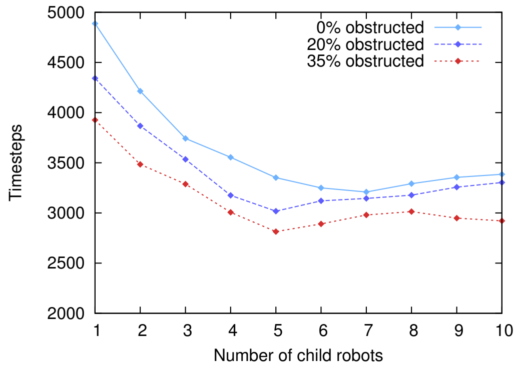

Fig. 5. Required timesteps for the total coverage of different environments with a varying number of robots.

be steered into line-of-sight which requires more timesteps. We furthermore observe that the coverage time decreases in obstructed environments. Table 1 shows the number of vertices and parent vertices in our environments. By increasing the ratio of the obstacles, the number of parent vertices is also increasing. This is because the obstacles obstruct direct views and more parent vertices are required to keep all child vertices in line-of-sight. Despite the higher number of parent vertices, the coverage time on the obstructed maps is shorter due to the lower number of child vertices that have to be covered.

	0 % obstr.	20 % obstr.	35 % obstr.
$\lvert V \rvert = P$	3600	2858	2317
$\lvert V_p \rvert = Q$	71	78	86

Table 1. Number of vertices P and parent vertices Q at differently obstructed maps.

To further investigate these results, we distinguish between *coverage timesteps* that are performed by the child robots around the parent vertices, and *coordination timesteps* that are necessary to guide the child robots to the parent robot and to move together in a group. Figure 6 depicts how these two phases are distributed. As expected, the coordination timesteps increase with the number of child robots that have to be organized.

An essential parameter for performing the coverage in the team of heterogeneous robots is the maximal line-of-sight radius r_{los} of the parent robot. Figure 7 depicts the necessary timesteps to cover the environment for different values of r_{los}. By increasing r_{los}, the number of parent vertices decreases (as shown in Tab. 2), and so is the number of coordination timesteps. The effect of the parameters r_{los} and d, requiring that any more than $\frac{r_{los}}{d}$ child robots first have to be steered into line-of-sight of the parent robot, is also apparent when varying r_{los}. For $r_{los} = 1.0\,m$ ($2.5\,m$), more than 2 (5) robots cannot further improve the coverage time. In case of $r_{los} = 7.5\,m$, we get a minimal coverage time by using 9 child robots. Conclusively the larger the value of r_{los}, the more useful is the employment of a larger number of child robots. As an exception, the coverage time for one single child robot at $r_{los} = 7.5\,m$ is larger than at $r_{los} = 2.5\,m$ because of a higher fragmentation of the child vertices within the larger line-of-sight radius.

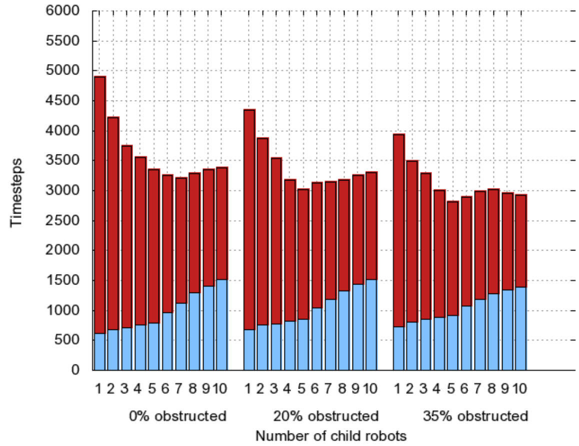

Fig. 6. Distribution of total timesteps for different environments. The lower parts of the boxes represent coordination timesteps, the upper part coverage timesteps.

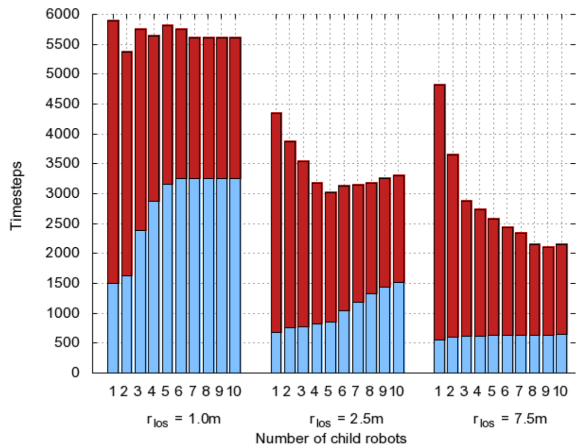

Fig. 7. Distribution of total timesteps at a map with 20% obstruction and different line-of-sight radii. The lower part of the boxes represent coordination timesteps, the upper part coverage timesteps.

As mentioned in Sect. 3.4, we extended the approach of Hazon and Kaminka (2008) by enabling the child robots to visit even vertices that are not part of the coarse grid. This is important for real-world tasks, but comes at the expense that the robots have to visit certain vertices twice. To determine the influence of this extension, we counted the number of these *redundant timesteps* in Tab. 2.

	r_{los}		
	$1.0\,m$	$2.5\,m$	$7.5\,m$
$\lvert V_p \rvert = Q$	364	78	24
Redundant Timesteps	20.11 %	12.06 %	10.68 %

Table 2. Number of parent vertices Q and percentage of redundant timesteps at different line-of-sight radii.

