Object-Specific Grasp Maps for Use in Planning Manipulation Actions

Franziska Zacharias, Christoph Borst, and Gerd Hirzinger

Abstract. Humans use learned knowledge to solve reaching tasks and to manipulate objects and tools. We believe that representations of manipulation characteristics of an object and of the reaching capabilities of a robotic arm can speed up low-level planners, like grasp planners. They also enable sophisticated scene analysis and reasoning for high-level planners, like task planners.

We present object-specific grasp maps to encapsulate an object's manipulation characteristics. A grasp planner is shown to use the grasps maps and a representation of the reachable workspace. The exploitation of the provided knowledge focuses the planning on regions of the object that are promising to yield high quality grasps. Speed ups of factor 2-12 are reported.

1 Introduction

In the course of life a human learns to use his arms but also to grasp and use tools and objects. Thus the human can rely on knowledge about the world and about himself to decide which regions are reachable for him and how to approach objects. It could be imagined that this knowledge is saved in the form of a model that can be referenced in action planning. Our goal is to use models of a robotic arm's capabilities and an object's manipulation properties to guide the task planning process at a high level of abstraction. The models can furthermore be used to parameterize path and grasp planners and enable them to solve a problem more quickly (Fig. 1). We introduced a representation of a robotic arm's reachable workspace [13] that characterizes regions w.r.t. their reachability from various directions. Using this representation it can easily be determined whether an object in the scene is reachable and from what

Franziska Zacharias · Christoph Borst · Gerd Hirzinger
German Aerospace Center (DLR), Institute of Robotics and Mechatronics,
Münchner Straße 20, 82234 Weßling, Germany,
e-mail: {franziska.zacharias,christoph.borst,
gerd.hirzinger}@dlr.de

directions. While an object may be reachable for the robotic arm, the question whether stable grasps can be realized from the corresponding approach direction is still open. An object like a coffee pot for instance is not graspable equally well in all regions of its geometry model. Grasps which are not suited to the hand kinematics (Fig. 2(a)) should be discarded early. Due to the object's orientation a prefered grasp (Fig. 2(b)) may not alway be reachable and other grasps (Fig. 2(c)) have to be used. Or all grasps may be unreachable (Fig.2(d)). A model of an object's manipulation properties should reflect these issues. A high-level planner could use this model to determine if good approach directions are accessible. In case of cluttered scenes, it could be analyzed whether a rearrangement of the scene or a repositioning of the mobile manipulator is required. In path planning research, workspace knowledge [14] and knowledge about the capabilities of a robotic arm [5] is used to speed up planning. However to the authors' knowledge neither current high-level planners like a manipulation planner nor low-level planner like a grasp planner use knowledge about an object's manipulation characteristics or representations of the reachable workspace. Instead state of the art grasp planners [9], [10], [7] integrate additional algorithmic frameworks to ensure particular characteristics. For instance a robotic arm's inverse kinematics is directly integrated into the grasp planner to ensure that grasps are reachable by the robot arm. In the long run this leads to huge monolithic planning systems. The representation of good grasps has also received attention. In manipulation planning often fixed manipulation points on the objects [12], predefined grasps [1] or simple rules for grasp generation are used [2] instead of versatile grasp planners. These methods discard a lot of manipulation possibilities especially for multifingered hands. Pelossof et al. [11] use an SVM approach to learn from a set of good grasps how to generate grasps for similar objects. Gienger

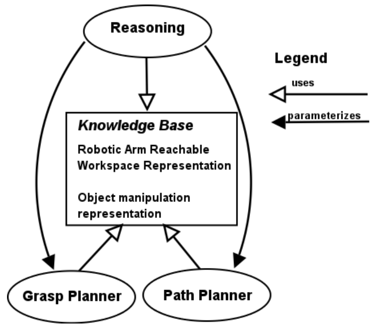


Fig. 1 Knowledge about the world and about the robot is used to guide and parameterize planning processes.

(a) (b) (c) (d)

Fig. 2 Various ways to reach for and grasp a coffee pot.

et al. [6] represent a continuous set of power grasps by a Rapidly-exploring Random Tree. The object-specific grasp information is used in motion optimization to speed up motion planning. However, these representations where not designed to be efficiently exploited by planners like grasp planners, or task planners.

