

Oriented Surface Reachability Maps for Robot Placement

Timo Birr, Christoph Pohl and Tamim Asfour

Abstract—For a robot to perform a grasping and manipulation task, it has to determine possible robot placements in the workspace, from which target objects or environmental elements relevant to the given task are reachable. This work presents a novel approach for finding placements for the mobile base of a humanoid robot in an unknown environment with multiple support planes. We propose a novel type of reachability map – the *Oriented Surface Reachability Map* – that takes inclined surfaces in the environment into account and has the same complexity as reachability maps designed for flat surfaces. The resulting robot placements are not limited to SE(2) but can be applied to arbitrarily oriented planes in 3D space. The proposed method was evaluated in simulation and on the humanoid robot ARMAR-6 in real-world grasping experiments. The results show that a placement can be found for over 80% of the poses that are reachable in complicated, simulated environments, with only a small runtime overhead.

I. INTRODUCTION

Manipulation is a central problem in robotic research. For a successful execution of manipulation tasks, target poses of the objects involved have to be reachable from the robot’s pose. Thus, an integral part of the planning of manipulation actions of a mobile robot consists of finding a suitable placement pose for the robot. This is especially important when robots are used in unstructured environments, where the exact workspace is unknown and areas to stand on are sparse. Humans have an intuitive understanding of their surroundings and can easily estimate, which surfaces provide a suitable support to manipulate objects. For example, they intuitively know to utilize the environment (e.g., ramps, ladders, stairs) to manipulate objects that cannot easily be reached from the ground. Generally, this is hard to achieve for mobile robots, as it is not trivial to distinguish between an obstacle and a plane on the ground that allows the robot to stand on, especially with noisy sensor data. Additionally, navigating an unknown environment with different levels of elevation is generally challenging, as paths that cross steep edges cannot be detected with classical collision detection. Therefore, we investigate a vision-based approach for the mobile base placement of a robot in unknown, nontrivial terrain, that facilitates the application of 2D collision-based path planners to partially planar 3D environments.

In our previous work [1] we introduced the concept of *Inverse Reachability Maps* (IRM) for the efficient query of collision-free placement poses of a mobile robot, based on

The research leading to these results has received funding from the German Federal Ministry of Education and Research (BMBF) under the competence center ROBDEKON (13N14678).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. timo.birr@student.kit.edu, asfour@kit.edu

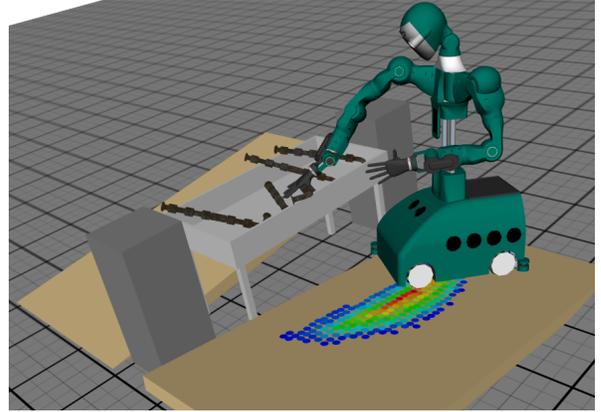


Fig. 1: Simulated robot on an inclined ramp grasping an object with visualized Oriented Surface Reachability Map.

the inversion of pre-computed reachability or manipulability. Even though this facilitates collision-free mobile manipulation in most indoor environments, there is no straightforward extension of the approach to uneven terrain. This significantly reduces the robot’s versatility in cluttered scenes, as all parts of the environment with a higher elevation than the ground level have to be considered as obstacles. While approaches for robot placement on uneven terrain [2] and inclined support surfaces [3] exist for legged robots, these cannot be applied to platform-based robots, which require sufficiently large placement areas and can only navigate on continuous surfaces, in a straightforward way.

This work proposes an approach that facilitates the estimation of feasible platform placements for manipulation tasks that have to be executed by robots with a mobile platform in unknown and unstructured environments. Possible support planes are extracted from point clouds using planar region segmentation that is based on the approach described in [4], which we extend to be more noise resistant. We combine this segmentation with an IRM to retrieve the *Oriented Surface Reachability Map* (OSRM), which allows the sampling of possible placement poses in piece-wise planar 3D terrain efficiently, as it only contains poses that respect the support constraints of the environment. Finally, the robot has to navigate to the determined pose, which can be difficult in situations with multiple levels of elevation. Therefore, we add obstacles for edges that cannot be traversed to the obstacle space, so that conventional path planning algorithms for 2D environments can be used. We implemented our approach using the robot framework *ArmarX* [5], evaluated it in simulation and on the humanoid robot ARMAR-6 [6].

