A SYSTEM TO MEASURE, CONTROL AND MINIMIZE END-TO-END HEAD TRACKING LATENCY IN IMMERSIVE SIMULATIONS

Giorgos Papadakis
Department of Electronic and Computer Engineering
Technical University of Crete
Chania, Crete, Greece
gpapadak@ced.tuc.gr

Katerina Mania
Department of Electronic and Computer Engineering
Technical University of Crete
Chania, Crete, Greece
k.mania@ced.tuc.gr

Eftichios Koutroulis
Department of Electronic and Computer Engineering
Technical University of Crete
Chania, Crete, Greece
efkout@electronics.tuc.gr

Abstract

System latency (time delay) and its visible consequences are fundamental Virtual Environment (VE) deficiencies that can hamper user perception and performance. In order to realize this goal, we present an immersive simulation system which improves upon current latency measurement and minimization techniques. Hardware used for latency measurements and minimization is assembled by low-cost and portable equipment, most of them commonly found in an academic facility without reduction in accuracy of measurements. We present a custom-made mechanism of measuring and minimizing end-to-end head tracking latency in an immersive VE. The mechanism is based on an oscilloscope comparing two signals. One is generated by the head-tracker movement and reported by a shaft encoder attached on a servo motor moving the tracker. The other is generated by the visual consequences of this movement in the VE and reported by a photodiode attached to the computer monitor. Visualization and application-level control of latency in the VE was implemented using the XVR platform. Minimization processes resulted in almost 50% reduction of initial measured latency. The description of the mechanism by which VE latency is measured and minimized will be essential to guide system countermeasures such as predictive compensation. The system presented in this paper will be used to investigate the effect of latency on spatial awareness states.

CR Categories: I.2.1 Virtual reality, I.2.1 Latency, I.2.1 [Tracking Systems, I.2.4 Presence and cognition;

Keywords: Latency measurement, minimization, tracking.

1. Introduction

End-to-end latency in a Virtual Environment (VE) is defined as the time lag between a user's action in the VE and the system's response to this action. VE lag comprises of four different types; user-input device lag, application-dependent processing lag, rendering lag and synchronization lag (Figure 1). The user input device lag is the lag introduced from the communication between the tracking system and the VE application. The application-dependent processing lag is the time required for the computation of the 3D model and depends on the complexity of the model and the application itself. The rendering lag is the time that passes until data sent from the VE application to the rendering hardware appears on a monitor or immersive display. The rendering lag depends on the scene and the viewpoint rendered at each time, so it varies through the VE application run-time. The synchronization lag is the total time that data is waiting during the necessary communication of involved input devices, in-between the parallel processing stages of the VE application. It is application-relevant and depends on rendering processing stages which are not well-synchronized to avoid delays of transmission. These stages are independent and it is possible that the input device deposits new tracking data shortly after the application reads the previous data. Thus, the application is busy processing the previous input before it reads and starts to process the new input, so that input data is delayed. Moreover, there is a fifth kind of lag, i.e. the frame-rate-induced lag, resulting from the fact that data displayed progressively become out of date, while the display is not updated fast. This type of lag is distinguished from other lag sources and is not considered as end-to-end latency. It is, though, perceivable by the users and results in slow frame rates while exposed to a VE to be considered unacceptable [Wloka 1995]. Excessive system latency is a well-known defect of VE and tele-operation systems [Ellis et al. 2004]. It is particularly troublesome for head-tracked systems since delays in head orientation measurement give rise to errors in presented visual direction.

Perceptible latency that is experienced by its visual consequences on a display is one of the most notable problems facing current VE applications [Ellis et al. 1999], [Garret et al. 2002], [Stanney et al. 1998]. High end-to-end latency can severely degrade users' performance in a VE [Ellis et al. 1997], [Ellis et al. 2002]. The RMS (Root Mean Square) tracking errors, which are an objective measure of user's performance, are caused mostly by visual latency, rather than spatial sensor distortion or low update rates [Ellis et al. 1999]. Latency also affects users' performance on 3D object placement tasks [Liu et al. 1993], [Watson et al. 1997]. Latency in a VE can also cause lack of accuracy during tracking tasks, motion sickness, and loss of immersion on users, as well as disorientation, discomfort, and even nausea [Kennedy et al. 1992], [Stanney et al. 1998]. While users can exhibit sensorimotor adaptation that might improve manual performance when time delays exist in situations where task preview is available [Cunningham et al. 2001a], [Cunningham et al. 2001b], the presence of delay has been shown to hinder operator adaptation to other display distortions such as static displacement offset [Held et al. 1966].

