

REALNAV: EXPLORING NATURAL USER INTERFACES FOR LOCOMOTION IN
VIDEO GAMES

by

BRIAN M. WILLIAMSON
B.S. University of Central Florida 2005

A thesis submitted in partial fulfillment of requirements
for the degree of Master of Science
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2009

Major Professor: Joseph J. LaViola Jr.

© 2009 Brian Williamson

ABSTRACT

We present an exploration into realistic locomotion interfaces in video games using spatially convenient input hardware. In particular, we use Nintendo Wii Remotes to create natural mappings between user actions and their representation in a video game.

Targeting American Football video games, we used the role of the quarterback as an exemplar since the game player needs to maneuver effectively in a small area, run down the field, and perform evasive gestures such as spinning, jumping, or the “juke”. In our study, we developed three locomotion techniques. The first technique used a single Wii Remote, placed anywhere on the user’s body, using only the acceleration data. The second technique just used the Wii Remote’s infrared sensor and had to be placed on the user’s head. The third technique combined a Wii Remote’s acceleration and infrared data using a Kalman filter. The Wii Motion Plus was also integrated to add the orientation of the user into the video game.

To evaluate the different techniques, we compared them with a cost effective six degree of freedom (6DOF) optical tracker and two Wii Remotes placed on the user’s feet. Experiments were performed comparing each to this technique. Finally, a user study was performed to determine if a preference existed among these techniques.

The results showed that the second and third technique had the same location accuracy as the cost effective 6DOF tracker, but the first was too inaccurate for video game players. Furthermore, the range of the Wii remote infrared and Motion Plus exceeded the optical tracker of the comparison technique. Finally, the user study showed that video game players preferred

the third method over the second, but were split on the use of the Motion Plus when the tasks did not require it.

Dedicated to my family and friends, they were always there to give me support and motivation.

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Joseph J. LaViola and the Interactive System and User Experience (ISUE) lab at the University of Central Florida for their support in this effort.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION.....	1
DEFINITION OF PROBLEM.....	2
HARDWARE DEFINITION	4
CONTRIBUTIONS	7
CHAPTER TWO: LITERATURE REVIEW.....	9
CHAPTER THREE: TECHNIQUES	15
RUNNING TASK AND EVASIVE TASK.....	15
KALMAN FILTERS	16
Position/Velocity/Acceleration Kalman Filter.....	16
Extended Kalman Filter	18
TECHNIQUE ZERO: COMPARISON WITH OPTICAL TRACKER	23
TECHNIQUE ONE: ACCELEROMETER ONLY METHOD.....	26
TECHNIQUE TWO: HEAD TRACKING METHOD	29
Integration with Wii Motion Plus	31
TECHNIQUE THREE: HYBRID METHOD.....	33
SOFTWARE AND COMPUTER USED	34
CHAPTER FOUR: ANALYSIS AND RESULTS.....	35

METRICS DEFINITIONS	35
ACCURACY DATA	37
RANGE DATA.....	38
EVASIVE RECOGNITION ACCURACY	42
MOTION PLUS ORIENTATION ACCURACY	43
USABILITY STUDY	46
Subjects and Apparatus	46
Experimental Task	46
Experimental Design and Procedure	48
Performance Results	50
Subjective Results	51
CHAPTER FIVE: DISCUSSION.....	54
CONCLUSION OF TECHNIQUE ANALYSIS	54
MOTION PLUS DISCUSSION	55
GENERAL OBSERVATIONS.....	56
FUTURE WORK	57
CHAPTER SIX: CONCLUSION	59
REFERENCES	I

LIST OF FIGURES

Figure 1 - Wii Remote Axis (picture courtesy of ISUE lab at UCF).....	5
Figure 2 – Natural Point TrackIR infrared tracker.....	7
Figure 3 -TrackIR attached to user	24
Figure 4 - Wii remote attached to user's leg	24
Figure 5 - Wii remote attached to user's chest	26
Figure 6 - Wii remote attached to user's head.....	29
Figure 7 – Accuracy of movements showing similarities in all but technique one	38
Figure 8 - Viewing frustums of infrared when looking at the screen	40
Figure 9 - Viewing frustums of infrared when looking straight ahead.....	41
Figure 10 - Accuracy of recognition for evasive tasks showing increases with the Motion Plus	42
Figure 11 – User’s yaw orientation with the Motion Plus and without infrared	44
Figure 12 - User’s pitch orientation with the Motion Plus and without infrared.....	44
Figure 13 - User’s yaw orientation with the Motion Plus and with infrared	45
Figure 14 - User’s pitch orientation with the Motion Plus and with infrared.....	45
Figure 15 - Maneuvering task example	47
Figure 16 - Evasive task example	48
Figure 17 - Mean time for users in completing the evasive gestures task	50
Figure 18 - Mean damage taken by users in the maneuvering task	51
Figure 19 - User Preference between techniques two and three	52

Figure 20 - User Preference between using the Motion Plus or not	52
--	----

LIST OF TABLES

Table 1 - User study test sequence.....	49
---	----

LIST OF ACRONYMS / ABBREVIATIONS

3DUI – Three Dimensional User Interface

6DOF – Six Degrees of Freedom

ANOVA – Analysis of Variance

API – Application Programming Interface

EKF – Extended Kalman Filter

IR – Infrared

ISUE – Interactive System User Experience

ODT – Omni-Directional Treadmill

USB – Universal Serial Bus

RMSE – Root Mean Squared Error

CHAPTER ONE: INTRODUCTION

The user interface for household video games has remained the same for many years, relying on joystick controllers with buttons that map the user's intentions into the virtual world. In the field of research in three dimensional user interfaces (3DUI) many forms of interaction has been defined, researched, and documented, though not adopted by the commercial user on a large scale.

Recently, however, the Nintendo Wii was released which revolutionized the console system, introducing new hardware into the common household and attracting new audiences into the gaming world (Brightman, 2009). The device can also leverage research in the 3DUI field to create new and natural user interfaces. However, the capabilities of the controller have not been scientifically examined to know what previous research in the 3DUI field applies to the hardware and what new techniques are possible.

Furthermore, this new hardware can open up a new field of exercising along with playing video games. Previous research has shown that game usage can link to obesity in children (Vandewater, Shim, & Caplovitz, 2003). While studies with the Nintendo Wii show that it may depend on how often the user plays the game to achieve an exercise workout, a 3D user interface can be used to make the game more natural and also increase activity levels (Ruberg, 2009) (Robertson, 2009).

In this thesis, we explore natural user interfaces for locomotion in video games using spatially convenient hardware. This term is defined as hardware that provides spatial data, contains a functional interface, and is convenient to install in the home (Wingrave, et al., 2010). Several techniques were developed, analyzed rigorously against a cost effective 6DOF tracker and the data presented for discussion and future work. The domain of American Football gaming was used to drive the requirements to which solutions could be developed.

Definition of Problem

First, we present a definition to the domain of American Football gaming and a statement of work. Our goal of a natural user interface means that the user should move intuitively, as though the actions in the virtual world were real, and have them respond in a way they expect. For this we found the quarterback's role to be exemplary in American Football gaming. It is this position that has to make the fastest decisions that will result in winning the game.

In traditional football based video games the quarterback is controlled by a joystick, with a complex series of buttons presented for all the many options the user can perform rather than natural movement. The user has to maneuver in a small area to avoid being tackled; either throws the ball or run it, and has certain moves that can assist in breaking away from their opponent.

In terms of locomotion, or the act of translating movement from the user's actions into the virtual world, this has always been a problem in 3DUI as it can be difficult to define a way that does not inhibit the user (Whitton, et al., 2005). For the quarterback of a football team they

have to rely on their ability to move to a position to avoid tackles and see the field if they are going to throw or run the ball successfully.

With this understanding of the domain, three major tasks were developed. First, the user must have a small area to perfectly maneuver in and have it mapped to the quarterback. For example, upon the start of a play if the user moves backwards the system should do the same, and if they move slightly to the right or left to dodge then the virtual world should follow. This falls precisely under the definition of the travel subtask, maneuvering (Bowman, Kruijff, LaViola, & Poupyrev, 2005).

The second task is that the user must be able to run a great distance in the virtual world without actually moving the same amount in their home. The domain mapping is the player may want to make a run down the field, but does not have the space in their living room to do so. As such, a new technique is needed beyond the maneuvering task, so this was called the running task.

Finally, as mentioned before, the quarterback has a set of moves that they can perform if an opponent is approaching to tackle. These maneuvers may be to jump over someone that has fallen in front of them, spin through a potential tackle, or juke their feet to throw off the direction they are heading. We needed to map these moves in some natural way for the user and called this the evasive task.

Several techniques were developed for the locomotion challenge based on hardware usage with each task integrated. While we focused on the quarterback, these tasks are important for other positions on the field, such as wide receiver, running back, etc. The quarterback has

other moves such as calling audibles and throwing or handing off the ball; however in this thesis we are only focusing on the locomotion aspect of the problem.

Hardware Definition

The hardware used was primarily based around the Nintendo Wii remote. This small device connects to a PC via Bluetooth and is accessed by software designed by community developers (Peek, 2008). This allows connection of up to four devices each giving output of buttons pressed, accelerometer data and any infrared information.

The accelerometer data is measured in “Gs” ranging from negative three “Gs” to positive three “Gs”. Furthermore, the accelerometers do not filter out gravity, so at any given moment when the Wii remote (Wiimote) is held still, the acceleration vector has a magnitude of one. With this, the orientation of the Wii remote can be determined by looking at how much of each axis contains the gravity vector when at rest. There are three axes, displayed in Figure 1 which delivers an instantaneous acceleration from the API.

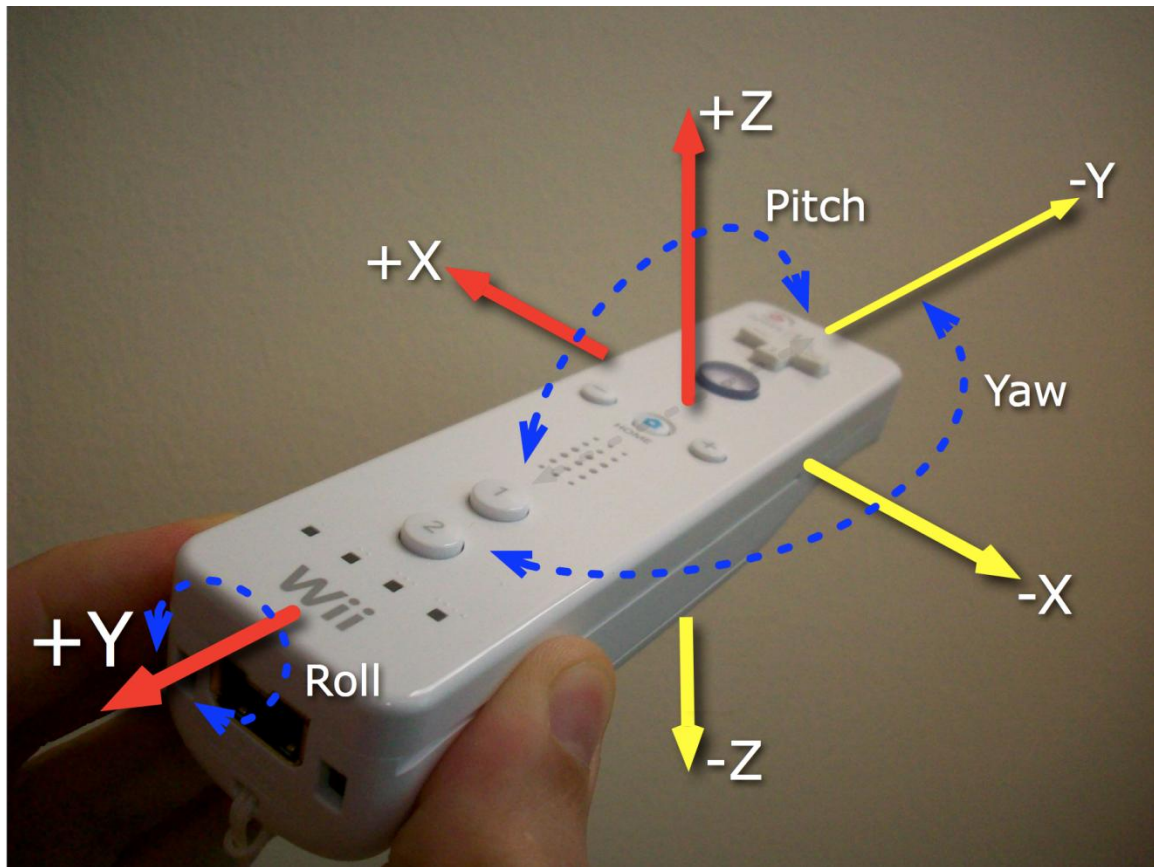


Figure 1 - Wii Remote Axis (picture courtesy of ISUE lab at UCF)

The infrared camera on the front of the Wii remote detects near infrared light and uses internal hardware to translate the blobs of light into points on a 2D plane. Natively, the sensor bars produce two beams of light that the Wii remote can see when pointed at the screen giving two points, their middle point, and their distance apart. This has the ability to be expanded to four points, but is currently not used by the at home Wii remote system. The Application Programming Interface (API) developed by the Wii remote open source community gives access to the point coordinates, both raw and normalized, to be used in order to determine 2D position (Peek, 2008).

