Minimizing Latency for Augmented Reality Displays: Frames Considered Harmful

Feng Zheng*[1], Turner Whitted†[2], Anselmo Lastra‡[1], Peter Lincoln§[1], Andrei State¶[3], Andrew Maimone‖[1], and Henry Fuchs**[1]

[1]The University of North Carolina at Chapel Hill
[2]TWI Research and UNC-CH
[3]InnerOptic Technology Inc. and UNC-CH

Abstract

We present initial results from a new image generation approach for low-latency displays such as those needed in head-worn AR devices. Avoiding the usual video interfaces, such as HDMI, we favor direct control of the internal display technology. We illustrate our new approach with a bench-top optical see-through AR proof-of-concept prototype that uses a Digital Light Processing (DLP$^{\text{TM}}$) projector whose Digital Micromirror Device (DMD) imaging chip is directly controlled by a computer, similar to the way random access memory is controlled. We show that a perceptually-continuous-tone dynamic gray-scale image can be efficiently composed from a very rapid succession of binary (partial) images, each calculated from the continuous-tone image generated with the most recent tracking data. As the DMD projects only a binary image at any moment, it cannot instantly display this latest continuous-tone image, and conventional decomposition of a continuous-tone image into binary time-division-multiplexed values would induce just the latency we seek to avoid. Instead, our approach maintains an estimate of the image the user currently perceives, and at every opportunity allowed by the control circuitry, sets each binary DMD pixel to the value that will reduce the difference between that user-perceived image and the newly generated image from the latest tracking data. The resulting displayed binary image is "neither here nor there," but always approaches the moving target that is the constantly changing desired image, even when that image changes every 50 µs. We compare our experimental results with imagery from a conventional DLP projector with similar internal speed, and demonstrate that AR overlays on a moving object are more effective with this kind of low-latency display device than with displays of similar speed that use a conventional video interface.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Realities

1 Introduction

In the past several decades, Augmented Reality (AR) has been shown to be potentially useful in a variety of areas, such as medicine, manufacturing, maintenance, navigation and telepresence. Many of these may benefit from head-worn, eyeglass-style displays, which are currently evolving rapidly [7, 1, 2, 6]. These displays optically combine the computer-generated image with the user's direct view of the surroundings ("optical see-through"), in contrast to smartphone- and tablet-based AR applications, which combine the computer-generated image with video imagery ("video see-through"). For head-worn displays, optical see-through with its direct and undegraded view of the surroundings is desirable and likely indispensable for extended use. However, it comes at a cost; unlike video see-through displays, which allow synchronization of real and virtual images by deliberately delaying the video stream, optical see-through AR must present synthetic imagery at the speed of "reality" to keep virtual and real objects aligned. Hence it must rely on minimal latency or on prediction techniques when computing synthetic imagery [21]. The latency in today's AR systems, even those optimized for low latency, often exceeds mere annoyance or distraction, and often makes optical see-through unusable. The debilitating effects are not just the magnitude of the offset between the intended and the achieved location of the computer-generated object, but also the change in the offset as a function of time – the synthetic object appearing to "slosh" or "swim" about the real scene [11]. While predictive tracking can significantly reduce the misalignment between synthetic and real imagery, errors are still present, especially during rapid changes in head pose [8, 27].

Unfortunately, latency accumulates throughout all the components of an AR system (tracking, application, image generation, scanout, display). This paper concentrates on the latency in the image scanout and display itself.

Today's most common display technologies (LCD, OLED, DMD) form images through various methods of controlling light: spatially, temporally, and in terms of wavelength (or even polarization). Historically, and until today, these capabilities have been internally "managed" by device designers, while end users have been limited to common display interfaces (VGA, DVI, HDMI). While these interfaces allow plug-and-play flexibility, they impose certain restrictions that are difficult to work around. Specifically, as this abstract layer is derived from the raster scan method (developed in the late 1930s for Cathode Ray Tube (CRT) television sets), it introduces almost an entire video frame of latency in the display device itself. For example, with DMDs, color imagery is almost always delivered via frame-sequential display—e.g., all pixels of the red channel displayed simultaneously, then all pixels of the blue channel, then all pixels of the green channel. Since the display interface is raster-scan-based, a DMD device has to receive an entire image before it can start to display even the first pixel of that image.

Even on simpler devices, such as a CRTs, the display of the bottom of the image occurs much later than the display of the top of the image. Raster scan is inherently unsuited for low-latency appli-

*e-mail: zhengf@cs.unc.edu
†e-mail: jtw@twilab.com
‡e-mail: lastra@cs.unc.edu
§e-mail: plincoln@cs.unc.edu
¶e-mail: andrei@cs.unc.edu
‖e-mail: maimone@cs.unc.edu
**e-mail: fuchs@cs.unc.edu

IEEE International Symposium on Mixed and Augmented Reality 2014
Science and Technology Proceedings

cations, unless scanout is performed at very high rates, which tends to cause memory access and high-power utilization issues.

