Accepted for 5th European Conference on Mobile Robots (ECMR), Örebro, Sweden, September 2011.

Real-Time 3D Perception and Efficient Grasp Planning for Everyday Manipulation Tasks

Joerg Stueckler, Ricarda Steffens, Dirk Holz, and Sven Behnke
Autonomous Intelligent Systems Group, University of Bonn, 53113 Bonn, Germany

***Abstract*— In this paper, we describe efficient methods for solving everyday mobile manipulation tasks that require object pick-up. In order to achieve real-time performance in complex environments, we focus our approach on fast yet robust solutions. For 3D perception of objects on planar surfaces, we develop scene segmentation methods that process Microsoft Kinect depth images in real-time at high frame rates. We efficiently plan feasible, collision-free grasps on the segmented objects directly from the perceived point clouds to achieve fast execution times. We evaluate our approaches quantitatively in lab experiments and also report on the successful integration of our methods in public demonstrations at RoboCup German Open 2011 and RoboCup 2011 in Istanbul, Turkey.**

***Index Terms*— Scene Segmentation, Grasp Planning, Mobile Manipulation**

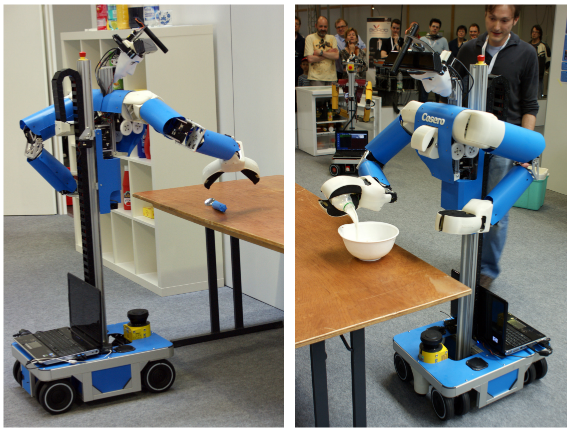

Fig. 1. Top: *Cosero* grasps a spoon and pours milk into a bowl of cereals. A 5 min video showing the complete demonstration at RoboCup GermanOpen 2011 is available at http://nimbro.net.

I. Introduction

Mobile manipulation tasks in domestic environments require a vast set of perception and action capabilities. The robot not only requires localization, mapping, path planning, and obstacle avoidance abilities to safely navigate through the environment. It also needs to integrate object detection, recognition, and manipulation. A typical requirement for a service robot is not just to achieve the task, but to perform it in reasonable time. While much research has been invested into the general solution of complex perception and motion planning problems, only few work has been focused on methods that solve the tasks efficiently in order to allow for continuous task execution without interruptions.

In this paper, we present fast methods to flexibly grasp objects from planar surfaces. To achieve fast performance, we combine real-time object perception with efficient grasp planning and motion control. For real-time perception, we combine rapid normal estimation using integral images with efficient segmentation techniques. We segment the scene into the support plane of interest and the objects thereon. Our perception algorithm processes depth images of a Microsoft Kinect in real-time at a frame rate of approx. 16 Hz. From the raw object point clouds our grasp planning method derives feasible, collision-free grasps within about 100 milliseconds. We consider grasps on objects from either the side or from above. The planned grasps are then executed using parametrized motion primitives. We integrate our approaches into a system that we publicly evaluate at RoboCup competitions. We further conduct experiments in our lab to demonstrate the robustness and efficiency of our approaches.

This paper is organized as follows: after a brief system overview in Sec. III, we detail our approaches to real-time 3D perception and efficient grasp planning in Sec. IV and Sec. V, respectively. In Sec. VI, we evaluate our approaches and report on successful public demonstrations at RoboCup GermanOpen 2011 and at RoboCup 2011 in Istanbul, Turkey.

II. Related Work

Many research groups currently develop systems for mobile manipulation in everyday environments. A very prominent example is the Personal Robot 2 (PR2) developed by Willow Garage [5]. It is equipped with two 7 DoF compliant arms and a parallel gripper with touch sensor matrices on the gripper tips. Similar to our approach, they derive feasible, collision-free grasps from the raw object point cloud [3]. They select the best-ranked grasp and plan a collision-free motion for the arm taking into account obstacles that are perceived by the robot's 3D sensors. While the authors demonstrate that the approach can robustly grasp a variety of objects in a wide range of configurations, the execution speed of the system for perception and grasping is still far slower than human performance.

