Toward a Plug-and-Play Vision-Based Grasping Module for Robotics

François Hélénon*†[1], Johann Huber*[1], Faïz Ben Amar[1] and Stéphane Doncieux[1]

Abstract— **Despite recent advancements in AI for robotics, grasping remains a partially solved challenge. The lack of benchmarks and reproducibility prevents the development of robots that can interact with open environments autonomously. The generalizing capabilities of foundation models are promising, but the computational cost is very high, and the adaptation capabilities demonstrated on real robots are still limited. This paper takes an opposite perspective by introducing a vision-based grasping framework that can easily be transferred across multiple manipulators. Leveraging Quality-Diversity (QD) algorithms, the framework generates diverse repertoires of open-loop grasping trajectories, enhancing adaptability while maintaining a diversity of grasps. This framework addresses two main issues: the lack of an off-the-shelf vision module for detecting object pose and the generalization of QD trajectories to the whole robot operational space. The proposed solution combines multiple vision modules for 6DoF object detection and tracking while rigidly transforming QD-generated trajectories into the object frame. Experiments on a Franka Research 3 arm and a UR5 arm with an SIH Schunk hand demonstrate comparable performance when the real scene aligns with the simulation used for grasp generation. This work represents a significant stride toward building a reliable vision-based grasping module that is transferable to new manipulator platforms and adaptable to diverse scenarios without further training iterations.**

I. INTRODUCTION

Recent advances in AI have made significant progress toward building autonomous robots to release humans from strenuous tasks. Those advances include natural language-conditioned planning [1], foundation architectures [2], and efficient optimization of controllers using generative models [3]. This progress suggests that the research field is getting closer to making robots operate in open-ended environments. However, some basic skills are only partially solved, and deploying them on a real robot requires significant engineering efforts to make them work in a given context. Grasping is an eloquent example of such a skill, as no off-the-shelf modules allow to address vision-based grasping on several grippers.

Data-greedy approaches are becoming the main paradigm in the field nowadays. Grasping is more and more addressed with generative AI methods [4][5][6], while the training of end-to-end controllers for mobile manipulators involves extremely large Transformers-based architecture [7][8]. Despite their promising generalization capabilities, the cost of these methods raises concerns about how to make such energy-demanding approaches sustainable [9].

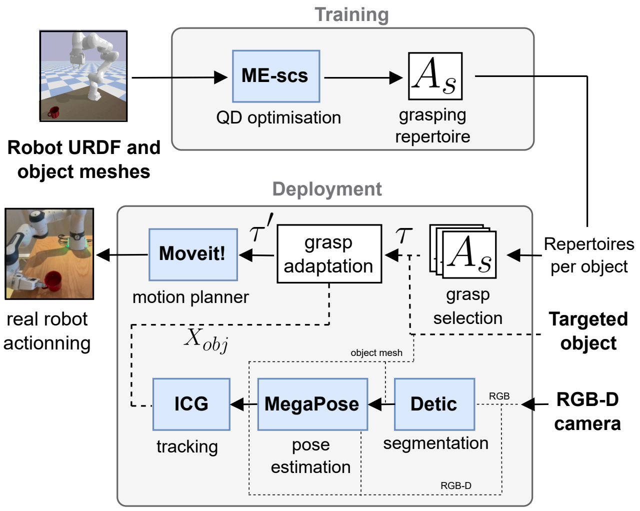

Fig. 1. **Overview of the proposed framework.** It involves utilizing 3D models of the robot, target objects, and RGB-D camera data. A diverse grasping repertoire is generated with ME-scs [17] in simulation. The integration pipeline predicts the object pose through a sequence of perception modules [30][31][32]. The selected grasping trajectory is transformed into the object frame and fed to a motion planner to generalize the trajectory to the whole operational space. This adaptable framework is compatible with various manipulators with minimal need for engineering efforts.

The usage of data-greedy methods, however, is recent in the history of Robotics [10]. For a long time, grasping was addressed with analytic-based approaches and motion planning [11]. The issue was the limited adaptation capabilities that the data-greedy approach could circumvent. The authors of the present work argue that many skills should be addressable with significantly less energy cost than the very large AI models, with enough generalization to provide robots the capability to solve a wide range of tasks in open environments while keeping a certain level of interpretability and modularity for further extension.

