# Object Representations by Graspability-Derived Regions

Safoura Rezapour Lakani, Antonio J. Rodríguez-Sánchez, and Justus Piater

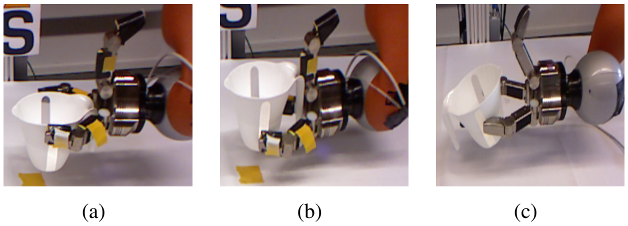
***Abstract*— Most human-made objects are composed of a configuration of parts whose design serves a certain functionality. As an example, a spatula is designed for scooping; the handle is the part of the object designed to grasp it in order to perform that operation. The functionality of an object's part and also the object can be related to its visual representation. In this paper we follow that idea 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 graspability characteristics. We propose a representation that encodes grasp parameters as well as the sensitivity of a grasp together with the object representation. We aim for a representation at the right level of granularity, specific enough to predict graspability with high accuracy but also generalizing to novel objects, while being computationally efficient. Our evaluation on publicly-available object sets shows that our approach is efficient and robust as well as transferable to previously-unseen 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 central contribution of this paper is a method for representing an object as a compound of parts that are characterized by their empirical graspability. During training, region descriptors are constructed and characterized by their grasp parameters (gripper pose), probability of grasp success, and sensitivity of the grasp to precise gripper placement. Novel objects are parsed into regions whose graspability can be estimated using the trained model.


(a)

(b)

(c)

Fig. 1. Grasping and local sensitivity of grasps. 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.

## 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.

Part-based methods [5], [6], [7], [8], [9] can overcome this problem. Exploiting object parts for robotic grasping has been addressed in the work discussed in [10], [11]. In the work mentioned in [10], object parts are segmented offline and grasps are associated to them. Next, an optimization procedure is used to find gripper pose in a novel object with similar part. There are mainly two problems with these methods. First, they consume a large computation time for gripper placement. 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 grasp 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. Learning the Graspability of Object Regions

### A. Characterizing a Potential Grasp by an Ellipsis

To train our model, we evaluate large numbers of grasps using a parallel-jaw gripper. Our method assumes that both contacts correspond to (groups of) points present in the point cloud. Since this is rarely the case in single depth images, we fuse two views taken by calibrated depth sensors with

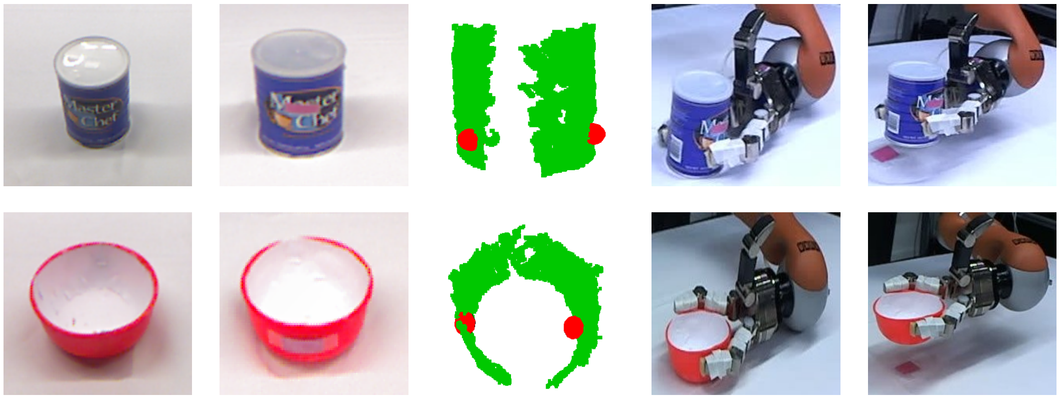


Fig. 2. Robotic grasping experiment to learn graspable object regions. From right to left: The images of the objects in two views, the computed contact pair, executed grasping and lifting object.

orthogonal views to maximize the observation. To obtain candidate grasps in the form of pairs of contacts, we proceed as follows.

First, the point cloud of the merged views is segmented into supervoxels [12] using an implementation provided by the Point Cloud Library[1]. Considering the fact that an object cannot be grasped from inside, we discard supervoxels pairs whose normal vectors point inwards. However, discarding these pairs is useful and reduces search space for grasping, but it is also noise prone. Normal estimation on borders or small object parts is very noise prone. Therefore, in this step we might object areas like rims or handles which are quite useful for grasping. Since, these areas are mostly on the borders or edges of the object, we include depth edges. We compute depth edges in each view using Canny edge detection. Then, we retain supervoxels that border on such depth edges, and consider all possible pairs among them as candidate pairs of grasp contacts.

