Active-Perceptive Motion Generation for Mobile Manipulation

Snehal Jauhri[1*], Sophie Lueth[1*], and Georgia Chalvatzaki[1,2,3]

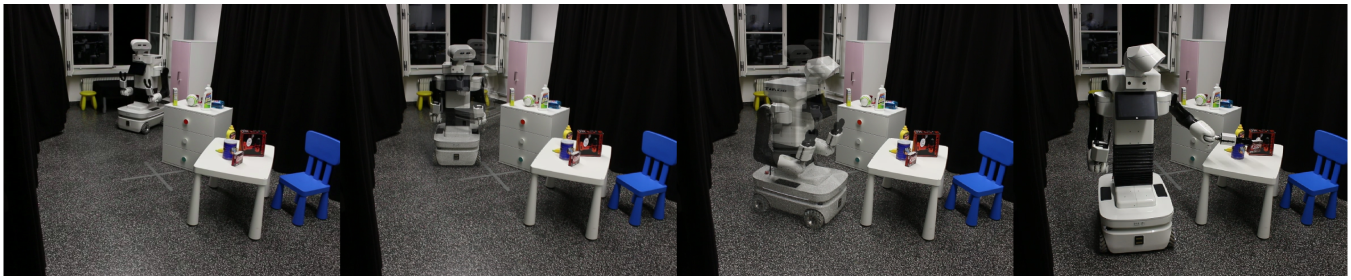

Fig. 1: Real-world execution of our active perception and grasping pipeline. The mobile manipulator robot, equipped with a head-mounted camera, efficiently explores the scene to detect grasps on the target object and effectively performs the grasp by evaluating its executability. Video demonstrations are provided at our project website: sites.google.com/view/actpermoma.

***Abstract*— Mobile Manipulation (MoMa) systems incorporate the benefits of mobility and dexterity, due to the enlarged space in which they can move and interact with their environment. However, extracting task-relevant visual information in cluttered environments, such as households, remains challenging. In this work, we introduce an Active Perception (AP) pipeline for mobile manipulators to generate motions that are informative toward manipulation tasks, such as grasping in unknown, cluttered scenes. Our proposed approach, Act*Per*MoMa, generates robot paths in a receding horizon fashion by sampling paths and computing path-wise utilities. These utilities trade-off maximizing the visual Information Gain (IG) for scene reconstruction and the task-oriented objective, e.g., grasp success, by maximizing grasp reachability. We show the efficacy of our method in simulated experiments with a dual-arm TIAGo++ MoMa robot performing mobile grasping in cluttered scenes with obstacles. Also, we demonstrate the transfer of our mobile grasping strategy to the real world, indicating a promising direction for active-perceptive MoMa.**

I. Introduction

We envision a future where embodied agents, such as mobile manipulators, autonomously operate in everyday environments like households. However, due to the unstructured and unpredictable nature of the real world, some robots will actively gather information about their surroundings through embodied sensors whilst operating [1]. While current advances in AI and machine learning for robotics have unlocked new capabilities for table-top manipulation [2]–[4], or language-driven navigation and manipulation [5]–[9], mobile manipulation in unknown (or partially known) scenes poses significant challenges [10]–[12], as the MoMa robot needs to consider both scene reconstruction and task-oriented objectives.

In AP for mobile robots, the robot's objective is typically reconstruction, i.e., obtaining volumetric information about the scene/target object [13]–[16]. Often, this is achieved via a Next-Best-View (NBV) strategy [17]–[21], primarily choosing viewpoints based on information gain (IG) [17] that minimizes uncertainty by exploring unobserved regions. A good overview and comparison of different IG formulations for NBV is provided in [22]. Notably, an AP approach that only considers movement to the NBV with the most information gain can lead to unnecessarily large motions. Hence, the authors of [23] consider IG over paths using a graph-based approach, while in [24], a receding horizon viewpoint and path planning method is proposed, a formulation that adapts to newly observed information.