This paper introduces a modular, adaptable, vision-based grasping framework that can be leveraged to make robots learn to grasp known objects with a limited computational cost. Based on recent works in Quality-Diversity (QD) methods [17][18], the proposed framework builds repertoires of diverse grasping trajectories for a given robotic manipulator and a set of objects. At deployment time, a vision pipeline of open-source modules predicts the targeted object state (6DoF pose, including position and orientation). A trajectory is then selected and adapted relatively to the predicted object pose. Experiments conducted on two robotic platforms show that this approach can efficiently be applied to different grippers and robot arms. The presented pipeline will be made publicly

* equal contribution

† corresponding author

[1] Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique, ISIR, F-75005 Paris, France {helenon, huber, benamar, doncieux}@isir.upmc.fr

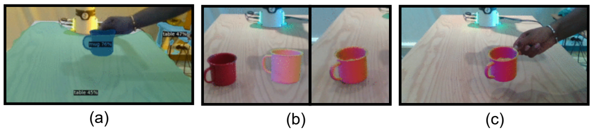

Fig. 2. **Object 6DoF pose detection pipeline.** (a) The scene is first segmented to isolate the targeted object using Detic [30]; (b) Megapose [31] does a 3d model matching to predict the 6DoF pose; (c) ICG [32] tracks the object pose to generalize the 6DoF tracking to any pose and allow retrial after failure.

available. More details can be found on the project website[1].

II. METHOD

Fig. 1 gives an overview of the proposed framework. It consists of a training step in simulation and a deployment step in the physical world. In the training step, a QD optimization method generates a set of grasping trajectories [17]. The deployment phase raises two key challenges: no off-the-shelf algorithms can robustly do the 6DoF object pose estimation, and the reach-and-grasps trajectories generated with QD are limited to a fixed initial object pose.

A. Training

Quality-Diversity methods are algorithms that optimize a set of diverse and high-performing solutions to a given problem [29]. Recent works show that those methods can be used to generate repertoires of diverse reach-and-grasp trajectories [17] that can successfully be transferred in the physical world [18]. Such an approach allows the generation of a diversity of trajectories that can fit a large variety of scenarios without new training iterations.

The training part is based on previous works in grasping with QD: the input is the 3D model of the considered robotic manipulator, as well as the 3D models of the targeted objects. Both are included in a simulated scene, on which a QD optimization method is applied to generate a repertoire of diverse and robust grasping trajectories A_s [17]. The most promising trajectories can be selected among the thousands of generated ones using dedicated quality criteria [18]. The output is, therefore, a set of repertoires containing hundreds of grasping trajectories for each of the targeted objects.

B. Deployment

The deployment part of the grasping module takes as input the data from an RGB-D camera, the name of a targeted object, and the skill repertoires containing the 3D model of the objects and the grasps.

6DoF pose estimation. The estimation of the object 6DoF pose X_{obj} is conducted in 3 steps (Fig. 2). Given an object name on a GUI, the targeted object is first localized on the RGB image using an open-vocabulary segmentation module [30]. The identified region is used to restrict the search space of a 6DoF pose estimation module called MegaPose [31]. MegaPose matches the object 3D model projection in the image with the current data acquired through the RGB-D camera. As soon as the predicted pose converges, a low processing tracking module [32] (running on CPU) is used to follow the object 6DoF trajectory in the RGB-D image, generalizing the detection to the whole robot field of view and allowing retrial after failure. It results in an accurate and fast estimation of the object's 6DoF pose.