More recently interest has been directed towards the subjective impact of latency on the users' reported sense of presence. Latency, as well as update rate, is considered as a factor affecting the operator's sense of presence in the environment [Welch et al. 1996], [Uno, Slater 1997]. Lower latencies were associated with a higher self-reported sense of presence and a statistically higher change in heart rate for users, while exposed to a stress-inducing (fear of heights), photorealistically rendered VE, involving walking around a narrow pit [Meehan et al. 2003]. The role of VE scene content and resultant relative object motion on latency detection has been examined by presenting observers in a head-tracked, stereoscopic head mounted display with environments having differing levels of complexity ranging from simple geometrical objects to a radiosity-rendered scene representing a hypothetical real-world setting [Mania et al. 2004b]. Such knowledge will help understand latency perception mechanisms and, in turn, guide VE designers in the development of latency countermeasures. In this study, a radiosity-rendered scene of two interconnected rooms was employed. Latency discrimination observed was compared with a previous study in which only simple geometrical objects, without radiosity rendering or a 'real-world' setting, were used employing formal psychophysical techniques which are far-removed from simulated tasks. The user is instructed to report the consequences of latency focused on differences between paired stimuli of varied tracking latency. They reveal that the Just Noticeable Difference (JND) for latency discrimination by trained observers averages ~15 ms or less, independent of scene complexity and real-world meaning. Such studies were, though, far-removed from real application scenarios of interaction with synthetic scenes or remote tele-manipulation applications because the user is required to solely focus on identifying the visual or other consequences of latency while no other task is performed. Moreover, there is always an issue that due to the intense nature of psychophysical experimentation, a small amount of users are normally tested, resulting in doubts concerning the generality of such results.

The work presented in this paper aims to implement an immersive simulation of minimum latency as well as a latency measurement mechanism. We extend past latency measurement and minimization techniques to build a portable, low-cost, custom-made but also accurate latency measurement system and, ultimately, create a VE with minimal head tracking latency. In the future, this system will be used in order to conduct experiments exploring the cognitive impact of latency.

The paper is organized as follows. Section 2 of the paper analyzes background research related to techniques for latency measurement and minimization. Section 3 presents a novel low-cost, custom-made portable latency measuring mechanism. Section 4 presents the resulting processes to minimize as well as add latency at will for experimental purposes. Section 5 presents future work on the effect of latency on 3D spatial awareness.

2. Background

2.1 Latency Measurements

In order to examine the effects of latency in a VE it is necessary that end-to-end latency is effectively measured minimized and controlled. Several measurement techniques have been introduced through years. In an early measurement method [Liang et al. 1991], an electromagnetic tracker was attached to a moving pendulum. Tracker readings were time-stamped and stored in a host computer. The computer monitor was displaying the current time of the clock that was generating the time stamps. A video camera was simultaneously recording the monitor display and the swing of the pendulum. The video was later analyzed frame-by-frame. End-to-end latency was determined by comparing the display time when the pendulum was passing by the zero-crossing point and the time stamp stored on the host computer. Later, the use of an oscilloscope instead of a video camera was introduced [Mine 1993], [Jacoby et al. 1996]. The oscilloscope was used to compare three inputs, estimating the end-to-end latency. The first input was deducted from a LED-photodiode pair that was marking the zero-crossing point of the pendulum. The second input was deducted from a Digital-to-Analog (D/A) converter attached to the host computer, reporting tracker position readings. Comparing these two inputs determined the input device lag. Additionally, a third photodiode was monitoring brightness changes on the system's display, while a specific polygon displayed was changing color from white to black and vice-versa at the time that zero crossings were reported to the system. Comparison between the first and the third input was used to measure the overall end-to end latency.

A slightly modified technique was used resulting in higher accuracy [Jacoby et al. 1996]. Instead of the first LED-photodiode pair responsible for monitoring and reporting the motion of the pendulum, a swing arm motor equipped with a shaft encoder was used. The arm repeatedly moved the tracker back-and-forth through a pre-set threshold point and the encoder reported crosses of the threshold to an oscilloscope. This input was, at first, compared with the input deducted from a photodiode monitoring the VE system's screen, which was displaying the same rectangular color transition as in [Mine 1993]. A calibration technique using a modified phonograph turntable to provide easily tracked periodic motion, reminiscent of the pendulum-based calibration technique of [Liang et al. 1991] was used by [Swindells et al. 2000]. Subseqneutly, the previous techniques were modified by directly monitoring the RGB analog output signals of the VGA, instead of using a photodiode in order to monitor the display [Hill et al. 2004]. Critical portions of the VE application code were also "trapped", thus, producing timestamps and signals to the oscilloscope. These signals were used to bridge internal and external measurements and provide information about timing at different stages of the VE execution [Hill et al. 2004].

