2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

A Combined Approach for Robot Placement and Coverage Path Planning for Mobile Manipulation

Fabian Paus, Peter Kaiser, Nikolaus Vahrenkamp and Tamim Asfour

***Abstract*— Robotic coverage path planning describes the problem of determining a configuration space trajectory for successively covering a specified workspace target area with the robot's end-effector. Performing coverage path planning for mobile robots further requires solving the problem of robot placement, i.e. determining of suitable robot base positions to perform the task. Finding an optimal solution is hard as both problems cannot be solved independently. Combined robot placement and coverage planning is particularly interesting if repositioning of the robot is costly or if simultaneous repositioning and end-effector motion is not desired.**

In this paper, we present a general approach for combined robot placement and coverage path planning that takes constraints like collision avoidance and static stability into account. In contrast to related approaches, we focus on mobile manipulation tasks that require a fixed placement for executing coverage trajectory segments. The approach is evaluated in two scenarios that exemplify the broad range of possible applications: The coverage of a building facade using a robotic manlift and the coverage of an industrial conveyer belt for maintenance tasks using the humanoid robot ARMAR-III.

I. Introduction

Coverage path planning in robotics is a well-understood problem with applications especially concentrated in the areas of industrial and agricultural robotics. Prominent examples include spray-painting robots in industrial assembly lines [1] or the coverage of agricultural fields using mobile robots [2] or drones [3]. Traditional methods can find solutions for such problems in two principle cases: 1) if the robot is statically mounted at a fixed position, e.g. in the case of an industrial spray-painting robot, or 2) if a mobile robot can freely move during its coverage task, e.g. in the case of a mobile agricultural robot.

In this work we investigate a third case, in which a freely movable mobile robot is employed under the assumptions that 1) repositioning of the robot produces high operational costs and 2) the coverage task is interrupted during the repositioning process. There is a wide variety of possible scenarios that expose the above characteristics, e.g. the coverage of a building facade using a mounted robotic manlift (see Fig. 2). A less obvious example is the cleaning of a large surface using a mobile robot with sophisticated kinematics, e.g. a humanoid robot. Fig. 1 displays such a scenario using the humanoid robot ARMAR-III [4] cleaning a conveyer belt. Precise end-effector operation during locomotion is a challenging field of research in humanoid robotics, justifying the need for temporarily fixed robot placements in this example.

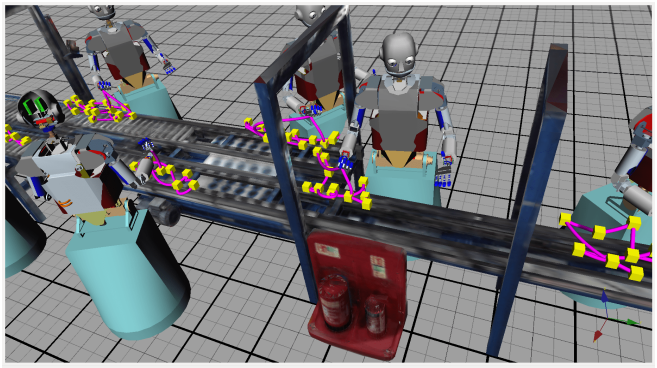


Fig. 1: ARMAR-III covering a portion of an industrial conveyer belt in simulation. Multiple placement poses and the coverage trajectories for the right hand are visualized. The trajectory is shown as a magenta line connecting the yellow guard points.

The approach proposed in this work aims at solving the problem of coverage path planning in combination with the problem of robot placement, which are usually studied separately. The combined problem of robot placement and coverage planning in a 3D workspace can be stated as follows: Given the kinematic properties of the robot, a structural description of the environment and the target surface, find a list $\mathcal{S}$ of partial solutions, each containing a robot pose $\boldsymbol{p}_i$ and a trajectory t_i which covers a portion of the target surface:

$$\mathcal{S} = [(\boldsymbol{p}_1, t_1), \ldots, (\boldsymbol{p}_n, t_n)] \tag{1}$$

In a valid solution, the combination of all partial solutions from $\mathcal{S}$ cover the whole target surface. An optimal solution to the combined problem minimizes two fundamental metrics: 1) the number of partial solutions with distinct robot poses and 2) the total cost of the coverage trajectories. Metric 1) determines how often the robot needs to be repositioned while metric 2) depends on the definition of path cost. Reasonable definitions include the path length, execution time or energy consumption. The movement cost between partial solutions is a potential third metric which will not be considered in this work.

The research leading to these results has received funding from the German Research Foundation (DFG) within the Transregional Collaborative Research Centre Invasive Computing (SFB/TR 89) and from the European Unions Horizon 2020 Research and Innovation programme under grant agreement No 643950 (SecondHands). We thank RUTHMANN GmbH & Co. KG and 3DIS GmbH for their support during the first author's master thesis.

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. {paus, asfour}@kit.edu

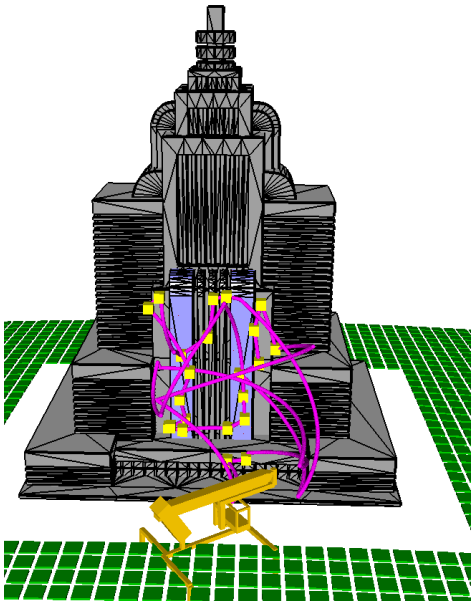


Fig. 2: The simulation result for a manlift covering a building's facade. The basket at the end of the kinematic chain follows the magenta trajectory during the coverage task. The yellow boxes indicate guard points.

