Gesture Controlled Virtual Lighting Band

Yu Qin yuqin@umich.edu

Introduction

Lighting band is a gesture-controlled music and lighting generation application. In the project, I focused on the natural gestures interaction design; the gestures are not only functional but also expressive. It is because current HCI systems mostly rely on functional, simple but repetitive hand movements like clicking, taping, swiping or zoom in and out to accomplish most computational tasks while the advent of VR AR technology is offering us a great opportunity to go back to natural and expressive body movement, which could be more efficient in the process of human-computer communication.

To unleash the power of the expression part of the gestures in the HCI system, I started from studying musical conducting gestures, which are great examples of universal, functional and expressive gestures used for real-time communication. By observing, annotating and abstracting several typical conducting gestures, I synthesized the core hand shapes and movements of conducting gestures that serve both functional and expressive purposes during the music playing, and then applied these highly abstract gestures into an HCI scenarios - this musical and lighting generation game.

Gestures used in this game include dual hands interaction, continuous hand movements, and hand shapes and orientation detections, which are not only technical functional but also natural and expressive by giving users the freedom to add their own interpretations in the process of interaction. The project is an exploration of how to apply rich musical conducting gestures into more general HCI using settings and how to unleash the power of natural hand movements into general HCI system.

Related Work

Gesture — hand movement is an important modality in human communication. It could be a good representation for Human's intention and emotion. However, most HCI systems seem to underestimate the natural expression power of the gestures while only treating hands as finger-level mechanical action executors like tapping on a button or clicking on a menu to accomplish series complex HCI tasks. In the article Hand Gestures for HCI, Axel Mulder pointed out the gestures we used in our daily communication, which he categorized as semiotic gestures, could be meaningful to be used in HCI system because this kind of gestures allows to users to act more naturally with low cognitive load, while the mouse gesturing failed to exploit the most valuable capabilities of hand movement[1].

Some applications did realize the current underestimation of full-hand or mid-air gesture interaction and tried to integrate full hand gesture into some HCI system. For example, U. V. Solanki and N. H. Desai designed Handmote allows users to use different hand shape to control home appliances, as shown in Figure 1[2]. It’s a good using scenarios but gestures used in the system are basically several static hand shapes, which could be unnatural for users to perform and entail high learning curve. There are some other similar applications like using gestures to control laptop, but all of these applications still stay within the techniques pattern — using low-level functional commands gesture to interact with the computational system, even they are full-hand involved compared to tapping or clicking.

Figure 1. Handmote using scenarios.

Unlike these hand-based remote control systems, some computational products indeed involving more natural hand movements. A typical example is the gesture-based conducting system called Conducting Master[3]. This project collected conducting gestures data from music professionals and then built a general conceptual model based on movement analysis and subjective questionnaires. Moreover, they applied the conceptual model to a real-time musical HCI scenario — gesture-based music retrievals, which proved the extensibility of this gesture system. However, the applications of such computational conducting gesture systems are mostly limited within the music-related scenarios like music control or music education because they are built from the professional perspectives of music.

In the project, I expanded the application of these conducting gestures to a more general game scenario while remaining the usability and naturalness of the interactions for non-musical people.

Gesture Design

One of the biggest challenges of building gesture interfaces is on high-level interpretation. The interface system isn't necessarily easy to learn but should effortlessly replicate after users learned it. The approach to building a comprehensive gesture interface system is to find a proper gesture mental model and apply it to proper tasks. Therefore, instead of directly using these conducting gestures into my application, I first did some researches on the mental model behind the conducting gestures system. My question is why the conductor uses this gesture and how I can abstract these gestures and apply them to a general HCI scenario.

The core function of these conducting gestures is to offer clear signals to the musicians about important playing-related points in real-time music playing: the starting point, the end pointing and the beats. These gestures are required to be obvious enough among the sequential movements of the conductor’s hands so that these musicians can differentiate and then interpret them correctly even from a long distance away from the conductor. The tricks conductors used to differentiate these gestures from others are using different hand shape, like palm face front indicating starts while palm face up means stop. Thus, in the game interaction design, I used this trick for selection and on/off tasks; the user uses two different hand shapes of the left hand — palm face down and up — to control the beating music clip play and stop. These two hand shapes are so differentiated with each other so that users won't easily get confused, so does the computer. Plus, there are three beating clips that are located in three different locations, which are visually indicated by three spheres with different colors. The user needs to use the direction of the left hand to tell the system which one they want to play or stop. These gestures as a system can still be naturally learned and performed by users in real-time.

Besides these functional gestures, the conductor performs some expressive gestures while interacting with the ensembles. For example, beyond the basic hand movements for music beat, they will use dynamic hitting stress for each beat to communicate their understanding of the music with the musicians: whether the music is edgy(hit hard between beating points) or smooth. Moreover, their hand movement range will indicate the overall feeling of music playing like vigorous or elegant. Similarly, in this music and lighting generation game, the right hand takes in charge of speed control. The movement speed of the right hand will cause the speed of music playing and the intensity of lights changed correspondingly. A bigger range of the hand movement will lead to more intensive lighting change, which creates a vibe that matches the player's potential emotional expression.

