

Bridging the Gap between Task Planning and Path Planning

Franziska Zacharias, Christoph Borst, Gerd Hirzinger
Institute of Robotics and Mechatronics
German Aerospace Center DLR
Oberpfaffenhofen, Germany
Email: Franziska.Zacharias@dlr.de

Abstract—Autonomous service robots have to recognize and interpret their environment to be able to interact with it. This paper will focus on service tasks such as serving a glass of water where a humanoid two-arm-system has to acquire an object from the scene. A task planner should be able to autonomously discern the necessary actions to solve the task. In the process, a path planner can be used to compute motion sequences to execute these actions. To plan trajectories, the path planner requires a pair of configurations, the start and the goal configuration of the robot, to be provided e.g. by a task planner.

This paper proposes a method to autonomously find the goal configurations necessary to acquire objects from the scene and thus makes an attempt to bridge the gap between task planning and path planning. The method determines where to grasp an object by analyzing the scene and the influence of obstacles on the intended grasp location. For the case where the goal object can not be grasped due to obstructing obstacles, a solution is proposed.

I. INTRODUCTION

A main goal in the development of autonomous service robots is, that they can be commanded by the human in a very natural way, like e.g. "Serve a glass of water!". Therefore a robot like the DLR robot "Justin" (fig. 1) must be able to decompose the task into a set of subtasks by itself. The subtasks and their ordering depend on the commanded task and on the scene. For serving water e.g. some command dependent subtasks are: finding a water bottle, grasping it, localizing a glass and pouring water in the glass. Depending on where the objects are in the scene, some movements by the two arm system have to be performed.

This paper will focus on the problem where to grasp an object. At first glance the problem seems solved by using a state of the art path planner to get the arm to the desired grasp position. But where is the desired grasp position? With a simple device like a parallel jaw gripper, the small number of possible grasping positions can be brute force sampled [1]. But for a dexterous hand like the DLR Hand II an almost infinite number of possible grasps can be generated [2], [3]. Hence, it is desirable to parameterized the grasp planner in a way that kinematic constraints of the arm and geometric constraints of the scene (obstacle objects) are taken into account. Unfortunately, information about these constraints cannot be extracted from most state of the art path planners although they obviously take them into account.

Two classes of path planners can be distinguished, the



Fig. 1. The DLR humanoid two-arm system "Justin", the target system for the task planning approach.

first class plans in configuration space and the second class plans paths in cartesian space. In configuration space, global probabilistic path planners as presented by Lavelle et al. [4] and Kavraki et al. [5] are state of the art. They are probabilistically complete, i.e. guarantee to find a solution whenever one exists provided enough samples. Local path planners [1], [6] also working in configuration space can be a good alternative to using global path planners. While they are not complete, they work well in scenarios composed of only few objects. Furthermore, local path planners have the advantages of being faster [1] and of generating smoother paths than global planners.

A planner in cartesian space like the one presented by Iossifidis et al. [7] plans the tool center point (TCP) movement. Compared to planners working in configuration space, the influence of the scene can directly be incorporated e.g. via an attractor dynamics approach. Incorporating the kinematics of a redundant robot in cartesian space is difficult, thus TCP planners in general are not complete. Nevertheless they are able to plan straight paths in cartesian space to the goal which are more intuitive for humans to predict than straight paths in configuration space which result in curved motions in cartesian space.

Everyone of these approaches needs a collision free, reachable start and goal configuration of the robot arm to be able to plan a geometric trajectory. The start configuration in general

is known. When intending to grasp an object, there is a huge number of possibilities where and how to grasp it. Thus there is an undetermined number of goal configurations, one of which has to be chosen for use by the path planner. Finding the best grasp might lead to an unreachable arm configuration while defining a certain arm configuration might lead to bad grasps. This paper tries to analyze the problem of computing a TCP position and orientation that is reachable and can be used to accomplish a given task (fig. 1).

