Lundell, Jens; Verdoja, Francesco; Kyrki, Ville

# Beyond Top-Grasps Through Scene Completion

*Published in:*
Proceedings of the IEEE Conference on Robotics and Automation, ICRA 2020

*DOI:*
10.1109/ICRA40945.2020.9197320

Published: 01/01/2020

*Document Version*
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*
Lundell, J., Verdoja, F., & Kyrki, V. (2020). Beyond Top-Grasps Through Scene Completion. In *Proceedings of the IEEE Conference on Robotics and Automation, ICRA 2020* (pp. 545-551). Article 9197320 (IEEE International Conference on Robotics and Automation). IEEE. https://doi.org/10.1109/ICRA40945.2020.9197320

# Beyond Top-Grasps Through Scene Completion

Jens Lundell, Francesco Verdoja and Ville Kyrki

***Abstract*— Current end-to-end grasp planning methods propose grasps in the order of seconds that attain high grasp success rates on a diverse set of objects, but often by constraining the workspace to top-grasps. In this work, we present a method that allows end-to-end top-grasp planning methods to generate full six-degree-of-freedom grasps using a single RGB-D view as input. This is achieved by estimating the complete shape of the object to be grasped, then simulating different viewpoints of the object, passing the simulated viewpoints to an end-to-end grasp generation method, and finally executing the overall best grasp. The method was experimentally validated on a *Franka Emika Panda* by comparing 429 grasps generated by the state-of-the-art Fully Convolutional Grasp Quality CNN, both on simulated and real camera images. The results show statistically significant improvements in terms of grasp success rate when using simulated images over real camera images, especially when the real camera viewpoint is angled.**

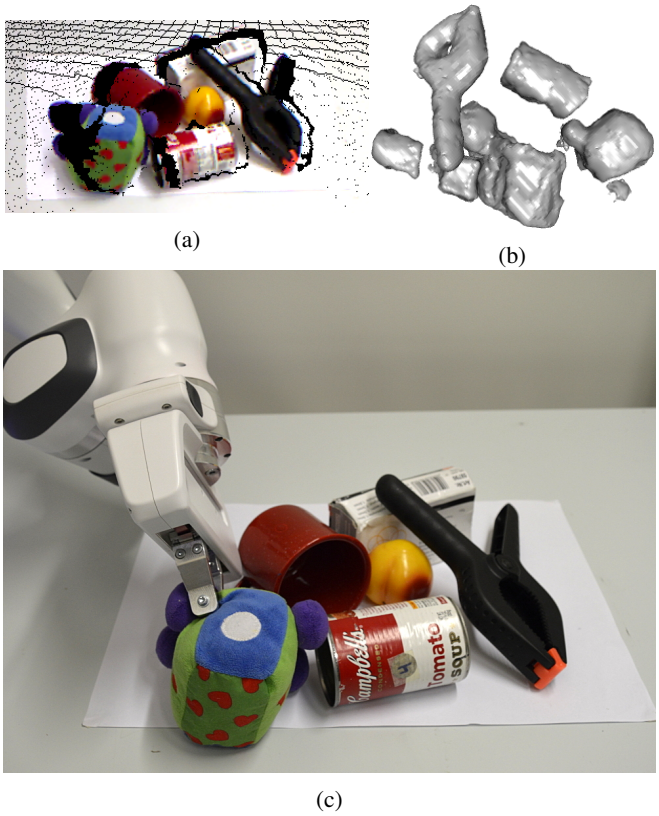
## I. Introduction

Robotic grasping has undergone a paradigm shift from analytical methods toward data-driven ones. Deep learning is the major driving force behind the shift and has given rise to a diverse set of methods [1]–[8]. These methods typically reach high grasp success rates (often above 90%) on a wide variety of objects while keeping the total computation time in the order of seconds, surpassing analytical methods by a large margin. However, to reach such a performance the grasp planning problem is usually constrained to the generation of top-grasps with four degrees-of-freedom (dof): one orientation and three translations. Top-grasps are good if the camera perceiving the environment is perpendicular to the plane supporting the target. However, as shown in this work, once the camera views a scene from an angle the performance drops. In such situations the grasping methods need to propose grasps in full six dof space to allow a robot to approach objects from any possible direction.

