Video-Based Measurement of System Latency

Ding He, Fuhu Liu, Dave Pape, Greg Dawe, Dan Sandin
Electronic Visualization Laboratory
University of Illinois at Chicago
{eric, liufuhu, pape, dawe}@evl.uic.edu , dan@uic.edu

Abstract

We describe an end-to-end latency measurement method for virtual environments. The method incorporates a video camera to record both a physical controller and the corresponding virtual cursor at the same time. The end-to-end latency can be concluded based on the analysis of the playback of the videotape. The only hardware necessary is a standard interlaced NTSC video camera and a video recorder that can display individual video fields. We describe an example of analyzing the effect of different hardware and software configurations upon the system latency. The example shows that the method is effective and easy to implement.

1. Introduction

This paper describes a simple to implement method for measuring end-to-end system latency in projection based virtual environments such as CAVEs® and ImmersaDesks® [Cruz93].

Interactivity is an essential feature of virtual reality systems. System end-to-end latency, or lag, is one of the most important problems limiting the quality of a virtual reality system. Other technological problems, such as tracker inaccuracy and display resolution do not seem to impact user performance as profoundly as latency [Ellis99]. In augmented reality, the system latency has even more impact on the quality of the virtual experience. Latency will make the virtual objects appear to “swim around” and “lag behind" real objects [Azuma95]. A prerequisite to reducing system latency is to have a convenient method of measuring it.

The system end-to-end latency is the time difference between a user input to a system and the display of the system’s response to that input. It can be the time delay from when the user moves the controller to when the corresponding cursor responds on the screen, or it can be the difference from when the user moves his or her head to when the resulting scene is displayed on the screen. The end-to-end latency is composed of tracker delay, communication delay, application host delay, image generation delay and display system delay [Mine93].

In this paper, we describe a video camera and recorder based measurement of the end-to-end latency of virtual reality systems. This latency measurement system uses an ordinary video camera to record movements of the tracked wand along with its virtual representation in a CAVE or an ImmersaDesk. The recording is viewed on a field-by-field basis to determine total delay.

2. Previous Work

Bryson and Fisher [Bryson90] drew a virtual cursor in the computer display according to the real position of the controller. They then superimposed a video image of the controller position and the video signal from the computer display using a video mixer. In one series of tests, by knowing the video frame rate, they calculated the time difference from a sudden movement of the controller and the following motion of the virtual image of the controller. In the second series of tests, they measured velocity of the sensor and displacement errors between the tracker and the virtual marker to estimate the time lag.

Liang et al. [Liang91] measured the latency of orientation data of electromagnetic trackers. The tracker sensor was affixed to a pendulum. The computer stored each reading and the corresponding time stamp. Simultaneously, a video camera recorded the pendulum swing along with a computer monitor displaying the current time of the clock used to generate the tracker time stamp. They then looked up the time stamps of zero position crossings in the stored tracker data and found the corresponding displacements. The displacement can be easily converted to lag time.

Mine [Mine93] analyzed and measured all components of the end-to-end latency in a HMD system. They also mounted the tracker sensor on a gravity pendulum and marked the zero position crossings by the swinging pendulum's optical interruption of an LED-photodiode pair. Tracker latencies were estimated on an oscilloscope by comparing the timing of the photodiode's zero-crossing transitions against the analog signal output from a D/A converter on the host. Furthermore, when the computer graphics application detected a zero crossing, it toggled a single polygon from black to white or vice versa on the screen. The signal from the second photodiode monitoring changes in the polygon's brightness was then compared with the first photodiode's zero crossing to provide an estimate of overall end-to-end system latency.

Adelstein et al. [Adelstein96] implemented an experimental testbed and method for precisely quantifying the components of tracker latency directly attributable to the transduction and processing internal to tracker sensors. Instead of using pendulums, they use a motorized rotary swing arm to sinusoidally displace the tracker sensor at a number of frequencies spanning the bandwidth of volitional human movement. During the tests, an optical encoder measured the swing arm angle coupled directly to the motor shaft. Both the actual swing arm angle and tracker sensor reports were collected and time stamped. Systematic biases including both software instruction execution time and serial data transmission time were subtracted from actual reports. The latency estimates of both position and orientation were derived from a least-squares fitting of each encoder and tracker sensor record to an ideal sinusoidal model.

