ECE 454 Mobile Computing Lab

Final report

Badger Remote Mouse

Team Member:
Yuye Wang, Xiufeng Xie,
Chun Wang, Po-Jung Chiu

1.Introduction

We start with the problem that, for laptop users, it is inconvenient to carry a mouse around. For presenters, there is not always clicker available when needed and it is expensive to buy one. For programmers and writers, we want to free their hands from constantly moving from keyboard to the mouse. We come up with an idea to solve the problem using mobile device, since smart phone has becoming one item that most people are used to carry around everyday. Therefore, a windows phone application is made, integrated with the solutions in a single piece of software.The implementation of the project last for three months throughout the 2012 fall mobile computing course offered in University of Wisconsin.

3.Background and motivation

Mobile devices have been much more widely used in recent years, especially the laptop computers and the smartphones. Many people, including students like ourselves, bring laptops to school for school work and do presentations often. However, it is very inconvenient to bring an additional mouse with us all the time. Traditional mouse requires a physical surface for use and the touchpads on laptops are small and inconvenient to use. In addition, many people carry their smartphones with them all the time. Therefore, we came up with the idea of using smartphones as mice on our computers.

It is often observed that a programmer or students with heavy typing works need to switch between keyboards and mice back and forth to adjust the cursor positions. While they take their hands off the keyboards, the typing process gets interrupted and lowers the productivity which is already a top concern of the user. Additional, there might be some physically impaired people who are not able to control mice with their hands. Therefore, we came up with a solution, Leg Control Mouse, which allows users to move their cursors around while they hands maintain on keyboards. This reduces the interruption in typing jobs and effectively lowers the time wasted on switching between keyboards and mice.

Another phenomenon we see often at schools is that students have to do presentations often, yet the traditional clicker available on the market is very expensive and not economic efficient for students to purchase them. In addition, using a mouse or having a classmates to control the presentation slides for them is also not an effective way to do presentations, therefore, we came up with the ideas of incorporating Clicker and 3-Dimensional Mouse in our mobile phone application.

Lastly, for those who prefer to use original style 2-Dimensional Mouse and the Touchpad Mouse, we also include them in our application as complementary functions for users who have these preferences.

4.related work

We have researched the applications in current market . Several related applications are found. Such as the “Remote Mouse” app, most of the apps claimes they perform the function of a mouse and can be a replacement ot physical mouse. However, we found these applications’ functions and usage are pretty limited.

First, most of the current remote mouse app are developed under IOS or Android platform, and there are very few ones developed on windows phone. Since we believe, as there are a large number of people using PC running windows operating system, and also there is trend for closer connection of usage among the windows devices, we believe the need to implement such application on windows phone.

Secondly, most of the current remote mouse application only performs the function of a touchpad and keyboard, However, a physical mouse’s functionality is very different from a touchpad. People are more used to hold it and have the sense of moving the whole device around and sense the feel of clicking. Therefore, our application intend to provide the user experience like using a real mouse.

Thirdly, there is no application on phone that can perform the control with 3d direction. Like the wii controller, we found such controlling mode very useful for when doing control in a far distance.

Fourthly, all current applications in market now only have one or two functions, which has no way to fulfill the functionality of a physical mouse in different occasions. Therefore, our app will provide much better usage and functionalities than any of the current applications.

5.System Design & Implementation

The block diagram of our application is illustrated in Fig.1.