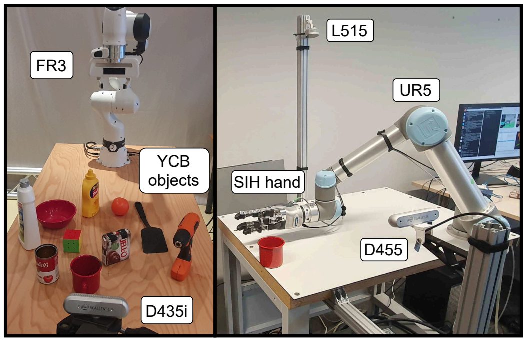

Fig. 3. **Experimental setups.** To demonstrate the framework flexibility to platforms, experiments have been conducted on an FR3 arm with a parallel gripper and on a UR5 arm with an SIH 5-fingers hand. The 3 RGB-D cameras have been indifferently used to demonstrate both hardware and point-of-view robustness. The 10 YCB objects [19] are used in both setups.

Trajectory adaptation. A grasping trajectory τ is then selected with respect to the addressed scenario (e.g., higher robustness, grasping a specific part of the object). In the experiments were exploited the best-performing grasps w.r.t. the fitness criterion proposed in [17]. The reach-and-grasp trajectory is described as a succession of end-effector states. However, those QD-generated trajectories are limited to the initial condition of the simulated scene. Generating trajectories for a new object pose would have a significant computational cost. We circumvent this limitation by rigidly adapting the trajectory τ to the object frame, making the trajectory valid for any object state in the operational space. The end-effector must reach the first step of the approach phase and then follow the path until the object is grasped. The approach trajectory is truncated if some states are not reachable by the manipulator. Trajectories leading to collisions are discarded. Details can be found in Appendix I. The resulting τ' adapted trajectory can be provided to a motion planner [33] to complete the grasp.

III. EXPERIMENTS

Robots and scene. To evaluate the proposed pipeline, experiments are carried out on a 6-DoF Universal robot (UR5) and a 7-DoF Franka Research 3 (FR3) (Fig. 3). The FR3 is equipped with a parallel 2-fingers gripper. The UR5 gripper is an SIH dexterous hand. Grasp learning and control of the SIH hand are made with synergies primitives (thumb-index, thumb-mid, thumb-index-mid, all-hand). ROS is used to orchestrate the different modules: robot and gripper control, the camera sensors, and perceptual modules. Each robot is mounted on a table modeled as a collision plane.

Sensing Experiments with the FR3 was conducted with a static Intel®Realsense™ Depth Camera D435i. For the UR5 and the dexterous hand, a Realsense D455 and a Realsense

[1] https://qdgrasp.github.io/

L515 are alternatively used to assess the robustness to the camera point-of-view. Cameras are fixed at various mounting positions (from the top, at 45°, from the side – Fig. 3). All cameras are hand-eye calibrated with an ArUco marker.

Hardware compute specifications Trajectory loading and transformations, 6-DoF tracking processing, and control of robots are made on a DELL laptop (a 12 cores Intel^R Core^{TM} i7-10850H). Deep learning perceptual modules are run on a remote desktop PC with a dedicated GPU (Graphical Processing Unit), an Nvidia TITAN X 12GB for the FR3, and an NVIDIA RTX 2080 for the UR5.

Dataset generation Grasping repertoires are generated on the Pybullet simulator [34]. ME-scs, a variant of MAP-Elites [35], is used to generate the grasps, as it appeared to be the most efficient QD method on this task [17]. The experiments are conducted on a dozen of YCB objects [19]. As the YCB objects' center of mass and inertia matrix are not correctly specified in the original dataset, we computed them by getting the average position of mesh vertices and by assuming that the objects' density was $1.5kg/m^3$.

Evaluating adaptation in simulation. To quantify the augmentation potential of the adapted trajectories, the trajectory adaptation was first simulated for three objects (mug, power drill, and pudding box) in the FR3 scene. The FR3 working space is defined as a box in front of it. This space is divided into equal-sized cells, defining different positions and orientations for the objects. Five trajectories are randomly sampled for each object and then adapted for each pose. For each pose, the number of successfully adapted trajectories is measured (i.e., the planner found a solution) – indicating the ability to generate a diversity of grasping at several positions in the working space.

Real world study. For each object, we randomly sampled grasps among the best-performing ones with respect to Huber et al. quality criterion [17], promoting diversity of object states. Each trajectory was tested for different object states. Overall, we collected about 300 trajectories.