The final piece of Nintendo hardware utilized is the recently released Wii Motion Plus. This device is a simple hardware upgrade and will be shown to improve the recognition of user orientation into video games. It is a plug-in to the Wii remote that includes multiple gyroscopes providing angular rate data back to the system. This information can be used in order to determine the orientation of the Wii remote without relying on the accelerometer's reporting the gravity vector. The integration of this device was worked on by students in the Interactive System and User Experience lab as modification to the community's Wii remote API (Peek, 2008).

Also hardware used for the cost effective 6DOF tracker was the Natural Point TrackIR device (Natural Point, 2009). We used this device over a more complex six degree of freedom tracker as it still closely resembles the cost and hardware used by the Wii remote. This consists of an infrared camera surrounded by infrared light shining outward. It then uses reflectors placed on a hat to detect points of lights. It uses three points with one offset, as seen in Figure 2, to determine position and orientation. The built in API gives position data in X, Y, and Z along with orientation of the head in yaw, pitch and roll data.

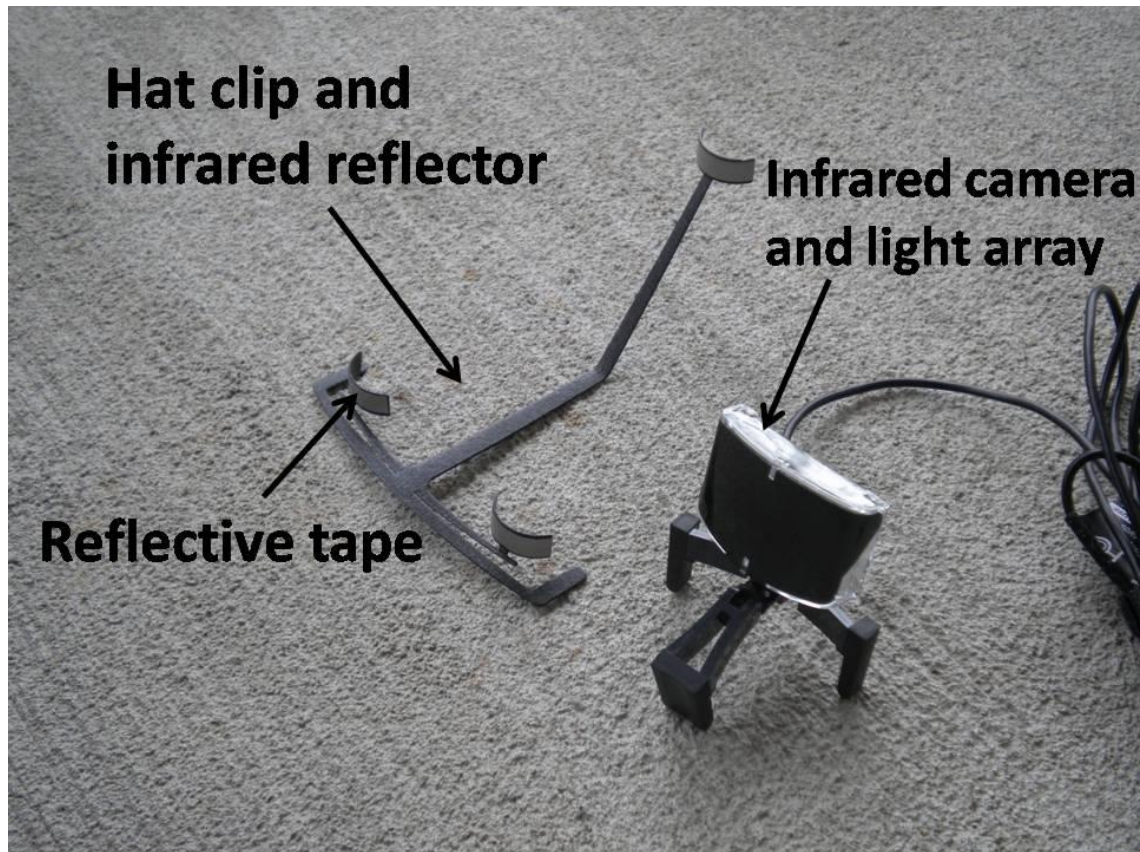


Figure 2 – Natural Point TrackIR infrared tracker

Contributions

With the problem and hardware defined, we began work on constructing a natural locomotion system dubbed RealNav. Three iterations of this software were developed, with the intent of performing research on the ideal case and hardware usage; also two of these iterations are made both with and without the Wii Motion Plus to show what the hardware can add.

The contributions made are the analysis of the Wii remote hardware in solving locomotion problems in video games. This is done by integrating position trackers using data present from a single Wii remote and analyzing motion data to decide if a user is running in

place or performing one of the three evasive gestures. Furthermore, we describe a method for navigation techniques integrating the Wii Motion Plus with an Extended Kalman Filter. Finally, results from the user study give knowledge of how the users could envision using the Wii remote and what they expect from it.

In the second chapter we cover other literature in the 3DUI field as it relates to travel and locomotion, and recent research with the Wii remote hardware. Next, the third chapter covers the techniques developed, including the comparison technique and Kalman filters. Proceeding into the fourth chapter is a definition of how the techniques were analyzed and their results. Finally, the fifth chapter gives a discussion based on the results and the sixth wraps everything up in a conclusion section.

CHAPTER TWO: LITERATURE REVIEW

In the 3DUI book navigation is broken down into two components, travel, which is the motor component and way-finding, the cognitive process of finding a path (Bowman, Kruijff, LaViola, & Poupyrev, 2005). For our research travel was the main focus with the path defined as down the field. Furthermore, travel tasks can be broken down into exploration, the user not knowing the goal, searching where the user has a specific goal in mind and finally maneuvering, small precise movements in a confined area (Bowman, Kruijff, LaViola, & Poupyrev, 2005).

The tasks of travel has also been decomposed and classified into three major components (Bowman, Koller, & Hodges, 1997). The first is direction which can be determined by gaze steering, gesture pointing, discrete selection, or 2D pointing. The next major task is velocity, which may be constant, gesture based, explicitly selected, scaled to the user's environment or adaptive. Finally, there are the input conditions, whether by constant travel, continuous input, start and stop inputs or an automatic start and stop. With these definitions in mind, a developer is able to quickly analyze their tasks at hand and make selections for how their interface will perform.

For our major task, physical locomotion techniques was to be considered, which is broken down with walking, walking in place, and vehicles or other techniques (Bowman, Kruijff, LaViola, & Poupyrev, 2005). In one publication, real walking, walking in place and joystick movements were compared (Whitton, et al., 2005). They discovered that more natural forms, such as real walking, have better performance in terms of precision and speed than walking in

place, which is still better than joystick movement. Other research studied four forms of locomotion: real walking, virtual walking with six degrees of freedom, virtual walking with only head movement tracked, and analog joystick movement (Zanbaka, Lok, Babu, Xiao, Ulinksi, & Hodges, 2004). Once again, users were more comfortable with real walking. Additionally, one paper performed an experiment on the effects of walking in place on presence in virtual reality (Slater, Usoh, & Steed, 1995). The overall result was that this method works well and does not remove the user's presence from the virtual world. Thus, when possible actual walking should be used to take advantage of these performance gains but walking in place is a viable substitute when real space is not available.

Furthermore, steps can be taken on how the travel development techniques can increase presence in the system. In recent research they discovered it beneficial to move the user forward at the speed in which they walked (Feasel, Whitton, & Wendt, 2008). Also, another publication discovered that if the camera moves as the head does when walking, such as up, down, and orthogonally swaying, the user's presence is increased (Interrante, Ries, & Anderson, 2007). These were considered when developing the RealNav techniques.

Walking as a solution to the travel task is successful as long as there is hardware available for tracking the user. In one technique they developed a system described as involving optical or ultrasonic sensors mounted in a fixed location that detect sensors attached to the user, which the 3DUI book refers to as an outside-looking in method (Foxlin, 2002) (Bowman, Kruijff, LaViola, & Poupyrev, 2005). The other possibility, inside-looking out, involves a system such as Hi-ball which involves an optical or ultrasonic device placed on the user that

looks to fixed mounted sensors (Welch, Vicci, Brumback, Keller, & Colucci, 2001). This solution's effect on presence has been further examined, where users walked over a virtual pit with the Hi-Ball tracker versus other travel techniques (Usoh, et al., 1999). User's felt much stronger fear when the Hi-ball tracker was used to produce a walking technique over others.

These methods work well for smaller areas of movements, especially indoors with great precision, but would fail for larger or longer movements. In an augmented reality paper they needed large scale tracking in outdoor environments for mobile augmented reality (Höllerer, Feiner, Terauchi, Rashid, & Hallaway, 1999). In this scenario, GPS was used to track the user, which can be blended with inertial data to provide more accurate and frequent updates.

For the development of RealNav, actual walking in the environment was the ideal solution for the maneuvering task as it needs precision and is in a closed in space. Other methods were needed, however, when this technique simply was not possible.

Another method of locomotion in 3D user interfaces is of walking in place. As stated in the 3DUI book "this technique is a good compromise because users still physically exert themselves" and the environment is no longer a limitation (Bowman, Kruijff, LaViola, & Poupyrev, 2005). However, there are some limitations in that the motion of real walking is lost to the user's senses.

The implementations of this technique can have many variations. In one publication they placed position trackers on the feet of the user and used neural networks to determine if a user was walking compared to other motions of the feet (Slater, Usoh, & Steed, 1995). In these experiments the neural networks were able to detect the walking motion correctly 91% of the

time on average. In other methods, such as GAITER more sophisticated algorithms worked with multiple sensors to determine walking (Templeman, Denbrook, & Sibert, 1999). Some methods can be entirely different, such as the development of special sandals that allowed the user to shuffle in place to move forward rather than the up and down motion of other walking methods (Iwata & Fujii, 1996).

As stated previously, walking in place has been shown to still maintain presence in the virtual environment more than an entirely virtual locomotion technique (Usuh, et al., 1999). However, while not as effective as normal walking, the systems still perform well when the user must travel further than their physical space allows, as is discussed in our implementation in RealNav.

A final form of locomotion to discuss is devices that simulate walking. This is desired when realistic walking is needed, but the environments traveled are greater than the physical environment provided. These systems are that “provide a real walking motion and feel while not actually translating the user’s body” (Bowman, Kruijff, LaViola, & Poupyrev, 2005). In a very simple sense a treadmill works except for when a user needs to turn which has been accomplished in the past with merely a joystick (Brooks, 1986).

Other more advanced methods allowed the user to slowly turn their head to change direction which would then cause the treadmill to rotate as well (Noma & Miyasato, 1998). However, the user could not turn quickly or sidestep with such a design. Another design is the Omni-Directional Treadmill (ODT) and the Torus treadmill (Darken, Cockayne, & Carmein, 1998) (Iwata, 1999). These focus on the idea of two sets of rollers moving orthogonally to each

other giving the treadmill to move in any arbitrary horizontal direction. These work well, but still cannot handle sudden turns and other maneuvers a person may make.

A novel approach that didn't use a treadmill was the GaitMaster (Iwata, 2001). This detected the user's motion with force sensors that moved hard platforms around so that the user felt a "ground" surface at the correct location of each step. However, this technique is very complex and has serious safety issues to resolve (Bowman, Kruijff, LaViola, & Poupyrev, 2005).

For RealNav, these devices were not implemented as spatially convenient low cost solutions were part of the design. Any customized treadmill mechanism would push the design costs too high for our intent.

This use of the Wiimote in 3D user interface research is one of growing popularity, started as users began breaking into the mechanics of the device (Lee, 2008). In recent publications, two Wiimotes were used to control the animation of a virtual character through arm, hand and leg movements (Shiratori & Hodgins, 2008). It also described how correlation of the data between the two Wiimotes could recognize a small but natural gesture set. While this gave guidance to Wiimote placement and usage, especially in terms of physical walking, it did not fit our domain of sports games.

Also making use of the Wiimotes are systems for multi-wall virtual reality theatres, which handled how the device could be used with the system including pointing techniques (Schou & Gardner, 2007). Another system was developed with use of the Wii remote for navigation of complex three dimensional medical data such as MRIs (Gallo, De Pietro, & Marra, 2008). On the gaming note the device has also been explored for musical instruments and for

dancing based games (Kiefer, Collins, & Geraldine, 2008) (Charbonneau, Miller, Wingrave, & LaViola, 2009).

With all of this literature previously developed we see potential for the use of Wii remote and 3D user interface research to build new systems and video game concepts in the home. Though much has been done with locomotion, particularly real walking and walking in place, nothing has merged this capability and research with the Wii remote. As mentioned before, this hardware is spatially convenient, bringing the research home for the average video game user.

CHAPTER THREE: TECHNIQUES

Three techniques were developed along with a fourth as a comparison to test against in chapter four. Generally, the running task and evasive task had the same solutions in all techniques, with minor modifications introduced depending on what data was present. We will present these first in an overview, then go through each technique, the idea behind it, and the development that resulted in a final software piece to be tested.

