Teleoperated Robot Grasping in Virtual Reality Spaces

Jiaheng Hu[1], David Watkins[1], and Peter Allen[1]

***Abstract*— Despite recent advancement in virtual reality technology, teleoperating a high DoF robot to complete dexterous tasks in cluttered scenes remains difficult. In this work, we propose a system that allows the user to teleoperate a Fetch robot to perform grasping in an easy and intuitive way, through exploiting the rich environment information provided by the virtual reality space. Our system has the benefit of easy transferability to different robots and different tasks, and can be used without any expert knowledge. We tested the system on a real fetch robot, and a video demonstrating the effectiveness of our system can be seen here: `https://jiahenghu.github.io/research/vr_teleop/`**

I. INTRODUCTION

Robot teleoperation is a useful tool for handling tasks that are hard to be completed autonomously by robots, such as the DARPA Robotics Challenge [1]. Among different teleoperation methods, virtual reality based methods stand out as they provide an intuitive way for the user to examine the state of the robot and send commands [2][3][4][5]. In spite of this, teleoperating robots with high degree of freedom to complete dexterous tasks such as grasping remains difficult, mainly due to the following reasons: 1. The high DoF of the robot makes it hard to create 1-1 mapping from the controller inputs to robot actions. 2. Different robots have different structures, making it difficult to switch between them. 3. Even given a 1-1 mapping between the controller inputs and robot actions, teleoperating a high DoF robot in cluttered environment would still be hard as it is usually unrealistic for the user to precisely predict the forward dynamics of the robot.

In this work, we extend upon Ros Reality [3] and create a robot teleoperation system that allows the user to easily and intuitively control a 13 DoF fetch robot to grasp objects in an unstructured environment. Instead of mapping controller inputs to robot actions, we only convey the user's intention of which object to grasp to the robot through utilizing the rich information provided by the virtual reality space. Thus, the user only interacts with the environment, and the robot plans the low-level controlling commands automatically, making our system easily extendable to different robots, and intelligent enough to handle dexterous grasping tasks.

II. METHODS

The general workflow of our system can be summarized by Algorithm 1. During each controlling cycle, the robot takes a pointcloud scan of the environment and produces a segmented pointcloud of objects detected in the pointcloud

[1]Computer Science Department, Columbia University, New York, NY 10027, USA {`jh3916,djw2146`}`@columbia.edu`, `allen@cs.columbia.edu`

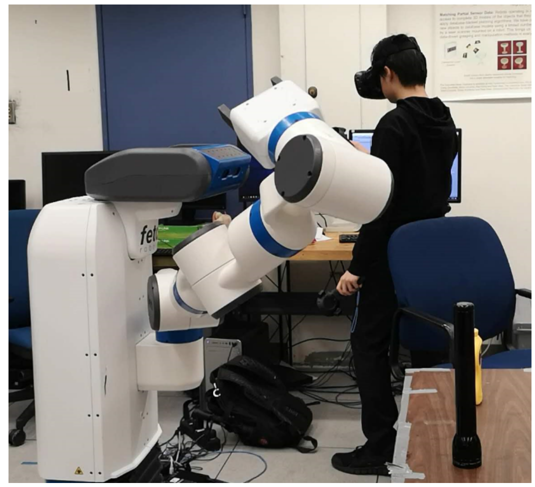

Fig. 1. The Hardware Setup: the user is holding two vive controllers and wearing a vive headset, and the robot is facing a table with objects to grasp

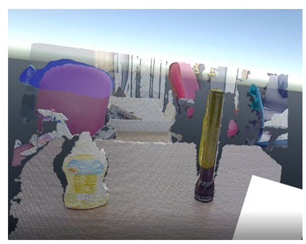

Fig. 2. Scene Segmentation

Fig. 3. Object Selection

Fig. 4. Grasping the Object

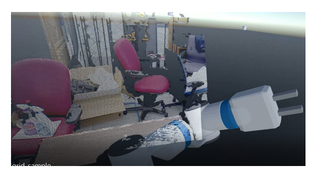

Fig. 5. Lifting up the Object

(Fig 2). The system then projects the segmented poincloud into the virtual reality space and waits for the user to select a certain mesh in the scene by pointing a virtual laser at it (Fig 3). Next, the partial pointcloud of the selected is shape-completed into a 3D mesh, and used to generate suitable grasps. A grasp is selected and the robot executes the grasp (Fig 4) and picks up the object (Fig 5).

