”Real Time Walking Detection Interface For The Virtual Environment”

Salvador Barrera
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
salvador@img.cs.titech.ac.jp

Hiroki Takahashi
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
rocky@img.cs.titech.ac.jp

Masayuki Nakajima
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
nakajima@img.cs.titech.ac.jp

Abstract

Currently, most virtual reality systems use upper body parts to interact with objects in the virtual environment. This situation is caused by technological limitations of current interface devices. The goal of this project is to develop and build a hands-free navigation system to be integrated into virtual environments. One of the most important fields in virtual realty (VR) research, is the development of systems that allow the user to interface with the virtual environment. The most intuitive method for moving through a virtual landscape is by walking. The implementation of a walking interface for a virtual reality system also allows a greater range of biomechanical experimentation and game research. Systems ranging from different platforms have already been implemented to produce virtual walking; however, these systems have been designed primarily for use with head mounted display systems. We believe that hands-free navigation, unlike the majority of navigation techniques based on hand motions, has the greatest potential for maximizing the interactivity of virtual environments, due to more direct motion of the feet. To make this possible, we created a simple device using acceleration sensors to detect ankle movements within the virtual environment. The sensors are attached to the foot and detect movement based on direction. This experimentation could prove beneficial in future virtual gaming. Validation of our approach is given by discussion and illustration of some experimental results.

Figure 1. Previous Foot Input Device

1 Previous work

Input devices on virtual environment have had an increasing demand on the research area in virtual reality [1]. The majority of the Virtual Reality input devices are hand-operated, such as the keyboard, mouse, joystick, pen-tablet, etc. Today’s interactive games, however, require more active and natural participation on behalf of the user. For this reason, researchers are going to investigate other new body parts as input devices.

This paper introduces a new approach to a virtual reality input devices focusing on the foot. We can use our feet for basic input operations, giving freedom to other body parts.

This led us to create a simple sensing device for detecting

Figure 2. ”Joyfoot” Foot input device based on foot motion sensing

ankle motions [2]. Our first version of foot input was based on detecting the orientation of the sole, converting it to a directional signal. This implementation had rotary encoder sensors on each foot sole as shown in Figure 1.

We implemented the interface to cope with the flaws in the design. More specifically, the first prototype suffered from lack of sensitivity in backward weight shifts. After many experimental results, we designed a new version of the foot input device known as the “Joyfoot” as shown in Figure 2.

“Joyfoot” permits users freedom for changing motion directions naturally. This allows users to input two operations simultaneously. Examples include pointing out an object and changing position at the same time. Users can express where they want to go or what they want to do trough natural movements[4]. This also allows the user to move or walk without making any step or hand movement, freeing other parts of the human body.

2 Methodology

“Joyfoot” consists of some acceleration sensors attached to a sandal for detecting motion. The sensors are connected to a strap directly below the knee with rubber bands.

The acceleration sensors sense foot motions and translate that action into movement. Specifically, the acceleration sensors measure the direction and amount of the foot’s motion as shown on figure 3 and 4.

According to the amount and the direction of the foot, the angle of the ankle is detected and translated to an electrical signal[5], the PC detects changes in voltage and calculates the direction data and inputs it into graphic system.

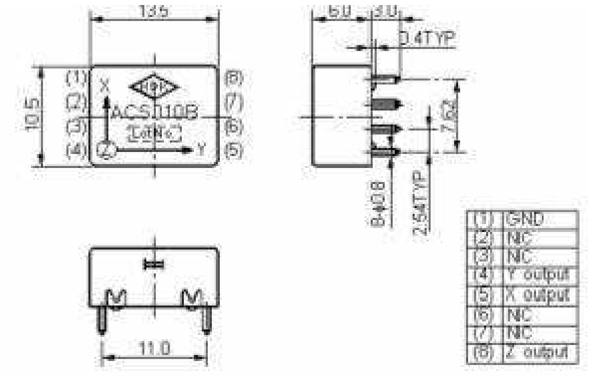


Figure 3. Acceleration Sensor Diagram

Since the level of voltage of the interface is in an analog form, we need a sampling process to convert it into a digital signal that our PC can manipulate. “Joyfoot” is connected to an A/D conversion board which takes care of the conversion in a rate that provides the user with high play ability and unnoticeable response times.

