Operating Spot in Virtual Reality

Jacob Epstein

Motivation

- Develop a Virtual Reality (VR) interface for performing navigation and manipulation tasks with Boston Dynamics' Spot robot
- Reasons this is useful
  - Remote operation with increased spatial awareness
  - No real-world limits on UI!
- A secondary motivation - demonstrate that our VR interface can be extended to a variety of different robots
  - Valkyrie, Baxter, Fetch, now Spot, Hubo??

Methods and Approach

- VR environment front-end created using Unity Game Engine
- ROS back-end
  - Used the spot-ros wrapper for the spot sdk
- ROS.NET for bridging between the two

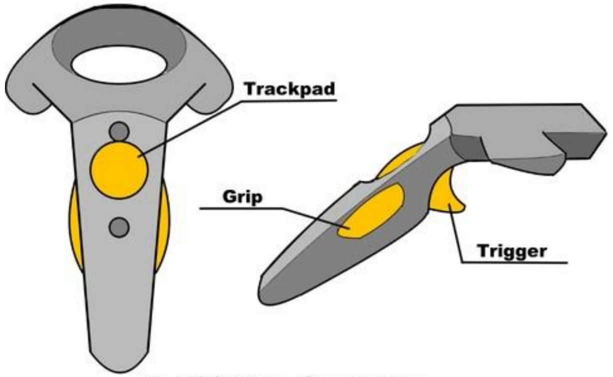
ROS

ROS.NET

unity

Results

- Interface for basic control of spot (claim, stand, sit, dock, undock, etc.)
- Interfaces for navigation and manipulation* with Spot
- Interface for controlling the orientation of spot's body
- Display
  - Spot's body and arm
  - pointclouds
  - objects with AprilTags
  - GraphNav maps



a) HTC Vive Controller

Theme Of The Summer: Extend VR interfaces for Navigation, Manipulation, Display, Control to work with Spot while dealing with hardware-based constraints; in this case, having a limited number of buttons on the VIVE controller.

