Object Turning for Barcode Search

Derk REMBOLD Vladislav BELOGROUDOV Heinz WÖRN

Institute for Process Control and Robotics (IPR)
University of Karlsruhe, D-76128 Karlsruhe, Germany
e-mail: drembold@ira.uka.de
http://wwwipr.ira.uka.de/

Abstract

In this paper, a flexible manipulation system used in a catalog sale business is presented. One application of such a system is the object turning operation for the barcode search. The regrasping procedures are paid great attention due to its significance in object turning. Focus on this work is laid on regrasping tasks with several different grippers.

1. Introduction

In co-operation with a big catalog sale company, IPR conducts a project aiming at studying the use of robots for handling returned products in the stockrooms of the company. On average the customers return about 20% of the company products, which are subsequently sent to a special facility in Germany. From there the products are unloaded from trucks to conveyer belts. The products are examined individually for defects, sorted and stored in the stockrooms.

In such a facility, many processes, in particular, the sorting task, have to be carried out manually which are interrupting other automation processes. A sorting task can be viewed as following: products are at first put on pallets on a conveyer belt with barcode side facing up; as the products are carried by the conveyer belt to a certain point, the pallets tip over to throw the product on one of the slides located next to the conveyer belt. As the product reaches the end of the slide, a worker picks it up and places it for further transportation, again with the barcode side facing up, into a box, Fig.1.

Such a process can be automated by using a robot system, [7]. The automated process consists of the following steps:

a) a laser scanner located above the slide scans for the object thrown on the slide.
b) the laser scanner transmits the position and orientation of the product to a robot system. Then the robot system plans the grasping operation
c) a camera system monitors whether the barcode side of the product is facing up; if yes, the robot picks the product and places it into a box; if no, the robot turns the product until the camera system signals the barcode side is facing up. Then the robot picks the product and places it into a box.

Notice that the main goal of the robot system is to pick the object and place it into the box. Since sometimes the barcode is not on the top of the object, the robot system must be able to turn the object until a barcode is found. Only the functionality for turning the object until a barcode is found is topic of this paper.

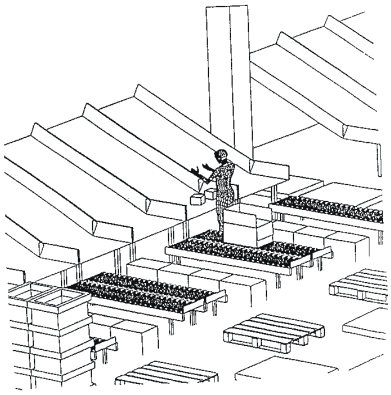


Fig. 1: Cell in Catalog Sell Business with Slides

1.1. Objective

The goal of this work is to build a robot system which can automate the above described process. The sizes of the boxes can vary largely and can change during the life-time of the facility. It is very difficult and expensive to design a general gripper which is capable of picking and turning every box due to mechanical limitations and constraints. For this reason we decided to use a gripper changing system, presently consisting of two parallel grippers. With these two grippers we are able to grasp all boxes used in the catalog sale business. However due to the repeating change of boxes, it must be possible to include easily new grippers. Grasp planning algorithms in previous works are usually limited to only one gripper. We extend the algorithms to all parallel grippers. Also we developed a system architecture where grasp planning algorithms for different grippers can be dynamically included into to control architecture of the robot system.

1.2. Previous Works

Previous works in turning objects using robots include an approach [11] which presents a system that is able to regrasp an object with a parallel gripper for assembly operations. However, this system is not suitable for our application, since they focus on off-line planning for assembly problems. [8], [9] and [10] offer an alternative regrasp planner that divides the regrasp algorithm into an off-line and an online-phase. In the off-line phase are planned valid grasp operations for an object, while in the on-line phase picking operations, placing operations, obstacles avoidance and the regrasp sequence are computed. Since we use mainly cuboidal objects in practice (however our algorithms are designed for any polyhedral objects) an off-line computation of all possible regrasp operations does not show a great advantage here.

Our goal is to study object turning operation in barcode search having a family of grippers from a gripper changing system. If necessary we must be able to include another gripper to the gripper changing system. This differs from the goals of [8], [9], [10] and [11]. Their goals are to study a regrasp operation for a place operation in an assembly task and their approaches and algorithms are limited to only one gripper.

