STUDENT SUMMER INTERNSHIP TECHNICAL REPORT

Performance Evaluation of Augmented Teleoperation of Contact Manipulation Tasks

DOE-FIU SCIENCE & TECHNOLOGY WORKFORCE DEVELOPMENT PROGRAM

Date submitted:

September 14, 2018

Principal Investigators:

Anibal E. Morales (DOE Fellow Student)
Florida International University

Dr. Young Soo Park, Mentor
Argonne National Laboratory

Florida International University Program Director:

Leonel Lagos Ph.D., PMP®

Submitted to:

U.S. Department of Energy
Office of Environmental Management
Under Cooperative Agreement # DE-EM0000598

FIU | Applied Research Center
Florida International University

ABSTRACT

Argonne National Laboratory is researching an enhanced telerobotic concept known as augmented teleoperation which includes augmented virtual fixtures for precise and dexterous manipulation. The proposed technology would incorporate a human-robot interface that can assist personnel with deactivation and decommissioning activities in areas with high radiation. The enhanced telerobotics concept has been implemented on a dual-arm robot system, known as Baxter, by integrating the robotic operating system (ROS) with 3D sensors and a visual-haptic operator interface. However, improved capabilities for precisely placing the virtual fixtures is needed before the technology can be operationally deployed. To this end, the objectives for the summer internship included preparing an experimental setup for contact manipulation, test operation, evaluation of performance under various test conditions, as well as the development of a feasible control/operation strategy, which required hardware fabrication and setup, and camera calibration through the modification and implementation of the ROS.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Across the United States, the Department of Energy Office of Environmental Management (DOE EM) manages a variety of nuclear facilities that are used for multiple objectives, including nuclear weapons stockpile maintenance, production of plutonium-238 oxide fuel for space exploration, and even production of radiopharmaceuticals for the medical industry [1]. Shielded radiation containment chambers, known as hot cells, are used at many of these facilities to safely handle radioactive materials.

DOE EM has been given the mission to remediate the nation's Cold War environmental legacy resulting from five decades of nuclear weapons production and government-sponsored nuclear energy research. This effort includes the deactivation and decommissioning (D&D) of thousands of radioactively contaminated facilities to reduce the imminent health and environmental hazards associated with radioisotope particles. Some locations, such as the Plutonium Fuel Form (PuFF) Facility in building 235-F at the Savannah River Site (SRS), were used by DOE and the National Aeronautics and Space Administration (NASA) to design and manufacture plutonium spheres and pellets for deep-space mission probes [2].

SRS 235-F has been shown to contain as much as one kilogram of plutonium oxide in the form of very small particles, making it one of DOE's most hazardous radioactive facilities to decontaminate, dismantle, package, and remove [3]. Conversely, other locations, such as the Fukushima Daiichi Nuclear Power Station in Japan, have experienced catastrophes where the deployment of robotics and remote systems have been critical to achieving the needed response.

Telerobotic systems were first invented for deployment at such nuclear facilities to assist human workers with managing D&D tasks from a safe location, removed from highly radioactive areas. Telerobotics is an area of robotics that is focused on the control of semi-autonomous robots from a distance. The use of telerobotic systems for nuclear waste cleanup has had multiple obstacles and limitations due to the high-radiation environments and heavy, yet dexterous contact manipulation. Thus, telerobotic research incorporates virtual-reality (VR) and augmented-reality (AR) to address these technical challenges.

The research presented here focuses on the camera calibration technique used to ensure an overlap of the real environment and a simulated environment captured by cameras on a robot. This camera calibration is crucial for the overall structure of an enhanced teleoperator interface, which includes a multi-nodal AR, virtual fixtures, and system integration on the robotic operating system (ROS). The camera calibration will permit the enhanced telerobotic operation system to achieve dexterous telerobotic manipulation capabilities with simple and rugged robots to be deployed in complex and hazardous remote environments for cleanup and D&D applications.