In this paper we advocate "de-abstracting" this display interface layer and exposing the technology underneath to the image-generation process. This will permit the image generation processors to "get closer" to the control of the photons in the display, achieving dramatically lower overall latencies.

After a review of related work in Section 2, Section 3 proposes a general low-latency algorithm that is independent of the target display device. Section 4 introduces the basics of the display device being used and specializes the algorithm to match that device's specifications and limitations. We demonstrate our prototype system in Section 5, draw conclusions in Section 6 and discuss future directions in Section 7.

2 Related Work

Researchers have identified the need for minimal total system latency in both VR and AR applications [19, 5]. To avoid certain deleterious effects of VR (such as what is commonly known as "simulator sickness"), it is desirable to keep system response to head motion roughly as fast or faster than the vestibuloocular reflex, one of the fastest reflexes in the human body at 7 ms to 15 ms [3], which rapidly stabilizes the retinal image at the current fixation point by rotating the eye in response to head motion. For example, the developers of the Oculus VR headset recommend "20 ms or less motion-to-photon latency" [28]. To help developers reach that goal, they have recently reduced the latency of the Oculus Rift tracking subsystem to 2 ms [4]. Even smaller total latencies are recommended when a VR experience conveying a high sensation of presence is needed: to avoid any perception of scene motion due to latency, values as low as 3 ms should not be exceeded [12, 13]. A NASA study investigating the utility of head-worn displays for flight deck "Synthetic/Enhanced Vision Systems" concludes that commonplace "head movements of more than 100 °/s would require less than 2.5 ms system latency to remain within the allowable [Head-Up Display] error levels" [9].

Touch-based interaction with displays also represents a form of AR, in that the user should ideally perceive display elements as being affected by touch as if they were tangible objects (e.g. when dragging). Previous work in this related area covers both user perception and task performance; its conclusions include that "there is a perceptual floor somewhere between 2 – 11 ms, below which users do not notice lag" and that "latencies down to 2.38 ms are required to alleviate user perception when dragging" [15, 18].

It is important to note that until now, all approaches striving to reduce rendering latency—even unusual ones such as frameless rendering [10]—have been applied to displays with standard video interfaces, such as VGA, DVI, or HDMI.

3 Approach

We minimize latency by updating selected parts of the displayed image—those that require the most change—instead of the complete image. Updating arbitrary individual pixels is generally not feasible; ideally, we would update small groups of pixels in parallel at a bandwidth as high or higher than current full-frame bandwidth. This leads to higher update rates, albeit of smaller display regions. While no currently available display accommodates this update mode, we propose a broad framework for the ideal device and then specialize the algorithm for an existing one.

The goal of this algorithm is—at every update of the display—to bring the image that is perceived by the viewer closer to an estimate of the latest true image, as determined by the tracker. We call this estimate of the true image the Desired Image. Producing the Desired Image by conventional rendering would be challenging at the rates at which we want to update the display, which is on the order of tens of thousands of updates per second. We propose rendering from polygons (or other primitives) at as high an update rate as a GPU can produce, and then computing a 3D warp from two nearby rendered images [16] to approximate the desired image. If a 3D warp at the desired update rate is not possible, then adding another, computationally less expensive approximation with a 2D warp is a possibility. Thus we have a sequence of rendering steps (see Figure 1), each computationally less demanding and updating at a faster rate than the previous one. We aim to achieve through this mechanism a total rendering latency of under 0.1 ms. (Note that our prototype does not fully implement this rendering pipeline. It is introduced for the integrity of discussion.)

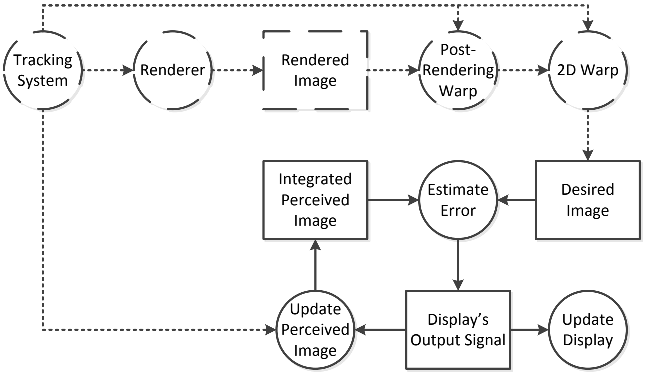


Figure 1: Data path framework. While the whole approach comprises many stages, each operating faster than the prior stage, our current prototype implements only the solid-line stages.

We must also maintain an estimate of what the user perceives. Since the display is updating very rapidly, the estimate of the perceived image must be an integral of what the viewer has seen over a short period of time in the past. We call this the Integrated Perceived Image. Abstractly, the algorithm works as follows.