The combined problem can be trivially solved using traditional coverage algorithms by extending the configuration space with the robot pose. This approach is viable if the robot can change its pose with little cost and the coverage task can be executed during locomotion. In this work however, we assume significant cost for robot repositioning, e.g. in the case of a robotic manlift, and the infeasibility of the coverage operation during locomotion.

The remainder of this work is structured as follows. In Section II we discuss related work regarding robot placement, coverage path planning and mobile manipulation. Section III describes our approach to solving the combined problem and Section IV presents the experimental results for the two chosen evaluation scenarios, manlifts and the humanoid platform ARMAR-III in simulation. We conclude the paper in Section V.

II. Related Work

In Section II-A we discusses different approaches to find suitable poses which allow the robot to reach a certain part of the workspace. Subsequently, in Section II-B we present related work considering the problem of planning trajectories which cover 3D surfaces with the robot's end-effector. Section II-C discusses related approaches for mobile manipulation.

A. Robot Placement

In order to determine whether a robot pose is suitable for execution of a given task it is necessary to have a representation of the workspace. Often a discrete model of the reachable space is used to capture the robot's capabilities. Guan and Yokoi present a method which uses randomized sampling to approximate the reachable space of a humanoid robot [5]. The workspace is represented using discrete cubes. Zacharias et al. introduce information about the directional structure of the workspace [6]. They sample points on spheres which are embedded in discrete workspace cubes. If an inverse kinematics solver finds a solution to a sampled point, it is marked as reachable. The structure of reachable points on the spheres is used to fit shape primitives in order to visualize and understand the directional nature of the robot's workspace. This model was used to find positions for mobile manipulators which allowed the execution of task specific 3D trajectories with high probability [7]. Vahrenkamp et al. try to find robot base poses by inverting the reachability representation and calculating the intersection with a 2D ground plane [8]. Their approach was evaluated in simulation for grasping and door opening tasks on a humanoid platform. The same approach was used by [9] to select stances for grasping tasks. They focused on foot placement for the bipedal robot NAO. Associating robot base poses with the probability of successfully executing a manipulation task was termed action-related places (ARPlace) in [10]. The authors evaluate discrete base poses by applying a predictive model which was learned from experience. The presented approach is able to handle uncertainties in robot pose and environment estimation by incorporating those into the probability model.

B. Coverage Path Planning

Coverage path planning is employed in different domains, e.g. spray-painting automobile parts [1], coverage of agricultural fields [2] and inspection of structures on the ocean floor [11]. It has been an area of active research for the past years. One of the most relevant surveys by Choset and Pignon presents approaches in which the robot covers its 2D workspace [12]. A later survey by Galceran and Carreras includes references to 3D surface coverage [13]. We will be focusing on coverage path planning for 2D surfaces embedded in a 3D workspace.

Most approaches divide the coverage planning problem into two subproblems. The first step is to determine a set of viewpoints which cover the whole target surface. This subproblem is also known as the art gallery problem [14] in which the minimal set of guards should be determined who can view the entire gallery. Then these viewpoints need to be connected via a closed walk. This second step resembles the traveling salesman problem (TSP) [15]. In this problem a salesman searches for the shortest route between cities which travels through every city and both starts and ends in his hometown. Since both subproblems in general are NP-hard, a common approach is to use randomized sampling to find suitable solutions in an acceptable time (e.g. [16], [3]). The authors in [17] term the first subproblem as *coverage sampling problem (CSP)* and the second one as *multi-goal path planning (MPP)*.

In [18] a lazy method for MPP is presented. Their approach is based on the assumption that path planning between individual goals dominates the runtime cost compared to the computation of approximate solutions to the TSP. They use

a lower bound estimation of the path length between goals as an heuristic to calculate candidate TSP solutions and only employ the complete path planner for edges in candidate solutions.

Another promising approach solves the coverage path planning as a whole. *Random Inspection Tree Algorithm (RITA)* is a randomized algorithm which iteratively improves the coverage path [19]. It is probabilistically complete and asymptotically optimal but requires more computation time than the divide-and-conquer strategies.

C. Mobile Manipulation

In [20], the goal is to maximize the manipulability of the manipulator during locomotion. This is achieved by a planning and control algorithm which was designed for mobile platforms. Motion planning for mobile manipulators can be implemented by separating the degrees of freedom into two groups. The first group is responsible for moving the mobile base whereas the second group moves the end-effector. The authors in [21] use this approach and present methods for generating motions which follow given end-effector paths.

Many constraints need to be considered while generating motions for mobile manipulation. Brock et al. combine in [22] obstacle avoidance and task-oriented motion using the elastic strip framework. Elastic roadmaps [23] are able to cope with a dynamically changing environment by encoding connectivity information of the workspace. Compared to configuration space planners this reduces the complexity of the problem and therefore allows for frequent updates to the data structures.