2. EXECUTIVE SUMMARY

This research work has been supported by the DOE-FIU Science & Technology Workforce Initiative, an innovative program developed by the US Department of Energy's Environmental Management (DOE-EM) and Florida International University's Applied Research Center (FIU-ARC). During the summer of 2018, a DOE Fellow intern Anibal E. Morales spent 10 weeks doing a summer internship at Argonne National Laboratory in the greater Chicago area, Illinois, under the supervision and guidance of Dr. Young Soo Park, Program Lead, Robotics and Remote Systems. The intern's project was initiated on May 21, 2017 and continued through July 27, 2017 with the objective of preparing an experimental setup for contact manipulation, test operation, and evaluation of performance under various test conditions, as well as developing a feasible control/operation strategy, which required hardware fabrication and setup, and camera calibration through the modification and implementation of the robot operating system (ROS).

3. RESEARCH DESCRIPTION

System Overview

Figure 1 shows a system overview of the Baxter robot in teleoperation. Here the architecture of the augmented telerobotic operation system, guided by virtual fixtures, can be seen. The user uses a haptic-feedback hand controller to operate the Baxter robot, which can be seen in Figure 2. Baxter is a unilaterally controlled slave robot, which works on the task in the environment. A 3D camera provides visual feedback of the slave environment to the master side. The AR system constructs and displays the slave robot environment model based on the visual feedback. Virtual fixtures are generated and overlaid onto the multi-modal human percepts. The haptic sense is conveyed via a haptic device, while the visual display is conveyed via a 2D monitor. This configuration can provide a precise and stable teleoperation, assuring the virtual fixtures are placed precisely. The integrated display is implemented using RViz, a ROS-based VR environment.

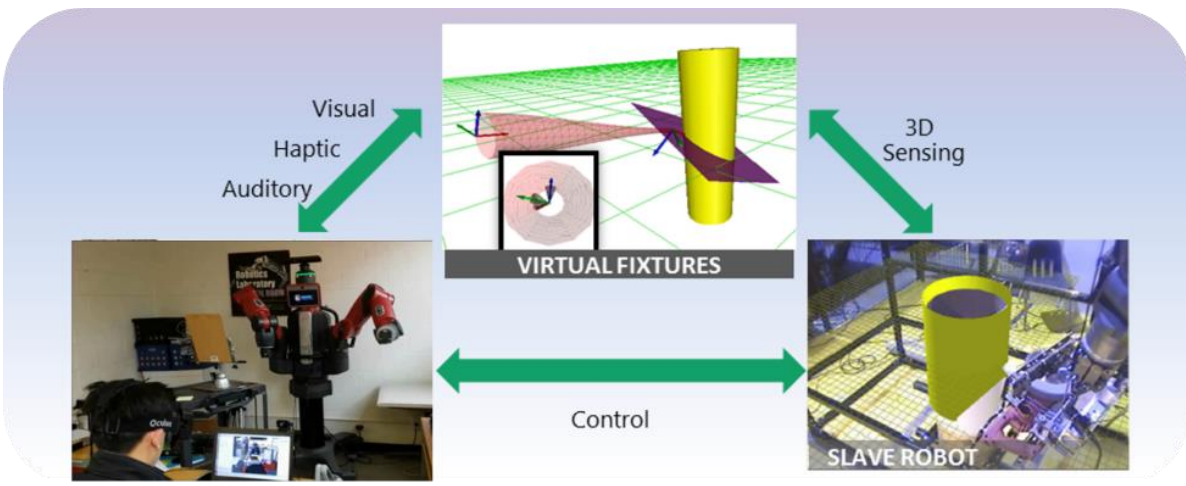

Figure 1. Augmented teleoperation system overview.

Figure 2. Baxter robot inside simulation.

The telerobotic system has many components and elements. The user interface and the slave robot environment were remote from each other, connecting online for decommissioning applications. Due to the remote aspect of the project, communicating user commands and various feedback data in real time is complicated and difficult to implement. These circumstances call upon the integration of an operating system robust enough for non-expert users. The ROS framework helped achieve the current status of the teleoperation system thanks to the utilization of its various tools.

