Generating Robust Grasps for Unknown Objects in Clutter Using Point Cloud Data

Hongzhuo Liang
Department of Informatics
University of Hamburg
liang@informatik.uni-hamburg.de

Shuang Li
Department of Informatics
University of Hamburg
sli@informatik.uni-hamburg.de

Michael Görner
Department of Informatics
University of Hamburg
goerner@informatik.uni-hamburg.de

Jianwei Zhang
Department of Informatics
University of Hamburg
zhang@informatik.uni-hamburg.de

ABSTRACT

For service robots in mundane environments, grasping unknown objects from clutter is an essential, yet complex, capability. In this work, we build on the recently popular *Grasp Pose Detector* system for generating grasp locations and propose multiple algorithmic improvements to achieve better performance with more complex gripper geometry. We validate the method in grasping experiments where objects are presented in isolation as well as in dense clutter, and demonstrate a 20% improvement in success rate of attempted grasps over the unmodified Grasp Pose Detector, achieving a rate of 91.1% in our setup. Additionally, our system can generate feasible grasp candidates that do not collide with the support surface in cases where the original method failed to grasp.

CCS Concepts

•**Computing methodologies** → *Robotic planning;*

Keywords

grasp pose detection; point clouds.

1. INTRODUCTION

The past decades have seen a lot of research that enables robots to grasp objects under controlled laboratory conditions as well as in mundane environments. In practice it remains difficult to generate stable grasp poses for unknown objects in cluttered environments. One common approach to implement this relies on a perception module to extract useful grasp poses from vision data. The resulting grasps are then used with traditional motion planning algorithms to pick up the object. Given a specific gripper geometry and models of graspable objects, one can define and approximate the grasp wrench space to judge whether or not a given candidate grasp will be stable [3]. Popular methods in recent years use neural networks to judge the quality of given grasps [5, 9]. However, in order to make use of such estimators for grasp stability, a major bottleneck is the generation of reasonable candidate grasps. Moreover the representation of the grasps for the estimator plays an important role. To get stable results with a tractable number of sampling attempts, most published approaches therefore restrict the set of possible grasp poses to a useful subset, for example top grasps [7, 8] or 2D variations around a low number of grasp points [10].

This paper describes several algorithmic contributions to enhance the generation and selection of efficient unrestricted grasps from point cloud data. Whereas a lot of previous work restricts grasps to lower-dimensional manifolds by reducing the degrees of freedom of the grasp description, we retain the full 6D pose description, but instead sample more grasp points in regions of the point cloud that are easier to reach by the manipulator. To get a collision-free grasp, we use a suitable selection of approach position. To explore more grasp poses in close vicinity we utilize a three-dimensional grid search and restrict suitable approach directions to avoid collisions with the support surface. Furthermore, in the final selection of the executed grasp we favor grasps that remain stable under small pertubations by considering the local neighborhood of a candidate in its quality value.

We evaluated these proposed methods in a series of experiments in the setup depicted in Figure 1. The results demonstrate that our improved algorithm provides significantly more successful candidate grasps and improves overall grasp performance.

2. RELATED WORK

In the past, analytical methods have been applied to analyze the geometry of the environment in sensor data and generate force closure grasps for giving friction coefficients [2, 6]. More research focuses on using data-driven methods to generate fast and robust grasps for unknown objects

HCR 2018 July 14–16, 2018, Shanghai, China

DOI:

(a) (b)

Figure 1: Experimental setup for grasping unknown objects. (a) Grasp in dense cluster. (b) Objects used in our experiment.

[1]. Saxena *et al.* were among the first to use a neural network to generate grasp candidates directly from vision where the network was trained on a synthetic corpus. Klingbeil *et al.* identify good grasps in point cloud data by looking for regions which resemble the interior of their gripper for given orientation and gripper spread. Recently, ten Pas *et al.* [10] proposed the grasp pose generation method (GPG) which samples a number of candidate grasps in a 3D point cloud and a grasp pose detection (GPD) method that assigns quality scores to each generated candidate through a convolutional neural network. Their proposed generator yields many useful, but also many infeasible candidate grasps. After including a number of technical refinements in their final grasp selection strategy, they achieve a grasp success rate of up to 93% in their setup. Note that this high grasp success rate is based on a multi-view point cloud from a mobile camera that is generated through computationally expensive SLAM methods.