One viable option to achieve full 6 dof grasping with current state-of-the-art grasping methods is to mount a camera on the robot itself and have it scan the scene from multiple viewpoints. However, not only is such an approach slow as the robot needs to first plan where to move and then physically move there but also the robot might self-occlude the view of the camera, rendering the method useless. A novel alternative, which is studied in this work, is to simulate different viewpoints of the object to be grasped and feed these to the methods proposing 4 dof grasps. As shown in Fig. 1, such a solution enables a robot to grasp objects from directions different from the one of the real camera.

This work was supported by the Strategic Research Council at Academy of Finland, decision 314180.

J. Lundell, F. Verdoja and V. Kyrki are with School of Electrical Engineering, Aalto University, Finland. {firstname.lastname}@aalto.fi



(a) (b) (c)

Fig. 1: (a) shows a point-cloud of a cluttered scene acquired with a real camera while (b) shows a simulated top-view of the same scene but shape completed. (c) shows that with our method, successful grasps can be generated from approach directions different from the camera viewpoint.

TO this end, we present a grasping pipeline that uses the state-of-the-art Fully Convolutional Grasp Quality CNN (FC-GQ-CNN) [2] to propose grasps. The pipeline first segments a point-cloud of the scene into objects. Then the shape of each object is estimated and placed in a physics simulator. In the context of this work, we refer to this as *scene completion*. Inside the physics simulator, a set of depth images are sampled from different viewpoints and fed to the grasp proposal method generating a set of grasp candidates. The grasp candidate with the highest score is then executed on the robot.

The proposed grasping pipeline is experimentally validated on a *Franka Emika Panda* by benchmarking it against grasps proposed on real depth images on single object grasping and grasping in clutter. The results of 429 grasps on single object grasping show statistically significant improvement in terms

of higher grasp success rate when planning is performed on simulated depth images compared to planning solely on real camera images. Similar results were also evident when grasping in cluttered scenes.

The main contributions of this paper are: (i) a novel grasp planning pipeline that enables existing 4 dof end-to-end methods to propose full 6 dof grasps, (ii) a method to densely sample simulated depth images, and (iii) an empirical evaluation of the proposed method against state-of-the-art on real hardware, presenting a statistically significant improvement in terms of higher grasp success rate using the proposed method.

## II. Related Works

To date, many grasping methods rely fully or in part on deep learning. Some methods only use deep learning to extract additional information about objects with *e.g., shape completion* [9], [10] or tactile information [11] and then use analytical methods to plan the actual grasp [12], while others employ *data-driven grasp planning* in an end-to-end fashion to generate grasps directly from images [1]–[8]. We will review both shape completion and end-to-end data-driven grasp planning as both are vital parts of our grasping pipeline.

### A. Deep Shape Completion

In the context of shape completion from incomplete point-clouds, most recent improvements come from the adoption of deep learning. For instance, different works have explored tailored network structures [9], [13], [14], semantic object classification to aid the reconstruction [15], the integration of other sensing modalities such as tactile information [11], or the exploitation of the network uncertainty [10].

In the context of robotics grasping, [9], [11] and [10] are the most interesting as they not only focus on shape reconstruction quality but also on grasping accuracy. In this work, we make use of our previous shape completion network [10] to complete objects but instead of planning grasps with analytical methods—which is computationally expensive—we turn to data-driven grasp planning.

### B. End-To-End Data-Driven Grasp Planning

The general interest in end-to-end data-driven grasp planning came after the pioneering work by Saxena *et al.* [16] where they trained a logistic regression model to directly predict good grasping points from a monocular image. To train the logistic regressor they used a large amount of synthetically labeled images of objects and the corresponding grasping location.

The use of synthetic data to train the sensor-to-grasp map was later used in a wide variety of similar methods [1]–[8]. For instance, Mahler *et al.* [1] used a data-set containing millions of synthetic antipodal top-grasps on a wide variety of objects to train a Grasp Quality CNN (GQCNN) that generates a grasp from a depth image in the order of seconds. The GQCNN was later improved in [2] through the use of on-policy data and a fully convolutional network structure called FC-GQ-CNN. The state-of-the-art FC-GQ-CNN was faster than GQCNN while sampling about 5000x more grasps and was thus used to generate grasps in this work.

Another line of research in end-to-end data-driven grasp planning is Reinforcemnt Learning (RL) [17]–[20] where the goal is to learn the sensor-to-grasp map directly through trial and error on the robot. Models learned with RL can attain a high grasp success rate without any hand-labeled data-sets, but the extensive interaction time needed to learn the model, which can be months on physical robots [19], is a bottleneck. Although some work have reduced the real-world interaction by using simulation [20], the learned models still needs fine-tuning on physical hardware to reach similar grasp success rate as methods that uses supervision [1], [2].