Taking into account the RGB signal instead of the photodiode readings of the monitor does not fairly offer an accurate measurement of the end-to-end latency, as it does not correspond to what the user actually sees. Relevant research attempts to minimize latency by assessing its level relevant to the internal system components, and reorganizing the communication between them more efficiently.

Recent estimating methods make use of video analysis comparing movement of the head tracker and the resulting movement of a simulated image on a screen. The simultaneous movements are captured using a video camera [Steed 2008] or encoded by photodiode readings of luminance gradients (one gradient that the tracked object is moved across and another gradient that is produced by the VE) [Di Luca 2010]. However, such estimation methods result in potentially less accurate measurements than using oscilloscope readings of electronic signals from the VE host computer and, therefore, cannot be further expanded in order to provide information concerning latency increases throughout the processing of the interactive VE itself. This information is essential in order to be able to understand how these different stages contribute to the overall latency and thus, be able to reorganize these components in order to achieve minimal latency.

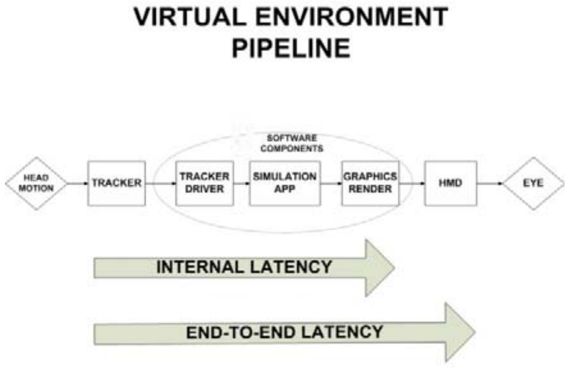

Figure 1. *The VE visual pipeline.*

2.2. Latency Minimization

In a VE, the expected minimum achievable latency is the sum of lag contributed from each of the three pipeline components, i.e. the tracking system component, the application-dependent processing component and the rendering component [Wloka 1995] (Figure 1). According to previous research on latency minimization, the initial latency measured (45 ± 1.8 ms) was by far higher than the minimum predicted by adding known latencies of the system components [Hill et al. 2004]. The additional lag was found to be caused by the synchronization lag and lies between the three other subsystems. Part of this lag lies between the graphics application and the display rendering subsystem and is mostly caused by the synchronization that occurs between the buffer swapping and the v-sync signal [Hill et al. 2004]. The rendering subsystem video memory (VRAM) consists of two buffers, the back buffer where changes are made to the image and the front buffer from which the image is displayed at the screen. Moreover, most of the modern graphic cards, in fact, triple-buffered the image, adding an extra frame of delay to the VE system. Though this triple buffering may have been useful for high performance 180 Hz CRT displays, this was unnecessary for the 60Hz Head Mounted Displays (HMDs) used in previous work [Hill et al. 2004], as well as, for the NVIS nVISOR SX111 to be used in future work. Triple buffering and vertical-sync can be disabled though, by turning off the proper settings in the graphics card's control panel. To prevent image tearing, the v-sync signal may be reused, without restricting the buffers swap rate, ensuring that swapping is regulated in order that only one swapping occurs at each v-sync. Reuse of the v-sync signal is achieved by connecting the V-sync signal from the graphics card VGA output to the computer's parallel port of the computer and having the VE application poll the port using the UserPort kernel driver [hill et al. 2004], [UserPort].

Lack of synchronization also occurs between the application software and the tracking device. This asynchrony consists of two components, one that varies at each update and one that remains constant. The varying component, called dynamic asynchrony, results from the absence of synchronization between the tracker device readings and the updates of the graphics application. The tracker sampling sequence is usually not identical to the graphics v-sync rate. This asynchrony has been eliminated [Hill et al. 2004] by synchronizing the tracker readings with the v-sync signal, after doubling the signal frequency (60 Hz) in order to match the one of the tracker (Polhemus Fastrack, 120Hz). The constant component, called static asynchrony, is the time that passes when the data processing is completed, however, at the same time the data remain idle waiting for the next monitor update cycle. Avoiding this "ageing" of processed data, implies receiving data from the tracker the last possible instant, necessary for completing all the computation needed to display the next frame in time. Since the update rate of the screen in [Hill et al. 2004] was 60Hz and the tracker update rate was 120 Hz, at each update cycle of the graphics subsystem, the tracker reports data two times. Software and hardware modifications can be used in order that the graphics subsystem skips the first reading of the tracker and uses the "fresher" second one, such as internally clocked "sleep" functions of the graphics application. This modification resulted in the elimination of up to 8.3 ms of added latency (half of the screen update cycle) [Hill et al. 2004]. The final modified VE resulted in a constant latency of 8.5ms for a simple ~100 polygon test environment and 13 ms for a more realistic ~35k test environment, without taking into account the refresh (frame) rate latency. An important fact is that both of these measurements were in the 8-20 ms range of perceptual tolerance for latency in a head tracked HMD-based VE system, however, both scenes were of fairly low polygon count [Mania et al. 2004a], [Adelstein et al. 2003], [Ellis et al. 2004].

