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

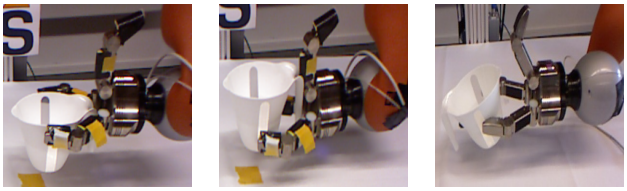
Grasping is an important functionality in robotic manipulation tasks such as stacking, assembling objects, object placement, screwing or pouring, just to name a few. In robotic manipulation scenarios, objects are initially perceived as an image or a point cloud. From the visual information, the robot 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.

After computing gripper pose based on each contact pair, we performed robotic grasping experiment as shown Figure 2. The experiment is performed, by executing a grasp based on computed gripper pose and lifting the object. 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/

B. Model Graspability from Vision

Considering the grasping experiment, we want to figure out the features which characterize the graspability. We consider four 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,

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

Since the normal vectors point outwards, the collinearity $\text{colli}(l, n)$ is always negative. In our learning procedure, we are interested in working with positive values, therefore the negative sign of the collinearity value is considered.

c) Normal distribution with respect to the neighbors: In the ideal case for grasping, the contact normals are completely collinear with the contact points thus the angle between them is exactly 180 degree. However, this depends on the convexity of the object surface. Therefore, the information of the surface must be encoded along with the collinearity. In the grasping case, mostly the angle between the contact normals is maximum among its neighbors. To

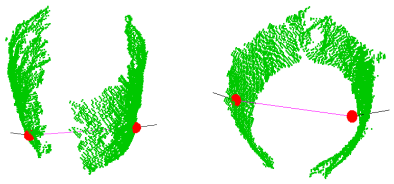


Fig. 3. Contact points and normals for a successfully grasped object.

this end, a relationship based on neighboring contact normals must be defined. We considered a feature for representing the relation between contact normals of an ellipse with respect to their neighbors. Therefore, we compute the angle between the neighboring contact pairs of an ellipse. We then compute the Gaussian distribution of these angles. Next, we compute the probability of the ellipse contact normals with respect to the computed distribution.

d) 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.

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) decomposition into graspable regions and 4) computing grasp parameter, gripper pose. We explain each step in detail in the following.

e) Obtaining candidate pairs: The first step in the inference is to obtain the candidate pairs. We collect them in the same way as discussed in Section III from supervoxels and depth edge points. Since, we represented a grasp with an ellipse, the next step is to fit an ellipse to the contact pairs. As we work with 3D pointclouds, there are many ellipses which can be fitted. Therefore, we consider the ellipse which can be best fitted to the surface of object in the area of contact points. Furthermore, based on our grasping experiment, there are elevations which the object cannot be grasped, for example from the table. We encode this two criteria into a score, and select the ellipse with the maximum fitting score in a acceptable elevation,

$$e^* = \underset{e}{\operatorname{argmax}}\, \text{score}(e) \tag{1}$$

$$\text{score}(e) = \alpha p(\phi|g) + (1-\alpha)(\text{fit}(e)) \tag{2}$$

We select the ellipse e^* which has the maximum score as given in Eqn 1, where ϕ indicates the elevation of the fitted plane. $p(\phi|g)$ is the probability of an elevation for grasping, which is obtained from our grasping experiments. α is set to 0.5 in our experiment.

f) Detecting graspability and grasp sensitivity: Given candidate pairs and fitted ellipses, we compute the features as described in Section III. We then exploit our trained classifier to filter out the graspable versus non-graspable contact pairs and ellipses. Next, we use our trained regression model and assign grasp quality probability to each graspable ellipse based on their computed features.