Further systems that perform object manipulation in cluttered environments have been reported by Srinivasa et al. [8, 7]. In [8], the authors present a robotic busboy system in which a mobile tray delivers mugs to a statically mounted manipulator. The manipulator grasps the mugs and loads them into a dishwasher rack. A real-time vision system that is designed for the mugs estimates the pose of the mugs on the tray. Since the objects are known, valid grasps on the mug are precomputed. The grasp planner then selects online a best feasible grasp from several criteria like reachability and collision avoidance. The authors report on a total duration of 51 sec in average for executing a grasp and releasing the mug in the dishrack. With the robot HERB [7], the vision system has been extended to more general object recognition and motion planning. While object recognition is aborted after 1 sec, the planning of motions is reported to take several seconds. Our approach is not restricted to recognizable objects.

Jain and Kemp develop EL-E [4], a mobile manipulator that shall assist the impaired. EL-E consists of a Katana manipulator on a vertical linear actuator mounted on a Erratic differential drive. While we extract object information in real-time from a depth image sensor, they segment measurements of a 3D laser using connected components labelling to find object clusters above table height. Similar to our approach, they perform top grasps along the object's principal axis. However, side grasps are not considered. If an object is too high or too wide to fit into the gripper, they also consider overhead grasps on top-most points on the object. To ensure that the grasping motion is not in collision, a cuboid volume from the manipulator base to the object is checked for obstacles.

Morales et al. [11] propose a system that selects feasible, collision-free grasps on objects from a database. They determine the set of feasible grasps on the object from its CAD model in an offline phase. After the object has been recognized and localized with a stereo vision system, a grasp simulation framework (GraspIt! [6]) is used to select a collision-free grasp among the potential grasps on the object. The authors report 5 ms computation time for the recognition of objects in a database of 5 objects. The time for planning of collision-free, feasible grasps in GraspIt is reported to range from seconds to several minutes in [6].

III. System Overview

A. Design of Cognitive Service Robot Cosero

Domestic environments are designed for the specific capabilities of the human body. It is therefore natural to endow the robot with an anthropomorphic upper body scheme for similar manipulation abilities. Furthermore, the actions of the robot become predictable and interpretable, when they are performed human-like. In such environments, robots also have to interact closely with humans. By its lightweight design, Cosero is inherently less dangerous than a heavy-weight industrial-grade robot. Finally, the robot should also possess natural sensing capabilities, e.g., vision and audio, since humans design their environments salient and distinguishable in such perception channels. We focused the design of Cosero on such typical requirements for household settings.

We equipped Cosero with an omnidirectional drive to maneuver in the narrow passages found in household environments. Its two anthropomorphic arms resemble average human body proportions and reaching capabilities. A yaw joint in the torso enlarges the workspace of the arms. In order to compensate for the missing torso pitch joint and legs, a linear actuator in the trunk can move the upper body vertically by approx. 0.9 m. This allows the robot to manipulate on similar heights like humans.

Cosero has been constructed from light-weight aluminum parts. All joints in the robot are driven by Robotis Dynamixel actuators. These design choices allow for a light-weight and inexpensive construction, compared to other domestic service robots. While each arm has a maximum payload of 1.5 kg and the drive has a maximum speed of $0.6\,m/sec$, the low weight (in total ca. 32 kg) requires only moderate actuator power. Compared to its predecessor Dynamaid [9], we increased payload and precision of the robot by stronger actuation.

Cosero perceives its environment with a variety of complementary sensors. The robot senses the environment in 3D with a Microsoft Kinect RGB-D camera in its head that is attached to the torso with a pan-tilt unit in the neck. To improve the robustness of manipulation, the robot can measure the distance to obstacles directly from the grippers. We attached infrared distance sensors to each gripper that point downward and forward in the finger tips. Another sensor is placed in the palm and measures distance to objects within the gripper.

B. Mobile Manipulation in Everyday Environments

We develop Cosero to perform a variety of mobile manipulation tasks in everyday environments. For mobile manipulation, we combine safe navigation of the robot through the environment with motion control methods for the upper body.

