Fast Vision-based Grasp and Delivery Planning for unknown Objects

Johannes Baumgartl and Dominik Henrich
Chair for Applied Computer Science III (Robotics and Embedded Systems), University of Bayreuth, Germany

Abstract

Fast manipulation of unknown objects is a basic ability on the way to the robotic co-worker, because it is mostly impractical to have a model of each object in the environment. This paper addresses the problem of grasping and delivering unknown objects by an industrial robot from a table using one single camera image. The delivery planning for each object is done with respect to the packing problem, with the additional constraint that the objects must be regraspable after their delivery. The aim is to build up a system, which has for example the ability to tidy up a table. For this aim, we present two planning algorithms, one for the grasp planning and one of the delivery planning. The performance and robustness of the system is validated by test runs with several objects.

1 Introduction

For human beings it is easy to grasp unknown objects. For a robot, this task is one of the most challenging problems. But it is one important step to reach the goal of a robot as a helper or co-worker. Nobody can imagine the objects a robot has to deal with, if its task is to tidy up an apartment or an assembly shop. In such situations, a robot must have the ability to grasp unknown objects and to deliver this object into a storage device.

The aim of this work is to grasp and deliver unknown object with a small industrial robot. The challenge is to do it fast, i.e., that a human does not recognize planning times during the execution. Here we will concentrate on grasp and delivery planning. We are using a lightweight industrial robot, a parallel jaw gripper and a standard webcam attached to the robot (Figure 1). All planning steps use only one image of the scene. The task for the object placement is, that the objects are placed in a bounded, rectangular and flat delivery area, like a tray or an ark, so that they can be regrasped and unpacked again. The challenge is to place as many objects as possible in this area.

The outline of this paper is as follows. In Section 2 the related work is discussed. Our two planning algorithms are presented in Section 3. In the experiments section, the solution is tested, whether it satisfies the demands on the planning time, execution velocity and robustness. In Section 5 the work is discussed and finally summarized in Section 6.

2 State of the Art

Grasp planning can be divided into two kinds of grasps, form closure ones [7] and force closure grasps. We will concentrate on force closure grasps. Additionally the related work is grouped based on the two questions: What is the execution time of the planner?

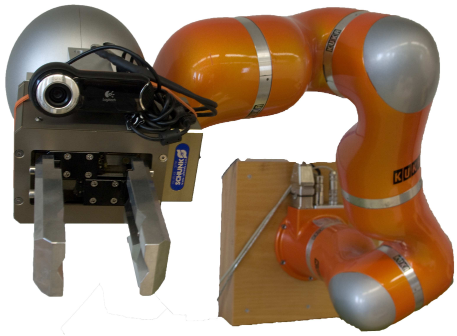


Figure 1: The used hardware, consisting of a 7-DoF robot, a parallel jaw gripper, and a colour camera

Takes the planner positioning errors and sensor uncertainties into account during the planning?

If the grasp planning is performed for known objects, there are many solutions including quality metrics; the paper [22] offers a good survey. These metrics cannot be applied to our planner, as they need proper information about the object's shape and other detailed information like the surface texture or the material of the object.

In [6, 20] uncertainties, based on the positioning of the grippers fingers, are taken into account. They define independent grasp regions on polygonal objects. To apply a force closure grasp it is sufficient to place one finger somewhere each region. They do not handle the problem that a finger is not only a point, but also a rigid body, like a parallel jaw gripper.

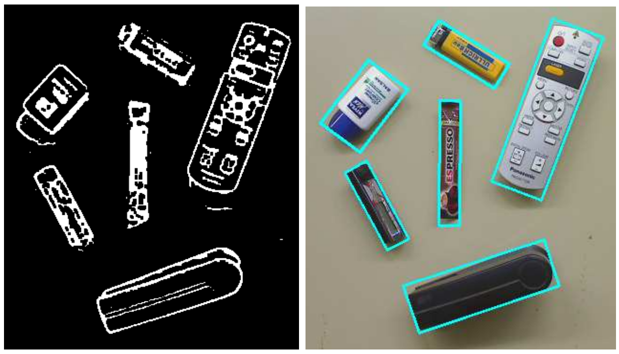
To handle novel objects there are many solutions, which try to learn how to grasp a class of objects, for example [4, 11, 21]. Jiang et al. [18] introduce a grasping system that plans on a single coloured depth image for a parallel jaw gripper. They use a two-step learning algorithm to compute the grasp configuration. The computation time of those algorithms is to long for our constraints.