The reason that I design dual hands interaction is for two reasons. One is to avoid the interference between these gestures especially when the user is moving the hand to change the speed of lighting and music, it won't accidentally trigger the start and stop commands because only the left hand can control the start or stop. Another is that it allows users to interact with the system in more diverse ways; they can manipulate the speed of music at the same time while using another hand to start or stop one of them, which lead to more interesting audio output. What's more, the system only captures the key landmarks of these gestures, which are the palm direction of the left hand, the hand direction of the left hand and the speed of the right hand movement so there are no restricted hand shapes and movement trails. It turns out to leave spaces for users to create their own specific gestures in the process of interacting with the system, which could be natural and delightful. The basic gestures of the system are shown in Figure 2.

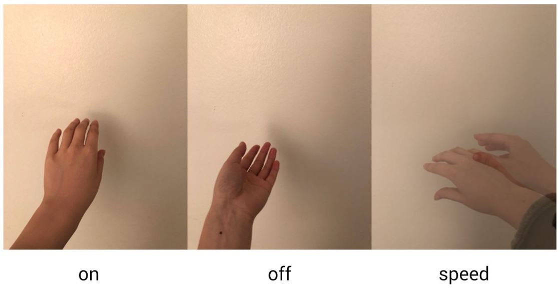

Figure 2. Three gestures for on/off/speed.

Implementation

Based on the high-level design framework, I built the application from collecting the low-level gesture signal by Leap Motion to multimodal interactive user interfaces programmed on Unity 3D, displaying in Figure 3.

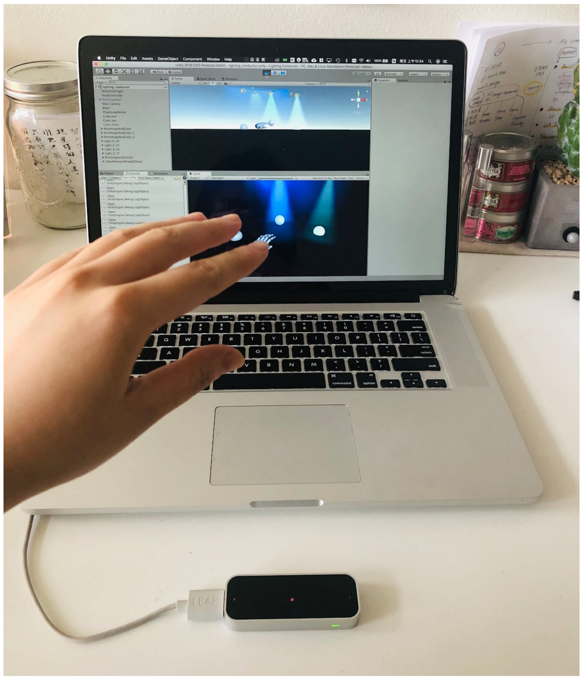

Figure 3. Unity working space and Leap Motion.

For the gesture processing part, Leap Motion provides its own gesture recognition and modeling algorithm. After importing Leap Motion Plugin into Unity, it will generate two rigid hands in the real-time game scene to represent user's two hands detected by the Leap Motion camera. These two hand models provide basic biological info like the right hand or left, which I used to detect which hand the user is using. Plus, detailed hand movement data— velocity for each part of the hand, the direction of the hand and the orientation of palm can be easily calculated through the built-in functions provided by Leap Motion and Unity C# libraries.

The technical logic of the gesture interaction in this game works as follow. First, the camera will differentiate the two hands' movements. For the left hand, it only detects palm direction and orientation to trigger selection and on/off features. Right hand's speed is detected for controlling the playing speed of the music clips and the intensity change of the lights.

The feedbacks from the application are not only audio but also visual. The initial game scene is a dark stage including three lighting and floating spheres which indicate the location of sounds. Users firstly direct their full left hand to the position of the sound they want to play, after doing start or stop gesture, a sound is playing or stopping as well as a corresponding light is opened or closed. Also, the lighting intensity is mapping to the movement range: the bigger range, the more intensive change. Different visual feedbacks are displayed in Figure 4.