III. Approach

Our proposed algorithm tries to find a suitable solution to the combined problem of robot placement and coverage path planning by employing a divide-and-conquer strategy and making use of heuristics to keep the computation affordable. We first describe our developed algorithm in Section III-A. In Section III-B and III-C, we present our heuristic planning approaches to the reachability and the coverage problem.

A. Algorithm

Let C be the configuration space of the robot, $\boldsymbol{p}_{\text{global}} \in SE(3)$ its global pose, $\boldsymbol{q}_{\text{transport}} \in C$ a safe transport configuration in which the robot can change $\boldsymbol{p}_{\text{global}}$ freely, and $\boldsymbol{p}_{\text{eef}} : SE(3) \times C \to SE(3)$ a function mapping the robot's pose and configuration to the corresponding end-effector pose. It is assumed that the $\boldsymbol{p}_{\text{global}}$ is independent of the configuration. We further assume that the environment is known and is represented as a triangular mesh in 3D space. The part of the environment which should be covered by the end-effector will be called target surface $S_{target} \subset \mathbb{R}^3$. Only a specific subset $G \subset SE(3)$ contains possible solutions to the robot placement subproblem, e. g. a humanoid robot can usually only be placed on the ground. In this paper we constrain G further by requiring it to be a finite set.

The goal is to determine a list $\mathcal{S}$ of partial solutions. A partial solution $(\boldsymbol{p}, t)$ consists of a global pose $\boldsymbol{p} \in G$ and an end-effector trajectory t which covers a part of the target surface. The trajectory t can be described as a curve $c_t : [0,1] \to Q$ where $c_t(0) = c_t(1) = \boldsymbol{q}_{transport}$. A point on the target surface $\boldsymbol{x} \in S_{\text{target}}$ is covered by a partial solution if the end-effector comes sufficiently close to the point while following the corresponding trajectory t:

$$\text{covered}(\boldsymbol{x}, \boldsymbol{p}, t) = \exists_{u \in [0,1]} \|\boldsymbol{x}_{\text{eef}}(p, c_t(u)) - \boldsymbol{x}\| < \delta,$$

where $\boldsymbol{x}_{\text{eef}}(\boldsymbol{p}, \boldsymbol{q}) \in \mathbb{R}^3$ is the positional part of the end-effector pose $\boldsymbol{p}_{\text{eef}}(\boldsymbol{p}, \boldsymbol{q})$ and δ is an application dependent distance threshold. We only consider positional errors since coverage task usually require a specific end-effector orientation, e.g. inspecting a pipe with a mounted camera or cleaning a table. These constraints need to be enforced by the coverage planner during trajectory generation. The quality of a solution is determined by the number of partial solutions as well as the total cost for the end-effector paths. In this work, we define the path cost as the distance traveled by the end-effector in workspace coordinates.

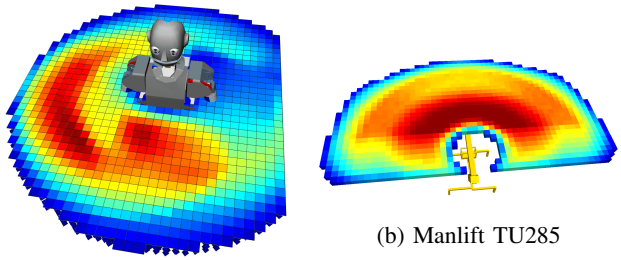
Algorithm 1 shows the high-level logic of the proposed approach. In the first step we uniformly sample points from the target surface. The set of target points $T \subset S_{\text{target}}$ will be used to estimate the progress of the coverage planning. We also keep track of all the points which have already been covered by partial solutions using the set $T_{\text{cov,all}} \subseteq T$. Next, we associate with each pose $\boldsymbol{p} \in G$ the set of probably reachable target points using a precomputed reachability map RM. This relation is stored in $R : SE(3) \to \mathcal{P}(T)$. After this initialization phase the main loop of the algorithm continues until either all target points have been covered or R is empty. During the loop we determine the pose $\boldsymbol{p}_{\text{max}}$ which potentially covers the biggest subset of the remaining target points. Then the coverage planner tries to find a trajectory t which covers as many points in $R(\boldsymbol{p}_{\text{max}})$ as possible. The actually covered points are denoted by T_{cov}. External and internal constraints like collision avoidance and stability requirements are taken into account by the coverage planner. The chosen pose $\boldsymbol{p}_{\text{max}}$ and the covered points T_{cov} need to be removed from R. This achieved by updating every entry $(\boldsymbol{p}', T') \in R$ to exclude the covered points $(\boldsymbol{p}', T' \backslash T_{\text{cov}})$ and removing the entry if there are no reachable points left. An update to the set of covered target points $T_{\text{cov,all}}$ is made by adding the points in T_{cov}. As the last step of the loop $(\boldsymbol{p}_{\text{max}}, t)$ is added to the list of partial solutions. Finally, after the loop has finished, we determine the achieved coverage rate c.

Note that the problem of choosing appropriate placement poses is similar to the set cover problem [24]. Let F be the finite family $\{R(\boldsymbol{p}) \mid \boldsymbol{p} \in G\}$ of finite sets. Then a solution to the set cover problem is the subset $F' \subseteq F$ so that $\bigcup_{S \in F'} S = \bigcup_{S \in F} S \subseteq T$. Our proposed algorithm resembles the greedy approach to this problem since we always choose the entry with the maximal cardinality. This may lead to suboptimal solutions but is necessary since we can only determine the set of actually covered target points after the time-consuming coverage planning.

