Dynamic Blanking for Virtual Reality Image Displacement

Anthony M. Pelot*

Stanford EE 267, Virtual Reality, Course Report, Instructors: Gordon Wetzstein and Robert Konrad

Abstract

In this paper we will describe a method for introducing two dimensional image displacement in a virtual reality head mounted display. This displacement can be used to correct the displayed image to account for head movements made since the orientation data was initially obtained prior to rendering the displayed frame. This method is novel in that it can be implemented extremely late in the display pipeline, on the head mounted display (HMD), rather than requiring implementation in the rendering system driving the display. This allows for the most up to date orientation data to be used and reduces or augments the requirement of extended prediction methodologies to estimate the orientation of the head at the time the frame is displayed.

Keywords: virtual reality, timewarp, image displacement, motion to photon latency, hdmi, blanking

Concepts: •**Computing methodologies** → **Virtual reality;** *Perception;*

1 Introduction

Generating a display image that corresponds with head rotation is a key fixture of virtual and augmented reality. One of the most important metrics when designing these systems is the motion to photon latency. Any movement of the head will result in a corresponding change in the perspective that must be displayed for the user, but processing that movement and generating a new perspective can introduce a significant amount of latency between the actual orientation of the head and the image the user perceives. Minimizing this latency is important for both maintaining presence in the virtual environment [Held and Durlach 1992], as well as preventing fatigue and discomfort due to conflicts between vestibular and visual stimulus [Akiduki et al. 2003].

Motion to photon latency has three primary factors. The first is the latency of the inertial measurement unit (IMU) in reading sensor data and fusing it into an orientation in space [LaValle et al. 2014]. The second source of latency is the rendering pipeline executed by the GPU to determine the appropriate perspective and view of the 3D world and generate a frame of appropriate pixel data. The final factor in latency is the time that the data is displayed on the screen.

This final factor has been addressed by the introduction of low persistence displays [Zielinski et al. 2015]. This effectively turns off the pixels at a much earlier time, reducing motion to photon latency from the other direction by not displaying the image for as long on the display.

This project intends to minimize the remaining latency by bridging the gap from IMU data observation to pixel display on the screen without requiring any changes to the rendering process.

We will be analyzing the viability of two dimensional image displacement at a very late stage in the display pipeline to correct for interim head movements by manipulating the HDMI signal as it is transferred to the display driver. By manipulating the HDMI timings with a dynamic blanking signal and utilizing larger blanking

---

intervals, we can create a buffer that allows for warping the output frame to more accurately align with the current head orientation.

By buffering the incoming frame and generating new timings at a significantly higher pixel clock rate, we can output the active pixel data for the frame at a later time and with a small amount of shifting determined by orientation data from an inertial measurement unit that can be read after the start of transmission of frame data by the GPU.

Because this method only provides 2D displacement, parallax effects due to head movement are not accurately represented. In addition, because no additional rendering of out of view elements has been done, shifting the image will leave a blank area with no rendered data on one edge of the screen. For these reasons, orientation correction using this method should be limited to very small corrections. Since this method does not interfere with the rendering pipeline and takes place on the HMD itself, it can be easily combined with other existing methods of timewarp to improve orientation correction.

2 Related Work

The problem of motion to photon latency in VR/AR displays has been the subject of research for many years. Original implementations of an image displacement system were mechanical in nature and used on the HMD units for aircraft pilots [Burbidge and Murray 1989; So and Griffint 1992].

With the advent of digital display systems, an interest in digital VR/AR environments spurred additional research such as that done by Mazuryk and Gervautz [Mazuryk and Gervautz 1995] in which they implement both a prediction system based on previous movements as well as a simple 2D image displacement in the frame buffer based on the predicted motion. The method of displacement used is very similar to the method we are proposing, but takes place at the end of the rendering phase. Rendering a slightly larger image than needed was used as a solution to the problem of unrendered areas on one edge of the screen and could be easily incorporated in the system we are proposing if desired.

More advanced 3D image displacement has also been investigated. Research at the University of North Carolina in Chapel Hill [Mark et al. 1997] rendered low frequency reference frames in a predictive manner and then used a separate combination algorithm to derive the current frame from two reference frames. This method benefits from less frequent rendering of scenes, but still replaces the rendering latency with the latency of the combination algorithm.

The most recent method of image displacement is currently implemented in the Oculus Rift. Using a process they refer to as late orientation warping [Binstock 2015], the Rift delays obtaining orientation data until the last possible moment in the rendering pipeline. This bypasses much of the latency involved with rendering the scene and allows fresh IMU data to be used, but still only uses IMU data available 3ms before scanout of the frame.

Oculus has also implemented a feature they term Asynchronous Timewarp [Antonov 2015]. This method runs a separate thread in parallel to the rendering engine and grabs the most recently rendered frame just before each VSYNC and displaces the image. The

primary goal of this method is to fill dropped frames, so that if rendering for a particular frame takes longer than the frame length, the previous frame can be displaced and used instead to minimize judder from dropping frames. This method is not used to reduce overall motion to photon latency.