1) Motion Control: We developed omnidirectional driving for Cosero's eight-wheeled mobile base [9]. The linear and angular velocity of the drive can be set independently and can be changed continuously. We determine the steering direction and the individual wheel velocities of the four differential drives, which are located at the corners of the rectangular base, from an analytical solution to the drive's inverse kinematics.

For the anthropomorphic arms, we implemented differential inverse kinematics with redundancy resolution [9]. We also developed compliance control for the arms [10]. For our method we exploit that the servo actuators are back-drivable and that the torque which the servo applies for position-control can be limited. Compliance can be set for each direction in task- or joint-space separately. For example, the end-effector can be kept loose in both lateral directions while it keeps the other directions at their targets. With these methods Cosero can perform a variety of parameterizable motions like grasping, placing objects, and pouring out containers.

2) Mobile Manipulation: In typical everyday mobile manipulation scenarios, the workspace of a statically mounted manipulator is too small. One possible solution to achieve a larger workspace is to construct robots with a restricted manipulator workspace but to extend it with a mobile base. Since the robot is not statically mounted to the environment, it has to estimate its pose in reference to static parts of the environment like walls, dynamic objects, and persons. We propose a coarse-to-fine strategy to align the robot to the objects involved in mobile manipulation. For example, when the robot grasps an object from a table, it first approaches the table roughly within the reference frame of the static map. Then, it adjusts in height and distance to the table. Finally, it aligns itself to bring the object into the workspace of its arms.

Cosero grasps objects on horizontal surfaces like tables and shelves in a height range from ca. 0.3 m to 1 m [9]. It

(a) Example table scene (b) RGB point cloud (c) Detected objects

Fig. 2. (a) Example table top setting. (b) Raw point cloud from the Kinect with RGB information. (c) Each detected object is marked with a random color.

also carries the object, and hands it to human users. We also developed solutions to pour-out containers, to place objects on horizontal surfaces, to dispose objects in containers, to grasp objects from the floor, and to receive objects from users.

When handing an object over, the arms are compliant in upward direction so that the human can pull the object, the arm complies, and the object is released. For receiving an object from a person, we localize the handed object with the depth camera and drive towards it. As soon as the object is reachable with the arms, the robot grasps it.

IV. Real-time 3D Perception

A typical task for a domestic service robot is to fetch and deliver objects. This involves detecting objects in the robot's workspace and recognizing them. In household environments, objects are usually located on planar surfaces such as tables. In [9] we proposed to use the laser scanner in the lower torso to detect such objects. This approach, however, is not able to perceive valuable 3D information like the objects' height or principal axes. We therefore developed real-time 3D scene segmentation with the RGB-D camera. In order to identify objects, we extract texture and color information.

A. Real-Time 3D Scene Segmentation

Our approach to object detection processes images of a depth camera such as the Microsoft Kinect at frame rates of approx. 16 Hz. This enables our system to extract information about the objects in a scene with a very low latency for further decision-making and planning stages.

We base our approach on fast planar segmentation of the scene. We achieve the high computational speed of our approach by combining rapid normal estimation with efficient segmentation techniques. The basic idea of the normal estimation method is to determine local surface normals from the cross product of two tangents to the surface. For each pixel in the depth image, the tangents are estimated from local pixel neighbors. In the simplest case, both tangents could be calculated from just the horizontal and vertical neighbors, respectively. However, this approach would be highly prone to measurement noise. The tangent estimates should therefore be averaged in an image neighborhood. By using integral images, such averaging operations can be processed rapidly in constant time independent of the neighborhood size. The overall runtime complexity of this approach is linear in the number of points for which normals are computed.

One way to find all planes in a scene would be to extract planes from the local surface normals in a two-stage process [2]. First, one could find clusters in normal orientation which then are separated by clustering in plane distance to the origin. Since we assume here that the robot as well as the objects in the environment are typically standing on horizontal surfaces, we instead focus our method on local surface normals close to the vertical direction. On all points with such normal orientation and within a 3D region of interest, we apply efficient RANSAC [1] to determine the horizontal plane. Then, we find the points above the plane and extract object clusters which projections lie within the convex hull of the support plane. For clustering we assume that there is a small space between the objects. The size of this space has to be chosen carefully, since due to occlusions, parts of an object may be disconnected. Fig. 2 shows a typical segmentation result for a table-top scene. A multi-object tracker is constantly updated with the detected objects.

