Hannah Xue (yx348)

February 21, 2016

Target Acquisition with Leap Motion: Design and Implement

1 Introduction

Leap Motion is a device which makes detecting natural hand gestures easy. In this report, three interaction designs (hand rotation, screen tap and clap) with Leap Motion are implemented and evaluated with a simplified Fitts' Law task (12 circles). Hand rotation performed the best both in speed and accuracy.

2 Design Process and Evaluation

2.1 Explore the design space

Leap Motion uses two monochromatic IR cameras and three infrared LEDs to detect hands in a roughly 1 meter hemispherical area[1]. It provides a variety of built-in gestures, including circle, swipe, screen tap and key tap[2]. It is also easy to detect hand and finger coordinates and rotation angles. Multiple hand detection is supported.

Leap Motion comes with an App Home[3], similar to Apple's App Store. Developers can develop and upload their Leap Motion App. I tried several apps on the list. Most of them involve natural gestures like grab, pick and point. Some music games try to mimic a keyboard tap gesture. However, the detection of those complicated gestures are not very accurate. It is easy to unintentionally carry out actions, thus causing frustration for users.

2.2 Brainstorm designs

Leap Motion actions can be generally categorized into finger gestures, hand gestures and hybrid gestures.

Finger gesture:

- A natural thought is to use one finger movement to control the cursor. The hand should remain steady. In this case a click action could be bending the finger. It could also be drawing a cross or a circle.

---

[1] "Analysis of the Accuracy and Robustness of the Leap Motion Controller". PubMed Central.

[2] https://developer.leapmotion.com/documentation/javascript/api/Leap.Gesture.html

[3] https://apps.leapmotion.com/

- Using two fingers to control the cursor (there could be two cursors or the mid point of two fingers coordinates could be the cursor position). Moving them closer would be the click action.

Hand gesture:
- Using one hand position to control the cursor. Rotate or shake the hand to click. This is similar to the first interaction with finger gesture.
- Using two hands and clap to click. This is similar to the second interaction with finger gesture.

Hybrid gesture:
- A single hand grab gesture could be used to ‘catch’ the target the user wanted to click.
- Similarly, a screen tap or key tap gesture could be used for click action and hand movement controls the cursor.