Lozano-Perez et al. [8] recognize that to solve pick-and-place tasks, choosing how to grasp an object is crucial. They state that the choice of a good target position depends on the environment. The hand position and resulting arm configuration have to be free of collision and reachable. In principle, also the availability of a path between start and goal configuration should be checked. Given the fact that there can be a huge number of goal configurations and that a complete path planner's worst-case time-complexity is exponential, checking the availability of a path for each possible goal configuration is computationally too expensive. The question of the existence of a path will not be addressed in this paper, but the goal configuration will be chosen in a manner that is beneficial for path planners and paths in general are found easily. Lozano-Perez et al. operate with a parallel jaw gripper only in the domain of polyhedral models and thus have only a constrained set of possible grasps for an object. They pick a grasp location and try to plan a path, repeating the process if no path could be planned to the grasping location. Using the service robot "Justin" we have a highly redundant robot with dexterous hands. Thus the set of grasps for an object is infinite and the brute force method of iterating over the whole set of grasps is not feasible.

Wösch et al. [1] use an 8-DOF robot arm and a parallel jaw gripper. They define by hand a set of approach TCPs, sample the TCP orientation and check whether there is one among them that results in a collision free robot configuration. Depending on the number of TCPs, they report that the determination of an optimal approach TCP in this brute force manner takes longer than the actual path planning. Thus a feasible solution is needed to close this gap.

Iossifidis et al. [7] try to attain straight line end-effector trajectories. An attractor dynamics approach to trajectory formation is used, which influences the velocity as well as two angles characterizing the TCP orientation. If obstacles obstruct the direct path to the target object, they try to grasp the object from above. Obstacles are avoided by choosing an inverse kinematics solution where the elbow position is high. Although they report that the trajectories are collision free in all experiments, this fact is not guaranteed. Using this method to grasp a bottle for a pouring task, before being able to pour, the bottle has to be regrasped.

The approach presented by Terasaki et al. [9] again operates with a parallel jaw gripper on objects taken from an assembly scenario. Due to their choice of objects and gripper, the number of possible grasps is limited. The target object is analyzed to obtain a set of candidates for grasping positions.

The space around each grasping position is described by a so-called open space characterization which represents approach directions by pyramidal slices labeled with a measure of the available free space in that direction. This characterization is used to select among the grasp candidates. If no grasp could be found, the method stops.

Observing the human grasp process it becomes evident that objects are grasped only in a certain area with respect to the human's own position and orientation. The grasp placement is influenced by the presence of obstacles [10]. If the target object is obstructed, it is also an option to displace the obstructing obstacles before applying a safe grasp to the target object. We integrated these aspects in the methods described in sections II and III.

To plan trajectories we use a local path planner [6] since for scenes of low complexity local path planners tend to be fast, provide smooth trajectories and can cope with changing scenes. We assume that we have a model-based object recognition that can label objects. Section II presents a method to analyze the scene using object recognition and to decide where to grasp the object in the presence of obstacles. Section III deals with the issue of how to react if the target object is inaccessible. Having solved the problem of placing a grasp and determining a collision free reachable goal configuration, all prerequisites of a path planner are met and an autonomous task planning process can schedule the path planner when composing the action plan to solve a given service task.

Section IV presents some experiments for scenarios with a varying number of obstacles. Section V concludes the paper and outlines future work.

II. OBSTACLES INFLUENCE THE TCP PLACEMENT

Human grasps depend on and adapt to a given environment [10]. In this section we present a method that chooses where to grasp an object depending on the robot's kinematic structure, surrounding obstacles and the position of the robot relative to the target object. It models a right arm behavior since we intend to use the method on the two arm system "Justin". Modeling right and left arm behaviors is reasonable in view of the human model and to avoid interference between the two arms (fig. 1). To concentrate on the problem at hand "Where should the robot place a grasp?", this first version of our algorithm projects the 3-D scene onto the table surface and bases its decisions on the resulting representation.

To determine an area of preferred grasping directions two approaches are proposed. Using a brute force approach, we sample the circumference of the object to be grasped and compute the inverse kinematics for each sampled position. We then search for the largest connected region of valid inverse kinematics solutions and search this stretch for good grasp approach directions (fig. 2).