A. *Virtual Reality Space*

The Virtual Reality portion of this project is an extension of Ros Reality[3], through which users can visualize a robot and the surrounding scene using a HTC Vive headset. Two

Algorithm 1: Teleoperated Grasping

while *robot active* **do**
  Scan Scene;
  Scene Segmentation;
  if *User Selected A Certain Object* **then**
    Complete the Mesh of the Given Object;
    Plan Grasp;
    Execute Grasp;
  else
    Continue;
  end
end

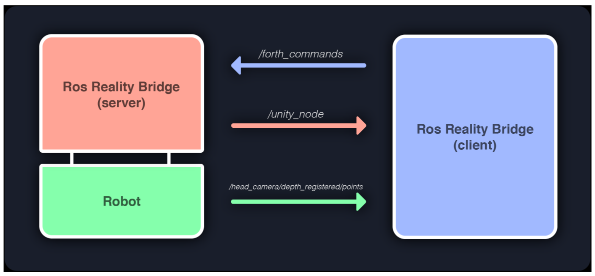

Fig. 6. Modified Ros Reality Scheme

additional hand controllers are used to send commands to the robot. The setup is shown in Fig. 1, with the user wearing the headset and holding the controllers. In order for the the system to be easily transferable to different robots, we modified Ros Reality to be more modular, shown in Fig. 6. Point clouds from the robot's depth camera are broadcast and rendered as meshes in the scene as well to allow the user to see what the robot sees. Using the Vive controllers, a user can control all aspects of the robot. More specifically, controller input is sent over the ROS topic /forth_commands, which the server interprets to command the robot.

To improve pointcloud transmission speed and thus reduce delay, we made further modifications to the original system including: 1. Changing pointcloud encoding to BSON encoding; 2. Limiting the pointcloud update frequency; 3. Changing the web socket protocol to .NET protocol; 4. Setting up direct connection between robot and the client. With these modications, we were able to reduce pointcloud delay from 10 seconds to less than 1 second.

B. Scene Segmentation

Our scene segmentation algorithm is based on [6]. Our program takes in an RGB-D pointcloud and returns a semantic segmentation of the input pointcloud. We further utilized domain randomization [7] to reduce noise in the generated semantic label.

C. Shape Completion

The shape completion program we used is a follow-up from our previous work [8][9]. We trained a convolution neural network (CNN) to predict the 3D mesh of an object given its partial pointcloud obtained from scene segmentation. A CNN is preferred over methods such as Gaussian Process Implicit Surface (GPIS) Completion [10] due to its faster run time, which is very important in our teleoperation loop. To further improve run speed, we set up a back-end shape completion server running over ros service so the front end doesn't need to run with GPU.

D. Grasp Planning

We used GraspIt [11] for grasp planning. The program takes in a completed mesh and returns a proper grasp. Due to time concerns, we used grid sampling instead of simulated annealing for grasp sampling, so that we are only sampling around the principal axis of the given mesh. This allows us to increase sampling efficiency by an order of magnitude of 10 comparing to simulated annealing and correspondingly reduce run time from around 50 seconds to less than 3 seconds.

III. CONCLUSIONS

We developed a system that is capable of grasping and navigation tasks through teleoperation and can be seamlessly transferred to additional robotic agents. It is easy and intuitive to use, and can be further extended to serve as a powerful tool for collecting data under a user monitored fashion, that could be useful in fields such as computer vision and robot learning.

REFERENCES

[1] C. Dellin, K. Strabala, G. C. Haynes, D. Stager, and S. Srinivasa, "Guided manipulation planning at the darpa robotics challenge trials," in *Proceedings of 2014 International Symposium on Experimental Robotics (ISER 2014)*, June 2014.

[2] J. Byrn, S. Schluender, C. Divino, J. Conrad, B. Gurland, E. Shlasko, and A. Szold, "Three-dimensional imaging improves surgical performance for both novice and experienced operators using the da vinci robot system," *American journal of surgery*, vol. 193, pp. 519–22, 05 2007.

[3] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, "Ros reality: A virtual reality framework using consumer-grade hardware for ros-enabled robots," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 07 2018.

[4] J. I. Lipton, A. J. Fay, and D. Rus, "Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing," 2017.

[5] M. Mallwitz, N. Will, J. Teiwes, and E. A. Kirchner, "The capio active upper body exoskeleton and its application for teleoperation," in *Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation. ESA/Estec Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA-2015)*. ESA, 2015.

[6] Y. Xiang and D. Fox, "Da-rnn: Semantic mapping with data associated recurrent neural networks," 2017.

[7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," 2017.

[8] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," 2016.

[9] D. Watkins-Valls, J. Varley, and P. Allen, "Multi-modal geometric learning for grasping and manipulation," 2018.

[10] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," 01 2007.

[11] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec 2004.