[1] presents another approach in turning objects. An orienting mechanism tilts a table to turn an object until it reaches a stable position. A planner computes a sequence of tilting operations to minimize the uncertainty in the part's orientation. [4] presents a manipulator with two joints and a conveyor which knocks objects onto new faces. The goal is to design a sensorless part feeder which orientates 3D objects. The goal of [5] is to orientate 3D objects from an initial position to an desired one. For this [5] uses a simple manipulator consisting of a fixed plate and a movable plate. The object is placed between the plates. By moving one plate the object can be moved to new positions. [1], [4] and [5] discuss less expensive methods to orientate an object. However a robot is needed anyway to pick and place the object in the catalog business application. Therefore we use the robot to perform the turning task.

2. Determining a Grasp

In order to avoid the collision between the gripper and its environment (e.g. the slide), it is necessary to determine the grasp region for each object.

Firstly, the object's position and orientation need to be acquired and converted to a computer model. Since the success of a gripping action does not only depend on the position and orientation of an object, but also on the environment of the robot and its kinematics, the last two factors need to be treated in the same manner.

The determination of the grasp region will be established for each kind of gripper in two steps [6], [9]:

- finding of a symbolic grasp
- finding of a numeric grasp

The grippers (parallel four-finger gripper, parallel two-finger gripper) are presented and the symbolic and numeric grasp are explained in great detail.

2.1. Parallel Gripper

Two kinds of parallel grippers are used in our project, see Fig. 2: the first kind has four fingers with a length of 0.15m for each finger and it cannot close completely; the second kind has two fingers with a length of 0.15m for each finger and it can close completely for any cases. These two kinds of grippers can be applied for tasks depending on their opening ranges.

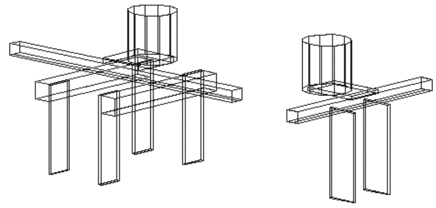


Fig. 2: Two Kinds of Parallel Grippers

Finding of a Symbolic Grasp

For grasp planning, the robot system models the objects and the grippers as polyhedrons, which are represented by a combination of geometric primitives which are also referred as *grasp features*: such as faces, edges and vertices. A symbolic grasp is realized when any pair of an object's grasp features F_1, F_2 is parallel approachable by a parallel gripper, e. g. two parallel facets of the object. More details for finding a symbolic grasp are specified in [6], [7] and [9].

Finding of a Numeric Grasp

Given a symbolic grasp a local grasp operation is realizable, however not necessarily a global grasp operation. Another prerequisite for realization of a grasp operation is a non empty gripping zone (GZ). A gripping zone GZ is said to be found when the projections of two grasp features F_1, F_2 on the grasp plane GP, which equally divides the space between the grasp features, intersect with one another [9]:

$$GZ = \prod(GP, F_1) \cap \prod(GP, F_2)$$

where $\prod$: $\Re^3 \times \Re^3 \rightarrow \Re^2$ is the projection operation. The grasp determination as a result, is reduced to a problem involving only three parameters: the position parameters ξ, ψ and the orientation parameter α. Growing is used to find the configuration space of the moving fingers of the gripper which are projected to the grasp plane GP. This yields a polygon GF_α (see Fig. 3) depending on its orientation α on the grasp plane GP:

$$GF_\alpha = \prod(GP, GF[0,0,0,\alpha])$$

where GF[x, y, z, α] is the polyhedral model of the gripper fingers; x, y, z is the position of the gripper, α is the orientation of the gripper about the normal vector of the grasp plane GP. Note that the grasp surfaces of the fingers must be parallel to the grasp plane, in order to perform the above operation.

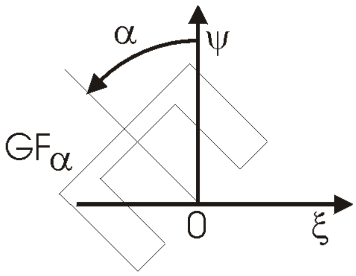


Fig. 3: Projection of Jaws of Four-finger Gripper on Grasp Plane GP for α

The *gripper configuration* can therefore be derived as a set difference „-“ between the gripping zone GZ and the gripper finger polygon set GF_α [2]:

$$CO^{\alpha}_{GF_\alpha}(GZ) = GZ - GF_\alpha$$

This gripper configuration can be interpreted as a region where the origin of the gripper fingers can be moved, so the gripper is able to grasp the object. However, there is no guarantee that the gripper will not collide with an obstacle, e. g. a slide. Therefore, the configuration space of the obstacles has to be subtracted from the gripper configuration.