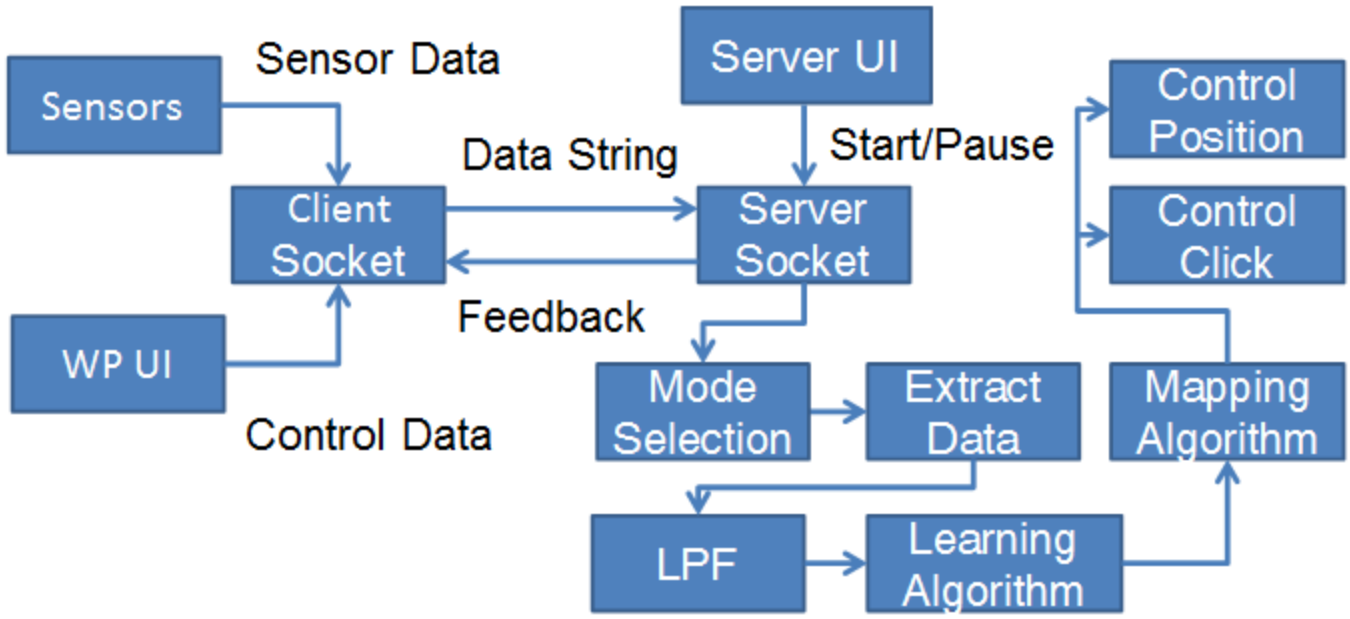

Fig.1 System Diagram

In the following part, we will introduce the design of our system in the manner of operation flow.

5.1 Update Control Data (WP)

WP updates its sensor data periodically (interval between updates is 60ms), including the reading of accelerometer in X,Y,Z-axis and its attitude of Roll, Pitch, Roll.

In 2D and 3D mouse, we also define certain variables to control mouse button click, the value of which will be changed by tapping the displayed button on the touch screen. Before sending data to server, WP will pack current sensor data and other control data such as those variables to mouse button click control into a control data string.

The header of the data string is used as the indicator of the operation mode, different value of int variable at the beginning of the data string is associated to different operation mode:

–1: PPT clicker
–2: 2D Mouse
–3: 3D Mouse
–4: Leg Control
–5: Touchpad

5.2 Send Data to Server (WP)

WP will send control data to PC (server) through Wi-Fi socket connection, in order to ensure the reliability of our application, we use TCP protocol for the transmission. Since UWNet will block our application, we set up the PC as a virtual 802.11 AP.

In order to ensure the sending frequency of control data will never exceed the processing ability of the server, we have designed an interactive data sending mechanism, in which the client will send data to the server only on receiving the feedback from the server about the data transmitted last time.

