Object Representation based on Grasping

Safoura Rezapour Lakani, Björn Ommer,
Antonio J. Rodríguez-Sánchez, Sandor Szedmak, Senka Krivic and Justus Piater

***Abstract*— Most human-made objects are composed of a configuration of parts whose design serves a certain functionality. As an example, a spatula is designed for scooping; the handle is the part of the object designed to grasp it in order to perform that operation. The functionality of an object's part and also the object can be related to its visual representation. In this paper we follow on that idea in order to infer the functionality of an object through its representation by parts. The focus is on graspable human-made objects; thus, the object representation is related to their graspable characteristics. Our evaluation on a robotic grasping scenario shows that our approach is efficient and robust as well as transferable to previously unseen, novel objects.**

I. INTRODUCTION

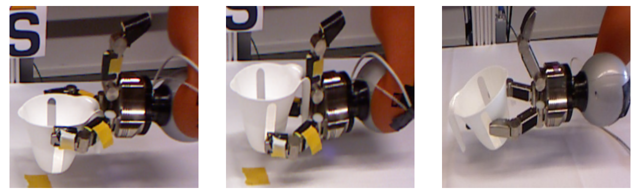
Grasping is an important functionality in robotic manipulation tasks such as stacking, assembling objects, object placement, screwing or pouring, just to name a few. In robotic manipulation scenarios, objects are initially perceived as an image or a point cloud. From the visual information, the robot must know how to grasp the object. An object can be grasped in many different ways as shown in Figure 1. The robot has to detect the graspable regions from the visual representation. From the three grasps in Figure 1, the one in Fig. 1(c) is the most *sensitive*. In other words, this grasp has a lower probability of being successful. Regions of lower sensitivity are more suitable for grasping, and vice versa.

Furthermore, it is not only the sensitivity of a particular region that has an effect on graspability, but also the sensitivity of its neighboring regions. Therefore, in addition to predicting graspable regions, their sensitivity in terms of a grasp-success probability should be obtained. This information is useful for grasping the least sensitive regions, and can reduce search time for finding such regions. Regions can be associated with a grasping sensitivity that, when considered along with their neighboring regions' sensitivities, has an effect on graspability. In this way, we can reduce the search space and focus on the less sensitive regions.

The main contribution of this paper is twofold: 1) Our method associates objects with their grasp parameters , and 2) encodes these, along with estimated grasp success probabilities, in the object representation. This allows for efficient grasp-parameter inference, avoiding the need to compute suitable gripper poses by separate means.

II. RELATED WORK

Robotic grasping has been closely associated with visual characteristics of objects. Grasping has been associated to a small set of object points (*patches*) [1], [2], [3]. These patches are either learned based on geometrical properties of object surface such as surface normals and curvatures [1], [3], [4] or from RGB edges [2]. These features are then used to classify graspable object patches. Often, these approaches provide fairly good detection results but their search space is very large.



(a) (b) (c)

Fig. 1. Grasping and local sensitivity of grasping. The container can be grasped in many ways. The grasp in Fig. 1(a) is stable after lifting. Also, when the gripper moves down as in Fig. 1(b), the grasp is still stable. But the grasp in Fig. 1(c) is more sensitive than others and the object might fall during lifting. Encoding grasp sensitivity information in a local neighborhood of a region is quite useful for finding the most stable graspable regions.

Part-based methods [5], [6], [7], [8], [9] can overcome this problem. Not only do they reduce the search space during recognition, but they also provide a framework for the generalization of grasping among different object categories that share the same parts In these methods, parts are usually segmented offline . Next, they either use an optimization procedure for finding gripper pose [10] or they make use of the object pose [11]. There are mainly two problems with these methods. First, they rely mostly on a motion planner for gripper placement which needs a large computation time. Second, the parts which are segmented offline are not necessarily useful for grasping.

To overcome these two deficiencies, we propose 1) a representation of objects based on their grasping parameters thus reducing computation time for finding the gripper pose. and 2) a method for segmenting the object into graspable regions that convey information relevant to the grasping task.