This work focuses on AP for mobile grasping. For grasping with static manipulators, recent AP methods adopt grasp quality metrics to choose the next robot viewpoint that minimizes uncertainty in the grasp pose estimation [2], [25]. However, these methods have several drawbacks when considering their application to MoMa. First, a MoMa robot can move *further* within the scene and reach (almost) *any* viewpoint. Thus, wasteful motions can be especially costly, and information gained *during* movement has to be considered, opposing NBV-only approaches. Second, the reachability of grasps and viewpoints is crucial when formulating planning for MoMa, since grasps of high quality could be challenging to reach and might cause unnecessary robot movements.

We propose an effective and efficient approach to visually informative motion generation for mobile manipulators performing tasks in unknown, cluttered scenes, focusing on mobile grasping. Our method, Active-Perceptive Motion Generation for Mobile Manipulation (Act*Per*MoMa), plans over collision-free paths generated toward objects of interest in a receding horizon fashion. We compute path-wise utilities over robot poses, balancing the collection of visual information gain to infer good grasps (*exploration*) and task-specific information, i.e. grasp executability (*exploitation*). Our holistic pipeline is illustrated in Fig. 2.

[*] Authors contributed equally

This research received funding from the European Union's Horizon program under grant agreement no. 101120823, project MANiBOT, the German Research Foundation (DFG) Emmy Noether Programme (CH 2676/1-1), and the Daimler Benz foundation.

[1] Computer Science Department, Technische Universität Darmstadt, Germany [2] Hessian.AI, Darmstadt, Germany [3] Center for Mind, Brain and Behavior, Uni. Marburg and JLU Giessen, Germany {snehal.jauhri,georgia.chalvatzaki}@tu-darmstadt.de, sophie.lueth@stud.tu-darmstadt.de

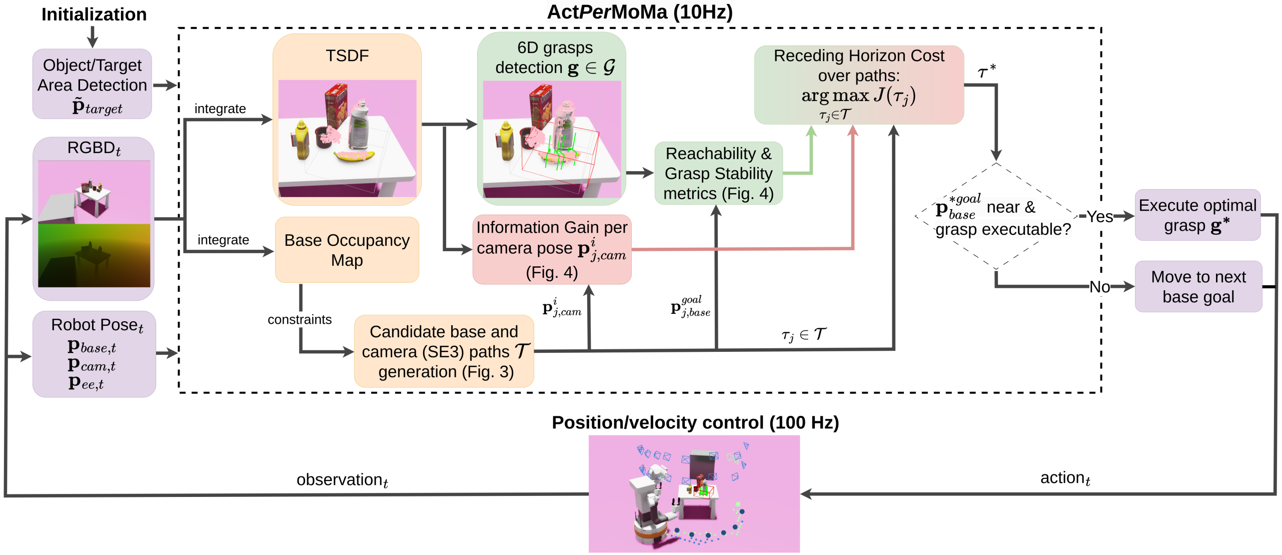
Initialization
ActPerMoMa (10Hz)
Object/Target Area Detection $\tilde{\mathbf{p}}_{target}$
RGBD$_t$
Robot Pose$_t$ $\mathbf{p}_{base,t}$ $\mathbf{p}_{cam,t}$ $\mathbf{p}_{ee,t}$
integrate
integrate
TSDF
Base Occupancy Map
constraints
6D grasps detection $\mathbf{g} \in \mathcal{G}$
Information Gain per camera pose $\mathbf{p}^i_{j,cam}$ (Fig. 4)
Reachability & Grasp Stability metrics (Fig. 4)
Receding Horizon Cost over paths: $\arg\max_{\tau_j \in \mathcal{T}} J(\tau_j)$
τ^*
Candidate base and camera (SE3) paths $\mathcal{T}$ generation (Fig. 3)
$\mathbf{p}^i_{j,cam}$
$\mathbf{p}^{goal}_{j,base}$
$\tau_j \in \mathcal{T}$
$\mathbf{p}^{*goal}_{base}$ near & grasp executable?
Yes
No
Execute optimal grasp $\mathbf{g}^*$
Move to next base goal
Position/velocity control (100 Hz)
observation$_t$
action$_t$