II. RELATED WORK

Most current placement approaches are based on the concept of *reachability maps* (RM), which were initially introduced in [7] under the name *Capability Maps* as a representation of the robot’s workspace. RMs represent a measure of the quality of a pose in the robot’s root frame in terms of reachability with the robot’s tool center point (TCP). An *Inverse Reachability Map*, first introduced in our previous work [1], represents the reachability of a given TCP pose w.r.t. the pose of the mobile base given in the frame of reference of the TCP. In our previous work, we utilized an IRM for the efficient generation of mobile base poses suitable for reaching a given TCP pose in a flat environment. Furthermore, in [8], the IRM is extended to bipedal humanoid robots with the assumption that the environment is flat. The authors in [9] further extend this concept into the *inverse Dynamic Reachability Map* (iDRM) that allows for fast querying of collision-free poses through an occupancy list that links the iDRM and the workspace. The iDRM is combined with its forward version in [2] for the motion planning of the upper and lower body of the humanoid robot Valkyrie. In [3], this approach is extended to also include placements on inclined surfaces. Instead of relying on a position-first-representation when creating a reachability map, the authors in [10] develop an *orientation-based reachability map*, which facilitates the online replacement of tools. In [11], an implementation of the IRM as a database, which also stores joint configurations, instead of a volumetric representation is proposed.

As we address the robot placement problem in unknown scenes, a suitable representation of the environment is needed. To represent environments with not exclusively planar surfaces, heightmaps are often used. For example, the authors of [12] use a *Conditional Random Field* for the spatio-temporal classification of terrain maps based on visual input. An extended version of elevation maps is introduced in [13], which classifies grid cells into four categories and quantifies the uncertainty of the height values with a Kalman filter. To account for the odometry drift of a quadruped robot, a robot-centric, probabilistic terrain elevation map that fuses incoming measurements in the local frame of the robot is described in [14]. The authors of [15] model the surface of a point cloud as a set of curved patches of sufficient size for a foot placement of a humanoid robot and use these for robot control, as well as localization and mapping.

III. ORIENTED SURFACE REACHABILITY MAPS

For a correct robot placement for platform-based mobile robots, the following criteria have to be fulfilled: (a) The robot has to be in steady and even contact with the ground, so the ground covered by the entire platform has to be planar. (b) The n target poses $T \in \text{SE}(3)^n$ of the robot’s TCP have to be reachable from a given placement pose. (c) The robot should not be in collision with the environment. In Section III.A, we give a formal definition of the OSRM that inherently fulfills criterion (a) and describe how we generate robot placement poses from the OSRM that fulfill criterion (b) and (c) in Section III.B.

A. Definition of the Oriented Surface Reachability Map

For the criterion (a) concerning the contact with the ground, the planar region segmentation described in Section IV is used. A planar region segmentation decomposes a given height map $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ into a set of planar and non-planar regions. A planar region $r = (\mathbf{n}, d, W)$ is given by a plane in Hessian normal form $\mathbf{n}^T \mathbf{p} + d = 0$, where $\mathbf{p} \in \mathbb{R}^3$ is position, $\mathbf{n} \in \mathbb{R}^3$ is the normal and $d \in \mathbb{R}^+$ is the shortest distance to the origin of the plane, and a set of grid cells $W \subset \mathbb{R}^2$ that belong to the planar region. Possible robot placement poses at a 2D-position $(x, y)^T$ are constrained to lie on the plane given by the planar region r that is uniquely defined for $(x, y)^T$. Therefore, the height z can be determined by the plane of r . Additionally, the orientation of the robot is constrained by \mathbf{n} , as it can only rotate by γ around the axis facing the direction \mathbf{n} and intersecting the plane given by r in $\mathbf{p}_{x,y} = (x, y, z)^T \in \mathbb{R}^3$. Consequently, the pose of the robot is limited to only three degrees of freedom, which can be described by the tuple $(x, y, \gamma) \in \mathbb{R}^2 \times [0, 2\pi]$. We define the case $\gamma = 0$ to be poses where the projection of the robot root frame’s y -axis $\hat{\mathbf{y}}$ on \mathbb{R}^2 is facing the same direction as the global y -axis. Due to these constraints, $\hat{\mathbf{y}}$ can be uniquely defined in terms of the components of \mathbf{n} .

