USER INTERFACE ISSUES FOR A BUILDING WALKTHROUGH SYSTEM WITH MOTION PREDICTION

Jiann-Rong Wu, Yuong-Wei Lei, Bing-Yu Chen, Ming Ouhyoung
Communication & Multimedia Lab.
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan. R.O.C. 106

Abstract

We discuss two user interface issues for a head-mounted display (HMD) based building walkthrough system: non-standard virtual reality(VR) input devices and perceived latency. We propose an useful procedure to help the user to attach various VR input devices easily. A Grey system based motion prediction method is applied to reduce the latency in our building walkthrough system.

Introduction

Traditionally, the user interfaces are designed to handle the 2D based manipulation. However, human's eye has six degree of freedom(DOF) in 3D-space. While the goal of user interface design is to make one feel as if he is moving in a realistic space in natural way, most interface devices are not intended for motion control in 3D, therefore the mapping of control actions to motion is not intuitive. For a building walkthrough system, we needs some advanced devices to manipulate in 3D, including such device as an HMD with a 3D tracker for exploring plus a treadmill for simulating walk in a 3D environment [1].

For a VR based system, such as building walkthrough, the user interface coordinates input devices with the display module. A flexible interface to deal with variety of input devices is important. However, as VR systems become more popular, many hardware manufactures develop their own VR devices and programming libraries. Lacking of a standard for these devices, the VR system programmer must pay more effort in adopting the devices and try to synchronize among them, since it is a time consuming process. The VR system developer needs a general procedure both to hook those devices easily and be system independent.

We have constructed a computer model of our newly constructed Computer Science Department building and developed a polygon-based walkthrough system, the SpaceWalker (Figure 1). The current system supports interactive walkthrough of the building model comprising 40180 triangles. Thanks to the visibility test to minimize the triangle's number for rendering, the update rate can get about 15-20 frames/sec on an SGI Indigo^2 Extreme graphics workstation. Even though the update rate is enough to work, it still has about 300ms overall latency; the tracker contributes approximately 70ms, the rendering pipeline 80ms, and the relatively low refresh rate LCDs in HMD contributes 150ms.

In developing an HMD based VR system, one of the critical problem is the perceived latency, which is the time delay between hand movement and its corresponding motion of the virtual object on the screen. The latency problem will be similarly serious when more devices attached in a VRML based 3D browser system on Internet.

To compensate for the latency, many proposed methods use prediction in motion tracking [2,3]. Several HMD based VR systems are implemented with head tracker prediction, where a look ahead algorithm is implemented using the 3D position and orientation as the input data. In 1994, we conducted an experiment on the latency and its compensation methods in a VR application using an HMD and a 3D tracker [4]. Both in the simulation and a flying target tracking experiment, the result of motion prediction of the tracker was better than the one without prediction. However, it is just an experimental result, and we would like to know how it behaves in a real application.

(a) The system overview. I/O devices: an HMD with a 3D tracker and a treadmill.

(b) The main window shows the scenario of the system, while the child window shows a user's motion.

Figure 1. The SpaceWalker system.

A General Procedure to Attach VR Devices

In the proposed procedure, there are three components included: the application, the device handler, and the shared library (Figure 2). The shared library creates a block of shared memory and defines five subroutines to deal with initialization, removal, and reading/writing of the shared memory. Both the application and the device handler components link with the shared library. The allocated shared memory can be taken granted as the bridge between the two programs for data transferring.

For each device, it has its own shared library and device handler components. These two components are pre-coded and already in the computer. All the things that the VR system designers have to do are forking a process to execute the device handler, get the data from the shared memory, and remove the device handler and the shared memory when the work terminates. When the device handler is forked and executed, it creates the shared memory and puts the data retrieved from the input device into the shared memory repeatedly.

In our building walkthrough project, we apply this method to attach (1) the 3D tracker to an HMD, (2) the external buttons for left or right turn, and (3) a shaft-encoder based treadmill system for measuring walking speed.

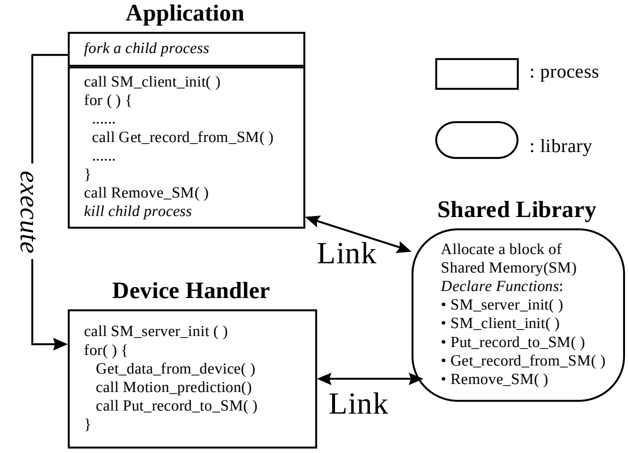

Figure 2. The block diagram of the general procedure to attach the input devices.