A major limitation in most end-to-end data-driven grasp planning works is that the planned grasp is only from the viewpoint of the camera, which effectively constrains the grasps to a subset of the complete 6 dof workspace (typically 4 dof grasps are considered). The work presented here lifts this limitation by shape completing the real objects, placing them into a physics simulator, and from there sampling different viewpoints of the object. Our method hence enables standard end-to-end data-driven grasp planning methods that suggest grasps from only one camera viewpoint to generate full 6 dof grasps from other directions such as the back of the object.

## III. Problem formulation

In this work, we address the problem of grasping unknown objects lying on a supporting surface with a robotic arm equipped with a parallel-jaw gripper. Information about the scene is obtained by a RGB-D camera whose pose is arbitrary but known relative to the robot.

Formally, let $\mathbf{x} = (\mathbf{c}, \mathbf{O})$ denote a state representing the environment, where $\mathbf{c} \in \mathbb{R}^6$ is the camera pose, and the set $\mathbf{O} = \{(\mathcal{O}_i, \mathbf{p}_i)\}_{i=0}^{N}$ contains the properties of the $N$ objects to be grasped, described by their pose $\mathbf{p} \in \mathbb{R}^6$ and model $\mathcal{O}$. The state is partially observable as $\mathbf{O}$ can only be indirectly and incompletely observed using the RGB-D camera. The camera produces a 2.5D point-cloud $\mathbf{y} = \{y_i\}_{i=1}^{H \times W} \in \mathbb{R}_{+}^{H \times W}$ which can be represented as a $H \times W$ depth image, assuming known camera intrinsic parameters.

Let $\mathbf{g} \in \mathbb{R}^6$ denote a parallel-jaw grasp, described by the 6D pose of the gripper center point and $S(\mathbf{g}, \mathbf{x}) \in \{0, 1\}$ be a binary-valued grasp success metric indicating, *e.g.,* force closure. Assuming a joint distribution $p(S, \mathbf{g}, \mathbf{x}, \mathbf{y})$, let $Q(\mathbf{g}, \mathbf{y}) = \mathbb{E}[S \mid \mathbf{g}, \mathbf{y}]$ be the expected value of the metric given a grasp $\mathbf{g}$ and a point-cloud $\mathbf{y}$. The quality $Q$ is intractable in most real cases. Therefore, it is modeled in data-driven grasp planning methods with a learned parametric model $Q_\theta$ with parameters $\theta$. The parametric model $Q_\theta$ is typically optimized either with supervised learning on synthetic [1] or real grasping data [4], or with RL [19].

Then, given a point-cloud $\mathbf{y}$ obtained from a known camera pose $\mathbf{c}$, the goal of most data-driven grasp planning

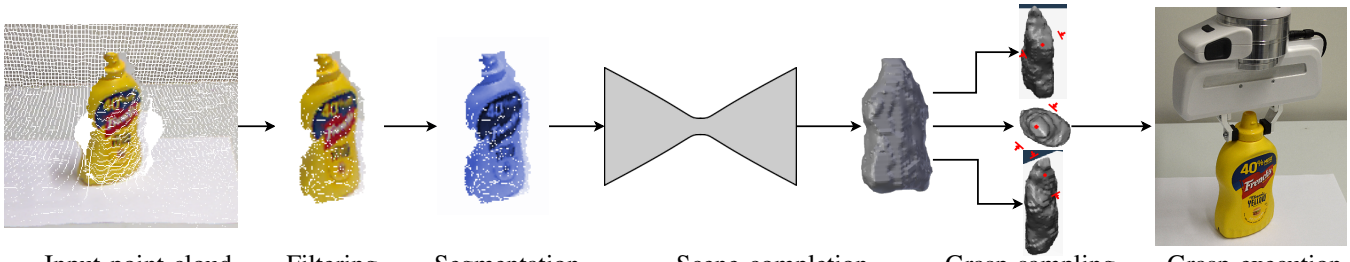



Fig. 2: The proposed grasping pipeline

methods is to find a grasp $\mathbf{g}^*$ such that:

$$\mathbf{g}^* = \arg\max_{\mathbf{g} \in \mathcal{G}} Q_\theta(\mathbf{g}, \mathbf{y}), \tag{1}$$

where $\mathcal{G}$ is a set of grasp candidates. However, such a formulation only accounts for grasps approaching the scene from the same direction as the camera, which is usually looking at it from top, leading to only top-grasps being proposed.