III. OBJECT REPRESENTATION FROM ROBOTIC GRASPING

A. *Learning from Robotic Grasping*

In order to learn graspable spots in objects, we performed robotic grasping experiment with two-finger grasps. Since, in one view both fingers of robot's gripper are not visible, we used two views with calibrated kinects. We provided the robot with pairs of candidate points. We obtained these pairs using normals and depth edges as following.

Fig. 2. Robotic grasping experiment to learn graspable object regions. From right to left: The images of the objects in two views, the computed contact pair, executed grasping and lifting object.

The pointcloud of the merged views is segmented into supervoxels based on the method [12] provided in the Point Cloud Library (PCL)[1]. Each pair of supervoxels are considered as candidate points. We filter out the pairs whose normal vectors point inwards. Furthermore, we computed depth edges in each view using Canny edge detection. We obtained the supervoxels which edges lie on them and consider all the possible pairs among them.

The obtained contact pairs specify the position of the gripper. In order to obtain the orientation, we fit an ellipse to the contact points in a fixed plane. The ellipse gives us only 2D rotation. In order to compute 3D rotation, we compute the third axis which is perpendicular to the principal axes of the ellipse. Considering e_x as the connecting axis of the contact points c_1 and c_2 and e_z as the other principal axis n which is determined by the ellipse fitting procedure, the third one e_y is computed as the cross product between e_x and e_y,

$$
\begin{aligned}
e_x &= c_1 - c_2 \\
e_z &= -n \\
e_y &= e_x * e_z
\end{aligned}
$$

This axis e_y can have two directions, we enforce it to have the same direction with the gravity axis g_v since our objects in training are positioned upright. e_x has also sign ambiguity, therefore we compute two rotation matrices based on different signs of e_x.

Given the contact pairs and their orientations, we perform robotic grasping on them as shown in Figure 2. Grasping is executed by holding an object and dropping it in a certain position. Each grasping is executed five times for a certain contact pair. We compute the grasping quality as the success probability of grasping for five executions. From the experiment, we get the graspable and non-graspable contact pairs as well as their probability of grasping. The grasping success probability is an important information for sensitivity of grasping in different regions of the object. Based on the experiment, we have two main information on the way to grasp the object and the grasping probability in different parts of the object. The former comes from the fitted ellipse for the gripper pose and the latter comes from the success probability of grasping.

[1] http://pointclouds.org/

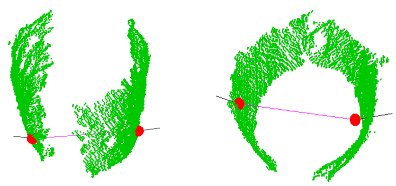


Fig. 3. Contact points and normals for a successfully grasped object.

B. Model Graspability from Vision

Considering the grasping experiment, we want to figure out the features which characterize the graspability. We consider five features for modeling graspability which we explain in following:

a) The length of ellipse principle axes: A grasp is represented with an ellipse based on the given contact points. The area of the ellipse is associated with the size of the gripper. Hence, the length of the principle axes of the ellipse determine the opening of the gripper and the convexity of the surface.

b) Collinearity between contact normals: Based on our experiments, it can be observed (Figure 3) that the contact normals are almost collinear with the contact points. We define collinearity as the inner product between the axis connecting contact points and the normals. Considering connecting axis as l and contact normals as n_1 and n_2, the collinearity is defined as,

$$
\begin{aligned}
\mathrm{colli}(l, n) &= l \cdot n \\
\mathrm{colli}(l, n_1, n_2) &= \frac{-\,\mathrm{colli}(l, n_1) - \mathrm{colli}(l, n_2)}{2}
\end{aligned}
$$

c) Symmetry: The grasped contact normals for a grasped ellipse have almost symmetry along the plane of the fitted ellipse. The symmetry is defined as the ratio between the inner product of plane normal and the contact normals. Considering plane normals as N for an ellipse e and contact normals as n_1 and n_2, the symmetry is computed as,

$$\mathrm{sym}(e) = \frac{N \cdot n_1}{N \cdot n_2} \tag{1}$$

In Eqn. 1, we always consider the larger value of the inner product in the denominator. Therefore, the symmetry value is always between zero and one. For graspable ellipses, this value is closer to one.

