2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Mobile Bin Picking with an Anthropomorphic Service Robot

Matthias Nieuwenhuisen[1], David Droeschel[1], Dirk Holz[1], Jörg Stückler[1],
Alexander Berner[2], Jun Li[2], Reinhard Klein[2], and Sven Behnke[1]

Abstract— **Grasping individual objects from an unordered pile in a box has been investigated in static scenarios so far. In this paper, we demonstrate bin picking with an anthropomorphic mobile robot. To this end, we extend global navigation techniques by precise local alignment with a transport box. Objects are detected in range images using a shape primitive-based approach. Our approach learns object models from single scans and employs active perception to cope with severe occlusions. Grasps and arm motions are planned in an efficient local multiresolution height map. All components are integrated and evaluated in a bin picking and part delivery task.**

I. Introduction

Removing individual objects from an unordered pile of parts in a carrier or box—*bin picking*—is one of the classical problems of robotics research. It has been investigated by numerous research groups over three decades [1], [2], [3].

Typical bin picking solutions consist of a 3D sensor mounted above the box, a compute unit to detect the objects, estimate their pose and plan grasping motions, and an industrial robot arm that is equipped with a gripper.

So far, bin picking robots are stationary. In order to extend the workspace of the robot and to make bin picking available for environments that are designed for humans, we implement bin picking using an autonomous anthropomorphic mobile robot. Mobile bin picking is made feasible by the advances in sensing, computing, and actuation technologies, but still poses considerable challenges to object perception and motion planning. Our robot Cosero has been designed for mobile manipulation and intuitive human-robot interaction tasks, which were tested successfully in RoboCup@Home competitions [4].

Here, our scenario is motivated by industrial applications. We consider the task of grasping objects of known geometry from an unordered pile of objects in a transport box. The grasped object is to be transported to a processing station where it is placed. Due to the operability of the robot in environments designed for humans, this scenario is easily transferable to a household scenario, e.g., clearing a shopping box. Solving this mobile manipulation task requires the integration of techniques from mobile robotics, like localization and path planning, and manipulation, like object perception and grasp planning.

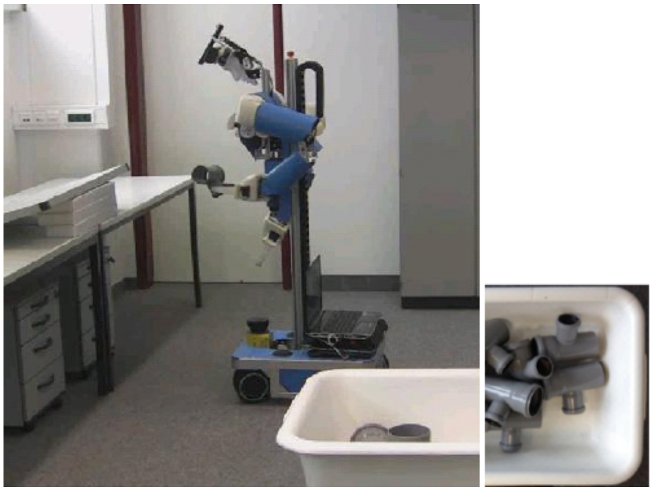

Fig. 1. Mobile bin picking scenario. Objects are grasped from a transport box (lower right) and placed on a processing station (upper left).

For mobility, we extend global navigation techniques by precise local alignment with the transport box and the processing station. For manipulation, we extend our previous work on shape primitive-based object detection and grasping [5] by the learning of object models, active perception, and object removal planning. We integrated all components to perform the complete task and evaluated the performance of the integrated system.

The remainder of this paper is structured as follows. In the next section, we review related work on stationary bin picking and mobile manipulation. Sec. III gives a high-level overview of our mobile bin picking system. Object detection and pose estimation are covered in Sec. IV and new objects are learned in Sec. V. Sec. VI and VII cover robot navigation and grasp planning, respectively. Detected objects are fed back to view planning to cope with severe occlusions (Sec. VIII). We report results on the experimental evaluation of our approach in Sec. IX.

II. Related Work

Despite of its long history, static bin picking is still an active research area. One recent implementation of Papazov et al. [6] utilizes a Microsoft Kinect sensor mounted above a table to acquire depth images of the scene. Object models are matched to the measured point cloud by means of a normal-based RANSAC [7] procedure. Papazov et al. consider tabletop scenes where multiple objects are arranged nearby, including the stacking of some objects. They select the object to be grasped based on the center of mass height (high objects are preferred). Each object is associated with a list of predetermined grasps, which are selected according to the

This research has been partially funded by the FP7 ICT-2007.2.2 project ECHORD experiment ActReMa.

[1] Autonomous Intelligent Systems, Institute for Computer Science VI,
[2] Computer Graphics, Institute for Computer Science II,
University of Bonn, Germany

978-1-4673-5643-5/13/$31.00 ©2013 IEEE

orientation of the gripper (grasps from above are preferred) and checked for collisions. The grasping is performed by a compliant lightweight robot arm with parallel gripper. Bley et al. [8] propose another approach of grasp selection by fitting learned generic object models to point cloud data. In contrast to our approach they manipulate separated objects.