Algorithm 1: Combined position and coverage planning

Input: S_{target}: Target surface
G: Possible placement poses
RM: Precomputed reachability map
Output: $\mathcal{S}$: List of partial solutions
c: Degree of coverage

$T = \text{samplePointsOnSurface}(S_{target})$;
$T_{\text{cov,all}} = \emptyset$;
$R = \text{determineReachability}(G,\, T,\, RM)$;
$\mathcal{S} = [\,]$;
while $(T \backslash T_{cov,all}) \neq \emptyset \wedge R \neq \emptyset$ **do**
  $\boldsymbol{p}_{\max} = \arg\max_{\boldsymbol{p} \in G}(\|R(\boldsymbol{p})\|)$;
  $(t,\, T_{\text{cov}}) = \text{planCoverage}(\boldsymbol{p}_{\max},\, R(\boldsymbol{p}_{\max}))$;
  $G = G \backslash \{\boldsymbol{p}_{\max}\}$;
  $R = \{(\boldsymbol{p}', T') \in G \times \mathcal{P}(T) \mid T' = R(\boldsymbol{p}') \backslash T_{\text{cov}}\}$;
  $R = \{(\boldsymbol{p}', T') \in R \mid \|T'\| > 0\}$;
  $T_{\text{cov,all}} = T_{\text{cov,all}} \cup T_{\text{cov}}$;
  $\mathcal{S} = \text{append}(\mathcal{S}, (\boldsymbol{p}_{\max},\, t))$;
end
$c = \|T_{\text{cov,all}}\| / \|T\|$;
return $(\mathcal{S},\, c)$;



(a) ARMAR-III

(b) Manlift TU285

Fig. 3: 2D cut of a reachability map for (a) the right arm of ARMAR-III using the joints in the right arm and the hip yaw and (b) the manlift RUTHMANN STEIGER ® TU285. The color of the blocks indicates the manipulability measure with which the end-effector reached the corresponding workspace voxel. Blue means low and red means high manipulability.

B. Reachability Planning

We determine how much of the target surface can be reached from all possible placement poses G with sufficient manipulability. This requires a model of the robot's reachable workspace which can be generated offline. The algorithm used to build the reachability map uses randomized sampling over the configuration space as described in [8] and [9]. Fig. 3 shows a part of the reachability maps for the humanoid robot ARMAR-III and a manlift.

The heuristic for determining the reachability is depicted in algorithm 2. It takes a discrete set of possible placement poses G which need to be defined according to task-specific requirements. The set of target points T approximates the surface to be covered. A precomputed reachability map RM is also required as an input to this algorithm. The heuristic

Algorithm 2: Heuristic for determining the reachability

Input: G: Placement poses
T: Target points to be covered
RM: Precomputed reachability map
Output: R: Reachability relation

$R = \{(\boldsymbol{p}, \emptyset) \mid \boldsymbol{p} \in G\}$;
forall $(\boldsymbol{p}, \boldsymbol{x}) \in G \times T$ **do**
  $v = RM.\text{getVoxelAt}(\boldsymbol{p}^{-1} \cdot \boldsymbol{x})$;
  if $v.quality > \lambda$ **then**
    $R = (R \backslash \{\boldsymbol{p}, R(\boldsymbol{p})\}) \cup \{\boldsymbol{p}, R(\boldsymbol{p}) \cup \{\boldsymbol{x}\}\}$;
  end
end
$R = \{(\boldsymbol{p}, R(\boldsymbol{p})) \mid \|R(\boldsymbol{p})\| > 0\}$;
return R;

Algorithm 3: Coverage planning

Input: $\boldsymbol{p}$: Placement pose
T: Set of target points to cover
$\boldsymbol{q}_{\text{transport}}$: Transport configuration
Output: t: Coverage trajectory
T_{cov}: Set of points which are covered by t

$S_{\text{selected}} = \text{selectGuards}(\boldsymbol{p}, T)$;
$(G, S_{\text{connected}}) = \text{constructGraph}(\boldsymbol{p}, \boldsymbol{q}_{transport}, S_{\text{selected}})$;
$t = \text{approximateTSP}(G)$;
$T_{\text{cov}} = \bigcup_{g \in S_{\text{connected}}} \text{coveredPoints}(g)$;
return (t, T_{cov});

iterates over all pairs of poses and target points $(\boldsymbol{p}, \boldsymbol{x}) \in G \times T$. We check whether $\boldsymbol{x}$ is potentially reachable from the pose $\boldsymbol{p}$. This check is implemented by querying the voxel which encodes the manipulability for the current target point. If the value exceeds a given threshold λ we mark the combination of pose and target point as reachable by updating the entry in the relation R. Finally, poses with no reachable target points are removed from R.

C. Coverage Planning

The coverage planner uses the methods presented [16] in which the problem is subdivided into the coverage sampling problem and the multi-goal path planning. Algorithm 3 shows the overall approach. Using a placement pose $\boldsymbol{p}$, a set of target points T and a transport configuration $\boldsymbol{q}_{\text{transport}}$ it computes a trajectory t which covers the subset $T_{\text{cov}} \subseteq T$ with the robot's end-effector.