Motion Prediction

Once various devices are attached to the VR system successfully, we may think that the system can work but unfortunately, in most cases, it can't be used for human beings. One of the big problems mentioned in VR research is the latency. Especially, in an HMD related system if the latency is relatively large, it will cause motion sickness. One proposed method is to use prediction in tracking for achieving the dynamic registration and trying to eliminate motion sickness.

Grey System Based Motion Prediction

In real world, the behaviors of most systems are uncertain. The effects of other systems to a system under monitoring are also unclear. Thus, it is hard to exactly analyze and predict the behaviors of such systems. In Grey System theory, the system model is established under a sequence of measured raw data which is generated by the system with unclear system characteristics. The observed sequence is used to generate a generating sequence on Grey Generating Space, and a Grey differential model (GM) is applied to fit the generated sequence. By using the established GM, we can predict, analyze, and program the behaviors of the original system.

In most cases, the raw data observed by measuring the system is lack of relation and insufficient to establish Grey Model. Some manipulations on the raw data are needed to get a more regular data sequence, and the so-obtained sequence is called the generated sequence.

The most commonly used Grey Generating Space is the Accumulated Generating Space (AGS), and the operation of AGS is called Accumulated Generating Operation (AGO). Let $x^{(0)}$ be the original sequence and $x^{(i)}$ be the generated sequence for $i > 0$. The AGO is defined as:

$$x^{(i)} = \text{AGO}\, x^{(i-1)}, i > 0 \quad (1)$$

$$x^{(i)}(k) = \sum_{m=0}^{k} x^{(i-1)}(m), i > 0 \quad (2)$$

Let $^{(0)}$ be the original sequence, and $^{(1)}$ = AGO($x^{(0)}$), then assume they satisfy the following first order Grey differential model, GM(1,1), with single variable:

$$x^{(0)}(k) + a\, z^{(1)}(k) = b, k = 1, 2, \ldots \quad (3)$$

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k-1)}{2}, k = 1, 2, \ldots \quad (4)$$

which is obtained from the following differential equation:

$$\frac{dx^{(1)}(t)}{dt} + a \bullet x^{(1)}(t) = b \quad (5)$$

By solving a, b, and the differential equation, we can get the prediction function for the original grey system:

$$\hat{x}^{(1)}(k) = \left(x^{(0)}(0) - \frac{b}{a} \right) e^{-ak} + \frac{b}{a}, \text{for} k \geq 0 \quad (6)$$

Applying prediction length, k, in terms of the update rate as k to the prediction function, we can get the predicted data in AGO domain, and thus the predicted data $\hat{x}^{(0)}(k)$ can be obtained from Eq.(7).

$$\begin{cases} \hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1), \text{ for } k > 0 \\ \hat{x}^{(0)}(0) = \hat{x}^{(1)}(0) = x^{(0)}(0) \end{cases} \quad (7)$$

Conclusion

We had planned the construction of a walkthrough system for three years. Only when solution tools corresponding to above two issues are ready, can we build the whole system effectively in a short time. It seems that we have proposed effective ways in dealing with the user interface issues in developing and head-mounted display based virtual reality system.

Reference

[1] Y.-W. Lei, "Image-Based and Polygon-Based Techniques for Interactive Walkthrough of Architectural Models," *Ph.D. proposal, National Taiwan University*, 1995.

[2] J.-R. Wu and M. Ouhyoung, "Reducing The Latency In Head-Mounted Displays By a Novel Prediction Method Using Grey System Theory," *Computer Graphics Forum*, Vol. 13, No. 3, pp. c503-c512, 1994.

[3] R. Azuma and G. Bishop, "Improving Static and Dynamic Registration in an Optical See-through HMD," *SIGGRAPH'94 Conference Proceedings*, pp. 197-204, 1994.

[4] J.-R. Wu and M. Ouhyoung, "A 3D Tracking Experiment on Latency and Its Compensation Methods in Virtual Environments," *The Eighth Annual Symposium on User Interface Software and Technology (UIST'95)*, ACM Press, pp. 41-49, 1995.