Closed-Loop Next-Best-View Planning for Target-Driven Grasping

Michel Breyer, Lionel Ott, Roland Siegwart, and Jen Jen Chung

***Abstract*— Picking a specific object from clutter is an essential component of many manipulation tasks. Partial observations often require the robot to collect additional views of the scene before attempting a grasp. This paper proposes a closed-loop next-best-view planner that drives exploration based on occluded object parts. By continuously predicting grasps from an up-to-date scene reconstruction, our policy can decide online to finalize a grasp execution or to adapt the robot's trajectory for further exploration. We show that our reactive approach decreases execution times without loss of grasp success rates compared to common camera placements and handles situations where the fixed baselines fail. Video and code are available at `https://github.com/ethz-asl/active_grasp`.**

I. Introduction

Planning grasps using on-board sensing is an essential skill for robots to intelligently interact with the unstructured environments outside of assembly lines. While tremendous progress has been made in predicting grasps from partial point cloud data [1], [2], retrieving a target object from clutter remains challenging. One example is shown in Fig. 1 where a manipulator with a wrist-mounted camera is tasked to grab the orange packet of cookies. The item is clearly visible from the initial view, but the graspable region is occluded by the surrounding objects. In order to complete the task, the robot has to move its sensor and actively explore the environment to discover a grasp on the target.

Previous works have proposed active perception systems to aid grasp synthesis. However, they often completely separate exploration and grasp detection [3], [4], or require an initial grasp hypothesis to guide viewpoint selection [5]–[7]. In this paper, we focus on designing a single policy that efficiently discovers stable grasp configurations on a partially occluded target object in clutter.

Our method combines next-best-view planning [8] and real-time grasp detection [9]. We continuously integrate new sensor measurements into a volumetric map of the scene, detect grasps, and compute the next viewpoint based on a count of voxels that are likely to belong to the target object. By re-planning at a rate of 4 Hz, our approach is able to quickly adjust the robot's trajectory based on the updated information, reducing overall execution times. In addition, a task-driven termination criterion is used to let our algorithm decide whether to execute a detected grasp at any point along the robot trajectory. However, to reduce failures, we only commit to a grasp configuration once it remains stable even after integrating new observations.

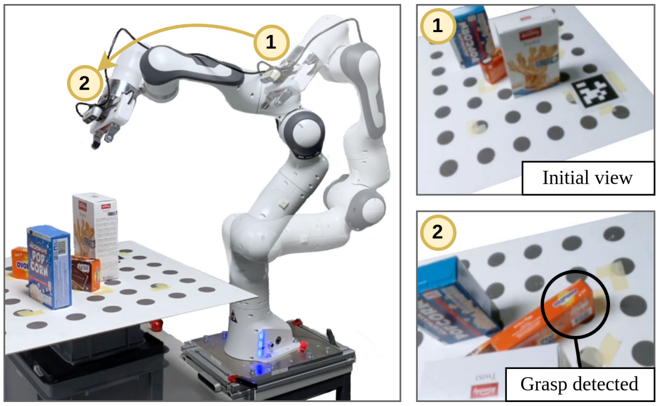

Fig. 1. Detecting grasps on the orange packet using the wrist-mounted camera is hindered by occlusions from surrounding objects. By moving the sensor, our next-best-view grasp planner successfully discovers a grasp on the target item.

We validate our approach in simulation and on a real robotic setup and compare our results to several baselines that represent common camera placements for grasp detection. We found that for scenarios that favor top-down grasps, our approach achieves similar success rates compared to top-down camera placements while on average reducing the search time by around 40%. In addition, we show that our method can adapt to more diverse situations, solving problems where the fixed baselines fail.

In summary, the contributions of this work are:

1) an online next-best-view planner for discovering grasps on a target object in clutter,
2) a task-driven termination criterion based on past grasp predictions,
3) experimental results showing our approach handling diverse situations faster and more robustly compared to baselines.

