

Minimizing Latency for Augmented Reality Displays: Frames Considered Harmful

Category: Research

ABSTRACT

This paper presents initial results of a new approach to image generation for low-latency displays such as those needed in head-worn AR devices. Our approach avoids the usual video interfaces, such as HDMI, in favor of directly controlling the internal display technology—LCD, OLED, DLP, or similar. We illustrate this new approach with a bench-top optical see-through AR proof-of-concept prototype that uses a DLP projector whose Digital Micro-mirror Device (DMD) imaging chip is directly controlled by a computer, similar to the way a RAM (memory chip) is controlled. We show that a perceptually-continuous-tone dynamic multi-grey-level image can be efficiently composed from a very rapid succession of binary (partial) images, with each (partial) binary image calculated from the continuous-tone image generated with the most recent tracking data.

Since the DMD projects only a binary image at any instant, 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 is to maintain an estimate of the image the user currently perceives, and at every opportunity allowed by the control circuitry, set 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 microseconds. 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 wide 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, in contrast to smartphone- and tablet-based AR applications, which combine the computer-generated image with video imagery of the surroundings. For head-worn displays, this “optical see-through” feature (in contrast to the “video see-through” of smartphone apps) is desirable since it offers a direct and undegraded view of the surroundings, likely indispensable for extended use. However, this desirable feature comes at a cost; unlike video see-through displays, which allow synchronization of real and virtual images via software control, optical see-through designs cannot delay the real “image,” so to keep virtual and real objects properly aligned, they must rely on minimal latency or prediction techniques in the computation of the synthetic imagery [21]. The latency in today’s AR systems, even those optimized for low latency, often exceeds mere annoyance or distraction, and makes many optical see-

through applications 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 has made significant improvements in reducing the discrepancy between the synthetic and real imagery, errors often still remain, especially during rapid changes in user head pose [8, 26].

Unfortunately, latency accumulates throughout all the components of an AR system (from tracking, to application, to image generation, scanout, and 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 great plug-and-play flexibility, they impose certain restrictions that are difficult to work around. Specifically, as this abstract layer is a raster scan interface (going back to late 1930’s CRT television technology), it introduces almost an entire video frame of latency in the display device itself. For example, with DMD (digital micro-mirror display - DLP), color imagery is almost always achieved with frame-sequential color—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 VGA/DVI/HDMI layer is raster-scan, 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 applications, 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-abstraction” 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.

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” [27]. 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

To reduce latency, we wish to update selected portions of the display (those that require the most change) instead of a full frame at a time. Updating arbitrary individual pixels in the display may not be feasible, but we want to at least be able to update small groups of pixels in parallel, at a display bandwidth that is as high or higher than the frame-oriented bandwidth available now. This leads to updates at higher rates, albeit of smaller regions of the display. Unfortunately there is no such ideal device that enables this type of update available to us today, but we can posit an algorithm that would work well with such a device, and then specialize the algorithm for existing (or upcoming) devices.

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.

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 position and produce the Desired Image.
2. Create an Error Image from the Desired Image and the Integrated Perceived Image.
3. From the Error Image, select the area of the display 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.

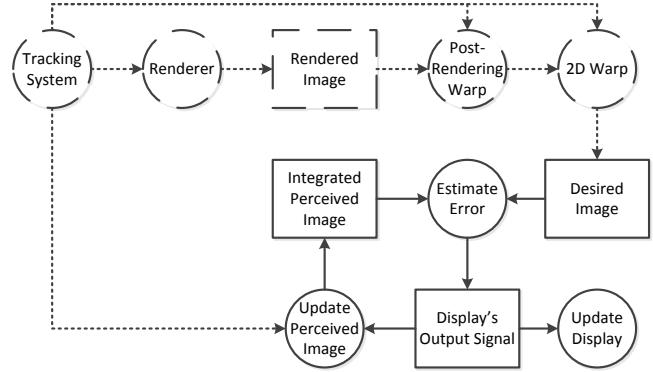


Figure 1: Data path framework. While the whole approach comprises many stages, each operating faster than the prior stage, our prototype implements only the solid-line stages.