First the guard points S_{selected} which cover a subset of T are sampled. A guard point g consists of a configuration $\mathbf{q} \in C$ and a set of points coveredPoints$(g) \subseteq T$ which are covered by the end-effector in this configuration. The guards are sampled in workspace coordinates close to the target points and are checked for collisions and stability. In order to determine a configuration which reaches the sampled point inverse kinematics are employed. In the next step we try to construct a complete graph between the configurations of the guard points and the transport configuration using a BiRRT

Algorithm 4: Selection of guards

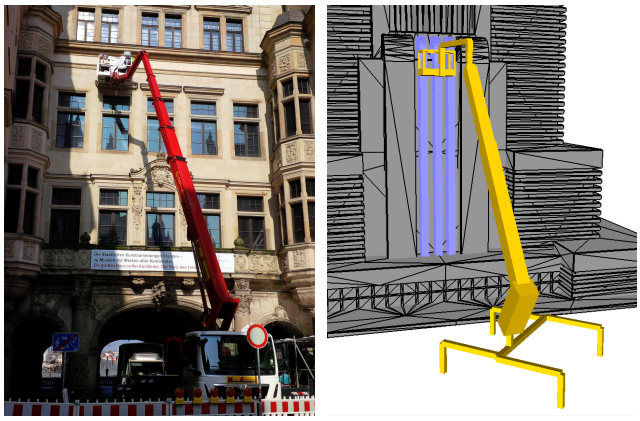
Input: $\boldsymbol{p}$: Placement pose
T: Set of target points to cover
Output: S_{guards}: Set of guard points
$S_{guards} = \emptyset$;
$T_{\text{cov}} = \emptyset$;
$f = 0$;
while $(T \backslash T_{cov}) \neq \emptyset \wedge f < f_{max}$ **do**
  $\boldsymbol{x}_{\text{sample}} = \text{sample}(T \backslash T_{\text{cov}})$;
  $S_{\text{pot}} = \text{sampleGuardsNear}(\boldsymbol{x}_{sample}, \boldsymbol{p}, n_{\text{guards}})$;
  if $S_{pot} \neq \emptyset$ **then**
    $g_{\text{best}} = \arg\max_{g \in S_{\text{pot}}}(\|\text{coveredPoints}(g)\|)$;
    $S_{guards} = S_{guards} \cup \{g_{best}\}$;
    $T_{\text{cov}} = T_{\text{cov}} \cup \text{coveredPoints}(g_{best})$;
    $f = 0$;
  else
    $f = f + 1$;
  end
end
return S_{guards};

motion planner [25]. During this step collision avoidance and stability are ensured. Also the resulting paths are post processed with a probabilistic path shortening algorithm [26] and the final path length in workspace coordinates is calculated. It may not be possible to connect some guard points to the graph G due to constraints. Therefore the subset of connected guard points $S_{\text{connected}} \subseteq S_{\text{selected}}$ is also computed. An approximate solution to the traveling salesman problem on the graph G is calculated by building a minimum spanning tree and doing a pre-order walk on this tree. The cost of the resulting route is at most twice the cost of the optimal solution [16]. This step could be improved by using the Concorde library [27]. Finally, the set of covered points T_{cov} is the union of the points covered by the connected guards $S_{\text{connected}}$.

The method for guard selection is shown in Algorithm 4. The main loop tries to find guards until the requested target points T are fully covered or the guard sampling failed multiple times (threshold f_{max}). Each iteration a random point $\boldsymbol{x}_{\text{sample}}$ is chosen from the remaining target points and potential guards S_{pot} are sampled near this point. The guard sampling generates n_{guards} potential guards in proximity of $\boldsymbol{x}_{\text{sample}}$ using inverse kinematics. From these potential candidates the best guard g_{best} is chosen according to the number of covered target points and added to the result set S_{guards}. If no inverse kinematic solution could be generated the failure counter f is increased. Instead of random sampling a more regularized approach can yield faster results if the target surfaces are mostly flat.

IV. Results

We present results of our approach for manlifts and for the humanoid robot ARMAR-III in simulation. Both scenarios were implemented using the Simox toolbox [28] and evaluated on a mobile Intel quad-core processor running at 2.6 GHz. For the manlift we chose a simple scenario to explain how the different steps of our approach work. The ARMAR-III scenario will be more complex and the analysis of the results more in-depth.



(a) (b)