g) Grasp-based decomposition: From the provided ellipses along with their grasp probabilities, we merge the neighboring ones which have the same probability together into a region. In addition to the graspable ones, we merge also the non-graspable ellipses into the same region. In this step, there might be overlapping regions, due to the overlapping ellipses which can cause uncertainty. In order to reduce the uncertainty, we compute a matrix which indicates whether each two ellipses and hence their regions overlap or not. The overlapping regions provide us with different representations of the object based on the structure of the object. We utilize these regions along with their probabilities for computing grasp on objects. The regions are only a combination of ellipses with the same grasp probability.

h) Computing grasp parameter: Provided with object regions and their grasp probabilities, we compute the most probable region for grasping. We then compute the grasp for

Feature Type	Ellipse Axes Length	Collinearity	Normal Distribution	Gradient of the ellipse movement	Combined
Precision	94.85%	95.83%	94.04%	94.53%	93.84%
Recall	96.42%	97.29%	93.43%	95.87%	94.59%
F1 norm	95.41%	96.43%	93.69%	95.1%	94.08%
Accuracy	95.59%	96.61%	95.08%	95.59%	94.91%

TABLE I

CLASSIFICATION PERFORMANCE ON 10 FOLD CROSS VALIDATION BY LABELED GRASP EXAMPLES. THE PERFORMANCE OF EACH INDIVIDUAL FEATURES AND THE COMBINATION OF THEM IS REPORTED IN EACH COLUMN.

the most graspable region. The grasps are encoded as ellipses inside each region. There are many ellipses in a region which all of them have the same grasping probability. The only difference between them are those which are on the boundary of the regions, due to the uncertainty of their graspablitiy based on their neighbors. Therefore, the middle ellipses are more certain for grasping. We consider only the middle ellipse inside the region, and compute the gripper pose based on that. The gripper pose is computed as mentioned in Section III. The region-based grasp computation is a computational boost in our approach due to two reasons. For one, regions carry information about grasp probability and we favour for the one with the maximum probability. For another, the gripper pose is encoded into the region representation, and we consider only one ellipse among them.

V. EXPERIMENTAL RESULTS

We evaluated our approach on two dataset, IKEA kitchen object and YCB [14] object dataset. 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. The learning procedure is already explained in Section III. The grasping contact points and their success probabilities are available on (grasp database[2]). From only this small set of objects, we obtained many grasping examples. We evaluated our method in three different scenarios:

- Offline experiment for evaluating grasp classifier
- Grasping experiment on simple geometrical shape objects with varying poses
- Grasping experiment on novel complex shape objects

A. Offline Experiment

We evaluated the accuracy of the grasp classification on different features as discussed in Section III. The performance is reported for classification of graspable versus non-graspable grasp ellipses with 10 fold cross validation. The performance is measured based on precision, recall, F1 norm and accuracy. The results are given in Table I. The performance of each individual feature as discussed in Section III is given in Table I. Furthermore, the performance of their combination is reported in the last column of the table. As can be observed in the results, for detecting graspable versus non-graspable ellipses, each individual feature is already enough for this purpose. The measures such as the length of ellipse axes which are correlated with the opening and size of the gripper play an important role in detecting graspable ellipses. The combined one does not provide us with a better performance. The reason can be due to the confusion of the combination of features. The combination based on decision trees or random forest might give better performance which are considered as the future work.

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

Fig. 4. Training objects for grasping experiments.

VI. CONCLUSIONS

All in all, we proposed a method for representing objects based on grasping. We encoded grasping parameters (gripper pose) as well as quality of grasping in terms of its local sensitivity into our object representation. We focused on a representation in the right level of granularity. To be more precise, our representation is neither too local, which cannot be generalized to novel objects neither too global, which is dependent to the global geometrical shape of object. The representation is independent of the global object pose and is generalizable to novel and even complex objects.

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)*. IEEE, 2007.

[6] L. Zhu and A. L. Yuille, "A hierarchical compositional system for rapid object detection." in *Advances in Neural Information Processing Systems (NIPS)*, 2005, pp. 1633–1640.

[7] B. Ommer and J. Buhmann, "Learning the compositional nature of visual object categories for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, pp. 501–516, 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.