

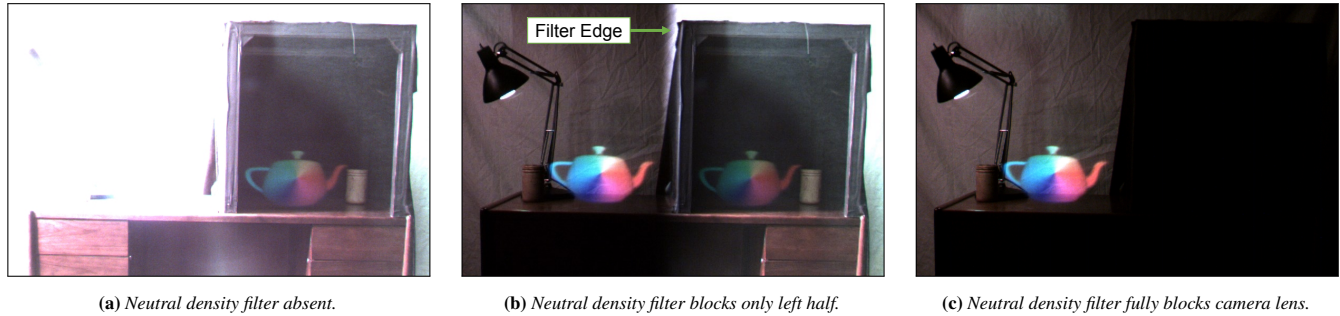
# Scene-Adaptive High Dynamic Range Display for Low Latency Augmented Reality

Peter Lincoln<sup>1</sup> Alex Blate<sup>1</sup> Montek Singh<sup>1</sup> Andrei State<sup>1,2</sup> Mary C. Whitton<sup>1</sup> Turner Whitted<sup>1,3</sup>  
Henry Fuchs<sup>1\*</sup>

<sup>1</sup>UNC-Chapel Hill

<sup>2</sup>InnerOptic Technology, Inc.

<sup>3</sup>NVIDIA Corporation



**Figure 1:** A pair of colored teapots from Demonstration 1. The teapot on the left is displayed 256 times brighter than the one on the right. To capture both simultaneously in a standard 8 bit camera, we optimized the exposure for the righthand darker teapot in condition (a). To demonstrate the HDR display output, a neutral density filter with 98 % attenuation is moved in front of the camera, without changing exposure.

## Abstract

For generalized augmented reality to be feasible, the augmenting elements must be visible in varied environments and under rapidly changing, high dynamic range lighting, from bright sunlight to deep shadows. We present a high dynamic range, optical see-through, augmented reality display that dynamically adjusts the brightness of the virtual imagery to match the current brightness of the real scene. Critical components include the spatial brightness sensor array and the positional brightness image intensity matcher. The color, scene-adaptive HDR display system is based on a high-rate (15 kHz) DMD projector using a high-speed RGB LED illuminator, each color with independent 16 bit intensity control for each binary DMD frame. The critical input to the intensity matching algorithm is the output of an array of high sensitivity light sensors. This paper discusses the implementation of the system and reports performance via still and video demonstrations under a variety of lighting conditions.

**Keywords:** digital micromirror display, high dynamic range, augmented reality, optical see-through

**Concepts:** •Computing methodologies → Mixed / augmented reality; Graphics processors; Virtual reality;

\*email: {plincoln, blate, montek, andrei, whitton, jtw, fuchs}@cs.unc.edu

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.

I3D '17, February 25-27, 2017, San Francisco, CA, USA

ISBN: 978-1-4503-4886-7/17/03...\$15.00

DOI: <http://dx.doi.org/10.1145/3023368.3023379>

## 1 Introduction

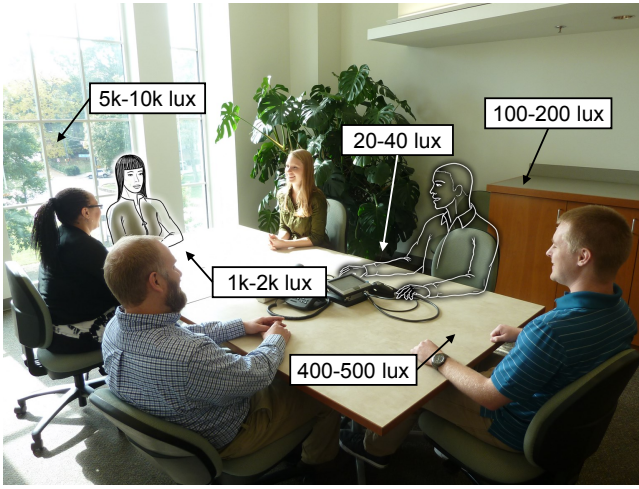
There is no way to deny that the real world is a high dynamic range (HDR) environment (see Figure 2 for an example). Consequently, high dynamic range displays for augmented reality (AR) are necessary because the user needs to comfortably and simultaneously perceive both the real world and virtual imagery in close visual proximity. Furthermore, because users may rapidly move their head and eyes naturally between various parts of their environment, they may rapidly transition between bright and dim areas. We have constructed a display whose dynamic range challenges that of the real world without compromising latency.