The second method is motivated by the observation that humans grasp an object usually in a certain area with respect to themselves. Therefore the heuristic approach we propose depends on the robot's position. The object's coordinate system is oriented so that its y-axis is determined by the line

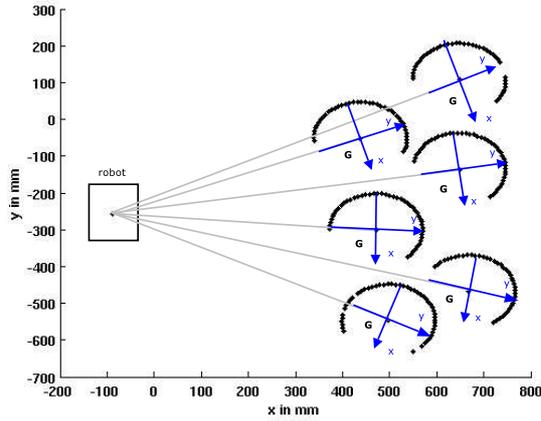


Fig. 2. A scenario (see fig. 3) where 6 objects are distributed across a table is analyzed concerning kinematic reachability. The object footprints are represented as circles. The inverse kinematic was able to find solutions at those locations where to circle has gaps. The circumference of the object was sampled with 5 degrees resolution. The object frames (blue) resulting from the heuristic method are sketched into the plot.

connecting the position of the robot base and the center of the target object (fig. 3). Grasps are searched in the 4th quadrant of the resulting object coordinate system.

Fig. 2 compares both approaches. Six objects, their circumference represented as circles, are distributed across the table (robot-to-table scenario see fig. 3). The inverse kinematics found solutions at those locations where the circle has gaps. The object coordinate systems resulting from the heuristic approach are sketched by hand into the plot. Since the shown grasping approach directions and TCP orientations correspond to a robot arm-hand system configured as a right arm, the solutions for the inverse kinematics are mostly available on the right hand side of the objects as seen from the robot point of view. We observe that the heuristically determined region covers the biggest connected stretch where inverse kinematic solutions are available.

The solutions lost through applying the heuristic mostly are farther away from the robot. If due to obstacles no grasp can be found in the region specified by the heuristic, in most cases no collision free solutions for the inverse kinematics will be found for grasps in regions farther away from the robot. In those situations the robot arm will collide with the objects that already obstructed the regions nearer to the robot. The ability to reach around obstacles in regions far away from the robot is kinematically limited. The heuristic method is clearly faster than the brute force method. Furthermore, it is evident that the brute force method does not scale to 3D.

In the following, we will use the heuristic method to determine the area to search for grasps. Figure 3 shows the DLR robot "Justin" and a table set with objects. A bottle is represented by the green object. The robot tries to grasp it in the area labeled G , according to the heuristic method presented above. Once the coordinate system of the target object is oriented according to the heuristic, the remaining movable obstacles

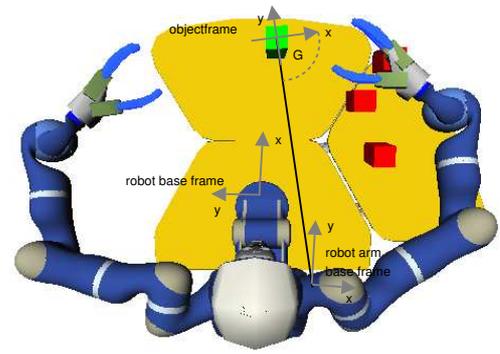


Fig. 3. "Justin" approaches the green target object. The orientation of the target object frame depends on the robot position. The heuristic method aims at placing the grasp in the region outlined by the dashed quarter of a circle.

are transformed into the target object frame and their influence on the grasp placement is determined.

A potential field approach could in principle be used to deflect the grasp approach direction of the TCP. We refrained from using potential fields for a number of reasons. A potential field approach incrementally draws the robot towards the goal using gradient descent along the current potential. While this is acceptable for mobile robots which use the potential field to e.g. generate velocities, we need quick decisions concerning only the grasp approach direction. The actual motion of the robot arm is planned using a path planner. Moreover, Borenstein et al. [11] reported problems with potential fields when the clearance between obstacles was small. The resulting repulsive forces could be such that the robot was pushed away although it would be able to pass. Potential fields thus would need to be finely tuned to guarantee geometrically consistent decisions. Since passing objects in close vicinity is essential in manipulation tasks, we chose to adapt another approach.

