

Supplementary Material for Learning to Perceive 3D Objects by Poking

I. POKING POLICY

The z axes of the world and the robot coordinate systems are perpendicular to the plane and the y axes are perpendicular to the line connecting the camera and the center of the world coordinate system. The line p_1p_2 is one of the longer side of the bounding box B that rotates the object clockwise when being poked. The r in line 8 and 11 in Algorithm 1 represents the angle between the poking direction and the longer side of the bounding box B . Instead of poking in the perpendicular direction ($r = 90^\circ$), we empirically add or subtract 15° to bring more translation for the object. $1/3$? The z values of the start and end points are defined by keeping the distance between the end of the gripper and the plane a constant value. The computed start and end points are examined before execution to avoid expected collision between the gripper and the object. After each iteration of poking, the oriented bounding box B is updated via point cloud registration on the foreground, which is defined as the set of pixels whose optical flow norm is greater than a pre-defined threshold. The point cloud registration is achieved by scene flow estimation and refined by ICP. The scene flow is obtained from the back-projected optical flow which is estimated with a trained optical flow estimation network [1]. The poking order is determined by sorting the distance between the center of the object proposal and the right-down corner of the plane increasingly.

II. DATA COLLECTION

The real-world setup is composed of a UR5 robot with a cylinder pusher tool and an RGB-D Kinect camera visualized in Fig. 1. The relative pose between the camera and the base of the robot arm T_{ca} is computed as follows:

$$T_{ca} = T_{gc}^{-1} T_{ga}, \quad (1)$$

where T_{gc} is the relative pose between the gripper and the camera and T_{ga} is the relative pose between the gripper and the base of the robot arm. In practice, T_{ga} is read from the Robot Operating System (ROS). T_{gc} is obtained by aligning the model of the gripper and an RGB-D image. The alignment is achieved via differentiable rendering. Representing the 3D model of the gripper as G and an RGB-D image with the manually-annotated segmentation mask of the gripper as M , T_{gc} is obtained by optimizing the following objective function:

$$\mathcal{L} = \sum (R(G) - M)^2, \quad (2)$$



Fig. 1. **The real-world setup.** Poking is achieved via a UR5 robot and the RGB-D video is recorded by a Kinect camera.

where R is a differentiable neural renderer [2] that produce the mask of the 3D object.

After the optimization, we manually tune the distance between the gripper and the camera to align the model of the gripper and the back-projected point cloud for more precise calibration.

III. SUPPLEMENTARY VIDEO

In the supplementary video, we first describe the task, then provide the recorded video during the poking process, and finally visualize the application of object pose estimation and robotics grasping to demonstrate the applicability of our method in real-world scenarios.

REFERENCES

- [1] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” in *European conference on computer vision*. Springer, 2020, pp. 402–419.
- [2] H. Kato, Y. Ushiku, and T. Harada, “Neural 3d mesh renderer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3907–3916.