GyroTouch: Complementing the Multi-Touch Display

Francisco R. Ortega **Armando Barreto** **Naphtali Rishe**
Malek Adjouadi **Fatemeh Abyarjoo**

Florida International University, Miami,FL.
forte007@fiu.edu, barreto@fiu.edu, ndr@acm.org, adjoudi@fiu.edu, fabya001@fiu.edu

ABSTRACT

We present our findings to yield a more intuitive user interaction for navigating 3D worlds. GyroTouch was created to augment multi-touch gestures with other devices. In this paper, a multi-touch display is complemented by a gyroscope to enhance 3D navigation. Our approach uses multiple to sensor to address 3D rotations and 3D Translation used in Navigation. This work also includes a two sample algorithm for estimating angular velocity for any given axis.

Author Keywords

Multi-Touch; MEMS; 3D Navigation, 3D Interaction, HCI.

ACM Classification Keywords

H.5.2 Input devices and strategies: Interaction styles

INTRODUCTION

The emergence of new widely accessible input technologies (e.g. Microsoft Kinect, Leap Motion, iPad) in the past few years has created opportunities to improve user interaction. When developing gestures to aid the navigation of 3D worlds, one can find several methods, including multi-touch interaction. However, one of the primary limitations of multi-touch displays is the 2D nature of their surface. This limitation can be circumvented by creating custom 2D gestures to map the 3D equivalent input actions. In pursuit of a more realistic 3D experience, augmenting or complementing the multi-touch display gives the users a more natural interaction. We have explored the development of a fast, natural and accurate real-time 3D navigation technique using multi-touch by augmenting (or complementing) it with commodity devices (e.g., Nintendo WiiMote, Leap Motion), in order to find a more intuitive user interaction. This work is aligned with the $1 algorithm[7] and the Rubine algorithm[4] in their attempt to show how simple solutions using commodity devices can be as efficient as more complex algorithms such as Hidden Markov Models. We showed an algorithm in [3] which demonstrated a simple approach to recognize some touch gestures.

The proposed solution uses standard multi-touch gestures (e.g., rotate with two fingers) and a Microelectromechanical System (MEMS) that has a 3-axis Accelerometer, a 3-axis Gyroscope and a 3-Axis Compass. This approach can be used to complement the multi-touch displays which may lead to a better user experience.

The work is novel when it comes to complementing multi-touch devices with gyroscopes. The ability to use additional sensors in a 'watch' wristband is realistic and not invasive. In addition, Algorithm 1 uses simple computations to estimate the rotation angle for any given axis using only two samples received from the MEMS module.

RELATED WORK

Multi-touch interaction for 3D environments has been explored before for domain-specific applications (e.g., [1]). There have also been attempts to augment the multi-touch experience. For example, Z-Touch[5], developed by Takeoka et al., captures a depth map to add the z-axis to the touch display. However, the Z-Touch has limitations and it is not a commodity device. Augmenting the touch with a force sensor has also been tried[2]. Vision has also been used to complement touch with Microsoft Kinect [2]. Similarly there are Apple iOS or Android OS devices that allow users to combine touch with some MEMS components (e.g., Accelerometer).

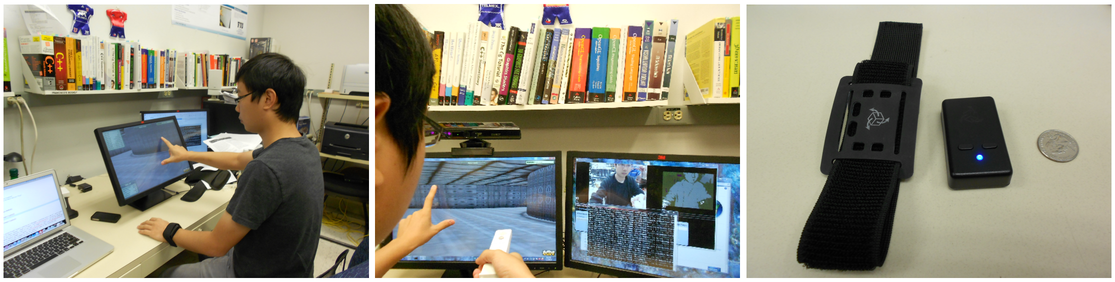
GYROTOUCH

We have developed **GyroTouch** using Visual Studio running on a Windows 7 platform with a 3M 22" multi-touch display and a MEMS module by YEI Technology (3 Space Sensor, shown in Figure 1c). Currently, we are using the 3 Space Sensor Wireless in the non-dominant hand of the user, as shown in Figure 1a. For our 3D rendering, we have used OpenGL and OGRE3D (OGRE3D engine shown in Figures 1a and 1b). We believe that combining both devices gives us the freedom to use our hands for some of the rotations and translations, while keeping the tactual feedback intact for other gestures.