Next, we compute a the least-squares ellipse containing the two contact pairs. 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 $e_x$.

Candidate grasps computed in this way are then executed by our robot (Figure 2), which moves the hand to the given pose, closes the grip, and lifts the object. Each candidate grasp is executed five times, yielding a coarse estimate of the grasp success probability.

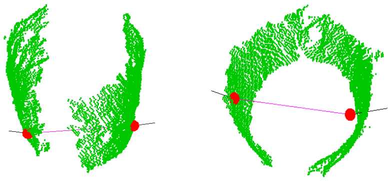


Fig. 3. Contact points and normals for a successfully grasped object.

This procedure provides us with two important pieces of information associated with a fitted ellipse, the gripper pose and the probability of grasp success.

### B. Features for Predicting Graspability

Our objective is to predict graspability from ellipses extracted from novel objects. To this end, we extract the following features that we expect to be predictive.

*a) Lengths of ellipse principle axes:* A grasp is represented with an ellipse based on the 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 of contact normals:* For high-quality grasps with a parallel-jaw gripper, contact normals are close to collinear with the closing direction of the gripper. We measure this collinearity as the average inner product between the axis $l = c_1 - c_2$ connecting the contact points and the contact normal $n_1$ and $-l$ and the contact normal $n_2$:

$$\mathrm{colli}(l, n_1, n_2) = -\frac{1}{2}\left(l^{\mathrm{T}} n_1 + -l^{\mathrm{T}} n_2\right)$$

Since the normal vectors point outwards, the collinearity $\mathrm{colli}(l, n)$ 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.

[1] http://pointclouds.org/

*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. This information can come from surface curvature or normal distribution in a local neighborhood. 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) Sensitivity:* Our ellipse-based grasp representation does not encode the sensitivity of grasping to the precise placement of the contacts. We define this sensitivity based on the relative sizes of neighboring ellipses. 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 included in the grasp representation. We model it by considering six movements of the ellipse along the principal axes and along their normal. The displacement in each direction is equal to the supervoxel resolution. The total length of displacements along a direction is equal to the length of the respective axis of the ellipse. For the normal axis, the length is considered as the maximum length of the principle axes. In each movement, we locally fit an ellipse and compute the ratio between its area and the area of the reference ellipse. The closer the ratio to unity, the less sensitive the grasp in that direction, and vice versa. These six ratios constitute a six-dimensional feature vector. In each direction, we compute the mean of the area ratios.

### C. Training a Grasp Model

We are interested in detecting graspability and sensitivity of grasping. We splitted this problem into two parts. For the graspability, we train a classifier which detects graspable versus non-graspable data. We train a classifier on the basis of the above features based on graspable and non-graspable training data. For the sensitivity, we considered only the sensitivity feature from the graspable data. Next, we group the graspable data based on their grasp success. We then perform clustering on each group based on the sensitivity feature. We used Spectral Clustering [13] for this purpose. Now, each cluster encodes a different grasp sensitivity. This grasp sensitivity can be computed directly from the mean sensitivity feature in each cluster. Considering the cluster means as $\mu_c$, the predicate indicating grasp as $g$ and movement along each direction $m$, $p(g|c)$ is computed by a marginalization over all the movement directions,

$$\begin{aligned} p(g|c) &= \sum_{m \in M} p(g, m|c) \\ &= \sum_{m} \frac{p(g, m, c)}{p(c)} \\ &= \sum_{m} \frac{p(g|m, c)p(m, c)}{p(c)} \\ &= \sum_{m} p(g|m, c)p(m|c) \end{aligned}$$

$p(m|c)$ is uniformly distributed $p(m|c) = \frac{1}{M}$. $p(g|m, c)$ is the probability of grasping along the direction $m$ and is computed as based on a Gaussian distribution with unit mean and fixed standard deviation. We considered unit mean, since the sensitivity is encoding the area ratio, and in the best case the ratio should be one.

$$p(g|m, c) = \mathcal{N}(\mu_{cm}; \mu, \sigma)$$

## IV. Inferring Graspable Regions For 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 of contacts, 2) detecting graspability and grasp sensitivity, 3) decomposition into graspable regions, and 4) computing the gripper pose for grasping. In the following, we explain each step in detail.

*e) Obtaining candidate contact pairs:* The first step is to obtain 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 that can be fitted. Therefore, we consider those ellipses that can be best fitted to the surface of object in the area of contact points. The best fitted ellipse has the minimum area among the others. Furthermore, based on our grasping experiment, there are elevations at which the object cannot be grasped, for example close to or from below the table. We combine these two criteria in a score, and select the ellipse that maximizes the fitting score at an acceptable elevation:

$$e^* = \underset{e}{\operatorname{argmax}}\, \alpha p(\phi|g) + (1 - \alpha)\, \operatorname{fit}(e),$$

where $\phi$ indicates the elevation of the fitted plane, and $p(\phi|g)$ is the probability of an elevation for grasping, which is obtained from our grasping experiments. The weight $\alpha$ is set to 0.5 in our experiments.