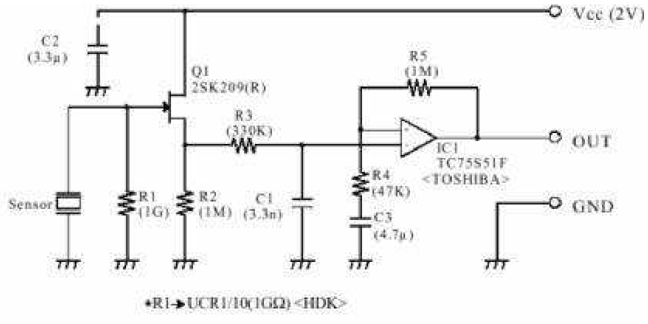


Figure 4. Circuit Diagram

The PC detects changes in voltage, calculates and sends direction data to the graphic system, according to increases or decreases in voltage.

3 Calibration and Data Transformation

Calibration is also needed before using the ”Joyfoot”. This requires input of five key-points: Center (C) to serve as the neutral position, Front (F), Left (L), Back (B) and Right (R). The vectors CR, CF, CL and CB divide the sensor plane to four parts, which are mapped to the four quarters of XY. Each of these vectors is moved, rotated, sheared and re-scaled to coincide with the vectors x, y, -x, -y of the target system. The transformation algorithm takes into account the possibility that the key points form a left-handed coordinate system, and manipulates the sensor values.

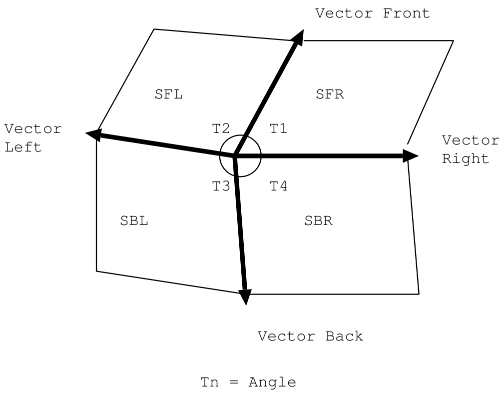


Figure 5. System's sensing plane

These points form a region of all the possible sensing values for each angle of the user's foot. This is called The user's sensing plane. Vectors from the Center point to the four other points are used to decompose arbitrary directions to a regular coordinate plane shown in Figure 5.

In order to map the system's sensing plane to a right-handed coordinate system, we calculate a transformation matrix for each quarter (eg. Front-Center-Left) of the sensing plane using the following formulas.

$$T_n = InvCos \frac{\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} * \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix}}{\sqrt{\alpha_1^2+\beta_1^2} * \sqrt{\alpha_2^2+\beta_2^2}} \qquad (1)$$

$$Vector \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \qquad (2)$$

$$Dot \quad product \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} * \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} \qquad (3)$$

$$Length \quad of \quad the \quad vector \sqrt{\alpha_1^2 + \beta_1^2} \qquad (4)$$

We need to calculate the angle for each quarter, for every quarter we have two vectors as shown in figure 5. Where α and β represents (x,y) depending on each quarter? and? $\sqrt{\alpha_n^2 + \beta_n^2}$ represents the lengths.

This transformation aims in making the Center Right and the Center left vectors collinear to xx axis, as well as the center front and center back vector collinear to the YY axis. The transformation also scales then sensor coordinates in order to normalize the input to one.

On the other hand, the program takes five points, center, right, left, front and back. For every point we start from zero, if it is right or left in their axes in order to know if it is perpendicular or not.

Sometimes the calibration fails and the system can not perform the conversion due to user error. In such a case, the conversion software will warn the user and the calibration will be repeated. Optionally, the user can then choose to use a logarithmic function to increase the stability around the center without diminishing the responsiveness of the interface.