Our display cascades a high-frame-rate binary digital micromirror display (DMD) with a high brightness RGB LED. In addition to modulating (spatially and temporally) the DMD, we separately, but directly, drive the LED's brightness so that the final number of levels of light output is roughly equal to the product of those from the two mechanisms. In addition, the two-stage light synthesis approach is compatible with previous work in low latency AR [Lincoln et al. 2016] with minimal motion-to-photon latency; our latency averages 124  $\mu$ s.

Bringing such a display into a high dynamic range real environment obviously requires a mechanism for spatially matching the brightness of the virtual with that of the real. For this we add an array of HDR light sensors coupled to the display. Scene-aware brightness measurements are then used to regulate the brightness of virtual objects.

In developing this system, we had three primary design goals:

1. Implement a display with dynamic range sufficient to match the real world.
2. React to positional HDR changes in physical scene brightness by changing the HDR brightness of the virtual scene.
3. Provide low motion-to-photon latency (on the order of 100  $\mu$ s).



**Figure 2:** A sample scene of people in a room with high dynamic range, captured by a standard camera, annotated with relative brightness measurements from physical points around the scene, and augmented with two simulated people.

## 2 Related Work and Background

Commercially available HDR desktop displays support 10 bit or 12 bit primary color precision. While this is a substantial improvement over the 8 bit primary that has dominated the industry for decades, it does not provide sufficient range to match real lighting environments.

Cascading controllable backlights with spatial light modulators (SLMs) is an established method of achieving high dynamic range. The original implementation [Seetzen et al. 2003] employed a patterned backlight composed of an array of monochrome LEDs placed behind a tri-color LCD. In this case, the resulting image is the product of the low-resolution backlight times the high-resolution SLM. A similar display employs a projector as the back image and an LCD as the front image [Pavlovych and Stuerzlinger 2005]. The update rate, and therefore the latency, for such arrangements is limited by the slowest element, generally the LCD. However, the basic idea of mixing controllable sources with spatial light modulators is also central to the work we report in this paper. It is worth noting that in AR applications, SLMs have also been employed to reduce the dynamic range of real scenes [Wetzstein et al. 2010; Mann et al. 2012], a reminder that extreme dynamic range may not always be desirable.

The requirement of low latency in AR and VR displays has prompted a rethinking of both displays and rendering pipelines. One approach is to place a post-rendering warp stage between the graphics processor and the display [Regan and Pose 1994]. This allows the most recent tracking information to bypass the latency of the rendering engine and update images as they are being passed to the display [Pasman et al. 1999; Itoh et al. 2016]. The low latency requirement has also led to the use of faster devices, such as digital micromirror (DMD) reflective displays, which can be updated at rates of 8 kHz to 32 kHz. Combining these two ideas drastically reduces misregistration between real and virtual components of AR scenes [Zheng et al. 2014]. The pixels of these faster displays are binary (either fully reflective or not reflective) and must be driven by processes such as pulse width or pulse density modulation (PWM or PDM) to yield a grayscale image. As a consequence, several binary frames must be integrated to produce a single grayscale frame. Recent research has demonstrated that up-

dating view parameters at the DMD binary frame rate can produce a 6 bit grayscale image in AR environments without noticeable latency [Lincoln et al. 2016]. A similar scheme has been applied to OLED displays driven in binary mode at a rate of 1700 Hz [Greer et al. 2016].

### 2.1 Pulse Train Modulation and SLMs

The output of each binary frame of a DMD comprises the light reflected by the set of mirrors in the “on” position. Given a light source of *constant* intensity, generating intensities other than zero or one hundred percent requires some form of time-domain modulation, wherein, over a series of binary frames, a pixel is in the “on” position for a time proportional to the desired intensity. Pulse Train Modulation (PTM, or simply Pulse Modulation) encompasses a family of techniques for converting a discrete-time binary signal into an analog signal; this family includes modulation schemes such as Pulse Width Modulation (PWM) and Pulse Density Modulation (PDM). In each scheme, the pattern of “on” and “off” pulses varies, but the ratio of “on” pulses to the sum of “on” and “off” pulses is proportional to the generated intensity.