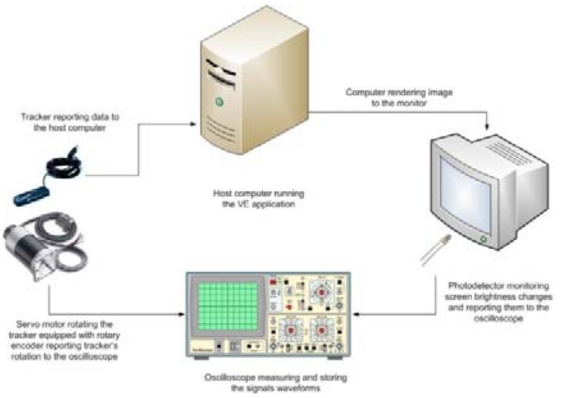

Figure 2. *The proposed data-acquisition system for measuring the end-to-end latency.*

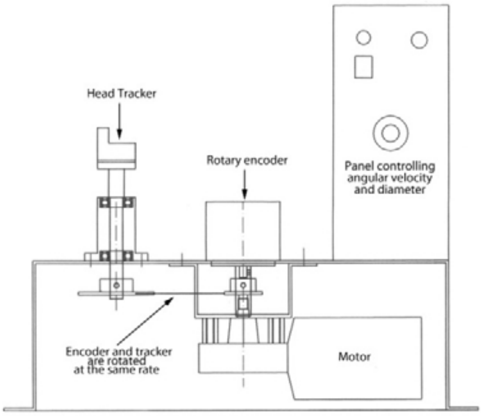

Figure 3. *The diagram of the servo-mechanism which is used to control the tracker rotation.*

3. A System for Latency Measurement

The VE visual pipeline (Figure 1) includes all processing steps starting from the head tracker input up to the display of the VE scene on the output device. The tracker acquires the current position and orientation information of the head. This data is deposited, through the tracker driver into shared memory, making data available to the next component in the pipeline, e.g. the simulation application. The simulation application retrieves this data from the shared memory and performs all the application related calculations that may be related to the physical simulation or the graphical user interface of the application. These calculations may be viewpoint dependent and may impact the final viewable image. All calculations that do not rely on the user's viewing position, or other sensed input action can be pre-computed externally of the VE visual pipeline and therefore do not contribute to the overall end-to-end latency.

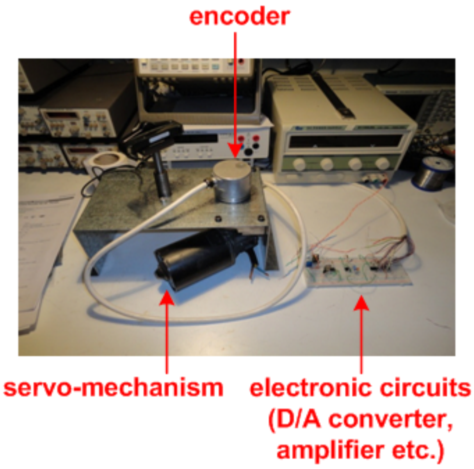

Figure 4. *The laboratory-built prototype comprising the tracker rotation mechanism with the shaft encoder attached and the signal-conditioning electronic circuits.*

After the completion of these calculations the geometry of the scene is transferred to the Graphics Processor Unit (GPU) of the graphics hardware. In the case of stereoscopic visualization, viewpoint transformations specific to each channel of the stereo viewpoint are performed. The resulting image (or images in case of stereo viewpoints) is rasterized into pixels and drawn into a temporary buffer of the graphics hardware video ram (VRAM), e.g. the back video buffer. After the pixels are drawn, the back buffer is swapped to the front video buffer of the VRAM. The use of the back buffer at the VRAM serves as a staging area where images are assembled before scanned onto the display in order to avoid visual discontinuities and artifacts that would otherwise occur if parts of the front buffer were changed during the scan-out process. After the image is swapped to the front buffer, pixels are scanned out onto the display in rows from top of the screen to down and from left to right within each row. Thus added latency at this final step varies, being lower at the upper left portion of the screen and higher at the lower right portion. In order to simplify the description of contributed latency, here we refer only to the latency at the uppermost left portion of the screen. At each refresh cycle of the screen a pulse called V-sync is generated. Synchronization between buffer swapping and the v-sync signal prevents “image tearing” by timing video time swaps to match the vertical blank interval of the monitor. The v-sync pulse indicates when the front buffer is ready to accept the next video frame, and only then the back buffer is being swapped. When synchronization between the back/front buffer swap and the v-sync pulse is used the maximum frame rate of the graphics redraw is limited by the interval between successive v-syncs, no matter how quickly video buffers can be filled with newer image information.