Choi et al. [9] proposed a Hough voting-based approach that extends point-pair features [10], [11], which are based on oriented surface points, by boundary points with directions and boundary line segments. Choi et al. use a structured-light 3D sensor mounted on an industrial arm to acquire point clouds of small objects in a transport box, estimate their pose and grasp them with high success rate. Another extension of Drost et al. [11] has been proposed by Kim and Medioni [12]. They consider visibility in between the paired surface elements to sort out false matches.

While the above methods for object detection work best with objects that contain distinctive geometric features, other approaches for object detection rely on the decomposition of point clouds into geometric primitives. The method proposed by Schnabel et al. [13], for example, is based on RANSAC and efficiently detects planes, spheres, cylinders, cones, and tori in the presence of outliers and noise. Another work in this direction is Li et al. [14], which build a graph of primitive relations and constraints. Assuming symmetry and consistent alignments of shapes in man-made objects, the orientations and positions of detected shapes are iteratively refined.

The above approaches require dense depth measurements. In contrast, Liu et al. [3] developed a multi-flash camera to estimate depth edges, which is mounted on an industrial robot arm. Detected edges are matched with object templates by means of directional Chamfer matching and objects are grasped with a three-pin gripper that is inserted into a hole at a success rate of 94 %.

Some research groups used mobile robots to grasp objects from piles. Klingbeil et al. [15], for example, utilized a Willow Garage PR2 robot to grasp unknown objects from a pile on a table and read their bar-codes to demonstrate a cashier checkout application. Because the dense packing of objects in a pile poses considerable challenges for perception and grasping, Chang et al. [16] proposed pushing strategies for the interactive singulation of objects. Gupta and Sukhatme [17] estimate how cluttered an area is and employ motion primitives to separate LEGO bricks on a pile.

Manipulation in restricted spaces like boxes and shelves leads to difficult high-dimensional motion planning problems. To this end, Cohen et al. [18] proposed a search-based motion planning algorithm that combines a set of adaptive motion primitives with motions generated by two analytical solvers.

All the above approaches are demonstrated with a static robot. In contrast, Chitta et al. [19] proposed an approach to mobile pick-and-place tasks, which integrates 3D perception of the scene with grasp and motion planning. The approach has been used for applications like tabletop object manipulation, fetching of beverages, and the transport of objects. In these applications, objects stand well-separated on horizontal surfaces or are ordered in feeders.

Other systems for which mobile pick-and-place has been realized include HERB [20], developed at the Intel Research Lab Pittsburgh. HERB navigates around a kitchen, searches for mugs and brings them back to the kitchen sink. Rollin' Justin [21], developed at DLR Oberpfaffenhofen, Germany, grasped coffee pads and inserted them into the coffee machine, which involved opening and closing the pad drawer. The Armar robots [22], developed at KIT, Germany, demonstrated tasks in a kitchen scenario that require integrated grasp and motion planning. In the health care domain, Jain and Kemp [23] present EL-E, a mobile manipulator that assists motor impaired patients by performing pick and place operations to retrieve objects. In Beetz et al. [24] a PR2 and the robot Rosie, developed at TU Munich, cooperatively prepare pancakes, which involves mobile manipulation and the use of a tool.

In most of these mobile manipulation demonstrations, the handled objects are well-separated. To the best of our knowledge, a mobile bin picking application has not been realized so far.

III. System Overview

We consider a scenario where unordered parts need to be grasped from a transport box, as shown in Fig. 1. The objects are transported to a processing station and placed there.

For the experiments, we use our cognitive service robot Cosero [25], shown in Fig. 1, which navigates on an eight-wheeled omnidirectional base and has an anthropomorphic upper body with two 7-DoF arms that end in grippers with two Festo FinGripper fingers. Due to the Fin Ray effect, the finger tips passively bend inwards, creating a closure around a grasped object. To extend the workspace, the upper body of the robot can be twisted around the vertical axis and lifted to different heights. With only 32 kg, Cosero has a low weight, compared to other service robots. Nevertheless, its arms can lift a payload of max. 1.5 kg each. The robot senses its environment in 3D with a Microsoft Kinect RGB-D camera in the pan-tilt head. For obstacle avoidance and tracking in farther ranges and larger field-of-views, it is equipped with multiple laser-range scanners, of which one in the chest can be pitched and one in the hip can be rolled. Cosero's main computer is a quadcore notebook with an Intel i7-Q720 processor.

The mobile bin picking task is divided into the cognition phase where the robot explores the transport box and recognizes the top-most objects, the pick-up phase where the robot grasps a top-most object out of the transport box, and the place phase where the robot places the object on the processing station.