1. Query the tracker and produce the Desired Image.
2. Create an Error Image from the Desired Image and the Integrated Perceived Image.
3. In the Error Image, select the area with the most error.
4. Update the selected display region to reduce the error.
5. Update the Integrated Perceived Image.
6. Loop to step 1.

The Error Image may be as simple as a per-pixel difference, or alternatively a perceptual metric. The display update step is heavily dependent on the capabilities of the target device. For example, the device that we have been using (see Section 4) can instantaneously display only binary images, and forms continuous-tone images by pulse-width modulation.

4 DMD as a Low Latency Display

The most accessible display technology for our approach is the digital micro-mirror device (DMD) manufactured by Texas Instruments as Digital Light Processing (DLP^{TM}). Low level, rapid display using DMDs has been demonstrated by numerous groups [20, 17, 14]. We used the TI Discovery 4100 Development Kit [25] with a DLP7000 [24] DMD chip capable of displaying 1024 × 768 pixels.

To construct a low latency image generation pipeline with this DMD device, we assume a high-speed tracker that can deliver the user pose with 1.5 ms of latency (only slightly faster than the current Oculus Rift tracker [4]), and a renderer that can generate the Desired Image for that user pose with 0.1 ms latency, as discussed in Section 3. These leave a display latency budget of 0.4 ms if we are not to exceed the perceptual floor of 2 ms [15, 18].

4.1 DMD Chip Basics

A DMD chip is primarily a random access memory device with an array of deformable mirrors. The 2D memory on the chip is split into two buffers, each with single-bit-sized elements: one buffer that the processor can write into (the "back buffer") and one buffer which controls each pixel's mirror (the "front buffer"). To copy from the back buffer to the front buffer, the processor must assert a Mirror Clocking Pulse (MCP). On the DLP7000, the controlling processor can assert this pulse at any time, though it operates on one, two, or four blocks of 48 rows each, or on the whole array simultaneously. This DMD cannot accept another MCP while processing a prior MCP for 4.5 µs, and it cannot accept updates to any buffer (front or back) on a block undergoing an MCP for 12.5 µs, after which the mirrors of that block will have stabilized. This combination of back buffer writes and MCPs allows pipelining of buffer updates and mirror commits. Since the pixel clock for this DMD is maximally 400 MHz, and one row requires 16 cycles, this means that an entire block is written in $(16 \times 48)/(400\,\text{MHz}) = 1.92\,\mu\text{s}$. Note that the MCP cycle time is 4.5 µs, longer than a single block update; as a result, it is more efficient to update two or four blocks between MCPs.

Therefore, with this DMD chip, the maximum latency from the start of memory writes to photon output for a single block (i.e. assert an MCP for one block only) is 14.42 µs, which supports our target latency of 0.4 ms for the entire frame (16 blocks).

4.2 Standard DMD Projector Basics

Typical DMD projectors uniformly illuminate the entire mirror array. Controlling each pixel's mirror deflection angle between the two powered states causes the light to either exit the projector (On) or hit an absorbing baffle (Off). The intensity of light that a user perceives at a given pixel is simply a function of the percentage of time that the pixel's mirror is in the On state. Given an 8-bit intensity value, the duty cycle executed may take the form of different durations for each bit. For example, to process one 8-bit value, the state of the most significant bit could control a mirror for $1/2$ of the frame time, the next bit for $1/4$, ... and the least significant bit for $1/256$. This basic mode supports only gray-scale imagery. DMD projectors often provide color though color-sequential methods, usually by spinning a color wheel in front of the light, or by alternating among multiple illumination LEDs. While a single color is active, the controller executes the mirror sequence for the intensities of that color. In this way, these projectors only emit one color at a time; for a 60 Hz projector, the colors may alternate at 180 Hz.

These DMD projectors control the duty cycles of the mirrors based on the video input they receive. Typically this input is supplied via a DVI, HDMI, VGA, or DisplayPort connection. All of these connections supply video in a raster scan format, in which a complete frame arrives, pixel-by-pixel, over a full frame time (e.g. $1/60$ s). Since most DMD projectors feature a color-sequential display and use pulse-width modulation to achieve varying intensities, they must buffer a complete full-color frame before starting to load the DMD's back buffer, resulting in a latency of at least one frame time by the interface alone, which is much longer than would be desirable for an optical see-through head-mounted display (HMD).

4.3 Low Latency Custom DMD Projector

In order to reduce the latency between image production and display, one needs lower-level control over the DMD projector than is afforded by a conventional video interface. Our experimental DLP7000 projector does not support color, so we describe here the algorithm for generating gray-scale images, which can be extended to support color (see Section 7).