Fig. 2: Act*Per*MoMa pipeline. Using a rough initial knowledge about the target area or target object position $\widetilde{\mathbf{p}}_{target}$, we continuously plan and execute informative motions for the mobile grasping task. At every timestep t, the RGBD information from the head-mounted embodied camera is integrated into a scene TSDF for both grasp detection and information gain computation. Using the currently known free space for movement of the robot base, we sample candidate robot paths $\mathcal{T}$, including both base and camera poses, towards the target. For each candidate path $\tau_j \in \mathcal{T}$, we compute the information gained from camera views $\mathbf{p}^i_{j,cam}$ in the path, and the reachability of stable detected grasps from the final base poses $\mathbf{p}^{goal}_{j,base}$ in the path. We trade-off these objectives with a receding horizon cost J_τ and take a step of the optimal path τ^* for execution at every timestep.

II. Active-Perceptive Motion Generation for Mobile Manipulation

We consider scenarios where a MoMa robot is placed in a previously unseen environment and is tasked with picking up a target object placed on a surface among clutter. To achieve this task, the mobile manipulator needs to use its mobile base to move in the scene, an RGBD camera to gather information about the scene, and an arm/end-effector to execute 6DoF grasps. Without loss of generality to different physical MoMa designs, we can simplify the robot's state as a combination of its mobile base pose $\mathbf{p}_{base} \in SE(2)$, its camera pose $\mathbf{p}_{cam} \in SE(3)$, and its end-effector pose $\mathbf{p}_{ee} \in SE(3)$.

We assume to know a rough target object position in the form of an approximate bounding target object position, the center of which we denote by $\widetilde{\mathbf{p}}_{target}$. This can be achieved by exploring and using an RGB object detector or by evaluating a user instruction like 'Pickup the object from the right corner of the table', although no prior scene information is required for our method. From the point cloud of the embodied camera, we build a volumetric representation of the scene, a 3D Truncated Signed Distance Function (TSDF). This TSDF is used to plan motions in the observed environment and to detect grasps. We consider example scenarios visualized in Fig. 2 and Fig. 3.

A. Candidate goals & paths generation

Our objective is to move towards the target object in the most informative and time/energy efficient manner and grasp it. We measure efficiency w.r.t the total distance traveled and viewpoints visited. At each time step, we sample candidate paths for the robot and the corresponding SE(3) camera poses, which always look at the target area, and evaluate utilities over those paths. For this, we sample N_b goal base poses $\{\mathbf{p}^{goal_i}_{base}\}_{i=0}^{N_b}$ near the approximate target object position $\widetilde{\mathbf{p}}_{target}$ within a radius that affords robot reachability [26], [27]. Every time step, we ensure that base goals are collision-free by performing a simple collision check with the scene's continuously generated TSDF grid or base occupancy map.

