Accessible Robot Control in MR

Mixed Reality Project Midterm Report
Supervised by: Eric Vollenweider
November 12, 2022

GROUP MEMBERS

Deheng Zhang | Longteng Duan | Ganlin Zhang | Guo Han




This project aims to help people with arm or hand amputation to operate the Spot robot using HoloLens. More specifically, we plan to design and implement a pipeline that enables people to move the robot, control the robot arm and grasp items by eye tracing and head motion. The pipeline of the project is briefly illustrated in Fig. 1.

I. CURRENT STATUS

As shown in the demonstration video, we can now use eye gaze assisted by voice to control the motion of the Spot robot. Detailed achievements are elaborated below.

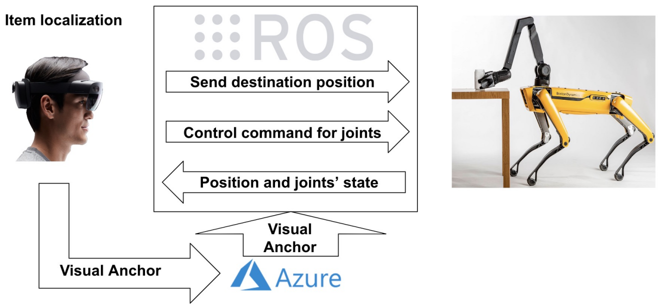

Fig. 1: Project Pipeline

Eye Gazing Acquisition The eye gazing ray is obtained through the eye gaze API of HoloLens. Specifically, we calculate the intersection of the eye gazing ray with the meshes of the world, and identify the intersection point with a cursor such that the user can easily find the results of this intersection.

Voice Control Allowing all commands to be made without hand gestures is challenging. To overcome this problem, we use voice control to help the user interact with the HoloLens. These commands include create/delete spatial anchor, start/stop sending commands to Spot robot, etc. Also, we also show the result of the recognition of voice to user so that user know whether he/she sends the command successfully or not.

Communication In our project, we have two main devices, HoloLens and Spot robot. To establish communication between them, we use ROS to transfer the messages. In Unity, we use *Unity Robotics Hub*, which enables Unity to send ROS messages. In this way, the Spot robot can receive the commands from HoloLens in real time.

Co-locolization The intersection point sent by HoloLens is in the Hololens' world frame. However, the world frame of HoloLens and Spot robot are different. Thus, in order to align these two frames, we need something in both frames to establish a connection. We use Azure Spatial Anchor to do this job. Some visual based anchors are created by HoloLens and sent to Azure. In Spot end, it also captures images and query them in the Azure cloud. In this way we can get the poses of anchors in both frame and know the transformation of them.

Eye Gaze Control Integrating the above functions all, now we can use HoloLens to get the position focused by eyes, and send it to the robot to let it move to the desired position.

II. IDENTIFIED PROBLEMS

Since our target users cannot use their hands to give instructions to the HoloLens, we can use neither a graphic user interface for item selection nor an intuitive controller for robot motion control. Therefore, we need to utilize the multifunctional input provided by the HoloLens such as eyegaze and voice. Besides, we need to send and request the spatial anchor to / from the Azure cloud for coordinate transformation between HoloLens frame and robot frame. As a consequence, we encountered more problems than expected.

Unity Version The first problem we faced is the incompatibility between the eye gaze module and the spatial anchor module. It turns out to be a problem caused by the Unity Editor version 2020.3.10. After updating the version to 2020.3.40, this problem is solved.

Eye Gaze Cursor Another challenge is that the eye-gaze cursor is problematic. We move the cursor to the intersection point between the eye gaze and the mesh in the *Update()* function for every frame. However, the eye gaze intersection interface provided by the *MRTK* takes all game objects into account. When the cursor is moved to the intersection point, the eye gaze intersects with the cursor, and this point is mistakenly used by the *MRTK* as a new hit point to move the cursor. As a result, the cursor will directly move to the camera. We solved this problem by thresholding the distance between the current cursor position and the hit position. If the distance is large enough (the hit point is impossible to be on the cursor), the cursor will be moved to the intersection point.

Voice Control Package The voice control is not accurate enough when we first added this feature. Even though we have changed the confidence level, the performance only improved slightly. It turns out to be the interface *IMixedRealitySpeechHandler* we initialy used is not optimal for voice recongnition accuracy. After changing it to the *SpeechInputHandler* component, the problem is solved.

Coordinate Alignment The Unity, Azure Spatial Anchor, and ROS use different coordinate systems. Unity uses left hand coordinate, while Azure and ROS use right hand coordinate. The X-Y-Z order is also different for Azure and ROS. Besides, during the coordinate transformation between the world frame and the local frame of the created anchor, we found that the local scale of the anchor affects the transformed coordinate as a feature of the Unity. Therefore, we spent a lot of time to align the coordinates correctly.

Spatial Anchor Accuracy The anchor position is computed by matching the feature captured by HoloLens camera and the feature captured by the Spot robot. However, due to the error produced by depth estimation and camera distortion, the hit position cannot be calculated accurately in the robot's frame.

III. NEXT STEPS

Solve Legacy Problems As mentioned before, the transformed / co-localized target location in the ROS frame is not accurate enough. To avoid giving the robot an erroneous destination and improve the user experience, this problem should be relieved. One possible solution is to initialize more anchors to improve the result's robustness.

Next Milestone - Robot Arm Control Our next milestone is to enable the Spot robot to move near the target item and allow the user to control the robot's arm to grasp the object using head. The head motion will be caught by the camera movement. The captured motion will be then mapped to the Spot arm by calling the ROS's API. Predictable challenges include motion mapping accuracy, robot arm collision avoidance, and user-friendly grasping process design.

User Interface Implementation Considering our project topic is accessible robot control in mixed reality, the HoloLens in our case primarily serves for interaction and localization purposes. We want the user to focus more on reality and our UI design is expected to be straightforward. Ideally, it will provide alert information and necessary guidance.

User Study The target group for our object is people with arm/hand amputation, so an emphasis on user study should be stressed. Fortunately, normal people can mimic the physical constraints our target people face relatively easily. We will evaluate the using experience by considering aspects of the user interface, operation smoothness, and instruction clarity of our application.