CS5641 Final Report --- **Leap Motion Hand Gesture Matching**

Ridwan Ahmed Khan, Jinxiang Liu

Abstract:

The basic idea of this project is to recognize hand gestures by using leap motion. Leap motion has some functionalities to recognize some basic gestures. However, in this project, hand motions of one hand will be captured and matched with another hand. For example, right hand of a person is put over the leap motion and goes from one point to another, then another person tries to perform the same travel with one hand at the same time. The movement pattern of two hands will be matched given some error limit. If both hands' movement patterns matches continuously it will play a soothing sound, if it turns to be not matching so much, it plays another a bit harsh and finally for mismatch a harsher sound.For another extension to the sound part, the movement of hand is translated into music notes. So the same type of movement should play the same sound. In this way, it could be easily to confirm that two movements are matched by listening the sounds of both movements. Besides, the hand movements have been modelled graphically by using Unity 3D. When the hand moves over a leap motion, a in-time hand model moving will be shown on screen. Also the matching measurement based on different features will be shown on the screen. Therefore, the user can receive the audio visual feedback.

Project description:

1. Project overview: Describe what your project does in-detail (with minimal technical detail)

The main achievement of this project is matching two movements of hand by using leap motion. Leap motion is designed to track hands. Based on this functionality of leap motion, this project focuses on hand gesture recognition through matching two hand movements. In this project, one hand's motion is captured and matched with the motion of other hand at the same time. Since it is hard to require the movements of the two hands are exactly same in terms of the positions of the hands, then the small errors are allowed when comparing hand motions. Besides developing the program to match two motions of hands, this project models and shows hand movement graphically. Therefore, the user can see their hand motions on screen in time. Since unique sound profiles are added to identify match of hand movements, and the result of the matching is shown on the screen, then the audio visual feedback is given to the user continuously.

2. Implementation: Describe the technical details of your project. If you use a separate section for this, a good name for it might be "Implementation".

Following shows the details about how to achieve each goal in this project.

a. Track hand movements

We chose C# in Unity 3D and Leap Motion SDK to track hand movements. According to the guide of leap motion, the real time data of hand movements was obtained. Such data includes the relative position of palm, the direction and palm normal of the hand movement, and the velocity of the motion. These are all vectors in 3D space.

b. Compare and match movements of two hands

After obtaining in-time data about relative position of hands, the task is comparing and then matching the hand motions. Because two hand gestures can be considered as the same when the exact positions of the hands are not matched but the changes of the hand's places are identical, this project compares the changes of the position rather than the exact place in order to achieve the goal. Considering it is hard to require the movements of two hand are completely matching, the small errors are accepted. In this project, the user can make up any gestures, one hand's motion is captured and then matched with the motion of other hand at the same time, therefore, there is no need to store the collection of the gestures' data to match with. The features based on which match of hands' movements are matched are four vectors - PalmNormal , PalmPosition, Direction and Velocity. PalmPosition represents the center point of the hand in 3D space, Direction and PalmNormal combined gives us the yaw, pitch and roll of the hand. Velocity represents the how slow or quick the hand is moving while doing the gesture. Now mismatch of these vectors of one hand from one frame to next frame is calculated. The same calculation is done for the other hand. The total mismatch of both hands should be the same if both hands are doing the same gesture at the same time. If there is difference between the mismatches of both hands, that is counted as error of that frame. In this way error of all the frame is calculated and at the end it is divided by the total number of frames to calculate the final error. If the final error is greater than a threshold value, then whole gesture is considered mismatch otherwise it is a gesture match of both hands. While calculating mismatch, different feature is given different weights. PalmPosition is given the highest weight , then Distance and PalmNormal and the least weight is given to velocity. The reasoning behind this heuristic is, PalmPosition identify the position of the hand in 3D space, it is of highest priority whereas velocity may suddenly change very rapidly from one frame to another as it is gesture movement , so the total velocity mismatch may turn into a very high value even if the gesture is the same with some variation in speed. That is why velocity is given the lowest weight among the features.

