Virtual Wiialtiy - Redux

Multimedia 4B03 Thesis Outline

Joel Clemens

For
Prof. Rockwell,
Prof. Hamilton.

Monday January 21, 2008.

Proposal.

The concept of virtual reality is not a new one, but even so, why has this field been so un-tapped and limited to large scale vendors. The reason, I believe, is accessibility; VR equipment can be very expensive for the average home user. This is the main goal of this project; however, I will try and go beyond average head tracking into full motion tracking as well. Free software and open source programs plus common latest generation game console controllers will be used to obtain affordable results.

The Nintendo Wiimote was used in previous research and development, as its motion sensing and IR dot tracking seemed to be useful. However, I concluded that it is impossible to obtain 'yaw' type motion (turning left/right) using the motion sensing or IR tracking the wiimote has available. Thus I migrated to the PS3 controller which instead of IR tracking, contains a gyroscope that allows the tracking of yaw. (The title of this project is tentative, hence the 'Redux')

This would allow anyone with a head mounted display, to convert it to full head tracking. The cost of this has been found to usually start around 800$. It is still being developed, but I have already enabled the 50$ PS3 Sixaxis controller to provide Yaw, Pitch and Roll tracking. This is ideal in for immersion in flight simulators, and driving type games. Depending of the software, it can also be used to expand one's monitor to the size of an entire room.

Upon further development, using either the accelerometers or wiimote sensor tracking, walk motion tracking should also be able to be possible. This could provide an enriched augmented reality scenario. The final goal is to have the user walk along a narrow beam across a virtual abyss. But in the real world, there is only the narrow beam on the floor. This 3d virtual area can be designed for almost any 3d game engine.

Outline.

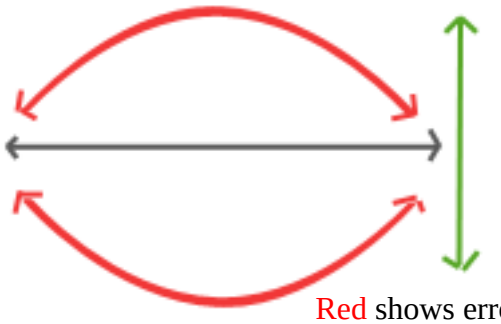
Due to hardware limitations of the wiimote, this project will now be relying on the Sony Playstation 3 console controller known as the PS3 Sixaxis. This controller has the same motion detection as the wiimote, being accelerometers for forward/backward, side and up/down motion. However the one critical difference is that it has a gyroscope that can detect rotation on a vertical axis, known as yaw. The sixaxis also includes 2 joysticks that can also be used in this project. The name sixaxis comes from the ability to detect up, down, left, right, forward and backward motions, thought it can detect much more than that. Though the PS3 controller also uses Bluetooth wireless technology to connect to the console, unlike the wiimote, it is not readily compatible with PC Bluetooth drivers. Currently users are able to interface the controller with a common USB cable. Wireless communication has been only successful on Linux systems, (because of the open source nature of Linux Bluetooth drivers). Wireless may be possible in the future for windows machines, but for now, having a cable will not defeat this project.

The program that is used to manipulate the inputs of the controller is called GlovePIE. It is not open source, but it is free to use. Its programming language is basic and easy to understand. It runs the script in a constant loop at a default rate of 40 times per second. Since most games use a mouse to look around, the motion detection from the PS3 controller can be remapped onto the mouse driver. Below is a simple script that translates motion from the sixaxis onto the mouse. Rolling is an option that I have only been able to control with a joystick axis, more explanation to come.

```
//1024x768 resolution in OpenArena results in 1024 Y mickies and
//2048 horizontal mickies (a mikie is a pixel unit for DirectX)
//ratio is 1024 mickies per 180 degrees (5.6889)
//ration  2048 mickies per 360degrees (5.6889)
pie.FrameRate = 70 //increases the speed of GlovePIE to 70Hz
//calibration indicator (light turns on when yaw is successfully calibrated
if abs(sixaxis.Yaw) >1
  sixaxis.led1 = 0
else
   sixaxis.Led1 = 1
endif
//calibration - set controller still on flat, LEVEL surface, and press circle btn
//see glovepie documentation for rawgyro explanation
If sixaxis.circle
  sixaxis.RawGyroZero = smooth(sixaxis.RawGyro)
  sixaxis.yaw = 0
  var.rolloffset = smooth(RemoveUnits(sixaxis.smoothroll) , 5)/70
//the 70 is for 70degrees of motion that I want mapped to the joystick axis
//(which ranges from -1 to 1)
  var.pressed = 1
endif
//only executed if calibrated and button is not being pressed
//this is what actually reads the data from sixaxis
if (var.pressed) & (sixaxis.Circle = 0)
  mouse.DirectInputY = (smooth(RemoveUnits(sixaxis.smoothPitch*-568) , 5))/100
  mouse.DirectInputX = (RemoveUnits(sixaxis.yaw*568))/100
endif
ppjoy.Analog0 = smooth(RemoveUnits(sixaxis.smoothroll) , 5)/70 -var.rolloffset
```