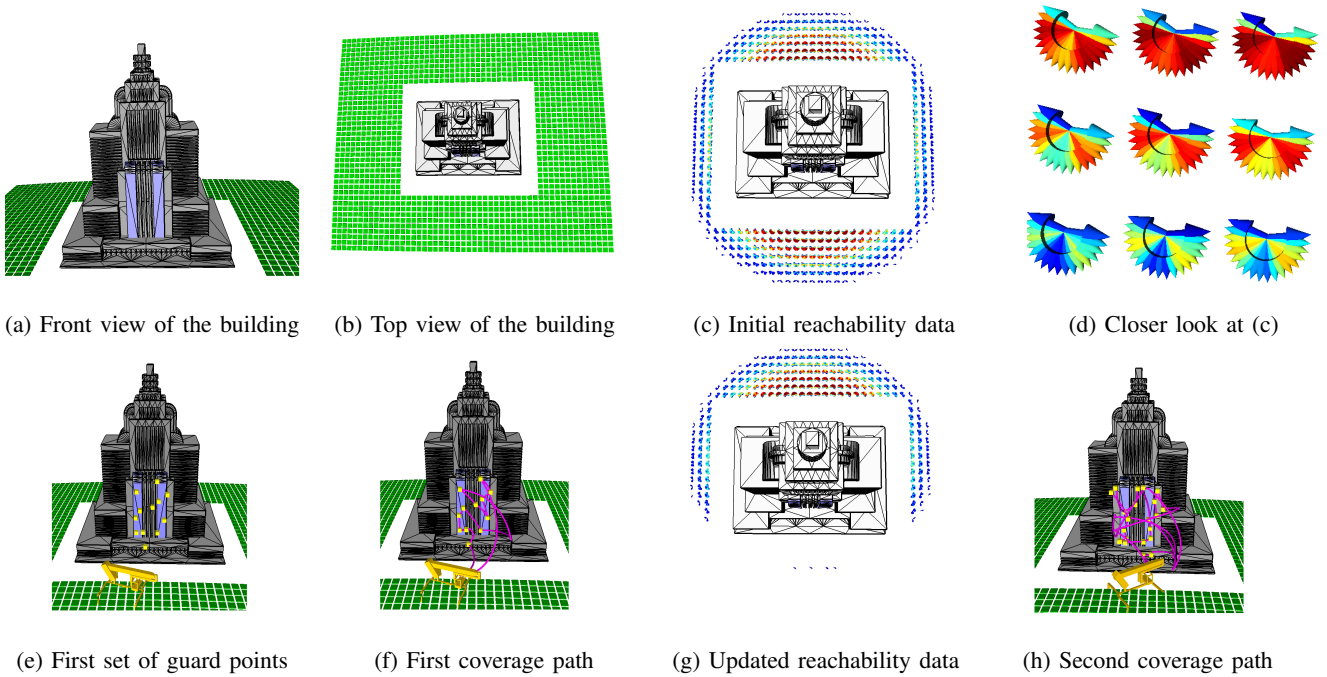
Fig. 4: (a) A photo of the manlift RUTHMANN STEIGER ® TU285 in action; (b) the simplified model of the same manlift consisting mainly of kinematics and basic geometry for collision detection.

A. Manlifts

We tested the proposed algorithm for a simplified model of the manlift RUTHMANN STEIGER ® TU285[1]. In Fig. 4 a photo of a real manlift and the corresponding simplified model are illustrated. This model consists of five revolute joints and one prismatic joint which in combination allow the basket at the end-effector to cover a horizontal range of approximately 20 meters and a height of 28 meters depending on the carried load. One of the revolute joints is constrained by the fact that the basket at the end-effector has to be kept parallel to the ground to guarantee safe transportation.

Fig. 5 shows the different stages of the proposed approach. The task for the manlift is to cover four separate surfaces on the building. Two target areas on the front side of the building are highlighted in blue (see Fig. 5a, the other two target surfaces are symmetrical on the backside). The green squares in Fig. 5a and Fig. 5b represent the discrete placement poses which are used by the algorithm to determine the reachable target points. The resulting reachability information is shown in Fig. 5c. Each pose is represented as a color-coded arrow which represents the position and orientation of the robot. The color blue indicates a very low count and red a high count of reachable target points. A closer look a the individual arrows is illustrated in Fig. 5d. Once the highest rated placement pose is picked, a coverage path for the reachable target points is planned (see Algorithm 1). The first step is to sample guard points which are highlighted as yellow boxes in Fig. 5e. The resulting end-effector trajectory of the coverage path is displayed as a magenta curve in Fig. 5f. Since this

[1] www.ruthmann.de/main.php?target=tu285_steiger&lang=en



(a) Front view of the building (b) Top view of the building (c) Initial reachability data (d) Closer look at (c)

(e) First set of guard points (f) First coverage path (g) Updated reachability data (h) Second coverage path

Fig. 5: Different steps of the combined algorithm for robot placement and coverage planning are shown in simulation for the task of a manlift which covers target surfaces on a building.

path only covers the target surfaces on the front side, the algorithm proceeds to search for a second partial solution. The reachability information is updated by removing the already covered target points (see Fig. 5g). Guard points and a coverage path are computed for the back side of the building and shown in Fig. 5h. Hence, the algorithm is able to solve the given coverage task with two partial solutions.

An overview of the properties of this concrete solution is shown in Table I. Although both sides of the building are symmetrical the randomized sampling approach leads to different results. Since the first partial solution needs ten guard points less than the second one, the path through all guard points is considerably shorter for the first partial solution. To tackle this problem a pruning step for the guard points could be introduced as previously done by [29]. Table II shows the runtime of different parts of the algorithm. The most time consuming subtask is the graph construction because it uses a motion planner to find paths between all the guard points while ensuring collision-free and stable trajectories. The subtask which takes the second most time is the coverage sampling. In order to find suitable guard points the corresponding robot configuration needs to computed using inverse kinematics. The reachability calculation and the TSP approximation for the final route can be computed much faster because complex constraints need not be considered in these phases.

Name	Guard point count	Path length
Partial solution #1	12	70,2 m
Partial solution #2	22	131,0 m
Total solution	34	201,2 m

TABLE I: Properties of the partial solution and the total solution are listed for the manlift scenario. It displays the number of guard points which were sampled and the path length in workspace coordinates.

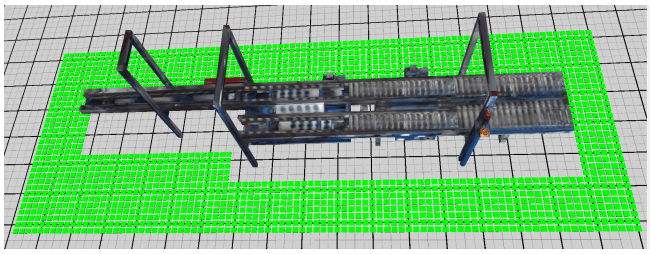
Name	Reachability calculation	Guard sampling	Trajectory planning	TSP approximation
Partial #1	-	6,6 s	35,0 s	0.07 s
Partial #2	-	12,5 s	63,1 s	0.15 s
Total	0.62 s	19,1 s	98,1 s	0.22 s

TABLE II: This table shows the runtime of the different parts in the manlift scenario. Reachability calculation is done only once therefore no times are reported for the partial solutions.