Algorithm 1: Low Latency Binary Projector Image Generation

```
Denote Desired Image as $I_{\mathbf{d}}$, Integrated Perceived Image as
$I_{\mathbf{u}}$, Error Image as $I_{\mathbf{e}}$, and Binary Projector Image as $I_{\mathbf{p}}$
for every pixel $\mathbf{x}$ at time $t$ do
    Compute $I^t_{\mathbf{u}}(\mathbf{x}) = 4\sum_{t'=t-64}^{t'=t-1} I^{t'}_{\mathbf{p}}(\mathbf{x}) - 1$
    Compute $I^t_{\mathbf{e}}(\mathbf{x}) = I^t_{\mathbf{d}}(\mathbf{x}) - I^t_{\mathbf{u}}(\mathbf{x})$
    Compute $I^t_{\mathbf{p}}(\mathbf{x}) = \begin{cases} 1 & \text{if } I^t_{\mathbf{e}}(\mathbf{x}) > 0 \text{ or } I^t_{\mathbf{d}}(\mathbf{x}) = 255 \\ 0 & \text{otherwise} \end{cases}$
```

Unfortunately, the DLP7000 only supports updating entire frames, rather than rows, blocks, or small groups of pixels. It can update and display a full binary image at 22 727 Hz (slightly over 44 µs per update). A custom controller could theoretically execute 4-block MCPs every 4.5 µs. If certain blocks did not require updates (no change to the back buffer), then the entire image could be updated with four 4-block MCPs in $4 \times 4.5\,\mu\text{s} = 18\,\mu\text{s}$, or 2.5 times faster than on the experimental projector. In many AR overlay applications, opportunities for partial-screen updates are frequent, as virtual objects often do not cover the entire display.

As noted earlier, applying these specifications, capabilities, and limitations of a DMD leads to a specialization of the abstract algorithm from Section 3, starting at step 3:

Select Area with Greatest Error. For a custom controller using this DMD, the selectable areas would be among the four 4-block regions of the array; however, with the experimental projector controller, the only selectable area is the entire array.

Update Display Region. While the desired image may have multiple intensity bits, the DMD is limited to a single output bit per pixel: On or Off. This simplifies the output decision based on the error: for each pixel, if the Integrated Perceived Image is dimmer than the Desired Image, turn on the pixel, otherwise turn it off.

Update Integrated Perceived Image. In order to generate each pixel of the new Integrated Perceived Image, we integrate over a selected number of the most recent Binary Projector Images. We determined empirically that using the latest 64 binary frames is sufficient for our experimental setup, though future user studies can refine the duration of this integration window.

The DMD-specialized algorithm is summarized in Algorithm 1. As long as we can feed it appropriate desired images at the DMD's maximal load and pulse rate, we should be able to display a smooth, low-latency, gray-scale image stream.

5 Preliminary Results

The DLP7000 can rapidly update the entire DMD (rather than a subset of it) in 44 µs, for an update rate of 22 727 Hz [23]. Alas, its host interface cannot transfer a binary image in 44 µs, so in order to evaluate dynamic imagery in real time, we had to pre-calculate binary images and pre-load them into the projector's local RAM. This RAM has a capacity of 43 690 binary images, which corresponds to 1.92 s at the above update rate.

Figure 2 shows our experimental setup, with a proof-of-concept optical-see-through AR display. In addition to the DLP7000, we also used a conventional 60 Hz DMD projector (a DLP Lightcrafter version 2 [26]) for comparison. Either projector can frontally illuminate a flat surface, or it can project onto a rear-projection panel viewed through a beam-splitter to provide augmentation to a scene. We used a camera (iPhone5S, due to its ability to capture 720p imagery at rates as high as 120 Hz) to record a user's monoscopic viewpoint. The target scene consists of a rotating turntable, which

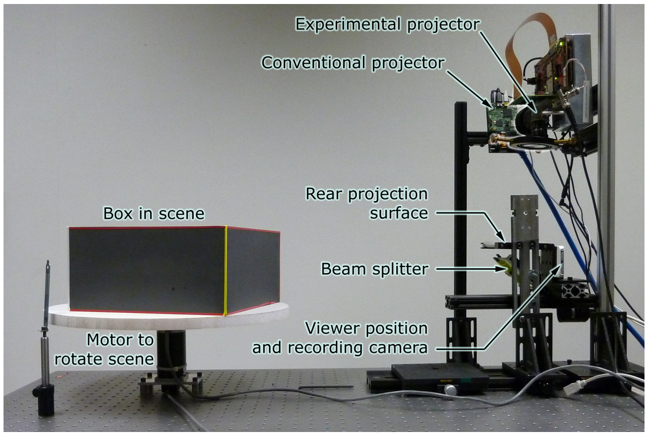


Figure 2: Experimental setup.

can be moved either by hand, with its motion tracked by a shaft encoder, or by a computer-controlled stepper motor. Objects such as a box or a pyramid are placed on the platter to provide a moving scene to test the effectiveness of AR overlay and registration. This setup is analogous to, but simpler to control experimentally than a tracked HMD user. In particular, we can take advantage of a very-low-latency tracker with controlled, repeatable motion.