V. Efficient Grasp Planning

Objects in everyday manipulation scenarios are highly variable in shape and appearance. Furthermore, the configuration of objects and obstacles in a scene is strongly unstructured. It is therefore challenging to develop a grasp planning method that can cope with any encountered situation. Our approach is specifically suited for rigid objects which shape is symmetric along the principal axes of the object. We also assume that the center of gravity roughly coincides with the center of the object. While many objects meet these assumptions, our approach can also yield robust grasps for objects that violate the constraints.

We developed flexible grasping motions to grasp objects from the side or from above. When the robot encounters a new situation, it plans and executes a feasible collision-free grasp on the object of interest. The robot perceives the scene with its depth camera. It interprets the raw point representation of the objects on the grasp surface which is provided by our real-time 3D perception method (s. Sec. IV-A).

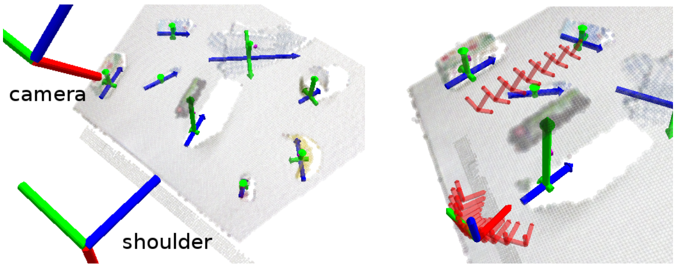

Fig. 3. Left: We extract object pose and shape properties from the object points. The arrows mark the bounding box of the objects by the principal axes. Right: We rank feasible, collision-free grasps (red, size prop. to scores) and select the most appropriate one (larger RGB-coded coordinate frame).

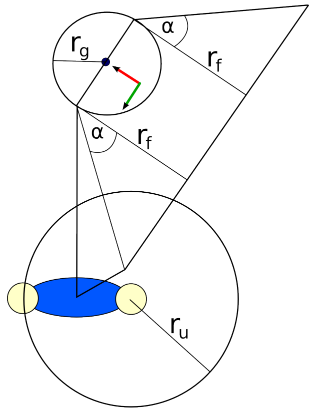

Fig. 4. For each sampled grasp (final position: black dot, pre-grasp at frame origin, x-direction: red arrow, y-direction: green arrow) we check for collisions that may occur during the execution of the grasp motion primitive. All points on obstacles are projected into the horizontal plane. We require the region around the shoulder (right yellow circle) within upperarm length distance r_u to contain no obstacles. We further require that the gripper and the forearm can move towards the object by checking a cone with opening angle α and forearm length r_f behind the grasping pose. We extend the cone towards the robot's center position to cover the area swept during the reaching motion. At the final grasp position (black circle) the gripper is not in collision, when there is no obstacle within a distance of r_g.

A. *Grasp Motion Primitives*

We distinguish two kinds of grasps for which we apply parametrizable motion primitives. Side-grasps are designed to approach the object along its vertical axis by keeping the parallel grippers aligned horizontally. To grasp objects from the top, we pitch the end-effector by $45°$ downwards to grasp objects with the finger tips.

Both kinds of grasps are flexible in the orientation around the vertical upward direction. However, we limit the yaw orientation to a range between $0°$ and $90°$ (for the right arm) due to kinematic constraints of the robot arm and torso. Orientations beyond this range are grasped in the closest limit angle. Alternatively, the robot can simply choose its left arm to grasp within the reachable range.

The motion primitives approach the pre-grasp poses on a direct line with open gripper. We establish the yaw orientation at the pre-grasp pose by smooth interpolation along the reaching trajectory. Once the pre-grasp pose is reached, the side-grasp motion primitive simply approaches the object and closes the gripper. For the top-grasp motion, we do not establish the pitch orientation of the pre-grasp pose until the pre-grasp position has been reached. We assume that the pre-grasp positions are placed at a fixed distance (0.1 m in our case) behind the grasp position along the grasp direction. We use the IR distance sensors in the gripper to determine premature contact with the object or the support plane. In such a case, the approach of the object is stopped. After the object has been grasped, the end-effector moves back to its initial pose.