ROS Integration in Teleoperation

Having an overview of the telerobotic system, one needs to understand the telerobotic system integration in ROS. Each component or node works independently and shares data in an asynchronous manner based on the ROS framework. ROS works by publishing and subscribing information through messages over the network [4]. This telerobotic system had 8 nodes, but they can be grouped into the following: slave robot, RGB-D sensor, RViz, haptic device, and user command interface. The interrelationship between them can be seen in Figure 3, which shows the ROS-based message communication of the telerobotic system.

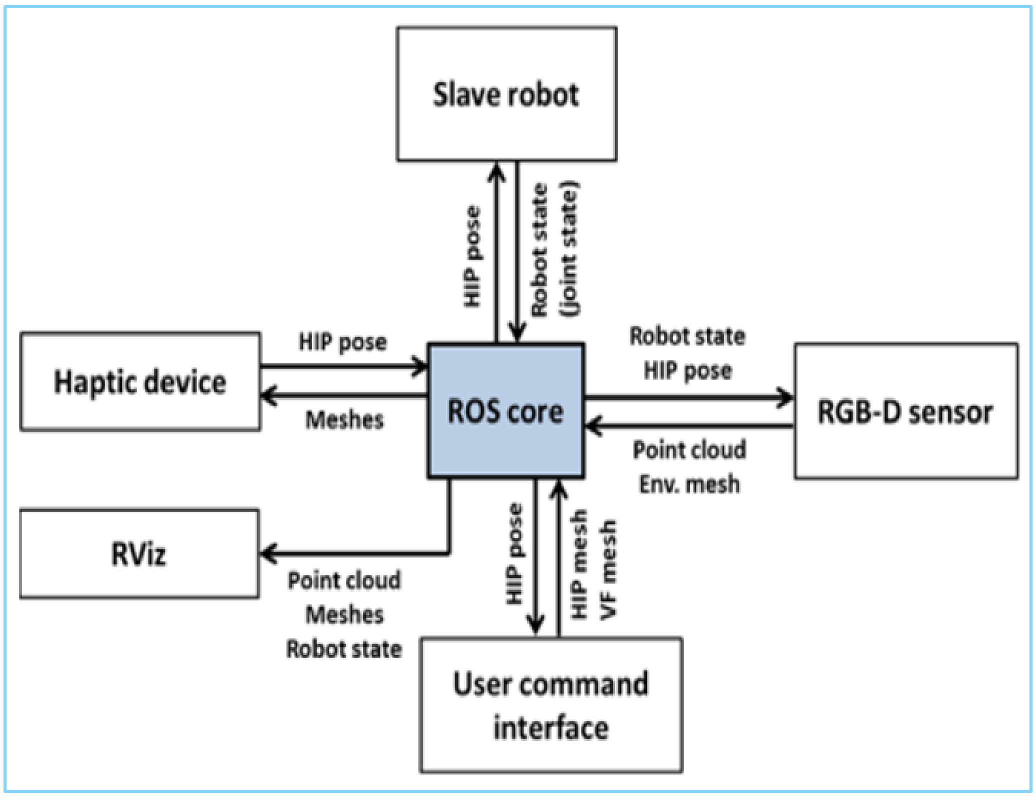

Figure 3. ROS structure, along with nodes.

The slave robot group subscribes pose of the haptic interface point (HIP) and the user command interface to track HIP with its end effector, according to the user command. This part publishes the robot joint status information. The red, green, blue, and depth (RGB-D) sensor publishes 3D point clouds of the observed environment for visual feedback, and it also reconstructs 3D polygonal meshes from the point cloud data and publishes it for haptic feedback. This group subscribes the HIP position and robot joints state to reconstruct the 3D environment.