3. GRASP DETECTION

The grasp detection approach [4, 10] proceeds by first sampling points in the input cloud and generating candidate grasps around them. The generated candidates are encoded and their quality is estimated by a previously trained CNN. Eventually, a heuristic method chooses a grasp from the top-scoring candidate set that is executed on the robot. The following subsections describe our optimizations to GPG/GPD in detail.

3.1 Data Preprocessing

Single acquired point cloud view from RGB-D cameras generally include a non-negligible level of noise and data outside our defined region of interest. Here we denoise the input clouds by voxelizing, remove outliers and clip the cloud around a defined region of interest. Notably, this last step removes the support surface of graspable objects from the input. We don't segment objects from these point clouds because we generate grasp candidates using geometry information rather than the pose and the shape of an object.

3.2 Candidate generation

In this subsection, we describe four modifications for the generation process that facilitate the generation of more high-quality candidates.

3.2.1 *Attentional sampling process*

As already noted in [10], in cluttered scenes top grasps show a higher success rate than side grasps. Also, grasps are not allowed to collide with the support surface. Whereas previous work exploited this information only during candidate selection, we modify the sampling process already. We divide the workspace $\mathbf{p}$ into three parts, $\mathbf{p_t}$, $\mathbf{p_m}$ and $\mathbf{p_b}$, where $\mathbf{p_b}$ denotes the point cloud from the bottom up to the gripper height of $\mathbf{p}$, $\mathbf{p_t}$ denotes a similar margin of the point cloud from the top of $\mathbf{p}$, and $\mathbf{p_m}$ denotes the remaining part of $\mathbf{p}$ excluding $\mathbf{p_b}$ and $\mathbf{p_t}$. If the height of $\mathbf{p}$ exceeds three times the gripper height, we uniformly sample 60% of all candidate points randomly from $\mathbf{p_t}$ and the rest from $\mathbf{p_m}$. Otherwise, we uniformly sample all points from $\mathbf{p_t} \cup \mathbf{p_m}$. Figure 2 illustrates the different parts of a point cloud and an example grasp pose F with the hand superimposed at the origin. As demonstrated in Figure 3, this process samples more candidates in the top region of the point cloud.

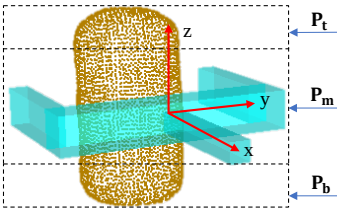


Figure 2: Grasp coordinate frame $F(i)$ in sample point i with hand superimposed at the origin. Different areas $\mathbf{p_t}$, $\mathbf{p_m}$ and $\mathbf{p_b}$ as defined in this work are indicated.

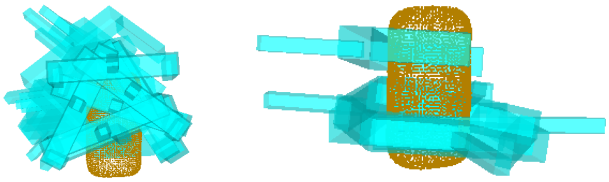
3.2.2 *Three-dimensional grid search*

For each point i that is sampled from the input cloud we compute a local reference frame $F(i)$. To improve the variance of generated grasps for this point, GPG performs a two-dimensional grid search along the y axis and around z. To cover even more local variations, we extend this grid search to three dimensions and also include rotations around the y axis. Figure 3 illustrates positive grasp candidates generated for 5 sample points by the two methods. As can be seen, the extended grid search in 3(a) yields a higher number of grasp candidates which are better distributed over the graspable object surface. Even so, it also generates a number of infeasible grasps that have to be taken into account later on. Some of these will be corrected in the next step and some remain to support the neighbor-based selection described below.

3.2.3 *Projection of invalid candidates*

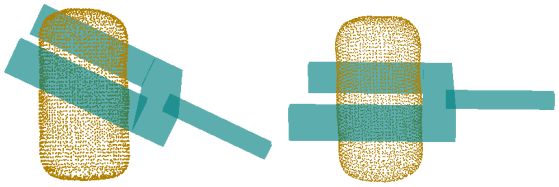
One remaining issue with the generated candidates is that while they respect possible collisions of the fingers and the environment, they do not always allow for feasible gripper approaches. In particular, the system generates grasps that approach objects from below, enclosing an angle of less than 90◦ between the approach direction and the normal of the support surface. In these cases we simply project the infeasible candidates to align their approach direction to the support surface. This is illustrated in Figure 4.

