Complete coverage path planning of mobile robots for humanitarian demining

Abstract— **The paper presents a path planning algorithm for a non-circular shaped mobile robot to autonomously navigate in an unknown area for humanitarian demining. For that purpose the path planning problem comes down to planning a path from some starting location to a final location in an area so that the robot covers all the reachable positions in the area while following the planned path. In robotics literature this problem is called the complete coverage of an area. Without knowing the map of the area or with incomplete map information this problem also refers to the exploration of unknown areas, where the robot needs to incrementally build the area map and calculate new paths in order to complete its exploration task. Based on our previous complete coverage algorithm of known areas we have developed a complete coverage algorithm capable of operating in unknown areas with known border dimensions. The proposed algorithm uses occupancy grid map representation of the area. Every free cell represents a node in the graph being searched to find the complete coverage path. The proposed algorithm finds the complete coverage path in the graph accounting for the dimensions of the mobile robot, where non-circular shaped robots can be easily included. The complete coverage path is followed by the dynamic window algorithm, which includes robot's kinematic and dynamic constraints. The algorithms are implemented under the ROS (robot operating system) and tested in the stage 3D simulator for mobile robots.**

Index Terms— **autonomous mobile robots, path planning, coverage path planning, exploration**

I. INTRODUCTION

The path planning problem of a mobile robot for humanitarian demining application comes down to planning a path from a start position to a final position in an area so that the robot inspects all the reachable positions (nodes in a graph) while following the planned path. The problem of finding the path that visits all nodes in a graph is called the complete coverage path planning (LaValle, 2006). Finding an optimal path that visits every node in a graph exactly once is NP-hard problem known as the traveling salesman problem. Therefore, approximate or even heuristic solutions are used for the complete coverage path planning task.

For the humanitarian demining application most approaches are semi-autonomous, i.e., use remote steering and control (Hemapala, Belotti, Michelini and Razzoli, 2009) or move a robot randomly with some simple heuristics (Kopacek, 2002). Approach presented in (Cobano, Ponticelli and Santos, 2008) focuses on irregular terrain with low vegetation and does not include obstacles. In (Acar, Choset, Zhang and Schervish, 2003) it is shown that complete coverage achieves coverage in shorter time than random coverage.

Some approaches consider partially or completely unknown environments in the complete coverage path planning problem (i.e. exploration task) (Gabriely and Rimon, 2003), (Luo and Yang, 2008). On the other hand, some approaches rely on the *a priori* knowledge of a 2D map of the environment and cope with unknown obstacles detected by range sensor (De Carvalho, Vidal, Vieira and Ribeiro, 1997), (Qiu, Song, Zhang and Liu, 2006). A common approach is decomposing the environment into subregions (cell-decomposition (Latombe, 1991)), selecting a sequence of those subregions, and then generating a path that covers each subregion in turn. Most methods assume convex polygonal environments and perform exact cell decomposition (Huang, 2001), (Choset, 2000), (Wong and MacDonald, 2003), which can be very time consuming in changing environments. Methods based on the approximate cell decomposition (i.e. grid maps) are less time consuming, but suppose that the mobile robot has the dimensions of exactly one cell within the grid map (Gabriely and Rimon, 2003), (Zelinsky, Jarvis, Byrne and Yuta, 1993), (Choi, Lee, Baek and Oh, 2009). With such a crude representation the details of the environment are lost and a sequence of free cells leading through a narrow passage could also be lost although the mobile robot could pass through it.

Most complete coverage planning algorithms assume circular shaped mobile robot, and there is little work reported for complex non-circular shaped mobile robots. Some methods are based on templates (Hofner and Schmidt, 1995), (De Carvalho et al., 1997) – the robot combines a number of "basic motion macros" used to maneuver a robot with holonomic constraints. Generating a feasible trajectories to completely cover a bounded area for a car-like mobile robot is presented in (Guo and Balakrishnan, 2006).

This paper presents a new complete coverage path planning algorithm for complex shaped mobile robot and capable of operating in unknown areas of known border dimensions for the application of humanitarian demining. The algorithm is an extension of our previous complete coverage D* algorithm (CCD*) developed for circular shaped robots operating in known indoor environments with moving obstacles (Đakulović, Horvatić and Petrović, 2011). The proposed algorithm uses decomposition of the unknown area into squared cells of equal size and finds the complete coverage path that covers all reachable cells. The complete coverage path is integrated with the dynamic window obstacle avoidance algorithm (Fox, Burgard and Thrun, 1997) to produce smooth robot trajectory considering robot's kinematic and dynamic constraints.

The rest of the paper is organized as follows. Section II describes robot and environment representation for the humanitarian demining. Section III briefly reviews related algorithms including our previous algorithm for complete coverage path planning. Section IV presents the proposed complete coverage planning algorithm. Test results and conclusions are given in Section V and VI, respectively.

II. Robot and environment representation

A. The robot

In this paper, we assume usage of the humanitarian demining mobile robot MV-4 of DOK-ING company (Fig. 1), although developed algorithm is generally applicable to other robots. The dimensions of the prime mover together with the attached flail tool for activating mines are given in Table I (taken from `www.dok-ing.hr`). The simulation setup with the robot model is shown in Fig. 2 The simulated robot has on-board laser range sensor with 360° field of view. The maximal range used for mapping of unknown obstacles are set to 8 m, although outdoor laser range sensors provides much higher ranges (e.g. 30 m). The limitation of only 8 m ranges data assure more reliable map update especially in an uneven terrains (detecting of the ground). Additionally, the range data are collected in sequence with different time stamp from angle -180° to 180° so the positions of detected obstacles got distorted while the robot is moving and especially rotating, where distortion grows with distance from the robot. The robot has differential drive, i.e., it can rotate in place, and can move in forward and backward direction.

TABLE I

Technical data for the MV-4 mine clearance system.