The configuration space of the obstacles is constructed as follows: given two planes, perpendicular to the opening direction of the gripper fingers (parallel to grasp plane GP), which are laid through the maximum opening positions of the gripper fingers or through the ends of the gripper palm. Both planes span a gripping volume GV. Any obstacle located within this gripping volume GV is projected to the grasp plane GP to generate a set of obstacle polygons B_i. The configuration space of all obstacles are generated by growing the obstacle polygons with the gripper polygon GF_α:

$$CO^{\alpha}_{GF_\alpha}(B) = \bigcup_i (B_i - GF_\alpha)$$

where the “-” operator corresponds to set difference.

An approachable gripper configuration CA^α is computed with the polygon difference operation “/”:

$$CA^\alpha = CO^{\alpha}_{GF_\alpha}(GZ) / CO^{\alpha}_{GF_\alpha}(B)$$

The gripper configuration CA^α can be viewed as a region where the origin of the gripper finger is able to grasp the object without colliding with an obstacle. The theories described above can only be applied to convex objects. The palm of the gripper as an obstacle is not described here, as well. A more detailed description can be found in [3], [9].

3. Turning the Object

In object turning operation for finding barcodes, we need information about stable poses of a given object. Such a stable pose will be called *placement space* PS^φ where φ is an angle of an object’s rotation about the normal of the deposit place (at the end of slide), [11].

A transformation of an object from a placement space PS_i^ϕ with fixed angle of rotation $\phi=\phi_0$ to a further placement space PS_j^φ of the object is required:

$$T(\varphi) = {}^{PS_j}T_{PS_i}$$

Given an approachable gripper configuration CA^α for the initial placement space PS_i^ϕ with $\phi=\phi_0$, we need to define another gripper configuration CA_i^α for a complete turning operation to a further placement space PS_j^φ.

With the help of the transformation T(φ) derived above we transform CA^α to a region $R^\varphi \in \Re^3$ on the grasp plane GP^φ, $R^\varphi \subset GP^\varphi$, of the object at the placement space PS_j^φ. The projection of the region on the grasp plane GP^φ will result to the gripper configuration of the turned object:

$$\overline{CA}_i^{\alpha} = \prod(GP^\varphi, R^\varphi)$$

Yet, this gripper configuration is not necessarily approachable (the gripper can collide against an obstacle after it was moved by the turning operation). Therefore, the obstacle configuration of the placement space PS_j^φ of the turned object has to be taken into account:

$$CO^{\alpha}_{GF^\varphi_\alpha}(B) = \bigcup_i (B_i - GF^\varphi_\alpha)$$

while GF_α^φ is the projected gripper, B_i represents a projected obstacle and „-“ is a set difference operator. An approachable gripper configuration, also called *grasp space* GS_i^α, for the complete turning operation from one placement space PS_i^ϕ with a fixed angle of rotation $\phi=\phi_0$ to a following placement space PS_j^φ is:

$$CA_i^\alpha = GS_i^\alpha = \overline{CA}_i^{\alpha} / CO^{\alpha}_{GF^\varphi_\alpha}$$

where “/” is the polygon difference operator.

Algorithm

We create a list of placement spaces PS_i^φ, i = 1 .. n, where n is the number of stable poses of an object.

The next problem we are facing is how to carry out the turning operation. The turning operation from one placement space to another is constrained by [9]:

- kinematic of the manipulator
- obstacles causing collision

Since the probability to find a barcode on a large face is high there will be an additional constraint:

- large faces should be observed first

In case of an object with many facets it takes significant amount of time to examine all possible positions of the object. However, viewing all faces of an object is not necessary. It's practical to define a parameter for visibility which describes how many facets can be examined. Now the computing time can be reduced significantly. After each manipulating step we examine which facets were viewed from the camera position. If all of them have been checked we stop the turning operations. This is a very important addition because in the case of complicated object this accelerates computing.

Our next goal is to optimize the sequence of the object's turning operations until the camera system detects the barcode. We approach the problem using an iterative deepening search.

The turning operation of the object will be described as a triple:

$$(PS_i^{\phi}, GS_i^{\alpha}, PS_j^{\varphi}),\quad i \neq j,\quad \phi = \phi_0$$

where PS_i^{ϕ} is the placement space for an initial stable pose, GS_i^{α} is the grasp space and PS_j^{φ} is the placement space for the following stable pose in the turning process.