For pose detection, we first give the name of the object to the open-vocabulary semantic module, which then feeds the 6-DoF pose estimation pipeline with the cropped RGB-D data. Visual ambiguities can lead to the wrong initial pose estimation depending on initialization and the current view of the object. To mitigate this limitation, objects are oriented so that several faces are in the camera's field of view, and the initial pose estimation is reinitialized until convergence. Then, by leveraging the 6-DoF tracker, we placed the object at the target location. A GUI was developed to monitor the whole process[2].

IV. RESULTS & DISCUSSION

Evaluating adaptation in simulation. Fig. 4 shows the results obtained in simulation for the FR3 robot with the 2-finger gripper after applying a perturbation to the object pose, either purely translated (upper row) or both translated and rotated (lower row). Most randomly sampled trajectories can successfully be adapted to grasp the object. The heatmap shows that the failures occur more frequently when the object is near the limits of the robot's reachable space: being too close results in self-collisions while being too far makes the robot near its joint singularities. Translating the object is likely to result in successful adaptations. It can be noticed that variations along the z axis can more frequently lead to a failure. Similarly, rotating the object can lead to an invalid grasp because of self-occlusions (e.g. trying to grasp the hidden handle of a mug) or object poses that are outside of the collision-free and reachable space (e.g. a robot cannot grasp a cup by inserting fingers in the containing part if the object is flipped on the surface). This limitation can be addressed by regrasping or trajectory selection.

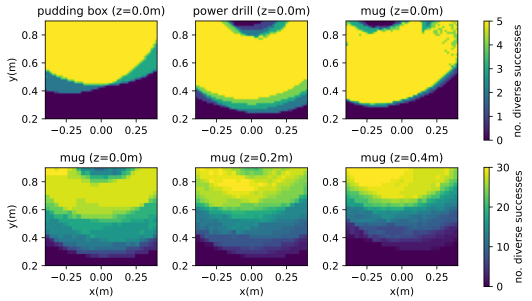

Fig. 4. **Adaptation of diverse trajectories.** Results obtained in simulation on the FR3 robot by randomly picking 5 reach-and-grasp trajectories from a learned repertoire and different object poses. (Upper row): 2500 positions in the xy grid at $z = 0$ and for a fixed orientation. (Lower row): 625 positions and 6 orientations per position - 2 rotations around the y axis and 3 around the z axis. The maximum number of transferable trajectories per pose is then 5x6=30. The rigid transform adaptation method generalizes the grasps to the whole operational space. Failures occur when rotations prevent some grasps (e.g. collisions or reachability constraints)

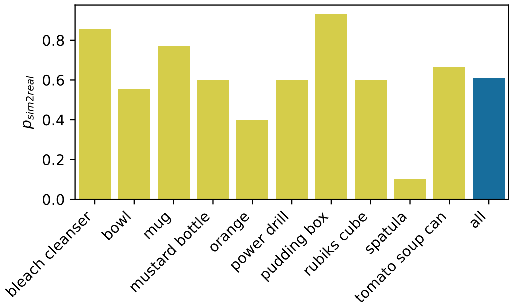

Fig. 5. **Successful sim2real transfer ratios on the FR3 robot.**Results are similar to those obtained in similar experimental conditions with objects fixed at simulated pose [18], validating the proposed adaptation framework.

Real world study. Fig. 5 shows the success rate of adapted grasping trajectories for all the tested setups. Most objects, even complex ones such as the power drill, show a grasp success rate of over 50%. The average success ratio is around 60%, which is similar to the transferability ratio obtained in the same experimental setups, except from the object pose, which matches the one in the simulation [18]. Note that the most challenging objects are the bowl, the orange, and the spatula, primarily because of vision failure. Fig. 6 illustrates one example of failure modes that were observed

[2] https://cloud.isir.upmc.fr/s/sqXpAtrrkSiM3SX

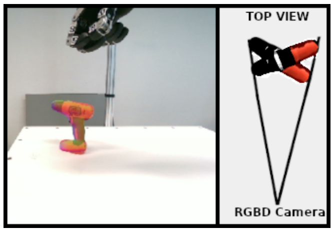

Fig. 6. **Object pose ambiguities.** Ambiguities appear depending on the view, the object, and the distance to the camera. Here, the power drill is too far from the camera. Depth measurements cannot solve ambiguities. The predicted orientation is wrong over the z-axis.

