Head Gesture Classification for Video Game Interactions

Samuel Sowell (alex4936@stanford.edu) and Rysen Otomo (rysotomo@stanford.edu)

Often times, when people begin playing a video game, the learning curve can be steep as it can be hard to pick up the nuances of a game. Using head gestures is a simple and intuitive way to have a player interact with the game while ensuring that they understand the current objective. The ability to use head gesture recognition can also provide a more immersive way to interact with in-game non playable characters. For our 3-unit project we implement head gesture classification when using a head mounted display. We display our ability to classify head gestures as "Nod Yes", "Nod No", "No Motion", and "Undefined Motion" in a virtual environment created via Unity. We use two VRduinos. One is used to read raw IMU data from the gyroscope while the second VRduino is used in conjunction with Unity to control the cameras set up in our environment. Both VRduinos were mounted on the front of the HMD.

Figure 1: One Angle of the created Unity environment

In this section, we discuss the head gesture classification algorithm used in this project. Our prediction method relies on the data gathered from the gyroscope on the VRduino. We took the absolute value of the raw gyroscope acceleration readings in the x and y directions at each timestep and averaged them with the previous 24 time steps. Then we checked if the averaged value was above a certain threshold (raw value of 40) in both directions to make our prediction. If the value in the y direction was greater than 40 and the value in x was less than 40 we predicted the user was nodding "yes". We predicted "no" if the value in the x direction was greater than 40 and the value in the y direction was less than 40. If both were less than 40 we predicted "no motion" and if both were greater than 40 we predicted "undefined motion" signifying that the user was looking around and not purposefully making a gesture. We found 40 to be a suitable threshold after running experiments on the gyroscope values and recording the

averages while simulating behavior for “yes” and “no” movements. For updating time steps we used a simple sliding window approach with two 25 value arrays that stored the absolute value the x and y measurements at a given time step. When we proceed to the next time step we simply shifted the values over one cutting off the 0th index in the process and adding the current timestep to the end of the array. We determined that the sliding window approach was a good way to ensure that we would be able to make predictions in real-time. The real-time prediction was printed to the serial monitor during IMU updates. Then, while a user explores our environment, we are able to ask them if they recognize any important landmarks, such as, the red ball or the sand man.

If time had permitted, we would have looked into more advanced head gestures for classification, and we would have implemented a more robust prediction model that would have used machine learning techniques to make classification. While creating our model, we decided to focus on something that could easily make real-time predictions. Unfortunately, executing python code on the VRduino is not possible.

Overall, we were pleased with the performance of our model as we were able to consistently predict the correct head gesture. We thank the EE 267 teaching staff for their help and insight while working over the quarter, and for providing the code necessary for VRduino to interact with the Unity environment.