Dimensions	(Length x Width x Height) mm
Prime Mover	3005 x 1530 x 1470 mm
Prime Mover With Flail (Clearing arm pulled in)	4455 x 2015 x 1470 mm
Prime Mover With Flail (Clearing arm extended)	5145 x 2015 x 1470 mm

Fig. 1. The MV-4 mine clearance system. Courtesy of DOK-ING company `www.dok-ing.hr`.

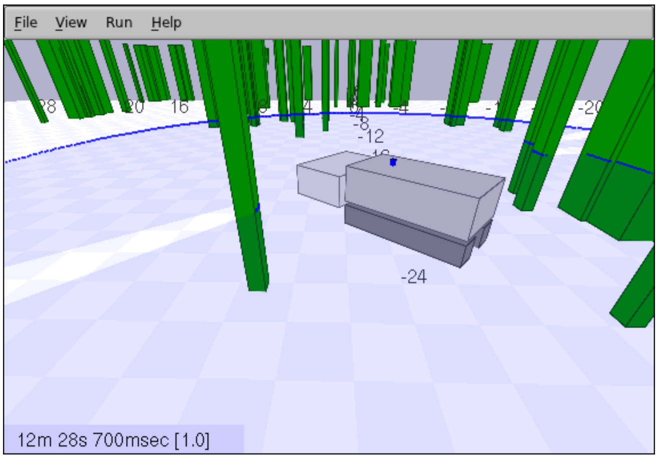

Fig. 2. The simulation setup in the Stage simulator – 3D view of the robot model and the part of the random environment.

B. The occupancy grid map

Two-dimensional (2D) occupancy grid maps are usually used to represent a continuous environment by an equally-spaced grid of discrete points (Thrun, Burgard and Fox, 2005). An occupancy grid map is created by approximate cell decomposition of an area. The whole unknown area with known borders is divided into squared cells of equal size e_{cell}, which are abstractly represented as the set of M elements $\mathcal{M} = \{1, \ldots, M\}$ with corresponding Cartesian coordinates of cell centers $c_i \in \mathbb{R}^2$, $i \in \mathcal{M}$. Each cell contains occupancy information of the part of the environment that it covers, which is continuously updated as the robot detects obstacles within the corresponding cells. The occupancy function $o(i)$ is used for representing the set of all occupied cells noted as $\mathcal{O} = \{i \in \mathcal{M} \mid o(i) = \infty\}$. Free cells have value $o(i) = 1$ or it can have higher values closer to obstacles as introduced in our previous work (Seder and Petrović, 2007).

The real shape of the robot represented in the grid map is shown in Fig. 3. It is usually assumed that the robot shape can be approximated by the circle, which position coordinates are planned. In case of more complex robot shapes (e.g. rectangle or ellipse), the path planning algorithm should also plan the robot orientation (LaValle, 2006), which significantly increases computational burden. In order to avoid robot orientation planning, in this paper we assume that the real shape of the robot can be approximated by two circles. One circle covers

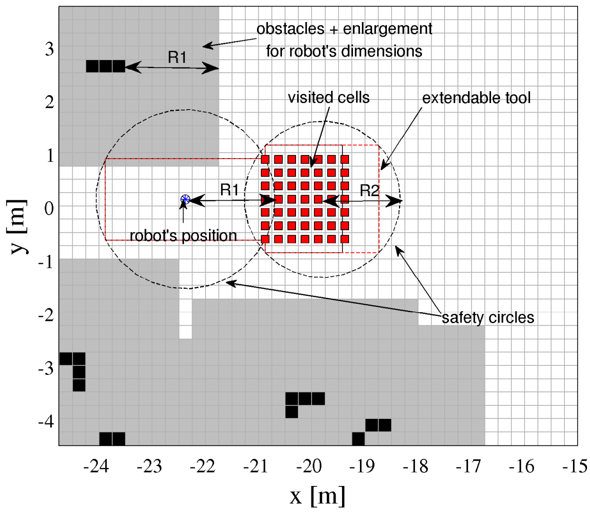

Fig. 3. The occupancy grid map with the robot real shape. The size of the cell is $e_{\text{cell}} = 0.25$ m.

the robot's vehicle (prime mover), and another circle covers the robot's flail tool for demining. By introducing the two circles and with certain adaptation of the obstacle avoidance module it is sufficient to plan only position coordinates. The larger circle (R_1 in Fig. 3) is used for obstacle enlargement. The robot's position is considered to be in the center of the prime mover. To allow the robot to be located within any free cell, all obstacles in the grid map are enlarged for the integer number of cells defined by the radius of the larger circle. For simplicity, obstacles are enlarged according to L_1 norm. For the path planning it is assumed that the robot needs to inspect the whole area by its tool and not by its mask. Further, it is assumed that the visited nodes are within the approximated squared shape of the tool, which is a little bit narrower than the real tool dimensions. On the other hand, while following the planned path, nodes that are visited are determined conservatively, i.e., the ones that are covered with the real shape of the robot, as shown in Fig. 4.

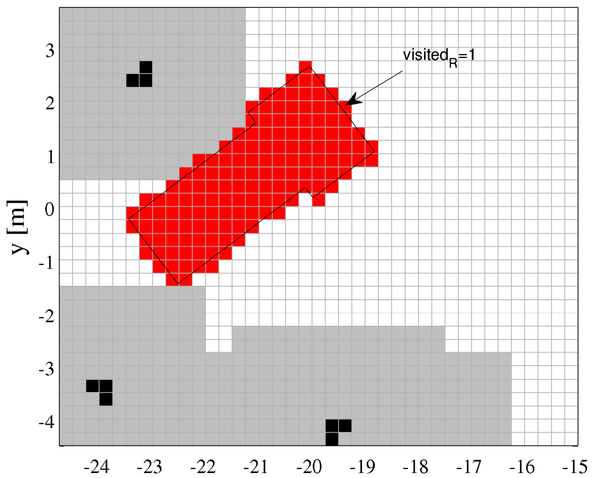

Fig. 4. The occupancy grid map with the visited nodes by the robot's mask.