Running Task and Evasive Task

As stated previously, walking in place seems like an adequate solution for the Running Task, so this was implemented in every technique. We decided to go with running rather than walking, as the user would not be walking down the football field in an actual game (unless very confident).

For every technique, the accelerometers were used to detect large upward movement on the vertical axis, with vertical depending on the orientation of the Wii Remote. We relied upon the system to filter out the gravity vector when needed. Generally though, when upward acceleration passed a threshold and was repeated, the user was considered to be running. After this state was entered all upward acceleration would then result in the virtual world moving forward slightly.

The result was that the virtual world moved as fast as the user was running in place. Faster running motions would move quickly forward, motivating the user to run in place as best as they could. Also, since the focus was locomotion and not navigation, the user moved strictly forward when running in place.

The evasive tasks relied on the user performing the natural gesture and having the software recognize it. For the jump, it was the gesture of the user jumping into the air, for spinning they would turn a full revolution and for the juke they would step to one side and then launch onto the other foot.

Kalman Filters

A Kalman filter was developed for the third technique in order to determine position, velocity, and acceleration from both the infrared and accelerometer data that was available. Furthermore, an extended Kalman filter was designed to implement navigation techniques using the Wii Motion Plus in order to account for noise from the gyroscopes. Both of these were based on filters designed in previous dissertation research (Azuma, 1995).

We used this approach because the filters are capable of both reducing noise in the model, problems that the accelerometers and gyroscopes face, and determining hidden states given measurements. This is done by performing a time update step, which predicts the next state based on the process model defined, and a measurement update step which takes in the actual measurement. A Kalman gain is calculated, which determines the best way to blend the time update prediction and the measurement to get the most accurate result. The details of these filters are defined in the following sections.

Position/Velocity/Acceleration Kalman Filter

For this filter the time update step is

$$\hat{x}_t = X_{t-1} * A_{\Delta t} \quad (1)$$

$$\hat{P}_t = A_{\Delta t} * P_{t-1} * A_{\Delta t}^T + E$$

where

$$X_t = \begin{bmatrix} X \\ \dot{X} \\ \ddot{X} \end{bmatrix} \quad (2)$$

$$A_{\Delta t} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

Also, the state vector X_t represents the state position, velocity and acceleration and $\hat{P}_t$ is the predicted covariance matrix, coming from the last state P_{t-1} . E is a constant error matrix for the prediction step itself. The matrices listed above are all time dependent, moving from a previous time state to a new one and with the fundamental matrix being built with respect to the change in time.

This shows that the state vector consists of the position, velocity, and acceleration. By multiplying this with the fundamental matrix, the next state is predicted based on changes in time.

The next step is to perform the measurement update which is

$$K_t = \hat{P}_t * H^T * (H * \hat{P}_t * H^T + R)^{-1} \quad (3)$$

$$X_t = \hat{X}_t + K_t * (Z_t - H * \hat{X}_t)$$

$$P_t = (I - K_t * H) * \hat{P}_t$$

where

$$Z_t = \begin{bmatrix} X \\ \ddot{X} \end{bmatrix} \quad (4)$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Also, K_t is define as the Optimal Kalman Gain and R is an error covariance matrix representing noise from the devices. The covariance matrices and initial values were based on the data observed in other research and adjusted slighted based on manual optimization (Azuma, 1995).

There was a predictor step added in from Azuma's dissertation based on predicting the state based on previous data when measurements are unavailable. This is essentially the time update step, except no states are actually changed within the filter. It was designed as

$$X_p = X_t + \dot{X}_t * (p - t) + \frac{1}{2} * \ddot{X}_t * (p - t)^2 \quad (5)$$

where X_p is the predicted state, X_t is the position from the last measurement update, $\dot{X}_t$ is the velocity, $\ddot{X}_t$ is the acceleration, p is the point of time in which a prediction is made and t is last point in time in which the Kalman filter's time update and measurement update steps were performed.

Extended Kalman Filter

For determining orientation from the Wii Motion Plus, non-linear equations were used and as such required an extended Kalman filter. The equations between the two filters did not change, and should be references for time update (1) and measurement update (3). However the state vectors, fundamental matrix, and many other variables were all re-defined for this model.

Also, a quaternion was used to represent the orientation of the Wii remote with the first value being the scalable real number and the following three representing the imaginary vector. Furthermore the angular velocity and angular acceleration were tracked in the filter. This is

$$X_t = \begin{bmatrix} Qw \\ Qx \\ Qy \\ Qz \\ \omega 0 \\ \omega 1 \\ \omega 2 \\ \dot{\omega} 0 \\ \dot{\omega} 1 \\ \dot{\omega} 2 \end{bmatrix} \quad (6)$$

where Qw , Qx , Qy , and Qz are the quaternion states, $\omega 0$, $\omega 1$, and $\omega 2$ are the angular velocity and $\dot{\omega} 0$, $\dot{\omega} 1$, and $\dot{\omega} 2$ are the angular acceleration.

To determine the process model we began with a non-linear equation to describe the derivative of the quaternion. This was done by performing a quaternion multiplication between Q and ω where the angular velocity's components are set to a quaternion with the Qw term being set to zero. This was then integrated with respect to time resulting in

$$\int_{t-1}^t \frac{1}{2} * (Q * \omega) dt \quad (7)$$

The derivative and integral that would predict angular velocity used a linear equation based on the angular acceleration that was calculated. The acceleration itself was presumed constant.

In order to make the equation linear so that it could work in the Kalman filter, the process model became a Jacobian matrix of the above equation. Every row was taken as the partial

derivative with respect to the other row. This was combined with the integration with respect to time and making the diagonal all ones so that the previous state was added to the current one.

The result is

$$A = \begin{bmatrix} 1 & \frac{-\omega 0 * \Delta t}{2} & \frac{-\omega 1 * \Delta t}{2} & \frac{-\omega 2 * \Delta t}{2} & \frac{-Qx * \Delta t}{2} & \frac{-Qy * \Delta t}{2} & \frac{-Qz * \Delta t}{2} & 0 & 0 & 0 \\ \frac{\omega 0 * \Delta t}{2} & 1 & \frac{\omega 2 * \Delta t}{2} & \frac{-\omega 1 * \Delta t}{2} & \frac{Qw * \Delta t}{2} & \frac{-Qz * \Delta t}{2} & \frac{Qy * \Delta t}{2} & 0 & 0 & 0 \\ \frac{\omega 1 * \Delta t}{2} & \frac{-\omega 2 * \Delta t}{2} & 1 & \frac{\omega 0 * \Delta t}{2} & \frac{Qz * \Delta t}{2} & \frac{Qw * \Delta t}{2} & \frac{-Qx * \Delta t}{2} & 0 & 0 & 0 \\ \frac{\omega 2 * \Delta t}{2} & \frac{\omega 1 * \Delta t}{2} & \frac{-\omega 0 * \Delta t}{2} & 1 & \frac{-Qy * \Delta t}{2} & \frac{Qx * \Delta t}{2} & \frac{Qw * \Delta t}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

The measurement taken from the system was assumed to know the orientation to some degree and contain the angular velocities. This resulted in an observation matrix H that only took out the angular acceleration data from the state vector. Both of these are

$$Z_t = \begin{bmatrix} Qw \\ Qx \\ Qy \\ Qz \\ \omega 0 \\ \omega 1 \\ \omega 2 \end{bmatrix} \quad (9)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Furthermore, a method was taken from previous research that involved a prediction step (Azuma, 1995). This would be used when a good measurement was not available, so the process

model had to be used given the previous data. Essentially, this was similar to the time update step, but without the actual changes to the state of the filter.

The equation begins with the derivative of quaternion Q being represented as the multiplication between the previous state Q and matrix $M(t)$ defined as

$$\dot{Q} = \frac{1}{2}(Q * \omega) = M(t) * Q \quad (10)$$

where

$$M(t) = \begin{bmatrix} 0 & \frac{-\omega_0(t)}{2} & \frac{-\omega_1(t)}{2} & \frac{-\omega_2(t)}{2} \\ \frac{\omega_0(t)}{2} & 0 & \frac{\omega_2(t)}{2} & \frac{-\omega_1(t)}{2} \\ \frac{\omega_1(t)}{2} & \frac{-\omega_2(t)}{2} & 0 & \frac{\omega_0(t)}{2} \\ \frac{\omega_2(t)}{2} & \frac{\omega_1(t)}{2} & \frac{-\omega_0(t)}{2} & 0 \end{bmatrix} \quad (11)$$

The prediction of the next state would then be to integrate the derived state $\dot{Q}$ given

$$Q_p = e^{\int M(t)dt} Q_{tc} \quad (12)$$

where Q_p is the predicted state and Q_{tc} is the state since the last measurement update.

The research then points out that the integral of an angular velocity component composing $M(t)$ would result in

$$\begin{aligned} \int_{t_c}^p \omega_0(t)dt &= \int_{t_c}^p [\omega_0(t_c) + (p - t_c) * \dot{\omega}_0(t_c)]dt \\ &= (p - t_c)\omega_0(t_c) + \frac{1}{2}(p - t_c)^2 \dot{\omega}_0(t_c) \end{aligned} \quad (13)$$

where p is the present point and time and t_c is the last point in time that a measurement update occurred.

To simplify $M(t)$ the integral components of angular velocity are simplified into

$$a = \frac{1}{2}(p - t_c)\omega_0(t_c) + \frac{1}{4}(p - t_c)^2\dot{\omega}_0(t_c) \quad (14)$$

$$b = \frac{1}{2}(p - t_c)\omega_1(t_c) + \frac{1}{4}(p - t_c)^2\dot{\omega}_1(t_c)$$

$$c = \frac{1}{2}(p - t_c)\omega_2(t_c) + \frac{1}{4}(p - t_c)^2\dot{\omega}_2(t_c)$$

$$d = \sqrt{a^2 + b^2 + c^2}$$

We can then rewrite (12) as

$$Q_p = e^B * Q_{t_c} \quad (15)$$

where

$$B = \begin{bmatrix} 0 & -a & -b & -c \\ a & 0 & c & -b \\ b & -c & 0 & a \\ c & b & -a & 0 \end{bmatrix} \quad (16)$$

We then use a Taylor expansion on (15), resulting in

$$e^B = I + M + \frac{M^2}{2!} + \frac{M^3}{3!} + \frac{M^4}{4!} + \dots \quad (17)$$

where I is the identity matrix.

By working out the values of M^2 and M^3 the following pattern emerges between the even and odd exponents.

$$M^2 = -d^2 * I \quad (18)$$

$$M^3 = -d^2 * M$$

$$M^4 = -d^4 * I$$

$$M^5 = -d^4 * M$$

etc ...

We then group the even and odd exponential pairs, factor out I and $\frac{M}{d}$ and find (17) can be simplified into

$$\begin{aligned} e^B &= I \left(1 - \frac{d^2}{2!} + \frac{d^4}{4!} - \frac{d^6}{6!} + \dots \right) + \frac{M}{d} \left(d - \frac{d^3}{3!} + \frac{d^5}{5!} - \frac{d^7}{7!} + \dots \right) \\ &= I * \cos(d) + \frac{M}{d} * \sin(d) \end{aligned} \quad (19)$$

Combining (19) with the original derivation (15) we get a final simplified equation to predict the next state when a measurement is unavailable. This is

$$Q_P = \left[I \cos(d) + \frac{M(t_c)}{d} \sin(d) \right] Q_{tc} \quad (20)$$

which can calculate a predicted quaternion Q at time P given a valid time update and measurement update steps performed on the extended Kalman Filter at time T .

Technique Zero: Comparison with Optical Tracker

The first technique created was meant to form a standard for the other techniques to conform to in performance. The hardware used in this method was the Natural Point TrackIR head tracking system and two Wii remotes (Natural Point, 2009). The head tracker was placed upon the user's head as seen in the figure below, and Wii remotes were attached to the legs.



Figure 3 -TrackIR attached to user



Figure 4 - Wii remote attached to user's leg

For the maneuver task this was done by mapping the positional data from the TrackIR API to realistic virtual space. In this way, the movement the user performs within the camera's field of view is mapped easily into the system. With the orientation data, this was found to be very noisy and disruptive to the user to see. To solve this problem an alpha-beta filter was utilized shown in

$$\hat{X}_n = (\beta * \hat{X}_{n-1}) + (\alpha * \hat{X}_n) \quad (21)$$

where

$$\alpha + \beta = 1 \quad (22)$$

Next was the running task, which involved taking two Wii remotes and strapping them to the user's legs. From this perspective, the upward acceleration of jogging could clearly be seen, and after data analysis a threshold was developed to determine when the user was running. This was then translated into forward movement down the field.

Finally the evasive tasks were implemented. For jumping it allowed for the Natural Point TrackIR's upward position to be turned on, which also allowed a bouncing head as the user jogged as is shown to be more realistic (Interrante, Ries, & Anderson, 2007). When the vertical position was seen to sharply increase, a jump flag was set in the system. Spinning was handled similarly, looking for the yaw orientation of the head to go one direction, stop (as the user is out of field of view) and then begin again from the other direction. As with jumping a flag was then set into the system. For the last gesture, the juke, sideward acceleration was monitored on the Wii Remotes looking for one to move sharply to the side followed by the other. This only set a flag and did not need to be displayed.