Since the ratio of pulses is tied to the generated intensity for a pixel, to generate an intensity with  $n$  bits of depth generally requires  $2^n$  pulses—an exponential amount of time for a full integration. Supposing a modulation function  $m(d, s)$ , where  $d$  is the desired intensity and  $s$  is the step index, the generated intensity  $g$  for a single color could be represented as a summation, shown in Equation 1 below, where  $b$  is the brightness of the illuminator. Each iteration of the summation represents one step of the integration cycle.

$$g = \sum_{s=0}^{2^n-1} b \times m(d, s) \quad (1)$$

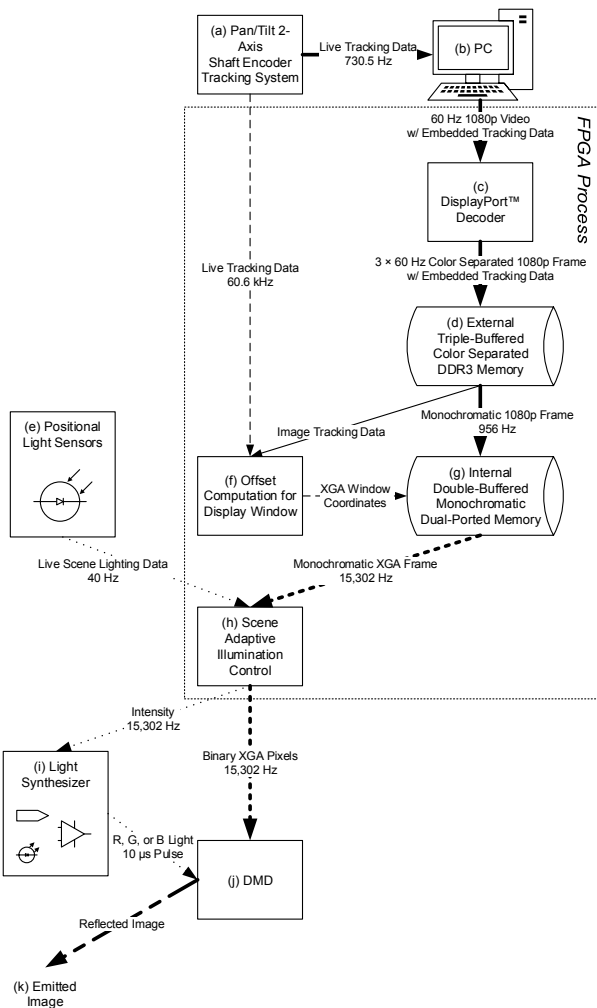
For PWM and PDM, the value of  $b$  is constant, since a constant illuminator is used. A simple form of  $m$  for PWM can be as shown in Equation 2.

$$m_{\text{PWM}}(d, s) = \begin{cases} 1 & \text{if } s < d \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In general, using these modulation schemes on DMDs with constant intensity illuminators can be thought of as executing these repeated summations, where each step ( $s$ ) operates as one binary frame.

### 2.2 Low Latency AR

Because we implement elements of a previous system [Lincoln et al. 2016], it is useful to review it. That system used a combination of conventional PC rendering with a post-processing FPGA (Field Programmable Gate Array) render process to display a 6 bit grayscale virtual scene using a DMD. Physical and virtual scenes were optically combined in the HMD (Head Mounted Display) using a small rear-projection screen, focusing optics, and an optical combiner (half-silvered mirror). To track the user, a pair of high-resolution optical rotary shaft encoders provided very low tracking latency of the pan and tilt axes. The tracking data guided the 3D render process on the PC and the latency compensation algorithm on the FPGA, which realigned the virtual scene back to the physical scene by performing an image-space 2D translation operation. Overall, that system was able to provide an XGA resolution ( $1024 \times 768$ ), approximately  $30^\circ$  horizontal field-of-view augmented scene with an average motion-to-photon latency of about 80  $\mu$ s at a binary frame rate of over 15 kHz. However, that system

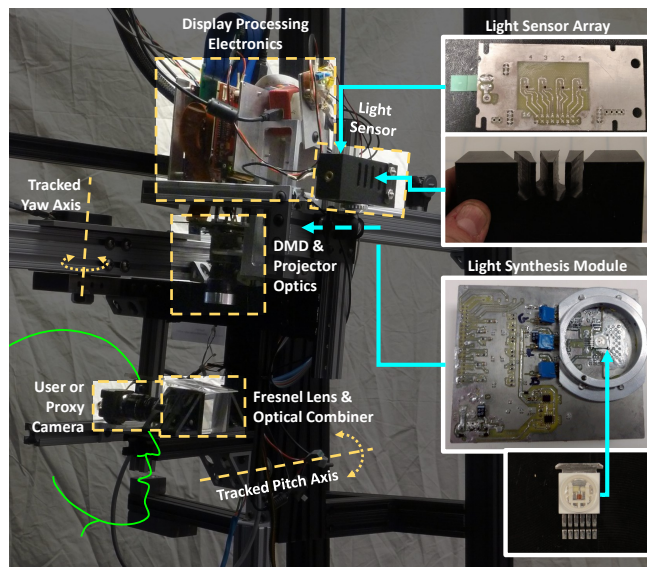


**Figure 3: Rendering pipeline.** Dashed lines indicate the tracking data path ((a), (f), (g), (h), (j), (k)). Thick lines indicate the video content path ((a), (b), (c), (d), (g), (h), (j), (k)). Lastly, dotted lines indicate the illumination path ((e), (h), (i), (j), (k)). Segments of composite features (e.g., thick dashed) indicate overlap (e.g., video and tracking) in the paths.