5.1 Experiment 1: Latency

Our first experiment compared the latency of the conventional 60 Hz DMD projector with that of the low-latency experimental DLP7000. A simple three-axis cursor was positioned in 3D space at the tip of the physical pyramid model on the turntable. This cursor was rendered for each projector and for each rotational position of the platter, at intervals of $1/3°$. The platter's position was tracked by a shaft encoder and the appropriate image was displayed as soon as a shaft encoder pulse was received. The pulse is input to the DLP7000 via a 1-bit pin; thus only unidirectional random motion is supported, while there is no such limitation for the conventional projector. Figure 4 shows the results for conventional and experimental projectors as the user rotated the turntable at a maximum rate of $2/3$ Hz. As expected, the conventional projector's image lagged noticeably behind its intended location at the tip of the pyramid; the experimental projector's cursor remained registered.

5.2 Experiment 2: Low Latency Gray-scale Imagery Using Binary Image Generation

In the second experiment, we projected a rotating gray-scale test pattern (see Figure 6a) containing a combination of text, lines, gradients and photos. The resulting set of Binary Projector Images was displayed at full speed in a continuous 360° loop so the results could be examined visually. As expected, the imagery was rotating smoothly, exhibiting increased motion blur near the edges of the spinning test pattern and very little motion blur near the center. No artifacts were observed. Figure 5 shows a selection of frames from this experiment: Desired Images, Integrated Perceived Images, Error Images, and Binary Projector Images. Figure 6 and the accompanying video[1] show the observed dynamic results.

5.3 Experiment 3: AR Imagery on Moving Object

Since our experimental projector requires pre-loaded (and therefore pre-computed) binary images, the real-life bidirectional motion of the object in this third experiment must be known in advance. Therefore, instead of moving turntable and object by hand unidirectionally, we moved it with a PC-controlled stepper motor, through a predefined series of angular positions, in both directions and at varying speeds. Figure 3 shows one such motion profile covering the experiment pictured in Figure 7. The sequence lasted 1.92 s, during which 43 690 binary images were displayed.

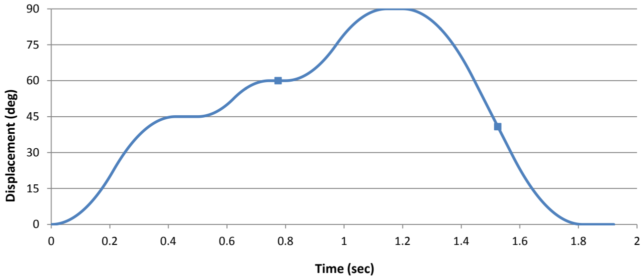


Figure 3: The rotation motion path used. The two dots along the curve indicate the time instants shown in Figure 7.

It is important to note that this experiment evaluates visual quality, not latency or registration. Note in Figure 7 and in the accompanying video[1] that the imagery is sharp when the cube is still, and is appropriately motion-blurred when the cube is moving rapidly.

6 Discussion and Conclusion

The proposed low-latency update algorithm produces visually pleasing results. Rapid updates decrease or eliminate the "swimming" artifacts induced by latency, and the imagery shown by our proposed display is more natural and resembles motion blur, which is more acceptable to viewers. Without the current hardware's limitations, we expect even better results because we could prioritize updates on portions of the display, rather than updating the full binary DMD array as shown.

We believe that to achieve low-latency in displays, we must abandon full-frame updates, which necessarily induce excessive latency (unless the update rates are extremely high). This means that we must also move away from the frame-based legacy display interfaces that are modeled on decades-old CRT technology.

7 Future Work

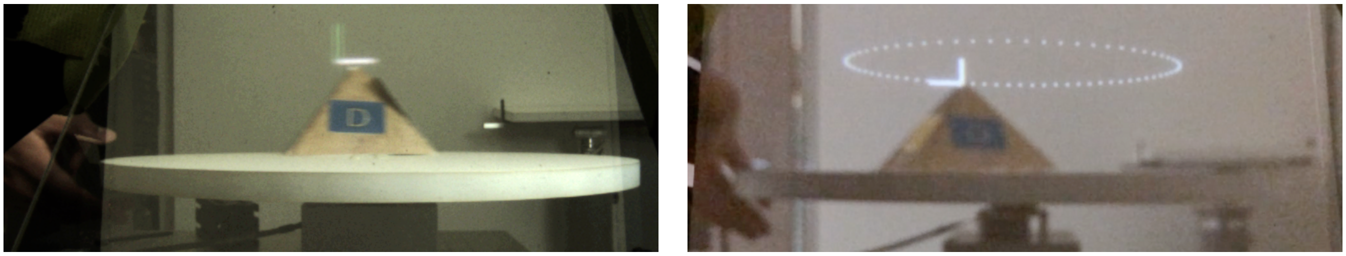
We will next focus on developing a real-time display by designing custom hardware to control the DMD directly, bypassing limitations of the current control hardware. The implementation will be on a high-performance FPGA, with a high degree of parallelism. Our Binary Projector Image generation algorithm is easy to parallelize and can be implemented as a fixed-function pipeline with simple integer math. It requires little memory as it involves only a small number of most recent projected images and two gray-scale images (one integrated Perceived Image and one Desired Image).