II. Related Work

Recent advances in object-agnostic grasp synthesis have been mostly driven by deep learning approaches that locate grasps directly in single depth image observations [1], [2]. However, the robustness of such systems can be severely impacted by the amount of available visual information. For example, Gualtieri et al. [10] showed that fusing observations from many viewpoints significantly increased precision and recall compared to two static cameras. One way to tackle the challenge of partial observations is to train models to complete the occluded regions [11], [12]. However, high levels of occlusion still motivate an approach that physically moves the sensor to gather more information.

Active perception [13] studies the problem of determining sensor placements that maximize the information collected

This work was supported in part by the Luxembourg National Research Fund (FNR) 12571953 and ABB Corporate Research.

Autonomous Systems Lab, ETH Zurich, 8092 Zurich, Switzerland, {mbreyer, lioott, rsiegwart, chungj}@ethz.ch.

from the environment and has been used in various applications such as mapping [14], [15], object search [16], [17], pose estimation [18], [19], and 3D reconstruction [20], [21]. A common approach is to select the "best" next view according to some Information Gain (IG) measure [22]. However, choosing a metric can be challenging and is typically highly task specific [23]. In this work, we take inspiration from the rear side voxel IG formulation proposed by Isler et al. [8] for volumetric reconstruction of objects by a mobile robot.

Active vision systems have also been used to aid grasp synthesis. Arruda et al. [5] propose an IG formulation that maximizes surface reconstruction close to the contact points between the object and a given grasp. Gualtieri and Platt [6] directly use the output of their grasp detection pipeline to construct an object-class-specific database of informative viewpoints while Morrison et al. [7] collect data online guided by the entropy in accumulated pixel-wise grasp predictions. Common among these works is that the next view is driven by an initial grasp detection.

In contrast, Kahn et al. [4] plan sensor motion to discover grasp handles within occluded regions of the scene following a classic sense-plan-act approach. Chen et al. [24] train a reinforcement learning policy to increase the visibility of a target object in clutter and to stop exploration to plan grasps.

We combine ideas from both lines of work. Our next-best-view grasp planner explores occluded parts of the object while continuously generating grasp hypotheses, and we use the distribution of past predictions to fix a final configuration.

While most grasp planners are target-agnostic, some works have looked into picking a specific object, either through reinforcement learning [25], [26] or by using instance segmentation to match grasps with the target object [27]. We follow the latter approach and assume that a bounding box of the desired item is provided.

III. Approach

We consider the problem of moving a depth camera attached to the hand of a robotic arm in order to find a parallel-jaw grasp on a given target object. We make the common assumption that the calibration between the optical center of the camera and the end effector is known. In addition, since we are interested in picking a specific object, we assume that we start with a partial view and a 3D bounding box of the target.

Our goal is to find a policy which continuously processes the latest sensor measurements and either returns a stable grasp detection or an informative view towards which the robot should move. An overview of our system is shown in Fig. 2. At every time step t, we integrate the current point cloud observation y_t and camera pose x_t into a Truncated Signed Distance Function (TSDF) [28] reconstruction of the scene and compute voxel-wise grasp affordances [9]. Next, we compute the view x_t^* from which we expect to observe occluded parts of the target item, which is then tracked by a Cartesian velocity controller. Since our primary goal is grasp success (rather than object reconstruction, etc.), we use a stopping criterion based on the history of grasp predictions to determine whether a suitable configuration has been found. By evaluating the policy at a fixed rate, the system can continuously move towards views based on expected information gain, but will execute a grasp whenever the stopping criterion is met. We first present the grasp detection and view planning components in Sections III-A and III-B, before we combine them into a fully autonomous exploration and grasping system in Section III-C.