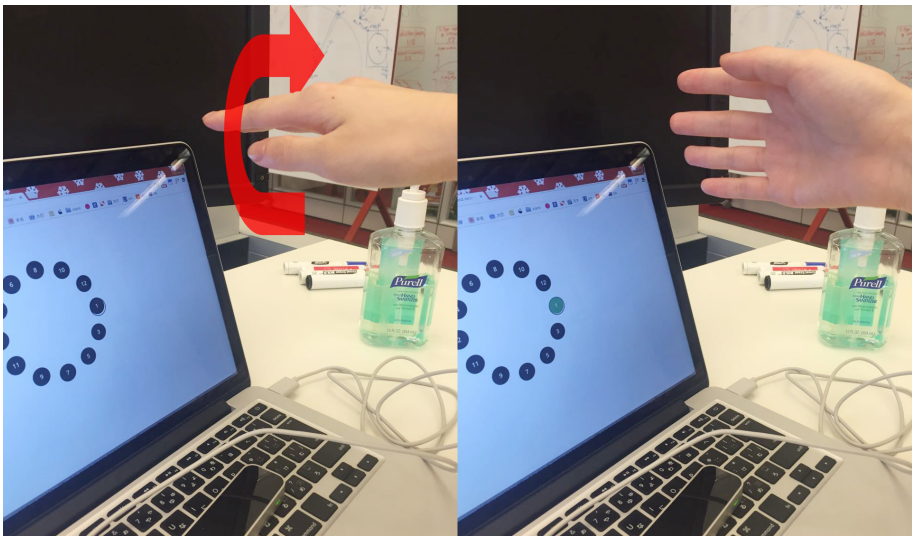

FIGURE 1. INTERACTION TECHNIQUE: HAND ROTATION

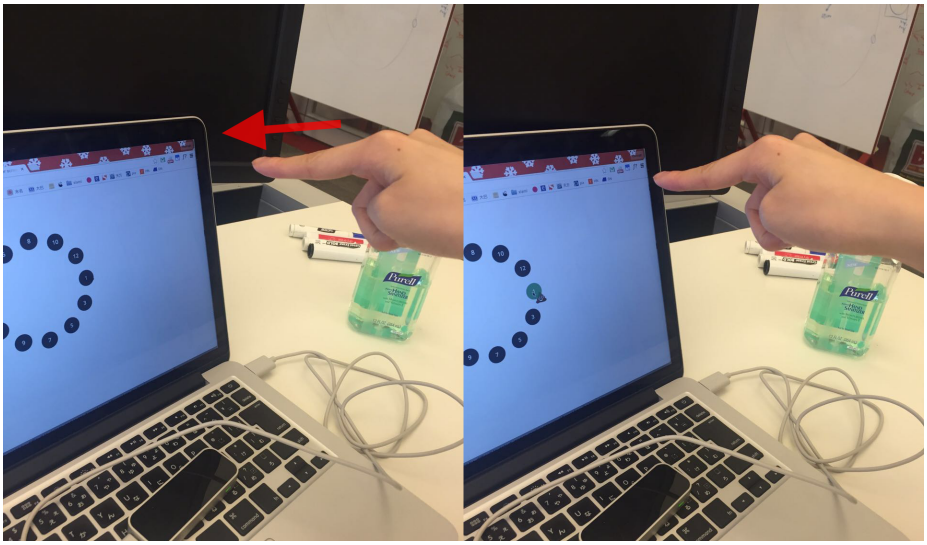

FIGURE 2. INTERACTION TECHNIQUE: SCREEN TAP

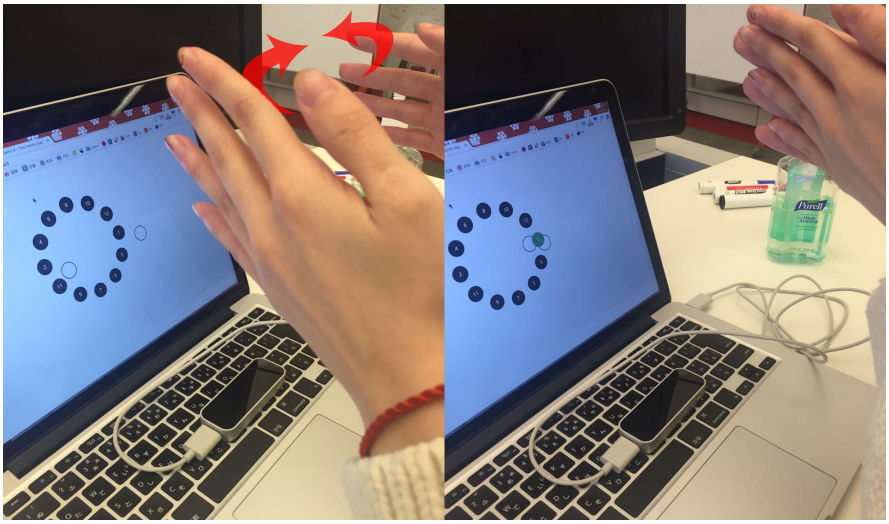

FIGURE 3. INTERACTION TECHNIQUE: CLAP

2.3 Converge

Finger gesture is the most intuitive. However, not only does it require user to remain their hand steady, the click action is time-consuming. Bending the finger would not be accurate since it is easy for Leap Motion to mistake it as cursor movement. Drawing a circle or cross also has a low accuracy, and is very slow. Thus finger gestures are not considered for prototyping.

Although hand gesture is strenuous, Leap Motion can detect hand position much more accurately since it is a larger target.

Rotating hand (Figure 1) is faster than shaking hand, and will not affect the x and y coordinate of the hand, which is important for accuracy. It might not be explorable for users without extra instruction, but is possible to be an effective interaction approach. Similarly, clapping (Figure 3) is also quick and accurate to detect for the device. However since it involves two hands it is more work for the users, and less accurate because the ‘clap’ action has to happen at the exact x and y coordinate mapping to the correct location on the screen.

Hybrid approaches combine the pros from both sides. From the technical side a grab gesture might be difficult to detect, and to me it is not very different from the hand rotation approach, thus I decided to prototype the screen tap approach (Figure 2). Ideally the cursor coordinates would be accurate, and the screen tap action should be accurate and fast, since it only affects the z coordinate of the hand and will not change the cursor position.

To conclude, I decided to prototype the hand rotation, clap and screen tap approaches. All of them should be easy to learn with a one line instruction or even only some implications from the UI.

2.4 Prototype

I have already explored the Leap Motion JavaScript APIs during the exploration phase, which I found easy to use, so I decided to code the designs directly. After testing it myself I found the detection area of Leap Motion relatively small, so I decided to cut down the Fitts' Law target numbers to 12.

Since I'm only comparing the three interaction techniques, I fixed D value to 300 pixel and W value to 50 pixel, which is approximately 1.4 inch and 0.3 inch respectively on a Mac retina display[4].

Participants
I asked three classmates Haiwei (CS'16), Daniel (CS'16) and Sam (CS'16) from Cornell Tech to try all the three prototypes.

Apparatus
The prototypes are made in HTML5/CSS/JavaScript/JQuery with Leap Motion JavaScript Library and can run under any modern desktop browser directly. No extra tool or library is used. The experiments are run under Google Chrome Version 48.0 with a 13-inch Macbook Pro, El Capitan V10.11.

Live demos: https://github.com/hanax/leap_motion_interaction_demos

Procedure
I explained to them the coordinate system of Leap Motion, and the moving and clicking interactions I applied in each prototype. I then asked them to complete the Fitts' Law task with each prototype 3 times. The first try is not counted. The prototype automatically recorded the time they spent to complete each click and also the accuracy rate.