In this paper we take a first step towards this goal and introduce a simple representation of an object's manipulation capabilities. We show how a grasp planner can use this representation and a previously published representation of a robot arm's reachable workspace to avoid wasting planning efforts on regions of the object to be grasped that do not participate in reachable high quality grasps.

2 Grasp Maps for Use in Grasp Planning

In this section we demonstrate the benefits of including object information into grasp planning. We summarize the grasp planner introduced by Borst et al. [4]. We describe the derivation of simple grasp maps and adapt the grasp planner presented by Borst et al. [4] to use this object knowledge. We assume that a geometric triangulated model of an object is available. No preprocessing steps are necessary.

2.1 The Grasp Planning Algorithm

The main aspect for a planned grasp is to hold an object firmly and safely, also in the presence of disturbances acting on the object, e.g. during transport motions. Grasps that have this property are called force-closure grasps. Unbound forces in the contact points are assumed. In this paper a grasp is composed of k point contacts with friction, with one contact per finger (precision grasp). We focus on multi-fingered force-closure precision grasps. Borst et al. [4] presented an intuitive measure for evaluating the quality of a grasp. They furthermore proposed an efficient generate and test algorithm to plan good but not optimal grasps for robotic hands independent of a robotic arm in very short time.

The grasp quality measure is based on the two concepts of the grasp wrench space GWS and the object wrench space OWS. A wrench is composed of the forces and torques acting in a contact point. The GWS of a given grasp describes disturbances that a grasp can counterbalance. The OWS of a given object describes which disturbances may occur. The largest factor by which the OWS can be scaled to just fit in the GWS was identified as an intuitive physically motivated measures of a grasp's quality. Based on this measure a generate and test grasp planner was introduced. It includes the following steps:

1. Contact points are generated by randomly sampling a face of the triangulated object model (uniform distribution).
2. Heuristic fast prefiltering steps roughly check force closure and gripper constraints. The tests are conservative since no solutions are discarded.
3. Grasp quality and robustness is evaluated
4. Gripper configuration is determined
5. Collision checks of the hand with the object to grasp are performed

All combined the grasp planner can produce good grasps for arbitrary 3D objects online. Depending on the object complexity to generate a grasp the planner needs between 20 ms-180 ms on an Intel Pentium D 3 GHz with 2 GB main memory.

2.2 Grasp Maps

A grasp planner based on random sampling strategies produces high and low quality grasps. We believe that some regions of an object's geometry contribute to high quality grasps more than others. When thinking of objects like the coffee pot (Fig. 2), a martini glass or a cup this immediately becomes obvious. In the grasp map, we intend to represent how specific regions of an object participate in manipulation. The geometric object model will be attributed with region specific aspects. Did a specific region participate in force closure grasps? What quality did these grasps have? The grasp planner presented in the last section places contact points on the object without any knowledge of the kinematics of the robotic hand. Therefore a lot of grasps are sampled that are kinematically infeasible or cause collisions between the object and the hand. The object-specific grasp maps will provide this knowledge to the grasp planner at the earliest stage. Using these grasp maps a grasp planner can automatically bias the generation process to concentrate the search for grasps in regions promising to yield higher quality force-closure grasps faster.

We illustrate the grasp map generation using an object constructed of primitive shapes (Fig. 3). A cube, a cylinder and a cone are concatenated. Using the grasp planner described in the preceding section we generate a number of n grasps per geometric object model. For each face we register its participation in valid force closure grasps. Furthermore we iteratively compute for each face the minimum, the maximum and the mean of the grasp quality measure for the grasps it participated in. Using these attributes important manipulation characteristics can be visualized. Fig. 3 shows two views of the grasp map for our example object. In Fig. 3 (left) the color encodes the absolute frequency of a face participating in a force closure grasp. Regions that were often part of grasps are light gray. Those that were seldom used are dark gray. In Fig. 3 (right) the color encodes the mean grasp quality value from low quality (dark gray) to high quality (light gray). This object shows a clear regional preference w.r.t. where high quality grasp can be found i.e. they are most often found on the cube part. Combining the information in both pictures it can be seen that high quality grasps are also found on the cylinder but seldom.