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 below) 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 appears to be the digital micro-mirror device (DMD), manufactured by Texas Instruments as Digital Light Processing (DLP™). Low level, rapid display using DMDs has been demonstrated by numerous groups [20, 17, 14].

Let us consider how we would construct a low latency image generation pipeline with this DMD device. We assume we have a high-performance tracker that can give us the user pose at any time with 1.5 ms 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 Basics

A DMD chip is primarily a random access memory device with an array of deformable mirrors covered by a protective transparent cover. 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”). DMDs come in many sizes and resolutions. For our experimental projector, we used the TI Discovery 4100 Development Kit [24] with a DLP7000 [23] DMD chip capable of displaying 1024×768 pixels. This DMD imaging device’s IC has an active area of $20.736\text{ mm} \times 11.664\text{ mm}$, where each mirror is a $13.68\text{ }\mu\text{m}$ square. When energized, each pixel’s mirror can tilt between two states, depending on the value of the front buffer’s bit ($\pm 12^\circ$ from the unpowered, flat state).

To optimize writing speed into the buffers, multiple adjacent pixels can be written in parallel. The DLP7000 supports writing 32 adjacent pixels per clock pulse, but further requires that the entire row of 1024 pixels be filled as well, through a sequence of 16 clock cycles where data is specified on both rising and falling edges of the pixel clock. 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 $4.5\text{ }\mu\text{s}$, and it

cannot accept updates to any buffer (front or back) on a block undergoing an MCP for $12.5\ \mu\text{s}$, 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, 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 $4.5\ \mu\text{s}$, 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 $14.42\ \mu\text{s}$, 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 grayscale 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 illuminated 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 HDMI, DVI, 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. $1/60\ \text{s}$). Since most DMD projectors feature a color-sequential display and use duty cycles 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 AR HMD display.

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 input connection. As mentioned in Section 4.1, we used a TI Discovery 4100 Development Kit with a 1024×768 pixel DMD chip for our experimental projector. Our projector does not support color, so we only describe here the algorithm for generating grayscale images, though it could be extended to support color (see Section 7).

Unfortunately the experimental projector only supports updating entire frames, rather than rows, blocks, or small groups of blocks. It can update and display the entire binary image at 22 727 Hz (slightly over $44\ \mu\text{s}$ per update). A custom controller could theoretically execute 4-block MCPs every $4.5\ \mu\text{s}$. 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 the experimental projector. In an AR overlay application with an optical see-through HMD, opportunities for partial-screen updates are likely as the virtual objects presented may not cover the entire screen.

Applying these specifications, capabilities, and limitations of a DMD leads to a modification of the abstract algorithm from Section 3, starting at step 3:

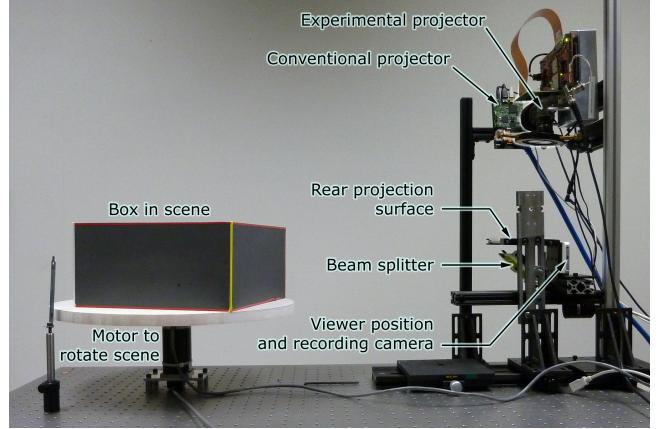


Figure 2: Experimental setup.

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 the new Integrated Perceived Image, for each pixel, we integrate over a selected number of the most recent binary images projected. We determined empirically that using the last 64 binary frames is sufficient (64 illuminated binary frames, followed by all dark frames in $1/120\ \text{s}$ saturated our capture camera). Future user studies can refine the duration of this integration window.

As long as we can feed the above algorithm appropriate desired images at the DMD's maximal load and pulse rate, we should be able to show the user a smooth, low-latency, grayscale appearance.