There was also the issue of the gravity vector and how it was filtered out. In this method the orientation of the Wii remote is relatively static as the legs are normally going straight up when left standing still. Thus, a vector was programmed in for the upward orientation the remotes were placed.

In conclusion this technique was developed effectively with all of the data present. It mapped the user well and was a successful demonstration of what was expected when the user would take part in RealNav. However, it had no concern for cost or placement of the hardware.

Technique One: Accelerometer Only Method

First we decided to include the least amount of hardware and intrusion upon the user which was to use single Wii remote and only the accelerometers. This was so the user did not have to be looking at or near their sensor bar and still have range of movement for as long as the Bluetooth had range.

The single Wii remote was placed at the user's chest for the first iteration as seen in the figure below. This was different from the comparison technique's location of the legs as the chest accelerations were moving with the center of mass and could be better trusted to the user's intentions.



Figure 5 - Wii remote attached to user's chest

The first iteration was to analyze the signal from the accelerations in order to determine the direction the user was moving. At first the thought was for forward movement the Z axis

would experience positive accelerations and negative accelerations would relate to backward movement. However, it was immediately noticed that positive and negative accelerations occurred together as the user moved and then stops, resulting in deceleration.

The signal analysis path was continued by noticing the pattern that forward movement always began with positive acceleration and backward movement always began with negative accelerations. With this knowledge a version of RealNav was developed that looked for accelerations beyond a certain threshold and then looked if it was positive or negative to determine forward or backward movement. Next, the system moved in that direction until the accelerations settled back to near stable as that meant the user had stopped. This was then coupled with sideways movement allowing eight cardinal directions to move in.

While this technique did work, it suffered greatly from latency as it had to determine the way the user was moving after they had already begun accelerating. Also, it could only determine the eight cardinal directions, exact angles were not recognized, nor was the precise time the user stopped moving. Finally, sudden direction changes were not detected as the user had to stop and wait for the system to consider it stable.

Although this iteration had an interesting design and worked, it was not considered good enough when compared to the cost effective 6DOF tracker. With this knowledge a more simple solution was considered of double integration of the acceleration in order to determine position. The result of the integral is

$$\hat{X}_n = \hat{X}_{n-1} + \hat{V}_n * t + \frac{1}{2} \hat{A}_n * t^2 \quad (23)$$

$$\hat{V}_n = \hat{V}_{n-1} + \hat{A}_n * t$$

where $\hat{X}_n$ is the position, $\hat{V}_n$ is the velocity, $\hat{A}_n$ is the acceleration, and t is the time elapsed since the last update.

The system worked well under two major assumptions, that the data contained little noise and the gravity vector was completely removed. For noise removal, an alpha beta filter (21) was put in which proved to introduce minor, but acceptable, latency with an increase in accuracy. With the gravity vector its calibration was updated every time the accelerometer data stayed near 1.0. This was taken as being steady, so no matter how the Wii remote was rotated the gravity vector was known. The full system worked well when performing the maneuvering task, with some minor accuracy loss when the user leaned as they moved, due to the gravity vector no longer being known until movement stopped.

Running in place was determined when upward acceleration beyond a threshold was seen at a frequent rate. This placed the system into a state in which all accelerometers were ignored the user moved forward every time there was an upward motion. It had latency in kicking off as the system waited for steady upward movement, but overall would pick up and move forward.

For the gestures matters became very complicated, as the accelerometer data was already being used to determine so many other states. They were not implemented as pure heuristics and extreme motions would be the only way without making them ambiguous. While it is possible to put them into the system, it is very limited in the ability of growth and was thus excluded.

Technique Two: Head Tracking Method

This technique was designed around making use of the infrared sensors. We determined that the user would be looking at the sensor bar if placed near their television, and the range would have to be tested to determine how effective it is. For this, a hat was used with Velcro to mount the Wii remote to the head facing forward as seen in the figure below.



Figure 6 - Wii remote attached to user's head

With the sensor bar turned on, the Wii remote API would then send back two points in the X, Y plane which were normalized. Also, a midpoint was calculated between the two points and with this, horizontal and vertical movements were easily mapped by just taking the differences in the midpoint coordinates.

For depth mapping a solution was developed of just taking the distance between the two infrared points because as the user gets closer to the sensor bars, the points would grow further apart, and as they stepped away they would grow closer together. The movement was not linear, however, and mapping out the depth changes into an excel chart revealed a parabolic movement.

The solution was to take the square root of the distance changes, which made the depth map linearly to the user's movements. This was shown to be an accurate solution for the maneuvering task when the infrared could be seen.

For the running task, accelerometer data upon the same axis as the gravity vector was recorded for the upward movement of walking in place. This was assumed to be similar to the accelerations seen when jumping, so both actions were observed to find distinguishing features in the data. Also collected was data on where the virtual camera of the system was currently positioned, as this should differ greatly in vertical displacement between running in place and jumping.

What was seen was that if the standard deviation of a set of data on the accelerometer axis, in this case Z would rise sharply and the standard deviation on the vertical position stayed near zero, the user was running in place. Two threshold values were set in a gesture recognition class which would flag if running in place was occurring. Should the system receive this feedback during the update loop, it would move the user forward slightly.

With the evasive tasks, first jumping was distinguished by seeing a large increase in the standard deviation of the vertical position. For juking the mean accelerometer data on the X axis, or sideways movement, was watched if it went over a threshold in either direction. Finally, spinning relied on the fact that the sudden spin would result in the horizontal position of the camera to shift around at great pace. This meant observing the standard deviation of the horizontal position of the virtual camera.

Integration with Wii Motion Plus

This method was selected as being capable of using the Wii Motion Plus as the infrared information was essential for correction of drift on the yaw axis.

A tracker class was built to produce the yaw, pitch, and roll of the user's head based on the data from the Motion Plus and the corrections provided by heuristics and the Extended Kalman Filter.

First, the angular rates were read from the Motion Plus with a simple calibration added in. Next, if the Wii remote's accelerometer data was relatively stationary pitch and roll were calculated from the detected gravity vector. This equation came from the Wii Linux website (Motion Analysis, 2009) in pseudo code and is

$$\begin{aligned} roll &= \arctan2(a_x, \sqrt{a_y^2 + a_z^2}) \\ pitch &= \arctan2(a_y, \sqrt{a_x^2 + a_z^2}) \end{aligned} \tag{24}$$

where a_x, a_y , and a_z are the axis data from the accelerometers. This created an equation reliable against singularity and other rotational problems, but made pitch only move from zero and ninety degrees. For this reason, function

$$f(pitch) = \begin{cases} pitch, & a_z \geq 0 \\ pitch + \left(\frac{\pi}{2} - pitch\right) * 2, & a_z < 0 \end{cases} \tag{25}$$

was applied.

This was a heuristic approach that if the Z axis of the Wii remote accelerometers were reporting the device being upside down, the angle was corrected to produce a correct value.

The values from (24) and (25) were then sent as measurement updated with the angular rates to the Extended Kalman Filter discussed above. The result was a corrected pitch and roll angles taken from the quaternion output. Yaw was excluded, as the measurement update was not as reliable to occur as often as the tilt determined from the gravity vector. Should this data not be available, the predictor step for the Extended Kalman filter was used.

With yaw, a heuristic approach was taken based on if infrared data was available at all. If this was the case, the previously reported yaw value was overwritten based on the X value of the infrared sensor, as this meant the remote was near zero degrees, pointing forward at the infrared. If the data was not present, the angular rates went through Euler integration with respect to time to determine the yaw value.

These three angles were then placed through an alpha beta filter similar to the one shown in (21) for smoothing purposes, especially of the yaw value as it was corrected with infrared. For the final integration into the system, roll was discarded as it was visually distracting from the user.

Also, the motion plus data had a threshold value created to signal to the system if the user was rotating their head or not. This was done to resolve any ambiguities in why the infrared data may be moving, whether the user is physically moving horizontally or vertically, or if they are rotating their head. If rotating, camera movement was ignored; otherwise, it took the normal effect described above.

Finally, the Motion Plus data proved valuable in recognizing the spin gesture in the evasion task. If the yaw angular rate had a sudden and consistent increase, the user was considered to be spinning and was flagged as such.

Technique Three: Hybrid Method

It was immediately decided that techniques one and two both had their pros and cons. Technique one let the user travel anywhere in the maneuvering task, but lacked in accuracy to the user's movements. For technique two, it could map the maneuvering task to the user with little hardware, but if the infrared was lost the system would come to a grinding halt.

The hybrid technique was then designed to merge the best of both worlds. A Kalman Filter was implemented as described above to blend the position data from the infrared and accelerometer data to result in the best accuracy and range combination. If the user stepped outside of the infrared, the prediction step from the Kalman Filter described in (5), was called, passing in the change in time since the last good measurement, and any updates to the accelerometer data. These updates were not sent into the actual state of the filter, however, and were only used until the user stepped back into range of the infrared.

Because of its similarities to technique two, the same gesture recognition class was used to resolve both running in place and the evasive gestures. Also, a version implementing the Wii Motion Plus was created using the exact same methods above.

With all of the techniques developed for a natural locomotion based user interface, an analysis was needed to determine which the best for the domain was. Furthermore the data

gathered will be presented so that future research with the Wii remote hardware in other domains can be determined and the ideal method can easily be selected.

Software and Computer Used

A single PC was used for device interaction and software creation and execution. This was an off the shelf commercial machine with a Bluetooth USB device.

The software used for these tasks involved basic drivers on a Windows Vista machine, including Bluetooth interface drivers. Furthermore the APIs for the TrackIR (Natural Point, 2009) were provided along with the Brian Peek Wii remote software API (Peek, 2008) for device interface.

All code was developed in Microsoft Visual Studio 2005 in the C# language. They were built upon the Bespoke Framework (Varcholik, 2009) which itself was built upon the XNA framework developed by the Microsoft Corporation. All models rendered were free to download from the web and created by various authors and used as needed.

CHAPTER FOUR: ANALYSIS AND RESULTS

With three techniques, the question arose of which one was the best, leading to a need for quantitative data that could be presented. This information may also be used for other applications depending on which domain they are being used for. To begin, metrics were defined on what data to gather and how it would be obtained. Followed by that is the presentation of these results.

Metrics Definitions

First we describe the methods in which metrics were gathered, followed by the data produced when applied to each technique. They are provided first in the following list proceeded by in depth definitions.

- Accuracy of technique given movement in four cardinal directions and complex movement
- Range of infrared hardware
- Correct gesture prediction both with and without the Motion Plus
- Orientation analysis of the Motion Plus both with and without infrared
- User Study Results

Metrics were first gathered for accuracy in the maneuvering task for each method. This was done by placing an X on the floor for the user to stand on. They would then move in one of the four cardinal directions and then move back to the X, having this mapped in the virtual world. They would then press a button that would cause the system record in the virtual world

the magnitude of the difference in starting and ending position. This was repeated for each direction 15 times and averaged for final results.

With that data, the recovery accuracy was also recorded. This was how well the RealNav system could have a more complex movement and still recover back to the original spot. It was originally designed for the infrared systems when the user moved outside of the infrared field of view and then back to the X. Similarly to the metric mentioned above, the user would move in a horizontal cardinal direction at least three steps or until the infrared was lost. They would then move back and record the difference in the starting and stopping positions in the virtual world.

After this, we examined the range of infrared for the Wii remote and compared it to the TrackIR's hardware. This was to give an idea of the box to which the user could move around in and still have the system track the position accurately. It was decided to map out several points in order to fully understand the frustum's bounds.

The first set of points were the closest the user could be while centered in front of the screen, followed by the farthest distance they could move and still have visible IR. Next, were the points moving horizontally, both near and far, and both looking at the screen still and looking straight ahead. The straight ahead view was more theoretical, while the looking at the screen more likely for someone engaged in the virtual world.

Also metrics were taken regarding the Wii Motion Plus. First, accuracy of the gesture recognition (joking, spinning, and jumping) with and without the Motion Plus was gathered. This was to show whether the extra data could resolve ambiguity issues. The next metric was on the ability for the Wii Motion Plus to give correct orientation in pitch and yaw both with and

without the infrared device. This measurement involved an Intersense head tracking system to give true measurements which could then be compared with the Motion Plus reported orientation (Intersense Inc., 2009).

Finally, subjective data was gathered on a preferred method. This involved a simple set of tasks involving dodging thrown objects using the natural maneuvering interface, along with performing the evasive gestures and running in place. A small set of users tested this and provided data on what they preferred and why.

Accuracy Data

All of the accuracy data was based upon the ability for the system to return to zero in terms of yards. The correlation between virtual units and yards was calculated on how much movement was needed to proceed down the hundred yard virtual field. For the first test, the four cardinal directions were taken with one step each and then returning to the starting position in the real world. In the virtual world, the difference between start and ending was recorded. This was done fifteen times and then averaged for each direction and repeated for more complex movements that involved several steps. The distance was measured in yards as this is the common unit in American Football and was determined by taking the difference in virtual position from one end of the field to the other.