Instead, we propose to extend this framework to allow object grasping from any direction, even those not directly seen by the camera. For this, we need a function $\hat{\mathbf{x}} = C(\mathbf{y})$ as an estimate of the full state $\mathbf{x}$ from the point-cloud $\mathbf{y}$. Practically, this means understanding how many and what objects are in the scene (*i.e.,* segmentation), and then for each object proposing a model $\mathcal{O}$ and a pose $\mathbf{p}$ (*i.e.,* shape completion). Once a state estimate $\hat{\mathbf{x}}$ is available we want to evaluate the quality of grasps approaching from any direction and execute the first kinetically feasible one with highest quality. Our proposed pipeline to achieve this is described next.

## IV. Grasping Pipeline

The grasping pipeline shown in Fig. 2 consist of: (i) filtering and segmenting the real objects, (ii) shape completing each segment and add them to a physics simulator, (iii) generate and rank grasps from different viewpoints of the objects, (iv) execute the best ranked grasp on the real robot.

### A. Segmentation

The scene in the point-cloud contains an unknown number of objects lying on a table. We first remove points that are part of the background indicated by their distance exceeding a threshold, as well as points that belong to the supporting surface identified using the known pose of the camera with respect to the robot. We are then left with a filtered point-cloud $\bar{\mathbf{y}}$ containing only the view of the objects to be segmented.

Given the point-cloud $\bar{\mathbf{y}}$, we define an $N$-region segmentation as a partition $\mathbf{R} = \{\mathbf{r}_i\}_{i=1}^N$ of the points of $\bar{\mathbf{y}}$. More precisely, the regions must satisfy the following constraints:

$$\begin{aligned} &\forall y \in \bar{\mathbf{y}} \quad (\exists \mathbf{r} \in \mathbf{R} \mid y \in \mathbf{r})\,; \\ &\forall \mathbf{r} \in \mathbf{R} \quad (\nexists \mathbf{r}' \in \mathbf{R} \setminus \{\mathbf{r}\} \mid \mathbf{r} \cap \mathbf{r}' \neq \emptyset)\,. \end{aligned} \tag{2}$$

These constrains enforce that all points in the point-cloud $\bar{\mathbf{y}}$ have to belong in a region but no point can belong in two regions.

The pipeline is agnostic to the segmentation algorithm employed, but the assumption is that, after the segmentation step, each region in $\mathbf{R}$ contains points belonging to a different object to be grasped. The next step is to estimate each object's properties through scene completion.

### B. Scene Completion

Scene completion refers to the process of both shape completing each object in the scene and then placing them in a physics simulator according to their individual estimated pose. Shape completion refers to reconstructing the shape of an object from partial information about it in the form of a point-cloud. More precisely, a shape completion algorithm estimates $(\mathcal{O}, \mathbf{p})$ given a point-cloud $\mathbf{r}$. To shape complete objects we used the pre-trained fully convolutional hour-glass shaped Deep Neural Network (DNN) proposed in [10] whose input is a voxel grid of the point-cloud captured from the camera and output is a completed voxel grid. The completed voxel grid is post-processed into a mesh by merging it with the input point-cloud and running the marching cube algorithm [21].

The DNN in [10] also included dropout layers throughout that were active during run-time to generate a set of shape samples representing, through Monte Carlo sampling, the shape uncertainty. In this work, we also generate shape samples but average them together to get a mean shape, effectively ignoring the shape uncertainty. Although it would be possible to deactivate the dropout layers at run-time and only consider a point estimate of the shape, the benefit of using the mean shape is that it is smoother, removing sharp artefacts on the shape which many end-to-end data-driven grasp planning methods often rank as stable grasp points.

For each region $\mathbf{r}_i \in \mathbf{R}$ we generate, through shape completion, objects $(\mathcal{O}_i, \mathbf{p}_i)$. Together, all objects represents an estimate $\hat{\mathbf{x}}$ of the real environment state $\mathbf{x}$. The state estimate $\hat{\mathbf{x}}$, containing all objects represented as meshes, are subsequently placed in a physics simulator. The next step is then to sample grasps over the state estimate.