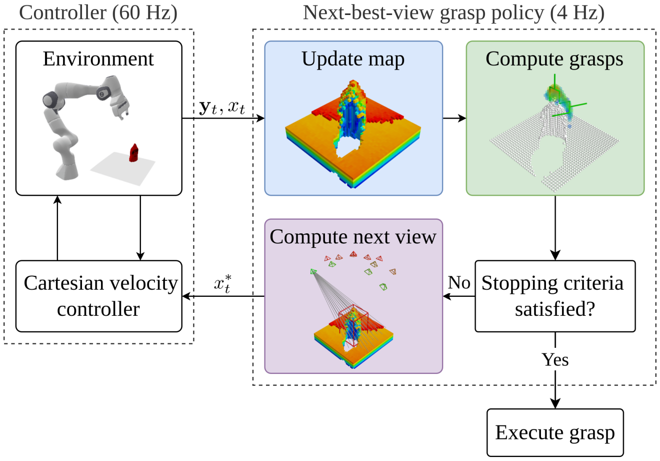

Fig. 2. Overview of the framework. Our policy continuously integrates sensor measurements into a volumetric map of the scene, computes grasps, and re-plans informative views until a stable grasp is detected.

A. Grasp Detection

We use the Volumetric Grasping Network (VGN) for grasp synthesis. VGN is a fully convolutional network that maps a voxel grid M to a grasp quality score Q, along with the associated orientation R and opening width W of a parallel-jaw grasp at every voxel. Given the output of the pre-trained network, we follow the steps of [9] to construct a list of promising grasp configurations ξ_i. Since we are only interested in grasps on the target object, we filter out the hand poses for which the fingertips do not lie within the target's bounding box. In addition, to avoid unreachable configurations, we remove grasps for which no Inverse Kinematics (IK) solution can be found. Finally, we mark the configuration ξ^* with the highest predicted grasp quality as the current best grasp.

B. Next-Best-View Planner

Let x be a view from a set of potential sensor placements $\mathcal{X} \subset SE(3)$. We define $\mathcal{G}_x$ as the predicted information gained by observing the scene from x. The goal of the next-best-view planner is to find the view with the highest predicted Information Gain (IG),

$$x^* = \underset{x \in \mathcal{X}}{\arg\max}\, \mathcal{G}_x. \tag{1}$$

The choice of world representation, sensor placements and information gain are crucial for any active perception task. In the following subsections, we present the different components of our grasp-driven view planner.

1) World Representation: We use a Truncated Signed Distance Function (TSDF) to represent a cubic volume of size l. A TSDF is a three-dimensional grid $\mathcal{M}$ of uniformly sized voxels storing the projected truncated signed distance to the nearest surface. We chose TSDFs due to their efficient incremental updates and noise averaging properties. In addition, we can directly share the map between the IG computation and grasp detection, avoiding the need for multiple maps.

2) View Generation: We distribute a set of view candidates $\mathcal{X}$ on a hemisphere placed on top of the target's bounding box as shown in the purple block in Fig. 2. The radius of the sphere is chosen such that the distances between the camera locations and the bounding box are greater than the minimum depth distance of the sensor. We use IK to discard views that are not reachable by the arm. While more sophisticated methods for selecting the viewpoints are possible, we believe that the evenly spaced views provide sufficient resolution for our purpose.

3) Information Gain: It was shown that surface reconstruction quality and grasp predictions are tightly coupled [10]. In the same way, when grasping using the VGN framework, the completeness of the TSDF reconstruction often has a large impact on grasp discovery and prediction accuracy. For this reason, we propose to use a variant of the rear side voxel IG formulation from Isler et al. [8].

Partially observed objects will cast shadows, i.e. voxels with negative distance values on the occluded side. We use ray casting to count the number of hidden object voxels that would be revealed by placing the camera at any particular viewpoint. More precisely, for each view candidate $x \in \mathcal{X}$, we generate a set of rays $\mathcal{R}_x$ cast from the optical center through each pixel of a virtual camera placed at x. Each ray r traverses a set of voxels $\mathcal{M}_r \subset \mathcal{M}$ until it hits an observed object surface. The gain $\mathcal{G}_x$ is then given by,

$$\mathcal{G}_x = \sum_{r \in \mathcal{R}_x} \sum_{m \in \mathcal{M}_r} \mathcal{I}(m), \tag{2}$$

where $\mathcal{I}(m) = 1$ if the voxel m is located within the target's bounding box and is storing a negative distance value, $\mathcal{I}(m) = 0$ otherwise.