The combination of prediction [LaValle et al. 2014] and timewarp on the Oculus Rift has resulted in drastic improvements in motion to photon latency [Kijima and Miyajima 2016; Evangelakos and Mara 2016]. We would like to continue to improve this by introducing our image displacement on the HMD.

3 Method

In order to push acquisition of IMU data even further out along the display pipeline, we have implemented a system that modifies the HDMI signal after it has entered the HMD. HDMI video is transmitted with both an active pixel area and a blanking area that is a legacy component of previous video formats. This blanking area was originally used to blank the cathode ray when returning to the beginning of a line or to the top of the screen, but has since been used in digital video formats to transmit additional data such as Closed Captioning data over air or audio data in other HDMI implementations. This blanking area is divided into the front porch and back porch margins for both vertical and horizontal directions. It also contains the HSYNC and VSYNC signals used to synchronize the displayed lines on the screen. A visualization of the active and blank pixel data in a typical video frame can be seen below in Figure 1 and a visualization of the timing for each frame can be seen in Figure 2.
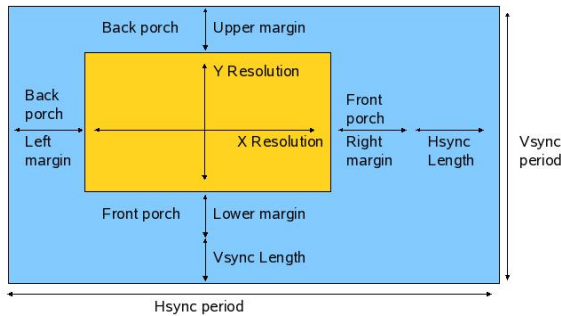

Figure 1: *Active and blank pixel data in typical video frame. Source: http://processors.wiki.ti.com/index.php/LCD_RGB_640x480_VGA_Addition*

The exact length of horizontal or vertical front porches and back porches vary between manufacturers and the only requirement is that the VSYNC and HSYNC periods are consistent, the pixel clock is appropriate for transmitting all the pixel data, and all frequencies are supported by the display being used.

By dynamically adjusting the length of the front porch and back porch without modifying incoming pixel data, the blanking signal can be de-asserted earlier during a scanline than it normally would be, and then reasserted earlier as well. This has the effect of transmitting blank pixel data for a small portion of the active region and then cutting off the normal active region early and effectively shifting the image on the display.

To test this system, a Digilent Nexys Video FPGA development kit was utilized. This development kit contains an Aritix-7 FPGA, speed grade 1, as well as HDMI in and HDMI out ports. A 1280x720 static test image was generated on the FPGA and transmitted over the HDMI out port to an external monitor (LG Flatron
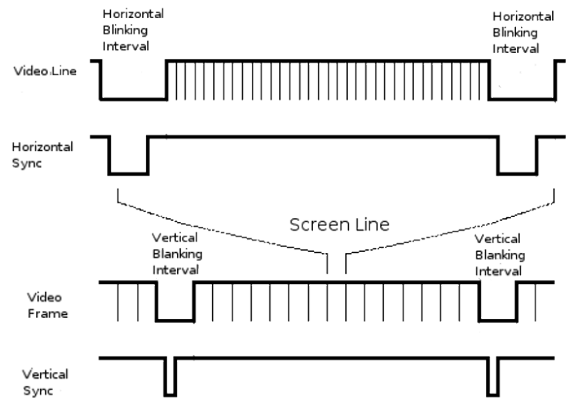

Figure 2: *Timing of video signals. Source: http://javiervalcarce.eu/html/vga-signal-format-timming-specs-en.html*

E2742) capable of 1920x1080 resolution.

The lengths of the front porch and back porch were set to be 150 pixels in every direction to show that larger values can effectively be used. To compensate for the larger total pixel area, a pixel clock running at 96.75MHz was generated on the FPGA.

The front porch and back porch for the horizontal scanline were modified so that a constant value of 20 pixels could be added or subtracted utilizing switches on the development board. Additionally, one of the switches activated a state machine that would increment or decrement the offset value once per frame between -20 and 20 to show that the dynamic blanking could be changed on every frame.

4 Results

The implementation allowed a static video image to be displayed. When the toggle switch was activated that added 20 to the horizontal front porch and subtracted 20 from the horizontal back porch, the image on the screen shifted to the left by 20 pixels. This left a bar, 20 pixels wide, of black on the right edge of the screen. A similar result was found when shifting the front porch and black porch the other direction.

When activating the state machine to automatically change the offset values each frame, the image appeared to shift constantly to the left and right, accurately updating the shift once every frame.