3.2.4 *Conservative approach depth*



(a) (b)

Figure 3: Candidate grasps generated by different methods. (a) Grasps sampled by attentional sampling and 3D grid search method. (b) Grasps sampled by uniform sampling and 2D grid search method.



(a) (b)

Figure 4: Example of invalid candidate projection. (a) A grasp candidate generated by original GPG. (b) The candidate projected to support surface.

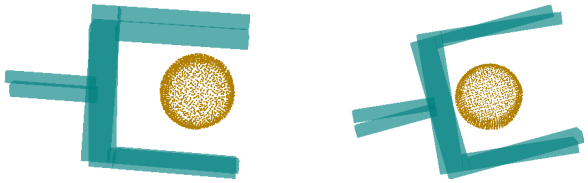
To fully define the sampled grasp, the last missing parameter specifies how far the respective grasp point should penetrate the volume inside the gripper. In original GPG, the gripper is pushed forward until it would collide with the perceived point cloud in order to generate a maximally stable grasp that encloses as much as possible of the object with the gripper. While this strategy can generate very secure grasps, this mostly applies to two-finger parallel grippers with limited finger length. When the length of the fingers exceeds the typical size of individual objects in a scenario, the fingers will often collide with unobserved parts of the scene and more often pick multiple objects in unstable grasps. Additionally, most adaptive grippers focus on a stable grasp volume between the fingertips, but not necessarily within the whole volume inside the gripper. This is the case for the 3-finger adaptive Robotiq gripper visible in Figure 1(a) when used for precision grasps.

In order to avoid these problems in a general way, we compute the approach depth by pushing the gripper further forward either until no collision with the perceived point cloud or until no further points are added to the volume inside the gripper. Figure 5 contrasts both criteria. Obviously, the modified strategy in Figure 5(a) is less aggressive and keeps the fingers of the gripper further away from possibly unseen obstacles.

3.3 Neighbor-based selection strategy

As previously discussed, good heuristics for the creation of the set of candidates are essential, because the final grasp is selected based on them. To rank these candidates, we apply the same strategy as GPD: candidates are represented by 15-channel images and assigned scores $\mathcal{S}_{cnn}$ by a pre-trained four-layer convolutional neural network.

To further improve the quality of this ranking, we notice



(a) (b)

Figure 5: Variations of approach depths: (a) Conservative strategy and (b) GPG's greedy strategy.

that robust grasps should remain good under minor local deviations. To account for this, we propose a neighbor-based extension to the overall scoring. If the single best pose is close to a region of low-scoring grasps, that pose should be penalized, whereas poses which are surrounded by other high-scoring grasps should be preferred.

We choose the final grasp as follows. First, we cluster the top N_s highest-scoring grasps based on distance and orientation. Candidates that are close to each other and roughly aligned in orientation are grouped together. For each cluster with a size below some n_c, we generate additional candidates nearby that are also scored and become part of the cluster. For each cluster, we also add the mean grasp pose as an additional grasp candidate. In contrast to the original GPD method, we retain the original set of candidates together with these average grasps to retain local variance. We define the average score of a cluster as $\mathcal{S}_k$.

To apply neighbor-based scoring, we define the final scoring function of a grasp $\mathcal{S}(g)$ as the weighted sum of its individual score $\mathcal{S}_{cnn}$ and its cluster's average score $\mathcal{S}_k$, also shown in Equation 1.

$$\mathcal{S}(g) = \alpha \mathcal{S}_{cnn}(g) + (1-\alpha)\mathcal{S}_k(g) \tag{1}$$

In all experiments, we used a value of $\alpha = 0.7$. Eventually, the grasp with the highest score $\mathcal{S}(g)$ will be selected.

4. EXPERIMENTS

We validated the efficiency and flexibility of our algorithm in two experimental conditions: Objects were presented to the robot in isolation as well as in a cluttered scenario. The experiments were carried out on a UR5 robotic arm with an attached Robotiq 3-finger adaptive robot gripper. We manipulated 8 different, previously unseen toys between 3cm and 10cm. The robotic setup and the objects can be seen in Figure 1. Input point cloud data is captured by a Kinect2 RGB-D camera from one fixed view. The whole system is implemented using the ROS framework and was tested on an Intel i9-7900X 3.30 GHz system with 128GB RAM. Inverse kinematics uses an analytical solver based on IKFast and Motion planning was implemented via the MoveIt! library.