C. Closed-loop Exploration and Grasping

For improved efficiency, we want to continuously process incoming sensor data and react to the updated information. For this reason, we evaluate our policy at a fixed rate. At each update t, we first update the TSDF map with the latest sensor measurements. Next, we evaluate VGN to determine the best grasp candidate ξ_t^* and compute the next best view x_t^* and associated IG $\mathcal{G}_{x_t^*}$ as described in the previous sections. We then decide whether to stop the policy execution or to continue exploring based on three stopping criteria.

First, we impose a time budget in the form of a maximum number of policy updates T_{max}. Second, we terminate if $\mathcal{G}_{x_t^*}$ is below a given threshold $\mathcal{G}_{\text{min}}$, since this means that we don't expect to gain a meaningful amount of additional information. This is designed to capture cases where no feasible grasp configurations can be detected even after fully exploring the target object. Third, we stop once VGN generates a grasp configuration that remained stable over several frames.

The intuition behind the third condition is that grasp detections can be imprecise and vary with updated map information due to occlusions, especially close to the boundaries of observed space. We formalize this as a function of the moving average of the predicted grasp quality at the voxel corresponding to the best grasp's location over a window of size T. More precisely, given the last T grasp quality tensors $Q_{t-T:t}$ predicted by VGN and the voxel m corresponding to ξ_t^*, we consider the grasp stable if,

$$\left(\frac{1}{T} \sum_{t'=t-T}^{t} Q_{t'}(m) \right) > \epsilon_\mu. \tag{3}$$

If one of the stopping criteria is satisfied, we abort the policy and, if found, return the best grasp configuration ξ_t^* to be executed by a separate controller. Otherwise, we set the next best view x_t^* as the target of a Cartesian velocity controller. The velocity controller runs in a separate control loop at a higher rate and generates velocity commands of fixed magnitude in the direction of the most informative viewpoint while ensuring that the camera maintains a minimum distance to the target object.

IV. Evaluation

The goal of our experiments is to answer the following two questions: (a) how does our active exploration approach compare in terms of efficiency and grasp success rates to common camera placement alternatives, and (b) how robust is our adaptive system against different object locations and amounts of occlusions compared to fixed baselines?

A. Experimental Setup

We evaluate our approach by attempting to grasp a target object in different scenarios (shown in Fig. 4). Our test environment consists of a 7-DoF Panda arm with a RealSense D435 rigidly attached to the end effector. For each scenario, we place the target object at a predefined location, surround it with distractor objects, and move the robot arm to a fixed initial configuration. Note that small perturbations to the initial configuration and object locations lead to some variance over multiple runs for a given scenario.

In addition to the real-world setup, we build a simulation environment in PyBullet [29] that mimics the real system. The simulation allows us to run quantitative experiments over a large number of randomly generated scenes following the "packed" protocol from [9]. For each scene, we render a segmentation mask of the initial view and choose the object with the smallest amount of visible pixels as target. Bounding boxes of the target are provided by the physics simulator.

Our policy is implemented in Python using ROS for interfacing the hardware. The extrinsic parameters of the depth sensor are calibrated using the toolbox from Furrer et al. [30]. We use the TSDF implementation from Open3D [31], TRAC-IK [32] for IK computations, and

TABLE I

Parameters used in our experiments.