The autonomous robot behavior is generated in a modular control architecture, using inter process communication infrastructure of the Robot Operating System (ROS) [26]. We implemented the mobile bin picking task as a finite-state machine. It monitors the state of task fulfillment and triggers individual behaviors in the appropriate order. The task starts with the robot navigating to the transport box. When the robot is in front of the transport box, it switches to a local navigation mode that accurately aligns it to the box. The next step is the acquisition of a 3D point cloud of the entire transport box, which is then processed by the object recognition module. The detected objects are fed to the grasp planner, which selects an appropriate grasp and plans trajectories for approaching the object and for removing it from the box. After the planned grasping motion is executed, Cosero navigates to the processing station using the environment map and local alignment with the processing station. Finally, our robot releases the object into the processing station. This process continues until the transport box is empty.

IV. Object Recognition

Our method for 3D object recognition is based on subgraph matching: First, we convert both the searched object and the scanned scene into an annotated graph. Nodes of the graph are instantiated for simple shapes (spheres, cylinders, planes) detected by an extended algorithm that is based on Schnabel et al. [13]. Edges connect those simple shapes which are neighbored in space and also store the relative pose of the primitives.

We localize objects in a scene by identifying parts of our search graph in the graph of the scene. Using the established graph correspondences, we calculate a rigid transformation to the assumed position and orientation of the object.

Finally, we verify this hypothesis with an object model (cf. Sec. V) positioned by this transformation.

A. Scan Registration

For perception of the transport box, we use a Microsoft Kinect camera that is mounted on the robot's pan-tilt unit. The sensor's limited field-of-view results in an incomplete scan of the box. Therefore, we employ scan registration of three overlapping point clouds from different views (middle, left, right) and align them to each other. We use the Iterative Closest Point (ICP) algorithm [27] to align the scans. To reduce the size of the model point cloud, only non-overlapping parts of consecutive scans are added to the model point cloud (see Fig. 3). After acquiring and aligning the scans, the model is stored in our planning representation (cf. Sec. VII-C).

B. Recognition

In the following, we give an overview of our recognition pipeline (referring to Fig. 2):

○ *Graph matching:* In a preprocessing step, we rapidly calculate surface normals and remove outliers in the data.

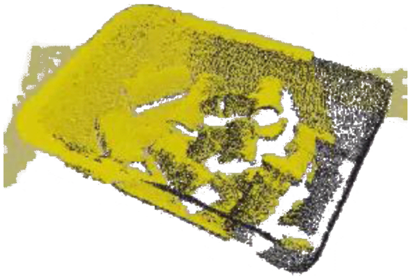

Fig. 3. An acquired 3D scan (yellow) is aligned to the model point cloud (black) using the ICP algorithm. New parts are added to the model.

Next, we detect shape primitives and establish the graph of spatially close primitives. We efficiently create the annotated shape graph of the scanned point cloud and apply our subgraph matching approach following ideas of [13]. First, a random start edge of the query graph is searched in the scene graph comparing the shape attributes. We go through the list of all similar edges and, for each edge, we try to expand the match to adjacent edges in the query graph and the scene graph simultaneously. This gives us a number of partially matched graphs. For the best ones, we compute a transformation matrix that encodes the estimated position and orientation of the object in the scene. Then, we use this transformation to verify our graph hypothesis.

○ *Computing position and orientation:* We compute a relative 6-DoF transformation towards a common reference shape position and orientation in our object model. For symmetric objects, we can take any transformation around the self-symmetry axis and compute a valid transformation. It is possible to detect all self-symmetries in the model generation phase (cf., [28]). We apply the transformation to the object model, check for sufficient overlap with the scanned points, and improve the accuracy by employing the ICP method of Mitra et al. [27] to register the object model to the points.

○ *Fallback solution for difficult scenarios:* In our experiments, we encountered situations, where even the active recognition was unable to identify a last remaining part in the transport box. An example for this is shown in Fig. 4: In this experiment, we emptied the box up to this one pipe connector, which is standing on its smaller pipe. For our graph-based recognition method, we need at least two primitives, to robustly detect an object and its pose, but here we are only able to scan just one cylinder. We developed a fall-back solution for such situations: If no further object can be detected, but the volume inside the box is not empty, we allow the recognition with just one primitive—in this case

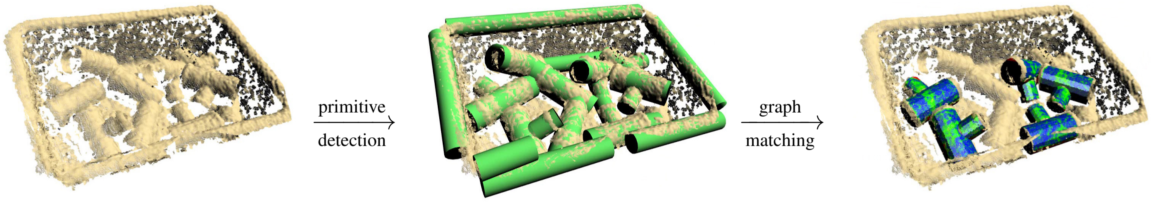

Fig. 2. Object recognition pipeline: Input scan - fast preprocessing, primitive detection - shape graph matching, transformation estimation and verification.

just one cylinder. Cylinders belonging to the box are ignored.