5 EXPERIMENTS

The TI Discovery 4100 kit can rapidly update the entire DMD (rather than a subset of it) in $44\ \mu\text{s}$ for an update rate of 22 727 Hz [22]. Unfortunately, the kit's interface to the host PC cannot transfer a new binary image in $44\ \mu\text{s}$, so in order to conduct real-time display experiments (necessary to evaluate dynamic imagery) we had to pre-calculate the binary images and pre-load them into the 4100's local RAM. This RAM has a capacity of 43 690 binary images, so the 4100 can run experiments lasting up to $1.92\ \text{s}$ each.

Figure 2 shows our experimental setup, with a proof-of-concept optical-see-through AR display. In addition to our experimental projector, we also used a conventional 60 Hz DMD projector (a DLP Lightcrafter version 2 [25]) 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 120Hz) to record a user's monoscopic viewpoint. The scene consists of a rotating turntable, which 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 user wearing an HMD. In particular, in this optical bench setup, we can take advantage of a very-low-latency tracker with controlled, repeatable movement.



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

(b) 1 kHz display.

Figure 3: AR registration of a moving object (pyramid). These frames were filmed by a 120 Hz camera through a beam splitter (see Figure 2).

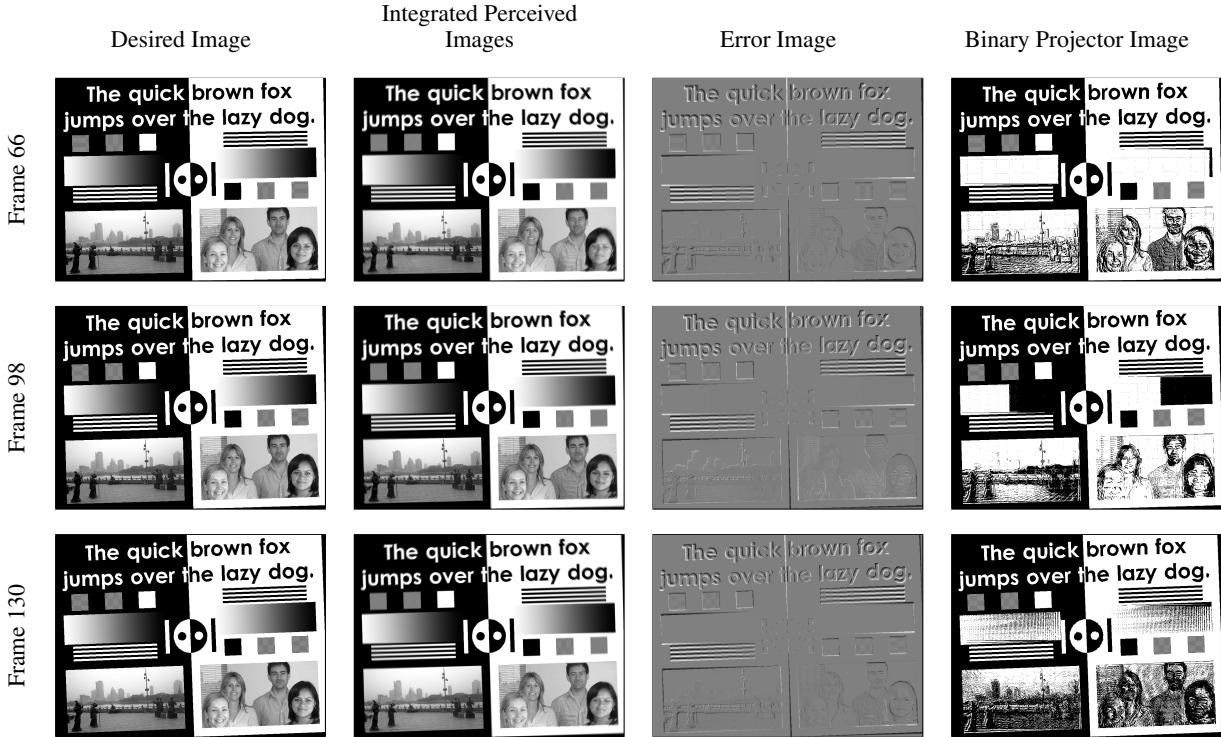


Figure 4: Sample images for a rotating test pattern as used with the experimental projector. The pattern (see Figure 6) rotates at 360°/s to produce the Desired Images. For clarity, a border has been added to each image.