The methods described above have some drawbacks and limitations. Specialized hardware (video mixer, pendulums, motorized rotary swing arm) is required in each of the techniques. Also, in some cases only certain latency components are measured, providing a partial systems analysis.

3. The Method

In our method the VR system displays the controller (wand) position and a fixed grid. The user moves the wand back and forth at moderate speed, while a video image from a camera shows both the real wand and a cursor representing the wand simultaneously. See Figure 1. The recorded video is analyzed to determine the lag between wand motion and the motion of the virtual image of the wand. The number of video fields delay between a grid crossing of the real wand and its virtual image, determines the total system delay with a resolution of 16.7 ms. Since typical VR systems experience latencies on the order of 40 - 150 ms [Bryson90] [Mine93], this resolution is sufficient for many applications.

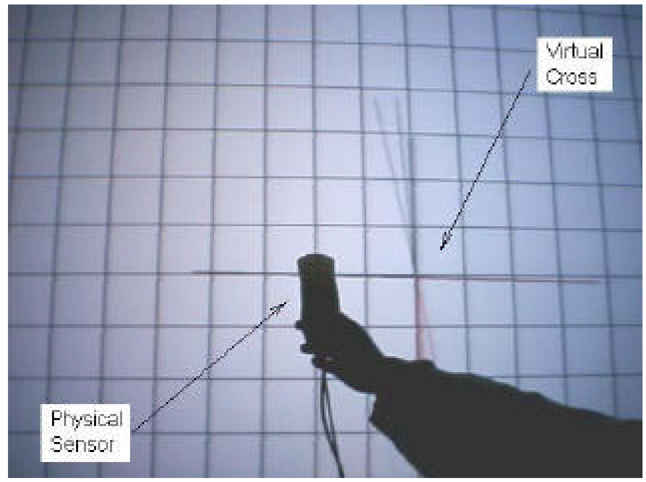
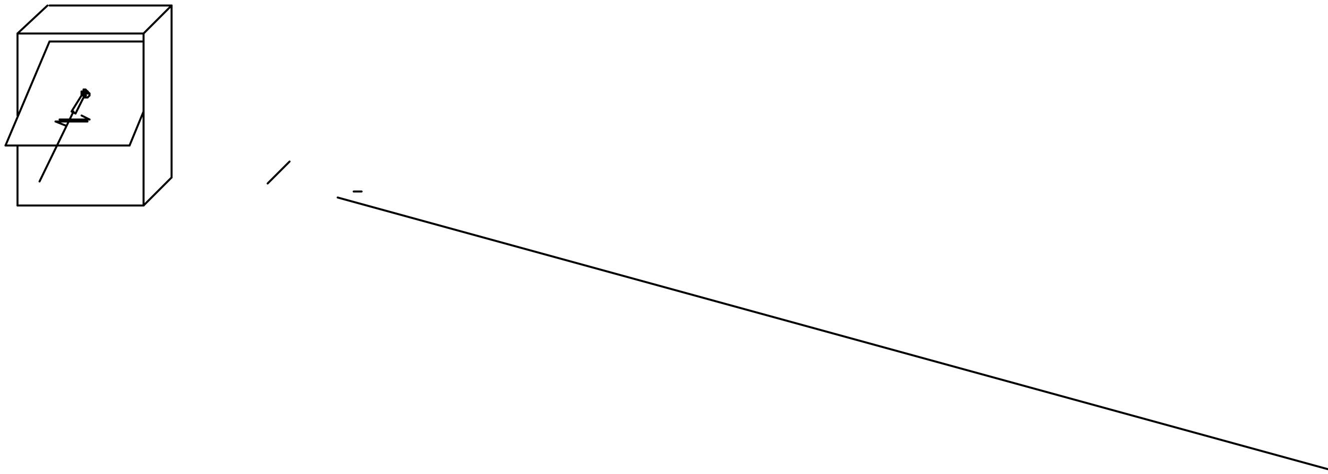


Figure 1. Physical sensor and virtual cross.

3.1 Description

This method is easy to implement in projection based virtual environments such as the CAVE or ImmersaDesk. The only equipment required is a video camera and video cassette recorder. The video recorder/player must be able to display fields and have a stable method of "jogging" between frames. This method includes all components of the end-to-end system latency.