5 Future Research

By showing that image displacement can occur by manipulating the HDMI signal on the HMD, we can consider the implications of utilizing an input signal sourced from a GPU displaying content rendered for VR/AR. Since front porch and back porch lengths are generated in the GPU, we would initially be restricted to utilizing whatever extra area is provided by the incoming signal for all shifts. Any change in the front porch and back porch areas at a midpoint in the HDMI signal would necessitate a corresponding change in pixel clock.

This provides an opportunity for further reducing the motion to photon latency in the system. By buffering the incoming active pixel data in a first-in first-out (FIFO) structure on the FPGA, we can then generate a much higher pixel clock that can approach the maximum value allowed by the display being used. The blanking margins can

then be sized such that the output for active pixel data occurs after a large amount of the frame has been buffered in the FIFO.

With appropriate timing, the final scanline of active pixel data can be sent to the display at the same time as it would have been sent natively, resulting in no net latency additions to the transmission of the frame. Because we are able to delay the start of the frame to the display by a significant amount of time, we can also delay reading the IMU data and adjusting the dynamic blanking signal a significant amount of time as well, resulting in a further reduction of motion to photon latency.

In a typical 720p display running at a 60Hz vertical refresh rate, the pixel clock necessary for transmitting only the 1280x720 pixels of active pixel data is about 55MHz. Typical 720p 60Hz displays run on a pixel clock of around 75MHz to account for the blanking area as well. If a display can be driven at a pixel clock of 200MHz, the actual time required to send the pixel data is only about 4.6ms of the 16.67ms frame. By aligning the active pixel data at the end of the frame time, over 10ms of additional motion to photon latency can potentially be eliminated.

Further investigation determining the extent to which current display technology could support this, as well as implementation of a system to dynamically generate the appropriate timing structures based on the input signal are required.

6 Conclusion

We have demonstrated a method by which an image displayed on a VR/HR HMD can be displaced to account for head motion after the initial rendering of the frame. While there exist many other systems to assist in reducing the motion to photon latency in head-tracked displays, this method is novel in that it can be completely implemented on the HMD with no interaction with the rendering pipeline.

This method has several limitations that do not allow it to be the only method used to reduce latency. As a 2D displacement, the system cannot produce parallax due to positional changes or display areas that were not part of the originally rendered scene. Because of this, the correction allowed by this system is limited to very small corrections.

The primary advantage of this method is the extremely late acquisition of accurate orientation data, even well after the frame has been sent to the display by the GPU. In combination with other existing methods such as prediction and rendering timewarp, this system can provide a last mile accuracy to correct any small prediction errors.

Acknowledgments

References

AKIDUKI, H., NISHIIKE, S., WATANABE, H., MATSUOKA, K., KUBO, T., AND TAKEDA, N. 2003. Visual-vestibular conflict induced by virtual reality in humans.

ANTONOV, M., 2015. Asynchronous timewarp examined. Oculus VR, LLC, https://developer.oculus.com/blog/asynchronous-timewarp-examined/.

BINSTOCK, A., 2015. Optimizing vr graphics with late latching. Oculus VR, LLC, https://developer.oculus.com/blog/optimizing-vr-graphics-with-late-latching/.

BURBIDGE, D., AND MURRAY, P. M. 1989. Hardware Improvements To The Helmet Mounted Projector On The Visual Display Research Tool (VDRT) At The Naval Training Systems Center. International Society for Optics and Photonics, J. Carollo, Ed., 52–60.

EVANGELAKOS, D., AND MARA, M. 2016. Extended TimeWarp latency compensation for virtual reality. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games - I3D '16*, ACM Press, New York, New York, USA, 193–194.

HELD, R. M., AND DURLACH, N. I. 1992. Telepresence. *Presence: Teleoperators and Virtual Environments 1*, 1 (jan), 109–112.

KIJIMA, R., AND MIYAJIMA, K. 2016. Looking into HMD: A method of latency measurement for head mounted display. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, IEEE, 249–250.

LAVALLE, S. M., YERSHOVA, A., KATSEV, M., AND ANTONOV, M. 2014. Head tracking for the Oculus Rift. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 187–194.

MARK, W. R., MCMILLAN, L., AND BISHOP, G. 1997. Post-rendering 3D warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*, ACM Press, New York, New York, USA, 7–ff.

MAZURYK, T., AND GERVAUTZ, M. 1995. Two-step Prediction and Image Deflection for Exact Head Tracking in Virtual Environments. *Computer Graphics Forum 14*, 3 (aug), 29–41.

SO, R. H. Y., AND GRIFFINT, M. J. 1992. Compensating Lags in Head-Coupled Displays Using Head Position Prediction and Image Deflection. *JOURNAL OF AIRCRAFT 29*, 6.

ZIELINSKI, D. J., RAO, H. M., SOMMER, M. A., AND KOPPER, R. 2015. Exploring the effects of image persistence in low frame rate virtual environments. In *2015 IEEE Virtual Reality Conference, VR 2015 - Proceedings*, 19–26.