Now the robot root frame’s x -axis $\hat{\mathbf{x}}$ can be determined by $\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \mathbf{n}$ to complete the rotation matrix $\mathbf{R}_{x,y,0} = (\hat{\mathbf{x}} \ \hat{\mathbf{y}} \ \mathbf{n}) \in \text{SO}(3)$. The orientation of the robot in the global frame for (x, y, γ) can be obtained by rotating $\mathbf{R}_{x,y,0}$ around the local z -axis by γ with $\mathbf{R}_{z,\gamma}$:

$$\mathbf{R}_{x,y,\gamma} = \mathbf{R}_{x,y,0} \cdot \mathbf{R}_{z,\gamma} \in \text{SO}(3)$$

Finally the 6D pose defined by (x, y, γ) is given by

$$\mathbf{p}_{x,y,\gamma} = \begin{pmatrix} \mathbf{R}_{x,y,\gamma} & \mathbf{p}_{x,y} \\ 0 & 1 \end{pmatrix} \in \text{SE}(3).$$

With this representation of robot poses in an environment with multiple support planes, the next step is to determine suitable placements, which ensure the reachability of a given TCP pose. The IRM is a mapping $\text{SE}(3) \rightarrow \mathbb{R}$ and represents the reachability of a given TCP pose w.r.t. a given pose of the mobile base in the frame of reference of the TCP. In our previous work [1], the intersection of the IRM with the xy -plane is used to build the *Oriented Reachability Map* (ORM). Similarly, we intersect the IRM with robot poses on the surface S of the environment provided by the planar region segmentation. The surface S is given by

$$S = \{ \mathbf{p}_{x,y,\gamma} \mid (x, y)^T \in W_{all}, \gamma \in [0, 2\pi] \} \subset \text{SE}(3),$$

where $W_{all} = \{ \mathbf{w} \in W, (\mathbf{n}, d, W) \in R \} \subset \mathbb{R}^2$ is the set of all points on a planar region and R is the set of all planar regions in the environment. The result is the *Oriented Surface Reachability Map* (OSRM) that maps the tuple (x, y, γ) to a reachability value. The OSRM is a three-dimensional distribution like the ORM, which keeps memory usage and computation time small and improves the efficiency in sampling possible poses for platform-based

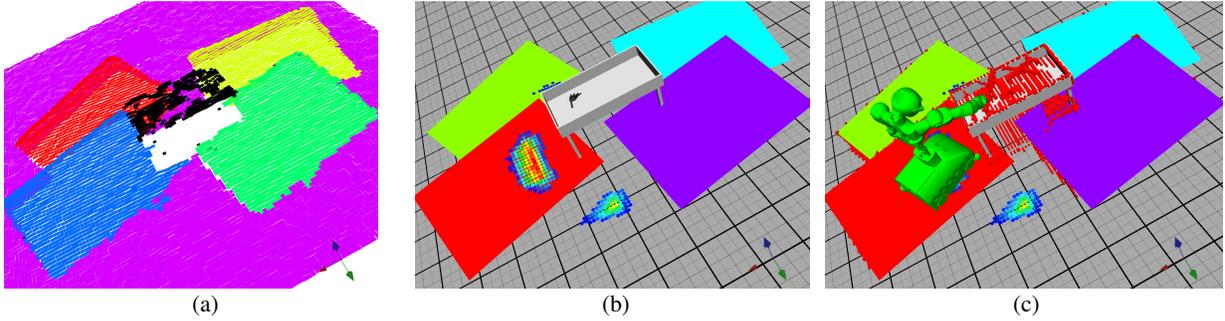


Fig. 2: Planar region segmentation of the scene with black cells representing non-planar regions and any other color representing planar regions (a). The OSRM drawn onto the scene where the reachability is summed over all orientations per position (b). The final robot placement and joint configuration (c).

robots. The algorithm for computing the OSRM is given in Algorithm 1. Similar to [1], the $\min()$ operator can be used to combine the reachability of multiple TCP poses.

The OSRM does not fully respect the criterion (a), as only the point of the robot’s platform specified by the base pose is definitely on the plane. Other parts of the platform could overlap with the edge of the planar region, which might result in infeasible or even dangerous robot placements. Therefore, we approximate the planar region by a polygon $c = (c_1, \dots, c_n)$, $c_i \in \mathbb{R}^2, 1 \leq i \leq n$ by applying a contour detection algorithm described in [16] on the planar region segmentation. Afterwards, we set the reachability of all poses, in which the robot’s base is not entirely within c to zero. The result is an OSRM that contains only placement poses that fulfill criterion (a).