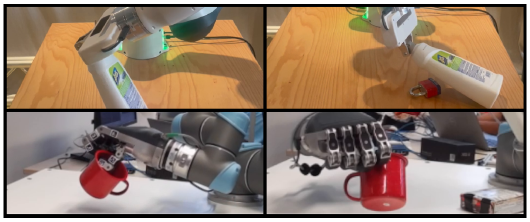

Fig. 7. **Examples of grasp diversity and trajectory adaptation.** The same trajectory adapted to two different object states for the panda parallel gripper (top) and the SIH Schunk hand (bottom).

for the power drill. As we use a single view, the partial point cloud is not always enough to disambiguate the pose. The depth measurements might also be too noisy, especially for objects small or far from the camera, making the 6-DoF pose prediction modules fall into local optima.

Those experiments validate the proposed approach, as the QD-generated trajectories are well generalized to the whole operational space (Fig. 7) with results comparable to those obtained at fixed object pose [17]. Moreover, the diversity of the generated repertoire is preserved, allowing the exploitation of the proposed framework in several scenarios without further training iterations (Fig. 8).

Cross-plateform transferability. A key component of the proposed framework is how easy it is to transfer to new platforms. While the recent works on robotic learning suggest that it might be possible to exploit foundation models to do cross-platform transferability efficiently [2], the learning methods that exploit such architectures require a tremendous amount of computation [8]. The approach proposed in the present paper generates the grasping repertoires offline and adapts them without additional cost, while the modular architecture eases human supervision of the grasping process. Exploring how the repertoires can be zero-shot transferred between robots and scenarios is also a promising perspective.

Computational cost. The training part requires building a repertoire per gripper and object once. It roughly takes 20 minutes on a 40-core CPU to generate hundreds of diverse trajectories. No further training is required at inference time. The trajectory adaptation and the variety of generated grasps provide adaptation capabilities to the robots. At inference time, Detic and MegaPose are the most computationally greedy modules. However, it works on an affordable consumer GPUs. Overall, the computational needs are far from the large infrastructure required by large end-to-end controllers [7][8].

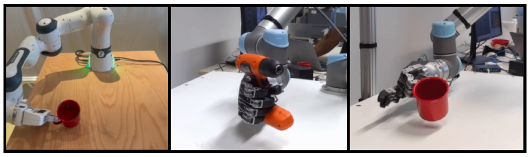

Fig. 8. **Example of diverse grasps.** The diversity produced by the QD method is preserved in the proposed framework so that many different tasks can be completed after having deployed the grasping trajectory.

Limitations. The most important bottleneck for adaptation to a new scene is the vision pipeline, as some of the exploited submodules have limitations. In our experience, the weakest part of the vision pipeline is the 6-DoF pose estimation. MegaPose and ICG are more robust on some objects (e.g. the mug) than others (e.g. the spatula, orange). They struggle in dense or noisy scenes. It can also require a few manual iterations to converge to a valid pose. Lastly, the 3D model matching part can make prohibitive errors in the object poses, especially for objects with ambiguous shapes and far from the camera (Fig. 6). The proposed modular framework can, however, easily be updated with a more robust 6-DoF pose estimator and tracker. This matter is a key challenge in the computer vision community [36].

While easily generalizing to different robotic platforms, the proposed framework requires a 3D model of the targeted object. Nevertheless, the ability to grasp ”known” objects in diverse manners opens many research paths for open-ended robotics. This limitation can be addressed by integrating vision-based surface reconstruction of unknown objects [37].