The latency of the VE system is the sum of the completion times required for each of the consecutive processes in the pipeline to be processed. In order to measure end-to-end latency, we have to also take into account the time that pixels take to be scanned onto the display; this is dependent on the hardware of the display. Thus, the term ‘internal latency’ indicates the latency contributed from all the processes of the VE visual pipeline except from the scan-out process.

In the proposed system, the end-to-end-latency is measured using a variation of the above techniques (Figure 2), designed to accurately measure latency of the InsertiaCube3 head tracker [InsertiaCube3], by utilizing a low-cost, custom-made portable measuring mechanism with relative angular resolution of 0.03° and internal latency of 2ms. Previous methods used a pendulum which was moving a 6 DOF (Degrees of Freedom) positional tracker about the 3-rotational axes (i.e. roll, pitch and yaw) and along the 3 positional axes (i.e. x, y and z). The orbit of the tracker movement was forming an arc and a photodiode or encoder was reporting crossings through a point. In the proposed system, the tracker is capable only to perform rotational 3-DOF movement tracking resulting in higher accuracy because of translational error control. In particular, the tracker movement at the measurements is restricted to rotational only, on one axis. The proposed data-acquisition system for measuring the end-to-end latency is illustrated in Figure 4. A custom made modular servo-mechanism, depicted in Figure 3, with a 14-bit, parallel-output digital rotary encoder attached to its shaft rotates the tracker back-and-forth within a preset threshold angle. The 14-bit resolution of the encoder matches the angular resolution of the tracker used in this study. The angular velocity and arc of the movement are fully controllable through a power supply and a double-pole/double-throw switch. The encoder output signals are interfaced to a D/A converter and then passed to the oscilloscope. The XVR VE application [XVR 2008] is configured such that passing through a threshold angle results in VE changes. Both the tracker and the VE application are zero-calibrated prior to the measurements.

The original scene is photorealistically illuminated using pre-computed radiosity textures and stereoscopically rendered, using XVR’s side-by-side stereoscopic rendering feature. The VE represents a room as shown in Figure . The polygon count of the scene was ~140,000 polygons. A box is superimposed at every frame on the uppermost left corner of the screen (Figure 5). The application is configured to change the color of the box from black to white and vice versa at each threshold crossing of the tracker. A photodiode with spectral sensitivity in the visible light is attached to the front of the monitor and it is used to measure the brightness changes of the superimposed box. Rather than using the Head Mounted Display (HMD) system to be utilized for future experimental work, we used a standard LCD monitor configured to refresh at 60Hz similar to the NVIS nVISOR SX111 that will be used in subsequent experiments. The small dimensions of the HMD displays make it hard to attach a photodiode on it. The refresh rate of both LCD and HMD displays is similar (60 Hz). Each monitor is configured to display the VE at 1280*1024 pixels resolution matching the resolution of the HMD displays. The photodiode output signal is amplified using an operational amplifier-based current-to-voltage converter. The laboratory-built prototype comprising the tracker rotation mechanism with the shaft encoder attached and the signal-conditioning electronic circuits is depicted in Figure 4. A digital oscilloscope with waveform storage capability is used to measure and store parallel digital samples of the D/A converter and amplifier output signals, corresponding to the tracker position and the display brightness level, respectively. An example of these signals is illustrated in Fig. 6.

The oscilloscope used in the experimental setup is configured to acquire 10.000 samples at a time frame of 1 second (i.e. one sample per 1/10 of a millisecond).

The samples acquired by the oscilloscope are downloaded to a PC through a USB communication interface. Implemented software compares the individual values of the signals measured, in order to calculate the time-shift between the passing of the tracker through the threshold angle and the black-to-white transition of the polygons. This time-shift is equal to the end-to-end latency of the system. Estimates (mean ± standard deviation) of the VE latency were derived from averaging measurements of a hundred back-and-forth threshold crossings by our rotation mechanism; fifty of them when moving the tracker from right to left and fifty vice-versa. The estimated latency of our system was measured to be 90ms ±10% before minimization processes were applied as described below, inclusive of the latency induced by the refresh rate of the screen (Figure 6).