The Borenstein's vector field histogram method [11] presents an alternative to using potential fields for collision avoidance. A polar histogram H of certainty values is built around the robot's momentary position. H is comprised of n angular sectors of width α . Each sector i corresponds to a direction $i \cdot \alpha$, $i \in [1, n]$ and represents the polar obstacle density in that direction. The robot then selects that sector i from all polar histogram sectors, where the obstacle density is smallest and drives in that direction. We adapt the approach of Borenstein et al. to our problem. The polar histogram H is now centered on the target object, oriented according to its coordinate system (see fig. 3). All other objects are considered to be obstacles. Only the region G is sampled with stepwidth α to generate candidate directions from which to approach the object. To compute the obstacle density values for a sector, the concept of obstacle influence is introduced. This replaces Borenstein's functions for density computation. Each obstacle has a region of influence. The region of influence should be such that the robot tries to grasp the target object from a direction with as much free space as possible to operate in. Considering fig. 4 (left), valid grasps for the target object (empty black

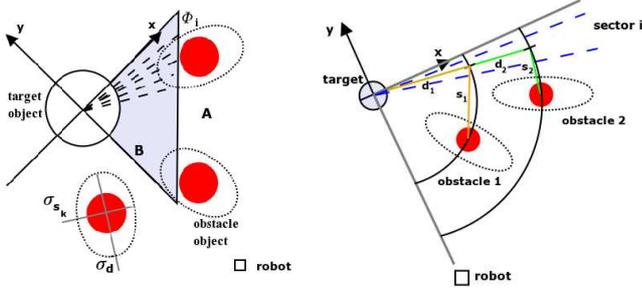


Fig. 4. (left) The region of influence (dashed ellipses) of obstacles (filled red circles) is illustrated. The goal object (empty black circle) is grasped in the shaded region. (right) The values used to compute the second half of eq. 3 are visualized. The position of the robot is symbolically sketched in both figures.

circle) should be placable in region B given enough free space while region A should be preferred since it is easier to approach. Easily reachable grasps ensure the existence of a path to the grasp and are beneficial for a path planner. For these reasons, we describe regions of influence by bivariate gaussian functions oriented towards the goal object (fig. 4 (left)). The influence p_{oi} of a single obstacle on a point in sector i characterized by d_k and s_k (fig. 4 (right)) is computed using the standard deviations σ_{s_k} and σ_d describing the extension of the bivariate gaussian, the distance d_k from the target object border to the currently considered obstacle k , and the distance s_k in sectors between the center of the obstacle and the current sector i . Eq. 1 states the gaussian function. The choice of the standard deviation σ_d in the direction of the target object depends on the size of the robot hand and is the same for all obstacles k . Given that we want to grasp the target object (fig. 4) in region B , the robot hand must fit into the clearance. We set $\sigma_d = 0.175$ m corresponding to the approximate width of the hand. The second standard deviation σ_{s_k} corresponds to the sectors spanned by an obstacle.

$$p_{oi}(d_k, s_k, \sigma_{s_k}) = \frac{1}{2 \cdot \pi \cdot \sigma_d \cdot \sigma_{s_k}} \cdot e^{-\frac{\left(\frac{d_k}{\sigma_d}\right)^2 + \left(\frac{s_k}{\sigma_{s_k}}\right)^2}{2}} \quad (1)$$

To represent the grasping preference, the influence p_g of the goal object is described as in eq. 2, representing a preference towards the middle of the shaded region of fig. 4 (left). The parameter p_b describes the base influence value exercised by a target object and is set to 0.02 in our experiments. The shaded region is sampled with a granularity α to obtain candidates for approach directions. The variable i describes the sector currently examined while the corresponding candidate approach direction is $\phi = i \cdot \alpha + \frac{\alpha}{2}$ with $i \in [0, \frac{90^\circ}{\alpha}]$, $i \in N$.

$$p_g(i) = \begin{cases} p_b \cdot \left(\frac{45^\circ - \phi}{45^\circ}\right) & \phi = \alpha \cdot i + \frac{\alpha}{2} \in [0^\circ, 90^\circ) \\ p_b & \text{otherwise} \end{cases} \quad (2)$$

The influences of all obstacles are mapped onto a one-dimensional representation of the target object's environment, the so-called polar histogram H . Each sector i accumulates influence values from a set of obstacles O . The influence values generated by the target object expressing the grasp

preferences and the influence of obstacles are combined to characterize each sector i (eq. 3). The resulting value $p(i)$ indicates how accessible sector i and thus the corresponding approach direction ϕ is for the robot hand. Equation 3 results in a smooth mapping of obstacle influences and the grasping preference to 1D.

$$p(i) = p_g(i) + \sum_{k \in O} p_{oi}(d_k, s_k, \sigma_{s_k}) \quad i \in [0, \frac{90^\circ}{\alpha}) \quad (3)$$

Figure 4 (right) visualizes the variables for accumulating obstacle influences p_{oi} for sector i for two obstacles. The distances in sectors s_1 and s_2 are shown from the sector of the obstacle to the sector currently considered. Furthermore the distances d_1, d_2 along the x-axis of the obstacle's coordinate frame are drawn into the sketch. The resulting accessibility value is obtained by inserting all variables in equation 3. We measure the individual obstacle influence at the points indicated using the assumption that if the robot hand is allowed to pass this point where the obstacle has the maximum blocking extension, it can also pass the remaining part of the sector.