Another possibility to deal with unknown objects is to approximate them by primitives like boxes or cylinders [14].

The real object is not taken into account in the grasp planning, so it is hard to guarantee a force closure and accurate grasp, as needed by the delivery planning.
Both [5] and [25] introduce a approach to grasp unknown objects based on visual information. The planning and modelling time takes more than a second. This does not meet our time constraints. However [5] uses the size of the surface patches to rate the best grasp. This minimizes the external torque.
Jain and Kemp [17] regard pick and place tasks from flat surfaces and they model the object as a point cloud. They use the footprint of the cloud to compute the maximum variance and try to grasp perpendicular and near the centroid. If that strategy fails, they search for the object's highest point and within its neighbourhood all points are selected witch have an nearly equal height. With these subset of points the grasp planning is repeated. The delivery task focuses on finding a good place in the neighbourhood where a human points to. Therefore it is evaluated if the object will be placed so far apart as the object can not fall off the table. The delivery task is different from that one we focus at, since we like to pack objects as close as possible to a given area.
Whelan and Batchelor [24] divide the domain of delivery planning into automated assembly, storage and transportation, material cutting and packing applications. We will concentrate on the packing of unknown objects into one single bounded area. There are solutions which handle non-rectangular objects like furniture, for example [8] and [23]. The runtime of these planners are too long for our needs. From now on we will concentrate on boxlike shaped objects.
Bischoff and Ratcliff [3] prove that the packing problem is NP-hard. Therefore heuristics and meta-heuristics are used to find a good solution. Hyde [15] introduces different meta-heuristics and offers a brief review of those methods. Unfortunately this methods are not suitable for unknown objects, because all objects must be present and their shape must be known before the planning and selection of the heuristic takes place.
There are many more solutions for the original container loading problem, for example [2, 9, 19]. All of them have to know the objects in general before planning.
We offer a solution to solve the two-dimensional packing problem for previously unknown objects, with respect to the constraint, that placed objects can be regrasped and unpacked out of the delivering area.

3 Planning Algorithms

In this section our grasp and delivery planner are introduced. The procedure of the planner is as follows: First, the object model is computed using a single camera image. The model of a object consists of its contour pixels (Figure 2a) and its minimum bounding box (Figure 2b). Second, the delivery planning is executed. Third, the grasp for the object is computed. The grasp planner is only launched, if the delivery planner could be executed successfully. Every time these steps are executed a new camera image is used.



(a) Contour pixel of the objects **(b)** Bounding boxes (light blue) around every object

Figure 2: Image based object model

Grasp Planning The grasp planner incorporates the object model and the assumption that the objects are located on a flat surface. The first step is to identify suitable grasp regions on the objects surface. The regions must be suitable to place a grippers jaw. Instead of using a single point to represent each region as in [21], we use straight lines, to take the physical size of the gripper into account. These regions are grouped into pairs, so that a grasp configuration can be computed. These pairs of grasp regions constitute the candidate set. Out of this set the best pair is chosen by some quality metrics and the gripper and robot configuration is computed to establish the real grasp.
The main component of the algorithm is a Hough transformation [12]. This transformation extracts edges on the objects, later used as grasp regions. These edges are automatically grouped by the Hough transformation in sets containing only parallel edges. Each pair of parallel edges is rated by quality metrics. One checks, if the two edges are long enough for the gripper jaws. The longer these edges are, the better is the rating for this pair. Another one checks, if the object fits in between the jaws of the gripper. This metric reduces the impact of the positioning inaccuracy of the robot and gripper. And the planner very robust against sensor noise.
Using the best rated pair, the configuration of the gripper and the robot is computed. With the Hough transformation, the orientation of the gripper is given. The position of the gripper and its opening diameter of the jaws are computed using the projection of the best candidate pair to the flat surface, where the object is located. The approach of the robot towards the object is force controlled to handle further uncertainties like the height of the objects.
The result is, that we are able to compute the full robot and gripper configuration from one image. The presented planning algorithm can easily be extended for n-finger grippers

or even hands by adapting the underlying function of the Hough transformation.