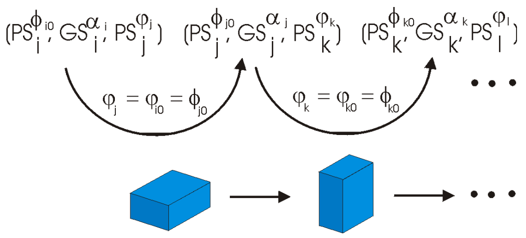


Fig. 4 List of Turning Operations for an Object

The following algorithm generates the list of all possible turning operations and computes a list of turning operations for optimal manipulation, Fig. 4. We build five necessary structures:

- list of <available> placement spaces
- list of <wrong ways> of turning operations
- <current way> of turning operations
- <current> placement space
- list of <viewed> faces

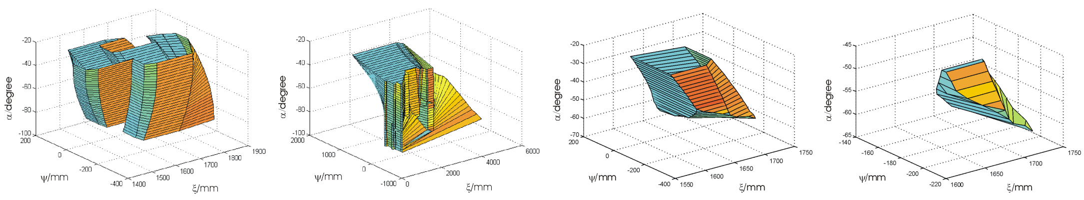
<available> consists of all placement spaces, which are to be considered but not used yet in <current way>. <current way> includes a sequence of turning operations (PS_i, GS_i, PS_j), where GS_i is a non-empty grasp space, therefore PS_i is excluded from <available>. <current> is the last placement space $PS_{current}$ of the last turning operation in <current way>, which represents the current pose of the object. <viewed> is a list of all faces viewed by the camera system previously and presently. Sometimes, a situation we call "deadlock" occurs; the turning operation cannot precede the next step because the grasp space GS is empty. To solve the problem, we need to move the current sequence of turning operations <current way> to <wrong ways>, and step back to a previous sequence of turning operations. In every step we search for next placement space PS_k. We compare all <available> placement spaces and choose the placement space with the largest faces viewed by the camera system, at the same time, create a valid turning operation (PS_k, GS_k, $PS_{current}$), which consists of a non-empty grasp space GS_k, with the placement space $PS_{current}$ from <current>. Also, we need to make sure that <current way> with this position equals to none of the <wrong ways>.

4. Example

For computing one single turning operation of a box located at the bottom of the slide, we need several configuration spaces, Fig. 5. The gripper angle α ranges from –25 degrees to –84 degrees. Angles in a range between –25 degrees to 0 degrees are not realizable, since the turning operation in such angle range will result to collision between gripper and walls of the slide. Fig. 5a shows the gripper configuration with a four finger gripper. Fig. 5b displays the obstacle configuration with the four finger gripper. By intersecting these two configuration spaces an approachable object configuration is formed, see Fig. 5c. Any point within this space corresponds to a point on the grasp plane, which can be reached by the four finger gripper from an approaching direction (expressed by α and the grasp plane) without colliding with the slide. At this point, the gripper is able to grasp the object. However, not all grasps points will result to a valid turning operations. Fig. 5d shows a approachable object configuration for the turning operation or the *grasp space*. The space becomes smaller because of the collisions which will occur during placing the turned object. Any point in the grasp space can be used as a valid grasp point to approach the object and turn the object from one placement space to another without colliding against the slide. A grasp point will be chosen from the following:

- distance to the center of gravity of an object
- overlapping area of a gripper with object
- distance to an obstacle

Sliding and rotating of the object within gripper are not considered here and will be subject of future work.



a) (b) (c) (d)

Fig. 5: Configuration Spaces (from left): Gripper Configuration (a), Obstacle Configuration (b), Approachable Object Configuration (c), Grasp Space (d)

5. Control architecture

Grasp planning algorithms and regrasp planning algorithms are realized as components, which can be loaded dynamically into the control architecture of our robot system. Presently the communication platform used for the gripper components is CORBA. The gripper components loaded into the control system receive the object's position and orientation from the camera system. They immediately send back an evaluation to the robot controller about their fitness to grasp the object. Then, the robot controller orders the gripper component with the best evaluation to calculate a collision free grasp. Here the current gripper attached on the robot must considered, because a change of a gripper is very time consuming. The chosen gripper component sends back a valid solution after calculating the gripping positions. Then the robot controller is planning and executing a trajectory for the grasp operation.

The regrasp component uses the services of the gripper components as well. If the camera system cannot find a barcode on the object, a regrasp operation is executed. The regrasp planner module computes the sequence of turning operations for searching the barcode. For each turning operation, a *grasp space* for picking and placing the object has to be calculated. The regrasp component sends orders to the gripper components which will return their solutions. While the regrasp planner computes the next turning operation, the robot will carry out previously generated program.

