
Gesture Recognition Using Microsoft's Kinect v2

Final Report

Andrew Weinfield

Computer Engineering Major

6/13/14

Table of Contents

Table of Contents	2
Introduction	4
Project Overview	4
Partners/Clients	4
Insights	4
Project Goals and Objectives	5
Goals	5
Objectives	5
Design	6
Hardware Architecture	6
Software Architecture	8
System Integration and Testing	9
User Prerequisites	9
Crossed Arms.....	9
Sitting vs. Standing.....	10
Number of Users in Frame.....	10
Implemented Gestures	11
Swipe Right/Left	11
Grab and Hold.....	12
Fling (Right Hand/Left Hand)	13
Pinch.....	14
User Testing	15

Analysis	16
Conclusion	16

Introduction

Project Overview

The goal of this project was to provide easy to use gesture recognition to be implemented as a subsystem in a COFFEE (Context Observer for Fast Enthralling Entertainment) system. Gesture recognition was achieved using the Windows Kinect v2. The SDK that was provided from Microsoft is only in the development stages so although the entire API is released, not all functionalities are available for use. Software development was done in Microsoft's Visual Studio using .Net programming.

Partners/Clients

The gesture recognition was developed on my own so there are no partners in that sense, but since it is part of a bigger project, I would consider Tony Lenz as a partner anyways. In reality though, Tony is the client that I designed this system for as use in his COFFEE system.

Insights

At the start of the project, the exact gestures wanted, were pretty vague. We (Tony and I) came up with the vision to have swiping gestures, flinging gestures, and a grab and hold gesture. Other than those few basic gestures, I had in mind possible

grab and stretch as well as grab and compress gestures. Besides recognizing when each gesture was detected, some of the gestures like grab and hold also needed to have an x-y position coordinated with them.

Project Goals and Objectives

Goals

- API needs to be easy to work with and implement
- Must accurately recognize when specific gestures are executed
- Gestures need to be easy and fun to use

Objectives

- Create an Event Handler to easily handle each gestures execution
- Gestures must feel natural and have a satisfactory completion rate for all different types of users