V. CONCLUSIONS

This paper proposes a framework to build a plug-and-play vision-based grasping module. It can easily be adapted to different robotic platforms and allows the robot to grasp objects in a diverse manner robustly. In future work, we plan to use dedicated quality metrics to improve the sim2real transfer ratio [18] and extend to dynamic adaptations of grasps. The pipeline will also be transferred to a mobile robotic manipulator within human-in-the-loop processes. We believe this modular framework to be an affordable alternative to the computationally greedy foundation-model-based approaches and a promising path to make different kinds of mobile manipulators interact with objects in the near future.

ACKNOWLEDGMENT

Ministry of Education and Research (BMBF) (01IS21080), the French Agence Nationale de la Recherche (ANR) (ANR-21-FAI1-0004) (Learn2Grasp), the European Commission’s Horizon Europe Framework Programme under grant No 101070381 and from the European Union’s Horizon Europe Framework Programme under grant agreement No 101070596. This work used HPC resources from GENCI-IDRIS (Grant 20XX-AD011014320). Authors deeply thank Pr. Sven Behnke and the members of the AIS lab of Bonn for their warm welcome and support with the SIH setup.

References

[1] Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., ...& Zeng, A. (2023, May). Code as policies: Language model programs for embodied control. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 9493-9500). IEEE.

[2] Team, O. M. (2023). Octo: An Open-Source Generalist Robot Policy.

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

[4] Urain, J., Funk, N., Peters, J., & Chalvatzaki, G. (2023, May). Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. ICRA 2023.

[5] Barad, K. R., Orsula, A., Richard, A., Dentler, J., Olivares-Mendez, M., & Martinez, C. (2023). GraspLDM: Generative 6-DoF Grasp Synthesis using Latent Diffusion Models. arXiv preprint.

[6] Chen, H., Xu, B., & Leutenegger, S. (2024). FuncGrasp: Learning Object-Centric Neural Grasp Functions from Single Annotated Example Object. arXiv preprint.

[7] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., ...& Zitkovich, B. (2022). Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817.

[8] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., ...& Zitkovich, B. (2023). Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15

[9] Thompson, N. C., Greenewald, K., Lee, K.,& Manso, G. F. (2020). The computational limits of deep learning. arXiv preprint arXiv:2007.05558.

[10] Newbury, R., Gu, M., Chumbley, L., Mousavian, A., Eppner, C., Leitner, J., ...& Cosgun, A. (2023). Deep learning approaches to grasp synthesis: A review. IEEE Transactions on Robotics.

[11] Sahbani, A., El-Khoury, S.,& Bidaud, P. (2012). An overview of 3D object grasp synthesis algorithms. Robotics and Autonomous Systems

[12] Bottarel, F., Altobelli, A., Pattacini, U., & Natale, L. (2023). GRASPA-fying the Panda: Easily Deployable, Fully Reproducible Benchmarking of Grasp Planning Algorithms. IEEE Robotics & Automation Magazine.

[13] Hodson, R. (2018). A gripping problem: designing machines that can grasp and manipulate objects with anything approaching human levels of dexterity is first on the to-do list for robotics. Nature Spotlight: Robotics.

[14] Yang, J., Tan, W., Jin, C., Liu, B., Fu, J., Song, R., Wang, L. (2023). Pave the Way to Grasp Anything: Transferring Foundation Models for Universal Pick-Place Robots. arXiv preprint arXiv:2306.05716.

[15] Fang, H. S., Wang, C., Gou, M., Lu, C. (2020). Graspnet-1billion: A large-scale benchmark for general object grasping. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition

[16] Qin, Y., Su, H., Wang, X. (2022). From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. IEEE Robotics and Automation Letters, 7(4), 10873-10881.

[17] Huber, J., Hélénon, F., Coninx, M., Ben Amar, F., Doncieux, S. (2023). Quality Diversity under Sparse Reward and Sparse Interaction: Application to Grasping in Robotics. arXiv:2308.05483

[18] Huber, J., Hélénon, F., Watrelot, H., Amar, F. B., & Doncieux, S. (2024). Domain Randomization for Sim2real Transfer of Automatically Generated Grasping Datasets. ICRA'24

[19] Calli, B., Walsman, A., Singh, A., Srinivasa, S., Abbeel, P., Dollar, A. M. (2015). Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. arXiv preprint arXiv:1502.03143.