We aim to obtain the optimal motion of the robot toward the target object by planning paths to the candidate base goals. Thus, we sample M candidate paths $\mathcal{T} = \{\tau_j\}_{j=0}^{M}$ to all the N_b candidate base goals $\{\mathbf{p}^{goal_i}_{base}\}_{i=0}^{N_b}$ using optimal path planners—in this work we plan over discretized grids with A*. Each path $\tau \in \mathcal{T}$ consists of base poses from the current robot base to the base goal and feasible camera poses $\mathbf{p}_{cam} \in SE(3)$ along the path, i.e., $\tau_i = \{\{\mathbf{p}^0_{base}, \mathbf{p}^0_{cam}\}, \{\{\mathbf{p}^1_{base}, \mathbf{p}^1_{cam}\} \ldots \{\mathbf{p}^{goal}_{base}, \mathbf{p}^{goal}_{cam}\}\}$. Example candidate paths are visualized in Fig. 3.

B. Receding-horizon control

We use a receding-horizon control formulation to generate the robot's motion to find and execute a grasp on the target object. At each time step, we choose the optimal path among the sampled candidate paths $\mathcal{T}$ and execute an action towards the first waypoint along this current optimal path. The re-computation of actions at every timestep ensures robot reactivity to newly observed scene information.

Formally, given the observation of the scene, i.e., the observed TSDF $\mathbf{o}_{\text{TSDF}}$, the detected set of grasps $\mathcal{G}$, and the sampled candidate paths $\mathcal{T}$, we compute the current optimal path $\tau^* \in \mathcal{T}$ based on the expected information gain J_{IG} and the utility of the grasps' executability J_{exec} in the paths:

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} J_{IG}(\mathbf{o}_{\text{TSDF}}, \tau) + J_{exec}(\mathcal{G}, \tau) \tag{1}$$

Utilities J_{IG} and J_{exec} are detailed in subsections II-C & II-D.

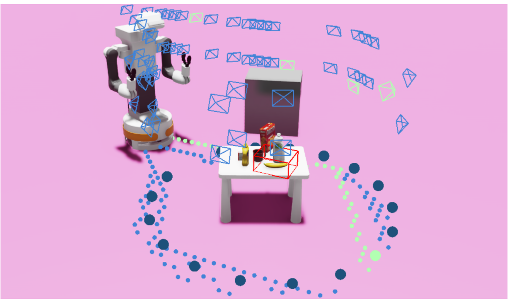

Fig. 3: Example scene with sampled candidate paths (blue) for the robot pose towards the target object (red box). The paths consist of SE(2) poses for the base and SE(3) poses for the head-mounted camera (visualized from the robot to the base goals). The current optimal path is highlighted in green.

For movement at every time step, we use the first waypoint along the chosen optimal path, i.e., $\{\mathbf{p}^{*1}_{base}, \mathbf{p}^{*1}_{cam}\} \in \tau^*$ and run a low-level controller that executes IK-based velocities for the robot base and the camera. If the optimal path τ^* contains exactly one waypoint, i.e., the robot is close enough to the final chosen base goal $\mathbf{p}^{*goal}_{base}$, we finally consider grasp execution. If the grasp execution utility J_{exec} is above a threshold, we execute the grasp with the highest utility (sec. II-D) by activating the arm/end-effector and planning a motion to the SE(3) grasp.