RViz subscribes all the information related to visualization and renders it to monitors. For instance, there are point cloud data from the RGB-D sensor, the pose, and geometric information (mesh) of HIP, robot model, and geometric information of user-defined virtual fixtures. The haptic device group publishes HIP pose, while it subscribes all the meshes to render the haptic effect. The user interface collects the user's commands and HIP pose, and it defines and broadcasts the user-defined virtual fixture and HIP meshes.

One of the key features of ROS, used for the teleoperation system, was the *tf* [5]. This feature functioned as a coordinate reference for independent components in a large-scale robot system. Each object had its own *tf* frame, and they are all connected to each other like a chain. This permitted transformation between components, such as the robot base and the RGB-D sensor coordinates, to be easily calculated. This aspect of ROS is instrumental for the camera calibration step.

Camera Calibration

In order to perform augmented teleoperation by providing augmented virtual fixtures for precise and dexterous manipulation, camera calibration had to be performed. The robot Baxter already had ROS incorporated into its system. It also had constant calculations of the positions and orientations of its components. As part of the system, the head camera, which consisted of an Xbox Kinect camera, would take a point cloud, providing RGB-D values for display on a 2D screen. These captured images and point clouds of the real environment were projected on an RViz simulation where the augmented teleoperation would take place. Inside the simulation, a Baxter robot model would also be projected, as seen in Figure 2. As part of a visualization expansion, a second camera to the left hand would increase the point cloud and the 3D reconstruction of the environment. This expansion would give Baxter a more precise and dexterous movement once a user manipulates the arms during its applications in decommissioning.

The camera calibration consisted of multiple steps. First was developing the launch files for turning on the cameras through ROS. Then, with a board that contained black and white squares, the user positioned the left-hand and head cameras at an angle, as seen in Figure 4. At this stage, the software began calibrating between them, recognizing the patterns and dimensions input before running the software. For this step, the board was moved around until enough images had been collected. Once enough images had been taken by the cameras, software was used to calculate the position and orientation values. These raw values have to be transformed into the Baxter position and orientation values, which was performed through a Python script that shifted each transformation.

Figure 4. Camera calibration process between left-arm camera and right (head-Kinect) camera.

The Python script transformed the raw values obtained from Figure 4 into Baxter's position and orientation. The raw values obtained consisted of the position and orientation of the left-hand camera in respect to the head-Kinect camera. In order to calibrate the cameras, the Python script transformed the head-Kinect and left-hand cameras back to the Baxter base. These transformations can be seen in Figure 5, where the Kinect is transformed to the RGB camera, transformed to the left-hand camera, and lastly transformed to the base of Baxter. Once these transformations were calculated, the resulting values obtained were input into a ROS file that would shift the simulated environment and the real environment. This file was then tweaked to ensure the real and simulated environments overlapped.

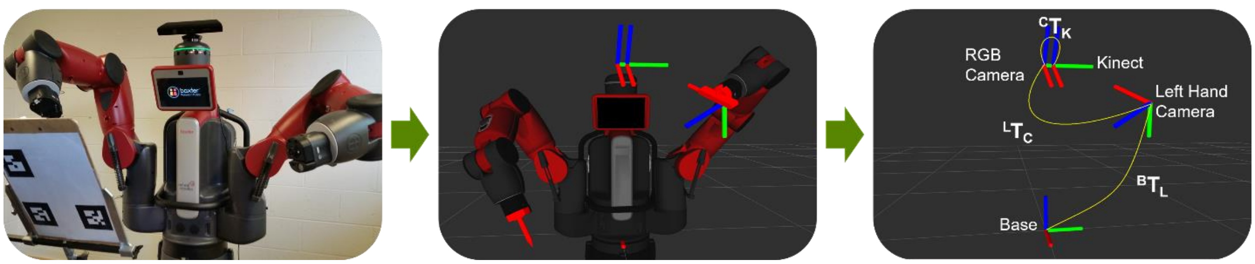

Figure 5. Baxter, Left-hand, and Kinect positions.