During the experiment, we waggled the wand controller in front of the screen. The virtual representation follows the wand, but with some latency. The distance from the wand to screen was kept as small as possible in order to reduce the parallax. The eyeglasses on which the head tracker sensor was attached were fixed beside the video camera so that the movement of virtual cross was only due to the wand movement. We changed the frequency of waggles from fast to slow in the normal range of a human being, around 2 – 0.5 Hz, to simulate the normal movements of the wand. The amplitude was approximately 3 feet. The frame rate of the application was 60 fps when running on a SGI Onyx, which will introduce a latency of 16.7 ms because of double buffering. In actual applications, the 3D scenes are often very complicated and contain thousands of polygons and rich texture; the frame rate in these cases tends to be below 60 fps. This will correspondingly increase the end-to-end system latency.



with less delay than the IRIX SGI. Delays in serial port processing by UNIX systems have been observed before [Mine93].

Please see the Appendix A for the detailed experiment data and statistical analysis.

5. Conclusions

This paper has presented an end-to-end latency measurement method for projection based virtual reality systems. This method is very simple to implement and uses off-the-shelf hardware. The results of our analysis have helped us to make configuration decisions of our tracking systems. For instance, it shows that the tracker PC does not introduce extra latency, but reduces the system latency, and that there may be a variation of latency with direction of movement.

6. Future Work

The most labor-intensive part of this method is reading of time differences between the virtual cross and the physical sensor from videotapes. It took more than 10 hours to review the set of data in the experiment described in this paper. Also, human reading will introduce a subjective component. We plan to make the reading procedure automatic by using computer vision technology.

In the analysis example, we found a problem of direction asymmetry. We will continue to explore whether it is due to the IS-600 tracker.

After InterSense provides the software with prediction implemented, we will redo the test to evaluate the effect of prediction.

Acknowledgments

The virtual reality research, collaborations, and outreach programs at the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago are made possible by major funding from the National Science Foundation (NSF), awards EIA-9802090, EIA-9871058, ANI-9980480, ANI-9730202, and ACI-9418068, as well as NSF Partnerships for Advanced Computational Infrastructure (PACI) cooperative agreement ACI-9619019 to the National Computational Science Alliance. EVL also receives major funding from the US Department of Energy (DOE), awards 99ER25388 and 99ER25405, as well as support from the DOE's Accelerated Strategic Computing Initiative (ASCI) Data and Visualization Corridor program. In addition, EVL receives funding from Pacific Interface on behalf of NTT Optical Network Systems Laboratory in Japan.

References

[Adelstein96] Bernard D. Adelstein, Eric R. Johston, Stephen R. Ellis. Dynamic Response of Electromagnetic Spatial Displacement Trackers. *Presence*, Vol. 5, No. 3, 1996. pp.302-318.

[Azuma95] Ronald T. Azuma. *Predictive Tracking for Augmented Reality*. Doctoral Thesis, University of North Carolina, February 1995.

[Bryson90] Steve Bryson, Scott S. Fisher. Defining, Modeling, and Measuring System Lag in Virtual Environments. *Stereoscopic Displays and Applications I, Proceedings SPIE 1256* pp. 98-109. 1990.

[Cruz93] Carolina Cruz-Neira, Daniel J. Sandin and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. *Proceedings of the 20th annual conference on Computer graphics (SIGGRAPH '93)* pp.135-142. 1993.

[Dudewicz88] Edward J. Dudewicz. Modern Mathematical Statistics. John Wiley & Sons. pp.716-721. 1988.

[Ellis99] S.R. Ellis, B.D. Adelstein, S. Baumeler, G.J. Jense, R.H. Jacoby. Sensor Spatial Distortion, Visual Latency, and Update Rate Effects on 3D Tracking in Virtual Environment. *Proceedings IEEE Virtual Reality (IEEE VR'99) Conference*, pp.218-221, Houston, TX, March 1999.

[Foxlin98] Eric Foxlin, Michael Harrington and Yury Altsuler. Miniature 6-DOF Inertial System for Tracking HMDs. *SPIE vol. 3362 Helmet and Head-Mounted Display III*, Orlando, FL, April 1998.