V. Example-Based Object Model Generation

In our previous work the object graphs were derived from exact CAD models. Though CAD models are typically available in industrial applications, the situation is different for the operation of a domestic service robot. To apply our model-based approach in a household scenario, object models have to be learned from example objects. To this end, the exemplars are placed into a designated learning board and a scan with the 3D sensor is taken. The learning board holds multiple instances of the object in different roughly known orientations, such that one scan is sufficient to obtain all information needed in later processing steps. All further processing is fully automated.

○ *Registration:* As a first step, we preprocess the scanner data, computing surface normals and removing outliers. We use the primitive detection to detect the largest plane and to register the scanned board to our internal model of the board area and expected poses of the parts. Then we remove all points of the board plane. All remaining points belong to the object instances. Next, we segment the remaining point cloud into the object parts and separate them into individual point clouds. The result is equivalent to the same number of scans taken from various viewpoints, with the difference that we know a rough transformation for all of the scans.

For the registration of the object point clouds we follow a coarse-to-fine approach. To improve the pose estimates of the object instances, we adapted the object detection and pose estimation method of Papazov and Burschka [29] to obtain a better initial transformation for the fine alignment. They compute point-pair features on a searched object and sample them in a searched scene, trying to find matches. Output is a pose estimate of the searched object in the scene. In our case, we use one of the segmented scans as searched object and the other one as target scene. This is a good setting for the method as we have a large overlap of the two objects—we always combine segments with close transformations and the target scene contains the segmented part only. In contrast to the original algorithm, we only allow transformations in the RANSAC step that are close to our estimated ones. For these reasons, we obtain very reliable pose results using this method. The fine alignment of the point clouds is now performed using ICP [27].

○ *Model Generation:* We search for shape primitives in the registered point cloud. Here, we set detection tolerance parameters very tight and get a clean set of shape primitives representing the object and a graph describing the relation of neighboring parts. The detected constellation of primitives is stored with the object and used for our object recognition method. The overall process of our registration and reconstruction pipeline is depicted in Fig. 5.

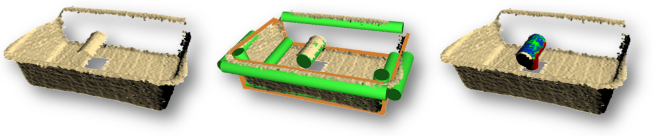

Fig. 4. Discovering the final remaining object: Scan of a pipe connector in the box, standing on the smaller pipe (left), The recognized primitives are not sufficient for the graph-based recognition (middle), Our fallback solution successfully identified the object in the box (right).

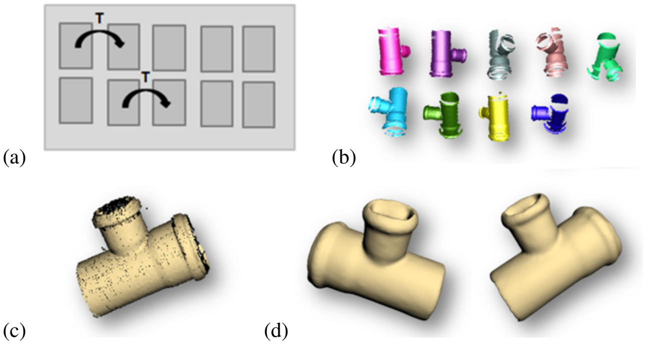

Fig. 5. Object reconstruction for model learning. Multiple instances of the object with roughly known poses are scanned simultaneously (a). The scan is segmented (b) and the resulting point clouds are registered using feature- and point-based techniques (c). Finally, the surface is reconstructed for visualization (d).

VI. Navigation

To cope with the challenges of mobile bin picking, we use a global-to-local strategy for approaching the transport box and fetching the work piece. Fig. 6 depicts the map for global localization and path planning and the local sensing used during the final box approach. We use a global navigation approach that utilizes a 2D map of the environment to roughly approach a pose in the map and a local navigation approach that accurately aligns the robot with the transport box and the processing station.

A. Global Navigation

For global navigation, we employ state-of-the-art methods for localization and mapping in 2D representations of the environment. Adaptive Monte Carlo Localization is used to estimate the robot's pose in a given occupancy grid map using a laser-range finder (see Fig. 6a). To plan a path from its estimated pose in the map to the target location, A* search [30] is applied to find short obstacle-free paths.

B. Local Navigation

In order to maximize the workspace of the robot and allow for active perception, an accurate alignment between the robot and the transport box is necessary. We use a 2D laser range finder that is mounted in the robot's trunk at a height of 80 cm to measure the distance and orientation to the transport box (see Fig. 6b).

To detect the transport box, we continuously extract line segments from the 2D laser scan by comparing the distance of neighboring points. We check the straightness of a line segment by principle component analysis and neglect line segments that exceed a given curvature. The closest remaining line segment that fits the dimensions of the transport box corresponds to the rim of the transport box. The box is approached by locally navigating to a predefined pose relative to the deduced box model.