Algorithm 1: Construction of the OSRM

Input: Set of planar regions R , List of target poses in global frame $T \in SE(3)^n$, Inverse Reachability Map IRM , Number of discrete orientations l

Output: OSRM

```

forall  $p_{target} \in T$  do
   $IRM \leftarrow IRM.setPose(p_{target})$ 
  forall  $r = (n, d, W) \in R$  do
    forall  $w = (x, y)^T \in W$  do
      forall  $\gamma \in \{0, \frac{2\pi}{l}, \dots, (l-1) \cdot \frac{2\pi}{l}\}$  do
         $OSRM(x, y, \gamma) \leftarrow$ 
           $\min(IRM(p_{x,y,\gamma}), OSRM(x, y, \gamma))$ 
      end
    end
  end
end
return OSRM

```

B. Generating Feasible Robot Placements

The OSRM can be used to sample possible poses using different strategies. For all strategies, the absolute value of the angle between \hat{y} and the projection of the direction towards the object on the supporting plane should not exceed a γ_{max} to preserve a natural orientation towards the target object. The strategies investigated in this work are:

- 1) *To-Object (TO)*: Selects a 2D-position randomly within the OSRM and orients the robot towards the target object if the summed reachability over all angles is greater than a defined threshold.
- 2) *Maximum-Reachability (MR)*: Selects a pose randomly within the OSRM if its reachability is greater than a defined threshold.
- 3) *Min-Reachability-To-Object (MRTO)*: The initial pose is chosen in the same way as *TO*. For a given position, the angle between the robot and the target object is iteratively incremented in the positive and negative direction, respectively, until the reachability of the OSRM entry is greater than a threshold.

As the IRM discretizes the workspace into voxels, it only provides a probability for the pose within a voxel to be reachable. For this reason, a selected placement pose has to be verified in order to be in accordance with criterion (b) by solving the inverse kinematics (IK) problem for the desired target poses w.r.t a given placement pose.

Finally, to verify criterion (c) we check whether the robot is in collision with the environment using its OctoMap [17] representation. A problem of this approach is that with naive collision checking, the robot is inherently in collision with the OctoMap, as it is always in contact with the ground. For this reason, we remove all occupied voxels from the OctoMap if the grid cell at their center’s xy -position belongs to a planar region r and their center’s distance to the plane of r is small enough. The entire algorithm for determining robot placement poses that meet all criteria is given in Algorithm 2. A visualization of an OSRM, with low reachability in blue and high reachability in red, can be seen together with a successful placement in Figure 2.

IV. PLANAR REGION SEGMENTATION

To obtain the planar regions used by the OSRM, we propose a segmentation based on the method presented in [4]. In this approach, the environment represented as a height map h , is segmented into planar regions. The authors use a region growing algorithm that adds neighboring cells to a region if their local normals have an inner angle smaller than a predefined threshold. Nevertheless, the approach tends to over- or under-segment a noisy scene if planes intersect smoothly. Especially an under-segmentation, i.e., if two

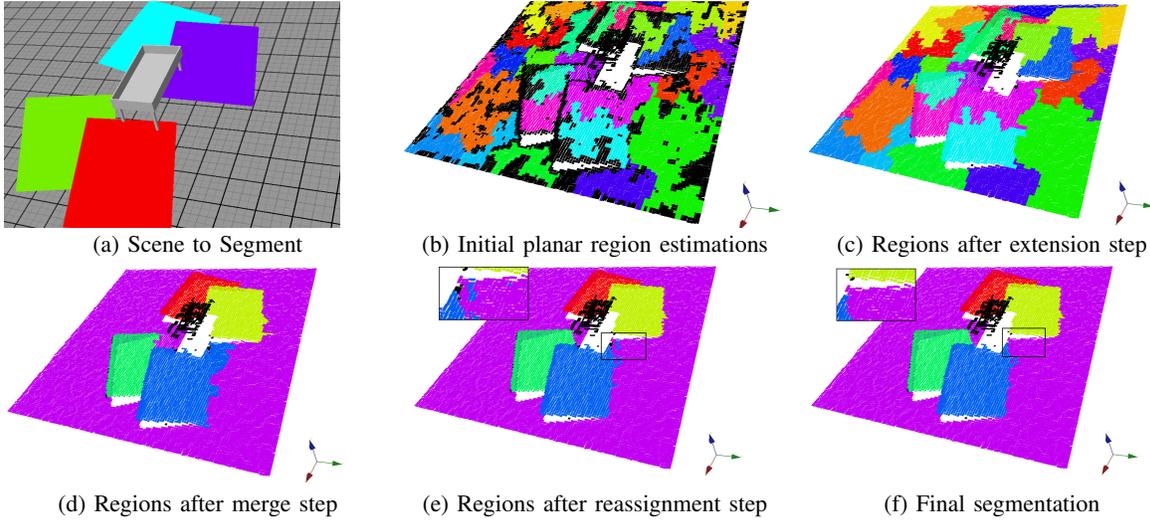


Fig. 3: Processing steps of a planar region segmentation with a resolution of 50 mm in a scene with Gaussian noise with a standard deviation of 25 mm added to the height map.