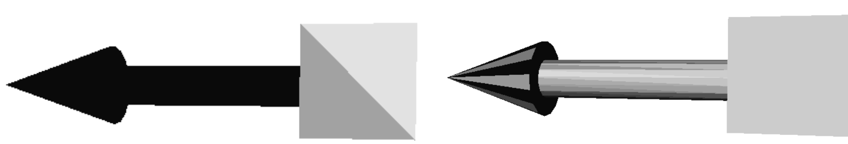


Fig. 3 (Left) Absolute frequency of a face participating in a grasp. (Right) Mean quality per face.

2.3 Feeding the Grasp Planner with Object Information

As a first step the object specific manipulation knowledge represented by the grasp maps will be used in a modified version of the grasp planner presented in section 2.1. We will show that by integrating the grasp map into the grasp planning algorithm higher quality grasp can be generated significantly faster, especially for complex objects. Borst et all. showed that for 4 or 5 finger grasps on a glass only 8% to 15 % of the sampled grasps were force closure grasps. Research in motion planning has already discovered workspace biasing methods and resulting non-uniform sampling distributions to lead to significant speedup in planning [3], [14]. Therefore we will modify the contact generation step of the grasp planner presented in section 2.1. Here contact points were placed on randomly sampled faces using a uniform distribution. Using the object's grasp map, we bias this sampling process to prefer regions of an object that promise to yield good force-closure grasps. We use the biasing method presented in [14]. Specifically we bias the distribution to favor faces that have a high max grasp quality measure and were often part of valid grasps. These two characteristics need to be combined. Otherwise given a high max grasp quality measure a face that e.g. participated only in one grasp would be treated the same as a face that was often part of valid grasps.

2.4 Discussion of Results

We evaluate the impact of using object-specific grasp maps in the grasp generation process using the objects in Fig. 4. For each object we generated the grasp map offline based on a set of 20000 grasps. The grasps were generated with the original grasp planner described in section 2.1. It should be noted that the coffee mug and the martini glass have interior regions which are also considered by the grasp planner. To measure the improvement we computed the time to generate one grasp and the mean grasp quality measure (Tab. 1). Since the grasp planner incorporates a random-sampling process the values were averaged using 1000 grasps. As expected using grasp maps for objects of rotational symmetry achieves practically no improvement. The reason for this is that valid force closure grasps are uniformly distributed across the object surface. However, a clear improvement w.r.t. time and quality was achieved for the martini glass and especially for the coffee mug. Here our assumption that not every region of an object participates in valid grasps is intuitive and valid. The object-specific grasp map prevents the grasp planner from exploring unpromising regions. For the coffee mug these regions are

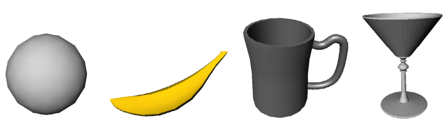


Fig. 4 Objects for testing the enhanced grasp planner.

Table 1 Time (in ms) and grasp quality measurements for all objects.

	original planner		grasp map planner	
	time	mean quality	time	mean quality
sphere	22	0.59	26	0.575
banana	17	0.09	24	0.098
martini glass	75	0.085	35	0.1
coffee mug	119	0.127	21	0.164

the inner regions and the handle. For the martini glass the regions on the stem are found to be rather insignificant for force closure grasps.

3 Reachable Grasps

Up to now it could be argued that a set of valid force closure grasps could equally well be generated for each object and saved in a database instead of biasing the grasp planner using grasp maps and computing grasps online. However as soon as the hand is attached to a robotic arm the advantages of using the grasp map strategy become evident. Reachable grasps cannot be saved for each object since they depend on the object's position w.r.t. the robot. Assuming a database of grasps for an object is available two strategies could be pursued. In the first strategy each grasp in the database would need to be checked for reachability using the robotic arm's inverse kinematics. They would furthermore have to be checked for collisions with the environment. Especially for cluttered scenes and complex objects that are difficult to grasp this can be an expensive process. A second strategy could be similar to that presented by Berenson et al. [1]. Here a database of grasps for each object is assumed to exist. The grasps are then ranked by combining several heuristic values that take the robot kinematics and the environment into account. However each grasp has to be tried out and evaluated till a valid reachable and collision-free configuration of arm and hand is found. In this section we show that reachable grasps can be generated without introducing heuristics. We generate grasps online by using the grasp map approach. By using a representation of a robot arm's reachable workspace we decouple the grasp planner from the inverse kinematics of the arm.