The values returned by the sensors simply refer to angles corresponding to distances between points on the foot and points around the knee. These values are not very useful when used directly. For this reason, we used the device to input a user's positional movement. We made a data transformation algorithm using two sensor values to calculate X and Y directional acceleration values shown in Figure 6. The basic idea of the data transformation algorithm is a space transformation. We used the following method to define the relationship between the sensor values and the XY values.

Figure 6. Calibration Equipment

The algorithm requires 5 points of basic postures (center, front, back, right and left) in sensor space. Those points are related respectively to points (0,0), (0,1), (0,-1), (0,1), (0,-1) in XY space.

The movement, rotation, shear and rescaling have to be calculated for each one of the four parts of the sensor plane and the results are stored in a transformation matrix.

The sensor's input value is multiplied every time with the correct part of the transformation matrix and then manipulated in various ways to include a customizable stability around the neutral point, a measure correction to make more efficient the corner positions, and the application of logarithmic functions to improve response time.

In this application, because we choose a game which only requires two dimensional input, we use only two sensor values. However for other cases we require three dimensional values.When the users change movement according to motion, the sensors receive information. Since we considered the structure of the human legs, it was possible to detect different movements.

4 System Architecture

We use a stereo viewing system for displaying. Various scanning modes are being investigated, which results in a number of systems incompatible to one another.

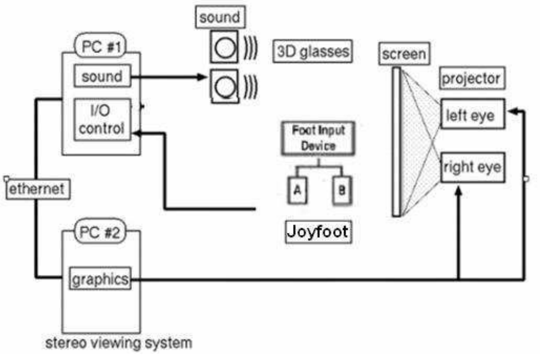
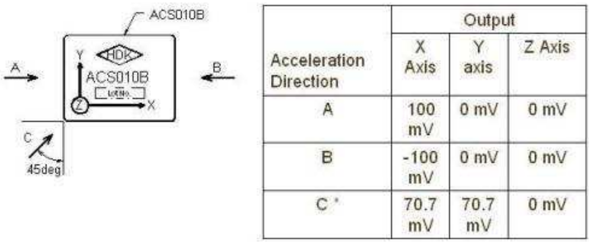


Figure 7. Overview of the Architecture System

We address the problem of the interconnection of such a device through standard conversions by a signal processing approach, namely the model of a universal standard converter, which is based on a layered functional architecture. The concept of a virtual standard is introduced for stereoscopic signals. When this machine receives direction data, it redraws a picture according to the data such as figure 7 and figure 8. In summary, the system receives direction data from the "Joyfoot" unit, reconstructs the scene picture and transmits it into the projector.

5 Application Game

In this section, we will show how the proposed approach can be applied to a Windows game application. A game player first puts on the device and stands in front of a 100 inch projected screen. Then, by using "Joyfoot" the player goes through a virtual Aerial Circus game, which has many routines? (Jumping, Trampoline, Bungy Swing, Flying rope), as shown in Figure 9. The game player must follow the instructions in order to select one of the games before start the game.



Acceleration Direction	Output		
	X Axis	Y axis	Z Axis
A	100 mV	0 mV	0 mV
B	-100 mV	0 mV	0 mV
C *	70.7 mV	70.7 mV	0 mV

Figure 8. As a result, the output voltage is proportional to the acceleration from the corresponding axis output terminal

The game rules are very simple and easily understood. This game requires the player to use his/her feet as shown in figure 9. Some excited game players might use their whole body for input, increasing the exhilaration of the gaming experience.