Algorithm 2: Generation of possible placements

Input: Set of target poses in global frame T ,
 OctoMap Env , height map h , maximum tries
 N , maximum placements M

Output: Set of possible placement poses P

Data: Inverse Reachability Map IRM
 $R \leftarrow calculatePlanarRegions(h)$
 $Env \leftarrow removeRegionsFromOctomap(Env, R)$
 $OSRM \leftarrow calculateOSRM(R, T, IRM)$
 $tries \leftarrow 0$
 $P \leftarrow \emptyset$

while $tries < N \wedge |P| < M$ **do**

$\mathbf{p}_{robot} \leftarrow OSRM.getRandomPose()$

for $\mathbf{p}_{target} \in T$ **do**

$q \leftarrow solveIK(\mathbf{p}_{robot}, \mathbf{p}_{target}, robot)$

if $!q$ **then**

$tries \leftarrow tries + 1$

continue

end

end

if $\neg inCollision(\mathbf{p}_{robot}, q, Env, robot)$ **then**

$P \leftarrow P \cup \mathbf{p}_{robot}$

end

$tries \leftarrow tries + 1$

end

planes are not correctly separated, poses a problem for our approach, as it would not allow an accurate estimation of the supporting planes. Therefore, we propose a modification of the algorithm to increase its robustness.

Let the distance between a grid cell $\mathbf{w} = (x, y)^T \in W$ and the plane defined by a planar region r be

$$dist(r, \mathbf{w}) = |\mathbf{n}^T \mathbf{o} + d|$$

with $\mathbf{o} = (x, y, h(x, y))^T \in \mathbb{R}^3$. Our modified procedure iteratively applies the region growing segmentation with growing thresholds. This initially leads to over-segmentation, so the resulting regions have to be seen as initial estimations for the planes of the planar regions. For the estimations to

be accurate, a region is only accepted if it has a minimum amount of points. The initial estimations are then grown into the surrounding non-planar regions in the *extension step*: A neighboring cell \mathbf{w}_n of a non-planar region is added to the planar region r if $dist(r, \mathbf{w}_n)$ is smaller than a threshold. To reduce over-segmentation, a region r is merged with a neighboring region r_n if their normals are similar and the distance between their centroid and the other plane is small enough. Additionally, in order to reduce artifacts due to wrong assignments of grid cells, we apply a *reassignment step*. For each grid cell \mathbf{w} assigned to r , the cell is reassigned to a neighboring region r_n if $dist(r_n, \mathbf{w}) < dist(r, \mathbf{w})$. Due to noise, some outliers may be added to a planar region using the *reassignment step*, which can be seen in Figure 3e. We correct this by first performing a connected component analysis on the grid of planar regions. We then remove all grid cells that do not belong to the biggest component of a region and add them to the closest neighboring region. The resulting method is computationally more expensive than the original one, but is more robust to noise and allows an easier selection of thresholds for the region growing. An example for the segmentation of a scene with randomly generated planes can be seen in Figure 3.

V. NAVIGATION USING PLANAR REGIONS

To facilitate the execution of an action from a given placement, the robot has to first navigate through an unknown environment with multiple support planes to reach the desired pose. This can be challenging, as traditional motion planning in 2D does not address two main issues: First, if all points in the environment with $z > 0$ are considered to be an obstacle, possible placement poses that are not on the xy -plane cannot be reached. Second, if the elevation difference between points in the environment is not considered, a resulting path could require the robot to traverse a steep edge and lead to a fall of the robot.

In our approach, we use the OctoMap of the environment as the basis for approximating the obstacle space. To address

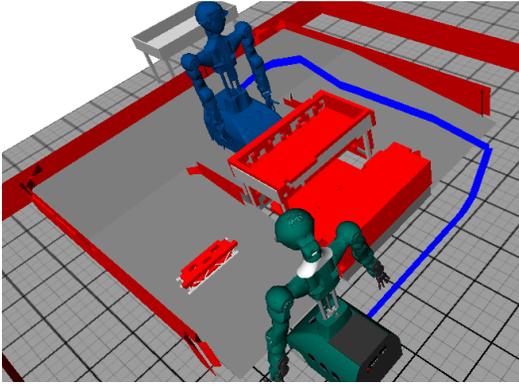


Fig. 4: Environment with collision sets (red), robot, path (blue), and goal pose (blue). The object on the left ramp is recognized by our algorithm and not seen as part of a planar region, so the robot navigates around it.

the first issue described above, the method described in Section III is used to remove all voxels from the OctoMap that belong to a planar region, as they are not obstacles but belong to possible support surfaces.