C. Information gain computation & grasp detection

Information gain computation: Our perception objective is to obtain enough information about the target object to grasp it. We continuously build a voxel-based TSDF representation $\mathbf{o}_{\text{TSDF}}$ of the scene and calculate the rear-side voxel IG IG_{rear} inspired by [2], [22]: For every viewpoint $\mathbf{p}_{cam}$, a set of rays R are generated by casting from a virtual camera placed at the respective view pose. Every ray r traverses voxels of the TSDF $v \subset \mathbf{o}_{\text{TSDF}}$ until it hits an observed surface. Therefore the rear-side IG is computed as $IG_{\text{rear}} = \sum_{r \in R} \sum_{v} \mathcal{I}(v)$, where $\mathcal{I}(v) = 1$ if the voxel is on the rear of an existing voxel and within the approximate target object bounding box.

Unlike [22], [2], we consider not only the Next-Best-View (NBV) but the IG *over paths* taken by the robot. Moreover, we also consider the cost of reaching the viewpoints in the paths by weighting the IG by the distance to the viewpoints $dist(\mathbf{p}_{cam})$ along the path. This takes care of our requirement that *information gained sooner is better than later*. We can thus calculate the total IG over each candidate path τ as

$$J_{IG}(\mathbf{o}_{\text{TSDF}}, \tau) = \sum_{\mathbf{p}_{cam} \in \tau} \frac{IG_{rear}(\mathbf{o}_{\text{TSDF}}, \mathbf{p}_{cam})}{dist(\mathbf{p}_{cam})^2}. \quad (2)$$

An example visualization of the rear-side voxel IG is provided in Fig 4.

Grasp detection: At every time step, we query grasps in the target object region using the observed TSDF of the scene using a grasp detection network. In this work, we use the VGN [28] grasp detection network to predict an SE(3) grasp pose for every 3D voxel of the TSDF along with a grasp quality q. We use a grasp quality threshold q_{th} hyperparameter to detect good grasps with a high likelihood of success. Given that the TSDF contains only partial/incomplete information, it is also important to consider grasp detector inaccuracy. Hence, as in [2], we also consider grasp *stability* by ensuring that a grasp predicted for the same 3D voxel on the TSDF has a high-quality score for n_{stab} steps.

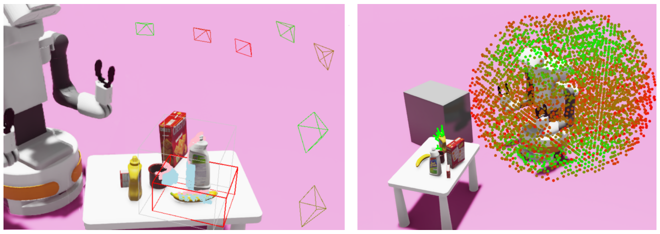

Fig. 4: Left: Example rear-side Information Gain (IG) for a candidate view. Pink voxels denote observed TSDF voxels. Blue voxels are on the rear side of the observed TSDF, which could be revealed by a candidate view. Views are colored red to green, denoting lower to higher IG. **Right:** Reachability map of the robot's left arm, reduced from 6 dimensions (SE(3)) to 3 for visualization. Red and green points denote lower and higher reachability. Current detected 6D grasps are visualized in green on a target object.

D. Reachability utility & grasp selection

To achieve the task, i.e., to grasp the target object, the robot also needs be positioned at a base goal where a high-quality grasp can be executed easily. We achieve this ability by computing a grasp reachability/ utility J_{exec} corresponding to each candidate path τ. The reachability of any SE(3) end-effector pose of a robot from a given base pose can be found by pre-computing a reachability map [29]–[31]. We refer to [29], [31] for a full description of reachability map computation. A visualization of the reachability map used in our approach is provided in Fig 4.