### C. Grasp Sampling

To obtain grasp candidates from all directions, we populate a scene in a physic simulator according to the state estimate $\hat{\mathbf{x}}$. Given the populated scene, we render $n$ depth images $\mathbf{Y} = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ of the objects from different viewpoints. To densely sample the scene we propose the sampling scheme visualized in Fig. 3, which is to approximate a sphere around

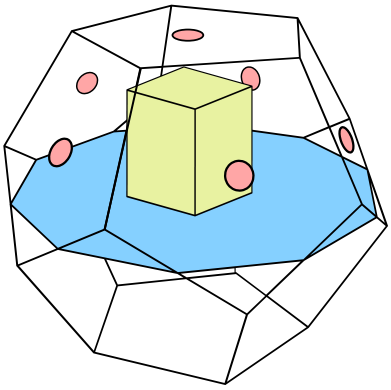


Fig. 3: The proposed dodecahedron sampling scheme. The object in yellow is lying on the blue plane. The sampled viewpoints (represented as red circles) are the midpoints of each face in the upper-half of the dodecahedron (best viewed in color).

the workspace with a dodecahedron and use the midpoints of each face in the upper half as the viewpoints. This amounts to $n = 6$ viewpoints in total and includes the top-view of the object that most end-to-end data-driven grasp planning methods are trained on. Of course, other sampling strategies can be devised to obtain an higher or lower number of viewpoints, as desired.

Next, we add noise to each simulated depth image $\mathbf{y}_i$ to make them more similar to ones acquired from physical cameras. The reason for adding noise is because many end-to-end data-driven grasp planning methods [1], [2], [5] use synthetic data to train $Q_\theta$ and adding artificial noise mimics depth images acquired from physical cameras which, in turn, improves the sim-to-real transfer. Similar to [1], we added both multiplicative and additive noise to each viewpoint resulting in the noisy depth image $\hat{\mathbf{y}}_i = \alpha \mathbf{y}_i + \epsilon$, where $\alpha \sim \Gamma(k, s)$ is a Gamma random variable modeling depth-proportional noise, and $\epsilon$ is a pixel-wise zero-mean Gaussian noise as explained in [22] with bandwidth $l$ and measurement variance $\sigma$ modeling additive noise. Experiments verified that adding noise to the depth images made the grasps more robust.

Grasps $\mathcal{G}_i$ are then generated on the set of noisy depth images $\hat{\mathbf{y}}_i$. The grasp $\mathbf{g}^*$ that achieves the highest utility among all candidates from all viewpoints is considered the best and, if physically reachable, is executed on the real robot. Formally, the best grasp is

$$\mathbf{g}^* = \underset{\mathbf{g} \in \mathcal{G}_i, i=1,\ldots,n}{\arg\max} \; Q_\theta(\mathbf{g}, \hat{\mathbf{y}}_i). \tag{3}$$

## V. Experiments

The two main questions we wanted to answer in the experiments were:

1) What is the impact of generating grasps from simulated depth images as opposed to real ones on grasp success and object clearance rate?

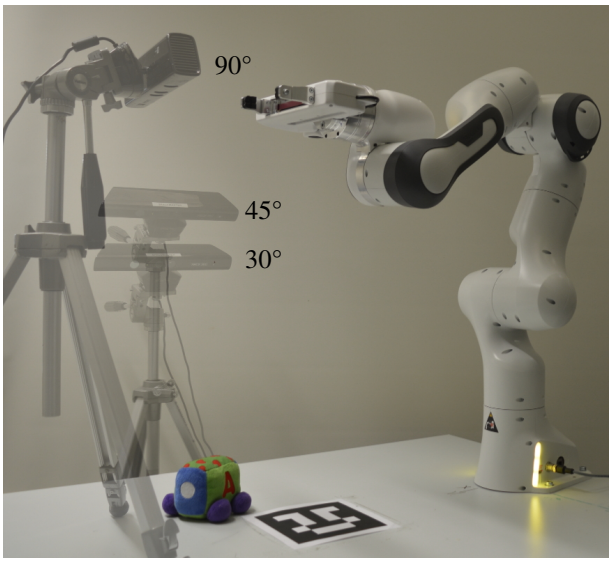



Fig. 4: The three camera viewpoints for single object grasping.

2) Is it beneficial to simulate angled viewpoints instead of top-views only?

In order to provide justified answers to these questions, we conducted two separate experiments. The first experiment evaluates grasp success rate on single object grasping while the second one evaluates the clearance rate in cluttered scenes.