To deal with the second issue, we model the segmentation in planar regions as a multigraph $G = (V, E)$. In this graph, a vertex $v = (r, c) \in V$ represents a planar region $r \in R$ with the polygon c approximating its contour. An edge $e = (\{v_i, v_j\}, \mathbf{p}_s, \mathbf{p}_e, t) \in E$ represents the adjacency between the regions represented by vertices $v_i, v_j \in V$. It contains the end points \mathbf{p}_s and $\mathbf{p}_e \in \mathbb{R}^2$ of the line approximating their common border and whether it can be traversed with $t \in \{0, 1\}$. To build the graph, we use the polygons approximating the planar regions obtained in Section III. If the polygons are close enough to each other, an edge is added between their corresponding vertices. In a second step, we need to detect all edges between regions where the height of the environment is continuous, and thus, can be traversed. This is the case if the planes of the neighboring regions intersect near their common border.

Finally, all non-traversable edges e can be added to the obstacle space in form of a thin box from \mathbf{p}_s to \mathbf{p}_e . An example for a path computed with a bidirectional RRT using the collision set determined by our approach can be seen in Figure 4. Though a more principled approach would be to use the planar region graph for a hierarchical navigation through the scene, our approach has the advantage that it works with conventional collision-based 2D path planners without adapting the algorithm.

VI. EVALUATION

The main part of the evaluation is done in simulation using randomly generated scenes. To incorporate uncertainty we added Gaussian noise with a standard deviation of 25 mm to the height map. The methods are also tested on the humanoid robot ARMAR-6 [6] to verify that the segmentation also works on noisy real-world data.

We use 5×10^7 random joint configurations to generate the IRM with a resolution of 50 mm in translation and 0.8 rad in orientation. The OSRM that is constructed has the same

resolution in translation but a resolution of 0.175 rad in orientation, which translates to 36 discrete orientations per 2D position. The height map used for the segmentation is generated from an intermediate OctoMap representation of point clouds from external cameras by setting the z -value of each grid cell to the highest z -value of a voxel center within the corresponding grid cell. As the resulting height map has only discrete z -values, we apply a Gaussian filter for smoothing. The OctoMap we use has a resolution of 25 mm and the height map has a resolution of 50 mm. Furthermore, we empirically chose the parameters for the planar region segmentation in a way that resulted in consistent results over all scenes considered.

A. Simulated Experiments

We evaluate our approach by computing placements for grasp candidates on randomly placed objects in a box. Additionally, there were multiple, randomly generated planes added to the environment. To generate ground truth data and estimate whether a grasp was reachable from any valid pose in the environment, we solve the IK problem for poses in a grid around the box. To reduce the number of calculations, only poses that are not in collision with generated planes were checked. For both the reachability estimation and the placement, a $\gamma_{max} = \frac{\pi}{4}$ is chosen.

For all grasps that are estimated to be reachable, placements are generated and the times for the OSRM-construction (t_{RM}) and the entire query consisting of t_a for all queries and t_s for all successful ones is measured. Furthermore, the timing of the stages for calculating the robot placements and the reason for a failure to determine a placement are measured. The results can be seen in Figure 5.

The first set of experiments is done without generating additional planes to compare our approach to an implementation that uses the ORM from [1]. This implementation uses a known environment for collision checking and queries placements similar to the *TO* strategy. As seen in Table I, our approach is marginally less successful in calculating placements using *TO* but is far better using the *MR* strategy. As poses, where the robot is not entirely within a planar region, are removed during the construction of the OSRM, many placements that result in collisions with the environment are already removed, too. This increases the construction time

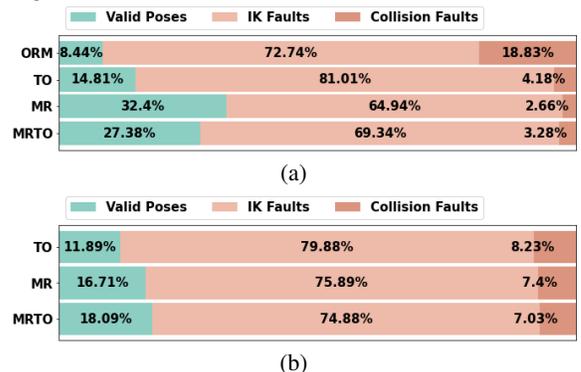


Fig. 5: Failure rates in flat (a) and complex scenarios (b)