In our pipeline, we pre-compute the reachability map $\mathcal{R}$ and query it for *each grasp* $\mathbf{g} \in \mathcal{G}$, when executed *from each candidate base goal* $\mathbf{p}^{goal}_{base}$ corresponding to our sampled paths $\tau \in \mathcal{T}$. The highest reachability over all grasps gives us a utility score J_{exec} for each candidate path τ and the corresponding grasp will be executed if the robot reaches the respective base goal. To ensure that the proximity of the robot to the base goal of the path $\mathbf{p}^{goal}_{base} \in \tau$ is also considered, we weigh the reachability utilities by the length of the path $len(\tau)$, resulting in

$$J_{exec}(\mathcal{G}, \tau) = \frac{\max_{\mathbf{g} \in \mathcal{G}} \mathcal{R}(\mathbf{g}, \mathbf{p}^{goal}_{base})}{len(\tau)} \quad (3)$$

E. Additional hyperparameters

To smoothly switch between the two objectives in (1), we weigh the IG and grasp execution utilities by factors w_{IG} and w_{exec}. Also, to avoid noisy grasps being used for movement and execution, we filter out unstable grasps, i.e., grasps that disappear after a few timesteps. Furthermore, to minimize oscillation between two base goals with similar overall utility (eq. 1), that move the robot in opposing directions, we introduce a momentum term that continues to move the robot in a direction unless the utility of another direction is significantly higher.

TABLE I: Ablations & Hyperparameter study – Complex scenes

Hyperparameters					
Approach	SR (%) ↑	AR (%) ↓	GFR (%) ↓	d_{total} (m) ↓	v_{total} ↓
Act*Per*MoMa-Quality (0.7)	**91.4**	**0.6**	8.0	**4.57 ± 2.39**	**16.47±8.78**
Act*Per*MoMa-Quality (0.9)	88.8	5.0	**6.2**	4.84±2.73	17.11±9.54
Act*Per*MoMa-StableGrasp (1)	**92.6**	2.2	**5.2**	**4.57±2.51**	**16.20±8.65**
Act*Per*MoMa-StableGrasp (5)	91.2	2.8	6.0	4.92 ± 2.48	17.47±8.80
Act*Per*MoMa-IGweight (3.0)	85.8	8.6	5.6	5.10±3.18	17.83± 10.38
Act*Per*MoMa-IGweight (0.2)	**92.6**	**1.8**	5.6	**4.31±2.27**	**15.60±8.39**
Act*Per*MoMa-momentum (0)	83.2	13.0	**3.8**	5.45±3.40	18.77±11.06
Act*Per*MoMa-momentum (700)	**90.6**	**3.8**	5.6	**4.56±2.53**	**16.28 ± 9.11**
Ablation					
Act*Per*MoMa[†]	92.6	1.80	5.6	4.31±2.27	15.60±8.39
Act*Per*MoMa-IG-only	**96.8**	**0.8**	**2.4**	3.52±1.61	12.87±6.31
Act*Per*MoMa-no-weights	48.4	42.0	10.6	9.05±3.79	13.28±10.44
Hard Grasps					
Act*Per*MoMa[†]	**61.8**	**29.6**	8.6	7.19±4.17	**24.81±13.24**
Act*Per*MoMa-IG-only	56.2	36.2	**7.6**	**5.77±4.11**	25.52±13.82

[†] Quality=0.8, StableGrasp=1, IGweight=0.2, momentum=800

III. Experiments

A. Experimental setup & metrics

We run our experiments in simulation and the real world on a dual-armed TIAGo++ mobile manipulator with a holonomic base and a head-mounted camera. In the simulated setup in Isaac Sim, the robot starts at a distance of 2 m w.r.t. the approximate target location. We consider two scenarios: a simple one, where a table is placed in free space and with 4 randomly spawned objects from the YCB dataset [32], and a more complex one, with 6 objects to create more clutter and with a random obstacle sampled around the table to obstruct the path of the robot.

We employ the following metrics and use ”rate” synonym to ”percentages of episodes”: **Success Rate** (SR), finishing with successful grasp execution, **Abort Rate**: (AR) ending without finding executable grasps in the given time budget, **Grasp Failure Rate** (GFR): ending in grasp failure, **total distance covered** (d_{total}), and the **total number of views** visited (v_{total})).