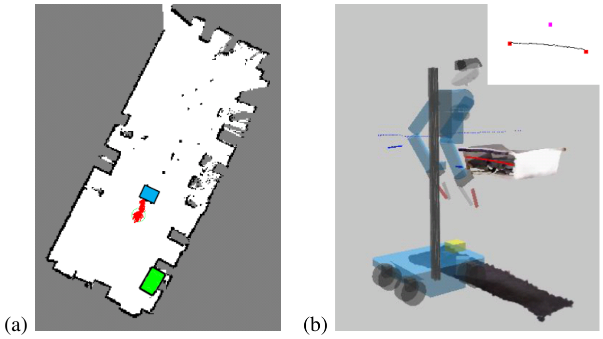

Fig. 6. Localization (pose estimates: red dots) and navigation to the box (blue rectangle) and processing station (green rectangle) is performed in a global frame using a known environment map (a). Local sensing and navigation is utilized to ensure a good alignment with the box/station. (b) depicts the robot's model, sensor input and result of the box detector during approach.

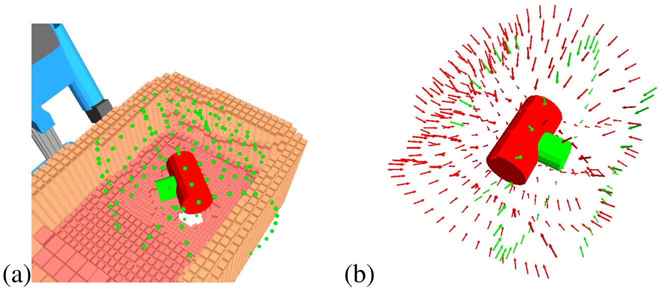

Fig. 7. (a) Selection of feasible grasps and arm motion planning is performed with a multiresolution height map. (b) For each shape primitive in an object compound, we sample grasps according to the parametric description of the shape primitives. For the grasps, we determine pre-grasp poses (visualized as arrows pointing in forward direction of the gripper; color codes correspondence to shape primitives). We discard grasps that are in collision within the object.

VII. Grasping of Shape Primitive Compounds

A. Grasp Planning

We plan grasps in an efficient multistage process that successively prunes infeasible grasps in tests with increasing complexity: In the first stages, we find collision-free grasp poses on the object, irrespective of the pose of the object and not considering its scene context (see Fig. 7b). These poses can be pre-calculated efficiently in an off-line planning phase. We sample grasp poses on the shape primitives. From these poses, we extract grasps that are collision-free from pre-grasp pose to grasp pose according to fast collision-check heuristics.

During on-line planning, we examine the remaining grasp poses in the actual poses of the objects to find those grasps for which a collision-free solution of the inverse kinematics in the current situation exists. We filter grasps before evaluation against our height map and finally search collision-free inverse kinematics solutions for the remaining ones. We allow collisions of the fingers with other parts in the transport box in the final stage of the grasp, i.e., in the direct vicinity of the object to grasp. The shape of the fingers allows for pushing them into narrow gaps between objects. If a valid solution is found, we employ motion planning to find a trajectory.

B. Motion Planning

Our grasp planning module finds feasible, collision-free grasps at the object. The grasps are ranked according to a score which incorporates efficiency and stability criteria. The final step in our grasp and motion planning pipeline is now to identify the best-ranked grasp that is reachable from the current posture of the robot arm. We solve this by successively planning reaching motions for the found grasps. We test the grasps in descending order of their score. For motion planning, we employ LBKPIECE [31].

We split the reaching motion into multiple segments. This allows for a quick evaluation if a valid reaching motion can be found by planning in the descending order of the probability that planning for a segment will fail.

C. Multiresolution Height Map

To speed up the process of evaluating collision-free grasp postures and planning trajectories, we employ a multiresolution height map that extends our prior work on multiresolution path planning [32].

Our height map is represented by multiple grids that have different resolutions. Each grid has $M \times M$ cells containing the maximum height value observed in the covered area (Fig. 7a). Recursively, grids with quarter the cell area of their parent are embedded into each other, until the minimal cell size is reached. With this approach, we can cover the same area as a uniform $N \times N$ grid of the minimal cell size with only $\log_2((N/M)+1)M^2$ cells.

Planning in the vicinity of the object needs a more exact environment representation as planning farther away from it. This is accomplished by centering the collision map at the object. This approach also leads to implicitly larger safety margins with increasing distance to the object.

D. Removal Planning

After the execution of the reaching motion, we check if the grasp was successful. If the object is within the gripper, a removal motion is planned with the object model attached to the end-effector using the detected object pose. We allow minor collisions of the object and the end-effector with the collision map in a cylindrical volume above the grasp pose. Finally, the work piece is deposited at the processing station. To reach it, global navigation and local alignment are used in the same way as for the box approach.

VIII. Active Recognition

The complete geometry of top-level parts in a box can in general not be acquired from a single scan due to occlusions. In particular, with increasing geometric part complexity large surface parts are likely to be occluded and cannot be acquired from a single view point. The range data obtained from a single scan is thus prone to being incomplete and provides only fragments of the actual part surfaces. Fragmentation leads to considerable uncertainty in the object recognition and thus threatens robustness of the object pose detection as described in Sec. IV. Furthermore, even if objects are recognized in the first scan, overlying objects can obstruct collision-free grasps if not yet recognized. To address the aforementioned problems, and to achieve robustness of object perception with respect to occlusions, we have developed an active object perception method for actively moving the scanning device to various view poses. This typically involves navigating locally with the robot's base.