### A. Experimental Setup

To perform the experiments we used the *Franka Emika Panda* robot and a Kinect 360° camera to capture the input point-clouds as shown in Fig. 4. We used an Aruco marker [23] for the extrinsic calibration of the camera. Once a point-cloud was captured, it was segmented, shape completed and finally placed into a physical rendering of the scene with the same transformation as in the real world. For segmentation we used the region growing method in PCL and for physical rendering MuJoCo [24]. For the zero-mean Gaussian noise $\epsilon$ we set $\sigma = 0.001$ and kernel bandwidth $l = 6$. For the depth-proportional noise $\alpha$, modeled as a Gamma distribution, we set $k = 5000$, $s = 0.0002$.

In both experiments we tested three different methods all using a pre-trained FC-GQ-CNN which is trained to recognize stable top-grasps from depth images [2]. The first method, which is the baseline, generates grasps with the FC-GQ-CNN on real depth images captured from a Kinect 360° camera. We benchmarked this against our method with two different sampling schemes for simulating depth images: one used the complete dodecahedron sampling method described in Section IV-C while the other sampled a depth image from a top-down view only. Henceforth we refer to the baseline method as FC-GQ-CNN, ours with the dodecahedron sampling as Simulated All-Grasps (SAG), and our with top-down sampling as Simulated Top-Grasps (STG).


Fig. 5: The 13 individually numbered objects used in the experiment. All objects, except 6 and 13, are from the YCB object set.

### B. Single Object Grasping

For single object grasping we compared FC-GQ-CNN with and without simulated depth images on the 13 objects shown in Fig. 5. To study the effect of the camera viewing angle on grasp performance, we ran the experiments on three different angles towards the grasping plane (30°, 45° and 90°) all shown in Fig. 4. For the 30° and 45° viewing angles the objects were placed in five different orientations: 0°, 72°, 144°, 216° and 288°, while for the 90° viewing angle, which corresponds to a top-down view, we only placed the objects at a 0° orientation. In total this setup amounts to 143 grasps per method.

To evaluate if a grasp was successful, the robot moved to the planned grasp pose, closed its fingers, and moved the arm upward 20 cm. Then, the arm moved back to the starting position, and once there rotated the hand $\pm 90°$ around the last joint. A grasp was successful if the object was within the gripper for this whole procedure and unsuccessful if dropped.

The experimental results for the different methods, which are analyzed for statistical difference with a one sided Wilcoxon signed-rank test, are presented in Table I. Over all viewing angles, the average grasp success rate is higher with the proposed STG and SAG compared to the baseline FC-GQ-CNN ($p < 0.0001$, $p < .05$). This result stems from the fact that the performance of FC-GQ-CNN deteriorates heavily when moving from a top-down view to an angled view. For instance, the relative performance drop for FC-GQ-CNN from a 90° viewing angle to a 45° is -42.22% and to 30° the drop is -28.89%. This is much higher compared to the performance drop for SAG, which is only -5% and -10%. The performance drop for STG is even less with -4.4% and -2.2% when moving from a 90° viewing angle to a 45° and 30° respectively. Together, these results show the importance of simulating depth images if the viewing angle of the real camera is not 90°.

Another interesting result from Table I is that STG, which simulates only top-down views, outperforms SAG which, in addition to simulating a top-down view, also simulates from angled viewpoints. One reason STG achieved a higher grasp success rate than SAG was that in many cases when an angled grasp was executed the gripper either tilted the object over or if the gripper decided to grasp a corner of an object the object simply slipped out of the gripper. Such situations were not common for top-grasps as the surface on which the object lies prevents the object from slipping and reduces the chance of it falling over. Although top-grasps seem more robust to external perturbations, we hypothesize that the performance difference between STG and SAG could be reduced if FC-GQ-CNN was also trained on angled viewpoints.

Finally, Fig. 6 shows clearly that STG performs better than average on all objects except for object 3 while SAG is above average on 7 out of the 13 objects. The performance of FC-GQ-CNN, on the other hand, is worse than average on 10 objects with an over 20% worse than average performance on objects 2, 3, and 12. One possible reason FC-GQ-CNN performs poorly on those objects is that grasping them from an angled viewpoint is much harder than grasping them from the top.

### C. Grasping in Clutter

In this experiment we studied the clearance rate of each method in a cluttered scene, meaning that the objective was to remove as many objects as possible within a given grasping budget. The grasping budget was set to 12 grasps and the objects we chose to use were 4, 6, 7, 8, 10 and 12 in Fig. 5 as these represent different shapes and sizes. To generate a cluttered scene, the objects were placed in a box that was shaken and emptied onto a table. An example scene is shown in Fig. 1. The physical camera perceiving the scene was set to 45°. To evaluate if an object was successfully removed from the scene we used the same procedure as in the single object grasping experiments except the last step to rotate the gripper was excluded for speed.