Design

Hardware Architecture

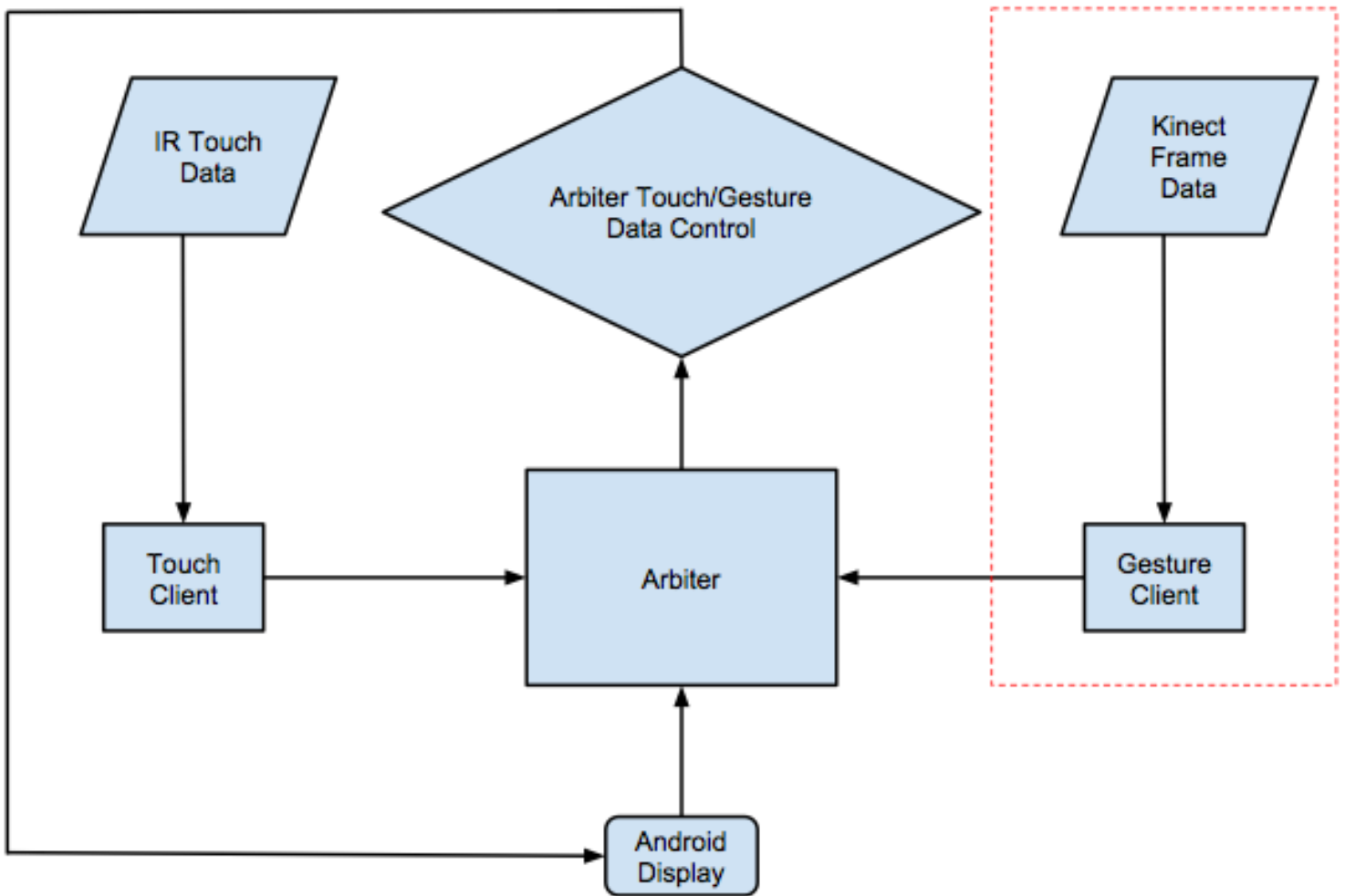


Figure 1: COFFEE Hardware Architecture

Figure 1 above is a block representation of the entire COFFEE hardware architecture. The area outlined with the dotted red line is the part of the system that is responsible for all of the Kinect gesture recognition. The rest of the diagram is a rough description of the entire COFFEE system. The *Kinect Frame Data*

represents the physical Kinect device and each from that is received from the Kinect. The *Gesture Client* is where all the analysis of the frame data is completed. For future reference, I will call a group of frames a “frame collection”. Each gesture is comprised of a frame collection but they are not necessarily all the same size.

Software Architecture

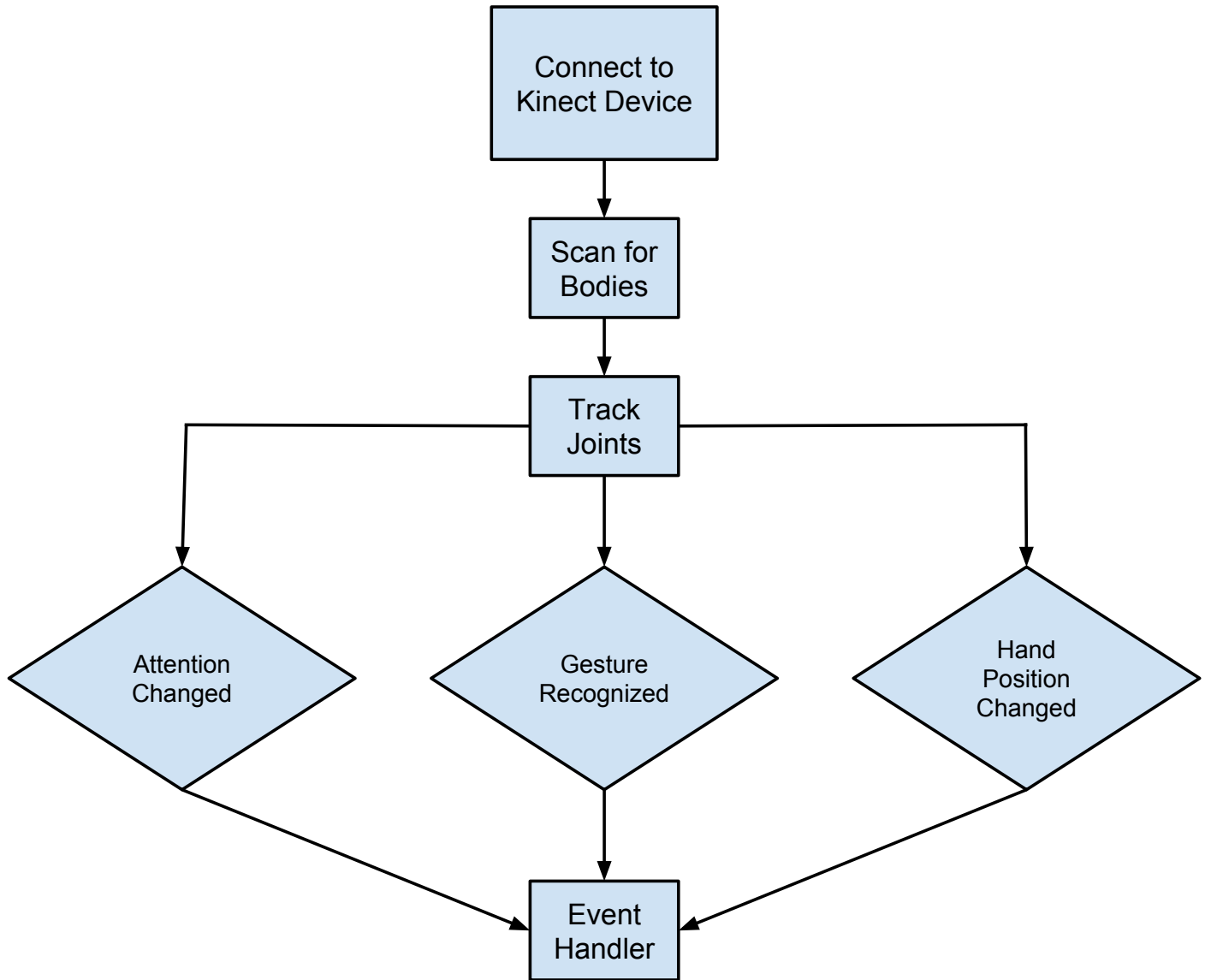


Figure 2: Kinect Client Software Architecture

Figure 2 above is a software architecture block representation of the Kinect Gesture Client only. Once the gesture client as connected to the Kinect device, it