As the bandwidth required to drive the DMD is very high, the control circuitry must be physically close to the DMD chip. We expect to supply images from a GPU to the controller (in pairs and with depth, to enable 3D warping [16]) over a conventional video interface, such as DVI [22]. Additionally, tracking data must be transmitted. The display controller should include circuitry to warp the received images (see Figure 1) to produce the Desired Images at high rates, as well as to compute the Perceived and Error Images at the same speeds. This direct rapid control may reduce latency as well as power consumption, and may result in higher image quality.

Extension to color images, via an approach similar to frame-sequential color, appears straightforward. We expect the next experimental projector to have three colored light sources, for instance red, green and blue LEDs. Switching between color channels could occur either at every update, perhaps every 50 µs, or less frequently if the system were to support mirror changes by blocks, as expected.

Longer-term plans include investigation of other display types that can be updated rapidly and are suitable for head-worn displays.

[1] http://youtu.be/dBFdBm9Ab9E



(a) Conventional 60 Hz color display. Note that the overlay is displaced significantly from the tip of the pyramid.

(b) Experimental display at 1 kHz. Without the need to operate at the maximum rate of 22 727 Hz, 1 kHz is enough to show the benefit of using this low latency display.

Figure 4: AR registration of a moving object (pyramid). These frames were filmed by a 120 Hz camera through a beam splitter (see Figure 2).

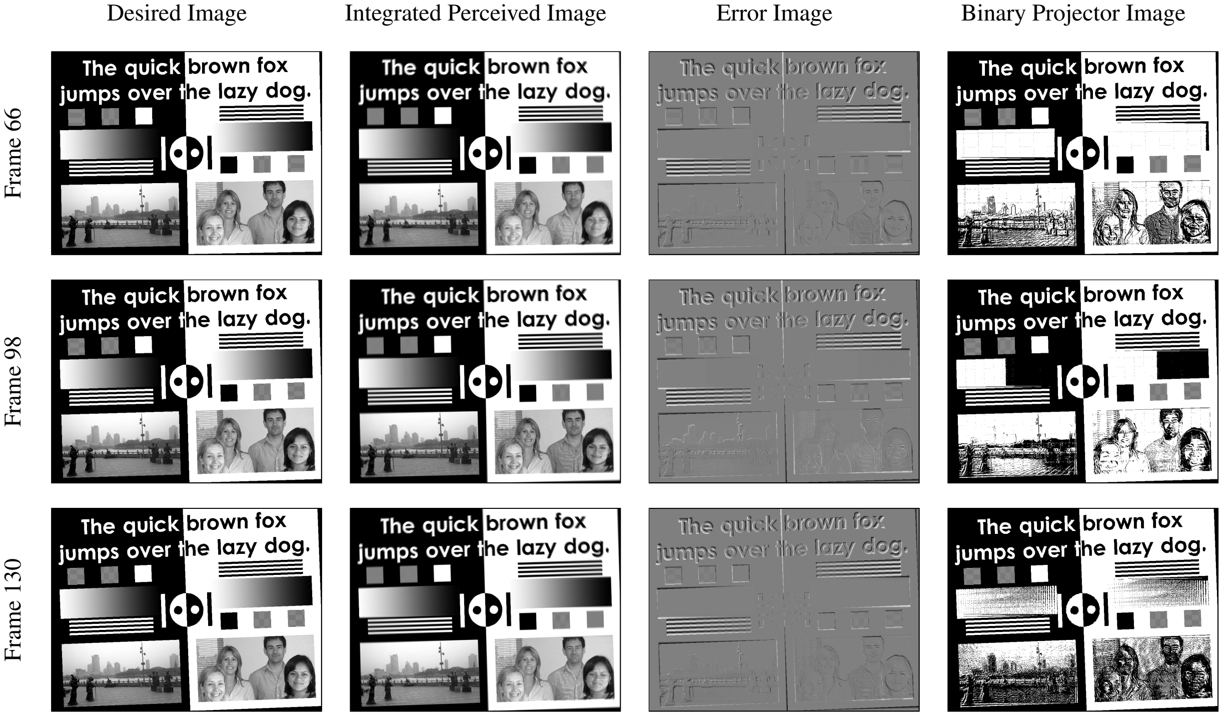


Figure 5: Sample images used by our algorithm when displaying a rotating test pattern with the experimental low latency projector. The pattern (see Figure 6a) rotates at 360 °/s to produce the Desired Images. For clarity, a border has been added to each image.

Finally, we plan to research approaches to a low latency equivalent of a device-independent interface, analogous to HDMI or DisplayPort for conventional displays. This would be an abstract interface to be used between a device-independent low latency renderer and a renderer-independent low latency display, enabling more of the proposed algorithm to be implemented in a GPU.