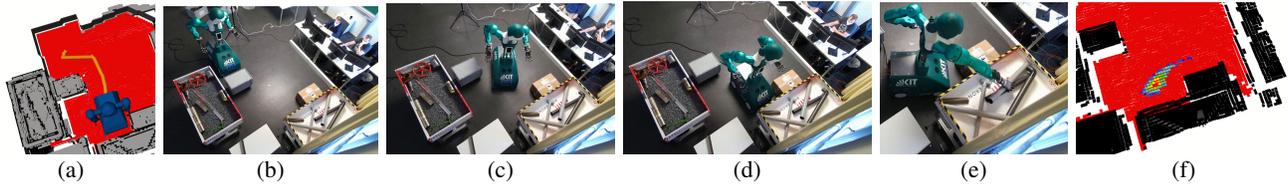


Fig. 6: Experiments on ARMAR-6. (a) shows the planned path in orange, the planar regions in red and the collision set in grey. (b) - (d) show stages of the executed path. (e) shows the grasp with the placement queried from the OSRM in (f).

of the reachability representation, compared to the ORM approach, but reduces the query time overall as there are fewer collision failures, as can be seen in Figure 5a. For reference, using grid search to find a suitable placement takes 2672 ± 3278 ms, whereas not using an OSRM and just placing the robot randomly takes 652 ± 358 ms.

For the second set of experiments, a number of scenes with different complexities is considered. We conducted experiments in six scenarios and evaluated the approach regarding two criteria, the number of generated planes and their different size: small (1 m–1.5 m), medium (1.5 m–2 m) and large (2 m–2.5 m). For each class, 10 scenes were randomly generated and over 500 grasp candidates are used to calculate placements. For all scenes, we evaluated our segmentation

TABLE I: Performance in flat scenario

Method	Success Rate [%]	t_{RM} [ms]	t_a [ms]	t_s [ms]
ORM	89.0	38 ± 17	1816 ± 1415	1788 ± 1319
<i>TO</i>	87.1	398	724 ± 513	538 ± 196
<i>MR</i>	97.5	± 256	515 ± 267	481 ± 151
<i>MRTO</i>	94.6		540 ± 235	495 ± 138

TABLE II: Performance in complex scenarios

Scene	Strategy	Success Rate [%]	t_a [ms]	t_s [ms]
Small 2 planes	<i>TO</i>	72.7	1308 ± 750	920 ± 332
	<i>MR</i>	87.8	1108 ± 915	924 ± 234
	<i>MRTO</i>	84.1	1065 ± 603	857 ± 370
Small 4 planes	<i>TO</i>	55.8	1430 ± 721	870 ± 196
	<i>MR</i>	77.9	962 ± 474	850 ± 213
	<i>MRTO</i>	80.4	954 ± 454	852 ± 214
Medium 2 planes	<i>TO</i>	74.1	1245 ± 748	815 ± 288
	<i>MR</i>	88.5	909 ± 469	752 ± 205
	<i>MRTO</i>	87.0	926 ± 436	776 ± 236
Medium 4 planes	<i>TO</i>	64.6	1345 ± 749	818 ± 289
	<i>MR</i>	86.1	928 ± 534	723 ± 240
	<i>MRTO</i>	81.8	954 ± 496	755 ± 259
Large 2 planes	<i>TO</i>	67.7	1295 ± 731	783 ± 279
	<i>MR</i>	85.8	957 ± 1008	736 ± 194
	<i>MRTO</i>	80.3	936 ± 470	725 ± 208
Large 4 planes	<i>TO</i>	61.6	1325 ± 750	667 ± 245
	<i>MR</i>	82.3	796 ± 455	610 ± 174
	<i>MRTO</i>	80.7	807 ± 380	672 ± 224

method by comparing the orientation of the generated planes to the estimated ones. Additionally, we checked whether the plane that lies within a given cell corresponds to the planar region our method assigned to it. Our improved segmentation is able to correctly predict the correct plane at a given grid cell with an accuracy of 97,8% and an average error in

normal orientation of 0.65° . Our approach finds a suitable robot placement for more than 80% of the grasps estimated to be reachable with the best strategy in any scenario. Generally, as can be seen in Table II, our approach is the least successful in scenes with high complexity and small planes. There are two reasons to explain this. One limitation of our approach is the representation of the environment as a height map with discrete cells, so the accuracy of the segmentation depends on the resolution. On the other hand, we make the assumption that planar regions do not overlap vertically, as only the highest occupied voxel is used to generate the height map used for the segmentation. Consequently, in the experiments, our approach does not find placements, in which the robot is partially under the box. Furthermore, *MR* is more effective than *MRTO* as it selects placements with higher reachability. The *TO* strategy is the least effective – in failures, computation time, and success rate, as it restricts possible placements.