3.1 Ensuring Modularity by Using the Capability Map

Humans use internal models to solve reaching tasks [8], to infer which regions are reachable from what directions. Possessing a similar abstraction of a robot arm's capabilities in its workspace is important for task planning. We previously introduced a model that can be used for this purpose [13] and will summarize its main points.

3.1.1 The Capability Map of the Robot

The theoretically possible robot arm workspace is enveloped by a cube and subdivided into equally-sized smaller cubes. Into each cube a sphere is inscribed and on this sphere n points are uniformly distributed (Fig. 6 (a), (c)). Frames are generated for each point and serve as the target tool center point (TCP) for the inverse kinematics of the robot arm (Fig. 6 (b)). The point on the sphere determines the z-axis of the TCP frame. The orientation of the x-axis and y-axis is subsampled. The result of the inverse kinematics is registered in the data structure that is visualized by the sphere. Reachable points on the sphere are visualized as black lines. The spheres visualize the reachability for a region and are therefore called reachability spheres. Fig. 6 (c) shows a sphere where only some points are reachable. The reachability sphere map is the aggregation of all spheres. It can be used to visualize and inspect the reachability across the workspace and to approximate the shape of the robot arm workspace. It was shown that the reachability sphere map of the DLR robot arm is regularly structured [13]. The lines on the reachability spheres form cone-like structures in outer workspace regions. These cones then open out and become cylinders in the center of the workspace. Towards the inner workspace regions the cylinders change into cones again. Due to this observation, cones and cylinders were fitted to the reachability spheres to capture the data. The quality of the fit was measured by

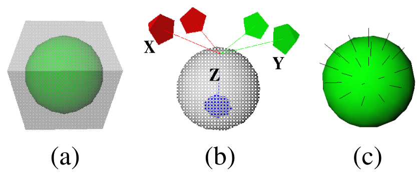


Fig. 6 Sphere inscribed into the cube (a), exemplary frames for a point on the sphere (b), valid inverse kinematics solutions on a sphere (c).

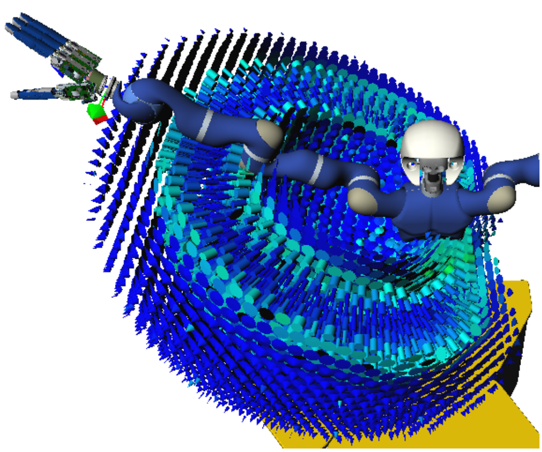


Fig. 5 The capability map for the right arm of the humanoid robot Justin.