Acknowledgments

The authors thank Kurtis Keller and Jim Mahaney for engineering advice and support. Lu Chen assisted with the implementation of an earlier prototype. Zachary Kaplan helped produce the accompanying video. Sharif Razzaque provided psychophysics insight. This research was supported in part by the BeingThere Centre, a collaboration between Eidgenössische Technische Hochschule (ETH) Zürich, Nanyang Technological University (NTU) Singapore, and University of North Carolina (UNC) at Chapel Hill. The Centre is supported by these three institutions and by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the Interactive Digital Media Programme Office.

References

[1] Lumus - Home. http://www.lumus-optical.com/.

[2] M2000AR | Vuzix. http://www.vuzix.com/consumer/products_m2000ar/.

[3] Vestibuloocular reflex testing. http://emedicine.medscape.com/article/1836134-overview, February 2012.

[4] Building a sensor for low latency virtual reality — Oculus Rift - virtual reality headset for 3d gaming. http://www.oculusvr.com/blog/building-a-sensor-for-low-latency-vr/, January 2013.

[5] G-SYNC – Technology – GeForce. http://www.geforce.com/hardware/technology/g-sync/technology, October 2013.

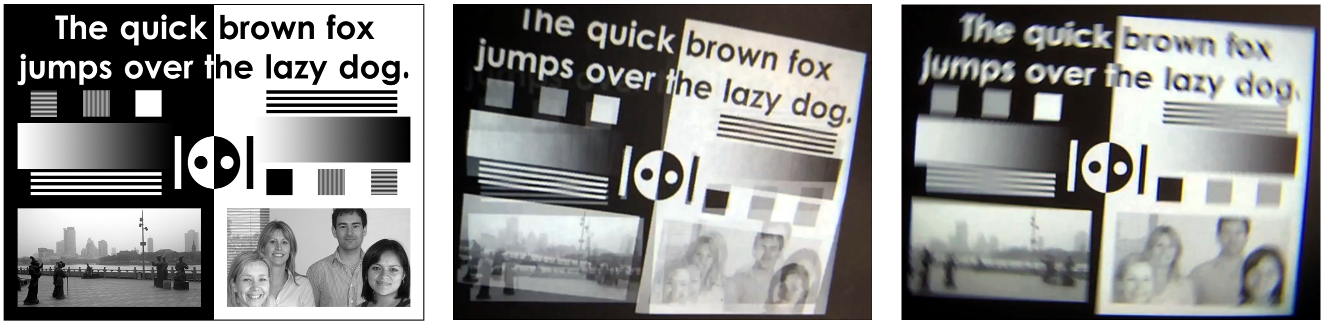
[6] Meta - spaceglasses. https://www.spaceglasses.com/, 2013.

[7] Google Glass - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Google_Glass, March 2014.

[8] R. T. Azuma. *Predictive Tracking for Augmented Reality.* PhD thesis, University of North Carolina at Chapel Hill, 1995.

[9] R. E. Bailey, J. J. Arthur III, and S. P. Williams. Latency Requirements for Head-Worn Display S/EVS Applications. In *Proc. SPIE*, volume 5424, pages 98–109, 2004.

[10] A. Dayal, C. Woolley, B. Watson, and D. Luebke. Adaptive Frameless Rendering. In *Proc. Eurographics Conference on Rendering Tech-*

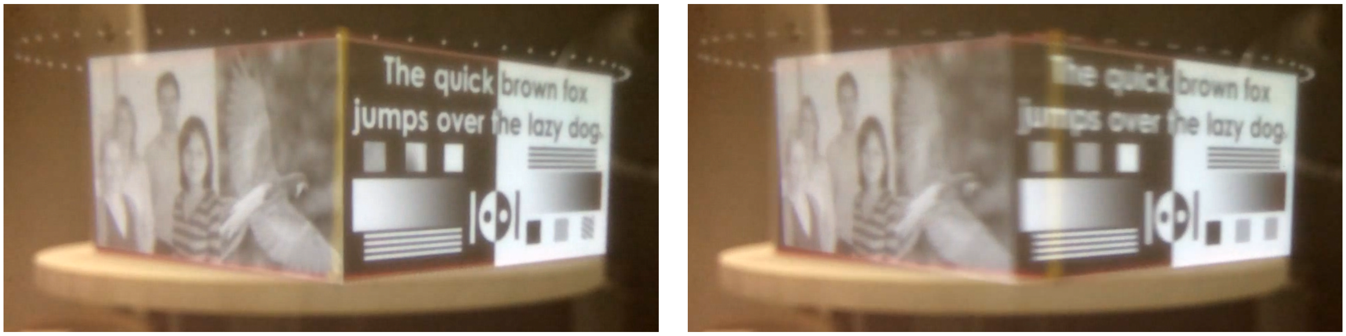


(a) Original Test Pattern. Photos from Wikimedia Commons: "'Leaving home' Statue in Ningbo" ©Siyuwj; "Client Advisory Team Photo" ©Mshannon, both CC-BY-SA-3.0 / GFDL.