d) Normal distribution with respect to the neighbors: The above features are local and only based on the grasped area. We consider a feature which is semi-global and characterizes the grasped area with respect to its neighbors. Therefore, we compute the angle between the contact normals. Then, we consider all the neighboring pairs which lie on the same plane of the fitted grasp ellipse. We compute the angle between each pair normals and obtain the mean and variance based on them. Then we compute the probability of the contact normals angle based on the computed semi-global Gaussian distribution. In practice, this probability might be very small in some cases, therefore, we consider the negative logarithm of this probability as the feature.

e) Gradient of the ellipse movement: Our ellipse-based grasp representation does not encode the sensitivity of grasping. This sensitivity is defined based on the neighboring area. As can be seen in Figure 1(a), moving upwards will lose the grasp, whereas moving downwards will keep the same grasp. This information must be encoded into the grasp representation. We modeled the movement by motion along the principal axes of the ellipse plus the third axis which is perpendicular to them. We then move along each axis, inwards and outwards with steps defined as the length of the ellipse axes. In each step, we consider the fitted ellipse in that area and compute the ratio between its area and the reference ellipse area. The closer the ratio to one, the less sensitive the grasp in that direction and vice versa. From the computed ratios, we construct a six dimensional feature vector, one dimension for each direction. In each direction, we compute the mean of the area ratios.

C. Training a Grasp Model

Considering features mentioned in Section III-B, we trained a regression model for predicting grasp quality of the trained graspable/non-graspable features. However, learning such a model is possible, but it does not give us a promising accuracy. The reason lies on the fact that, the training data are not balanced. In other words, we have plenty of negative, non-graspable data. Therefore, we train two models. One classifier for detecting graspable versus non-graspable data. And a regression model for computing the grasp quality based on only graspable data and their grasp success quality. As we discuss in Section V, we tried Support Vector Regression with different kernels (RBF, Sigmoid and Linear) for training such models, but unfortunately they did not lead to an acceptable accuracy. Therefore, we employed the method discussed in [13] for this purpose. We then use the trained classifiers in the inference step to assign label to each ellipse based on its grasp quality. Furthermore, we merge the neighboring ellipses into a region. And we use these regions for robotic grasping purpose.

Fig. 4. Training objects for grasping experiments.

IV. INFERRING GRASPABLE REGIONS IN NOVEL OBJECTS

In this section, we explain the inference step for decomposing a novel object into graspable regions and assigning grasping probabilities to them. This inference procedure is composed of multiple steps, 1) obtaining candidate pairs, 2) detecting graspability and sensitivity of grasping, 3) decompsoition into graspable regions and 4) computing grasp parameter, gripper pose. We explain each step in detail in the following.

f) Obtaining candidate pairs:

g) Detecting graspability and grasp sensitivity:

h) Grasp-based decomposition:

i) Computing grasp parameter:

V. EXPERIMENTAL RESULTS

We evaluated our approach on two dataset, IKEA kitchen object and YCB [14] object dataset. The grasping contact points and their success probabilities are available on (grasp database[2]). The experimental setup for grasping experiments consists of a robot with two KUKA 7-DoF Light-Weight Robot 4+ arms with servo-electric 3-Finger Schunk SDH-2 dexterous hands. There are two kinects for capturing RGB-D data which are located in opposite of each other.

For learning purpose, we performed robotic grasping experiments with two-finger grasps on five simple geometrical shape objects as shown in Figure 4.

A. Offline Experiment

- with labeled graspable vs non-graspable contact points for objects
- divide to training and test set
- in total 43 objects, we performed cross validation on the objects, precision, recall.