B. *Planning of Collision-Free Grasps*

The grasp planner selects a feasible collision-free grasp for the object of interest. It samples grasp candidates, removes infeasible and colliding grasps, and ranks the remaining grasps to find the most promising one.

The planner outputs a pre-grasp pose to parametrize the grasping motion. A grasp pose directly corresponds to the pose of the end-effector which we define as follows: We place the grasp at the center of the gripper. The x-axis and y-axis of the grasp pose align with the direction from wrist to finger tips and the direction from the right to the left finger, respectively.

1) Sampling of Grasp Candidates: We sample grasp candidates depending on pose and shape properties of the object. In order to determine these properties we project the raw points on the object into the horizontal plane and measure the principal axes of the point distribution. In addition, we determine height, center, and bounding box (aligned with the principal axes) of the object.

Once the shape and pose of the object are known, we determine feasible grasps on the object. For the side-grasps, we sample pre-grasp poses on an ellipse in the horizontal plane in equally sized angular intervals. The center and axes of the ellipse directly correspond to the properties of the object's bounding box. The diameters of the ellipse add the distance towards the grasp point to the diameters of the bounding box (0.1 m in our implementation). We grasp the objects as low as possible above the surface at a specific height. This makes the grasping more robust for measurement and control inaccuracies. Otherwise, the object could easily topple over, when the robot touches the object while moving in grasping direction. We set the grasp height to half the height of the gripper plus a safety padding of 0.03 m.

We sample the top grasps equidistant along both principal axes through the center of the bounding box. For kinematic constraints of our anthropomorphic arms, we constrain the pitch of the grasp to $45°$ in downward direction. We place the pre-grasp pose 0.1 m above the object's height, but at least 0.1 m above the support plane.

2) Filtering for Feasible and Collision-Free Grasps: Since the sampling stage does not consider any feasibility constraints or collisions, we filter the grasp candidates in a post-processing step. We take the following criteria into account:

- *Grasp width.* We reject grasps, when the object's width orthogonal to the grasp direction does not fit into the gripper.
- *Object height.* Side-grasps are likely to fail when the object is too small.
- *Reachability.* We do not consider grasps that are outside

processing stage	mean (std) in msec
normal estimation	7.2 (2.4)
scene segmentation	11.9 (1.4)
object clustering	41.6 (1.5)
grasp planning	98.1 (9.1)

TABLE I

COMPUTATION TIME OF INDIVIDUAL PROCESSING STAGES.

of the arm's workspace.

- *Collisions.* We check for collisions during the reaching and grasping motion.

Fig. 3 shows an example for grasps that satisfy our criteria.

One possible solution for collision checking would be to search for collisions of all robot limbs during the complete trajectory of the grasping motion. However, we propose to use simple geometric constraints to find all possible collisions (s. Fig. 4). While our method is more conservative, we can find collisions with only little computational effort.

We first project all points on obstacles into the horizontal plane. In order to avoid collisions of the upperarm, we search for collisions within a circle around the shoulder with a radius equal to the upperarm length. We further require that the gripper and the forearm can move towards the object by checking a cone with opening angle and forearm length behind the grasping pose. We extend the cone towards the robot's center position to cover the area swept during the reaching motion. Finally we search for collisions within a small circle at the final grasp position. The radius of this circle is set to the maximum diameter of the open gripper.

3) Ranking of Grasps: We rank the feasible and collision-free grasps for several criteria such as

- *Distance to object center.* We favor grasps with a smaller distance to the object center.
- *Grasp width.* We reward grasp widths closer to a preferred width (0.08 m).
- *Grasp orientation.* Preference is given to grasps with a smaller angle between the line towards the shoulder and the grasping direction.
- *Distance from robot.* We support grasps with a smaller distance to the shoulder.

Fig. 3 illustrates this process with example rankings.

From the ranked grasps, we find the best top- and side-grasps and select the most appropriate one. This decision depends on the relation of the object height to the largest extent of the object in the horizontal plane. We integrate a small bias towards the faster side grasps.