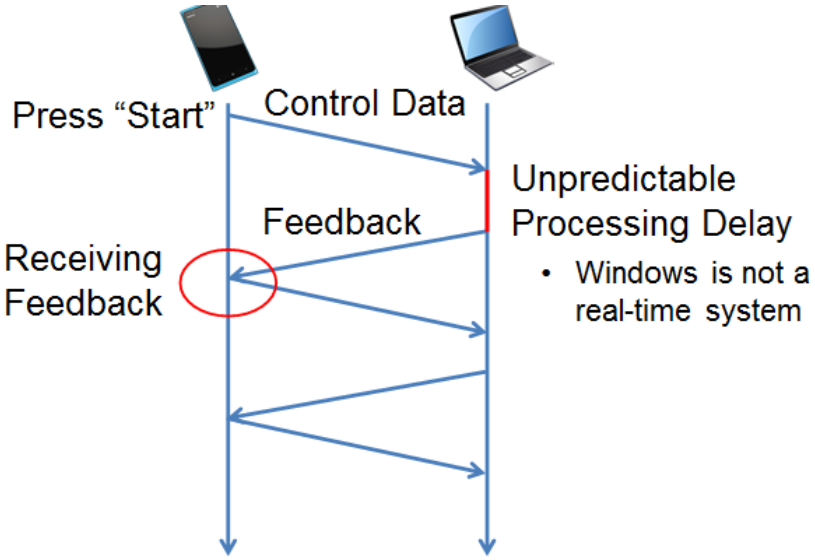

Fig.2 Interactive Data Sending Mechanism

Fig.2 Interactive Data Sending Mechanism

This data sending mechanism is a very important design in our application because it makes the application very stable. In our previous design, WP send control data to the server periodically, since Windows is not a real-time system, the processing time for the control data is not predictable, if the processing time for a control data string exceeds the interval between two control data transmissions, new data will arrive when the program is processing the old data, then the thread will be blocked and the application will crash with the exception “An existing connection was forcibly closed by the remote host socket.”

Note that it is not feasible to solve this problem by increasing the interval between two transmissions or try to transmit less data. From the background knowledge of real-time computation, the worst case execution time (WCET) is always much larger than normal case execution time, therefore the estimated WCET must be very conservative, which

will degrade the performance severely, moreover, even using a very conservative estimation for WCET, the runtime execution time can still exceed this value under some extreme conditions. In our application, we once tried to set the interval between transmissions to 100 ms, which is very conservative and has bad fluency, however, the application would still crash because the unpredictable processing delay in the server side can still exceed 100 ms. In fact, when we use the interactive data sending mechanism, the mouse control are more fluent than the periodic transmission method, because the interval between transmissions has no need to be conservative.

5.3 Extract Control Variables (PC)

The server(PC) will extract control variables from received data string.
–Operation Mode
–Accelerometer: X, Y, Z
–Attitude: Yaw, Pitch, Roll
–Mouse click indicator

5.4 Smooth Sensor Data (PC)

Since in leg control and 3D mouse mode, user will need to lift the phone with part of their body, which is definitely not a stable platform, there will be a lot of unconscious movement, thus in the sensor data there will be a lot of ripples, we need to smooth the sensor data for better mouse cursor control.

We use a first order low pass filter to eliminate the higher frequency ripple signals in the sensor data. In order to save the battery energy of WP, we do this data smoothing in the server side.

5.5 Linear Mapping Algorithm (PC)

5.5.1 Mathematical model

- Cursor location is (x,y)
- x: location of cursor in the x-axis of the screen, 0<x<Width
- y: location of cursor in the x-axis of the screen, 0<y<Height
- Roll: Rotation angle of WP
- Pitch: Pitch angle of WP

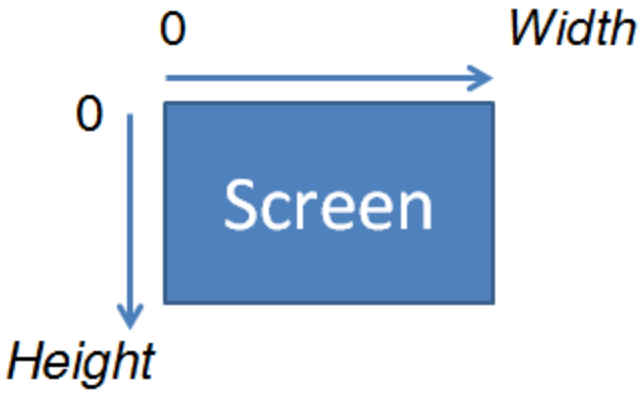

Fig.3 PC screen

5.5.2 Linear Mapping Equation

We use the linear mapping model to get current mouse cursor position (x,y).