To decide where to grasp the object, we search the approach direction ϕ with the minimum total influence value $p(i)$. If the influence value of sector i is below the accessibility threshold t , the target object is assumed to be unblocked and we try to grasp the object in that region. This position should then be supplied to a grasp shaper which shapes the grasp aperture and finger configuration to the available free space and the target object.

Should the influence value of every approach direction ϕ exceed the accessibility threshold t , the goal object is considered to be blocked. Sectors are furthermore considered blocked, if no solution for the inverse kinematic of the robot arm is available. In the next section we expand the presented method to deal with blocked objects.

III. SAFELY APPROACHING BLOCKED OBJECTS

If the target object is blocked by obstacles, especially when working with uncertain vision information, the safest way to grasp the target object is to clear some space by removing obstacles beginning with those nearest to the robot. Modifying the scene to suite ones needs is generally termed rearrangement planning [12].

To determine which object can be removed, the scene is analyzed with respect to the influence of object removals on the graspability of the target object. Each obstacle's accessibility is determined by the approach described above. A tree is built to find a sequence of valid obstacle removals. The tree building process stops as soon as the target object becomes accessible by one sequence of object removals. Tree nodes of the class leaf are inserted when the target object becomes accessible. They terminate a valid solution path. To determine an optimal removal sequence that path to a leaf is searched, that results in a minimum value for the accessibility measure of the target object. Thus the removal tree fullfills a function similar to the precedence graph combined with the permutation net proposed

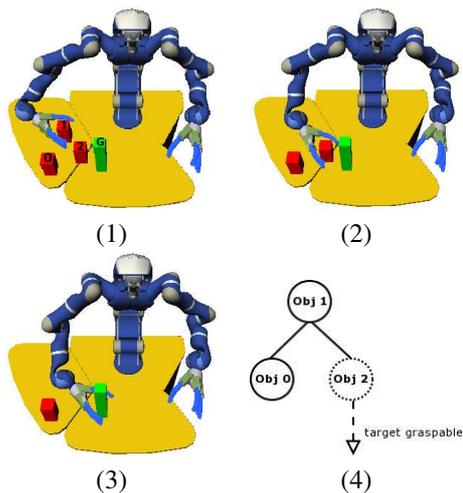


Fig. 5. The target object T is not accessible without removing some objects. The task planner builds a tree to decide which objects to remove. Leaf nodes terminating the removal of objects are marked by dotted circles.

by Ben-Shahar et al. [12] in determining the sequence in which to remove the objects. Ben-Shahar et al. use the precedence graph in planning rearrangement tasks for a simple 2D robot. All objects scheduled for removal appear in the corresponding manipulation path [13] that defines a sequence of transfer and transit movements. In the experiments presented in this paper, obstacles are removed to an appropriate position and the goal object is grasped.

IV. EXPERIMENTS

The methods presented in the previous section were integrated to form a module we call the task planner. It parameterizes the Baginski path planner [6]. For our experiments we used the DLR humanoid two-arm-system "Justin". The complete system has 43 degrees of freedom (DOF) and consists of two arms with 7 DOF each, two hands with 12 DOF each, a PanTilt unit for the head with 2 DOF and a torso with 3 DOF as well as one passively coupled link. "Justin" is mounted on a table set with objects. The following experiments were carried out in simulation. Objects on the table are represented by bounding volumes. This is necessary to use the setup as input for the path planner. Objects are assumed to be rotation symmetric, recognized and labeled.

For the first experiment the scene setup is as shown in fig. 5(1). Obstacles are colored red and labeled 0-2. "Justin" has the task to grasp the green object labeled T in the preferred sector as described in the previous section. As can be seen in fig. 5(1) the target object T is not accessible as accessibility was defined in section II. The task planner analyzes the scene and computes a tree of objects to remove (fig. 5(4)). In the initial set up, only object 1 is accessible and inserted into the removal tree. On the second tree layer object 1 has been removed. Now object 0 and object 2 are accessible but only the removal of object 2 results in the target object becoming accessible. Thus the task planner computes that to reach the target first object 1 and then object 2 have to be removed.