TSDF size	l	$0.3\,\mathrm{m}$
TSDF voxel count per side	N	40
Number of view candidates	$\lvert\mathcal{X}\rvert$	16
Policy rate		$4\,\mathrm{Hz}$
Maximum number of views	$T_{\max}$	80
Minimum IG	$\mathcal{G}_{\min}$	10
Grasp quality moving average window size	T	12
Grasp quality threshold	ϵ_μ	0.9
Linear velocity		$5\,\mathrm{cm/s}$

Numba [33] to accelerate ray casting. The parameters of our approach are listed in Table I. All experiments were run on a computer equipped with an Intel Core i7-8700K and a GeForce GTX 1080 Ti.

To evaluate grasps returned by our policy, we first plan a trajectory to the grasp pose using MoveIt. To avoid collisions with other objects in the environment, we extract a point cloud representation of the TSDF, compute clusters, and generate convex hulls for each segment which are added as collision objects to the MoveIt planning scene. Next, we close the gripper with constant force and count the grasp as a success if the target object was successfully lifted by $10\,\mathrm{cm}$. Occasionally, MoveIt fails to find a path to the grasp returned by our policy. Since online collision-checking is out of the scope of this work, we remove these trials from our experiments, though we include a more detailed discussion about this problem in Section V.

B. Evaluation Metrics

We evaluate the performance with the following metrics:

- Success Rate (**SR**). Ratio of runs where the target was successfully grasped.
- Failure Rate (**FR**). Ratio of runs where a grasp was detected, but failed during execution.
- Aborted Rate (**AR**). Ratio of runs where no grasp on the target object was found.
- Mean number of **views**. Number of policy updates.
- **Search time**. Time elapsed between receiving a bounding box and returning a grasp configuration.
- **Total time**. Includes the time to execute a detected grasp in addition to the search time.

C. Baselines

We compare our method against three baseline policies that correspond to common camera placement strategies.

- *initial-view*: Detect grasps using a single image captured from the initial view and execute the best grasp.
- *top-view*: Move the camera to the top of the view sphere described in Section III-B.2 and detect grasps from a single top-down image. This is typically an effective strategy for table-top manipulation [11].
- *top-trajectory*: Same as *top-view* but integrate images along the trajectory to the top view.

All baselines use the same controller described in Section III-C to generate robot motion.

TABLE II

Results from the simulation experiments.

Policy	**SR**	**FR**	**AR**	**Views**	**Search time (s)**	**Total time (s)**
initial-view	79	3	18	1 ± 0	1.4 ± 0.1	16.2 ± 2.2
top-view	90	3	7	1 ± 0	10.0 ± 1.4	22.5 ± 1.5
top-trajectory	91	2	7	34 ± 5	9.5 ± 1.3	22.5 ± 2.3
nbv-grasp	89	4	7	18 ± 12	5.9 ± 3.1	19.5 ± 3.4

D. Simulation Experiments

Table II reports the results from running 400 trials with each policy. We observe that the *initial-view* policy has the shortest total time since it does not explore the scene at all. However, this comes at the cost of a lower success rate as strong occlusions lead to a higher number of aborted runs. *Top-view* and *top-trajectory* resulted in similarly high success rates, even though *top-trajectory* integrates more information about the scene along its trajectory. For the generated scenes, the target can typically be picked from the top, making the top view an effective strategy to discover grasps. Finally, the proposed *nbv-grasp* policy has comparable grasping performance, but requires on average only little more than half the time to discover a successful grasp. The higher standard deviation in search time indicates that our policy adapts to the complexity of the scene, efficiently exploring and stopping once a stable grasp has been detected.

To investigate the influence of the stopping criterion, Fig. 3 shows SRs and mean search times for different values of the window size T. We observe that higher values of T result in higher success rates, as the policy is forced to explore longer, thus improving the accuracy of grasp detections in challenging cases. However, this also leads to increased mean search times.