Figure 9. Game Playing Scene

For instance to control the timing of somersault using Joyfoot, users should move right to control somersault timing. Points are given by 3 different movements. The first moves the acrobat, such as her knees when landing. It is a preparation step for jumping. When the acrobat has jumped high enough we move right to control the dummy.

Playing the game, players can see that "Joyfoot" is a very intuitive device and will understand the usefulness and fun of using both their feet in a VR gaming environment.

6 Impact Application Area

For desktop computing. “Joyfoot” could be used instead of a mouse as a pointing device, or as second pointing device. When used as the main pointing device, a user can move a pointer on the screen while typing a keyboard. In 3D modelling for example, a user often wants third hand. While the right hand is occupied moving a point, the left hand is also occupied typing a shortcut key to change operation mode. The modeler may also want to simultaneously change the viewpoint, which he could do using “Joyfoot”.

In Virtual Reality systems, “Joyfoot” is very powerful. When you move objects from one point to another point, such operations using both hands and feet are natural in the real world, and would incorporate actions such as grabbing, walking and dropping. In this case, assigning walking operations to foot-controlled input is a better solution than using hand actions for walking.

In another example, assume that we want to complete a more complex task in a VR environment. When a chief engineer checks a design for complex pipe distribution in a plant, using a VR system such as a CAVE, and finds a point requiring inspection, he may want to take a memo with his/her palm PC (which is connected to the VR host computer). His writing is related to the point coordinate and/or? the bad object immediately. Ideally, his/her hand should be ready for writing when he notices something. Therefore assigning movement operations to foot action in the VR environment is very important.

There is much demand to have hard-to-reach information accessible at awkward times: a craftsman may want to read a blueprint while working, a surgeon may want to watch a visualized CT data whilst operating, a engineer may want to read manuals while doing maintenance, and so on. “Joyfoot” will be a useful device as these ideas become a reality.

7 Experimental Results

As measurement results the order of Front (F), Left (L), Back (B), Right (R) is preserved, even though it is distributed and depends on how much the person can move his foot at the time they wear the input device , as shown in figure 10.

The direction sensitivity of measurement can be defined by the area of a quarter, divided by the area of all quarters. Since the system is consistent and relatively sensitive to any direction, we can define the direction sensitivity measured by the following formula.

$$\text{Sensitivity} = \frac{\text{Area of a Quarter}}{\text{Area of all four Quarters}}$$

$$\text{FR sensitivity is given by} = \frac{\text{SFR}}{\text{SFR + SFL + SLB + SBR}}$$

As an example of Front, Right

According to the relative sensitivity to directions, we conclude that from Forward to Backward there is a big movement and equal to the sensors range, not as right and left movement.

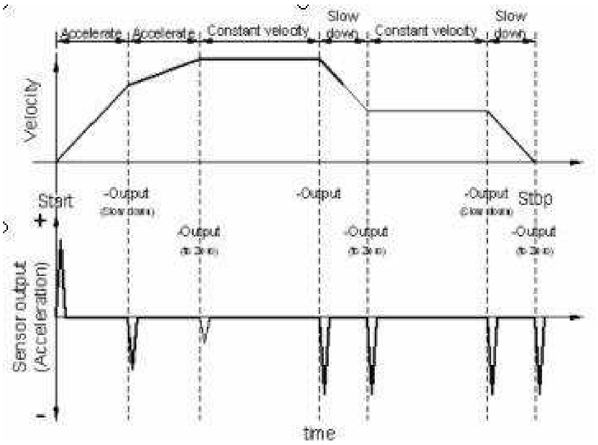


Figure 10. Output generated when acceleration was applied

As an experimental results we showed that a sensitivity range between 10 and 50 percent gave us an acceptable levels of control as shown in Figure 11.

8 Comments

The project presents a multi-modal interface that provides a natural space, where the user can interact with a virtual environment via the 3D graphics, audition and natural movement.