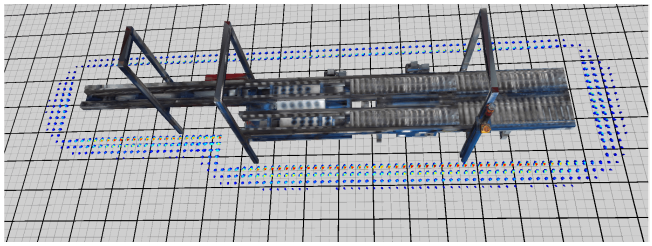
B. ARMAR-III

The second scenario we evaluated uses the humanoid platform ARMAR-III to clean an industrial conveyer belt with its right hand. The kinematic chain consists of eight revolute joints starting at the hip and ending at the end-effector. In contrast to the manlift scenario the area to be covered is much larger than the reachable workspace of the robot. Therefore more placement poses need to be determined in order to complete the task. Fig. 6 shows the scenario, the initial reachability information, the complete solution and coverage trajectories for some partial solutions.

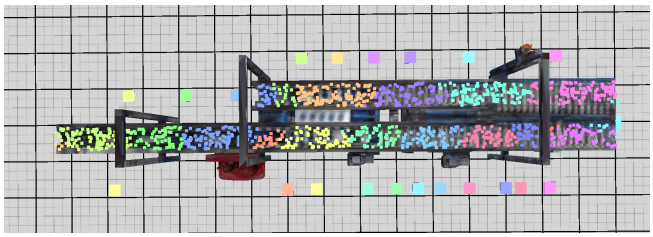
In this scenario we defined 5400 discrete placement poses around the conveyer belt and sampled the target surface



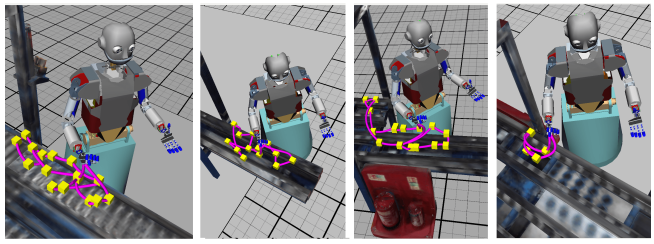
(a) Industrial conveyer belt with green placement poses



(b) Initial reachability indicators before the first partial solution



(c) Chosen placement poses and their covered target points



(d) Coverage trajectories for some partial solutions

Fig. 6: (a) The task for the robot is to cover the two conveyer belts by using the green placement poses. (b) The initial reachability information is used to determine the pose for the first partial solution. (c) The total solutions showing placement poses and the corresponding covered target points in the same color. (d) The partial solutions contain collision free coverage trajectories for the right arm of the robot.

Metric	Average	Standard deviation
Partial solutions	25.8	2.1
Guard points	164.3	3.9
Path length	51.6 m	2.3 m
Achieved coverage	96.7 %	0.6 %
Total runtime	362.5 s	52.5 s
Reachability calculation (runtime)	0.45 s	0.05 s
Guard sampling (runtime)	23.2 s	11.1 s
Trajectory planning (runtime)	330.5 s	43.3 s
TSP approximation (runtime)	0.06 s	0.02 s

TABLE III: Average metrics of 50 runs of the proposed algorithm for the ARMAR-III robot in simulation. Solution and runtime properties are shown.

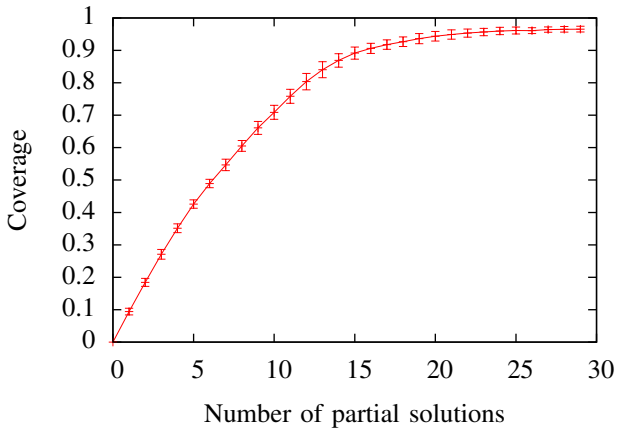


Fig. 7: Relation between the number of partial solutions and the achieved coverage. This graph shows the average coverage and the respective standard deviation over 50 simulated runs.

with 1200 points. We simulated our approach 50 times and computed different metrics for the solutions. Table III shows the important quality and runtime metrics of the computed solutions. It can be seen that the quality metrics (number of partial solutions, total path length and achieved coverage) do not vary much between different runs demonstrating the ability to reproduce consistent results despite the use of randomized approaches. Similar to the manlift scenario most of the runtime is spent during the trajectory planning between guard points. By using the lazy method for mutli-goal path planning in [18] the number of computed trajectories and therefore the total runtime could be improved.