E. Real-world Experiments

Table III shows the results from four grasping trials for each of the four real-world scenarios shown in Fig. 4. For scene (a), where the object and initial camera are favorably placed, all policies succeed in retrieving the target. However, we can see that the *nbv-grasp* policy operates more efficiently compared to the top-view baselines due to the early stopping criterion. Scenes (b) and (c) are designed similarly, with the target being visible, yet heavily occluded from the initial view. This is reflected in the degraded performance of the *initial-view* policy, which fails in three out of the four

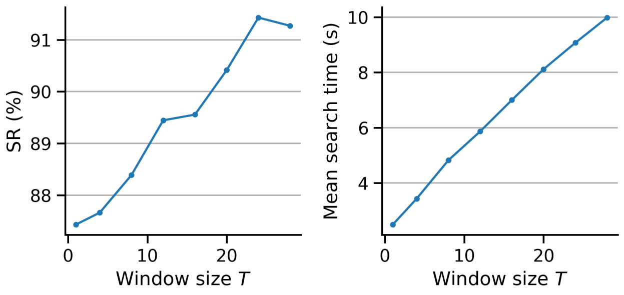

Fig. 3. Success rate and search time vs window size T. Larger values of T force the policy to explore longer, leading to higher success rates at the cost of longer search times.

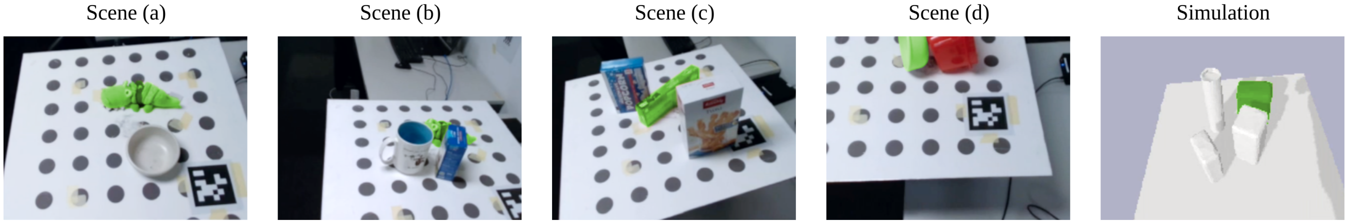

Fig. 4. Images taken from the initial camera view for each test scene with the target object colored in green. Note that the RGB image has a smaller field of view compared to the depth sensor of the RealSense.

TABLE III
RESULTS FROM THE REAL WORLD CASE STUDIES.

Scene	**Policy**	**SR**	**FR**	**AR**	**Views**	**Search time (s)**	**Total time (s)**
(a)	*initial-view*	4/4	0/4	0/4	1 ± 0	1.9 ± 0.7	16.5 ± 0.7
	top-view	4/4	0/4	0/4	1 ± 0	9.8 ± 0.6	23.6 ± 1.1
	top-trajectory	4/4	0/4	0/4	33 ± 1	9.8 ± 0.6	23.2 ± 1.0
	nbv-grasp	4/4	0/4	0/4	18 ± 7	6.3 ± 2.4	20.9 ± 2.6
(b)	*initial-view*	1/4	1/4	2/4	1 ± 0	1.9 ± 0.7	17.1 ± 0.4
	top-view	4/4	0/4	0/4	1 ± 0	13.2 ± 0.8	26.8 ± 0.9
	top-trajectory	4/4	0/4	0/4	46 ± 2	13.2 ± 0.9	27.4 ± 1.1
	nbv-grasp	4/4	0/4	0/4	43 ± 5	12.6 ± 0.9	26.4 ± 0.2
(c)	*initial-view*	1/4	0/4	3/4	1 ± 0	1.9 ± 0.8	17.5
	top-view	4/4	0/4	0/4	1 ± 0	13.5 ± 0.8	24.8 ± 0.9
	top-trajectory	4/4	0/4	0/4	48 ± 1	13.5 ± 0.8	25.0 ± 0.9
	nbv-grasp	4/4	0/4	0/4	30 ± 15	9.6 ± 3.1	23.1 ± 2.2
(d)	*initial-view*	1/4	0/4	3/4	1 ± 0	1.9 ± 0.8	34.5
	top-view	1/4	0/4	3/4	1 ± 0	11.0 ± 0.7	23.8
	top-trajectory	2/4	0/4	2/4	38 ± 0	11.1 ± 0.7	24.5 ± 0.5
	nbv-grasp	4/4	0/4	0/4	27 ± 4	8.5 ± 1.7	25.7 ± 5.3