[2] `https://iis.uibk.ac.at/public/GraspAnnotateDataset/`

Object	Proposed Method	Without Sensitivity	Depth edge points	Naiive Method
bottles	%	%	%	%
Boxes	%	%	%	%
Spatulas	%	%	%	%
Bowls	%	%	%	%
Mugs	%	%	%	%
Pitchers	%	%	%	%

TABLE I

QUANTITATIVE RESULTS FOR OFFLINE EXPERIMENT. GRASPABLE AND NON-GRASPABLE OBJECT REGIONS ARE LABELED BASED ON ROBOTICS GRASPING. THE DATASET IS DIVIDED INTO TRAINING AND TEST SET, 70% TRAINING AND 30% TEST DATA. LABELED OBJECT REGIONS ARE USED FOR TRAINING. WE THEN MEASURED THE ACCURACY OF THE PROPOSED SYSTEM IN RECOGNIZING GRASPABLE REGIONS ON THE TEST DATA BASED ON THE LABELING.

- compared with naiive approach of grasping based on random contact points selection
- compared with grasp selection based on only ellipses and no sensitivity
- compared with method grasping objects based on depth edge points like NG papar [2]
- see if planning is succesful and the contact points were labeled as graspable Table I

B. Grasping Experiment

- report on grasping based on different object categories, like balls, pitchers, containers, mugs.
- perform multiple trials, say 5 per object, and average accuracy
- compare with method in Section V-A
- show complexity level on object representation vs grasping success rate. report on grasping different geometrical shapes. from simple like cylindrical cans, bottles, or circular like balls to complex such as toys.

C. Generalization on Novel Objects

- report on wheter the fetures are generalizable. learn from simple geometrical shapes like cylindrical, spherical, cubics and generalize to more complex objects like pitchers, containers, toys, etc.

VI. CONCLUSIONS

- efficient grasping method because it decompose an object to a set of regions and considers only the central ellipse in each region
- grasping parameters are encoded into the feature representation hence no more inference is needed to obtain grasping parameters based on some optimization criteria [4], [10]
- can be further improved by using more views, by capturing different views from in-hand camera, (calibarted in pre-defined joints positions offline)
- computation of merging normals should be improved. that is the slowest part of the algorithm

APPENDIX

ACKNOWLEDGMENT

REFERENCES

[1] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” *IJRR*, 2015.

[2] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, “Learning to grasp objects with multiple contact points.” in *ICRA*. IEEE, 2010.

[3] A. Boularias, O. Kroemer, and J. Peters, “Learning robot grasping from 3-D images with Markov Random Fields,” in *IROS*. IEEE, Sept. 2011.

[4] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, “One shot learning and generation of dexterous grasps for novel objects,” *The International Journal of Robotics Research*, 2015.

[5] S. Fidler and A. Leonardis, “Towards scalable representations of object categories: Learning a hierarchy of parts,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[6] L. Zhu and A. L. Yuille, “A hierarchical compositional system for rapid object detection.” in *Advances in Neural Information Processing Systems*, 2005.

[7] B. Ommer and J. Buhmann, “Learning the Compositional Nature of Visual Object Categories for Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, 2010.

[8] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, 2010.

[9] C. S. Stein, M. Schoeler, J. Papon, and F. Wörgötter, “Object partitioning using local convexity,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[10] S. Stein, F. Worgotter, M. Schoeler, J. Papon, and T. Kulvicius, “Convexity based object partitioning for robot applications,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 3213–3220.

[11] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Y. Ng, “Learning to grasp novel objects using vision.” in *ISER*, 2006.

[12] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, “Voxel cloud connectivity segmentation - supervoxels for point clouds,” in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2013.

[13] S. Szedmak, E. Ugur, and J. Piater, “Knowledge Propagation and Relation Learning for Predicting Action Effects,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 623–629.

[14] A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel, “BigBIRD: A large-scale 3D database of object instances,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.