5.1 Experiment 1: Latency

Our first experiment compared the latency of a conventional 60Hz DMD projector (DLP Lightcrafter) with that of the low-latency Experimental Projector. A simple 3-axis cursor was positioned in 3-space on 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 of a degree. 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. Figure 3 shows the results for conventional and experimental projectors as the user rotated the turntable at the maximum rate of 2/3 Hz. As expected, the conventional projector's image lagged noticeably behind its intended location; the experimental projector's cursor stayed in the proper location.

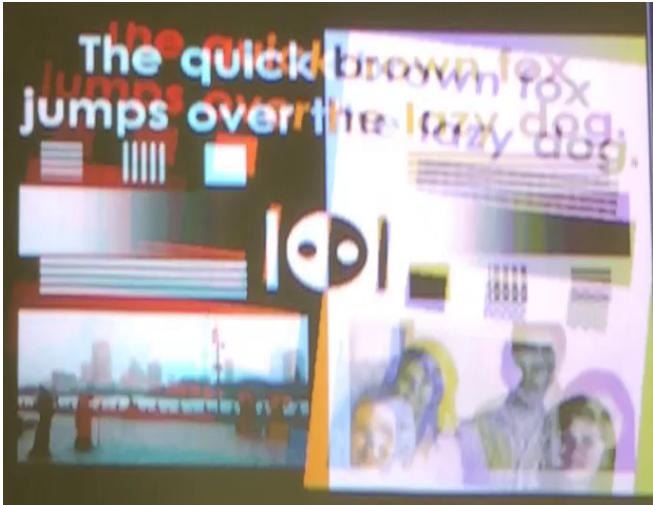
5.2 Experiment 2: Low latency gray-scale imagery using binary image generation

The second experiment consisted of spinning a test pattern (see Figure 6) that contained a combination of text, lines, gradients and pho-

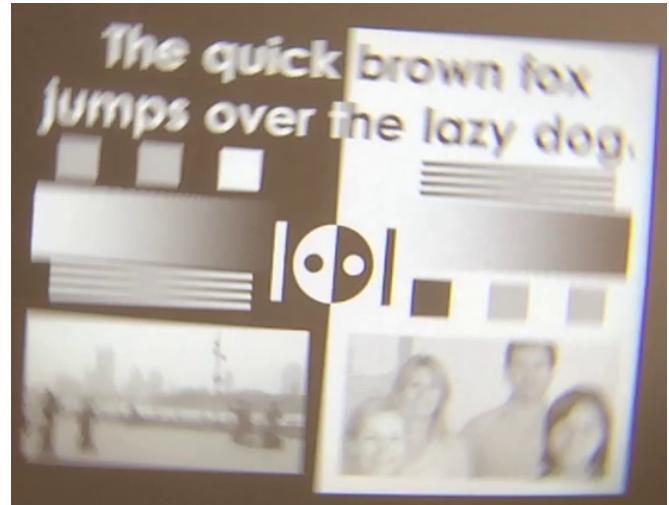
tos. The resulting set of Projected Binary Images was displayed at full speed in a full 360° cycle so the results could be examined visually. As expected, the imagery was rotating smoothly, exhibiting increased motion blur near the edges of the spinning test pattern and very little motion blur near the center. No artifacts were observed. Figure 4 shows a selection of frames from this experiment: Desired Images, Integrated Perceived Images, Error Images, and Binary Projected Images. Figure 5 and the accompanying video shows the dynamic results.

5.3 Experiment 3: AR imagery onto a moving object

Since our experimental projector requires us to pre-load (and therefore pre-compute) the binary images, the real-life motion of the object in this third experiment must be known in advance. Therefore, instead of moving the turntable (and object) by hand, we moved it with a PC-controlled stepper motor, through a predefined series of angular positions, in both directions and at varying speeds. Figure 8 shows one such motion profile covering the experiment pictured in Figure 7. The sequence lasted 1.92 s, during which 43690 binary



(a) Projected using a conventional DLP projector. Note that the image is generally sharp for a single color channel, though the color channels are spatially separated.