attempts. Since the target can be grasped from the top, *top-view* and *top-trajectory* successfully handle these cases. The *nbv-grasp* policy also succeeds in all cases, following a similar trajectory as the top-view baselines, as can be seen from the average search times and the accompanying video. Finally, the bowl placed on its side in scene (d) poses a challenge even for the top camera placement. Only the *nbv-grasp* policy consistently explores the side views of the scene and detects a grasp on the rim, highlighting the ability of our approach to adapt to scenes of various complexity.

F. Computation Times

Fig. 5 shows the computation times for one update of our *nbv-grasp* policy measured over 40 simulation runs. We observe that the IG computation accounts for the largest portion of total time. This could be improved by optimizing the ray casting implementation, however, the approach is still efficient enough to run online at 4 Hz.

V. DISCUSSION

The goal of this work is to enable a robot to efficiently retrieve a target object from clutter. We showed that our next-best-view policy often adapts a top-down strategy which is effective in many cases, but also generalizes to situations where the predefined camera locations fail. However, there remain several limitations leaving open directions for future research.

First, our policy considers the kinematic feasibility, but ignores collisions between the arm and the scene when computing grasp and view candidates. This can occasionally lead to situations where a grasp returned by our policy cannot be reached by the robot. This could be tackled with efficient collision checking, either based on geometric primitives [34] or learned collision functions [35].

Second, we assume the availability of bounding boxes for matching grasps to the target object. In a more integrated system, this could be replaced by object detection [36] or instance segmentation [37]. It would also be interesting to include both object and grasp detection into a single active perception formulation.

Finally, we focused our study on the case where the robot can pick the target without additional manipulation. However, in dense clutter grasps on the target can be blocked by other objects, requiring multiple interactions, e.g. removing the blocking objects [19], [38].

VI. CONCLUSION

In this paper, we presented a next-best-view grasp planner that efficiently searches for stable grasp configurations on a partially occluded target object. Our information gain-based approach explores occluded parts of the object, adapting to the complexity of the scene. The presented stopping criteria help to balance exploration and exploitation by terminating

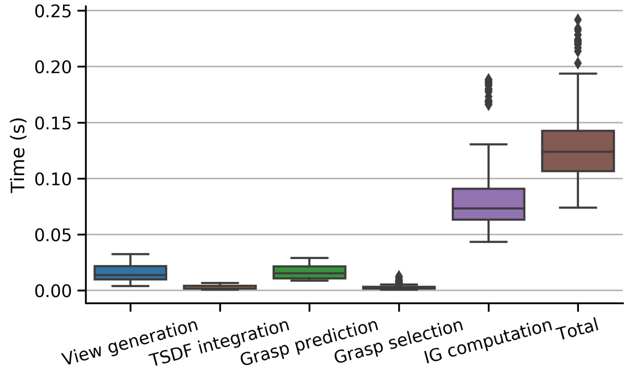

Fig. 5. Computation times for one update of our *nbv-grasp* policy measured over 40 simulation runs.

the search once a stable grasp candidate has been found while reducing the number of false positive detections by enforcing a constraint on the grasp prediction history. We showed the increased robustness and efficiency of our approach compared to common fixed camera placement alternatives in a series of simulated and real world experiments.

In future work, we plan to investigate how to extend our formulation to more complex rearrangement tasks.

References

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems*, 2017.

[2] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes," in *International Conference on Robotics & Automation*, 2021.

[3] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *International Conference on Robotics & Automation*, 2008.