For a more details, a full overview of experiments including the *real-world demonstrations*, ablations and hyperparameter study, see the full paper and the project website: https://sites.google.com/view/actpermoma.

B. Ablations & hyperparameter study

We present the results for (i) Act*Per*MoMa-IG-only: ablation without the grasp executability objective in which case we execute a grasp as soon as we are 0.85 m of the object; (ii) Act*Per*MoMa-no-weights: ablation without path-length-related scaling of the utilities (see eqs. 2, 3)). Additionally, we tune important parameters to find the best configuration for our method and present results for varying grasp quality thresholds (Act*Per*MoMa-Quality), grasp stability windows (Act*Per*MoMa-StableGrasp), IG weighting factors (Act*Per*MoMa-IGweight), and the momentum term that punishes oscillatory paths (Act*Per*MoMa-momentum).

Table I presents the results of our study for complex scenes. Here we see the benefits of our momentum term as high momentum in complex scenes leads to better performance and reduced path lengths. Our ablation places the method without the grasp-related utility higher than the full Act*Per*MoMa approach, but in the “hard” grasp scenario, we notice a significant ∼6% performance improvement when accounting for the grasps utility while planning.

TABLE II: Comparison with baselines

Approach	**SR (%)** ↑	**AR (%)** ↓	**GFR (%)** ↓	$d_{\textbf{total}}$ **(m)** ↓	$v_{\textbf{total}}$ ↓
Simple scenes					
Naive	95.2	**1.2**	3.6	**1.36±0.28**	**5.93±3.77**
Random	93.2	5.0	1.8	4.38±1.98	15.24±6.47
Breyer et al. [2]	92.0	8.0	**0.0**	3.71±1.78	12.56±6.12
Act*Per*MoMa (ours)	**95.4**	1.4	3.2	3.59±1.69	12.67 ± 5.39
Complex Scenes					
Naive	86.8	2.6	10.6	1.55±0.45	**8.72±8.89**
Random	90.4	5.8	**3.2**	3.97±1.56	13.98±5.39
Breyer et al. [2]	90.0	6.0	4.0	**3.32±1.47**	12.32±6.13
Act*Per*MoMa (ours)	**92.6**	**2.2**	5.2	4.57±2.51	16.20±8.65
Complex Scenes (Hard Grasps)					
Naive	43.8	49.2	7.0	6.06±3.81	27.25±13.95
Random	24.6	68.4	7.0	**3.82±2.23**	**16.29±11.07**
Breyer et al. [2]	47.2	48.8	**4.0**	5.11±3.83	24.03±14.88
Act*Per*MoMa (ours)	**61.8**	**29.6**	8.6	7.19±4.17	24.81±13.24

C. Baseline Comparison

We consider baseline methods with and without IG objective to show in which cases AP is necessary. Namely, we compare against (i) a naive approach (Naive) in which we navigate the robot towards the approximate target location and activate grasp execution if, within a 0.85 m distance from the object, a high-quality grasp has been detected; (ii) a random approach (Random) in which, we randomly select a feasible base goal each time step around the approximate target object location until a grasp has been detected. (iii) the method by Breyer et al. [2] adapted for mobile manipulation, in which we compute the IG per view (and not accumulated over paths) sampled on a hemisphere of radius 1m around the approximate object location. In this case, we always move to the viewpoint with the highest IG. If we are within reach of the object and a grasp has been found, we execute it.

Table II presents the results for simple and complex scenes. For simple scenes, the need for path planning seems to be alleviated, as an approach as simple as the Naive approach performs as well as ours, outperforming Breyer et al. [2] In complex scenes, the SR margin with which Act*Per*MoMa leads, widens. Nevertheless, the hard-grasp case shows the significant benefit of Act*Per*MoMa w.r.t baselines. Notably, we highlight the large benefit in finding grasps (at least ∼20% lower abort rate) compared to baselines.