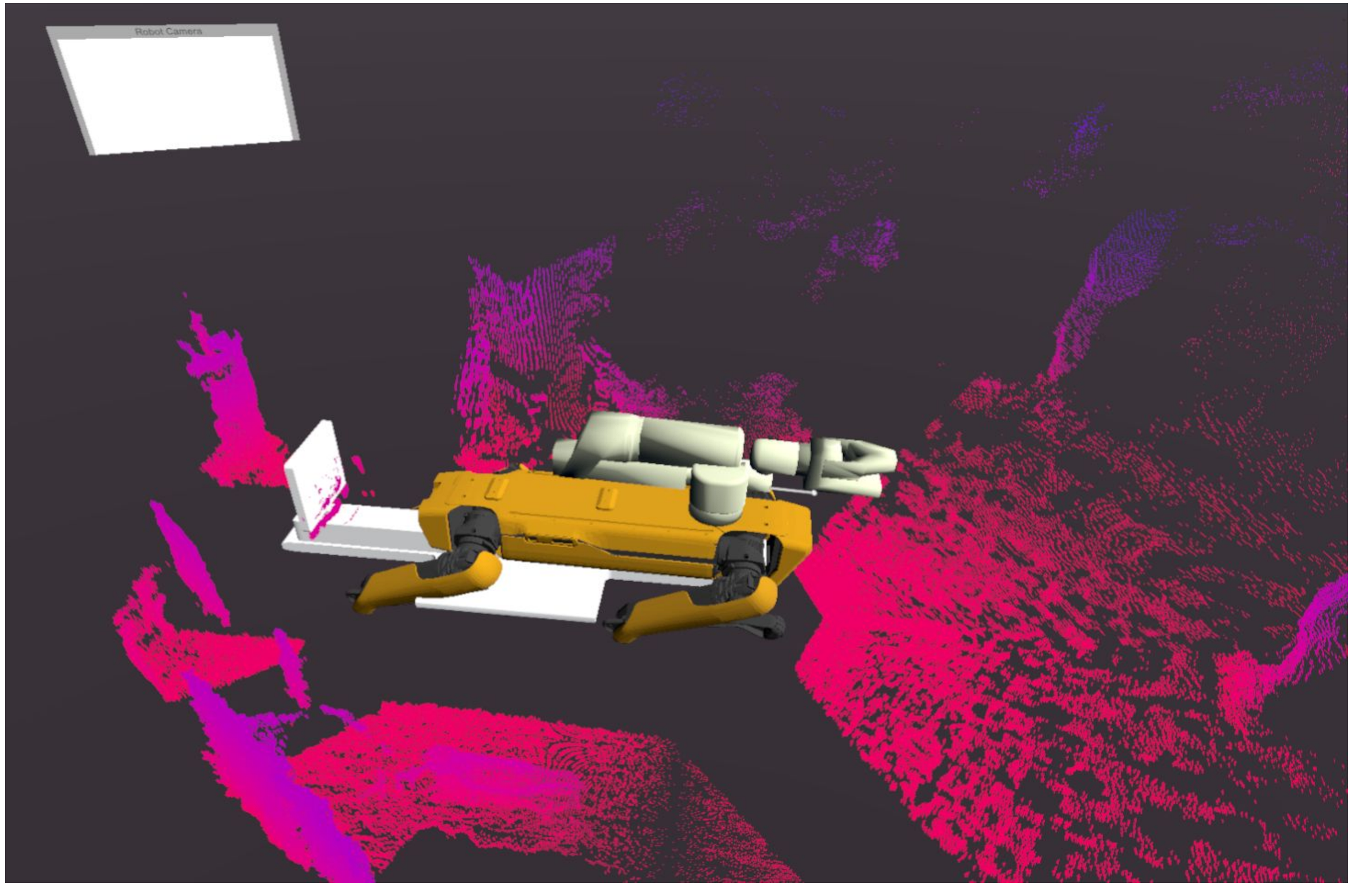
Control

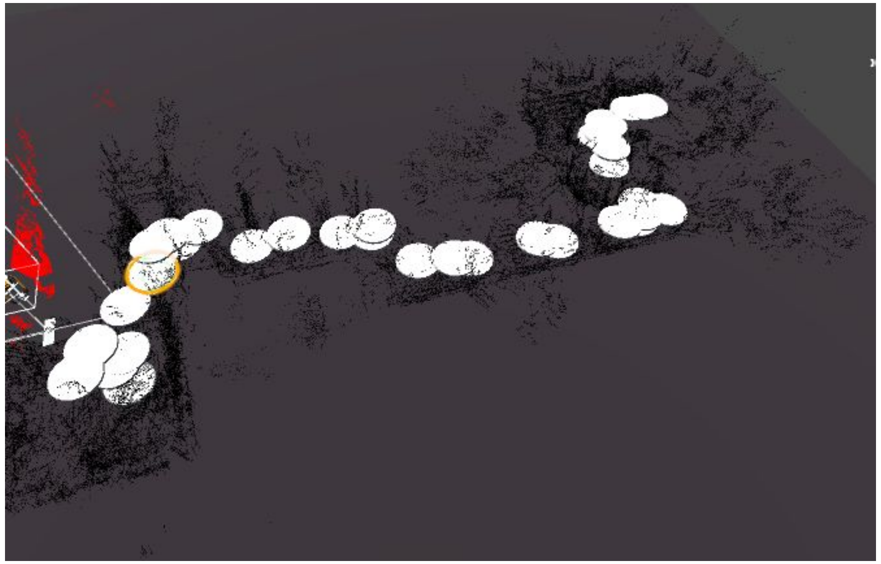
Navigation

Body Orientation

Manipulation (almost!)

Display

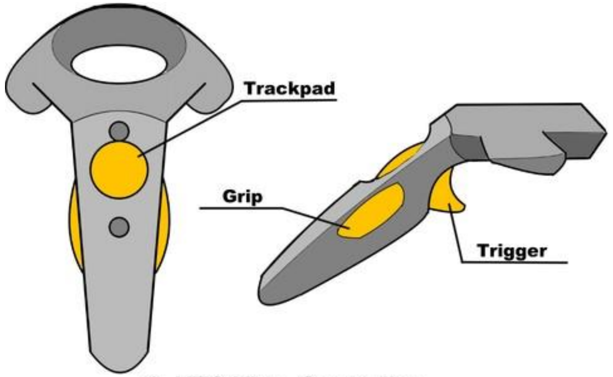




Conclusions

- Our VR interface can be extended to a variety of different robots, with a relatively low degree of effort required
- This is because our core Navigation, Manipulation, Display, and Control interfaces solve the limiting problem in elegant ways
- It's all about meeting the constraints of the hardware while still allowing for precise control

Limiting factor is the number of buttons on the controller!



a) HTC Vive Controller

How we solved the limiting problem:

Control: Wristwatch UI contains all necessary buttons and sliders needed for control

Navigation: Point and click placement of nav goals, plan and execute the trajectory with the Wristwatch UI

Manipulation: Ability to spawn in and quickly clone manipulation goals using the Wristwatch UI, trigger, and touchpad buttons. Plan and execute with the Wristwatch UI

Display: No need to press a single button! It's done automatically.

Spot-Specific Interfaces: Body Posing using a trackpad in the Wristwatch UI, GraphNav-based navigation with interactable waypoints

Future Plans

- **Short term**
  - An official lab fork of spot-ROS
  - Debug manipulation, add support for GraphNav-based navigation
  - Hubo!
- **Long term**
  - Study the differences between 2D and VR interfaces
  - Take the interface beyond the prototyping phase - redesign it with a specific application in mind (potentially one informed by what we find to be certain strengths specific to a VR interface)
  - Operating more than one robot at a time in VR?

Thank you for a fantastic summer!