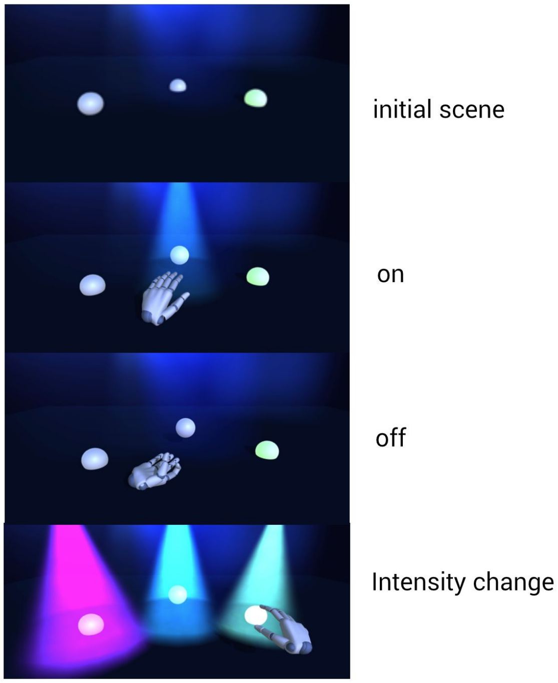

Figure 4. The initial game scene → on/off → the intensity of lighting change

User Study

This application was developed based on a good design intention, I then got five people to test the application, where I observed some interesting behaviors. In each round of user testing, the new user was allowed to explore the application on their own first. One interesting finding is that four of them were using their right hand to control the off/on feature for the first time. However, after I explained the idea of why I separated two hands to accomplish different tasks, they quickly learned and applied to the game playing, which is a good proof of the learnability and usability of the interaction techniques. Another interesting observation is that almost every participant was using different hand movements to control the speed. Even though the gesture detection is not that accurate and sometimes delay, it doesn't hurt how people enjoy the application at all. Some of them were triggered by the intensity changing of the light and changed their movement style correspondingly. One user testing image is displayed in Figure 5. The important takeaway from these observations is that finding a balance between precise gesture input and natural hand expression is crucial to a good natural gesture interface. The designer should consider the consequences of Inaccurate gestures detection while asking users to do some precise gestures. It's absolutely possible that we can turn the inaccuracy into the freedom and naturalness by extracting the most important landmarks in the movement.

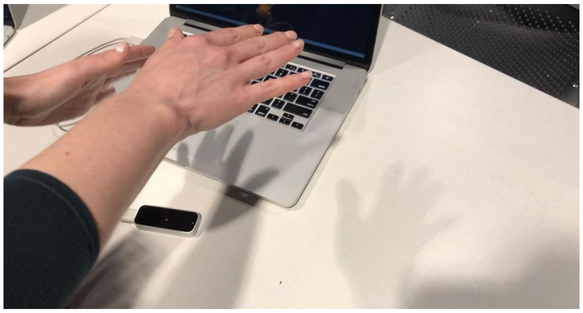

Figure 5. One user is using two hands to interact with the game

Discussion

The goal of this project is to explore how to unleash the power of natural hand movements into general HCI system. And the challenge of designing such natural gesture interfaces is —"There is no such thing as a universal gesture. "—as mentioned in the article Hand Gestures for HCI [1]. In this project, I didn't pay attention to the cultural part of gesture interpretation so it may cause different learning experiences. But instead of staying on the gesture design itself, I digger deeper to understand the mental model beneath these gestures, which could be universal for us as humans. For example, the speed of hand can easily map the speed of music beats, the speed of lighting changes mapping might be a little bit abstract but mostly comprehensive from the results of user study. What's more, another potential improvement for building more natural interactive system would be allowing multi-modal inputs: speech and gesture because gesture sometimes just acts as the complement in the process of communication[4].

This project, I designed two hands interaction techniques which entail dominant hand discussion. I assume that the dominant hand of most users are right hand so right hand in the system takes responsibility of controlling speeds which is the main task of the game while the left-hand deals with the less sequential and low-frequency task like on/off. But from the user study, I found that everyone was trying to use their right hand to control the on/off first and then adjusted to the left hand after my explanation, so maybe providing users with the freedom to choose which hand to control which features is a better solution.

The technical realization level of the project raised an important tradeoff question for gesture interfaces development. Leap Motion can't track hands in a consistently accurate manner. The detection delay and error will cause unexpected usability issues, however, it also brings the opportunity to release the naturalness of human hand movements. Seeking the balance fo the design intention and technique limitation is important for a good gesture or multi-modal interfaces. Fortunately, with the development of machine learning algorithms, the accuracy of gesture detection could not bother designers anymore. However, how to build the natural gesture mental model of your products should still be the first priority questions for designers to answer.

Reference

[1] A. Mulder, “Hand gestures for HCI”, Technical Report 96-1, vol. Simon Fraster University, 1996

[2]U. V. Solanki and N. H. Desai, "Hand gesture based remote control for home appliances: Handmote," 2011 World Congress on Information and Communication Technologies, Mumbai, 2011, pp. 419-423.
doi: 10.1109/WICT.2011.6141282

[3] Maes, Pieter-Jan, et al. "The “Conducting Master”: an interactive, real-time gesture monitoring system based on spatiotemporal motion templates." *International Journal of Human-Computer Interaction* 29.7 (2013): 471-487.

[4] Kettebekov, Sanshzar, and Rajeev Sharma. "Understanding gestures in multimodal human computer interaction." *International Journal on Artificial Intelligence Tools* 9.02 (2000): 205-223.