[20] Horaud, R., Dornaika, F., & Espiau, B. (1998). Visually guided object grasping. ieee Transactions on Robotics and Automation

[21] T. Weng, D. Held, F. Meier, and M. Mukadam. 2023. Neural Grasp Distance Fields for Robot Manipulation. IEEE International Conference on Robotics and Automation (ICRA) (2023)

[22] Urain, J., Funk, N., Peters, J., & Chalvatzaki, G. (2023, May). Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE.

[23] Yi, S. J., Lee, C. Y., Yi, J., Cho, H., Park, Y., Zhang, B. T., ... & Suh, I. H. Team Tidyboy RoboCup@ Home Domestic Standard Platform League Team Description Paper.

[24] Eppner, C., Höfer, S., Jonschkowski, R., Martín-Martín, R., Sieverling, A., Wall, V., & Brock, O. (2016, June). Lessons from the amazon picking challenge: Four aspects of building robotic systems. In Robotics: science and systems (pp. 4831-4835).

[25] Matamoros, M., Viktor, S. E. I. B., & Paulus, D. (2019, April). Trends, challenges and adopted strategies in robocup@ home. In 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) (pp. 1-6). IEEE.

[26] https://pal-robotics.com/robots/tiago/

[27] http://wiki.ros.org/Robots/TIAGo/Tutorials/MoveIt/Pick_place

[28] https://enchanted.tools/

[29] Cully, A., Mouret, J. B., & Doncieux, S. (2022, July). Quality-diversity optimisation. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 864-889).

[30] Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., & Misra, I. (2022, October). Detecting twenty-thousand classes using image-level supervision. In European Conference on Computer Vision (pp. 350-368). Cham: Springer Nature Switzerland.

[31] Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., ... & Sivic, J. (2022). Megapose: 6d pose estimation of novel objects via render & compare. arXiv preprint arXiv:2212.06870.

[32] Stoiber, M., Sundermeyer, M., & Triebel, R. (2022). Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[33] Coleman, D., Sucan, I., Chitta, S., & Correll, N. (2014). Reducing the barrier to entry of complex robotic software: a moveit! case study. arXiv preprint arXiv:1404.3785.

[34] Coumans, E., Bai, Y. (2016). Pybullet, a python module for physics simulation for games, robotics and machine learning.

[35] Mouret, J. B., & Clune, J. (2015). Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909.

[36] Guan, J., Hao, Y., Wu, Q., Li, S., & Fang, Y. (2024). A Survey of 6DoF Object Pose Estimation Methods for Different Application Scenarios. Sensors, 24(4), 1076.

[37] Xia, H., Fu, Y., Liu, S., & Wang, X. (2024). RGBD Objects in the Wild: Scaling Real-World 3D Object Learning from RGB-D Videos. arXiv preprint arXiv:2401.12592.

[38] Chen, Y. L., Cai, Y. R., & Cheng, M. Y. (2023). Vision-Based Robotic Object Grasping—A Deep Reinforcement Learning Approach. Machines, 11(2), 275.

[39] Zhou, W., & Held, D. (2023, March). Learning to grasp the ungraspable with emergent extrinsic dexterity. In Conference on Robot Learning (pp. 150-160). PMLR.

[40] Sefat, A. M., Angleraud, A., Rahtu, E., & Pieters, R. (2022, August). SingleDemoGrasp: Learning to Grasp From a Single Image Demonstration. In 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) (pp. 390-396). IEEE.

[41] Wang, P., Manhardt, F., Minciullo, L., Garattoni, L., Meier, S., Navab, N., & Busam, B. (2021, September). DemoGrasp: Few-shot learning for robotic grasping with human demonstration. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5733-5740). IEEE.

[42] A. Mousavian, C. Eppner, and D. Fox, "6-DoF graspnet: Variational grasp generation for object manipulation," in International Conference on Computer Vision, 2019.

[43] Roa, M. A., Argus, M. J., Leidner, D., Borst, C., & Hirzinger, G. (2012, May). Power grasp planning for anthropomorphic robot hands. In 2012 IEEE International Conference on Robotics and Automation (pp. 563-569). IEEE.