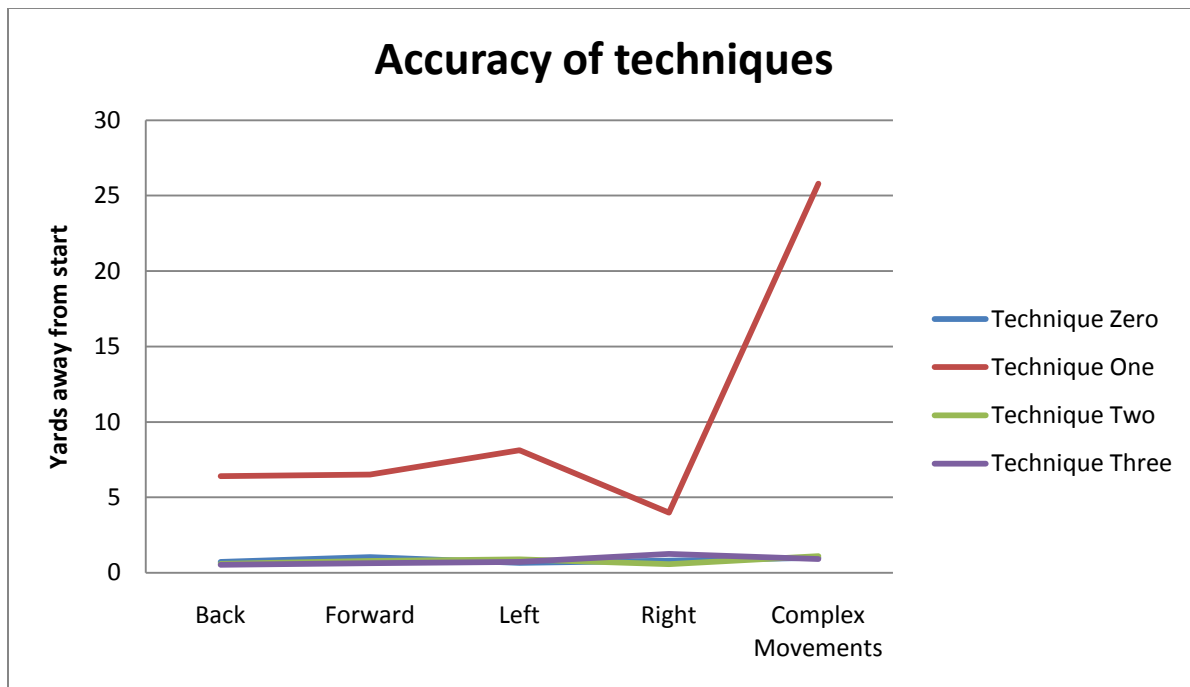


Figure 7 – Accuracy of movements showing similarities in all but technique one

As you can see in Figure 7 this data is relatively consistent for every technique except the first. While no technique touched exactly zero, the minor difference in yards is acceptable as the user rarely stepped back into the starting position perfectly. The first technique suffered heavily from drift, growing worse as more complex movements were introduced.

Range Data

Next, data was gathered on the range of the infrared systems. For the first technique that used only accelerometer data, Bluetooth range was the only restriction of concern which is far more than the user's ability to view the average screen. With this in mind, the infrared was measured for the two IR devices, the Wii Remote and Natural Point TrackIR.

This measurement was taken by loading up test software showing when the computer or device had track of infrared and when it didn't. Then, measurements were taken based on

horizontal movements both looking at the screen while moving and looking straight ahead. The idea being is that the user may be strafing to the side and looking straight, or may keep their head fixed on the screen which should have a longer range of infrared detection.

Also, the near-range for both devices was capable of incredibly close and unrealistic distances, such as the Wii remote being one foot and four inches from the screen. This was impractical though, as most users would not be able to see their screens, so distances were measured at two feet and nine inches away. Horizontal movement was performed both near and far creating a viewing frustum, a box in which the user could move and still detect infrared position. These are plotted in the figures below.

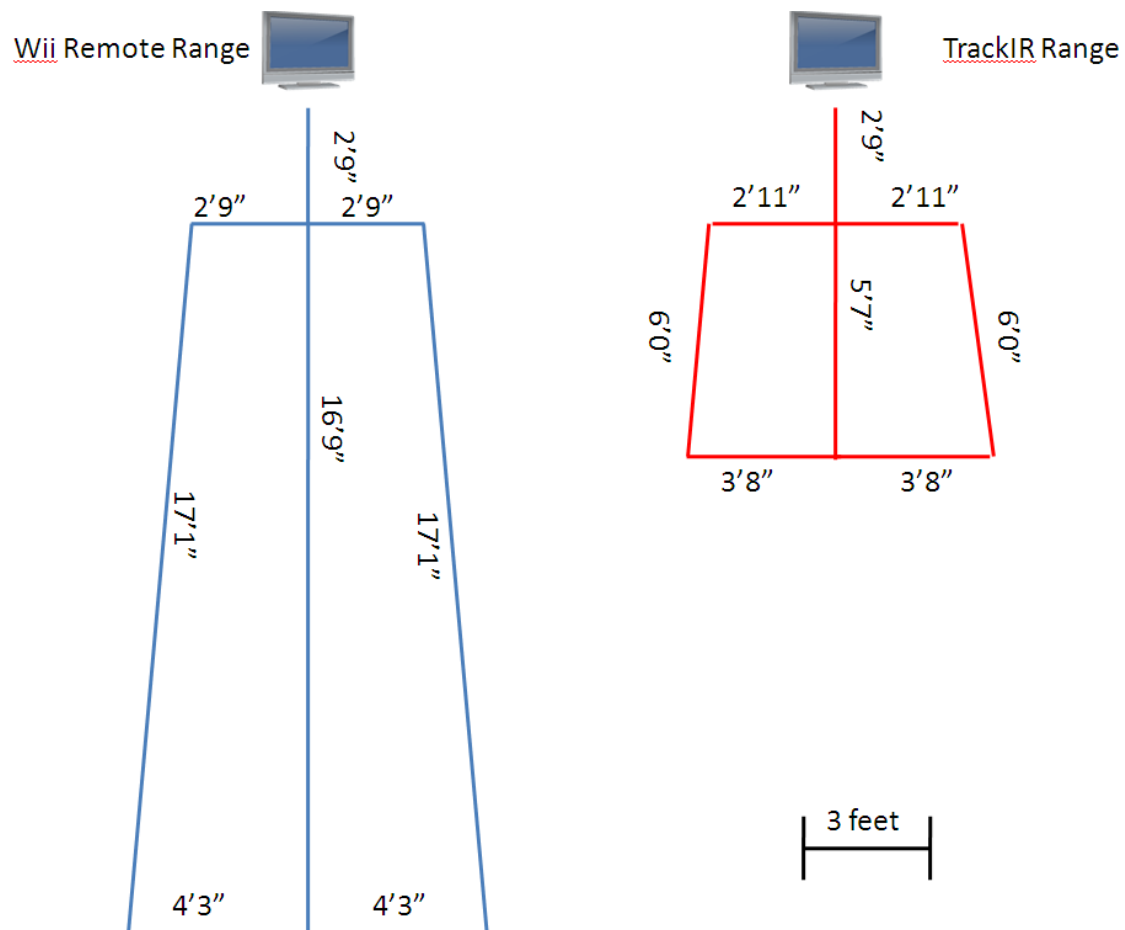


Figure 8 - Viewing frustums of infrared when looking at the screen

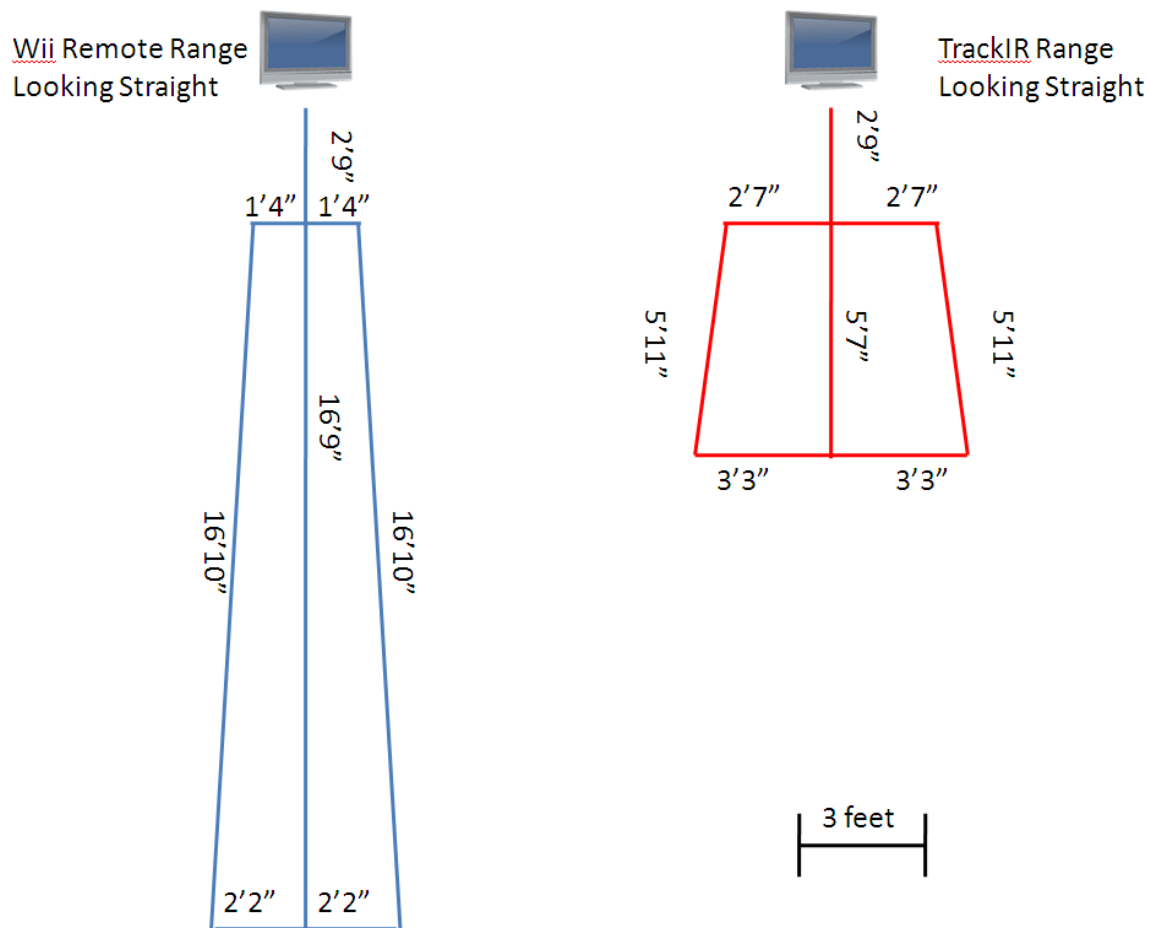


Figure 9 - Viewing frustums of infrared when looking straight ahead

There were a few discoveries from these graphs, first that the Wii remote has much longer range of infrared than the TrackIR, which is expected as the TrackIR device was built for desktop application. To the sides, however, the Wii remote is better when the head moves to continue looking at the screen, still picking up infrared at extreme angles. However, if the user continues to look straight, the data drops out much sooner than the TrackIR does.

Evasive Recognition Accuracy

For each evasive gesture the user was prompted to perform the gesture and given ten seconds to do so. If a different evasive maneuver was detected, the time immediately expired and the user moved on to the next session. This was repeated twenty-five times for each gesture and then once with the Wii Motion Plus heuristics and once without. The results are shown in the following figure.