6. Experimental Results

Experiments were carried out in a demo cell in our institute, using two parallel grippers, a KUKA robot, and a laser scanner. In Fig. 6, a single turning operation of a box is shown. Fig. 6a illustrates the approaching position of a four finger gripper before the turning operation. Fig. 6b indicates the gripping position of the gripper. After the object is turned the robot moves to the approaching position (Fig. 6c) and places the turned object to a designated place on the slide (Fig. 6d). In Table 1, the computation time of a single turning operation for several grippers are shown. The measurements have been taken by using a SGI, 270MHz MIPS R12000 Processor. Two available parallel grippers are tested: a two finger gripper and a four finger gripper, Fig. 2. Additionally, we present the computation times for a two finger parallel gripper from [10] for comparison. In order to generalize our algorithms to cover all parallel grippers of different shape and size, we have to sacrify computing efficiency. The use of different computer systems also creates difficulties in comparison of our two algorithms. Note, that the computation time for the grasp classes and the grasp space of a four finger parallel gripper is very high. This is caused by the fact that the four finger parallel gripper is represented as a non-convex polygon on a grasp plane, see Fig. 3.

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

Fig. 6: Turning Operation

Parallel Gripper	*Grasp classes*	*placement spaces*	*Grasp space*	*Total*
Two Finger	0.28	0.1	1.27	1.65
Two Finger [10]	0.21	0.05	n. a.	n. a.
Four Finger	0.9	0.1	6.17	7.17

Table 1: Computation Times for a Turning Operation of a Cuboidal Object for several Parallel Grippers in Seconds

7. Conclusion and Future Work

In this paper, the use of a robot system for searching barcode on an object is described in great detail. The work presented in this paper differs from previous work [7], [8], [9], [10] and [11] in the following ways:

- objective (finding a barcode)
- easy inclusion of new grippers into control system
- general grasp planning algorithms for parallel gripper
- object turning to all viewable sides
- on-line computation of turning operations

Such a system described here finds its applications in a catalog sale businesses, where objects are required to be placed into boxes with the barcode facing up. Computation for collision avoidance during grasping and computations for grasp points are done by using configuration space methods. Goal is to turn the object using the robot until a camera detects a barcode. A regrasp planner finds an optimal turning sequence, with which larger faces will be examined by the camera first, since it is more likely to find the barcode on them.

Acknowledgments

This research work was performed at the Institute for Process Control and Robotics, Prof. Dr.-Ing. H. Woern, Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, Department of Computer Science, University of Karlsruhe, D-76128 Karlsruhe, Germany. The research is funded by the Bmbf (Bundesministerium für Bildung, Forschung, Wissenschaft und Technologie, Germany). Also special thanks to Lucy Chen for her efforts in editing this paper.

References

[1] Erdmann, M., Mason, M. T., Vanecec, G. Jr. Mechanical Parts Orienting: The Case of a Polyhedron on a Table. Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California, April 1991, pp. 360-365.

[2] Lozano-Perez, T. Spatial Planning: A Configuration Space Approach. IEEE Transactions on Computers, Vol. C-32, No. 2, February 1983, pp. 108-120.

[3] Lozano-Perez, T., Jones, J. L., Mazer, E., and O'Donnell, P. A. HANDEY, A Robot Task Planner; Cambridge, Massachussetts; The MIT Press, 1992.

[4] Lynch, K. M. Toppling Manipulation. Proceedings of 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan, May 1999, pp. 2551-2557.

[5] Marigo, A., Chitour, Y., Bicchi, A. Manipulation of Polyhedral Parts by Rolling. Proceedings of 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997

[6] Pertin-Troccaz, J. Geometric reasoning for grasping: a computational point of view. CAD-Based Programming for Sensory Robots. NATO ASI Series, Vol. F50, Springer-Verlag Berlin Heidelberg 1988, pp. 397-423.

[7] Rembold, D., Fischer, T., Müller, M., Wörn, H.: A flexible manipulation system for handling a large variety of objects; CIACIM, Troy, October 1998.

[8] Röhrdanz, F., Wahl, F. M. Generating and Evaluating Regrasp Operations. Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 2013-2018.

[9] Röhrdanz, F. Modellbasierte automatisierte Greifplanung. Dissertation. Technische Universität Braunschwaig, Institut für Robotik und Prozeßinformatik, Germany. Shaker Verlag, 1998.

[10] Stoeter, S. A., Voss, S., Papanikolopoulos, N. P., Mosemann, H. Planning of Regrasp Operations. Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan, May 1999, pp.245-250.

[11] Tournassoud, P., Lozano-Perez, T. and Mazer, E. Regrasping, Proceedings of 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, March 1987, pp.1924-1928.