D. Limitations

An issue of methods that use reactive planning (such as ours) is that, depending on the resolution of the sampled base goals and the sampling frequency, the robot can get stuck in deadlocks trying to switch between base goals leading to oscillating motions. Although we introduce the momentum as a penalty for this behavior, some amount of deadlocks can still exist.

Another limitation is the limited volumetric information in the target area in very occluded scenes, making the IG computation difficult. This can prominently be observed in Breyer et al. [2], as very occluded objects that get partially discovered from different views often prompt a ‘zigzag’ path. Although we improve this behavior by not just considering the best NBV to decide where to go but instead using the whole spatial distribution provided by our sampled base goals, we consider future work with Reinforcement Learning agents leveraging a combination of our active and some random exploration to mitigate this effect.

References

[1] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, pp. 177–196, 2018.

[2] M. Breyer, L. Ott, R. Siegwart, and J. J. Chung, "Closed-loop next-best-view planning for target-driven grasping," 2022. [Online]. Available: https://arxiv.org/abs/2207.10543

[3] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[4] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[5] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 608–10 615.

[6] H. Wang, W. Wang, W. Liang, S. C. Hoi, J. Shen, and L. V. Gool, "Active perception for visual-language navigation," *International Journal of Computer Vision*, vol. 131, no. 3, pp. 607–625, 2023.

[7] D. Shah, B. Osiński, S. Levine *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*. PMLR, 2023, pp. 492–504.

[8] N. Yokoyama, A. W. Clegg, E. Undersander, S. Ha, D. Batra, and A. Rai, "Adaptive skill coordination for robotic mobile manipulation," *arXiv preprint arXiv:2304.00410*, 2023.

[9] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[10] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1647–1654.

[11] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4583–4590.

[12] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[13] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3d object reconstruction with positioning error," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, p. 159, 2014.

[14] C. Potthast and G. S. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 148–164, 2014.

[15] T. Zaenker, C. Lehnert, C. McCool, and M. Bennewitz, "Combining local and global viewpoint planning for fruit coverage," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.

[16] 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*, vol. 5, no. 2, pp. 1500–1507, 2020.

[17] S. Isler, R. Sabzevari, J. A. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3d reconstruction," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3477–3484, 2016.

[18] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1462–1468.

[19] D. Watkins-Valls, P. K. Allen, H. Maia, M. Seshadri, J. Sanabria, N. Waytowich, and J. Varley, "Mobile manipulation leveraging multiple views," 2021. [Online]. Available: https://arxiv.org/abs/2110.00717

[20] M. Naazare, F. G. Rosas, and D. Schulz, "Online next-best-view planner for 3d-exploration and inspection with a mobile manipulator robot," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3779–3786, apr 2022.

[21] L. Bartolomei, L. Teixeira, and M. Chli, "Semantic-aware active perception for uavs using deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3101–3108.

[22] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3d object reconstruction," *Autonomous Robots*, vol. 42, no. 2, 2018.

[23] T. Zaenker, J. Rückin, R. Menon, M. Popović, and M. Bennewitz, "Graph-based view motion planning for fruit detection," *arXiv preprint arXiv:2303.03048*, 2023.

[24] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[25] D. Morrison, P. Corke, and J. Leitner, "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8762–8768.

[26] S. Jauhri, J. Peters, and G. Chalvatzaki, "Robot learning of mobile manipulation with reachability behavior priors," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8399–8406, 2022.

[27] T. Birr, C. Pohl, and T. Asfour, "Oriented surface reachability maps for robot placement," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3357–3363.

[28] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. I. Nieto, "Volumetric grasping network: Real-time 6 DOF grasp detection in clutter," in *CoRL*, ser. Proceedings of Machine Learning Research, vol. 155. PMLR, 2020, pp. 1602–1611.

[29] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 3229–3236.

[30] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *ICRA*, 2013.

[31] A. Makhal and A. K. Goins, "Reuleaux: Robot base placement by reachability analysis," in *IRC*. IEEE Computer Society, 2018, pp. 137–142.

[32] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.