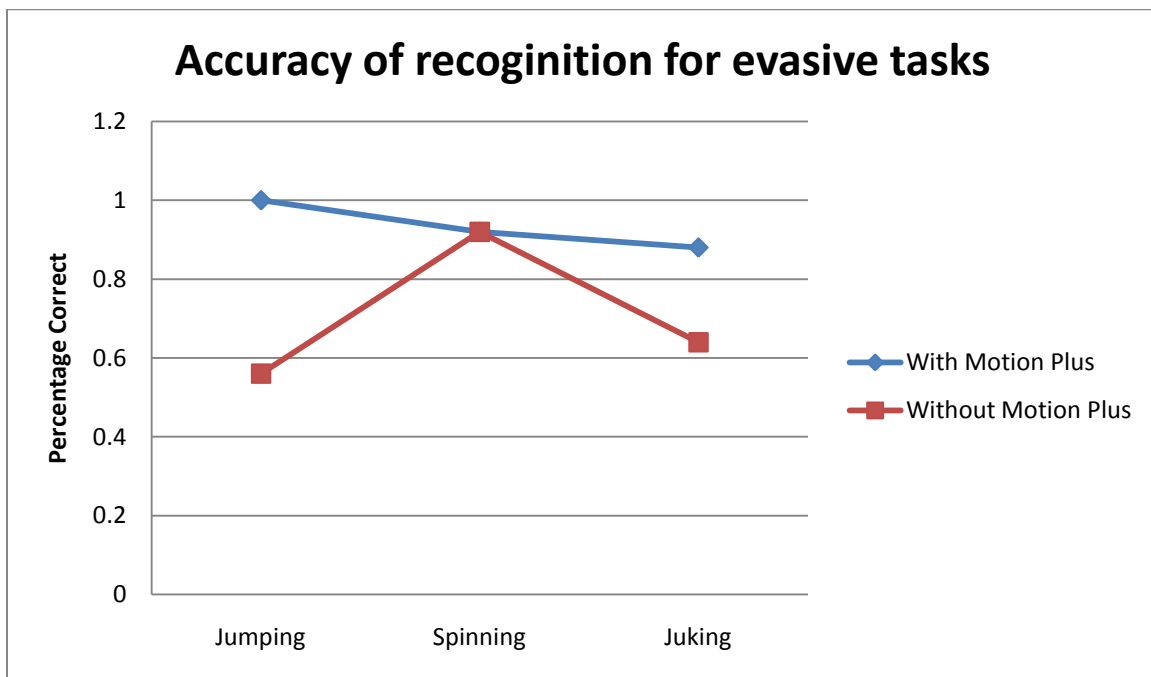


Figure 10 - Accuracy of recognition for evasive tasks showing increases with the Motion Plus

What can be seen here is that overall the percentage of correct recognition was far lower without the Motion Plus than with it. The spinning gesture was able to stay near the same, but was ambiguous with the other gestures in the absence of a Motion Plus. Though these gestures were developed heuristically, the added data from the extra hardware enhances the simple capabilities and shows potential for future growth with more gestures especially with more

advanced methods of recognition as has been seen in previous research (Hoffman, Varcholik, & LaViola, 2010).

Motion Plus Orientation Accuracy

In this experiment, we paired the Wii remote to the Intersense PC Tracker system for truth data (Intersense Inc., 2009). First, both pieces of hardware were made sure to reflect pitch and yaw in the same coordinate system. Second, two separate programs gathered data from them, attaching universal time stamps of the same format to the data. Then, with both systems running, the two devices were turned in a complete circle of rotation with respect to heading, and then pitched another full circle.

The first time was with the tracking system fully intact and the Wii remote without infrared to correct the Motion Plus. The experiment was repeated; this time with the Wii remote able to see the infrared when yaw approached zero degrees. The effect was to observe the effectiveness of the Motion Plus with and without infrared in relation to truth data of its orientation. The figures below show the results.

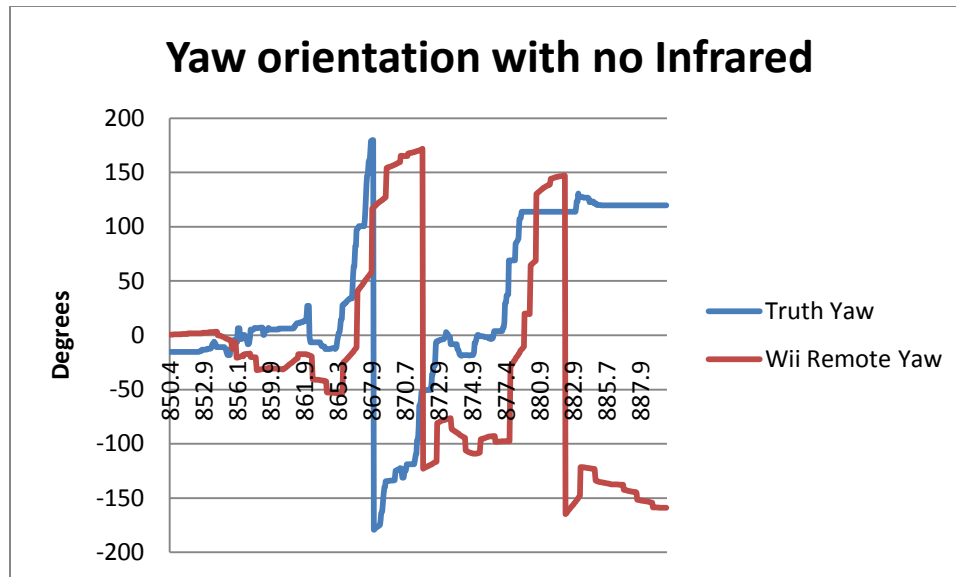


Figure 11 – User’s yaw orientation with the Motion Plus and without infrared

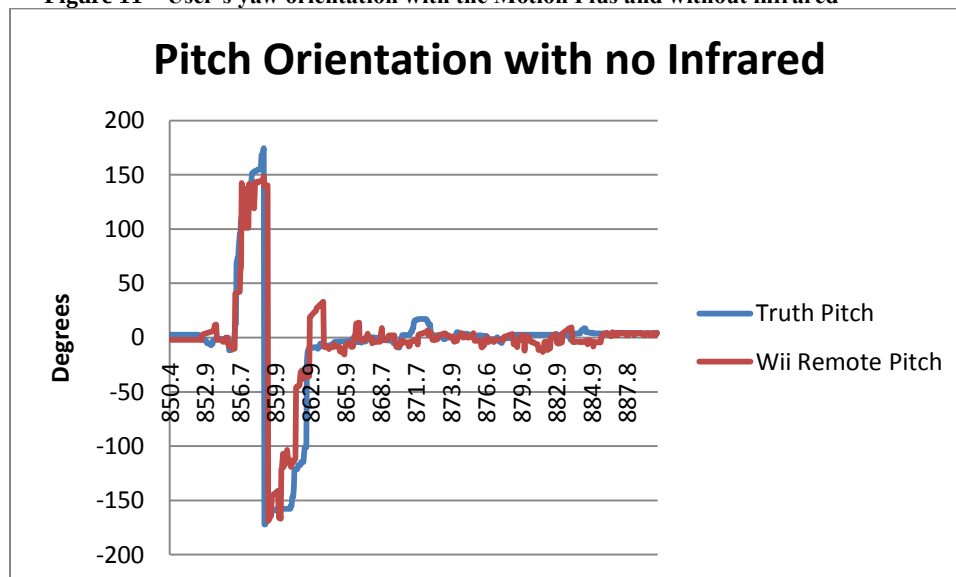


Figure 12 - User’s pitch orientation with the Motion Plus and without infrared

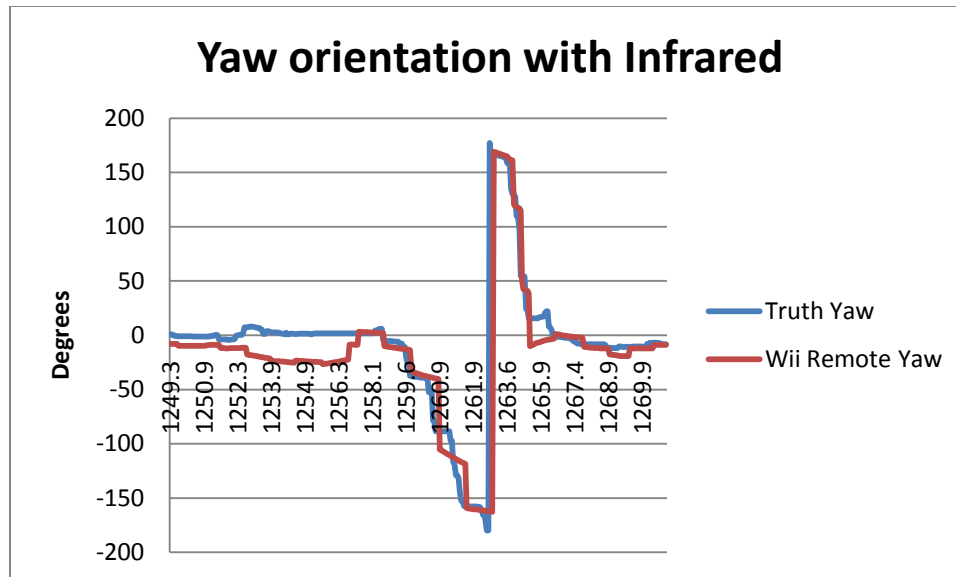


Figure 13 - User's yaw orientation with the Motion Plus and with infrared

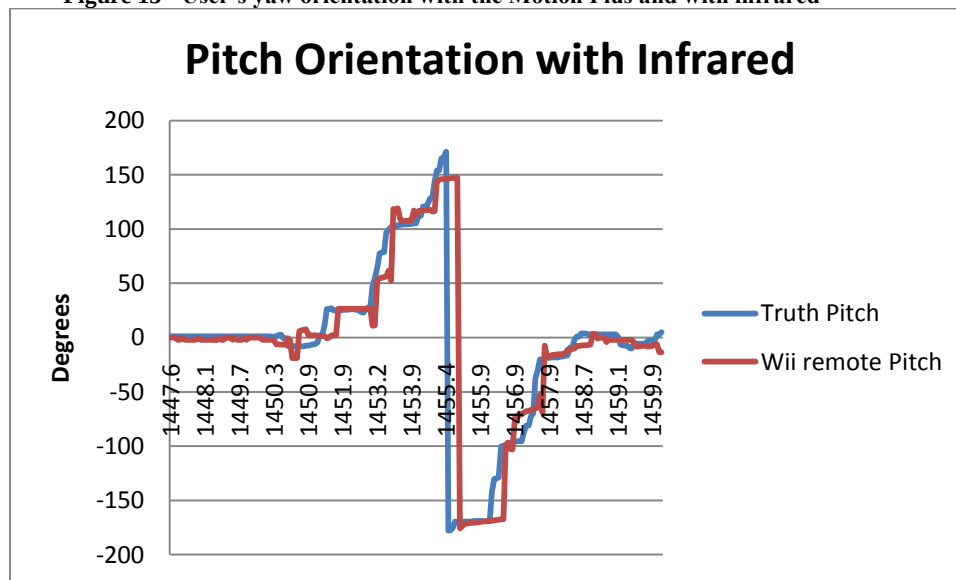


Figure 14 - User's pitch orientation with the Motion Plus and with infrared

The X axis in these figures represents time in seconds, rounded to the nearest tenth of a second. The sudden jumps in the data are signs of the non-linear orientation equations leaping from 180 degrees to -180 degrees. Furthermore, in respect to Figure 11 the gap of data seen at the end has been taken with the consideration of the data wrapping around at these points.

A pattern emerges that pitch data is accurate due to the measurement updates being available from the accelerometers. Yaw, however, can be nearly perfect with infrared involved; otherwise the drift can grow to 80 degrees in a mere 30 seconds. Even in 20 seconds it is near 65 degrees of difference. For the yaw angles, the calculated RMSE without infrared is 157.40 degrees and with infrared the RMSE is 16.33 degrees. The RMSE for the pitch orientation is 3.09 degrees.

Usability Study

For the user study performed, two sets of experiments were set up testing two variables. The first variable was whether users would prefer the second technique to the third technique and the second variable was preference to the Motion Plus providing head orientation even if it did not affect the challenge.

Subjects and Apparatus

Ten participants were recruited for this study by word of mouth, their demographics were six males and four females with a mean age of 25 and range of 19-28. They were placed in front of a computer screen with a hat on their head which had the Wii remote attached to it. The Motion Plus was added or removed as was needed for their test number.

Experimental Task

The first challenged designed tested the maneuvering aspect of the system. An object moved randomly on the screen in front of them, firing shots in the participant's direction (see

Figure 15). It was then their goal to dodge as many shots as they could in a preset time. This was referenced as the maneuvering task and an example can be seen in the following figure.

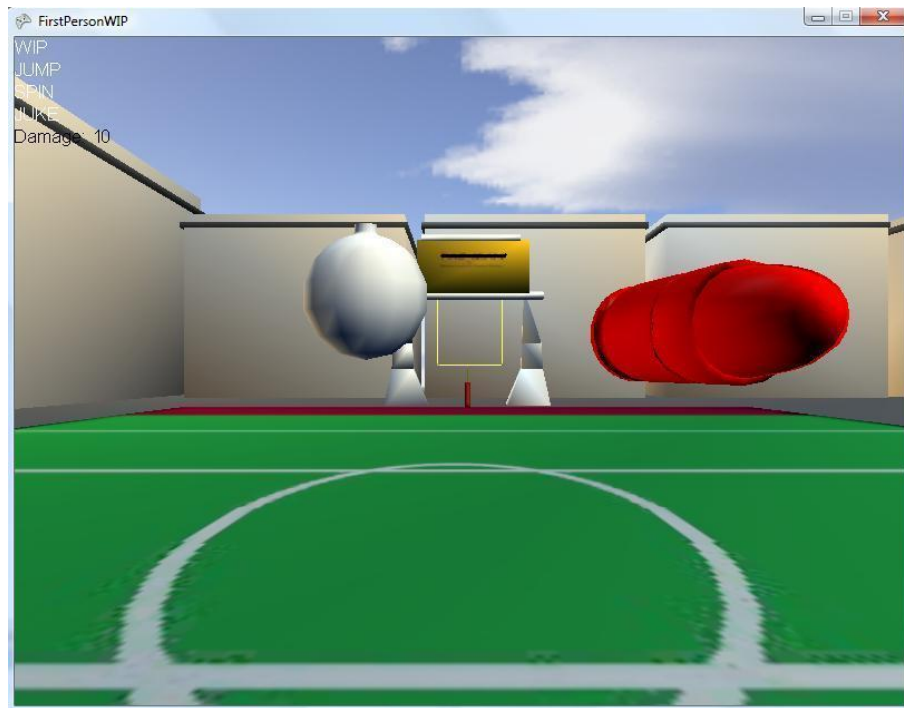


Figure 15 - Maneuvering task example

The second challenge involved evasive gestures and running in place. The player had to run in place down the field while facing several obstacles blocking their path. The first was a barrel on its side for them to jump over, the second was a barrel standing up for them to spin around, and the final was an opponent player pushing them back when they drew closer. This opponent could be defeated, however, if the player performed the juking gesture when they were near. Afterwards, they ran to the end zone, completing the trial. This became referenced as the evasive gestures task and is illustrated in the following figure.