[InterSense98] InterSense Inc. *IS-600 Series Precision Motion Tracker User Manual*. 1998.

[Liang91] Jiandong Liang, Chris Shaw, Mark Green. On Temporal-Spatial Realism in the Virtual Reality Environment. *Proceedings of the fourth annual symposium on user interface software and technology* pp.19-25. 1991.

[Mine93] Mark R. Mine. Characterization of end-to-end delays in head-mounted display systems. *Technical Report TR93-001*. Department of Computer Science, University of North Carolina at Chapel Hill.

Appendix A. An Example of Video-Based Measurement

In this experiment, we are mainly considering three factors that influence the delay in an IS-600 based tracking system: prediction, moving direction, connection type.

The following are tests we ran with different settings. Because there are several factors that influence the result of our test, we use the "two factors with replication ANOVA (Analysis of Variance)" method to analyze the data [Dudewicz88]. The tables list latency measurements as average numbers of video fields. For clarity, the number of data points represented in the tables has been reduced.

Test 1. Prediction

In this test, we considered 3 different settings: 0 ms, 25 ms, 50 ms. The connection method used was "with PC". The serial baud rate was 115200 bps. We used different prediction values as different treatments, and used different moving directions as different blocks.

	Pred. 0 ms	**Pred. 25 ms**	**Pred. 50 ms**
Up	1	1	1.4
	1	1	1.4
	1.5	1	1.5
	1.7	1	1
Down	3	1.9	2.5
	2.3	2	2.5
	2	2.2	1.6
	2	2.1	1.5
Left	1.5	2	1.9
	1.3	1.4	1.5
	1.1	1.8	1
	1.2	1.7	1.4
Right	1.9	2	2.5
	1.4	2	1.7
	1.9	2	2.1
	2.1	2.1	2.6

We define the null hypothesis H_0 as: there is no difference between different prediction settings. The F-test gives a p-value of 0.78601, which means, if we state that there is difference between different prediction settings, the error probability will be about 78.6%, making the hypothesis unacceptable. Therefore we conclude that the prediction does not influence the delay.

Test 2. Moving Direction

In this test, we considered 4 choices: up, down, left, right. The connection method used was "with PC". The serial baud rate was 115200bps. We used different moving directions as different treatments, and used different prediction values as different blocks.

	Up	**Down**	**Left**	**Right**
Pred. 0 ms	1	3	1.5	1.9
	1	2.3	1.3	1.4
	1.5	2	1.1	1.9
	1.7	2	1.2	2.1
Pred. 25 ms	1	1.9	2	2
	1	2	1.4	2
	1	2.2	1.8	2
	1	2.1	1.7	2.1
Pred. 50 ms	1.4	2.5	1.9	2.5
	1.4	2.5	1.5	1.7
	1.5	1.6	1	2.1
	1	1.5	1.4	2.6

We define the null hypothesis H_0 as: there is no difference between different moving directions. The F-test gives a p-value of 1.61E-08, which means, if we state that there is difference among different moving directions, the error probability will be very small. So we conclude that the moving direction does impact delay.

Test 3. "With PC" and "Without PC"

In this test, we considered two connection types: "with PC" and "without PC". We fixed the prediction value at 25 ms. The baud rate for both connection types was 115200 bps. We used different connection types as different treatments, and used different moving directions as different blocks.

	"Without PC"	"With PC"
Up	1.7	1.7
	2	1.7
	1.7	1.7
	2	1.5
	1.7	2.2
Down	1.8	1.5
	1.7	1.5
	1.5	1.5
	1.6	1.5
	1.5	1.5
Left	2.5	2
	2.5	1.5
	2	2.2
	3	2.3
	2.2	1.8
Right	2.5	1.8
	2.2	2
	2.2	1.8
	2.3	1.5
	2.6	1.9

We define the null hypothesis H_0 as: there is no difference between different connection types. The F-test gives a p-value of 0.000211, which means that we can state that there is a difference between different connection types.

The average delay without PC:
2.06 frames * 2 * 16.67 = 68.7 ms

The average delay with PC:
1.755 frames * 2 * 16.67 = 58.5 ms

According to the NTSC standard, the field time is 16.67 ms.