By increasing r_{los}, Q is decreasing, while the percentage of redundant timesteps is also decreasing. Most redundant timesteps are required at the border of the line-of-sight radius. Thus, fewer parent vertices imply fewer redundant timesteps.

	$r_{los} = 1.0\,m$	$r_{los} = 2.5\,m$	$r_{los} = 7.5\,m$
	10.43 min	10.76 min	148.1 min

Table 3. Computation times of the coverage process at different line-of-sight radii.

In Tab. 3, we measured the computation times of the entire coverage process for different line-of-sight radii on an AMD Athlon Dual Core with 1 GHz and 2 GB RAM on Scientific Linux 5.0. When $r_{los} = 7.5\,m$, the computationally most demanding factor is the geometrical determination of vertices that lay within the enhanced line-of-sight radius of the parent robot. We recall that we do not require online processing. In tasks that require the robots to cover the area multiple times, no new computation has to be performed.

Finally, it can be stated that a larger number of child robots improves the coverage time significantly. However, the optimal coverage time $\frac{P}{N}$ that could theoretically be obtained in the coverage timesteps can hardly be achieved since it is largely affected by the starting positions of the child robots that are in most real-world scenarios nearby each other.

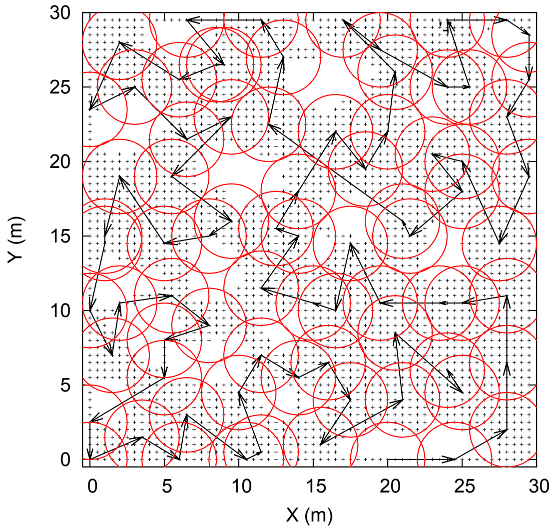

Fig. 8. Coverage example on a $30\,m \times 30\,m$ map, obstructed by 20% obstacles. Circles depict the line-of-sight radii ($r_{los} = 2.5\,m$) of the parent vertices that are set by the greedy algorithm. Black lines depict parent edges.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach to the area coverage task considering line-of-sight conditions. To the best of our knowledge, this is the first work that directly aims at deploying a team of heterogeneous mobile robots in coverage. Due to the cooperative behaviour of our robots, we could relax the requirement that all robots are able to localize within the environment. We showed that, by using a larger number of inexpensive and small child robots with restricted sensing and computational capabilities, area coverage can be performed in a fast and efficient way.

We furthermore investigated to what extent the line-of-sight radius of the parent robot influences the coverage time. It resulted that a larger number of child robots leads to a shorter coverage time as long as a reasonable ratio between the line-of-sight radius of the parent robot and the number of child robots was chosen. A further increase of the number of child robots does not necessarily lead to a shorter coverage time.

Interesting extensions to our work might be the deployment of multiple teams of mobile robots and a dynamic association of the child robots to further optimize the coverage time. Based on the results of this work, we will implement and investigate our method in our team of heterogeneous mobile robots.

REFERENCES

Arai, T., Pagello, E., and Parker, L.E. (2002). Guest editorial: Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5), 655–661.

Cao, Y.U., Fukunaga, A.S., and Kahng, A.B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4, 1–23.

Choset, H. (2001). Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4), 113–126.

Gabriely, Y. and Rimon, E. (2001). Spanning-tree based coverage of continuous areas by a mobile robot. In *Annals of Mathematics and Artificial Intelligence*, volume 31, 77–98.

Grabowski, R., Navarro-Serment, L.E., Paredis, C., and Khosla, P. (2000). Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots - Special Issue on Heterogeneous Multirobot Systems*, 8(3), 271–298.

Hazon, N. and Kaminka, G.A. (2008). On redundancy, efficiency, and robustness in coverage for multiple robots. *Robotics and Autonomous Systems*, 56(12), 1102–1114.

Howard, A., Parker, L.E., and Sukhatme, G.S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *Int. Journal of Robotics Research*, 25(5), 431–447.

Jarvis, R. (2006). Robot path planning: Complexity, flexibility and application scope. In *Proceedings of the 2006 Int. Symposium on Practical Cognitive Agents and Robots*, 3–14. Perth, Australia.

Kavraki, L.E., Svestka, P., Latombe, J.C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580.

Parker, L.E., Kannan, B., Tang, F., and Bailey, M. (2004). Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *Proceedings of IEEE Int. Conference on Intelligent Robots and Systems (IROS)*, volume 1, 1016–1022. Sendai, Japan.

Parker, L.E. (1998). ALLIANCE: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 220–240.

Rekleitis, I.M., Lee-Shue, V., New, A.P., and Choset, H. (2004). Limited communication, multi-robot team based coverage. In *Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 3462–3468.

Wagner, I.A., Lindenbaum, M.A., and Bruckstein, M. (1999). Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5), 918–933.