immediately starts to scan for bodies in its field of view. Once the first body has been recognized, frame data is immediately analyzed. The gesture client looks for three things when analyzing frame collections: has a gesture been completed, has the users attention changed, and has the location of the users hands changed. The exact method of determining this will be discussed further later. Once one of these three conditions is true, the appropriate event is thrown to the event handler. This process then goes back to the track joints stage and repeats until terminated.

System Integration and Testing

User Prerequisites

There are a number of required conditions that must be satisfied for the Kinect system to work up to specifications. If all of these are not met, the gestures may not work properly or might not work at all.

Crossed Arms

If the user is sitting with their arms crossed, the Kinect starts to get “confused” and pick up data that is not entirely accurate. For best results, have the user sit with their hands on their lap or to their side.



Sitting vs. Standing

The system works much better when the user is sitting rather than standing. It will still work when standing but all gestures are programmed and calibrated for a sitting user so the results may or may not be up to specifications.

Number of Users in Frame

The system will work if there is only one body in the Kinect's view. This is the requirement that is the most important because multiple people in the view of the Kinect will cause gestures to fire off randomly.

Implemented Gestures

The following is a list of the four gestures recognized by the gesture client. Each gesture description is immediately followed by a picture step-by-step representation of how to properly complete it.

Swipe Right/Left

The swipe gesture requires that you always swipe outside-to-inside. This means that with your right hand you would swipe from right to left and with your left hand you would swipe from left to right. The only other requirement for this is that you must keep an open hand the whole time. This gesture is hand specific so a different event will trigger depending on which hand the swipe is completed with.





Grab and Hold

The grab and hold gesture requires that you start with an open palm facing the Kinect sensor. You then close in your hand into a fist and hold it there. This will trigger the gesture and will track the location of your hand until it moves out of the Kinect's vision, or the closed fist is released. This gesture is not hand specific so there is only one event trigger regardless of what hand the grab and hold is completed with.





Fling (Right Hand/Left Hand)

The fling gesture is pretty straightforward and simple. You must start with your hand close to your body and move it outward towards the Kinect. This gesture is hand specific so a different event will trigger depending on which hand the fling is completed with.





Pinch

The pinch is completed by starting with your hand in a fist except for your index and middle finger which are kept pointing upwards. You then close your hand into a fist, and then open back up to the starting position. This gesture not hand specific so there is only one event trigger regardless of what hand the pinch is completed with.



User Testing

For user tests, a group of students (~30) were invited to come and use the complete functionality of the COFFEE system. There was a pre-planned demo that included both of the touch client and gesture client. For the purpose of this report, I will only be focusing on the gesture part of the demo. The first 15 or so students came from a somewhat more “technical” background while the second half were from a not so technical background. The demo started with getting both the swipe left and right down to navigate through a side-scrolling menu on the TV screen. Once this was achieved, the volunteered test subjects would then select the “picture” menu item by performing a fling with the right hand. Once in the pictures, more swipes were performed to navigate and chose a picture. While looking through pictures and music was playing, the pinch gesture would change to the next song in the playlist. To get back to the main menu, a fling with the left hand triggered the back button. They would then navigate to the “movies” menu item and select that in the same way (right fling). Once in and a movie was selected, they were instructed to perform the grab and hold gesture to move the mouse over the stop button. Once that was done, a successful fling with the right hand would stop the movie bringing an end to the demo.

Analysis

Too much surprise, the non-technical users seemed to pick up the gestures much easier than the technical ones. On average though, the gestures seemed to be something that got better as the users progressed through the demo. There seemed to be a learning curve and once they realized what the system was looking for, they could navigate the screen with much more efficiency.

The feedback about the gesture interface was also very positive. Although it may not of been the easiest to use at the start, the satisfaction that was obtained by being able to control XBMC (Xbox Media Center) though hand motions alone outweighed the negatives.

Conclusion

The project itself turned out to be a great success. The feedback from most of the users was very positive and in both the execution of the system and the concept. A lot of people said they would have been very interested in purchasing one for their home. As far as the gesture recognition itself, that seemed to be the most popular and I feel like it was overall a great success. The goals of having a swipe, fling, and grab and hold gesture were all achieved. The pinch gesture was an added bonus that could be created because of the extra time.

In conclusion this was a very exciting project. I gained a lot of very valuable experience in the programming sense as well as in project development. I am very happy that I decided to undertake this project as my senior project and am glad that the final outcome met and even exceeded the goals that were originally created.