C. Search graph

For searching a path the weighted undirected graph $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{W})$ is created from the occupancy grid map, where $\mathcal{N} = \mathcal{M} \setminus \mathcal{O}$ is the set of nodes (free cells), $\mathcal{E}$ is the set of edges $\mathcal{E} = \{\{i, j\} \mid i, j \in \mathcal{N}, i$ and j are neighbors$\}$, and $\mathcal{W}$ is the set of edge weights. Two nodes $i, j \in \mathcal{N}$ in the graph are *neighbors* if $\|c_i - c_j\|_\infty = e_{cell}$. The weights are defined as the cost of transition between neighbors

$$w_{i,j} := \|c_i - c_j\| \cdot \max\{o(i), o(j)\}. \tag{1}$$

A path in the graph $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{W})$ is defined as a succession of neighboring nodes. For example a path $\mathcal{P}$ of length L is defined by $\mathcal{P} = (n_1, n_2, \ldots, n_L)$ if $\{n_i, n_{i+1}\} \in \mathcal{E}$, $i = 1, \ldots, L-1$. The optimal path is the path with the smallest path cost $c(\mathcal{P}) := \sum_{i=1}^{L-1} w_{n_i, n_{i+1}}$.

III. Related algorithms

Here we present a simple idea of the coverage path calculation developed by (Zelinsky et al., 1993) on which we based our complete coverage D* (CCD*) algorithm. Then we briefly review the D* algorithm, which we used for cost values calculations, and our original CCD* algorithm applicable for circular robots navigation in known environments.

A. The basic idea of the coverage path algorithm

The cost values are assigned to the nodes in the graph by propagation of a wave front of values starting from the specified goal node G. For example, the goal node G gets value 1, its neighbor nodes get value 2, the neighbors of its neighbor nodes that do not have already assigned values get value 3, etc., see Fig. 5. Notice that diagonal and straight transitions are treated equally (L_1 norm). The complete coverage path is determined by following the largest values of costs of non-visited nodes starting from the specified start node S, see Fig. 5.

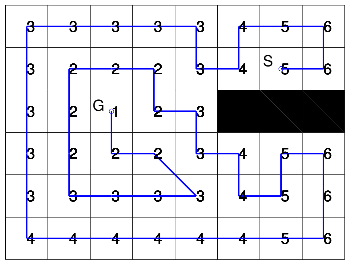

Fig. 5. An example of the complete coverage path from S to G by the PT algorithm.

B. The D* graph search algorithm

The D* algorithm (Stentz, 1994) is a well known graph search algorithm capable of fast replanning in changing environments. It is also known as dynamic version of the A* algorithm without the heuristic function, (Hart, Nilsson and Raphael, 1968).

For every searched node n, the D* algorithm computes the cost value $g(n)$ of the optimal path from the node n to the goal node G. The cost values can be the same as calculated by the wavefront algorithm if the graph weights are in L_1 norm. The execution of the D* algorithm can be divided into *initial planning* and *replanning* phases. Initial planning is performed if the robot is standstill at the start position and replanning is performed if the robot detects nodes with changed occupancy values during its motion. The number of expanded nodes is kept to a minimum and consequently the time of execution (for more details see (Stentz, 1994)).

C. The complete coverage D* algorithm

The CCD* algorithm developed in our previous work (Đakulović et al., 2011) uses the D* algorithm for assigning the cost values to the nodes. The CCD* algorithm is restated by Alg. 1. The D* algorithm is invoked from the robot's start position to assign cost values g to each node in the graph. The complete coverage path is determined by following the smallest values of costs of non-visited nodes starting from the start node S. By this the coverage path will go first close to the start position, where new obstacles are detected by the robot's sensor, and then outwards into the unexplored area. In order to avoid visiting already visited nodes the binary function **visited**$(n) = \{0, 1\}$ is used and its values are stored for each node. **visited**$(n) = 1$ is assigned to all cells which the robot's mask covers in the grid map with its dimensions when it is positioned in the node of the path. It is assumed that the robot's mask is a square of size $2M_R + 1$ cell sizes. In order to avoid overlapping of the robot's masks while robot follows the path the binary function **overlapped**$(n) = \{0, 1\}$ is used and its values are stored for each node. It is set to be twice as larger than the mask for determining visited nodes. Although it can be set to smaller square if we want to have more cells redundantly visited by the robot's mask (for example as a assurance for those nodes to be visited in non-perfect path following). The first node in the coverage path is the start node S. The next node in the coverage path is chosen from the candidate nodes set $\mathcal{N}_C$. The candidate nodes set $\mathcal{N}_C$ is defined to be non-overlapped nodes that are reachable ($g(n) < \infty$) and distanced from the previous node in the coverage path for the robot's square size in four straight directions through the grid. The next node in the coverage path is the closest (the node with the minimal cost g) node from the candidate nodes set (noted by M in Alg. 1). If the candidate nodes set is empty, then the A* search is performed to find (a) the closest non-visited node in the graph or (b) the visited node that is one cell near the enlarged obstacle and the robot mask number of cells M_R near the non-visited node. The case (b) is necessary because there can exists occupied nodes within the enlargement of obstacles that are reachable by the robot's mask but not by it's center. The auxiliary path $\mathcal{P}_{aux}$ to the found node is stored and added to the complete coverage path. The algorithm stops when there are no (a) or (b) cases left.