[4] G. Kahn, P. Sujan, S. Patil, S. Bopardikar, J. Ryde, K. Goldberg, and P. Abbeel, "Active exploration using trajectory optimization for robotic grasping in the presence of occlusions," in *International Conference on Robotics & Automation*, 2015.

[5] E. Arruda, J. Wyatt, and M. Kopicki, "Active vision for dexterous grasping of novel objects," in *International Conference on Intelligent Robots and Systems*, 2016.

[6] M. Gualtieri and R. Platt, "Viewpoint selection for grasp detection," in *International Conference on Intelligent Robots and Systems*, 2017.

[7] D. Morrison, P. Corke, and J. Leitner, "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *International Conference on Robotics & Automation*, 2019.

[8] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *International Conference on Robotics & Automation*, 2016.

[9] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric grasping network: Real-time 6 DOF grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[10] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *International Conference on Intelligent Robots and Systems*, 2016.

[11] J. Lundell, F. Verdoja, and V. Kyrki, "Beyond top-grasps through scene completion," in *International Conference on Robotics & Automation*, 2020.

[12] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-DoF grasp detection via implicit representations," in *Robotics: Science and Systems*, 2021.

[13] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, 2018.

[14] M. Popović, T. Vidal-Calleja, G. Hitz, J. J. Chung, I. Sa, R. Siegwart, and J. Nieto, "An informative path planning framework for UAV-based terrain monitoring," *Autonomous Robots*, 2020.

[15] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An efficient sampling-based method for online informative path planning in unknown environments," *IEEE Robotics and Automation Letters*, 2020.

[16] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *International Conference on Robotics & Automation*, 2019.

[17] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception," in *International Conference on Robotics & Automation*, 2020.

[18] S.-K. Kim and M. Likhachev, "Planning for grasp selection of partially occluded objects," in *International Conference on Robotics & Automation*, 2016.

[19] P. K. Murali, A. Dutta, M. Gentner, E. Burdet, R. Dahiya, and M. Kaboli, "Active visuo-tactile interactive robotic perception for accurate object pose estimation in dense clutter," *IEEE Robotics and Automation Letters*, 2022.

[20] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, 2015.

[21] J. Daudelin and M. Campbell, "An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-D objects," *IEEE Robotics and Automation Letters*, 2017.

[22] C. Connolly, "The determination of next best views," in *International Conference on Robotics & Automation*, 1985.

[23] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *International Journal of Robotics Research*, 2011.

[24] X. Chen, Z. Ye, J. Sun, Y. Fan, F. Hu, C. Wang, and C. Lu, "Transferable active grasping and real embodied dataset," in *International Conference on Robotics & Automation*, 2020.

[25] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," in *Conference on Robot Learning*, 2017.

[26] Y. Fujita, K. Uenishi, A. Ummadisingu, P. Nagarajan, S. Masuda, and M. Y. Castro, "Distributed reinforcement learning of targeted grasping with active vision for mobile manipulators," in *International Conference on Intelligent Robots and Systems*, 2020.

[27] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-DOF grasping for target-driven object manipulation in clutter," in *International Conference on Robotics & Automation*, 2020.

[28] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Conference on Computer Graphics and Interactive Techniques*, 1996.

[29] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016.

[30] F. Furrer, M. Fehr, T. Novkovic, H. Sommer, I. Gilitschenski, and R. Siegwart, "Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets," in *Field and Service Robotics*, 2018.

[31] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847 [cs]*, 2018.

[32] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *International Conference on Humanoid Robots*, 2015.

[33] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: a LLVM-based Python JIT compiler," in *Workshop on the LLVM Compiler Infrastructure in HPC*, 2015.

[34] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in *International Conference on Robotics & Automation*, 2012.

[35] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox, "Object rearrangement using learned implicit collision functions," in *International Conference on Robotics & Automation*, 2021.

[36] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[37] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation," in *Conference on Robot Learning*, 2019.

[38] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *International Conference on Robotics & Automation*, 2019.