VI. EXPERIMENTS

A. Quantitative Results

1) Run-Time Efficiency: In Table I, we summarize average run-times of several stages of our perception and grasp planning pipeline in the scene shown in Fig. 2. For a depth image resolution of 160×120, our table-top segmentation approach achieves an average frame rate of approx. 16 Hz. The experiments have been carried out on an HP Pavilion dv6 notebook with an Intel Core i7 Q720 processor. Using

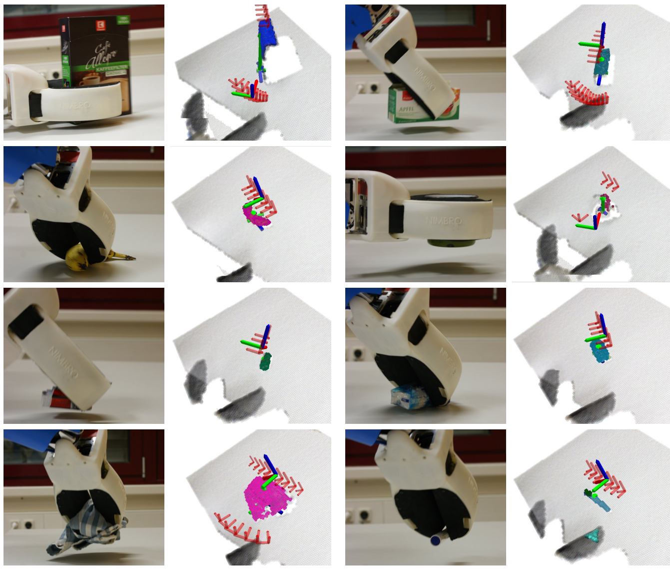

Fig. 5. Example grasps, segmentations, and grasp planning results for each of the 8 household objects used in the experiments.

the integral image approach, normals can be estimated rapidly for the 19200 image pixels within only 7.2 msec in average. The segmentation of the scene into vertical points, applying RANSAC to find the support plane, and determining the points above the support plane requires 11.9 msec (avg.). The clustering of the points into objects takes 41.6 msec (avg.). The computation time in this step depends on the number of objects in the scene. Our approach to grasp planning requires computation time in the same magnitude as the segmentation, i.e., 98.1 msec (avg.). The timings demonstrate that our approaches are very performant and yield results in short computation times.

We also measured the time for the complete object pick-up process. The robot has already approached the table. It perceives the objects on the table and plans a grasp on the closest object in front. It executes the grasp and moves the gripper back to its starting pose. The overall process takes approx. 15 sec for a side-grasp and 25 sec for a top-grasp.

2) Robustness: We evaluate the robustness of our perception and grasp planning pipeline in a series of experiments. We chose 8 typical household objects and executed 10 grasps with the right arm that cover the range of feasible object orientations for this arm. Fig. 5 shows an example grasp for each object. Table II summarizes the results of the experiment. The robot could grasp the most objects very reliably. For the tissues, it sometimes chooses a top-grasp along the shorter side of the object. In one situation it missed the object with this grasp. Our approach also estimates the tea box to be high enough to be grasped from the side in some configurations. Despite the fact that the clothes strongly violate our assumptions that objects are rigid and are shaped symmetric along principal axes, our method succeeds robustly for this object.

B. Public Demonstration

While one can assess the quality of individual system components in the laboratory, it is difficult to compare robot

object	side-grasp	top-grasp
filter box	10 / 10	0 / 0
tea box	1 / 1	9 / 9
banana	0 / 0	10 / 10
cup	10 / 10	0 / 0
chewing gums	0 / 0	10 / 10
tissues	0 / 0	9 / 10
cloth	3 / 3	7 / 7
pen	0 / 0	10 / 10

TABLE II

SUCCESS RATES (SUCCESS / TRIALS) WHEN GRASPING OBJECTS 10 TIMES IN RANDOM ORIENTATIONS.

systems with others. In recent years, competitions such as the DARPA Grand and Urban Challenges and RoboCup, play an important role in assessing the performance of robot systems.

The international RoboCup competitions include the @Home league for domestic service robots. In this competition, the robots have to perform tasks defined by the rules of the competition, in a given environment at a predetermined time. In addition, there are open challenges and the final demonstration, where the teams can highlight the capabilities of their robots in self-defined tasks. The simultaneous presence of multiple teams allows for a direct comparison of the systems by measuring objective performance criteria, and by subjective judgment of the scientific and technical merit by a jury.