1. Replacing input with a keyboard or mouse, foot input devices systems are intuitive for current operations using hands and feet naturally in the real world. 2. The design of

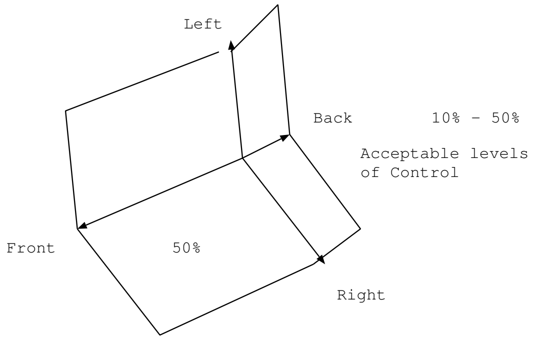


Figure 11. Measurement results show forward and backward movements equal to the sensors range

this interface opens the way to a new foot interface based on foot position and motion measurement. These techniques are shown to be simple and sufficiently accurate. As well, this link technology does not disturb the visual interaction and keep enough freedom for the user.

9 Conclusions

In this paper we have presented a new VR input device that can be objectively used in many VR applications. Compared with the traditional input devices such as keyboard and mouse, ”Joyfoot” opens a new wearable user interface technology with motion detected from the foot.

As an advantage of this device, we can conclude that by giving freedom to other parts of the body, users can interact easily in the virtual world, making hands-free from motion operation. Users can input two operations at one time. Foot operation is very useful for moving naturally around the virtual world. Working on a new foot input device has demonstrated the power of body sensing, giving freedom to our hands and making games interactive to users.

The streamlined configuration requires only a few sensors. The design of this interface also eliminates mechanical parts that could wear out. “Joyfoot” opens the way to a new foot interface based on foot position and motion measurement.

References

[1] S. Barrera, H. Takahashi, M. Nakajima, A new Interface for the Virtual World Foot Motion Sensing Input Device, Conference Abstracts and Applications, SIGGRAPH 2002 Computer Graphics Annual Conference Series, San Antonio: 141.

[2] H. Iwata, and Fujii,T. Virtual Perambulator: A Novel Interface Device for Locomotion in Virtual Environment: Proc of IEEE 1996 Virtual Reality Annual International Symposium.

[3] Y. Shinogawa, Kunii T.L., Nomura Y., Okuno T., Young Y. (1990) Automating View Function Generation for Walkthrough Animation Using a Reeb Graph, Proceedings Computer Animation 90 Springer, Tokyo: 227-238.

[4] Myers. B.A, “User Interface Software Tools”, ACM Transaccions on Computer Human Interaction, 1995 2(1): pp 64-103

[5] Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R., and Hart, J.C. The CAVE, Audio-Visual Experience Automatic Virtual Environment. Communications of the ACM, June 1992, pp 67-72

[6] Kennedy, R.S., et al. Profile Analysis of Simulator Sickness Symptoms: Application to Virtual Environment Systems. Presence, Volume 1, number 3, Summer 1992, pp 296

[7] Davis RB, Oumpw S, Deluca PA (1997)“Gait Data Reporting, Archiving and Sharing”, in Allard P, Coppozzo A, Lundberg A, Vaughan CL “Three Dimensional Analysis of Human Locomotion”, John Willey and Sons Ltd. Chapter 19, pp. 173-189.

[8] Myers, B.A., ”User Interface Software Tools.” ACM Transactions on Computer Human Interaction, 1995. 2(1): pp. 64-103.

[9] Barfield W. Zeltzer, D., Sheridan, T., Slater, M. (1995). Presence and performance within virtual environments. Virtual environments and advanced interface design (pp. 473-541). New York: Oxford University Press.

[10] Card, S.K., ”Pioneers and Settlers: Methods Used in Successful User Interface Design,” in Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context. Rudisill, et al., Editors. 1996, Morgan Kaufmann Publishers: San Francisco. pp. 122-169.

[11] Figueiredo, M., Bohm., Teixeira,J. Advanced Interaction Techniques in Virtual Environments. Computers and Graphics, Volume 17, number 6, 1993, pp 655-661