$$x = K_x \cdot Roll_{current} + b_x$$

$$y = K_y \cdot Pitch_{current} + b_y$$

In equations above, (Kx, bx, Ky, by) are predetermined constants. We can compute these constants based on the range of sensor data.

$$K_x = \frac{Width}{Roll_{max} - Roll_{min}} \qquad b_x = -K_x Roll_{min}$$

$$K_y = \frac{Height}{Pitch_{min} - Pitch_{max}} \qquad b_y = -K_y Pitch_{max}$$

5.5.3 Learning Algorithm

Hardcoded (Kx, bx, Ky, by) may not be easy to use for different users. Therefore we designed a learning algorithm which can compute the proper constant (Kx, bx, Ky, by) for each user based on the range of sensor data collected in the training phase. Specifically, the algorithm will keep updating the maximum and minimum value of both pitch angle and roll angle of WP during the training phase, as illustrated in Fig.4.

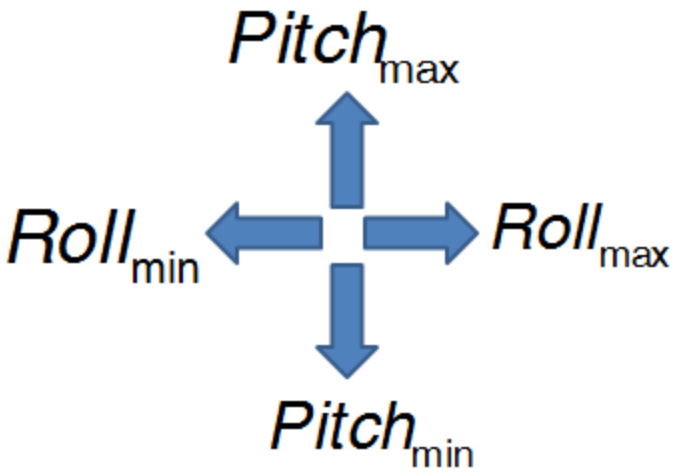

Fig.4 Variables in the learning algorithm

5.6 Set mouse cursor to the position (PC)

You can control the mouse cursor position using windows API. Import “user32.dll”, then it is very convenient to get cursor position and set cursor position. There are also some mouse events defined in this “dll” file, such as mouse button up and down, which is useful when implementing mouse button click function in our application.

5.7 Simulate Mouse Button Click (PC)

We have different ways to simulate mouse button click in different operation modes.

5.7.1 Leg Control

Touch screen is not available when using the leg control mode, therefore we have to use the reading of motion sensors as the control parameter for mouse button click, specifically, we use acceleration on different axis to control click of different mouse buttons.

However, this may cause interference between the control of mouse position and control of mouse button click. In order to alleviate the interference, we set a threshold for mouse button click control, if the absolute value of acceleration on Z-axis (up/down) exceeds the threshold, perform left button click. If absolute value of acceleration on X-axis (left/right) exceeds the threshold, perform right button click. Therefore, it will not interfere with cursor position control as long as you do it slow enough, but it is difficult to determine the threshold value, if the threshold is too large, it is difficult to trigger

click, if the threshold is too small, it will always interfere with cursor position control. After a lot of test, we use 0.3g as the threshold.

We explored 3 possible ways for leg control related method which are proved better than other methods:

Method	Phone location	User Status	User behavior	Easy to use
Thigh Control	Tied over thigh	Sitting	Move thigh	Good
Leg Control	Tied over ankle	Sitting	Move feet	Fair
Stand Control	Tied over ankle	Standing	Move body	Fair

5.7.2 2D/3D Mouse

Touch screen is available when using the 2D/3D mouse mode, we can display buttons on the touch screen of windows phone and use these buttons to control mouse button click. In our application, we use “Tap” event rather than “Click” event because it can better simulate double click of mouse button.