(b) Projected using a conventional gray-scale DMD projector. Note that the image is generally sharp within each consecutive frame, though these two frames are distinctly visible, which results in jumpy motion.

(c) Projected using experimental projector. Note that the center of the image is sharper while the outside edges are more blurred (no distinct frames are visible), which results in smooth motion.

Figure 6: A frame of the rotating pattern from (a) projected onto a flat surface and captured with a 120 Hz camera, (b), (c). The pattern rotates at a constant rate of 360 °/s.



(a) $t \approx 0.78$ s: The cube is idle, though shaking (not intended, due to mechanical instability).

(b) $t \approx 1.53$ s: The cube is rotating quickly ($\omega \approx 240$°/s.)

Figure 7: The cube—with AR augmentation—rotates on a computer-controlled motion platform. These images were recorded from the approximately calibrated viewpoint by a 120 Hz camera filming through the beam splitter (see Figure 2). Due to preliminary calibration inaccuracies, the virtual texture overlay is not registered to the real box indicated by the bright red and yellow wire frame (see Figure 2). It is important to note that this experiment is used to evaluate visual quality, not latency.

niques, pages 265–275, 2005.

[11] R. L. Holloway. Registration Error Analysis for Augmented Reality. *Presence*, 6(4):413–432, 1997.

[12] J. Jerald and M. Whitton. Relating Scene-Motion Thresholds to Latency Thresholds for Head-Mounted Displays. In *Proc. IEEE Virtual Reality*, pages 211–218, March 2009.

[13] J. J. Jerald. *Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays*. PhD thesis, University of North Carolina at Chapel Hill, 2009.

[14] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec. Achieving Eye Contact in a One-to-Many 3D Video Teleconferencing System. *ACM Trans. Graph.*, Jul. 2009.

[15] R. Jota, A. Ng, P. Dietz, and D. Wigdor. How Fast is Fast Enough?: A Study of the Effects of Latency in Direct-touch Pointing Tasks. In *Proc. ACM CHI*, pages 2291–2300, 2013.

[16] W. R. Mark, L. McMillan, and G. Bishop. Post-Rendering 3D Warping. In *Proc. ACM I3D*, pages 7–16, 1997.

[17] I. McDowall and M. Bolas. Fast Light for Display, Sensing and Control Applications. In *IEEE VR Workshop on Emerging Display Technologies*, 2005.

[18] A. Ng, J. Lepinski, D. Wigdor, S. Sanders, and P. Dietz. Designing for Low-latency Direct-touch Input. In *Proc. ACM UIST*, pages 453–464, 2012.

[19] M. Olano, J. D. Cohen, M. R. Mine, and G. Bishop. Combatting Rendering Latency. In *Proc. ACM I3D*, pages 19–24, 1995.

[20] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays. In *Proc. ACM SIGGRAPH*, pages 179–188, 1998.

[21] J. P. Rolland and H. Fuchs. Optical Versus Video See-Through Head-Mounted Displays. In *Presence: Teleoperators and Virtual Environments*, pages 287–309, 2000.

[22] G. Stoll, M. Eldridge, D. Patterson, A. Webb, S. Berman, R. Levy, C. Caywood, M. Taveira, S. Hunt, and P. Hanrahan. Lightning-2: A High-performance Display Subsystem for PC Clusters. In *Proc. ACM SIGGRAPH*, pages 141–148, 2001.

[23] Texas Instruments. ALP-4.1 controller suite for DLP® Discovery$^{\text{TM}}$ 4100. http://www.dlinnovations.com/doc/ALP4_Basic_Datasheet.pdf.

[24] Texas Instruments. DLP® 0.7 XGA 2xLVDS Type A DMD. http://www.ti.com/lit/ds/symlink/dlp7000.pdf, June 2013.

[25] Texas Instruments. DLP®Discovery$^{\text{TM}}$4100 chipset data sheet. http://www.ti.com/lit/er/dlpu008a/dlpu008a.pdf, June 2013.

[26] Texas Instruments. 0.3 WVGA Chipset — DLP Lightcrafter — TI.com. http://www.ti.com/lsds/ti/dlp/03-wvga-chipset-lightcrafter.page?DCMP=dlplightcrafter-en&HQS=dlplightcrafter, 2014.

[27] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. The Hiball Tracker: High-performance Wide-area Tracking for Virtual and Augmented Environments. In *Proc. ACM VRST*, pages 1–21, 1999.

[28] R. Yao, T. Heath, A. Davies, T. Forsyth, N. Mitchell, and P. Hoberman. Oculus VR Best Practices Guide v 0.008. http://static.oculusvr.com/sdk-downloads/documents/OculusBestPractices.pdf, March 2014.