Figure 16 - Evasive task example

Experimental Design and Procedure

To make sure ordering was not an effect, the following random order was designed. In this table, the number one stands for the second technique without the Motion Plus, the number two is for the second technique with the Motion Plus, three is for the third technique without the Motion Plus and four is for the third technique with the Motion Plus.

Table 1 - User study test sequence

User Number	First Test	Second Test	Third Test	Fourth Test
1	1	2	3	4
2	1	4	3	2
3	1	3	2	4
4	2	4	1	3
5	2	4	3	1
6	3	1	4	2
7	3	4	2	1
8	3	2	4	1
9	4	1	3	2
10	4	3	1	2

Each subject was presented one of the methods as given from the table, a practice run for the maneuvering task, then two normal trial runs that lasted exactly thirty seconds. After that, they were given a practice run for the evasive gestures task, followed by two trials, then a mandatory break to rest.

Users becoming tired became a significant factor in this, especially with the task involving running down the field. Because of this, only two trials could be performed, and breaks became mandatory as some users in the pilot study would not take breaks as long as they really needed.

Performance Results

Quantitative data was gathered from the two tasks as well. For the maneuvering task, the system recorded how many times they were hit by projectiles (ten points per hit) in the given thirty seconds. With the evasive gestures task, it was how long it took them to make it down the field. These results were then averaged between the two trials for the user, and then a sum average was taken for all participants. This data is shown in the figures below.

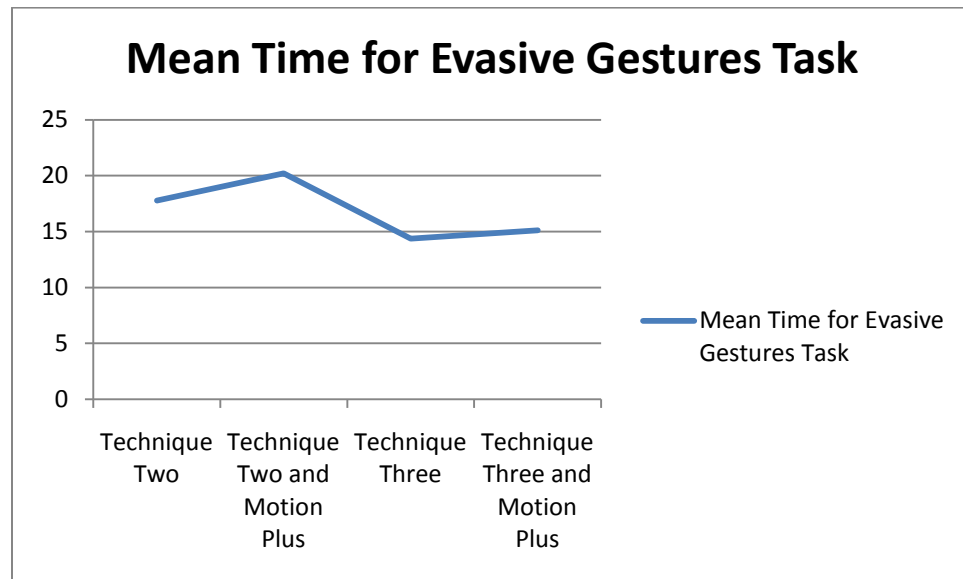


Figure 17 - Mean time for users in completing the evasive gestures task

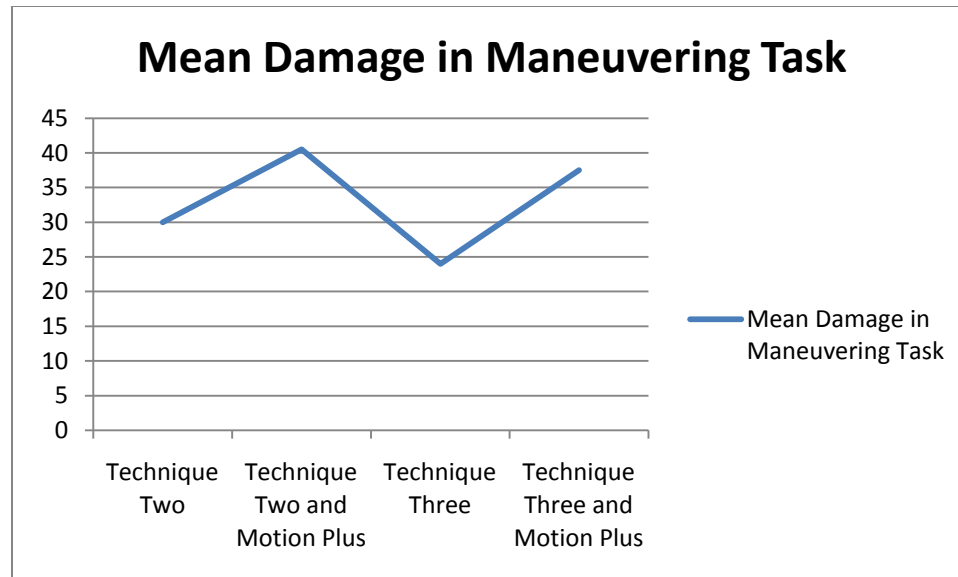


Figure 18 - Mean damage taken by users in the maneuvering task

This data shows a mean times hit of 3.3 with ranges from being hit zero to ten times and mean time to complete the evasive task of 15.85 seconds with ranges from 9.88 to 33.6 seconds. We performed an ANOVA analysis and found no significant difference between conditions for the maneuvering task ($F(3,7)=1.04$, $p = 0.39$) and the evasive task ($F(3,7)=1.41$, $p = 0.26$).

Subjective Results

Next we look to user preference which was gathered via a post-questionnaire administered after the four techniques were completed. They were asked to pick a favorite method for both the maneuvering and evasive gestures task and explain why. What they chose was broken out into the two variables being tested, which are whether the users prefer technique two to technique three and if they preferred a Motion Plus or not. The results are in the following figures.

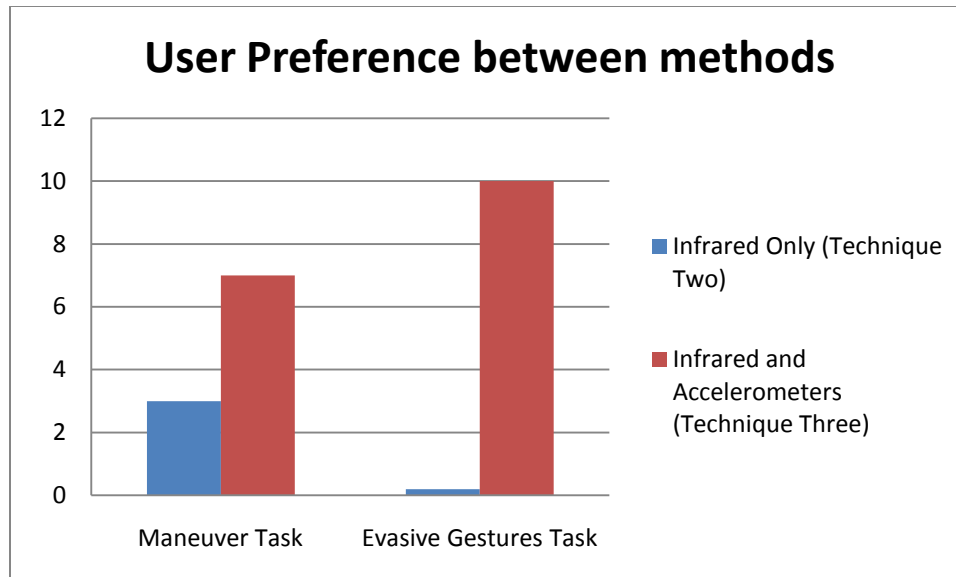


Figure 19 - User Preference between techniques two and three

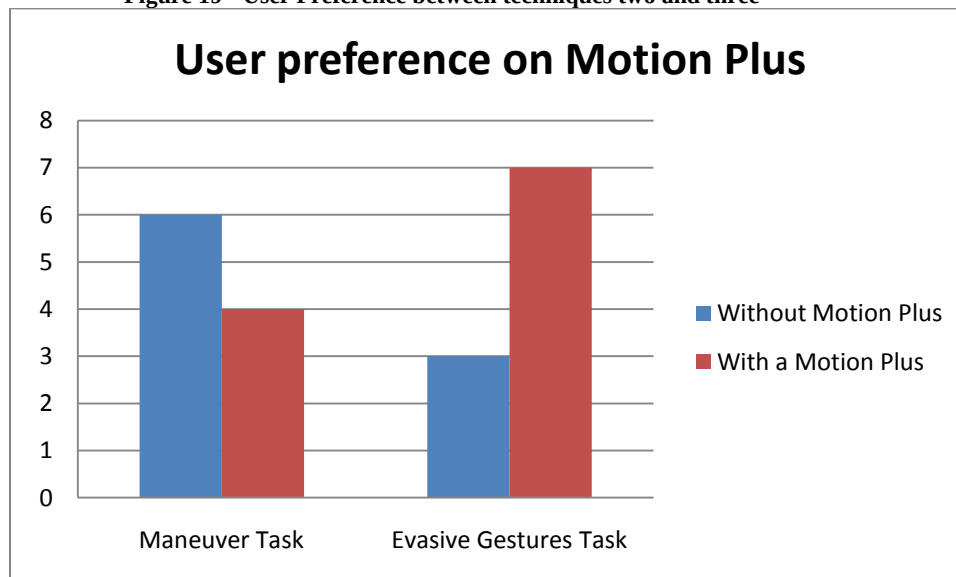


Figure 20 - User Preference between using the Motion Plus or not

Figure 19 shows that users preferred technique two to technique three with seventy percent picking it for the maneuvering task and every user picking it for the evasive gestures task. As for the Motion Plus, users were generally split for the maneuvering task, and chose it for the evasive gestures task seventy percent at the time.

In regards to the Motion Plus, most users verbal comments during the study were either expressing novelty toward it, such as one user amazed that they could look up and down, or annoyed by it, with one user stating that it gave them a headache. However, even those that saw it as a distraction in the maneuver task recognized that it helped with the gestures in the evasive gestures task, verbally commenting on how it appeared to be easier. This may suggest that unless the task calls for it, the Motion Plus is better with the data being used in the background than actually be rendered.

CHAPTER FIVE: DISCUSSION

With three techniques developed and analyzed, and the Motion Plus integrated on the last two, we look to the results from the analysis for discussion and future work.

Conclusion of Technique Analysis

After looking over the results, the third technique, which involved a Kalman filter blending infrared and accelerometers, stands out beyond the others, even the cost effective 6DOF tracker. It had further range than the standard, maintained its accuracy, recognized all of the gestures, and was preferred by users. The likely cause for the user preference is that there is no limit to the user's movement, especially when maneuvering. As one verbal comment was made, going from technique three to technique two, "I feel more constrained here; I can tell my movements are inside this small box".

This statement tells us that users ultimately want to be free to move to where makes sense to them. When the virtual world was restricted to the football field, their natural expectations kept them looking forward and moving on the field itself (no one tried to go into the stands for example), however if limited in what they wanted to do they did not prefer it.

For technique one, the system was not even presented to users as the accuracy was already determined to be too inadequate. While it was interesting to design, had no limitations in range, and could be placed anywhere on the user, it failed for the simple task of maneuvering and did not have enough data present to perform all of the evasive gestures in an acceptable manner.

As for technique two it performed well for evasive gestures and accuracy, but the bounds placed by infrared were a problem for most users. This may still be a preferable technique when position has to be perfect and the user won't be moving enough to really go outside of the infrared. However, it shouldn't be used when the system coming to a standstill by outside of infrared range negatively impacts the user.

With the TrackIR similar measurements came out for all categories. However, it did not have the range of the Wii remote infrared, was costlier, and more intrusive with two Wii remotes being placed near the user's legs. With this in consideration, the spatially convenient hardware was able to meet or surpass the cost effective 6DOF tracker.

Motion Plus Discussion

We also analyzed the Motion plus, both in integration and feedback. The accuracy measurements from the truth data shows that with infrared it can be accurate, but still had some jitter and inaccuracies to deal with. Regardless, we decided to include it in the user study to see if a preference emerged and what users generally thought about it.