4. RESULTS AND ANALYSIS

The principal task of the internship project consisted of performing camera calibration for the teleoperation system. This would help to place accurate virtual fixtures and control the robot in a dexterous manner for decommissioning applications. The purpose of the left-hand and head-Kinect camera calibration was to overlap the real and simulated environments. Part of the process was to automate camera calibration. Moreover, a clear and detailed step-by-step process for accomplishing this was needed for future reference.

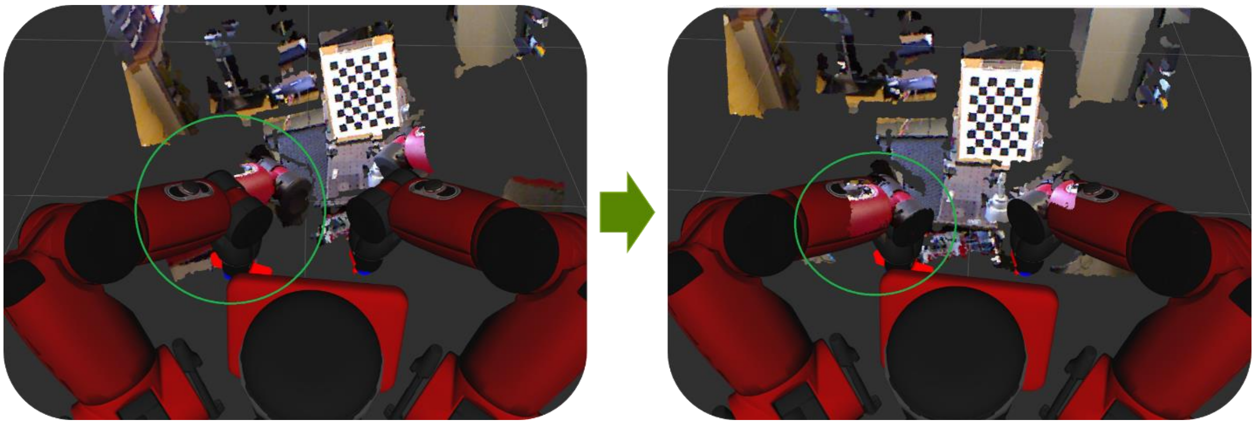

Figure 6. Before and after camera calibration of Baxter's cameras.