A. Pre-Processing and Registration

To fuse acquired scans and aggregate information for view planning, we incrementally build a volumetric model of the transport box. For efficiency, we sort out all measurements not belonging to the transport box employing the estimated box pose from our 2D box detection (cf. Sec. VI-B). We deduce an oriented bounding box and project it into the acquired 3D range scan. Only measurements within the bounding box are considered for further processing. Moreover, the estimate of the box pose allows for differentiating between the box and its content, as well as predicting the hitherto unseen transport box volume. For fusing and aligning acquired scans, we incrementally build a model of the transport box and register the acquired scans. The aligned scan points are added to the model while avoiding to add duplicate points in the model.

For representing the volumetric model of the transport box, we use a multiresolution voxel grid map, based on [33]. It is organized as an octree with leaves that model multiple attributes of the underlying volume, e.g., the object detection's interest in that region or the volume's occupancy. The latter allows us to explicitly model the difference between seen free volume and previously unseen (unknown) volume, as well as to integrate the identified regions of interest (to guide the further acquisition of scans). For efficiency, our algorithm operates only in the oriented bounding box of the transport box model.

B. View Planning

For planning the next best view, we consider

- previously unseen or unknown volume (in order to obtain more information),
- previously seen volume or occupied volume (in order to obtain a sufficient overlap for the registration),
- and the recognition results fed back into the model of the transport box (for focusing on regions where no objects have been detected or where the detected objects have only little confidence).

We consider the box to be explored and stop the exploration if no more unknown or unseen volume exists within the transport box.

○ *Sample generation and travel cost:* We apply a sampling-based approach for determining the next best view. We first generate a set of sample poses and then estimate, for every sample pose, the involved traveling cost and the expected information gain. The pose with highest utility, i.e., with a high information gain and low traveling cost, is selected as the next best view. Since approaching a new view pose close to the box only involves local navigation with the robot's omnidirectional base, we approximate the involved traveling cost by the Euclidean distance between the robot's current pose and the base positions of the sampled view poses.

○ *Identifying regions of interest:* In Sec. IV, we described our approach for detecting basic geometric shape primitives and objects composed of shape primitives. The object detection component gives us detailed feedback on all regions in an input point cloud to guide the further acquisition of scans to regions having no or only little confidence in detected objects. When a point is detected to lie on a primitive that is not belonging to the object's shape primitive compound we are searching for, the region around the point is less interesting than regions where object detections are still possible.

○ *Information gain estimation:* Classic approaches to view planning consider previously unseen volume and previously seen surface. We have developed an approach to view planning that seamlessly integrates with classic approaches, but also considers how interesting a certain region actually is.

For the task of detecting objects in a transport box, we integrate the possible outcomes of shape primitive compound detection into the built model of the transport box, and use it to derive regions of interest (Fig. 8). View planning then focuses on sensing regions of interest in addition to considering previously unseen volume and previously seen surface. The interest score is stored in our volumetric model in addition to the occupancy. Regions that possibly contain an object but where no object has been detected yet are more interesting. The final information gain for a sampled view is the sum of interest values over all cells visible from the sensor pose. For efficiency, we first extract all unknown cells (previously unseen volume) from the transport box, as well as all occupied cells (previously seen surface). If regions of modeled free space are considered interesting, they are handled extra and in addition to the extracted cells. We remove cells that lie outside the sensor's view frustum. For the remaining cells, we conduct a reverse ray-casting from the cell to the view pose to determine the cell's visibility. To further speed up this step, we limit resolution and depth in the tree. Furthermore, we only consider those segments of the ray that are contained in the oriented bounding box of the transport box model. Overall, our approach allows for focusing the acquisition of new range scans in regions where we expect to find objects. For planning these views, we can compute 100 samples per second.

IX. Experiments

We tested the integrated system in our lab with our cognitive service robot Cosero.

A. Model Learning

We reconstructed two different objects, the pipe connector and a cross clamping piece. All reconstructions could be performed within 12 to 15 seconds. We compared the parameters of the detected primitives in the reconstruction to ground truth parameters and observed deviations of about 2%