*f) Predicting graspability and grasp sensitivity:* Given candidate pairs and fitted ellipses, we compute the features as described in Section III-B. We then exploit our trained classifier to filter out the graspable versus non-graspable contact pairs and ellipses. Next, we assign the graspable

| Feature Type | Principle Axes | Collinearity | Normal Distribution | Sensitivity | 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. THE FOUR CENTRAL COLUMNS SHOW THE PERFORMANCES OF THE INDIVIDUAL FEATURES (CF. SEC. III-B); THE RIGHT-HAND COLUMN SHOWS THEIR COMBINATION.

ellipses based on their sensitivity feature to the to the closest cluster from our trained clusters. The assigned cluster identifier determines the ellipse type.

*g) Grasp-based region formation:* We merge adjacent ellipses which belong to the same cluster into regions. The regions might have intersection with each other. This intersection might be due to the structure of the object and the grasps associated with it. As an example, a mug can be grasped from its handle, its body, or even the both simultaneously. Therefore, obtaining intersecting regions is not a deficiency in our method.

*h) Computing grasp parameter:* Provided with object regions and their grasp probabilities, we select the region with the highest grasp success probability. We the compute the grasp for the selected region. The grasps are associated with the ellipses composing each region. There may be many ellipses inside a region, all of which share the same grasp success probability. Nevertheless, ellipses belonging to the same region differ in their centrality within the region and thus their sensitivity. In particular, central ellipses within a region are likely to be less sensitive than peripheral ellipses. Therefore, we always choose the central ellipse of a region to compute the gripper pose, as mentioned in Section III-A.

## 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

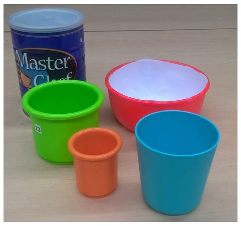


Fig. 4. Training objects for grasping experiments.

### 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.

### B. Generalization on Novel Objects

We performed robotic grasping experiment on novel objects which are composed of multiple parts and have complex shapes as show in Figure 5. We decomposed objects into regions based on grasp sensitivity as explained in Section IV. Next, we computed the grasp from the least sensitive region. The experiment is done three times per each object. We considered two measures for grasping performance: 1) grasp success rate on three time executions per object, and 2) the number of grasps explored for each object, the lower the better. Figure 6 shows grasping examples of the test objects. The quantitative results for our experiment is provided in Table II. As can be seen, our approach provided a very high grasp success rate, while exploring only small number of grasps per object. That is a computational boost for

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

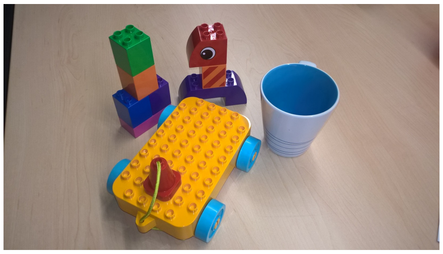


Fig. 5. Test objects for grasping experiments.

| Objects | Grasp Success Rate | Explored Grasps |
|---|---|---|
| Toy Giraffe | 100% | 1 |
| Mug | 100% | 2 |
| Toy Lego | 75% | 3 |
| Toy Car | 100% | 3 |

TABLE II

GRASPING PERFORMANCE ON NOVEL OBJECTS.

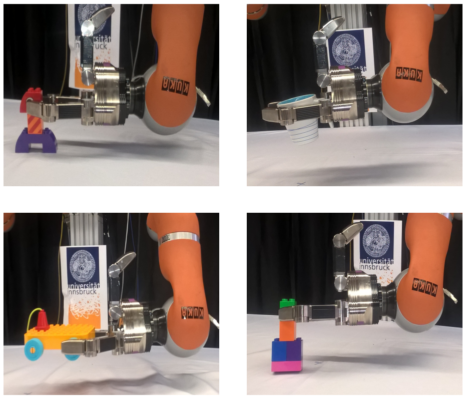


Fig. 6. Robotic grasping on novel objects with complex structure.

robotic grasping task and plays an important role in real-time grasping applications.

## VI. Conclusions

All in all, we proposed a method for representing objects based on graspability-derived Regions. 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.

Region-based grasp computation is computationally efficient 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.

## References

[1] A. Saxena, J. Driemeyer, and A. Ng, “Robotic grasping of novel objects using vision,” *The International Journal of Robotics Research*, vol. 27, no. 2, p. 157, 2008.

[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 3d images with markov random fields,” in *IEEE/RSJ International Conference on Intelligent Robot Systems (IROS)*, 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. Wörgötter, M. Schoeler, J. Papon, and T. Kulvicius, “Convexity based object partitioning for robot applications,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 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*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and D. Rus, Eds., vol. 39. Springer, 2006, pp. 33–42.

[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] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[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.