After user click the button on the touch screen, WP will pack mouse button click variable in the data string transmitted to the laptop, then PC will perform mouse click based on the received mouse button click variable.

5.8 Send feedback to WP (PC)

After performing mouse control, PC will transmit a data string back to WP, in normal case, it will transmit “connected”, if mouse button is successfully clicked, transmit “clicked”.

5.9 WP process the feedback (WP)

On receiving the feedback, WP will send current sensor data and other control data to the server. If the content of the feedback means “clicked”, WP will also vibrate as a feedback that the button is successfully clicked.

7.Limitation and challenges

7.1 Selection of Click Threshold

Problem: Simulating mouse button click without interfering with cursor position control. We use acceleration on different axis to control click. However, it is hard to find a good threshold value to determine the click motion. If the threshold value is too large, it will be difficult for the user to trigger clicking. On the other hand if the threshold is too small, the clicking motion may interfere with position control.

7.2 Feedback to User

Problem: As people are used to do sense the clicking of a mouse, such as a physical click feeling, the sound of clicking. However, for a touch screen phone, there is no way to provide the feeling of a real button.

Solution:Therefore, we set the a vibration for the button click. After a mouse button click, server sends a "clicked" signal back to WP. And after receiving the "clicked" signal, WP will vibrate.

7.3 Smooth Data

Problem: You thigh is not a stable platform. Sensor reading will ripple

Solution: A first order low pass filter is used to smooth sensor data. Eliminate the rippled caused by the inadvertent movement of users. However, using only first order mathematical equation can only provide a limited data smoothing. High orders of mathematical equation would improved the data smoothing and mapping performance. But we don't have enough time to do research on this and it is out of the scope of this class as well.

7.4 Precision of the sensor

Sensors equipped on Windows Phone are not precise enough.Use integration of acceleration, we cannot get current velocity of the phone.Causes unbalanced integration in 2D mouse control.

7.5 Wireless communication

Problem: client app sends data to server on timer-based communication sometimes causes the server to crash.

Solution: we change the communication protocol to have the client send data to the server only when it has received the server's response. In this design, the data transfer rate between the client and the server would not exceed the processing limit of the server.

8.Lessons and insights

The most significant takeaways from the project are the idea fast, simple, stable and user friendly.

Fast. For device like this remote mouse, which requires very high level of accuracy and calculation, we need to make sure it runs fast and smoothly. In order for it to operate fast, we put all the calculation and mapping at the server side, which reduces the memory uses on the phone. In this way, we also get to save the power of the phone to make it last longer.

Simple. A good application should be never complicated to use. We set the user interface to look simple but elegant, which give the user a sense of easy using. The flow of the usage are very simple and smooth, the panorama page is a good way leading the user through the setting and using it. Every user should be able to start using it without read though any instructions.

Stable. For such a utility device, the stableness is very important. We spend lots of time on testing and debugging in order to provide a stable usage of the application. These includes the network connection, the gesture capturing and making sure the application does not end unexpectedly.

User friendly. In order to adjust the habit of different users, we set the learning algorithm which learns the comfortable position of each user. Also, for different usage. The five functionalities are set and people can easily switch between them.

For further development, we will enhance our algorithm for smoothing the cursor control. And also thinking of other wireless connections as blue tooth as a way to save energy. However, for windows 7, the bluetooth API is not available. Therefore, we might switch to windows 8 operating system.