Algorithm 1 Coverage(S)

D*(S) *//Compute* $g(n),\ \forall n \in \mathcal{N}$
$C \leftarrow S$ *//Set the current node to* S
$\mathcal{P}_{aux} \leftarrow C$
$\mathcal{P} \leftarrow \emptyset$
while 1
    $\forall n \in \mathcal{P}_{aux},\ m \in \mathcal{M},\ \|c_n - c_m\|_\infty \leq M_\mathrm{R} \cdot e_\mathrm{cell}$
        visited$(m) \leftarrow 1$
    $\forall n \in \mathcal{P}_{aux},\ m \in \mathcal{N},\ \|c_n - c_m\|_\infty \leq 2M_\mathrm{R} \cdot e_\mathrm{cell}$
        overlapped$(m) \leftarrow 1$
    $\mathcal{N}_C \leftarrow \{n \in \mathcal{N} \mid \|c_n - c_C\|_\infty = (2M_\mathrm{R} + 1) \cdot e_\mathrm{cell}$
        and **overlapped**$(n) = 0$ and $g(n) < \infty\}$
    if $\mathcal{N}_C \neq \emptyset$
        find $M \in \mathcal{N}_C$ with minimal g
    else
        A*(C) and stop at **visited**$(M) = 0$
            or $\|c_M - c_o\|_\infty = e_\mathrm{cell},\ o \in \mathcal{O}$ and $\exists n$,
            visited$(n) = 0, \|c_M - c_n\|_\infty < M_\mathrm{R} \cdot e_\mathrm{cell}$
        if no such node M exists
            return $\mathcal{P}$
        end
    end
    $\mathcal{P}_{aux} \leftarrow \mathcal{P}_{aux}(C, M)$
    $C \leftarrow M$ *//Set the new current node*
    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_{aux}$
end

IV. The complete coverage planning for humanitarian demining

The proposed complete coverage planning algorithm for humanitarian demining is actually a modified version of our original complete coverage D* algorithm. While original CCD* algorithm is limited to path planning of circular shaped robots in known area, introduced modifications enables path planning of non-circular shaped robots in unknown areas. The pseudocode of the proposed algorithm is given by Alg. 2.

A. The CCD algorithm's modifications*

(mijenjam paragraf) Modifications of the CCD* algorithm, which enable path planning for complex robot's shape, include first planning the coverage path for the tool's center, i.e., the node in the center of the tool's squared shape (see Fig. 3), and afterwards deriving for the robot's position. The coverage path is determined by invoking the function Coverage given by Alg. 1 (line 7 of the Alg. 2). It is assumed that the tool's mask is a square of $2M_T + 1$ cell sizes, and the robot's mask is arbitrary. The first node in the coverage tool's path is the tool's center point when the robot is in the start position, noted by T, and the first node in the coverage robot's path is of course the start node, noted by S. The tool's center point is displaced from the robot's center point by

$$T = S + R(\theta) \cdot t_d, \tag{2}$$

where $R(\theta) = [\cos\theta \sin\theta; -\sin\theta \cos\theta]$ is rotation matrix for the robot's orientation θ, and t_d is translation vector between the robot and tool center point in the robot local reference frame. For the described robot model it is $t_d = e_{cell} \cdot [M_R + M_T;\ 0]^T$. To ensure that the robot reaches all planned position by its tool the path composed of robot's positions must be calculated such that the robot actually visits all reachable nodes by its tool while following that path. Denote paths composed of robot's and tool's positions by $\mathcal{P}_R$ and $\mathcal{P}_T$, respectively. The point in the robot's path is calculated according to the curvature of the tool's path, where 0-curvature denotes that path direction is not changing at point $i-1$, while 1-curvature denotes that path direction is changing at point $i-1$:

$$\begin{array}{ll} \mathcal{P}_R(i) = \mathcal{P}_T(i) - R(\alpha) \cdot t_d & \text{for 0-curvature} \\ \mathcal{P}_R(i) = \mathcal{P}_T(i-1) - R(\beta) \cdot t_e & \text{for 1-curvature,} \end{array} \tag{3}$$

where α is the angle between the line connecting $\mathcal{P}_T(i)$ and $\mathcal{P}_R(i-1)$ and the positive x-axis, and β is the angle between the line connecting $\mathcal{P}_T(i)$ and $\mathcal{P}_T(i-1)$ and the positive x-axis (orientation of the tool's path segment), and translation t_e

Algorithm 2 modified-CCD*(S)

1: $\forall n \in \mathcal{N},\ k(n) \leftarrow \infty,\ g(n) \leftarrow \infty,\ b(n) \leftarrow n,$
  visited$(n) \leftarrow 0$, **overlapped**$(n) \leftarrow 0$,
  visited$_R(n) \leftarrow 0$, **overlapped**$_R(n) \leftarrow 0$
2: $R \leftarrow S$ *//Initialization*
3: while 1
4: if $R = S$ or **changed-nodes**(R)
5: **D***(S) *//(Re)compute* $g(n),\ \forall n \in \mathcal{N}$
6: $T \leftarrow$ **robot-to-tool-transform**(R)
7: $\mathcal{P}_T \leftarrow$**Coverage**(T)
8: $\mathcal{P} \leftarrow$ **tool-to-robot-transform**$(\mathcal{P}_T)$
9: end
10: $R \leftarrow$**path-following**$(\mathcal{P})$
11: $\mathcal{N}_R \leftarrow \{n \in \mathcal{M} \mid n \in$ **robot-shape**$(R)\}$
12: for $\forall n \in \mathcal{N}_R$ **visited**$_R(n) \leftarrow 1$
13: $\mathcal{N}_T \leftarrow \{n \in \mathcal{N} \mid \|c_n - c_T\|_\infty \leq (2M_\mathrm{T}+1) \cdot e_\mathrm{cell}$
14: for $\forall n \in \mathcal{N}_T$ **overlapped**$_R(n) \leftarrow 1$
15: for $\forall n \in \mathcal{M}$
16: **visited**$(n) \leftarrow$ **visited**$_R(n)$
17: **overlapped**$(n) \leftarrow$ **overlapped**$_R(n)$
18: end
19: end

denotes displacement of the robot's point from the tool's point if both the robot's main body and the tool are aligned with their two perpendicular sides. For the described robot model it is $t_e = e_{cell} \cdot [M_R - M_T;\ 0]^T$. Such displacement assures that the robot is within the planed visited cells if its orientation is in the direction of the tool's path orientation β. Although the robot's orientations are not explicitly planned, they are achieved by following the planned x and y coordinates of the robot's path. Functions **robot-to-tool-transform** and **tool-to-robot-transform** in Alg. 2 (lines 6 and 8) denote transformations given by (2) and (3), respectively. The path is followed by the **path-following** function (line 10) which is an integration of the computed complete coverage path and the dynamic window obstacle avoidance algorithm (Fox et al., 1997), which is also adapted for non-circular shape of the robot.