[44] R. Grimm, M. Grotz, S. Ottenhaus, and T. Asfour, "Vision-based robotic pushing and grasping for stone sample collection under computing resource constraints," in IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 0–0.

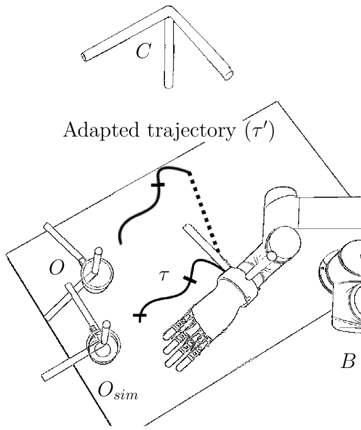

Fig. 9. **Notations and adaptation principle.** The robot base frame B and the world frame W are assumed equal. The robot has to grasp a mug (frame O) with a pose estimated by a RGB-D camera (frame C) and perception modules. The trajectory τ has been generated in simulation with the object at O_{sim}. The path followed by the end-effector is adapted from one pose to another, resulting in the trajectory τ'.

Supplementary Materials

Appendix I
Trajectory adaptation

Fig. 9 shows the used notations. Let B be the robot frame and W be the world frame. Here, we assume that $B = W$, as the robots considered in the experiments are fixed manipulators. Let O_{sim} be the object frame at initial conditions in the deterministic simulation, and O be the actual frame associated with the targeted object in the physical world (indifferently noted X_{obj}).

Let $\tau \in \mathbb{R}^{m \times n}$ be a selected trajectory, where m is the number of values to express a state pose, and n is the number of considered time steps. State-of-the-art QD methods generate open-loop trajectories conditioned on a specific object pose (O_{sim}). The trajectory is expressed as a succession of end-effector Cartesian positions and Euler orientations ($m = 6$). A QD method thus generates a set of trajectories $A_s = \{\tau_{i \in \mathbb{N}^{+*}}\}$. Each trajectory can be expressed as a sequence of end effector state through forward kinematics, such that $\tau = \left\{X_{i \in [0,\ldots,n-1]}\right\}$.

The trajectories are projected in the object frame (O) to generalize the generated repertoire to the whole operational space. Each repertoire can thus be interpreted as bundles of trajectories that can be reached to grasp the object in a certain manner. Let ${}^{W}e_s$ be the end effector state in homogeneous coordinates in W for a given 6D state X generated in simulation. The adapted state on the real object ${}^{W}e_r$ is defined such that:

$$ {}^{O_{sim}}e_s = {}^{O}e_r \tag{1} $$

The adapted state in W can be computed as follow:

$$
\begin{aligned}
{}^{W}e_r &= {}^{W}_{C}H \; {}^{C}_{O}H \; {}^{O}e_r \\
&= {}^{W}_{C}H \; {}^{C}_{O}H \; {}^{O_{sim}}e_s \\
{}^{W}e_r &= {}^{C}_{W}H^{-1} \; {}^{O}_{C}H^{-1} \; {}^{O_{sim}}_{W}H \; {}^{W}e_s
\end{aligned} \tag{2}
$$

where transformation matrices ${}^{b}_{a}H$ is the transformation matrix from a to b. The equation (2) allows to compute the adapted trajectory τ' from τ, considering that ${}^{O}_{C}H$ is obtained using the proposed vision pipeline, ${}^{C}_{W}H$ comes from the camera calibration, and ${}^{O_{sim}}_{W}H$ is provided by the simulation. Each trajectory is then filtered using the following criterion:

$$ f_c(\tau') = f_{IK}(\tau') \wedge f_{collision}(\tau') \tag{3} $$

where $\wedge$ is the logical and, $f_{IK} : \mathbb{R}^{m \times n} \to \{0, 1\}$ assesses that the trajectory is kinematically feasible with limited jump in joint space, and $f_{collision} : \mathbb{R}^{m \times n} \to \{0, 1\}$ assesses that no collisions happens between the robot and the environment or itself. The resulting τ' adapted trajectory can be deployed on the real platform using a motion planner [33].