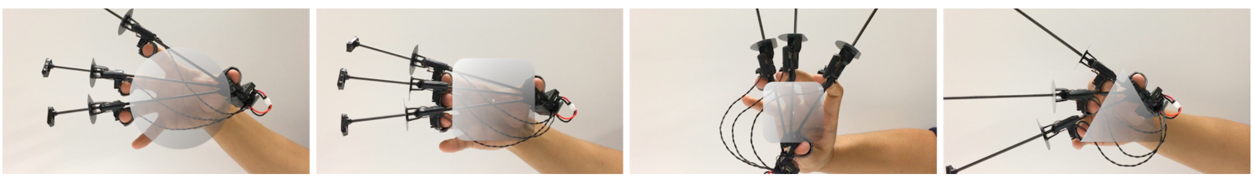
EE267 Proposal

Inrak Choi

BACKGROUND

Though VR has been explored in research contexts since the late 1950s, recent advances in display technology have made consumer VR a reality. While new devices such as the Oculus Rift or HTC Vive provide high resolution visuals, the user input devices have been limited to traditional game controllers and existing styles of gestural input. It is desirable to allow users to touch what they can see and physically manipulate virtual objects. However, current consumer input devices do not provide the kinesthetic force feedback that we experience when interacting with objects in the real world. We seek to meet these design objectives for the specific case of grasping virtual rigid objects in precision-based grips.



Wolverine is a mobile, wearable haptic device designed for simulating the grasping of rigid objects in a VR environment. In contrast to prior work on wearable force feedback gloves, we focus on creating a low cost and lightweight device that renders a force directly between the thumb and three fingers to simulate objects held in pad opposition (precision) type grasps. Leveraging low-power brake-based locking sliders, the system can withstand over 100N of force between each finger and the thumb, and only consumes 2.7 μWh (10 mJ) for each braking interaction. Small vibration motor on each finger generates vibrotactile feedback for informational cues. Integrated sensors are used both for feedback control and user input: Three magnetic encoders provide the position of each finger and Four IMUs provide the orientation of each finger and the thumb.

MOTIVATION

There have been many wearable haptic interfaces developed recently. One of the issues is “hand/finger tracking”. To give realistic force feedback and to render hands in VR with correct positions and orientations, precise tracking is important. However, due to the complexity and large number of degrees of freedom of hands, it is very challenging to track them.

Hand tracking methods can be divided into two groups: tracking with internal sensors and tracking with external sensors. Many wearable haptic devices have integrated sensors, such as encoders, IR sensors, ToF sensors, hall effect sensors, and IMUs, to sense distances and angles. The tracking performances vary depending on the design. Sometimes resolutions are not good or signals are too noisy. To overcome this, some people use external motion tracking systems. Usually they are composed of multiple optical or magnetic cameras. The cameras track markers on rigid bodies. Generally, the external sensors are more stable and accurate, but more expensive and the markers make the device bulky. There are pros and cons for sure.

In this EE267 project, we would like to compare the actual tracking performances between a set of internal sensors on a wearable haptic device and an external motion tracking system. As we have developed a wearable device, Wolverine, we are going to use this device for this comparison.

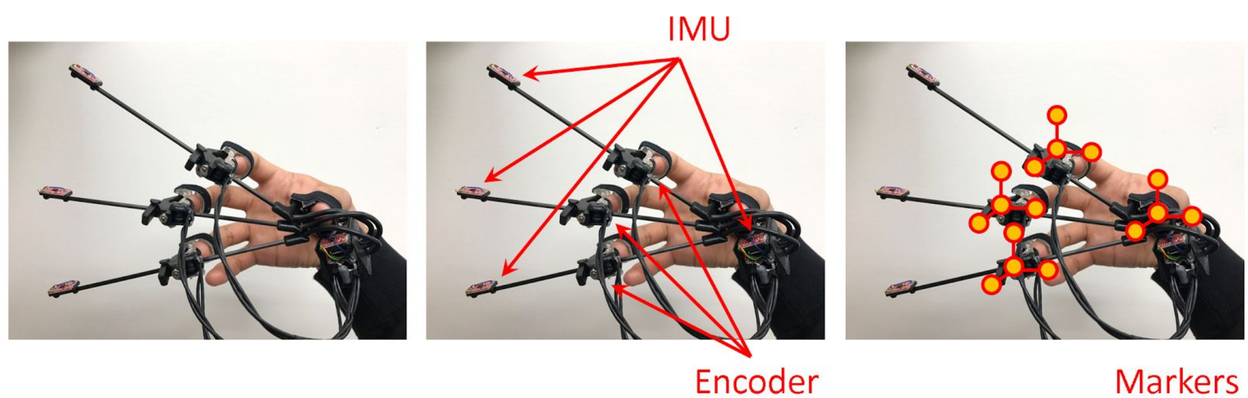
GOALS

I. Find position errors between the integrated sensors and external tracking system for hand rendering in a three dimensional space.
   - Integrated sensors: A set of 4 IMUs and 3 magnetic encoders
   - External tracking system: Optitrack system with markers

II. Find position errors & noises with both sensing system when vibration motor is on.

III. Summary pros and cons of each system in terms of sampling rate, latency, errors, etc.

PLANS



Isometric View | Mode 1 | Mode 2

I. The pictures above show 1) general shape of the device 2) tracking mode with integrated sensors 3) tracking mode with OptiTrack. In the mode with OptiTrack, we need to attach additional markers on the device.

II. For the mode with integrated sensors, we will write codes in Arduino for the microcontroller (Teensy 3.2) to read sensor values.

III. For the mode with OptiTrack, we will use Motive software.

IV. For rendering the hand, we will use CHAI 3D software. It will receive sensor values from Arduino and Motive.

V. We will see two images at the same time through a screen that rendered with the Mode 1 and Mode 2.

VI. We will print out the values of fingertip positions to analyze the position errors, noises, sampling rates, and latency.

VII. We will analyze the values.