Algorithm in Psuedocode :

Given features: *PalmPosition, PalmNormal, Distance and Velocity of hands*

Calculation:

For each hand in the frame:

For each feature

calculate change in value from the current frame to previous frame

Add all the change values for one hand

Compare the total change values of both hands and add it to total error using following weighted formula:

*Total error = Mismatch in PalmPosition for both hands * 3*

\+ *Mismatch in PalmNormal for both hands * 2*

\+ *Mismatch in Direction for both hands * 2*

\+ *Mismatch in Velocity for both hands * 1*

Per frame total error = Total error/ number of frames

If Per frame total error is above a threshold value then mismatch, otherwise it is a matching gesture ,

c. Model and show hand movement graphically

I used Leap motion Unity 3D version 2 core assets to model the hand graphically. I drew an skybox and in it camera and lights are positioned. The leap motion controller game object is placed in a way so that it is in camera's view point range. Hand models are added to the scene.

d. Adding unique sound identity to hand movement

I used Pure Data to design our sound profiles. From Unity 3D end, I send OSC (Open Sound Control) messages (UDP packets containing some values) to Pure Data. In Pure Data, I wrote a patch which will based on the value transmitted to it from Unity 3D scene will play three different sounds. One sound for if both hands' mismatch of movements are in safe threshold value, another harsher sound will be played if the mismatch value is above safe zone but not yet crossed the threshold value of mismatch and lastly a harsher sound is played when the mismatch value is above the threshold value. So in Unity 3D, I used SharpOSC library to send the OSC messages to Pure Data. In Pure Data, the message is unpacked and the mismatch value is extracted. Based on the value, three different MIDI notes are played.

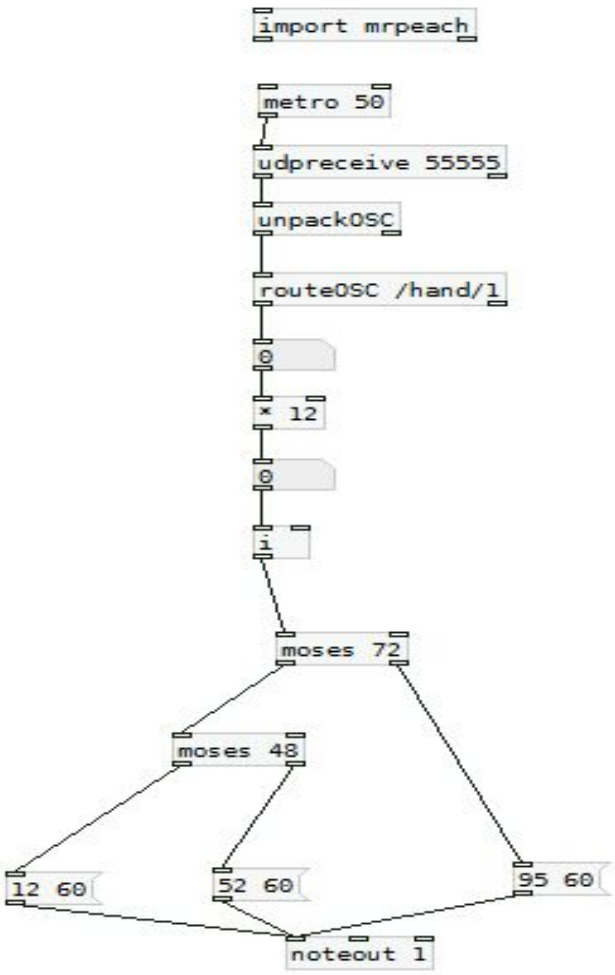


Figure: Pure Data Patch

3. Responsibilities: If you worked on the project with others, how did you split up work? Who did what?

We worked together to achieve the goal of tracking hand movements.In the first phase of the project, Ridwan extracted the leap motion data and showed the data in form application. Then Jinxiang calculated the distance metric of vectors which was shown in the form application. Then in the second phase and final phase of the project, modeling and showing hands in Unity, developing the algorithm for hand matching and coding for sound design was done by Ridwan. Jinxiang tried to help in these phases. It was difficult to develop in these phases as both of our laptops were so old that it cannot handle Unity 3D and Microsoft Visual Studio.