4. Minimizing and Adding Latency

In our system, certain latency-inducing features of the graphics subsystem were disabled. Triple buffering and V-sync were disabled from our graphics card control panel. Disabling these features resulted in the reduction of the overall latency dynamic asynchrony component, but introduced image tearing. Although image tearing may not be observable [Meehan et al. 2003], the XVR platform [XVR 2008] was configured to produce the higher frame rate possible approximately 160hz, independent from the monitor's maximum frame rate. Moreover the VE software was configured to retrieve tracker data directly from the tracker interface .dll at the last possible instant necessary for completing its computations. This eliminated the static asynchrony latency component described above.

After disabling these features, the end-to-end latency of our system was measured again using the same measurement technique described above. The new estimated latency of our VE was measured to be slightly below 50ms ±10%, a reduction of almost 50%, inclusive of the latency induced by the refresh rate of the screen. The estimated latency and the latency reduction is less or comparable to previous work, in this case using a more complex environment of high polygon count, accurate measurements via an oscilloscope and a custom-made, low-cost, portable system.

The ability to add a constant amount of latency in order to conduct future experiments of variable latency conditions was added to the system using a circular buffer inside the VE application for storing tracker positions and reporting them to the rendering thread on a later frame (Figure 7). This addition does not affect frame or tracking rate. Added latency is needed by subsequent experiments which require the addition of specific amounts of head tracking latency in order to assess their effect on spatial cognition.

5. Conclusions and Future Work

In this paper we presented a custom-made mechanism of measuring and minimizing end-to-end head tracking latency in an immersive VE. Our mechanism builds on previous mechanisms by using an oscilloscope to compare two signals, assembled by low-cost, custom-made and portable equipment. One signal is generated by the head-tracker movement and reported by a shaft encoder attached on a servo motor moving the tracker. The other signal is generated by the visual consequences of this movement in the VE and reported by a photodiode attached to the computer monitor. The end-to-end head tracking latency of the VE is the measured time-shift between these two signals. The presented system calculates this time-shift by off-line processing the tracker position and display brightness measurements stored in a computer t derived from the oscilloscope using a USB connection. Thus, an accurate measuring mechanism is provided, utilizing equipment commonly found in an academic facility. Subsequent software reorganizations to the VE system result in the reduction of the overall system latency resulting to a VE with minimal end-to-end head-tracking latency. The utility of simulation environments for training, such as flight simulators, or collaborative 3D design, as well as remote tele-operation manipulations is predicated upon the accuracy of the 3D spatial representation formed mentally. Spatial awareness is essential for human performance efficiency of tasks requiring spatial knowledge of an environment [Mania et al. 2010]. A central research issue therefore for such simulations is how participants cognitively represent 3D spatial elements and how their memory and recognition of such worlds corresponds to real world conditions.

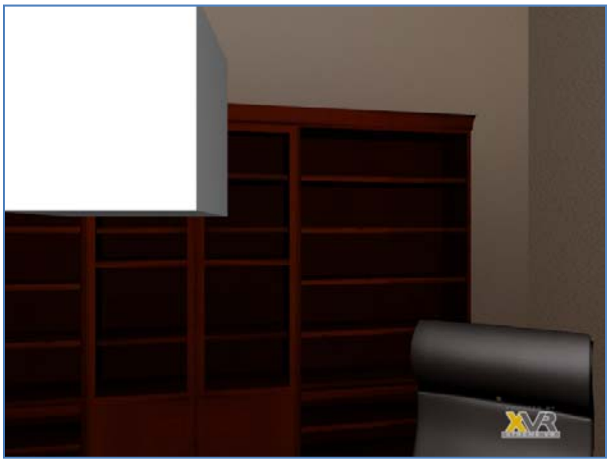

Figure 5. *Test scene for latency measurements.*

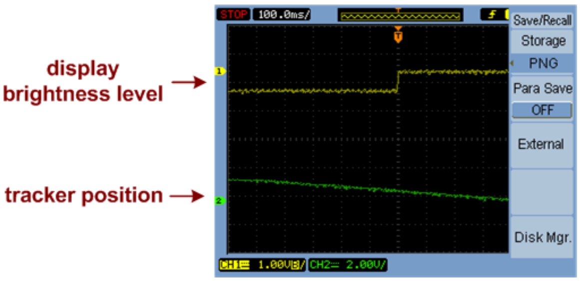

Figure 6. *An example of the signals measured using the oscilloscope, corresponding to the tracker position and the display brightness level, respectively.*

The system presented in this paper will be used to investigate the effect of latency levels, ranging from the minimum system latency to added latency, on spatial awareness states.