Delivery Planning The focus of delivery planning is on finding a solution for the two-dimensional online packing problem, so that the unknown objects can be regrasped later on. Since the objects are unknown, there is no possibility to perform an offline planning step as known from the classical container loading problem.

The locations where objects can be delivered are represented by free regions, selected by different criteria. Our solution approximates the grasped objects by their minimum volume bounding boxes. For regrasping the object by a standard gripper, we enlarge these boxes so that there is enough space between the already placed objects. The object is rotated, so that the bounding box is aligned with the boundary of the delivery area.

Before the first object is placed, the delivery area is represented by a free region covering the whole area. After the placement, the first free region is divided into three areas. The first is the bounding box of the object and the other two areas are the new free regions, with the property, that each region covers the maximum rectangular area inside the original free region. An object is heuristically placed at that corner of a free region, which is the closest to a corner of the delivery area. The free regions are stored in a list, sorted by the minimum distance between a free region corner and a corner of the delivery area. This order ensures that in doubt the free region is chosen, whose distance to a corner of the delivery area is minimal. For all further objects, one of the free regions in the list is selected, proceeding further on. In Figure 3 the state of the planner is shown after two objects (black boxes) were placed in the delivery area. The three free regions are represented as coloured boxes and their order in the mentioned list is shown on the right.

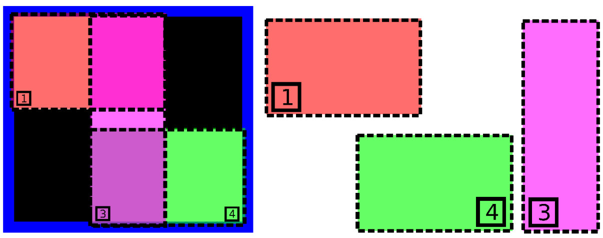


Figure 3: Left: State of the delivery area after placing two objects (black). Right: the order of the free-space list from left to right. The colours of the free-spaces vary, to to distinguish them. Their numbers show the order of their formation.

A free region is selected by the means of an objective function. We implemented three types: The first one (*firstfit*) takes the first free region the object's bounding box fits into. The second one (*bestfitVol*) computes the difference between the volume of the free region and the volume of the objects bounding box. The free region with the minimum difference is chosen. If there are more than one area with the same objective value the first one is chosen. The third objective function (*bestfitDist*) computes the distance between a free region and the objects bounding box for each axis. These two values are sorted lexicographically. And the free region with minimal objective value is chosen.

4 Experiments

In this section we evaluate the success of reaching our objectives. We investigate the performance of the planner by grasping and delivering several objects with our hardware setup. And we measure the runtime of each component, both planning and execution by the robot to evaluate whether we meet our most important objective being fast. We executed our planning algorithms with the 7 DoF KUKA LWR IV, a Schunk parallel jaw gripper and a standard webcam, mounted at the robot's wrist. Figure 4 shows the successes rates of the grasp planner and object modeller for a selection of our test objects. Both components were executed ten times on each object, always with a new camera image. A failure was counted when the real grip or the planning was not successful, caused by an inaccurate object model or selecting an unsuitable grasp region.

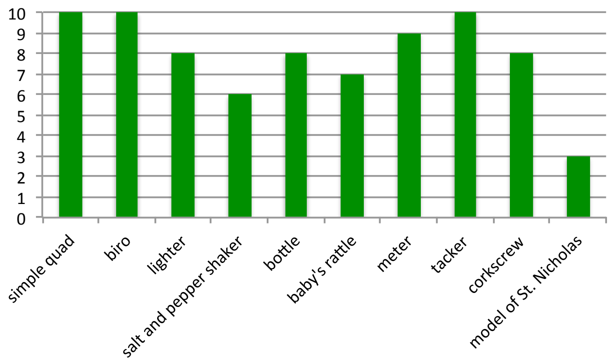
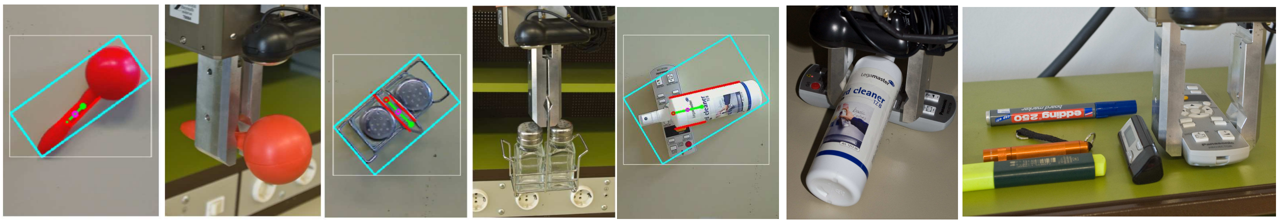