4. Evaluation: What was easy or hard about the project?

When the project was planned it seemed moderately difficult to implement. However, while starting the project we faced with many problems. The first issue, neither of our old personal laptops were good enough to handle heavy applications like Unity 3D, Microsoft Visual Studio. So Ridwan's lab desktop PC was used for development. Then we only had one leap motion for two persons team. Sharing it and completing some task was a bit difficult. For these logistic reasons, distribution of works was very difficult. While developing the project, it was difficult to decide on the development stack. Python was the initial choice but lack of knowledge on how to build GUI in python forced to switch the development stack to C#, Then we developed an WPF application without any hand model which just shows the tracking data. The next wrong decision was to choose C#'s native GUI library to draw hands. As tracking data is overflowing, it was difficult to draw a suitable stable hand. Then the development stack was changed fully to Unity. Part of the C# code of the WPF application code was used, however in Unity, the whole application was written from scratch. Then comes the next challenge which was to find proper hand model that support Leap motion device version 2 which is an older version. So it took a lot of time to find and add a compatible hand model to the scene using Leap Motion Core Assets for version 2. After adding the hand model, deciding on what features of hands to use was crucial. Also how to develop a matching/mismatching algorithm was a challenge. So it took a good amount of time to figure out an algorithm which will match hand movements efficient enough with using heavy arsenal like machine learning. For some time, I thought about using machine learning technique to match two hand movements, however as it is real time hand movement matching application it can cause a heavy burden on the whole application and may perform a little better than the algorithm proposed. So upon deciding pros and cons and time availability for development, no machine learning technique was used. However, I still believe using a well designed machine learning technique will bring out a better result than the proposed algorithm. Another challenging aspect of using the proposed algorithm, was to decide threshold mismatch value. It was decided using lots of trial and error processes. However it is not perfect. Storing frame data of hand movements was another daunting challenge at the beginning of the project which was solved by tracking and calculating mismatch of hand movements at the same time and in an effect nullifying the need of storing frame data. However, still later in the project development, all hand movements tracking data was written in a file along with the result and total mismatch. Now it seems easy

to store tracking frame data of hand movement in a file which will come helpful if we want to apply any machine learning technique in future development of the project.

5. Changes: How does your final project compare with your initial proposal? What changed? What didn't you get complete (it is OK if you don't get every single thing done!)?