Roll is important to have for proper VR, tilting the neck shoulder to shoulder is something that occurs naturally even if only a few degrees. When one's head is tilting and the image on the HMD doesn't match, it can create motion sickness. I have found an open source game engine that had code for controlling the roll of the view with a joystick axis. Modifying the source code and recompiling allowed more direct control with the PS3 controller; however motion at the extremities is very shaky.

Even though the script above does fully track 360 degree motion, my calculations are too simple to include the extra information necessary for linear motion while the controller is angled up or down. The script that I wrote for GlovePIE can be quickly tested with the desktop cursor behaviour while the script is running. See illustration below for clarification.



Black indications cursor motion while changing the yaw left/right

Green shows cursor motion when changing the yaw of the controller

Red shows erroneous cursor motion that occurs when the controller is tilted up or down while changing yaw left/right at the same time

This is occurring because the way pitch and yaw are calculated. They use gravity to determine which way is down according to the associated axis. However, the direction of gravity still needs to be included in a non linear calculation to make sure that the cursor stays straight when turning the controller left or right with a non-zero pitch.

Accomplishing a linear tracking system is a significant milestone. As this should match existing head tracking technology, it is enough to warrant a working project that can be submitted. Once this has been achieved, the next step is to create a custom map within a game to test the interface, and practice map generation skills. It will be a car driving map with a circuit roadway with minor obstacles. Valve's Half Life 2 engine comes with a vehicle that allows the user to drive steer and look freely 360degrees while seated in the car. Unfortunately, at this time, there does not seem to be a way to control the roll of the view directly.

Once the map and VR system performs to an acceptable level, further experimentation with motion tracking can then be resumed. The simplicity of the final map goal of simply walking in a straight line is intentional. Calculations required to get linear motion tracking from the accelerometers may be too difficult to develop within the time limit. It is possible to use the Wiimote to track motion within the confines of the

area to be walked on. This could be done by having the Wiimote stationary and set it up to track IR sources on the user. Johnny Chung Lee has had success with this mode of tracking. (see http://www.cs.cmu.edu/~johnny/projects/wii/ for more information)

If proper motion tracking cannot be achieved in time enough to develop the walk-across-abyss scenario map, then the driving course map will have to be submitted as the final project.

Time allotment

1 week – Obtain PS3 Controller, test and experiment with Pitch and Yaw

completed

2 weeks – find open source 3d shooter environment that have potential to enable roll control, modify code and experiment

***completed* Now capable of basic demonstration**

2 weeks – modify GlovePIE script for improved linear head tracking

in progress

2 weeks – build and test map for driving course.

pending

3 weeks – experiment with any potential motion tracking methods

pending
1-2 weeks – build abyss map and test.
pending
According to this time allotment, this should lead right up to the end of March in time for submission and presentation.

Appendix

Example sources of both Head Mounted Displays and existing head tracking technology
http://www.vrealities.com
http://www.i-glassesstore.com/

Sony Playstation 3 SIXAXIS controller being used wirelessly on modified Linux system
http://www.pabr.org/sixlinux/sixlinux.en.html

GlovePIE - used for creating scripts for the wiimote and PS3 controller
http://carl.kenner.googlepages.com/glovepie

Nexiuz, free open source 3d game based on the Quake engine,
I was able to modify this game to accept a roll command input
http://www.alientrap.org/nexuiz/