Supplementary Material for Parallel Monte Carlo Tree Search with Batched Rigid-body Simulations for Speeding up Long-horizon Episodic Robot Planning

Baichuan Huang Abdeslam Boularias Jingjin Yu

***Abstract*— We propose a novel Parallel Monte Carlo tree search with Batched Simulations (PMBS) algorithm for accelerating long-horizon, episodic robotic planning tasks. Monte Carlo tree search (MCTS) is an effective heuristic search algorithm for solving episodic decision-making problems whose underlying search spaces are expansive. Leveraging a GPU-based large-scale simulator, PMBS introduces massive parallelism into MCTS for solving planning tasks through the batched execution of a large number of concurrent simulations, which allows for more efficient and accurate evaluations of the expected cost-to-go over large action spaces. When applied to the challenging manipulation tasks of object retrieval from clutter, PMBS achieves a speedup of over $30\times$ with an improved solution quality, in comparison to a serial MCTS implementation. We show that PMBS can be directly applied to a real robot hardware with negligible sim-to-real differences. Supplementary material, including video, can be found at `https://github.com/arc-l/pmbs`.**

I. Appendix

A. Grasp Classifier Implementation Details

For our implementation of the grasp classifier (GC), we used Isaac Gym to collect grasp training data. Random objects are first sampled on the workspace, and then we discretize the workspace into a grid, where each point is the (x, y) of a grasp action a^g. We also discretize rotation into K angles uniformly. All robots in simulator will pick one grasp action and check the distance between two fingers as a signal of successful grasping. For each depth image, it is associated with hundreds of grasps. If the target can be grasped in at least n attempts, then the label is 1, and 0 otherwise (n = 5). We used two days of generating 20000 training data (a depth image focus centered on the target object and a label of can it be grasped) without human annotation. It is evaluated on test data with 93.45% accuracy if the R_c^* equals 0.7. The batch size is 256, learning rate is 0.1, epochs is 90, momentum is 0.9. We have successfully reduced the grasp evaluation time from 0.26 to 0.003 per image, making the parallel MCTS possible.

B. Grasp Network Implementation Details

For deciding whether to perform further push actions or to make an attempt to retrieve the target object, we resort to a *grasp network* (GN) that is fast and amenable to parallelization. GN is based on *fully convolutional networks* (FCNs) [1], [2] and customized to estimate the grasp probability for the target object [3], [4]. It takes an RGB-D image o_t as input and outputs dense pixel-wise values $P(o_t) \in [0, 1]^{H\times W\times K}$. H and W are the height and width of the o_t. To account the gripper orientation, we discretize θ of a^g into $K = 16$ angles, in multiples of (22.5°), so o_t has been rotated K times. GN presented in [2] is trained to estimate the grasp success rate for all objects. Further, binary mask of target object M is imposed using Mask R-CNN [5], to truncate the values as $P_m(o_t) = P(o_t) \cap M(o_t)$. In proposed system, we only interested the highest grasp probability. If $\max_{h,w,k} P_m(o_t)$ is greater than the preset threshold P^* (it is 0.75 in our case), then, the robot should grasp at location h, w with k as orientation of gripper. The backbone of GN is ResNet-18 FPN [6], [7], with convolution layers and bilinear-upsampling layers as described in [3], [4]. We used the pre-trained Grasp Network from [4]. The Grasp Network is a plug-in component for the system, can be replaced by other advanced methods as if a grasp probability and grasp action can be provided.

References

[1] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[2] B. Huang, S. D. Han, A. Boularias, and J. Yu, "Dipn: Deep interaction prediction network with application to clutter removal," in *2021 IEEE International Conference on Robotics and Automation*, 2021, pp. 4694–4701.

[3] B. Huang, S. D. Han, J. Yu, and A. Boularias, "Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement," *IEEE Robotics and Automation Letters*, vol. 7(1), pp. 231–238, 2022.

[4] B. Huang, T. Guo, A. Boularias, and J. Yu, "Interleaving monte carlo tree search and self-supervised learning for object retrieval in clutter," *arXiv preprint arXiv:2202.01426*, 2022.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144