B. Validation on ARMAR-6

We validate our approach on ARMAR-6 in a real-world environment. Therefore, we installed two external static Azure Kinect cameras and fused their point clouds to generate the OctoMap. In the experiments, we set up different configurations of the scene with various positions of obstacles. The results show that obstacles can be avoided in navigation and placement planning and that the robot can successfully execute the grasps using the suggested placements as can be seen in Figure 6. However, we acknowledge a more rigorous evaluation is necessary, as the experiments on ARMAR-6 were only a first proof-of-concept of the approach.

VII. CONCLUSIONS

In this work, we presented an approach that enables a mobile robot to find placements suitable for grasping and manipulation tasks on inclined surfaces in previously unknown environments based only on visual information. Our approach was consistently able to find more than 80% of placements for grasps estimated to be reachable with a comparable execution time to state-of-the-art approaches assuming a flat environment. In future work, the segmentation approach can be extended to segment multi-level surface maps and allow the OSRM to represent vertically overlapping planar regions. A more rigorous evaluation of our approach in real outdoor environments, as well as on other mobile robots should also be conducted.

REFERENCES

- [1] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot Placement based on Reachability Inversion," in *IEEE International Conference on Robotics and Automation*, 2013, p. 1970–1975.
- [2] Y. Yang, W. Merkt, H. Ferrolho, V. Ivan, and S. Vijayakumar, "Efficient Humanoid Motion Planning on Uneven Terrain Using Paired Forward-Inverse Dynamic Reachability Maps," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2279–2286, 2017.
- [3] H. Ferrolho, W. Merkt, Y. Yang, V. Ivan, and S. Vijayakumar, "Whole-Body End-Pose Planning for Legged Robots on Inclined Support Surfaces in Complex Environments," in *IEEE-RAS International Conference on Humanoid Robots*, 2018, p. 944–951.
- [4] P. Karkowski and M. Bennewitz, "Prediction Maps for Real-Time 3D Footstep Planning in Dynamic Environments," in *IEEE International Conference on Robotics and Automation*, 2019, p. 2517–2523.
- [5] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The Robot Software Framework ArmarX," *Information Technology*, vol. 57, pp. 99–111, 01 2015.
- [6] T. Asfour, M. Wächter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, R. Grimm, Y. Zhou, M. Grotz, and F. Paus, "ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real-World Scenarios," *Robotics Automation Magazine (RAM)*, vol. 26, no. 4, pp. 108–121, 2019.
- [7] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, p. 3229–3236.
- [8] F. Burget and M. Bennewitz, "Stance Selection for Humanoid Grasping Tasks by Inverse Reachability Maps," in *IEEE International Conference on Robotics and Automation*, 2015, p. 5669–5674.
- [9] Y. Yang, V. Ivan, Z. Li, M. Fallon, and S. Vijayakumar, "iDRM: Humanoid Motion planning with Real-Time End-Pose Selection in Complex Environments," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 271–278.
- [10] J. Dong and J. C. Trinkle, "Orientation-based Reachability Map for Robot Base Placement," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 1488–1493.
- [11] S. Osswald, P. Karkowski, and M. Bennewitz, "Efficient Coverage of 3D Environments with Humanoid Robots using Inverse Reachability Maps," in *IEEE-RAS International Conference on Humanoid Robots*, 2017, pp. 151–157.
- [12] S. Laible and A. Zell, "Building Local Terrain Maps using Spatio-Temporal Classification for Semantic Robot Localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4591–4597.
- [13] P. Pfaff and W. Burgard, "An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping," in *Field and Service Robotics*. Springer Berlin Heidelberg, 2006, vol. 25, pp. 195–206.
- [14] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic Terrain Mapping for Mobile Robots With Uncertain Localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.
- [15] D. Kanoulas, N. G. Tsagarakis, and M. Vona, "Curved patch mapping and tracking for irregular terrain modeling: Application to bipedal robot foot placement," *Robotics and Autonomous Systems*, vol. 119, pp. 13–30, 2019.
- [16] S. Suzuki and K. Abe, "Topological Structural Analysis of Digitized Binary Images by Border Following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, p. 32–46, 1985.
- [17] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, p. 189–206, 04 2013.