When an on-board range sensor detects obstacles, corresponding cells become occupied. Additionally, cells between the robot and the detected obstacles become free. Function **changed-nodes**(R) in Alg. 2 handles these changes in a way such that map representation of an unknown area is continuously updated. Afterwards, the path replanning process is initiated and certain parts of the area are included or removed from the path. To track which cells are visited by the robot while following the complete coverage path, functions **visited**$_R(n) = \{0, 1\}$ and **overlapped**$_R(n) = \{0, 1\}$ are used. Before each execution of the complete coverage path calculation values of functions **visited**(n) and **overlapped**(n) are rewritten by the new ones **visited**$_R(n)$ and **overlapped**$_R(n)$, respectively. Since D* stores information about previous calculations the minimal number of nodes are examined. The new coverage path is recalculated by the same procedure but with smaller number of non-visited nodes in the graph.

B. Illustration of algorithm's iterations

Hereafter, we illustrate on a simulation example how the proposed path coverage algorithm plans the complete coverage path for a complex shaped robot in an unknown area with randomly placed obstacles. Obstacles that are detected by the sensor readings from the robot's start position are inserted into the graph. Then, the D* search is performed from the start node S to calculate the cost values g for every reachable node. The cost values are shown in Fig. 6. The smallest value of the

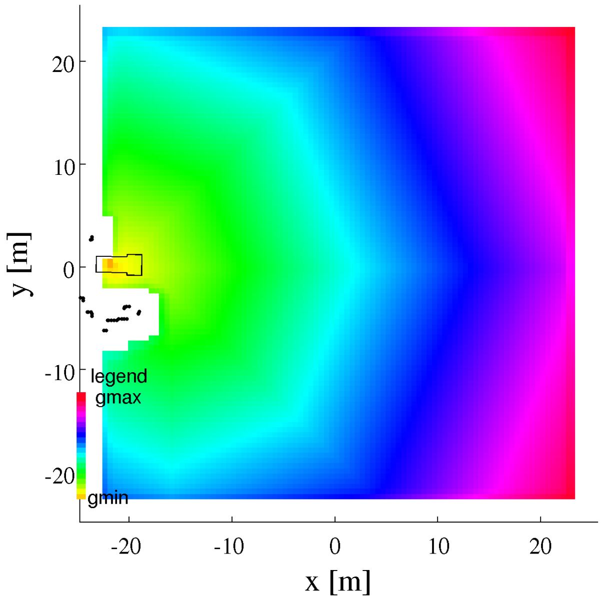

Fig. 6. The cost values by the D* algorithm – the smallest value at the robot's start position (-22 m, 0 m).

cost is at the robot's start position (-22 m, 0 m), $g(S) = 0$, and the larger values are at the most distanced nodes from the start node. Detected obstacles and white area around them are non-reachable by the robot's center, but some of them can be reached by the tool's mask.

(mijenjam paragraf) The first node in the coverage tool's path is the tool's center point, and the first node in the coverage robot's path is the start node. The tool's center point is distanced from the robot's position for the fixed length $(M_R + M_T) \cdot e_{cell}$ along the x-axis of the robot's local reference frame (robot's direction of moving forward), where $M_R = 6$ and $M_T = 3$. In the path planning step, smaller mask of squared shape (inner part of the real tool shape) is used for determining visited nodes (see Fig. 3), as opposed to the path following step where all cells that are covered by the real shape of the robot are used (see Fig. 4). Only for the first step of the algorithm visited and overlapped values are determined for complete robot mask. In all other steps those values are determined only for the tool. The next node in the coverage tool's path is chosen from the candidate nodes in the same way as in the original CCD* algorithm for the robot's path. The candidate nodes are defined to be non-overlapped nodes that are reachable ($g(n) < \infty$) and distanced from the previous node in the coverage tool's path for the tool square size in four straight directions through the grid. The next node is the one with the smallest cost value. An example of the first iteration of producing the coverage path is shown in Fig. 7. When each

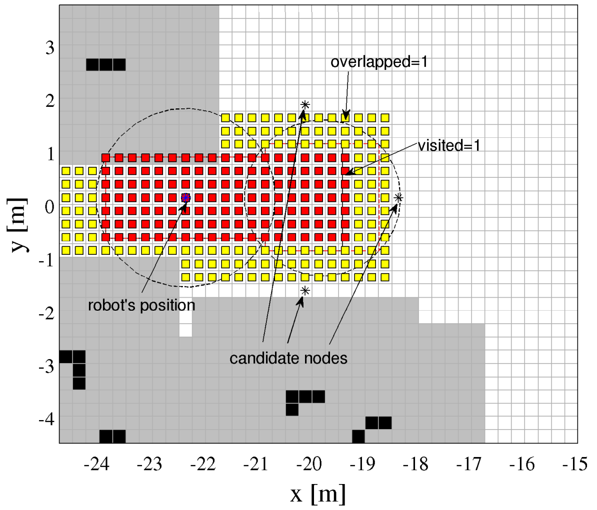

Fig. 7. The first iteration of the coverage path – assigning the visited and overlapped nodes and determining the candidate nodes.

node in the tool's path is determined, for example the i-th node of the tool's path, the i-th node of the robot's path is calculated according to (3). An example of this procedure is shown for the sixth iteration of the algorithm in Fig. 8. The

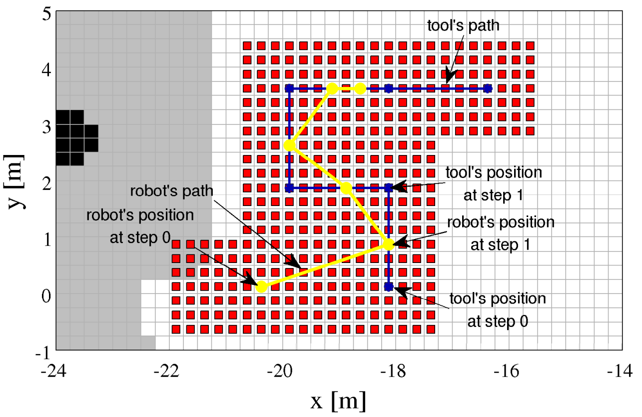

Fig. 8. The sixth iteration of the coverage path – determining the robot's path from the tool's path.

tool's path changes direction in points 0, 1, 2 and 3 and robot's points are calculated for the 1-curvature. In the 4th point the direction of the tool's path is not changing and the robot's point is calculated for the 0-curvature of (2). Described iterations continue until there is no surrounding candidate nodes. Then, like in the CCD* algorithm, the A* search is executed from the last node of the tool's path to find the next non-visited node (as explained in Section III-C).