4.1 Objects Presented in Isolation

We ran 10 trials for each object presented in isolation on the table. The results shown in Table 1 demonstrate that our improvements result in more grasp attempts and a higher success rate. Note that the number of attempts of some low objects (in this dataset the potato and the banana) is zero. In these cases we let the algorithm run for 5 minutes but no actionable grasp was generated. This is mainly because all

Table 1: Grasp test on isolated object

Objects to pick		Ball	Potato	Frog	Sponge	Onion	Banana	Melon	Bear	Total
GPD	Success/Attempts	8/10	0/0	7/10	7/10	10/10	0/0	5/10	9/10	46/60
	Success rate of grasp attempts	80%	Null	70%	70%	100%	Null	50%	90%	76.7%
Improved GPD	Success/Attempts	10/10	10/10	8/10	10/10	10/10	7/10	9/10	8/10	72/80
	Success rate of grasp attempts	100%	100%	80%	100%	100%	70%	90%	80%	90%

Table 2: Grasp test on cluster objects

GPD	Success/Attempts	32/45
	Num grasped/total objects	44/80
	Success rate of attempted grasps	71.1%
	Completion rate	55%
Improved GPD	Success/Attempts	51/56
	Num grasped/total objects	63/80
	Success rate of attempted grasps	91.1%
	Completion rate	78.8%

generated grasp candidates would collide with the table below the object. In contrast the conservative approach depth avoids these collisions and manages to grasp the objects.

4.2 Objects Presented in Dense Clutter

In the dense clutter condition we randomly put all objects in a heap and let the system pick them successively. We ran 10 trials in this task and the results can be found in Table 2. The improved GPD method achieved a success rate of 91.1% for attempted grasps and picked 78.8% of all objects. For both metrics, this clearly outperforms the original GPD implementation by at least 20%.

Note that the number of successes differs from the number of grasped objects in both cases. In some cases the gripper grasps more than one object at once, but as there is no concept of an object in the entire grasp generation process, this is an expected phenomenon.

5. CONCLUSIONS

This work proposed an improved grasp detection algorithm for the task of robust grasp planning for unknown objects. We sample grasp points via an attentional process in order to generate more grasp candidates in the graspable region of cluttered space. Searching grasp candidates by 3D grid search significantly boost the number of grasp candidates and a conservative estimation of the grasp depth effectively avoids collisions of the gripper with obstacles. Moreover, the projection of the approach direction of infeasible grasps also improves the quality of grasp candidates. Finally, a neighbor-based ranking strategy is used and supports the selection of robust grasps.

The experimental results demonstrate that our improvements can increase the number of attempted grasps, as well as achieve higher success rates for attempts and completion rate for pick tasks. Overall the improvements generate more robust and reliable grasp candidates.

Even so, it leaves many alternative pathways untouched. While sampling of candidates allows for more intuitive internal dynamics, it also requires a lot of computation. Creating 3D occupancy grid maps will be considered in future. An alternative improvement could update and verify grasp scores during grasp attempts to abort as soon as mistakes in the quality estimate become apparent.

6. ACKNOWLEDGMENTS

This research was funded by the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, TRR-169. And it was partially supported by project STEP2DYNA (691154).

7. REFERENCES

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-Driven Grasp Synthesis-A Survey. *IEEE Transactions on Robotics*, 30(2):289–309, apr 2014.

[2] C. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. In *International Conference on Intelligent Robots and Systems*, volume 4, pages 3692–3697. IEEE, 2003.

[3] C. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 319–325. IEEE, 2004.

[4] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt. High precision grasp pose detection in dense clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2016-Novem, pages 598–605. IEEE, oct 2016.

[5] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib. Grasping with application to an autonomous checkout robot. In *2011 IEEE International Conference on Robotics and Automation*, pages 2837–2844. IEEE, may 2011.

[6] M. Li, K. Hang, D. Kragic, and A. Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364, 2016.

[7] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. *Robotics: Science and Systems (RSS)*, 37(3):301–316, mar 2017.

[8] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3406–3413. IEEE, may 2016.

[9] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173, 2008.

[10] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt. Grasp Pose Detection in Point Clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, dec 2017.