GyroTouch allows the combination of multi-touch and MEMS devices. There is a mature project called VPRN[6] that is designed for Virtual Reality input devices. However, **GyroTouch** differs from VPRN because our aim is two-fold: First, we are designing a system that is more usable for a variety of desktop environments; second, our approach is designed with usability testing in mind.

The approach is to use our touch algorithm to detect swipe, zoom and rotate gestures for the touch. This allows us to use the touch for translating in x and y (using two-finger swipe). For z translation, we map a one-finger swipe (same direction as the y-axis) to the z translation. For the rotation about the z-axis (yaw), we use the 2-finger touch rotation as it is commonly done with touch tablets. In order to keep the interaction as natural as possible, we complement the touch with the gyroscope for rotations about the x-axis (roll) and the y-axis (pitch) using the gyroscope.

Our touch detection method, described in [3], consists of finding certain characteristics for each gesture using a very fast and simple algorithm. The gyroscope found in the MEMS, shown in Figures 1a and 1c, is used only to indicate the roll



(a) 3 Space Sensor (b) WiiMote (c) 3 Space Sensor & Strap

Figure 1: GyroTouch

and pitch rotations, where as the third rotation is indicated via the touch interaction. Algorithm 1 shows the integration over time of the gyroscope signal, using the current and previous samples, required to obtain the angle of rotation about the x-axis. The same applies for each of the other two axes. The sensor already provides data processed by a Kalman Filter. In addition, we filter data within a threshold, calculated when the sensor is initialized, in an idle position. The sampling rate varies depending on the sensor. We are using a sampling rate of 160 Hz (with a possible maximum for this device of 800Hz). This gives us the period ***T*** to be $1/FS$ or $1/160$. Since the data is already normalized, there is no need to use the ***midLevel*** and the ***unitDegree*** variables in Algorithm 1. Hence, we set them to 0 and 1 respectively. It is important to point out that this is not always the case for all devices. For example, for the WiiMote (as shown in Figure 1b), it is necessary to set the midLevel at 2^{13} and the unit degree at $8192/592$.

Algorithm 1 Rotation Algorithm for a Gyroscope

Ensure: midLevel=0 & unitDegree = 1 for 3Space Sensor

$roll \leftarrow rawdata.roll - midLevel$

$rot.x[0] \leftarrow rot.x[1]$

$rot.x[1] \leftarrow roll$

$omega.x[0] \leftarrow omega.X[1]$

$omega.x[1] \leftarrow roll.x[1]/unitDegree$

$x \leftarrow angle.x[1]$

$angle.x[1] \leftarrow x + T * ((omega.x[1] + omega.x[0])/2)$

return angle.x[1] as roll

CONCLUSION AND FUTURE WORK

We have shown a simple method for real-time 3D Navigation using multi-touch and gyroscope devices. Our next step is to provide additional gestures for new devices including multi-touch displays and MEMS. We will look at combining multiple devices to create a fusion algorithm to enhance the user experience. For example, how can the accelerometer, gyroscope and compass serve to improve the touch interaction? We will use our **GyroTouch** to evaluate user interaction when dealing with 3D navigation. Our experiment design will test different devices with GyroTouch, including multi-touch, gamepad and Leap Motion. We plan to provide our work to the community once is completed.

ACKNOWLEDGMENTS

This work was sponsored by NSF grants HRD-0833093, CNS-0959985, IIS-1052625, OISE-1157372, IIP-1237818, IIP-1330943. Mr. Francisco Ortega is the recipient of a GAANN fellowship, from the US Department of Education and the McKnight Dissertation Year Fellowship.

REFERENCES

1. Fu, C.-W., Goh, W. B., and Ng, J. A. Multi-touch techniques for exploring large-scale 3D astrophysical simulations. In *CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Request Permissions (Apr. 2010).

2. Lai, H. Using Commodity Visual Gesture Recognition Technology to Replace or to Augment Touch Interfaces. In *15th Twente Student Conference on IT* (2011).

3. Ortega, F., Barreto, A., Rishe, N., Adjoudi, M., and Abyarjoo, F. Poster: Real-Time Gesture Detection for Multi-Touch Devices. *IEEE 8th Symposium on 3D User Interfaces* (2013), 167–168.

4. Rubine, D. Specifying gestures by example. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (July 1991), 329–337.

5. Takeoka, Y., Miyaki, T., and Rekimoto, J. Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. In *ITS '10: International Conference on Interactive Tabletops and Surfaces* (Nov. 2010).

6. Taylor, R. M., II, Hudson, T. C., Seeger, A., Weber, H., Juliano, J., and Helser, A. T. VRPN: a device-independent, network-transparent VR peripheral system. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology* (Nov. 2001).

7. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology* (Oct. 2007).