The main premise of future work is that accuracy of memory performance per se is an imperfect reflection of the cognitive activity that underlies user performance. Memory, in the sense of 'information' for subsequent analysis, plays an important role in perceptual systems such as the visual, auditory, haptic and kinesthetic systems[Mania et al. 2003], [Mania et al. 2006]. Memory research has established that accurate memory recollections can be linked with the subjective awareness states "Remember", which is a recollection based on a mental image or a prior experience, and "Know", which is a general sense of knowing with no or little recollection of this sense [Mania et al. 2010].

Furthermore if something feels like it has been encountered or experienced recently, but nothing about this occurrence can be remembered is defined as "Familiar". On the other hand, there is the probability of a random "Guess".

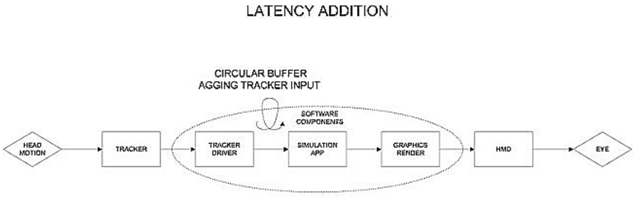

Figure 7. *Increasing end-to-end latency by aging tracker input using a circular buffer.*

Memory research has established the dissociation between 'remembering' and 'knowing' as an important means of differentiation in the quality of the experience participants have following recognition memory providing a demonstration that these responses can be made in a memory test.

Desirable variations of awareness states for specific application purposes, therefore, could be ultimately identified and generalized. It could be true, for instance, that for flight simulation applications it is crucial for trainees to achieve a high level of conscious recollections associated with mental images relating to instruments as opposed to recollections that are confident but not accompanied by the recall of visual images or memories of events. Psychology of memory should take on board subjective reports of conscious states and not just rely on more conventional measures of performance. This evidence has established that the essential subjectivity of remembering and knowing does not make reports of these states of awareness intractable to science.

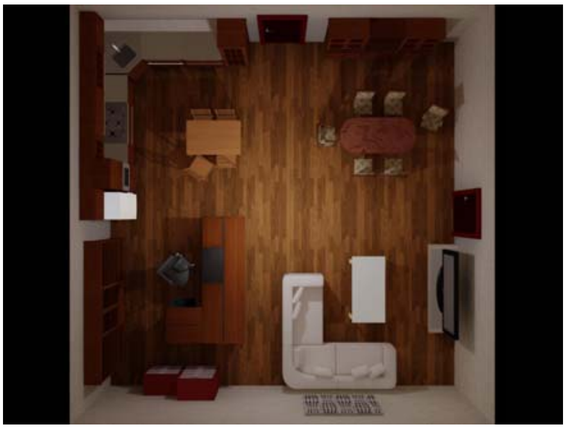

Figure 8. *The experimental scene (top-view).*

In subsequent experiments, participants, across three levels of head tracking latency and two types of objects placed in a photorealistically rendered scene, will be exposed to the VE and complete an object-based memory recognition task reporting associated states of awareness. Such considerations could also relate to human attentional resources and could ultimately lead to algorithmic optimizations related to computational savings of rendering, as well as indications of perceptual sensitivity to latency.

References

Adelstein, B. D., Johnston, E. R., Ellis, S. R. 1996. Dynamic response of electromagnetic spatial displacement trackers. Presence: Teleoperators and Virtual Environments, 5(3), 302-318.

Adelstein B.D., Lee T. G., Ellis S.R. 2003. Head tracking latency in virtual environments: psychophysics and a model. Proceedings, 47th Annual Meeting of the Human Factors and Ergonomics Society, pp. 2083-2087.

Cunningham D.W., Billock V.A., Tsou B.H. 2001 Sensorimotor adaptation to violations in temporal contiguity. Psychological Science, 12, 6, 532-535.

Cunningham D.W., Chatziastros A., von der Heyde M., Bulthoff H.H. 2001 Driving in the future: Temporal visuomotor adaptation and generalization. Journal of Vision, 1, 88-98.

Di Luca, M. 2010. New Method to measure end-to-end delay of virtual reality. Presence, Vol 19, No. 6, 569-584.

Ellis S.R., Adelstein B.D., Baumler S., Jense G.J., Jacoby R.H. 1999. Sensor spatial distortion, visual latency, and update rate effects on 3D tracking in virtual environments. In Proceedings of VR '99, pp. 218-221.

Ellis S.R., Bréant K., Menges B.M., Adelstein B.D. 1997. Operator interaction with virtual objects: effects of system latency. In Proc. HCI International, 973-976.

Ellis S.R., Mania K., Adelstein B.D., Hill M. 2004. Generalizability of latency detection in a variety of virtual environments. 48th Annual Meeting Human Factors and Ergonomics Society, New Orleans.