Figure 4: The graph shows the success of a grasp for several objects. For every object the text was executed ten times.

Robotic Experiments Figure 5a-c shows three of the test objects during the grasp execution and the corresponding camera image. These images show the grasp regions , the supporting line , which is used to rate the pair of grasp regions and the minimum bounding box.

The baby's rattle (Figure 5a) could be grasped successfully despite the fact that there is no strait line on the surface but rather a curved surface.

Even when the objects are located on top of each other (Figure 5c), the grasp planner is able to plan a proper grasp configuration and the robot executes the grasp successfully.



(a) Grasp of a baby's rattle **(b)** Grasp of a salt and pepper shaker **(c)** Grasp of overlapping objects **(d)** Delivery of the remote control

Figure 5: Sub-figures a-c show examples of the grasp planner and executer. The left image shows the camera image with the grasp regions (red), the supporting line (green) and the bounding box (light blue). Sub-figure d shows the delivery area while placing the remote control.

The object must be grasped exactly in the planned configuration, for the accuracy of the delivery. Each inaccuracy limits the delivery planner. In this case the bounding box around the object is not minimal, so the space-saving packing becomes worse. But this is a minor problem because crashing into another object or the environment is still avoided.

The result of tests is that the *bestfitDist* objective function has the best packing density up to 90% of the available space. The tests are based on the benchmarks from Bishoff and Ratcliff [3].

Figure 5d shows the delivering area filled up with several already placed objects. Currently the robot packs the remote control next to a black music-player. Note that the space between the gripper-jaws and the black music player is minimal.

Computation Time For the runtime measurements, we used a standard personal computer with a 2.66 GHz quad core CPU and 4 GB RAM. We measured the computation time of the grasp and delivery planner. Both planners including the computation of the object model were executed 6000 times. The average computation time of the grasp planner is 26.51 ms and the maximum was 34.45 ms. The maximum computation time of the delivery planner was 1.3 ms. This is together approximately the frame rate of the used camera.

5 Discussion

Since there is no general purpose robot for pick and place tasks in an unknown environment, like [17], we think that a bunch of behaviours for a wide set of situations needs to be developed. Combining these behaviours should give us a solution for such a robot. There are already some addressing the delivery of an object towards a person [17], overhanding an object to a persons hand [10, 13] and opening a door [16]. We offer a possibility to pack unknown objects to a given area, like a tray, as a part of the mentioned bunch of behaviours. These behaviour can be sued for butler-task or during the tidying up, while placing all objects together to get some free space.

All of these robots first need to grasp the unknown object. As shown in Section 2 there are many solutions. We focus on a fast solution with a minimum of sensor input, to reduce the costs and the complexity of the robot. Our approach is not limited to convex objects. Since we search for two parallel line segments with respect to a small curvature or noise, it does not matter whether the object is convex or has non-convex parts. The rattle in Figure 5a has convex parts and there is no truly even line on its surface boundary, anyhow the grasp planning is successful and the robots grip, too. Regarding the size of the gripper jaws minimizes the effects of external torque to the grip, like [5].

Our focus on minimizing the computation time of the planners is motivated out of the fact, that, during interaction with a human, the robot should act as expected, meaning pauses for planning during the movement or special movements for the object modelling should be non-existent. Our approach does not need or perform such movements.