V. Test results

The proposed algorithm was implemented in ROS (the robot operating system and the Stage simulator www.ros.org) with the MV4 robot model described in Section II. The AMCL algorithm (Adaptive Monte Carlo Localization) was used for robot localization. For path following a dynamic window based algorithm, described in our previous work was used (Seder, Maček and Petrović, 2005) with certain adaptations for two circle shaped robot. The robot was allowed to go backwards in some deadlock scenarios. A randomly generated map was used with dimensions 50 m x 50 m as the simulation map of an unknown area, see Fig. 9. The laser range readings used

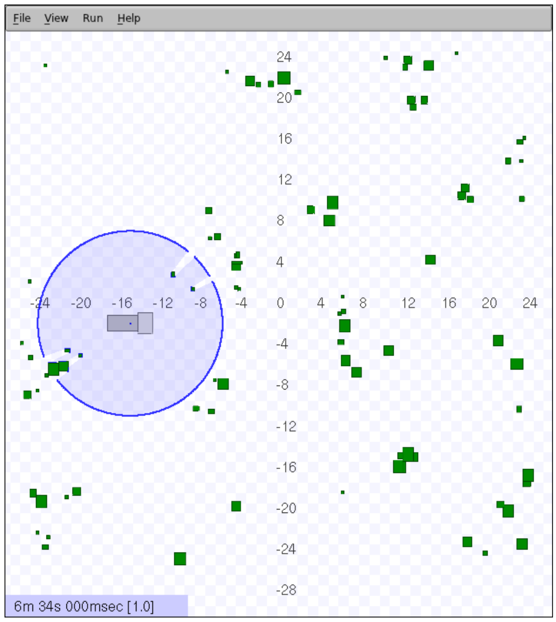

Fig. 9. The simulation setup in the Stage simulator – 2D top view of the robot model and the randomly generated simulation map of the unknown area.

in the simulation has full field of view (360 degrees) and was limited to 8 meters to cope only with obstacles in local vicinity of the robot. While the robot was moving, it detected unknown obstacles and replanned the complete coverage path.

(dodajem novi paragraf) Simulation tests are divided into two parts: (1) the initial planning and (2) replanning. The first part evaluate calculated initial path for the known map of the environment, while in the second part the same environment is used but the map is unknown and the path must be replanned as the robot detects obstacles.

A. The initial path planning

(dodajem novi section) Figure 10 shows initial tool's path and covered cells by the tool. Both tool's path and robot's path are presented in Fig. 11 with noted starting and finishing points in the paths (S_T and G_T for the tool's path and S_R and G_R for the robot's path). The length of the complete coverage path of the tool is 1451.3 m, and of the robot is 1384 m. Because of the transformations (3) between the tool's and robot's positions the robot's path is slightly shorter. The total reachable area by the tool is 1652.1 m^2 (26434 cells, where

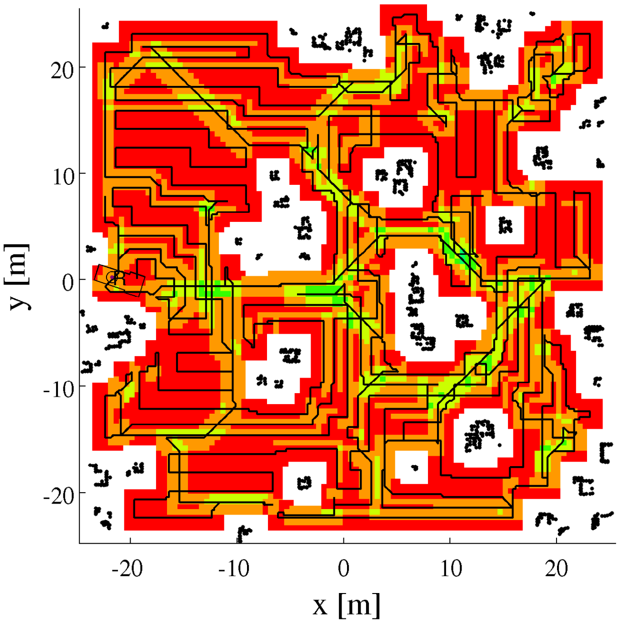

Fig. 10. The initial complete coverage path of the tool and the numbers of visits of each cell - lighter colors indicate higher numbers of visits.

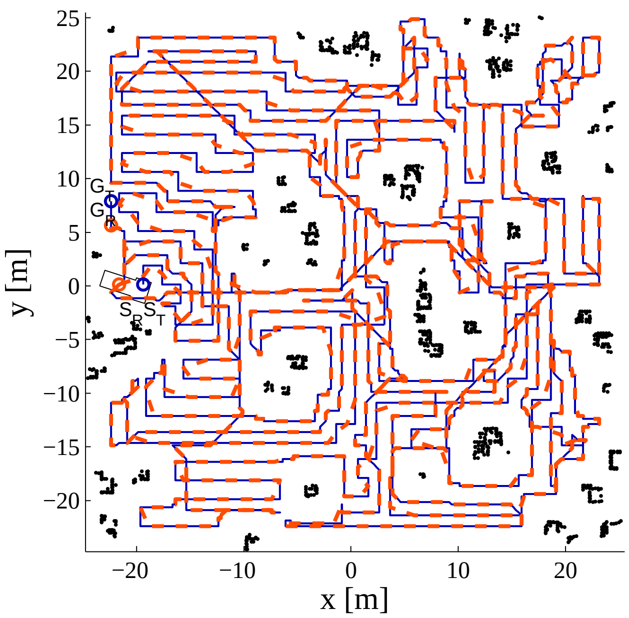

Fig. 11. The initial complete coverage tool's (solid line) and robot's paths (dashed line).