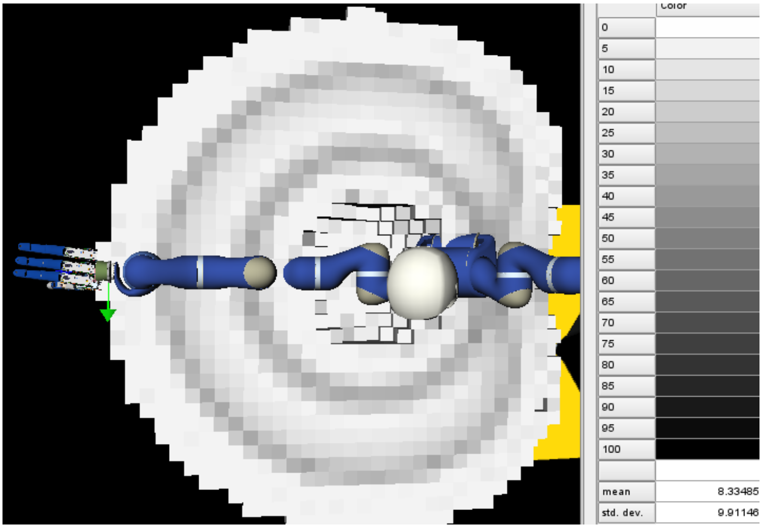
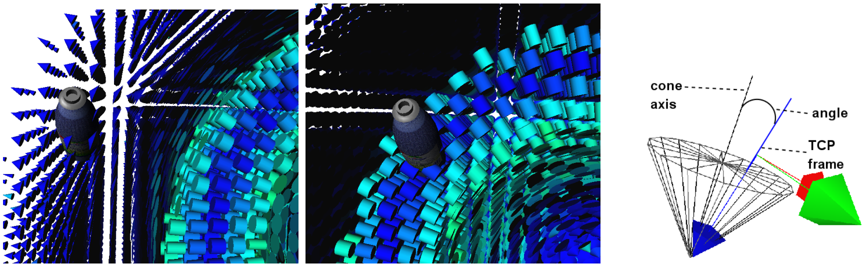


Fig. 7 Gradient of the SFE in grayscale for the capability map for the right arm of the humanoid robot Justin.

the relative error of an approximation, called the shape fit error (SFE). The capability map of the robot arm is a representation of the reachability sphere map. It is derived by replacing every sphere by the best fitting shape. Fig. 5 shows the capability map for the right arm of the DLR robot Justin. The location where the hand was attached to the robot arm serves as TCP for the map computation. It is marked by a coordinate system. The SFE is represented in grayscale in Fig. 7. Here cubes replace the cones and cylinders to allow the isolated inspection of the gradient of the SFE. Light gray indicates low error (0) and dark gray indicates high error (100). The mean SFE of 8.33 shows that the capability map is a valid representation of the reachability sphere map.

3.1.2 Filtering Grasps Using the Capability Map

Using the capability map, it can easily be determined whether an object of the scene is reachable and from which directions. Fig. 8 (a) shows a bottle placed near the outer border of the workspace. In this region cones are used to represent the directions from which the areas can be reached. Fig. 8 (b) shows the same bottle placed at the center of the robot arm workspace. Here cylinders are predominantly used to capture the reachability data. The possibilities to approach and manipulate the object are more numerous as is directly evident from the capability map. We provide the grasp planner with the capability map as a model of the robot arm's reachable workspace. The grasp planner then uses this model to predict the reachability of a grasp. A grasp is composed of 4 finger contact points on the object, 4 surface normals at these contact points and a frame T_H defining the position and orientation



(a) (b) (c)

Fig. 8 (a,b) A bottle in different areas of the workspace, (c) test for reachability.

of the hand in a reference coordinate system. This frame T_H is tested against the capability map to determine the reachability of the grasp. The frame is mapped to a region of the capability map. To test its reachability the inner product between the z-axis of the TCP frame and the axis of the respective shape is computed. This is then compared to the opening angle of the shape (Fig. 8 (c)). If the test fails the grasp is discarded as unreachable. The filtering step using the capability map is added to the *Fast Prefiltering* step of the grasp planner. To evaluate the achieved improvement we measure the time to generate one reachable grasp with the original system and our improved grasp planner that uses grasp maps and the capability map of the robot arm. The original grasp planner does not determine a grasp's reachability by a robotic arm. Therefore we first create a grasp using the original grasp planner and check it for reachability using the arm's inverse kinematics. If it is not reachable a new grasp is generated. The execution times averaged over 1000 grasps are listed in table 2. A clear improvement is visible. The objects were placed in the center of the reachable workspace. The speed up ranged from factor 2 to 6 depending on object geometry. For different positions and orientations of the objects in the workspace speed ups up to factor 12 were observed. The effect was more pronounced in the bordering workspace regions since here fewer grasps are actually reachable. Orientation dependency can be observed for objects like the coffee cup. The orientation of the handle in the workspace influences the speed up to range from factor 3 to 12 given the object position is kept.