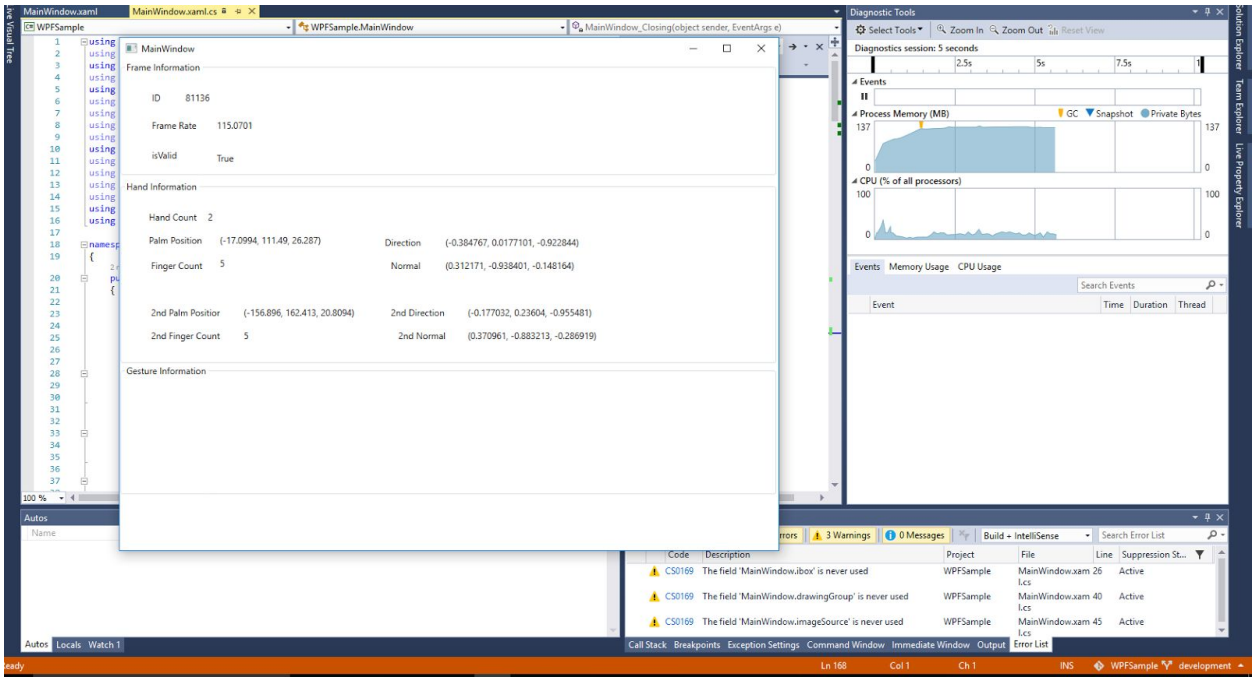
Initially, we wanted to compare two separated movements of hand that happened in different time period. But it was found that by using leap motion the numbers of frames for separated movements are different even their time capacities are same, therefore, how to choose frames that are used to compare become a challenge for us. Then we tried to simplify this task. We decided to capture and then compare the two movements of hand at the same time. Since the data of two hand motions can be compared immediately after they are obtained, then it is not necessary to store a collection of gestures' data to match with, which also become an advantage of our project. Other than this, we stayed on our goals for the project and completed it. May be some features of our application may need upgrades like visual and sound, still the project is ready with all the features stated in the project statement. We also mentioned an ambitious goal of applying machine learning technique in the project which we did not do at the end. But the proposed algorithm showed good result, so it is satisfactory to us.