(b) Projected using experimental projector. Note that the center of the image is sharper while the outside edges are more blurred.

Figure 5: A frame of the rotating pattern from Figure 6 projected onto a flat surface and captured with a 120 Hz camera. The pattern rotates at 360 °/s.



Figure 6: 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.

images were displayed.

It is important to note that this experiment is used to evaluate visual quality, not latency. Note in Figure 7 and in the accompanying video 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 produced visually pleasing results. Rapid updates reduce or eliminate the “swimming” artifacts induced by excessive latency, and the imagery produced 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 legacy frame-based display interfaces that are modelled on 1930s 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 the limitations of the current control hardware. The implementation will be on a high-performance FPGA, with a high degree of parallelism. Since the bandwidth necessary to drive the DMD is very high, the control circuitry must be physically close to the DMD chip. For this prototype, we expect to supply images from a GPU to the controller (in pairs, and with Z to enable the 3D warp [16]) over a conventional video interface. 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.

Simple extension to color image generation, via an approach similar to frame-sequential color, appears to be 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 well-suited for head-worn displays. Finally, we plan to research approaches to achieve a low-latency equivalent to a device-independent interface, analogous to DVI or HDMI for conventional displays. This would be an abstract interface that could 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.



(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 ($\omega \approx 240^\circ/\text{s}.$)

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.

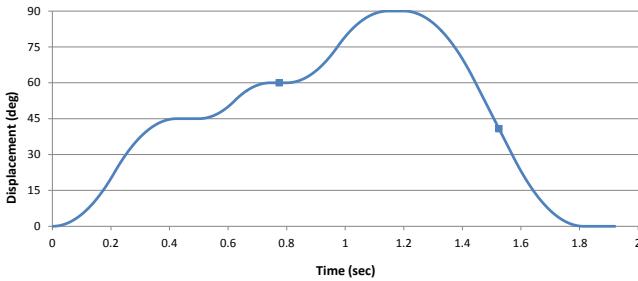


Figure 8: The rotation motion path used. The two indicated dots represent the time instants shown in Figure 7.

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, Dept. of Computer Science, 1995.
- [9] R. E. Bailey, J. J. Arthur III, and S. P. Williams. Latency requirements for head-worn display se/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 *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR’05, pages 265–275, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [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 *Virtual Reality Conference, 2009. VR 2009. IEEE*, 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, Department of Computer Science, 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 Transactions on Graphics*, 28(3):64:1–64:8, 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. of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’13*, pages 2291–2300, New York, NY, USA, 2013. ACM.
- [16] W. R. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D ’97, pages 7–16, New York, NY, USA, 1997. ACM.
- [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 *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST ’12, pages 453–464, New York, NY, USA, 2012. ACM.
- [19] M. Olano, J. D. Cohen, M. R. Mine, and G. Bishop. Combatting rendering latency. In *SI3D*, pages 19–24, 204. ACM, 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 *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’98, pages 179–188, New York, NY, USA, 1998. ACM.
- [21] J. P. Rolland and H. Fuchs. Optical versus video see-through head-mounted displays. In *in Medical Visualization. Presence: Teleoperators and Virtual Environments*, pages 287–309, 2000.
- [22] Texas Instruments. ALP-4.1 controller suite for DLP® Discovery™ 4100. http://www.dlinnovations.com/doc/ALP4_Basic_Datasheet.pdf.
- [23] Texas Instruments. DLP® 0.7 XGA 2xLVDS Type A DMD. <http://www.ti.com/lit/ds/symlink/dlp7000.pdf>, June 2013.
- [24] Texas Instruments. DLP®Discovery™4100 chipset data sheet. <http://www.ti.com/lit/er/dlpu008a/dlpu008a.pdf>, June 2013.
- [25] Texas Instruments. 0.3 WVGA Chipset — DLP Lightcrafter — TI.com. <http://www.ti.com/lit/ds/ti/dlp/03-wvga-chipset-lightcrafter.page?DCMP=dplightcrafter-en&HQS=dplightcrafter>, 2014.
- [26] 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 *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST ’99, pages 1–21, New York, NY, USA, 1999. ACM.
- [27] 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.