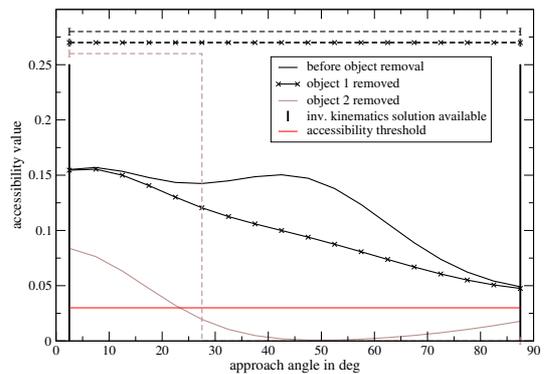


Fig. 6. The accessibility values $p(i)$ for experiment 1 are plotted across the heuristically determined approach angles. The red line represents the accessibility threshold $t = 0.04$. The black bars mark the area where an inverse kinematics solution is available. The dashed lines mark for each valid inverse kinematics solution, if the robot arm would be in collision. Those intervals are free of collision where the dashed line falls to zero.

For each object the computed approach direction and robot configuration are visualized in fig. 5(1-3). The polar histogram H of accessibility values for region G is shown in fig. 6. The graph shows the accessibility of the target object in the situation at the beginning of the experiment and after removing obstacle 1 and obstacle 2. The angles (in deg) in the polar histogram correspond to angles measured clockwise from the positive x-axis of the target object coordinate system from 0 to 90 degrees (compare fig. 3). The removal of object 1 and 2 results in a constant lowering of the consecutive curves till the brown curve falls below the accessibility threshold and the target object can be grasped. The vertical bars border the region where an inverse kinematics solution was available for the target object in the respective sector. Furthermore, the collision information after an object is removed is illustrated by dashed lines. Their color indicates the part of the experiment they belong to. Those region are free of collision where the dashed line is zero. It can be seen that only after the removal of objects 1 and 2 there are collision free goal configurations. The point where the brown curve falls below the accessibility threshold is nearly identical to the point where collision-free configurations become available. Since the configuration belonging to the approach angle with the minimum accessibility value is chosen, the presented method could be used as a fast heuristic alternative to replace the collision tests.

In the second experiment (fig. 7) the scene is much more cluttered. The green object to be grasped is surrounded by obstacles and not accessible. Automatically a valid sequence of object removals is computed to clear sufficient space to grasp the target object using the algorithms presented in sections II and III. Removals begin from the outer skirts and move inwards. Fig. 8 presents an analysis similar to fig. 6. The plotted collision information shows that there are collisionfree configurations when choosing approach angles between 42° and 62° , i.e. between object 3 and 4. While these are valid goal configurations, they are very hard to reach for path planners

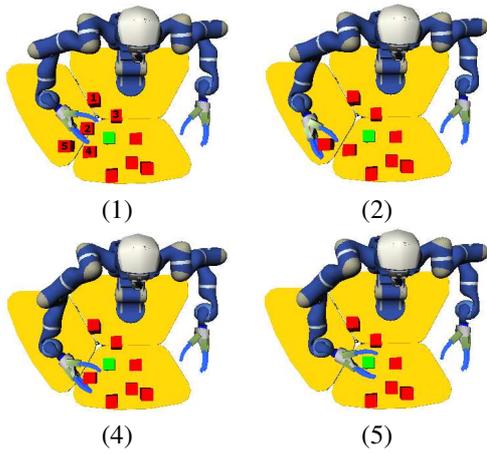


Fig. 7. The green object is to be grasped but is inaccessible. A space is cleared automatically to get to the object.

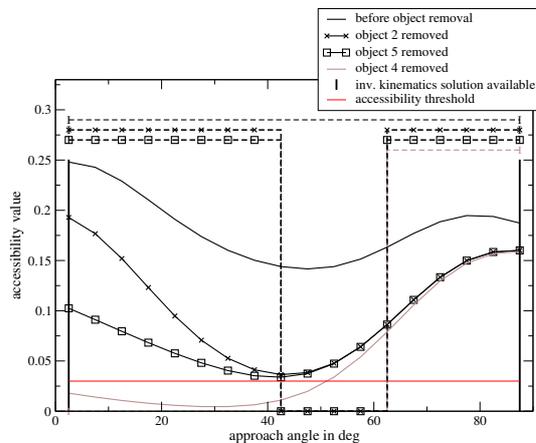


Fig. 8. Accessibility and collision values for experiment 2.

since they belong to the class of narrow passage problems. The presented method continues to remove objects and thereby creates a situation that is beneficial for path planners.