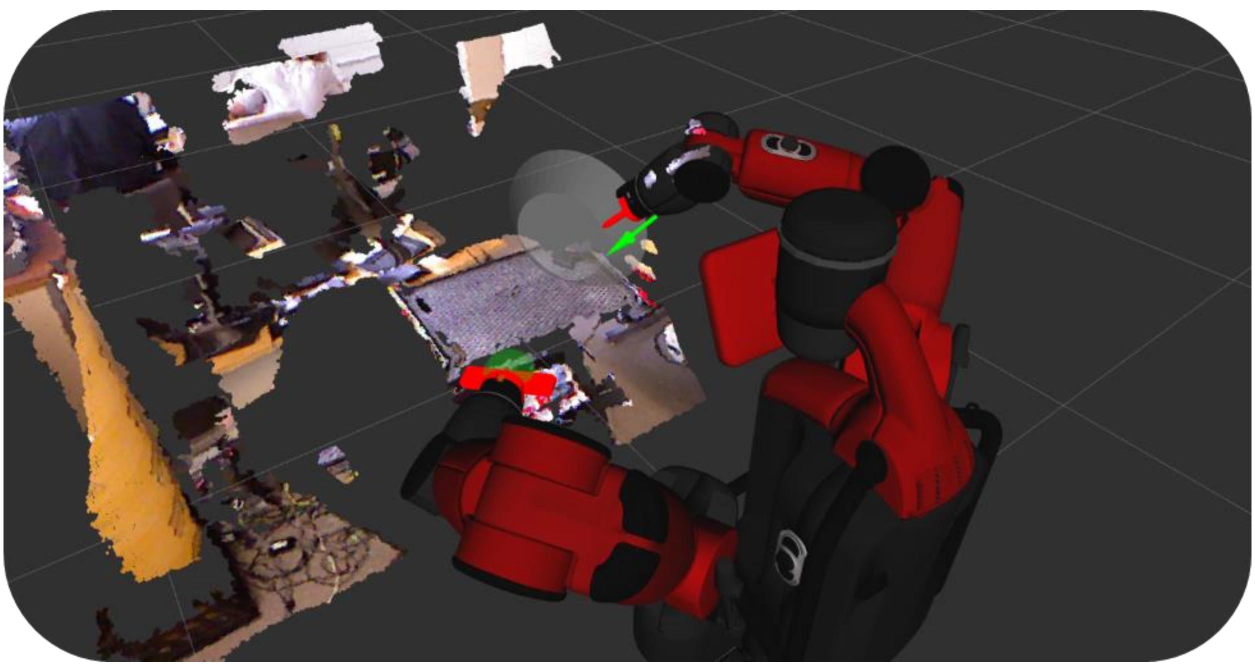

Figure 7. Baxter calibrated cameras and simulation.

As seen from the results in Figure 6 and Figure 7, camera calibration was able to correctly overlap the real with the simulated environments. Once the recalculated transformation values of position and orientation were obtained, these values were input to the according script that modified the environments in the simulation. The process of camera calibration is not currently completely automated so after the transformed values were input, manual editing had to be performed. Achieving overlap between the virtual and real environments was successful, but camera distortion or distortion parameters had to be considered and further analyzed to account for errors in the calibration process.

The automation process can be achieved through the use of Python scripts that can access the raw values obtained during the camera calibration steps. With this, the calibration process could become independent of the user once the calibration is initiated. These considerations should be considered for future steps in the project. Additional future steps would consist of adding a right-hand camera to improve the 3D reconstruction and the point cloud. Furthermore, an effective and efficient method to merge all of the point clouds obtained from the left-hand, right-hand, and head-Kinect cameras is needed.

5. CONCLUSION

During this research effort, a few problems were encountered. The first was incompatibility between different software systems. There are multiple ROS distributions which depend on which Linux system is being used. The project was run across a number of computers, each containing a different version of Linux, ROS, or other software required for the implementation of the program. Due to these issues, significant time and effort was spent troubleshooting software errors. Determining solutions to these issues was especially challenging since the cutting-edge teleoperation research being pursued could not look to previous projects for best practices and lessons learned. A recommendation for future efforts includes ensuring that there are no compatibility issues across the various platforms to optimize progress on the project.

Conversely, camera calibration, which is crucial for precisely placing the virtual fixtures, was successful after running the simulations. The real and simulated environments overlapped, permitting the utilization of augmented teleoperation and the dexterous manipulation of Baxter for D&D applications at nuclear facilities.

6. REFERENCES

[1] T. Ranklin, W. Kanne, M. Mouthan, D. Bickford, and J. Congdon. *Production of Pu-238 Oxide Fuel for Space Exploration*. Savannah River Site 50th Anniversary Symposium, Aiken, South Carolina, May 2000. Retrieved from http://www.c-n-t-a.com/srs50_files/179rankin.pdf

[2] Savannah River Nuclear Solutions. *SRS Building 235-F. Facts from the Savannah River Site*. Retrieved from www.srs.gov

[3] R. Arbitz, A. Duncan, M. Kane, R. Dewberry, and J Santos. *Planning for the Decontamination of the Plutonium Fuel Form (PuFF) Facility Savannah River Site (SRS), South Carolina*. Waste Management Conference, Phoenix, Arizona, March 2010.

[4] M. Quigley et al., “ROS: an open-source Robot Operating System,” In *Int. Conf. Robotics and Automation Workshop on Open Source Software*, Vol. 3, No. 3.2, pp. 5, May. 2009.

[5] T. Foote, “tf: The transform library,” in *Proc. Int. Conf. Technologies for Practical Robot Applications (TePRA)*, pp. 1-6, Apr. 2013.