cell edge is 0.25 m). The number of visits noted in Fig. 10 with corresponding colors are: 13889 (53%) cells are visited only once, 9752 (37%) cells are visited twice, 2483 (9%) cells are visited 3 times, 297 (1%) cells are visited 4 times and 12 (<1%) cells are visited 5 times. White area around obstacles remained nonvisited because the tool's mask is smaller than the robot's mask for which the obstacles are enlarged.

B. The path replanning

(dodajem naslov - novi section i mijenjam tekst) Since the robot does not follow the planned path ideally, for the replanning test the condition of assigning the values overlapped(n)=1 was changed to assure redundancy. We have chosen to have 2 cells of redundancy, i.e., nodes distanced from the nodes in the path for less or equal to $(2M_R - 2) \cdot e_{cell}$ are set to overlapped (changed line 7 in Alg. 1). Figure 12 shows six of many replannings while moving through the unknown area. The first one (upper left) is the initial planning. End nodes of the tool's path and the robot's path are noted by G_T and G_R, respectively. Comparing to Fig. 6 it can be noticed that the coverage path goes over the nodes with the cost values that are increasing along the coverage path. The path was changing at each replanning step and had more path direction changes with more obstacles detected. Due to non-perfect path following some parts of the area remind non-visited. Those parts were included in the new complete coverage path with the next replanning step. However, some cells were very hard to visit due to complex shape of the robot and rotation on the spot near certain obstacle configuration was not admissible by the dynamic window algorithm.

(tu tablicu) Table **??** shows evaluation parameters during 7 (of many) replanning steps, each row corresponding to each replanning step, and replanning step 2 is after replanning step 1 in time, etc. It is also noted which replanning step corresponds to which snapshot in Fig. 12. Total time needed to visit green area is noted by t_{sum} and it grows higher with the replanning step. The robot was traveling with average speed v_{ave} of about 300 mm/s. Maximal allowed speed was 500 mm/s for forward motion and 100 mm/s for backward motion. Maximal orientation speed was limited to 100 °/s. Number of nodes visited only once decrease with time since the robot is traveling over the same area trying to reach borders of obstacles. This can be noticed in Fig. 12 from snapshot 5 to snapshot 6, where the robot tried to visit cells near the border of obstacles, which was not successful in all cases since the robot needed also to rotate in place to reach the non-visited cells, which was not planned by the algorithm. For the snapshot 6 the robot covered total number of 30419 cells (1901.2 m^2, for $e_{cell} = 0.25$ m) of the total number of 30562 reachable cells. The length of robot's trajectory l_{traj} is increasing with time, while the length of the new calculated coverage path l_{path} that must be followed is decreasing with time. It can be noticed that the sum $l_{traj} + l_{path}$ is almost constant for the first 5 replanning steps and for the 6th and 7th it grows higher due to non-perfect path following.

VI. Conclusions

In this paper a new complete coverage path planning algorithm for humanitarian demining has been proposed. It was shown that it effectively plans the robot's path ensuring that the flail tool visits all reachable regions (cells) of the inspected area. The test results of the proposed algorithm have shown satisfactory behavior of the algorithm in the environment populated by unknown static obstacles. A few cells have stayed non-visited due to non-perfect path following. In our future work, a better path following algorithm will be developed and additional constraints will be included in the planning algorithm such as the minimization of the number of path