With Cosero, we won the RoboCup GermanOpen 2011 competition. In the finals, Cosero and Dynamaid prepared breakfast within the 10 min demonstration slot. Dynamaid fetched orange juice out of the refrigerator, which it opened and closed successfully. It delivered the bottle on the breakfast table. In the meantime, Cosero grasped a bottle of milk, opened the bottle, and poured the milk into a cereal bowl. Cosero disposed the empty bottle into the trash bin. It then moved to another table and successfully grasped a spoon with a top-grasp. A jury member placed the spoon in an arbitrary orientation. Cosero put the spoon next to the cereal bowl and finally waited for an instruction to leave the room. Another jury member pointed towards one of two exit doors using a pointing gesture. Cosero successfully recognized the pointing gesture and left the room through the correct door. The jury awarded us the highest score for the finals.

We also won the @Home competitions at RoboCup 2011 in Istanbul, Turkey. Early in the competition in the open challenge, Cosero demonstrated to prepare cereals to a jury of team leaders of other teams. In the demo challenge, Cosero cleaned up the appartement by picking up laundry from the ground and putting it into the correct laundry basket. A human user could before show in which baskets to put colored and white laundry using gestures. Afterwards, Cosero picked up 3 objects from a table using the perception and grasping pipeline proposed in this paper. In the first attempt to pick up a carrot, it had to choose a grasp perpendicular to the carrot's principal axis and failed to keep grip on the object. However, in the second attempt, it picked up the carrot successfully along its principal axis. Finally, it grasped a tea-box with a top-grasp. The objects have been placed randomly. We could convince the jury with this demonstration and achieved the highest score. [1]

VII. CONCLUSION

In this paper, we proposed highly efficient means to perceive objects on planar surfaces and to plan feasible, collision-free grasps on the object of interest. We integrate our methods into a mobile manipulation system, that robustly executes object pick-up in reasonable time without longer processing interruptions, i.e. interruptions in the milliseconds to seconds.

For object perception, we segment depth images of a Microsoft Kinect camera in real-time at a frame rate of up to 6 Hz. We demonstrated that our perception and planning modules yield their results in a very short time. In the integrated system this allows for short and steady execution of the task. Our experiments demonstrate that our method is fast yet robust.

In future work, we plan to integrate feedback from touch sensors into the grasp execution. By including top-grasps at high points on the objects we could further extend the range of graspable objects (bowl-like shapes, for instance). We could also improve our grasping pipeline through knowledge on how to grasp specific objects.

ACKNOWLEDGMENTS

This research has been partially funded by the FP7 ICT-2007.2.2 project ECHORD (grant agreement 231143) experiment ActReMa.

REFERENCES

[1] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[2] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke. Real-time plane segmentation using RGB-D cameras. In *Proc. of the 15th RoboCup International Symposium*, 2011.

[3] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones. Contact-reactive grasping of objects with partial shape information. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[4] A. Jain and C. C. Kemp. EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 28:45–64, January 2010.

[5] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. Bradski, K. Konolige, B. P. Gerkey, and E. Berger. Autonomous door opening and plugging in with a personal robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[6] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[7] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. C. Romea, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and J. M. Vandeweghe. Herb: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, January 2010.

[8] S. Srinivasa, D. Ferguson, J. M. Vandeweghe, R. Diankov, D. Berenson, C. Helfrich, and H. Strasdat. The robotic busboy: Steps towards developing a mobile robotic home assistant. In *Proc. of the Int. Conf. on Intelligent Autonomous Systems (IAS)*, 2008.

[9] J. Stückler and S. Behnke. Integrating indoor mobility, object manipulation, and intuitive interaction for domestic service tasks. In *Proc. of the 9th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2009.

[10] J. Stückler and S. Behnke. Compliant task-space control with back-drivable servo actuators. In *Proc. of the 15th RoboCup International Symposium*, 2011.

[11] A. Morales T., Asfour, P. Azad, S. Knoop, and R. Dillmann. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[1] Videos showing these demonstrations can be found at http://nimbro.net.