CMPUT 302: Human Computer Interaction

Team Discover

Daniel Kritsky, Dana Hurlbut, Tri Lai, Tanvir Sajed

Abstract

Recent advances in affordable sensors for 3d skeleton tracking has led to many hobby projects utilizing Microsoft's Kinect in novel ways. Thus, as part of our project in a Human Computer Interaction CMPUT302 course at the University of Alberta we decided to take on the job of improving the user experience with 3d modeling application using the Kinect. Several decisions that we made from the start was to make use of Kinect's skeletal tracking (particularly of the user's two hands) for direct manipulation in the graphical user interface - a better match for the user's mental model of manipulating objects in a virtual world, and to make use of Kinect's microphone combined with Microsoft's Windows Speech Recognition platform as an integral part of our user interface (replacing the traditional menu system). We used Blender for 3D visualization and planned to utilize a CAVE setup combined with a stereopsis setup to better show depth information. For our prototype we decided to implement mainly two features of functionality which would later be tested for performance relative to the traditional mouse and keyboard setup. The first is navigating in a virtual world environment, and the second is manipulating an object's location in one 3d viewpoint, simplifying the functionalities to just the most basic ones. We chose these two features as they were the essential basics for interaction in a virtual environment of which we have to test for user friendliness before continuing to develop and expand our application.

Implementation

As part of our software stack, we split up the functionalities and responsibilities into several different applications communicating with each other via a network TCP connection. We chose to use the TCP protocol for convenience of having a stateful server.

The server included Windows Speech Recognition combined with Kinect Skeletal tracking, which tracks the skeletal position of the right and left hand and continuously sends them over the network with the recognized spoken command in a comma separated value format.

The two clients connected to the server included the Blender game engine for visualizing the virtual world environment, and another separate GUI applications built using Windows

Presentation Foundation. To navigate, the user would have to speak out the word “Navigate” and use their right hand for orienting where to look, and their left for modulating the speed of their forward momentum. For manipulating an object’s location, the user would control the cursor with their right hand, speaking out “Move” to both select and start dragging the object. When dragging an object, the right hand would control the movement of the object within a 2d plane, and the left hand modulated the distance away from the user that the object would be. “Stop” could be spoken at any time to go back to the default mode where user is not controlling the software.

The GUI that we constructed, to be used with the visualized virtual environment, contains the following items:
1) A Blender game window to show the 3D environment
2) The list of usable voice commands on the left side of the screen.
3) The list of animations to show the user how to interact with our interface.
4) A video of depth information extract from Kinect with the user’s recognized skeleton to make the user more aware of how they are utilizing the user interface
5) Feedback to user voice commands included the recognized voice command and the system’s confidence. If highlighted green, then that means the command was recognized.

Statistical Analysis

In our experiments we wished to determine the efficiency, and ease of use between the traditional mouse use and using gesture and voice recognition in a 3D modeling environment. We had two experiments, each of which measured the time it took for each subject to perform the task at hand. To compare blender results with our product results, we used a paired t-test for each experiment.

Navigation

We gave a maze to each subject and timed them for how long it took them to get through the maze using blender, and using our software. Each subject was given a quick run through of controls, and had one run through to get use to the controls. We will test our hypothesis that a person will take longer to navigate the maze in kinect than blender.

$H0$: μkinect = μblender
$H1$: μkinect > μblender

Kinect	Blender
48s	45s
59s	45s

70s	42s
30s	40s
110s	45s
41s	35s

Our p-value is 0.080867, this means our test is not significantly significant. We must reject our hypothesis and accept the null hypothesis. Therefore the speed of the kinect navigation mode is the same as navigation in blender. We can conclude that our software is up to pair with the speed and ease of use with the mouse.

Object Movement

We had a preloaded screen with various primitive objects, and each subject had the task of picking up and moving each object to a target on the screen. As before, each subject was given a quick run through of the controls, and had one run through to get use to the controls. We will test our hypothesis that a person will take longer to put all the objects into the baskets in kinect than blender.

*H*0: µkinect = µblender
H1: µkinect > µblender

Kinect	Blender
210s	83s
200s	60s
230s	63s
105s	55s
165s	93s
101s	94s

Our p-value is 0.00647 this means our test is significantly significant. We must reject our null hypothesis and accept the null hypothesis. Therefore it takes more time to move objects using the kinect that it does using blender. We can conclude from this that our software is not up to par on accuracy when compared with the modern mouse.

Problems

Problems with our software mainly consisted of the inaccuracy and lag of the kinect. Voice recognition would sometimes be inaccurate, particularly for participants with an accent. Response time of a mouse was also much better than the kinect. It also required much more effort for the user to move objects using their arms than it did from just moving their hands on a mouse.

Using two hands was also a problem as it deviated from the standard approach used in 3D manipulation - using just one hand placed on the mouse. Using the right hand was intuitive, but the left was not. Some people had trouble using their left hand for controlling the forward momentum for either object moving or navigation; this broke the user's mental model.

We also had other problem not directly a cause of the kinect. To place the kinect in the CAVE we had to position it relatively high compared to the user; the angle caused some inaccuracy with the skeleton being tracked. We would like to try another experiment without such a big angle offset. Another problem was that our system did not calibrate to users of different heights, causing some problem for some of our shorter participants. Another problem that we sometimes ran into was that at some points the kinect system would stop responding unexpectedly, without any feedback, forcing us to restart the whole application.

Improvements

While experimentally testing our prototype, we had come up with some thoughts into how it could be improved. First of all, the inaccuracy of the speech recognition led to some frustrations during the experiment. We could look into finding out if it is possible to calibrate the speech recognition through training to each participant's' accent.

For our prototype we had a GUI which described which commands the user can utilize by speaking it out, but for our experiment we decided to focus on finding out how efficient direct manipulation using hand positioning compared to a traditional mouse system. To keep a tight focus we had to limit what we were experimenting to just moving objects and navigating. This meant that other features such as rotating or stretching an object were not used, which made the GUI less useful (and not particularly needed).

There was also some belief that part of the cause for the lag of voice commands may be because we send out frequent updates over the network, but did not have any prioritizing metrics for commands versus positional tracking. This might have meant that the clients in the server-client architecture might have been flooded with data, and could not keep up with it (leading to lag).

Conclusions

Our goal for this project was to prototype a system where a user would be able to utilize their actual body to drive a user interface with the hope that it would feel more intuitive than the classical mouse approach. In this respect, the system we designed for navigating a virtual environment could claim some success. While on average it was slightly slower than using Blender with a mouse, it was not statistically significant. This is in contrast with our object manipulation mode which does have statistically significant result to say that it is not as efficient as using a mouse. We also got some feedback to suggest that navigating using the Kinect is more enjoyable. For this reason, our prototyped system may actually be useful for our ultimate goal to attract new users to 3D modeling, but our object manipulation mode must be modified to be more intuitive.