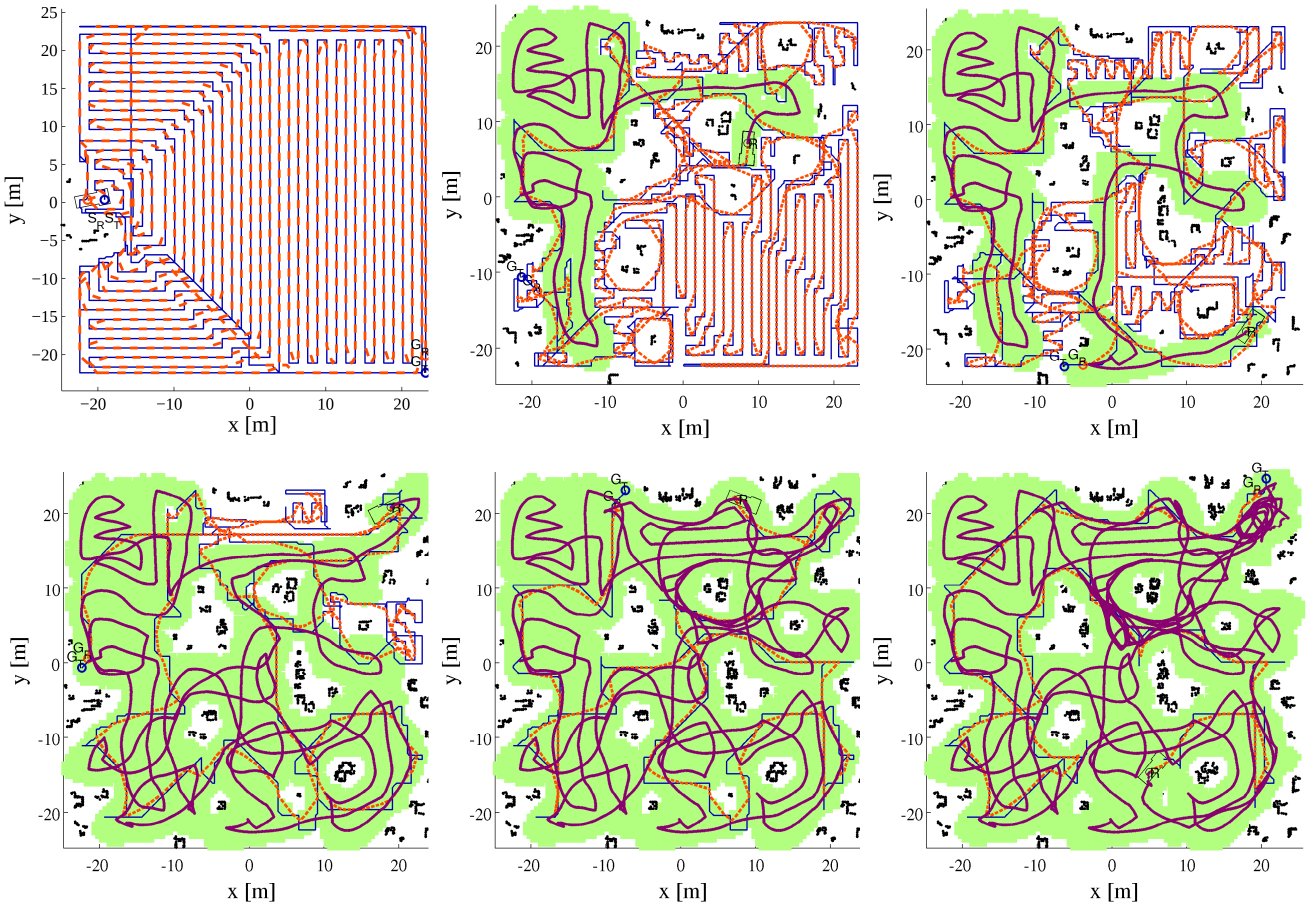

Fig. 12. Six snapshots of (many) replanning steps while the robot is moving through the unknown area and detecting unexplored obstacles – visited area is colored green, the new tool's path is noted by solid line, the new robot's path is noted by dashed line, and driven trajectory is bolder curve.

direction changes and planning also orientations for certain points near the obstacles.

References

Acar, E., Choset, H., Zhang, Y. and Schervish, M. 2003. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International Journal of Robotics Research* **22**(7-8), 441–466.

Choi, Y., Lee, T., Baek, S. and Oh, S. 2009. Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE. pp. 5788–5793.

Choset, H. 2000. Coverage of known spaces: the boustrophedon cellular decomposition. *Autonomous Robots* **9**(3), 247–253.

Cobano, J., Ponticelli, R. and Santos, P. 2008. Mobile robotic system for detection and location of antipersonnel land mines: field tests. *Industrial Robot: An International Journal* **35**(6), 520–527.

De Carvalho, R., Vidal, H., Vieira, P. and Ribeiro, M. 1997. Complete coverage path planning and guidance for cleaning robots. *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE 1997)*. pp. 677–682.

Đakulović, M., Horvatić, S. and Petrović, I. 2011. Complete Coverage D* Algorithm for Path Planning of a Floor-Cleaning Mobile Robot. *Proceedings of the 18th IFAC World Congress*. pp. 5950–5955.

Fox, D., Burgard, W. and Thrun, S. 1997. The dynamic window approach to collision avoidance. *Robotics & Automation Magazine, IEEE* **4**(1), 23–33.

Gabriely, Y. and Rimon, E. 2003. Competitive on-line coverage of grid environments by a mobile robot. *Computational Geometry* **24**(3), 197–224.

Guo, Y. and Balakrishnan, M. 2006. Complete coverage control for non-holonomic mobile robots in dynamic environments. *IEEE International Conference on Robotics and Automation, (ICRA 2006)*. IEEE. pp. 1704–1709.

Hart, P., Nilsson, N. and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* **4**(2), 100–107.

Hemapala, M., Belotti, V., Michelini, R. and Razzoli, R. 2009. Humanitarian demining: path planning and remote robotic sweeping. *Industrial Robot: An International Journal* **36**(2), 146–156.

Hofner, C. and Schmidt, G. 1995. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robotics and Autonomous Systems* **14**(2), 199–212.

Huang, W. 2001. Optimal line-sweep-based decompositions for coverage algorithms. *IEEE International Conference on Robotics and Automation (ICRA 2001)*. pp. 27–32.

Kopacek, P. 2002. Demining robots: A tool for international stability. *World Congress*. Vol. 15. pp. 1423–1423.

Latombe, J. 1991. *Robot Motion Planning*. Kluwer Academic Publishers. Dodrecht, Netherlands.

LaValle, S. 2006. *Planning algorithms*. Cambridge Univ. Pr.

Luo, C. and Yang, S. 2008. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neural Networks* **19**(7), 1279–1298.

Qiu, X., Song, J., Zhang, X. and Liu, S. 2006. A complete coverage path planning method for mobile robot in uncertain environments. *The Sixth World Congress on Intelligent Control and Automation, (WCICA 2006)*. Vol. 2. IEEE. pp. 8892–8896.

Seder, M., Maček, K. and Petrović, I. 2005. An integrated approach to real-time mobile robot control in partially known indoor environments. *32nd Annual Conference of IEEE Industrial Electronics Society (IECON 2005)*. pp. 1785–1790.

Seder, M. and Petrović, I. 2007. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. *IEEE International Conference on Robotics and Automation (ICRA 2007)*. pp. 1986–1991.

Stentz, A. 1994. Optimal and efficient path planning for partially-known environments. *IEEE International Conference on Robotics and Automation (ICRA 1994)*. pp. 3310–3317.

Thrun, S., Burgard, W. and Fox, D. 2005. *Probabilistic robotics*. Cambridge, Massachusetts: MIT Press.

Wong, S. and MacDonald, B. 2003. A topological coverage algorithm for mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*. Vol. 2. IEEE. pp. 1685–1690.

Zelinsky, A., Jarvis, R., Byrne, J. and Yuta, S. 1993. Planning paths of complete coverage of an unstructured environment by a mobile robot. *Proceedings of International conference on Advanced Robotics (ICAR 1993)*. pp. 533–538.