The experimental results are presented in Table II. These results show a clear improvement in average clearance rate using STG and SAG over FC-GQ-CNN. For instance, STG removed all objects in 9 out of 10 scenes and for the one scene it did not clear only one object was left. SAG, on the other hand, managed to clear 5 out of 10 scenes while FC-GQ-CNN cleared 2 out of 10 scenes.

For scenes where not all objects were cleared, the average clearance rate were 83.33% for STG, 76.68% for SAG, and 47.9% for FC-GQ-CNN. In these cases, FC-GQ-CNN manged to remove more than half of the objects in only 2 out of the 8 scenes. SAG and STG, on the other hand, removed more than half of the objects in all scenes.

Based on the presented results, we demonstrated that it is also beneficial to generate grasps from other viewing angles when removing objects in cluttered scenes. Together, both the result on single object grasping and grasping in clutter demonstrate that the performance of FC-GQ-CNN, a state-of-the-art end-to-end data-driven 4 dof grasp planning method, deteriorates heavily when viewing the scene from an angled viewpoint. However, through the use of shape completion and simulated viewpoints this is no longer the case.

## VI. Conclusions

We presented a grasping pipeline that enables end-to-end data-driven grasp planning methods which previously

TABLE I: Average grasp success rate on different viewing angles with test statistics and p-values of pair-wise one sided Wilcoxon signed-rank test between methods.

| Viewing angle | FC-GQ-CNN | STG | SAG | FC-GQ-CNN vs STG | FC-GQ-CNN vs SAG | SAG vs STG |
|---|---|---|---|---|---|---|
| 90° (top-down) | **69.23%** | **69.23%** | 61.54% | – | – | – |
| 45° | 40.00% | **66.15%** | 58.46% | T=52, p<.0005*** | T=144, p<.05* | – |
| 30° | 49.23% | **67.69%** | 55.38% | T=144, p<.05* | – | – |
| Average success rate | 46.85% | **67.13%** | 57.34% | T=450, p<.0001*** | T=768, p<.05* | – |
| Planning Time (s) | 2.3 | 2.0 | 12.3 | | | |
| Shape Completion Time (s) | – | 27.6 | 27.6 | | | |

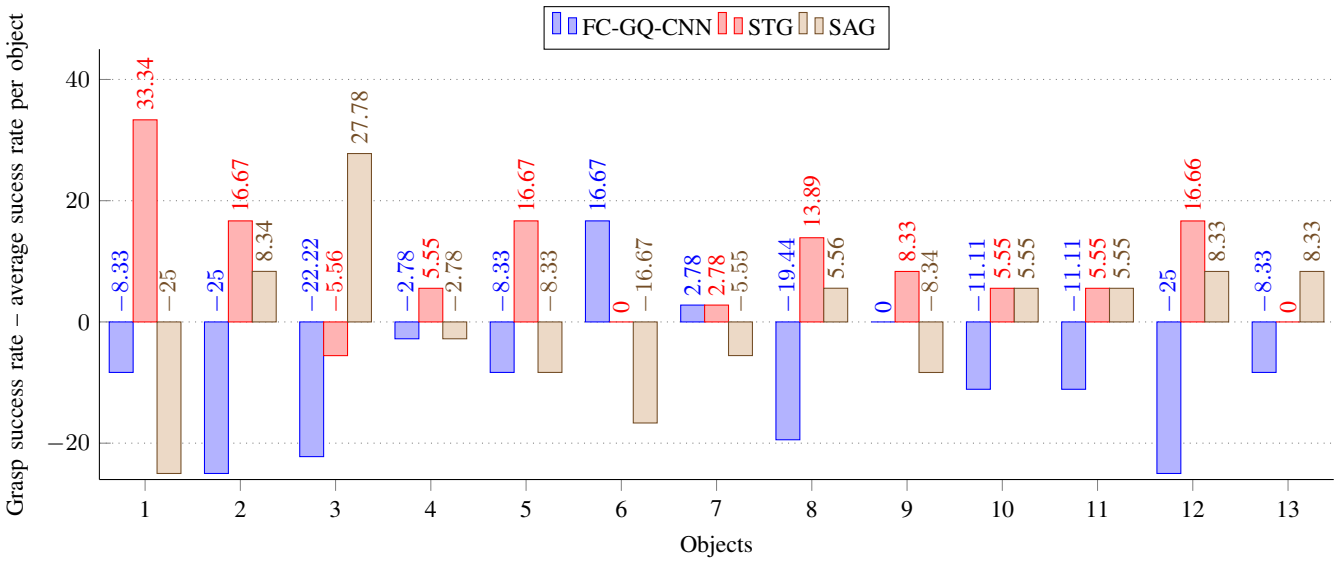



Fig. 6: Grasp success rate per object for each method minus the average success rate per object on each of the 13 objects used in the experiment.