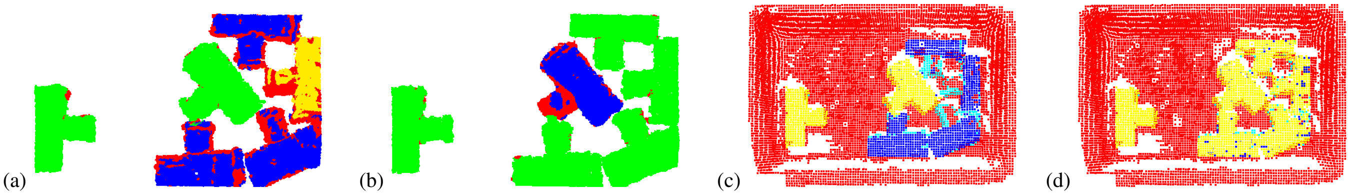

Fig. 8. Detecting objects and updating regions of interest, detected cylinders draw the robot's attention to closely inspect this region finally leading to the detection of all objects: (a) First scan with detected objects (green), detected cylinders (blue), primitives not being searched for (yellow), in the shown case a wrong primitive has been fitted due to yet missing information, and unassigned points (red). (b) After incorporation of a second scan more objects and cylinders are detected. (c) First model with two detected objects (yellow) and larger interesting regions on the right, color-coded from interesting (blue) to not interesting (red). (d) Updated model with six detected objects.

TABLE I

RESULTS FOR CROSS CLAMPING PIECE (CCP) AND PIPE CONNECTOR

Object Type	CCP	Pipe	Overall Average
Visible Objects	48	47	47.5
Pose Estimable	48	36	42
Ours	11 / 0	17 / 0	14 / 0
PPF	19 / 16.5	12.5 / 9.5	15.8 / 13

(true positives / false positives)

to 3%, using a precise 3D scanner. To judge the quality of our detected primitives for the object recognition, we compared the models learned from the scans with handcrafted models based on ground truth data. The observed recognition results were similar.

B. Object Recognition

We compared our method with the method from Papazov and Burschka [29] using point-pair features (PPF). For both object types, we created five piles of ten objects and scanned them. To annotate ground truth poses, we manually placed 3D models in the scene. For some visible objects, no manual pose could be estimated because of ambiguities (pose estimable in Tab. I). We averaged the results for the randomized PPF algorithm over ten runs. The comparative results are given in Tab. I. On average our method found slightly less objects. The advantage of our method is, that it did not produce any false-positives. Thus, it is better suited for our grasp planning approach.

C. Mobile Bin Picking

The transport box was filled with up to ten pipe connectors (Fig. 1). We split the complete task in single runs where the robot picks up one pipe connector and delivers it. In total, we have recorded 32 runs. In 28 runs, the robot could successfully grasp a pipe connector and deliver it to the processing station. In nine of these successful runs, the robot first failed to grasp an object, detected its failure, and performed another grasp. This was the case, for instance, when the object slipped out of the gripper after grasping. In four runs, the object was not successfully delivered to the processing station. In three out of the four failed runs the last object could not be detected. In one instance, the object slipped out of the gripper after lifting. This is caused by the fact that we have to allow collisions between the gripper and other objects during the grasp. Occasionally, these minor collisions can cause changes in the object's pose that can make the chosen grasp impossible or unstable. Fig. 9 shows some images from a successful run. Tab. II shows the mean and standard derivation of the measured phase durations for the 32 individual runs. Please note, that the timings for the grasp selection and motion planning within the cognition phase are averaged over the ten runs it took to clear one completely filled box. One can see that the longest phase is the cognition phase where objects are detected and the grasping motion is planned. This phase also includes the transmission of the sensor data to the object recognition module on a physically distinct computer. We omitted bin picking experiments with the cross clamping piece, as that industrial part is too heavy for our service robot.

TABLE II

TIME NEEDED FOR PHASES OF THE BIN PICKING DEMONSTRATION.

	Duration (in sec.)	
Phase	Mean	Std. dev.
Navigation (transport box)	20	8
Approaching (transport box)	16	11
Cognition phase	83	41
- Grasp selection	19.9	14.4
- Motion planning	3.8	2.6
Grasping	36	7
Navigation (processing station)	26	9
Approaching (processing station)	22	9
Putting the object on the processing station	18	2

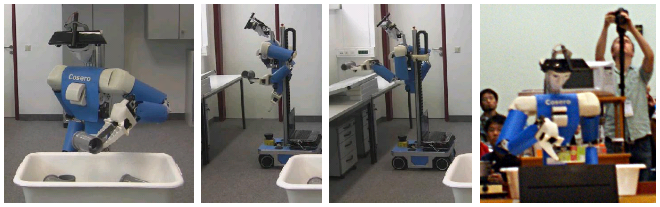

Fig. 9. Left: Cosero grasps a pipe connector out of the transport box, brings it to the processing station, and finally deposits it. Right: Public demonstration of mobile bin picking at RoboCup 2012.

D. Active Recognition

We conducted a set of experiments to demonstrate the feasibility of our approach. Fig. 8 shows typical examples where the active perception component leads to new object detections (a-b) or a complete model of the transport box and all objects contained therein (c-d). Object detections are successively integrated into the model. Regions where no object was detected, but which could contain objects are represented in the model and draw the robot's attention when planning the next view.

X. Conclusion

In this paper, we presented an integrated system for a mobile bin picking application. This requires a combination of navigation and manipulation skills, like global navigation, local precise alignment, 3D environment perception, object recognition and motion planning. We recognize objects using an efficient noise-resistant approach using RANSAC and sub-graph matching. In order to obtain the necessary shape composition graphs, we derive models automatically from 3D point cloud data or CAD models.