Table 2 Time (in ms) and grasp quality measurements for all objects.

	original		improved	
	time	mean quality	time	mean quality
sphere	120	0.58	95	0.576
banana	91	0.067	94	0.074
martini glass	306	0.075	158	0.084
coffee mug	395	0.130	68	0.157

4 Outlook and Future Work

In this paper we introduced the idea of object-specific grasp maps. We presented the capability map and the grasp maps as resources for a grasp planner. The randomized grasp planer was able to exploit the provided knowledge to bias the exploration towards the most promising regions. In the best cases a good grasp was found 12 times faster than before. In most cases also the mean grasp quality was significantly improved w.r.t. to the original planner. Thus the grasps produced with the grasp map based system have a higher quality than grasps produced with the original planner. In the long run we expect the capability map and the object-specific grasp maps to be a valuable resource for a number of planning subsystems (Fig. 1). Future versions of the grasp maps will contain not only information w.r.t. good finger positions but also identify good approach directions. In a cluttered environment a reasoning component could use the representations to determine whether an object is reachable and graspable at all. Using this information, a scene could be judged to be too difficult if a lot of approach directions are obstructed for the object manipulation. A rearrangement planner [12] could be triggered to simplify the scene setup and make it possible to achieve the task goal.

The quality of the grasps produced using the grasp maps is dependent on the features chosen, e.g. max quality measure, and their corresponding weights. To obtain an even better performance these weighting factors could be learned using the method described in [14]. If a robot often handles a specific object or if it does not know an object before hand and is able to build a geometric model online it is also imaginable to learn the grasp maps iteratively online and thereby improve the grasping process. This is a clear advantage compared to relying on a database of grasps.

References

1. Berenson, D., Diankov, R., Nishiwaki, K., Kagami, S., Kuffner, J.: Grasp planning in complex scenes. In: IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids 2007) (2007)
2. Berenson, D., Kuffner, J., Choset, H.: An optimization approach to planning for mobile manipulation. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (2008)
3. Boor, V., Oversmars, M., van der Stappen, A.: The gaussian sampling strategy for probabilistic roadmap planners. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1018–1023 (1999)
4. Borst, C., Fischer, M., Hirzinger, G.: Efficient and precise grasp planning for real world objects. Springer Tracts in Advanced Robotics 18, 91–111 (2005)
5. Diankov, R., Ratliff, N., Ferguson, D., Srinivasa, S., Kuffner, J.: Bispace planning: Concurrent multi-space exploration. In: Proc. Int. Conf. on Robotics: Science and Systems (2008)
6. Gienger, M., Toussaint, M., Goerick, C.: Task maps in humanoid robot manipulation. In: Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS), pp. 2758–2764 (2008)
7. Harada, K., Kaneko, K., Kanehiro, F.: Fast grasp planning for hand/arm systems based on convex model. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (2008)

8. Kawato, M.: Internal models for motor control and trajectory planning. Curr. Opin. Neurobiol. 9(6), 718–727 (1999)
9. Lopez-Damian, E., Sidobre, D., Alami, R.: A grasp planner based on inertial properties. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 754–759 (2005)
10. Miller, A.T., Knoop, S., Christensen, H.I., Allen, P.K.: Automatic grasp planning using shape primitives. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (2003)
11. Pelossof, R., Miller, A., Allen, P., Jebara, T.: An svm learning approach to robotic grasping. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (2004)
12. Stilman, M., Schamburek, J.U., Kuffner, J., Asfour, T.: Manipulation planning among movable obstacles. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (2007)
13. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: Representing robot capabilities. In: Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS), pp. 3229–3236 (2007)
14. Zucker, M., Kuffner, J., Bagnell, J.A.: Adaptive workspace biasing for sampling-based plaanners. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 3757–3762 (2008)