TABLE II: Results on the cluttered scene

| | FC-GQ-CNN | STG | SAG |
|---|---|---|---|
| Average clearance rate (%) | 58.33 | **98.33** | 88.33 |
| Planning Time (s) | **1.86** | 2.7 | 12.84 |
| Shape Completion Time (s) | – | 47.53 | 60.23 |

only generated 4 dof top-grasps from a single depth image to generate full 6 dof grasps from simulated viewpoints. The key component was the use of shape completion to model a partly observed object and place it into a physics simulator to simulate depth images from multiple viewpoints. We used FC-GQ-CNN to generate grasps and compared the 6 dof grasps generated with our pipeline to the 4 dof grasps proposed from a depth image captured by a real camera on both single object grasping and grasping in clutter. The single object grasping results show that generating full 6 dof grasps leads to a statistical significant improvement in terms of higher grasp success rate. Major improvements were also prominent for grasping in clutter when generating 6 dof grasps opposed to 4 dof ones.

Despite the good results, shape completion is a major computational bottleneck. Most computation time, however, is not spent on shape completion but on the post-processing of the completed voxel grid which could be improved with better hardware and optimized implementation. Another limitation is that the accuracy of shape completion is conditional on successful segmentation. The analytical region growing segmentation method used here is known to perform poorly in highly cluttered scenes [25]. Thus, the segmentation method would need to be replaced in such cases.

In conclusion the work presented here demonstrates that planning full 6 dof grasps brings significant advantages over 4 dof grasps. This, in turn, poses new interesting research questions. For instance, instead of simulating different viewpoints of the shape completed object, is it maybe better to plan directly on the object itself using, *e.g.,* mesh neural networks [26]? Or, is it possible to generate full 6 dof grasps directly from real depth image, removing the need for shape completion? These questions pave way for interesting future research avenues.

## ACKNOWLEDGMENT

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## REFERENCES


[1] 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," *arXiv preprint arXiv:1703.09312*, 2017.

[2] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.

[3] U. R. Aktas, C. Zhao, M. Kopicki, A. Leonardis, and J. L. Wyatt, "Deep dexterous grasping of novel objects from a single view," *arXiv preprint arXiv:1908.04293*, 2019.

[4] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[5] E. Johns, S. Leutenegger, and A. J. Davison, "Deep Learning a Grasp Function for Grasping under Gripper Pose Uncertainty," *arXiv:1608.02239 [cs]*, 2016.

[6] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning Object Grasping for Soft Robot Hands," *IEEE Robotics and Automation Letters*, 2018.

[7] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[8] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder *et al.*, "Domain randomization and generative models for robotic grasping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[9] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2442–2447.

[10] J. Lundell, F. Verdoja, and V. Kyrki, "Robust Grasp Planning Over Uncertain Shape Completions," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019.

[11] D. Watkins-Valls, J. Varley, and P. Allen, "Multi-Modal Geometric Learning for Grasping and Manipulation," *arXiv:1803.07671 [cs]*, 2018.

[12] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, 2012.

[13] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 85–93.

[14] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, "3d object reconstruction from a single depth view with adversarial learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 679–688.

[15] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[16] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic Grasping of Novel Objects using Vision," *The International Journal of Robotics Research*, 2008.

[17] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, "Deep Reinforcement Learning for Vision-Based Robotic Grasping: A Simulated Comparative Evaluation of Off-Policy Methods," *arXiv:1802.10264 [cs, stat]*, 2018.

[18] L. Pinto and A. Gupta, "Supersizing Self-supervision: Learning to Grasp from 50k Tries and 700 Robot Hours," *arXiv:1509.06825 [cs]*, 2015.

[19] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection," *arXiv:1603.02199 [cs]*, 2016.

[20] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.

[21] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.

[22] J. Mahler, S. Patil, B. Kehoe, J. Van Den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4919–4926.

[23] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, 2014.

[24] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[25] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7283–7290.

[26] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3d shape representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.