V. CONCLUSION

In manipulation planning as by Alami et al. [13] and rearrangement planning as by Ben-Shahar et al. [12] a task is given as a set of start and goal positions of every robot and object. Manipulation and rearrangement planners compute valid sequences of transit and transfer tasks to achieve a task as specified above. A task in the context of this paper is phrased more general, e.g. grasp bottle x . The main aim of this paper was to provide a method to automatically find a goal configuration to achieve the task. Thus we enabled a task planner to parameterize a path planner. The object to be grasped and the other scene objects influence the computation. As a first step we addressed the problem in 2D by allowing the TCP to be deflected by obstacles on an arc around the z -axis of the target object coordinate system. If the target object is not accessible, the scene is modified. The proposed method is used to determine the precedence of objects should a rearrangement

of the scene be necessary to solve the task. The method is also able to cope with an increased number of objects (fig. 7).

In future work the method will be expanded to work on the 3D scene directly instead of on a 2D projection. Trivariate gaussians can be used and a part of a sphere can be sampled. Clearly, with the added dimension the computational complexity rises. While the extended method could still be applied to simple scenes, it would probably not be feasible for very complex scenes as presented in fig. 7. Thus an integrated analytic solution that is computationally feasible is needed. This method should determine a desirable region to place grasps and filter the huge amount of grasps according to criteria such as the freeness of collision, the possibility of finding stable grasps and kinematic reachability.

When interacting with humans a service robot should act so that humans feel safe around it. In the opinion of the authors it is therefore important that the actions of a service robot do not seem awkward to users and that users are able to predict them to some extent. Thus a right arm behavior was shown in the experiments. In future analysis criteria measuring how natural grasp and arm positions are should also be taken into consideration.

REFERENCES

- [1] T. Woesch and W. Neubauer, "Grasp and place tasks for domestic robot assistants," International Workshop on Advances in Service Robots, Stuttgart, Germany, May 2004.
- [2] C. Borst, M. Fischer, and G. Hirzinger, "A Fast and Robust Grasp Planner for Arbitrary 3D Objects," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'99)*, Detroit, Michigan, May 1999, pp. 1890–1896.
- [3] A. T. Miller and P. K. Allen, "Grasplit!: A versatile simulator for robotic grasping," *IEEE Robot. Automat. Mag.*, vol. 11, no. 4, pp. 110–122, dec 2004.
- [4] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Oct. 1998, TR 98-11, Computer Science Dept., Iowa State University.
- [5] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 566–580, Aug. 1996.
- [6] B. Baginski, "Motion planning for manipulators with many degrees of freedom - the bb-method," Ph.D. dissertation, Technische Universität Muenchen, aug 1998.
- [7] I. Iossifidis and G. Schöner, "Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, USA, Apr. 2004, pp. 4295–4300.
- [8] T. Lozano-Pérez, J. L. Jones, E. Mazer, and P. A. O'Donnell, "Task-level planning of pick-and-place robot motions," *IEEE Computer*, vol. 22, no. 3, pp. 21–29, 1989.
- [9] H. Terasaki, T. Hasegawa, H. Takahachi, and T. Arakawa, "Automatic grasping and regrasping by space characterization for pick-and-place operations," *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS'91*, Osaka, Japan, Nov. 1991.
- [10] J. R. Tresilian, "Attention in action or obstruction of movement? a kinematic analysis of avoidance behavior in prehension," *Experimental Brain Res.*, vol. 120, no. 3, pp. 352–368, 1998.
- [11] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 278–288, 1991.
- [12] O. Ben-Shahar and E. Rivlin, "Practical pushing planning for rearrangement tasks," *IEEE Trans. Robot. Automat.*, vol. 14, no. 4, pp. 549–565, aug 1998.
- [13] R. Alami, J. Laumond, and T. Simèon, "To manipulation planning algorithms," *Algorithmic Foundations of Robotics*, pp. 109–125, 1995.