We further evaluated the relation between the number of partial solutions and the achieved coverage. Fig. 7 shows that the rate at which the coverage percentage increases decreases when more partial solutions are added. This is expected behavior since our approach chooses placement poses using a greedy method. Fig. 8 plots the coverage against the runtime of the algorithm. Although the runtime varies between different runs it can be seen that there is a linear correlation between these two variables. Since partial solutions with higher indices cover less target points the coverage planning is simpler and therefore faster compared to earlier partial solutions. This effect compensates for the fact that more partial solutions are needed to cover the same amount of target points in a later timeframe.

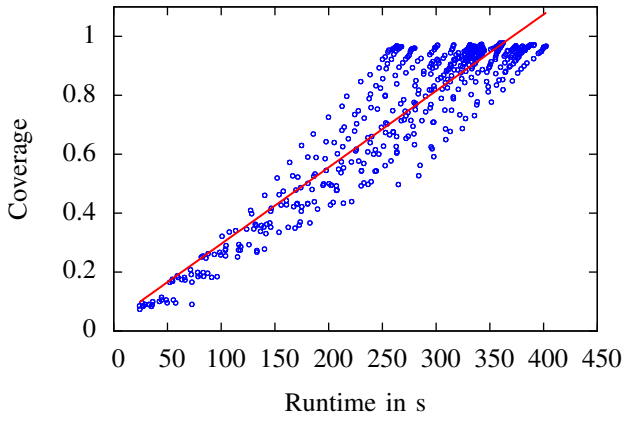


Fig. 8: Relation between the runtime and the achieved coverage. For 20 runs the runtime and coverage after every partial solution has been plotted as blue points. The red regression line shows the linear relation between the two values.

V. Conclusion

In this work we presented an approach for solving the combined problem of robot placement and coverage planning. The use of heuristics for the subproblems of finding appropriate placement poses and computing a coverage trajectory allowed efficient computation of suitable solutions. We evaluated the algorithm in simulation on a humanoid robot cleaning an industrial conveyer belt and on a manlift covering a building's facade demonstrating the broad area of possible applications. Further scenarios may include spray-painting car parts with a mobile robot, inspection of buildings and cleaning kitchen tables.

Currently we consider the number of placement poses and the trajectory costs as optimization criteria. But the travel cost between placements has not been considered yet. One possible extension would be to define a cost function which balances base and end-effector motion.

References

[1] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, "Uniform coverage of automotive surface patches," *International Journal of Robotics Research*, pp. 883–898, 2005.

[2] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *Journal of Field Robotics*, pp. 651–668, 2009.

[3] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6423–6430.

[4] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-III: An integrated humanoid platform for sensory-motor control," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2006, pp. 169–175.

[5] Y. Guan and K. Yokoi, "Reachable space generation of a humanoid robot using the monte carlo method," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 1984–1989.

[6] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3229–3236.

[7] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, "Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009, pp. 55–61.

[8] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1970–1975.

[9] F. Burget and M. Bennewitz, "Stance selection for humanoid grasping tasks by inverse reachability maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5669–5674.

[10] F. Stulp, A. Fedrizzi, L. Mösenlechner, and M. Beetz, "Learning and reasoning with action-related places for robust mobile manipulation," *Journal of Artificial Intelligence Research*, vol. 43, pp. 1–42, 2012.

[11] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *Journal of Field Robotics*, pp. 952–983, 2015.

[12] H. Choset, "Coverage for robotics – a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 113–126, 2001.

[13] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomus Systems*, vol. 61, no. 12, pp. 1258 – 1276, 2013.

[14] D. T. Lee and A. K. Lin, *Computational Complexity of Art Gallery Problems*. New York, NY: Springer New York, 1990, pp. 303–309.

[15] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," DTIC Document, Tech. Rep., 1976.

[16] T. Danner and L. E. Kavraki, "Randomized planning for short inspection paths," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 971–976.

[17] B. Englot and F. S. Hover, "Sampling-based coverage path planning for inspection of complex structures," *22nd International Conference on Automated Planning and Scheduling (ICAPS)*, 2012.

[18] M. Saha, G. Sanchez-Ante, and J. C. Latombe, "Planning multi-goal tours for robot arms," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 3, 2003, pp. 3797–3803 vol.3.

[19] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 4126–4133.

[20] Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of a mobile manipulator," in *Proceedings of the 31st IEEE Conference on Decision and Control*, 1992, pp. 2643–2648.

[21] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 2154–2160.

[22] O. Brock, O. Khatib, and S. Viji, "Task-consistent obstacle avoidance and motion behavior for mobile manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 388–393.

[23] Y. Yang and O. Brock, "Elastic roadmaps—motion generation for autonomous mobile manipulation," *Autonomous Robots*, vol. 28, no. 1, p. 113, 2009.

[24] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, no. 3, pp. 264 – 269, 1974.

[25] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, pp. 378–400, 2001.

[26] S. Sekhavat, P. Svestka, J.-P. Laumond, and M. H. Overmars, "Multilevel path planning for nonholonomic robots using semiholonomic subsystems," *International Journal of Robotics Research*, pp. 840–857, 1998.

[27] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, "Concorde TSP solver," 2006. [Online]. Available: http://www.math.uwaterloo.ca/tsp/concorde.html

[28] N. Vahrenkamp, M. Kröhnert, S. Ulbrich, T. Asfour, G. Metta, R. Dillmann, and G. Sandini, "Simox: A robotics toolbox for simulation, motion and grasp planning," in *International Conference on Intelligent Autonomous Systems (IAS)*, 2012, pp. 585–594.

[29] B. Englot and F. S. Hover, "Planning complex inspection tasks using redundant roadmaps," in *International Symposium on Robotics Research*, 2011.