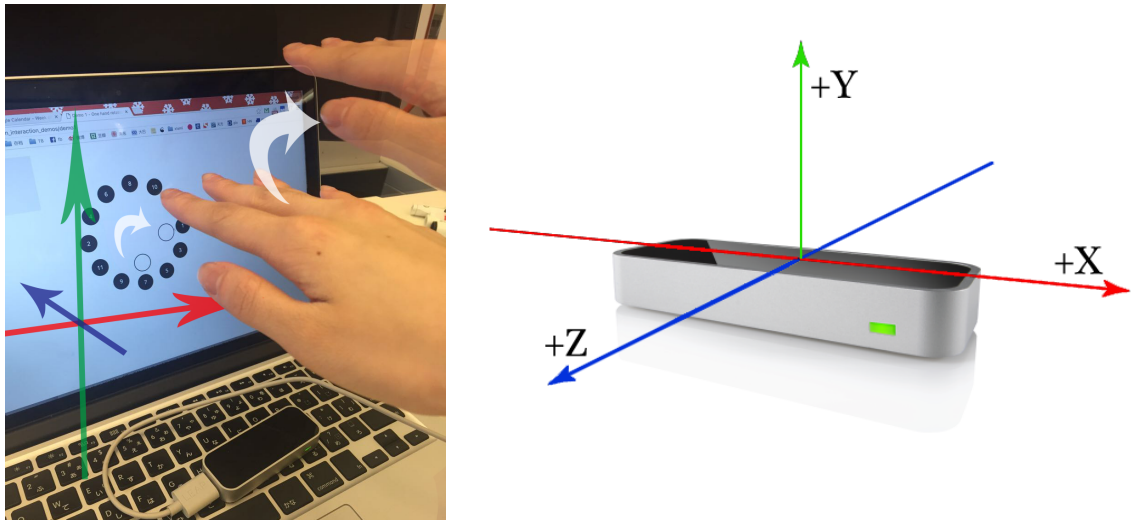

FIGURE 4. LEAP MOTION COORDINATE SYSTEMS[5] AND CURSOR CONTROL WITH HAND

[4] https://support.apple.com/en-us/HT202471

[5] https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Coordinate_Mapping.html

Feedbacks and results
For the first iteration I didn't explain the Leap Motion coordinate system (Figure 4), and the participants tend to mistake the z-axis as the y-axis. Haiwei said this is because he is used to the keyboard direction keys and naturally map that coordinate to the prototype. After I explained that to them they are able to control the cursor with Leap Motion well quickly.

For the screen tap prototype, it is almost impossible for them to complete the task. The tap gesture is detected poorly, and affects the cursor position greatly resulting in low accuracy. I applied the built-in screen tap gesture detection algorithm, which I assumed is already one of the optimal solutions. Due to the extreme low speed and accuracy, screen tap is not a good choice for the task.

Daniel and I myself found the clapping prototype easy to use and fast and accurate to click. Participants have a good intuition of the mid points of their two hands. However a big concern here is when two hands vertically overlap, Leap Motion always loses track of one of them. This is the main drawback of the speed and accuracy for this prototype. On average participants spent 32 seconds to click all the 12 circles and have an accuracy of 18%. Most of the misoperation is caused by the track lost by Leap Motion. Sam said this is the most fun prototype of the three.

All participants found the hand rotation approach the easiest to use. The average task completion time is 23 seconds with an accuracy of 75%.

2.5 Refine

After the preliminary evaluation the hand rotation prototype outperforms the other two. I think the main reason is that for Leap Motion a hand rotation is a big gesture, which is easy and relatively accurate to detect compared to a screen tap gesture. It also only involve one hand, thus largely decrease the possibility of losing track of detection.

One pros I expected it to have is the steadiness of cursor position, since the rotation action can trigger a click without affecting the x and y coordinates. However Daniel reported that since I required him to rotate their hand 90 degrees clockwise, he still tends to move his hand to the right during the rotation. This is also effort taking.

According to this feedback, I made the detect boundary from 90 degrees to 45 degrees. Though the 45 degrees rotation is a rather subtle gesture, it can still be detected correctly. I also move the circles to the middle of the screen to make it easier for Leap Motion to detect.

I then let Daniel and Haiwei to try that again. This time the results reached 21 seconds in speed and 82% in accuracy.

3 Conclusion

A preliminary conclusion after evaluation is that hand rotation is the best in both speed and accuracy of the three prototypes. The clapping prototype is the most innovative and

fun. It is slightly worse in time but has a significant low accuracy. This is limited by Leap Motion detection techniques. The screen tap prototype cannot be used to complete the task in a reasonable time.

After refining the hand rotation prototype, task completion speed increased to 21 seconds with a high accuracy. However since I tested it on the same group of users there might exist error due to practice. This worths further investigation.