In terms of technique selection, it was not preferred in the maneuvering task by sixty percent, with users split on finding it annoying or novel and interesting to use. Those that did not enjoy the device made comments about the head movement being more of a distraction since it did not relate to the task at hand. For the evasive gestures task, however, seventy percent selected it with some commenting that they can tell the gestures are recognized better. Users were also asked on the post-questionnaire to pick a number from one to seven on how much they

enjoyed using the Motion Plus. This result came out with an average of 4.2, users somewhat enjoying it, but not quite.

One technique for the future of the device is to not show the rendering of it unless the task absolutely requires it because as shown in the analysis of gesture recognition, the extra data helps with the evasive gestures implemented to a degree that users were able to observe the effect.

General Observations

There were other questions asked and results to look at from the user study, that lead to a section on general observations with the system. For example, to measure frustration with least favorite method, users selected their frustration levels on a scale from one to seven, with seven being very upset with the method. In this case, seven out of ten users selected a number higher than four, suggesting there was frustration with the least favorite techniques.

Also asked was whether participants would ever make use of this system to complement an existing exercise routine. They chose on a scale from one to seven the likelihood of such a use, with seven being very likely to use with exercise. The result was an average of 4.8 with eight out of ten users selecting a number higher than four. This suggests a potential use of 3D user interfaces to complement existing exercise routines.

Another question presented was whether the users could foresee themselves replacing the standard controller with a 3D user interface, especially for American football video games. Once again this was on a scale from one to seven and the result was an average of 4.3. This was selected by seven out of ten users, most saying they would use it sometimes in place of a

traditional controller and found the idea to be interesting. Those opposed to it listed some extreme reasons, with one stating they don't play video games and another saying the games they play don't lend themselves well to 3D user interfaces. The final opposed user stated that they preferred to play video games to relax and find moving around in the interface to not have the same affect.

Lastly some of the verbal comments made from the user study were of interest; participants greatly enjoyed the evasive gestures, particularly the juke gesture taking down an opposing player. Several users became absorbed in this task, leaping about and making comments on their success at taking down an opponent. With several users, aft the task, even though tired from running in place, they would make comments on how much fun they were having.

Future Work

Future work involved would be continued integration and perfection of the Motion Plus. The code used to determine yaw orientation from the infrared was more heuristic based than mathematics and an algorithm that could determine the angle in a general case. Also, the extended Kalman filter can be perfected; adding in control states that can be tweaked for optimal use would greatly improve the accuracy of the system. Finally, the roll angle was removed for this study as it was not found necessary, but should be integrated to create a true mapping to the user's head.

Also, the evasive gestures were mostly recognized heuristically, using the data available and the tasks to be integrated to determine what the user was doing. More sophisticated

methods, such as the Rubine algorithm and learning data can be integrated to create a more dynamic experience, with even more gestures being integrated.

Furthermore, there is future work that can be done with this system and a more advanced user study. Certainly a larger sample population can be selected from a higher demographic pool that would result in more accurate results, preferences, and discussions. This can lead to further analysis of RealNav being used for exercise, and other factors that can be added to make the game more enjoyable. Users seemed to prefer the juke gesture as it knocked down an opponent, perhaps all gestures could be used to deal with opponents, something more in the football realm.

Finally, the system itself can be expanded into the domain of American Football gaming. This thesis only dealt with travel down the field, but certainly other moves can be integrated and recognized. For example, jumping to catch a ball, or throwing a pass to a specific player based on head orientation and arm movements. Also, the game can take a third person perspective, with avatars moving on the field to further simulate the football game experience.

CHAPTER SIX: CONCLUSION

In this thesis we looked at the possibilities of natural 3D user interfaces with commercially available hardware and their possibility of locomotion solutions in video games. The domain of American Football gaming was used to drive the requirements, developing three tasks, maneuvering, running, and evasive gestures. Also, the hardware was picked as commercially available, particularly with focus on the Wii remotes.

With this, three techniques were developed as potential prototypes for the RealNav system. First though a comparison technique was built, which used the TrackIR (Natural Point, 2009) and two Wii remotes attached to the user's legs. The first technique was just a Wii remote placed near the user's core of mass that used only accelerometers. The second technique was with a Wii remote mounted to the head and using the infrared, and the third integrated infrared and accelerometers in a Kalman filter.

We also integrated in the new Motion Plus hardware with the Wii remote using an extended Kalman filter modified from one used in previous research (Azuma, 1995). With this device we were able to provide orientation data available in technique zero to techniques two and three. The first technique was not used as infrared was a necessity for correction of the yaw data.

Next the three techniques were analyzed in their accuracy and range of data against the cost effective 6DOF tracker, and user preference looked at from a user study. The Motion Plus was included in an analysis of gesture recognition accuracy, orientation accuracy of the device itself, and finally user preference as well. The results for the methods indicated the first

technique was too inaccurate for use, the TrackIR lacked the range of the Wii remote, and the third technique was preferred by users over the second.

As for the Motion Plus, it was as accurate as an advanced head tracking system (Intersense Inc., 2009) when infrared was present and could enhance gesture recognition probability. However, it was not preferred by most users when the task did not call for it, though they did recognize that it helped with gesture recognition.

Conclusions can then be made that the commercially available hardware, the Wii remote and Motion Plus, is capable of doing as much as a cost effective 6DOF tracker. This shows that systems can be developed for it to resolve the problem of locomotion in a 3D user interface, making something natural and fun for a user to play. The hardware now available in the home is capable of greatly expanding video games and will definitely pave the way for future technology.

REFERENCES

- Azuma, R. (1995, February). Predictive Tracking for Augmented Reality. Chapel Hill, North Carolina, United States of America: University of North Carolina.
- Bowman, D., Koller, D., & Hodges, L. (1997). Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. *IEEE Virtual Reality Annual International Symposium* (pp. 45-52). IEEE Press.
- Bowman, D., Kruijff, E., LaViola, J., & Poupyrev, I. (2005). *3D User Interfaces: Theory and Practice*. Addison-Wesley.
- Brightman, J. (2009, February 17). *Wii Attracting Young Boys, Older Women in Drove - Nielson*. Retrieved October 3, 2009, from GameDaily:
<http://www.gamedaily.com/articles/news/wii-attracting-young-boys-older-women-in-drove-ndash-nielsen-/?biz=1>
- Brooks, F. (1986). A Dynamic Graphics System for Simulating Virtual Buildings. *Workshop on Interactive 3D Graphics* (pp. 9-21). ACM Press.
- Charbonneau, E., Miller, A., Wingrave, C., & LaViola, J. (2009). Understanding visual interfaces for the next generation of dance-based rhythm video games. *ACM Siggraph Video Game Symposium* (pp. 119-126). New Orleans: ACM.
- Darken, R., Cockayne, W., & Carmein, D. (1998). The Omni-Directional Treadmill: A Locomotion Device for Virtual Worlds. *IEEE Virtual Reality Annual International Symposium* (pp. 12-19). IEEE Press.
- Feasel, J., Whitton, M., & Wendt, J. (2008). Low-Latency, Continuous-Motion Walking-in-Place., (pp. 97-104). Reno, NV.
- Foxlin, E. (2002). Motion Tracking Requirements and Technologies. In K. Stanney, *Handbook of Virtual Environments: Design, Implementation, and Applications* (pp. 163-210). Lawrence Erlbaum Associates.
- Gallo, L., De Pietro, G., & Marra, I. (2008). 3D interaction with volumetric medical data: experiencing the Wiimote. *1st annual conference on Ambient media and systems*. ICST.

- Hoffman, M., Varcholik, P., & LaViola, J. (2010). Breaking the Status Quo: Improving 3D Gesture Recognition with Spatially Convenient Input Devices. *To appear in IEEE VR*.
- Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., & Hallaway, D. (1999). Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. *Computers and Graphics* , 779-785.
- Interrante, V., Ries, B., & Anderson, L. (2007). Seven League Boots: A new Metaphor for Augmented Locomotion through Moderately Large Scale Immersive Virtual Environments. *Symposium of 3D User Interfaces* .
- Intersense Inc. (2009). *Intersense Inc - IS-900 systems*. Retrieved 10 16, 2009, from Intersense Inc - Precise Motion Tracking Solutions: http://www.intersense.com/IS-900_Systems.aspx?
- Iwata, H. (2001). GaitMaster: A Versatile Locomotion Interface for Uneven Virtual Terrain. *IEEE Virtual Reality* (pp. 131-137). IEEE Press.
- Iwata, H. (1999). Walking about Virtual Environments on an Infinite Floor. *IEEE Virtual Reality* (pp. 286-293). IEEE Press.
- Iwata, H., & Fujii, T. (1996). Virtual Perambulator: A Novel Interface Device for Locomotion in Virtual Environment. *IEEE Virtual Reality Annual International Symposium* (pp. 60-65). IEEE Pres.
- Kiefer, C., Collins, N., & Geraldine, F. (2008). Evaluating the Wiimote as a Musical Controller. *International Computer Music Conference*.
- LaViola, J., Keefe, D., Zeleznik, R., & Acevedo, D. (2004). Case Studies in Building Custom Input Devices for Virtual Environment Interaction. *VR Workshop: Beyond Glove and Wand Based Interaction* , 67-71.
- Lee, J. (2008). Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing* , 7 (3), 39-45.
- Motion Analysis*. (2009, September 14). Retrieved September 20, 2009, from WiiLi.Org Wii Linux: http://www.wiili.com/index.php/Motion_analysis
- Natural Point, I. (2009). *TrackIR :: head tracking view control immersion for flight racing and action simulator :: TrackIR :: Premium head tracking for gaming*. Retrieved September 20, 2009, from NaturalPoint :: Optical Tracking Cameras :: Motion Capture Solutions :: Hands Free Ergonomic Mouse Alternatives: <http://www.naturalpoint.com/trackir>

- Noma, H., & Miyasato, T. (1998). Design for Locomotion Interface in a Large Scale Virtual Environment - ATLAS: ATR Locomotion for Active Self Motion. *ASME Dynamic Systems and Control*, (pp. 111-118).
- Peek, B. (2008, June 7). *WiimoteLib -.Net Managed Library for Nintendo Wii Remote*. Retrieved September 20, 2009, from Brian's Blog - BrianPeek.com: <http://www.brianpeek.com/blog/pages/wiimotelib.aspx>
- Robertson, A. (2009). *Can You Really Get Fit With Wii Exercise Games?* Retrieved October 3, 2009, from WebMD: <http://www.webmd.com/fitness-exercise/features/can-you-get-really-fit-with-wii-exercise-games>
- Ruberg, B. (2009, January 29). *The Truth About Wii Fit And Weight Loss* . Retrieved October 3, 2009, from Forbes.com: http://www.forbes.com/2009/01/29/video-games-wii-technology-personal-tech_0129_wii.html
- Schou, T., & Gardner, H. (2007). A Wii remote, a game engine, five sensor bars and a virtual reality theatre. *19th Australian Conference on Computer-Human Interaction: Entertaining User Interfaces* (pp. 231-234). Adelaide: ACM.
- Shiratori, T., & Hodgins, J. (2008). Accelerometer based User Interfaces for the Control of a Physically Simulated Character. *ACM Transactions on Graphics* , 27 (5).
- Slater, M., Usoh, M., & Steed, A. (1995). Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality. *ACM Transactions on Computer-Human Interaction* , 2 (3), 201-219.
- Templeman, J., Denbrook, P., & Sibert, L. (1999). Virtual Locomotion: Walking in Place through Virtual Environments. *Presence: Teleoperators and Virtual Environments* , 598-617.
- Usoh, M., Arthur, K., Whitton, M., Bastos, R., Steed, A., Slater, M., et al. (1999). Walking > Walking-In-Place > Flying in Virtual Environments . *SIGGRAPH* (pp. 359-364). ACM Press.
- Vandewater, E., Shim, M., & Caplovitz, A. (2003). Linking obesity and activity level with children's television and video game use. *Journal of Adolescence* .
- Varcholik, P. (2009, July 28). *Bespoke Software >> 3DUI*. Retrieved October 3, 2009, from Bespoke Software: http://www.bespokesoftware.org/wordpress/?page_id=50
- Welch, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D. (2001). High Performance Wide-Area Optical Tracking: The HiBall Tracking System. *Presence: Teleoperators and Virtual Environments* , pp. 1-21.

- Whitton, M., Cohn, J., Feasel, J., Zimmons, P., Razzaque, S., Poulton, S., et al. (2005). Comparing VE Locomotion Interfaces. *IEEE Virtual Reality* , 123-130.
- Wingrave, C., Williamson, B., Varcholik, P., Rose, J., Miller, A., Charbonneau, E., et al. (2010). Wii Remote and Beyond: Using Spatially Convenient Devices for 3DUIs. *To appear in IEEE Computer Graphics and Applications* .
- Zanbaka, C., Lok, B., Babu, S., Xiao, D., Ulinksi, A., & Hodges, L. (2004). Effect of Travel Technique on Cognition in Virtual Environments. *IEEE Virtual Reality*, (pp. 149-156). Chicago, IL.