Ellis S.R., Wolfram A., Adelstein B.D. 2002. Large amplitude three-dimensional tracking in augmented environments: a human performance trade-off between system latency and update rate. In Proc. of the 46th Annual Meeting Human Factors and Ergonomics Society, 2149-2154.

Garret A., Aguilar M., Barniv Y. 2002. A recurrent neural network approach to virtual environment latency reduction. In Proceedings, IJCNN, Honolulu, HI.

Held R., Efstathiou A., Greene M. 1966. Adaptation to displaced and delayed visual feedback from the hand. Journal of Experimental Psychology 72, 6, 887-891.

Hill M., Adelstein B.D., Ellis S.R. 2004. Achieving minimum latency in virtual environment applications. In Proc. of IMAGE Society Annual Conference, Scottsdale AZ.

InsertiaCube3: INTERSENSE www.isense.com http://www.intersense.com/pages/18/11/

Jacoby R.H., Adelstein B.D., Ellis S.R. 1996. Improved temporal response in virtual environments through system hardware and software reorganization. Proceedings, SPIE Conference on Stereoscopic Displays and Applications VII, Vol. 2653, pp. 271-284.

Kennedy R.S., Lane N.E., Lilienthal M.G., Berbaum K. S., Hettinger L.J. 1992. Profile analysis of simulator sickness symptoms: application to virtual environment systems. Presence, 1 (3), 295-301.

Liang J., Shaw C., Green M. 1991. On temporal-spatial realism in the virtual reality environment. In proceedings of the 4th annual ACM symposium on User interface software and technology (pp. 19-25). New York: Association for Computer and Machinery.

Liu A., Tharp G., Lai S., French L., Stark L.W. 1993. Some of what one needs to know about using head-mounted displays to improve teleoperator performance. IEEE Transactions on Robotics and Automation 9, 5, 638-48.

Mania K., Adelstein B.D., Ellis S.R., Hill M. 2004. Generalizability of latency detection in a variety of virtual environments. Submitted to First Symposium on Applied Perception in Graphics and Visualization, ACM-SIGGRAPH, Los Angeles, August.

Mania K., Adelstein B.D., Ellis S.R., Hill M. 2004. Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity. In Proc. ACM Siggraph Symposium on Applied Perception in Graphics and Visualization, 39-47, ACM Press.

Mania K., Badariah S., Coxon M. 2010. Cognitive transfer of training from immersive virtual environments to reality. ACM Transactions on Applied Perception, ACM Press, 7(2), 9:1-9:14, ACM Press.

Mania K., Troscianko T., Hawkes R., A. Chalmers 2003. Fidelity metrics for virtual environment simulations based on spatial memory awareness states. Presence, Teleoperators and Virtual Environments, 12(3), 296-310. MIT Press.

Mania K., Wooldridge D., Coxon M., Robinson A. 2006. The effect of visual and interaction fidelity on spatial cognition in immersive virtual environments. IEEE Transactions on Visualization and Computer Graphics journal, 12(3): 396-404.

Meehan M., Razzaque S., Whitton M.C., Brooks Jr F.P. 2003. Effect of latency on presence in stressful virtual environments. Proceedings, IEEE Virtual Reality Conference, Los Angeles, pp. 141-148.

Miné M.R. 1993. Characterization of end-to-end delays in head-mounted display systems. Chapel Hill, NC, University of North Carolina.

Stanney K.M., Mourant R.R., Kennedy R.S. 1998. Human factors issues in virtual environments: A review of the literature. Presence, Vol. 7:4, pp. 327-351.

Steed A. 2008. A simple method for estimating the latency of interactive, real-time graphics simulations. VRST '08 Proceedings of the 2008 ACM symposium on Virtual reality software and technology. ACM New York, NY, USA.

Swindells, C., Dill J.C., Booth, K. 2000. System lag tests for augmented and virtual environments, Proceedings of the 13th Annual Symposium on UIST, ACM.

Uno M., Slater M. 1997. The sensitivity of presence to collision response. In Proc. of IEEE VRAIS, 95-101, 1997.

UserPort. http://hem.passagen.se/tomasf/UserPort/

Watson B., Walker N., Woytiuk P., Ribarsky W. 2003. Maintaining usability during 3D placement despite delay. In Proc. of IEEE Virtual Reality 2003, 133-140.

Welch R.B., Blackmon T.T., Liu A., Mellers B.A., Stark L.W. 1996. The effects of pictorial realism, delay of visual feedback and observer interactivity on the subjective sense of presence. Presence: Teleoperators and Virtual Environments 5, 3, 263-273.

Wloka M. 1995. Lag in Multiprocessor Virtual Reality. Presence, Vol. 4:1, pp. 50-63.

XVR 2008. VR Media, http://www.vrmedia.it