8.Self-evaluation and individual contribution (each person write your own)

8.1Yuye Wang

First, I worked on the touch pad module of the remote mouse. I did the touch gesture detection and the cursor position mapping. I came up with two mapping methods. One is to map the cursor position on the computer relative to the touch position on the phone. The second position is to move the cursor corresponding to the touch movement difference. I choose the second methods for the final touch pad module implementation, since it will be hard to click on specific spot due to the size of the touch screen, if using the first methods. Also, I multi touch function is implemented and a maximum of eight touch points can be traced in on the silverlight system. However, since such multi touch function is not necessary for our touch pad model, I eliminate the touch method to single point detection in order to increase the stability of

our app.

Second, I worked on the design and implementation of the user interface. I designed the process for using the app and I also drawn the background image and icons. I learnt the panorama, pivot and plain page implementation in windows phone. Also, added application bar which makes a smooth flow throughout the whole program. I did the integration of the each functions into the app.

Third, I set the main login page of the program, since one of the biggest change from our second stage and final stage is that we only need to connect the computer once for the app other than several times for every module. In order to prevent the user's confusion of starting modules without right connection, I set the feedback from the setting page, to make sure the user can only clicked into each section after getting good feedback from the server.

Fourth, as the leader of the team, I got chance to communicate with each team members and set up meeting and keep track of our progress. Lead the discussion from brainstorming, implementation, report writing and presentation. I also did the demo recording and editing for the final presentation.

8.2Xiufeng Xie

First, I finished the prototype of the system, including core modules such as getting motion sensor data of WP, socket communication between WP and PC, and control data processing in PC side.

Second, I worked on the 3D mouse and Leg Control function of our application. Designed the linear mapping algorithm to convert motion sensor reading to mouse cursor location on PC screen. I also did some exploration about the optimal place to put the phone when controlling the mouse cursor without using hands.

Third, I worked on the UI of the PC side program. Since windows form application is more user-friendly, we use this as the user interface. At the same time, we also have a windows console to display control data, just for debugging use. Since socket communication is an infinity loop, we have to use multi-thread in this program.

Fourth, considering the fact that different user will generate sensor data of different range because they may have different body structure and habits. Note that some

parameters in the linear mapping algorithm are related to the range of sensor data, the parameter values convenient for one user may not be easy to use for another use. In order to solve this problem, I designed a learning algorithm, WP will collect and learn the range of sensor generated by user in the training phase, then it will use the learnt data to determine proper parameters for the linear mapping algorithm, the user will choose when to stop the learning algorithm and switch to the linear mapping algorithm by clicking a certain button.

Finally, I designed the Interactive control data sending mechanism and solved the unexpected crashing problem in our application. Based on the background knowledge of real-time system, I realized periodical control data sending is the main reason of the crash because the worst case processing time for the control data is unpredictable, then I used an interactive way to solve this problem.

8.3Chun Wang

I worked on the clicker module of the remote mouse. Most of the function shares the same code and structure as 3D mouse. The Next-PPT button uses the left mouse click function and the Previous-PPT button uses the left-arrow key on the keyboard. I modified the server prototype introducing the parts for choosing modes for different function that the client would use. I also did the first and last integration of the app. Throughout the entire semester, I help other teammates on debugging and testing out individual modules. I also did some optimization when finalizing the app.

8.4Po-Jung Chiu

I mainly work on the 2-D Mouse functionality of our application and it was quite challenging due to the physical constraints of the sensor hardwares as well as the precision of the outputted data. Therefore I have to try many different ways to produce the best results in this functionality. In clicker and touchpad functionalities I've provided some help in developing stage. In other stages of our application, I provided some insightful suggestions and some possible solutions while performing functional tests on the performances of other tasks.

8.5 Overall

Overall our group worked as a very good unit, everyone got clear goals and contributed a lot throughout the program. Everyone has passion in our program and has put lots of effort and realized our idea from scratch. We are glad that our final product meet our original expectation and is able to perform all functions we came up when we confirmed our idea at the beginning. The successfulness of the project

management is that everyone got the chance to do what he/she is good at and make efficient progress and contribution towards the project. This course brought us together and we enjoyed the progress of the project development.