6. Results: Screenshots, diagrams, photos, links to videos, etc.

a. First phase of the application -

Project Link - https://github.com/ridwanahmedkhan/LeapHandMovement

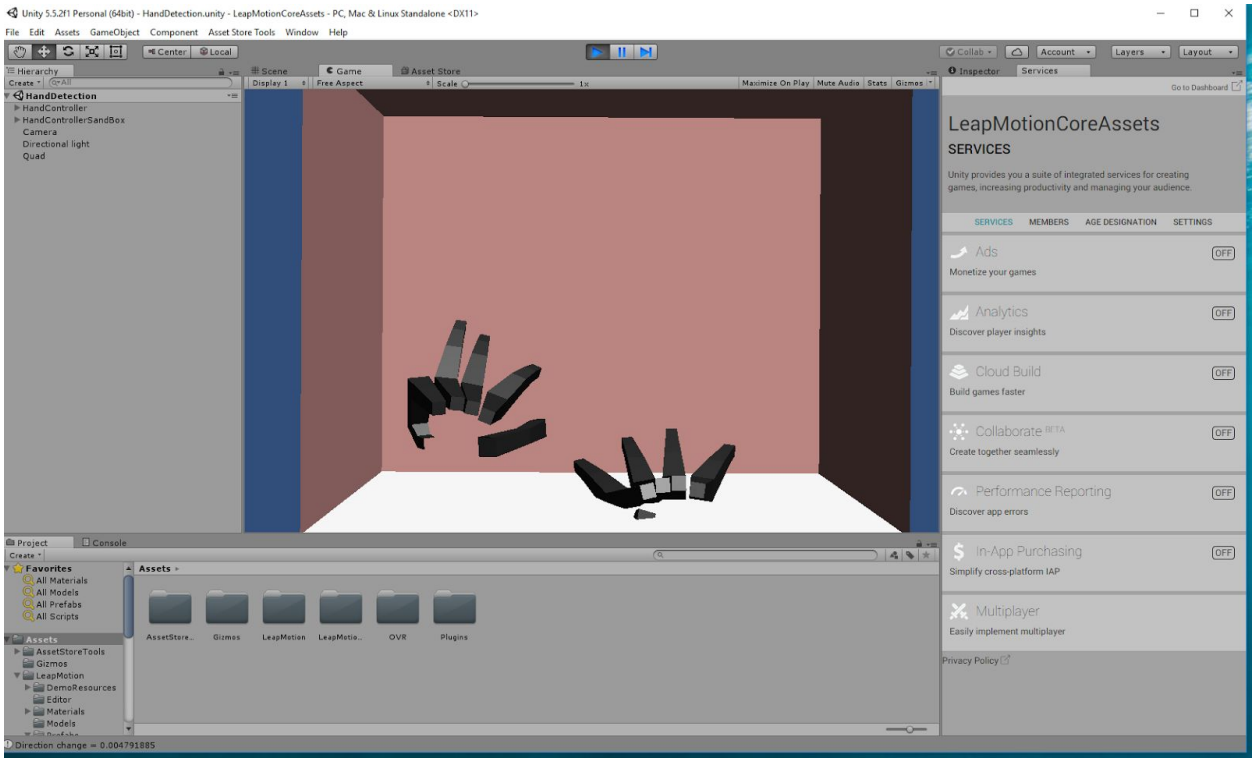
Screenshot -



b. Second Phase of the application -

Project link -
https://github.com/ridwanahmedkhan/LeapMotionUnityCustomHandGestureMatching/tree/master/LeapMotionCoreAssets

Screenshot-



c. Final phase of the application -

Project Link -
https://github.com/ridwanahmedkhan/LeapMotionUnityCustomHandGestureMatching/tree/master/LeapBetterHands

Video link-

https://drive.google.com/open?id=0B-x6yGv_leBwUk5qa0pXN3ZIVVE

7. Discussion: What did you learn from working on the project?

From this project, we have deeper understanding about the functionalities of leap motion and gesture recognition. Since this project also involves the programming of C# and Unity 3D, which we never use before, we gained the experience to develop using those. Also how to design informational audio was learned. In the process, how to use Pure Data and communication to it using network communication was learned. Learning how to develop GUI and functionality at the same time and knead those together was a valuable lesson.The most valuable experience that was learned from working on the project was how to think out of the box to find a simple solution and having a belief that there is solution to even the hardest problem.

8. Future work: Do you recommend that people attempt a project that is similar to yours in future years? Or, what general advice might you give to people who do a project in this class in the future?

It is a very interesting project. To extent this project, it includes following directions. Firstly, a collection of custom gestures library can be built manually which is used for matching user's custom gestures. Secondly, the audio and visual part in the project can be improved. Modeling hand and its movement without any considerable latency in a real life replica for virtual environment could be a option for the future work. Thirdly, although we simplify it, the matching of separated hand movements over a large time capacity still is a good direction to explore in the future. Overall, it is highly recommended to have further work on such topics.

References

[1] Official developer website of leap motion: https://developer.leapmotion.com/

[2] SharpOSC library - https://github.com/ValdemarOrn/SharpOSC

[3] Pure Data - https://puredata.info/

[4] Unity 3D - https://unity3d.com/

[5] Leap Motion Core Assets - https://github.com/leapmotion/LeapMotionCoreAssets