Grasping objects is realized as a multistage process from coarse, i.e., global navigation in the environment, to fine, i.e., planning a collision-free end-effector trajectory within a multiresolution collision map. Intermediate steps align our robot to the transport box and the processing station using local sensing and navigation and evaluate the graspability of objects using fast heuristics. To cope with occlusions in the unordered pile of objects in the transport box, we developed view planning techniques, involving active sensor positioning and navigation of the robot's base around the box.

We showed the applicability of our approaches in a mobile bin picking and part delivery task in our lab, where our service robot Cosero cleared a transport box with pipe connectors. A video summarizing our work is available on our website[1]. Among other skills, we demonstrated mobile bin picking in the @Home final of RoboCup 2012 in Mexico, where our robots convinced the high-profile jury and won the competition. In future work we aim at improving the object detection performance by extending our approach with contour primitives.

References

[1] K. Ikeuchi, B. K. Horn, S. Nagata, T. Callahan, and O. Feirigold, "Picking up an object from a pile of objects," in *Robotics Research: The First International Symposium*. MIT Press, 1984, pp. 139–162.

[2] K. Rahardja and A. Kosaka, "Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, 1996.

[3] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *Int. J. of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.

[4] J. Stückler, D. Holz, and S. Behnke, "RoboCup@Home: Demonstrating everyday manipulation skills in RoboCup@Home," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 34–42, 2012.

[5] M. Nieuwenhuisen, J. Stückler, A. Berner, R. Klein, and S. Behnke, "Shape-primitive based object recognition and grasping," in *Proc. 7th German Conference on Robotics*, 2012.

[6] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3D geometry matching for grasping of known objects in cluttered scenes," *Int. J. of Robotics Research*, vol. 31, no. 4, pp. 538–553, 2012.

[7] 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*, vol. 24, no. 6, pp. 381–395, 1981.

[8] F. Bley, V. Schmirgel, and K.-F. Kraiss, "Mobile manipulation based on generic object knowledge," in *Proc. IEEE Int. Symp. on Robot and Human Interactive Communication*, 2006.

[9] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3D sensor," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012.

[10] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification," in *Proc. Int. Conf. on 3-D Digital Imaging and Modeling*, 2003.

[11] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[12] E. Kim and G. Medioni, "3D object recognition in range images using visibility context," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, 2011.

[13] R. Schnabel, R. Wessel, R. Wahl, and R. Klein, "Shape recognition in 3D point-clouds," in *Proc. Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2008.

[14] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "Globfit: Consistently fitting primitives by discovering global relations," *ACM Trans. on Graphics*, vol. 30, pp. 52:1–52:12, 2011.

[15] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, "Grasping with application to an autonomous checkout robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011.

[16] L. Chang, J. R. Smith, and D. Fox, "Interactive singulation of objects from a pile," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012.

[17] M. Gupta and G. S. Sukhatme, "Using manipulation primitives for brick sorting in clutter," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012.

[18] B. Cohen, G. Subramanian, S. Chitta, and M. Likhachev, "Planning for manipulation with adaptive motion primitives," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011.

[19] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, "Perception, planning, and execution for mobile manipulation in unstructured environments," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 58–71, 2012.

[20] S. S. Srinivasa, D. Ferguson, C. J. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. V. de Weghe, "HERB: a home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.

[21] B. Bäuml, F. Schmidt, T. Wimböck, O. Birbach, A. Dietrich, M. Fuchs, W. Friedl, U. Frese, C. Borst, M. Grebenstein, O. Eiberger, and G. Hirzinger, "Catching flying balls and preparing coffee: Humanoid Rollin'Justin performs dynamic and sensitive tasks," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011.

[22] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simultaneous grasp and motion planning: Humanoid robot ARMAR-III," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 43–57, 2012.

[23] A. Jain and C. C. Kemp, "EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2010.

[24] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic roommates making pancakes," in *Proc Int. Conf. on Humanoid Robots*, 2011.

[25] J. Stückler, R. Steffens, D. Holz, and S. Behnke, "Real-time 3D perception and efficient grasp planning for everyday manipulation tasks," in *Proc. European Conf. on Mobile Robots*, 2011.

[26] S. Cousins, B. Gerkey, K. Conley, and Willow Garage, "Sharing software with ROS," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 12–14, 2010.

[27] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Symp. Geometry Processing*, 2004.

[28] A. Berner, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel, "A graph-based approach to symmetry detection," in *Proc. IEEE/EG Int. Symp. on Volume and Point-Based Graphics*, 2008.

[29] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *Proc. Asian Conf. on Computer Vision*, 2011.

[30] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[31] I. A. Sucan and L. E. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Algorithmic Foundation of Robotics VIII*. Springer, 2009, vol. 57, pp. 449–464.

[32] S. Behnke, "Local multiresolution path planning," *Robocup 2003: Robot Soccer World Cup VII*, pp. 332–343, 2004.

[33] K. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*.

[1] www.ais.uni-bonn.de/ActReMa