6 Conclusion

We have succeeded in grasping and delivering unknown objects located on a table [1] using only one image is used. The grasp planner uses the Hough transformation to extract the grasp regions and to group them into pairs. Further on, the robot and gripper configuration is computed directly on the results of the Hough transformation. The planner is robust against noisy sensor data, takes the real size of the gripper jaws into account to reduce the effect of external torque and the tolerance with respect to the positioning of the robot is ensured. The delivery planning of the unknown objects packs the objects with maximum density in a given rectangular area, allowing the robot to regrasp them. The experimental results show that the two planners have the capabilities to handle various objects even if they have no perfect planar surface. The overall computation time for the object modelling, grasp and delivery planning is less than 36 ms, which is approximately the frame rate of the used camera. The system is able to tidy up a table, to pack

these objects into given rectangular delivery area, for example a tray, and to regrasp every stored object. Hence we have introduced another behaviour on the way to a general purpose robots for pick and place tasks.
Our next steps are to include depth image information to both planners and to remove the constraint, that the objects need to be located on a table in a known environment.

References

[1] J. BAUMGARTL, *Schnelles Greifen und Ablegen unbekannter Objekte mit einem Industrieroboter*, Master Thesis, University of Bayreuth, 2011.

[2] E. BISCHOFF, *Three-dimensional packing of items with limited load bearing strength*, EJOR, 2006.

[3] E. E. BISCHOFF AND M. S. W. RATCLIFF, *Issues in the development of approaches to container loading*, Omega, 1995.

[4] L. BODENBAGEN ET AL., *Learning to grasp unknown objects based on 3d edge information*, CIRA, 2009.

[5] G. M. BONE ET AL., *Automated modeling and robotic grasping of unknown three-dimensional objects*, ICRA, 2008.

[6] J. CORNELLÀ AND R. SUÁREZ, *Fast and flexible determination of force-closure independent regions to grasp polygonal objects*, ICRA, 2005.

[7] J. CORNELLÀ AND R. SUÁREZ, *Efficient determination of four-point form-closure optimal constraints of polygonal objects*, T-ASE, 2009.

[8] J. EGEBLAD ET AL., *Heuristics for container loading of furniture*, EJOR, 2010.

[9] M. ELEY, *Solving container loading problems by block arrangement*, EJOR, 2002.

[10] J. KIM ET AL., *Advanced Grasp Planning for Handover Operation Between Human and Robot: Three Handover Methods in Esteem Etiquettes Unsing Dual Arms and Hands of Home-Service Robot*, ICARA, 2004.

[11] N. GORGES AND H. WÖRN, *Learning an object-grasp relation for silhouette-based grasp planning*, Advances in Robotics Research, 2009.

[12] P. HOUGH, *Method and means for recognizing complex patterns*, U.S. Patent 3.069.654, 1962.

[13] M. HUBER ET AL., *Human-Robot Interaction in Handing-Over Tasks*, RO-MAN, 2008.

[14] K. HUEBNER ET AL., *Grasping known objects with humanoid robots: A box-based approach*, ICRA, 2009.

[15] M. HYDE, *A genetic programming hyper-heuristic approach to automated packing*, The University of Nottingham, 2010.

[16] A. JAIN AND C.C. KEMP , *Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator*, RSS Workshop on Robot Manipulation: Intelligence in Human Environments, 2008.

[17] A. JAIN AND C.C. KEMP, *EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. Autonomous Robots*, Autonomous Robots, 2010.

[18] Y. JIANG ET AL., *Efficient Grasping from RGBD Images: Learning using a new Rectangle Representation*, ICRA, 2011.

[19] S. MARTELLO ET AL., *Algorithm 864: General and robot-packable variants of the three-dimensional bin packing problem*, ACM Trans. Math. Softw., 2007.

[20] M. A. ROA AND R. SUÁREZ, *Computation of independent contact regions for grasping 3-d objects*, T-RO, 2009.

[21] A. SAXENA ET AL., *Robotic grasping of novel objects using vision*, IJRR, 2008.

[22] R. SUÁREZ ET AL., *Grasp quality measures*, Universitat Politècnica de Catalunya, 2006.

[23] V. TORRA ET AL., *Container loading for northogonal objects: an approximation using local search and simulated annealing*, Soft Computing, 2009.

[24] P. WHELAN AND B. BATCHELOR, *Automated packing systems-a systems engineering approach*, Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, 1996.

[25] K. YAMAZAKI ET AL., *A grasp planning for picking up an unknown object for a mobile manipulator*, ICRA, 2006.