only provided 6 bit grayscale output, and its modulation scheme, like other  $O(2^n)$  PTM algorithms, required  $2^n - 1$  (where  $n$  is the depth of each color) binary frames to produce that output (about 4 ms). Extended to 16 bit RGB color, its modulation algorithm would require  $3 \times (2^{16} - 1)$  binary frames or 13.1 s using 15 kHz binary frames, which would be useless. Due to the similarity between that system and the system introduced here, an extended discussion of latency is presented in Section 4.2.

### 3 Approach

Our low latency rendering pipeline, presented in Figure 3, consists of three main data paths: tracking, video, and illumination. The video and illumination data paths are most relevant to our color and HDR design goals, but the tracking data path is necessary for latency correction. This process is based, in part, on prior work [Lincoln et al. 2016], which combined standard AR rendering on a



**Figure 4: System Assembly.** Display processing electronics include the TI DLP® Discovery™ 4100 Kit (Virtex®-5 FPGA), a HTG-777 FPGA board (Virtex®-7 FPGA), and a custom DisplayPort™ input and interconnect board. Projector components include the DMD chip and standard lens assembly. The remaining optics include the Fresnel lens and optical combiner (prism). Also present are a light sensor array and an RGB LED light synthesis module. Either a user or a camera can observe the optically combined physical and virtual visual targets.

PC with a post-rendering correction performed by the display hardware.

Our system (Figure 4) is a low latency, color, HDR, monocular, optical see-through, head mounted display for AR. The user views the virtual world through a system of lenses and views both the real and virtual worlds through an optical combiner (prism). Virtual imagery is generated by the PC, processed by the display hardware (an FPGA board), and displayed by a DMD illuminated by a light synthesis module. Tracking is provided by a pair of optical shaft encoders. The system’s awareness of physical scene brightness from the user’s perspective is provided by an array of high dynamic range light sensors.

#### 3.1 Low Latency Color HDR

Generating color HDR imagery on a DMD (or any binary SLM) at rates sufficient for low latency requires a process faster than the exponential methods described in Section 2.1. As an alternative, we directly control the brightness of our illuminator over a period of time via direct digital synthesis (DDS) and spatially select illumination levels.

For  $n$  bits of intensity resolution, we generate  $n$  levels of illumination over a period of  $n$  steps. Each step in the sequence is twice the intensity of the previous step. A sequence of  $n$  binary images is displayed on the DMD, synchronized with the illumination so that each pixel of the display temporally selects the powers of two corresponding to bits in its binary value. Integrating these powers of two over the  $n$  temporal steps yields the desired gray level for each pixel. Repeating the steps for each color produces a full color image with  $n$  bits per color. Expressed in the form of Equation 1,



our scheme follows Equation 3 below.

$$g_{\text{DDS}} = \sum_{s=0}^{n-1} b_{\text{DDS}}(s) \times m_{\text{DDS}}(d, s) \quad (3)$$

Functions  $b_{\text{DDS}}(s)$  and  $m_{\text{DDS}}(d, s)$  can be defined as shown in Equations 4a and 4b below, where  $\text{bit}(d, s)$  returns the 0-based  $s$ -th least significant bit (LSB) of the binary value of  $d$ .

$$b_{\text{DDS}}(s) = 2^s \quad (4a)$$

$$m_{\text{DDS}}(d, s) = \text{bit}(d, s) \quad (4b)$$

Combining Equations 3, 4a, and 4b yields Equation 5 below.

$$g_{\text{DDS}} = \sum_{s=0}^{n-1} 2^s \times \text{bit}(d, s) \quad (5)$$

Unlike the PTM techniques, our process is linear with the number of brightness levels.

The challenge of this approach is generating the  $n$  light levels at short intervals (i.e., a fraction of one binary frame’s duration). Typical methods for generating variable light levels with LEDs use PWM, but for 16 bit intensity generation at 16 kHz would require the LED’s intensity modulator to operate at over 6 GHz, which is infeasible for high-intensity LEDs.

A key enabler for the present display is a custom high-speed, precision, digitally-controlled, direct digital synthesis light module (shown at the lower-right of Figure 4). This module comprises a high-intensity RGB LED and three independent linear current mode driver circuits controlled by 16 bit digital-to-analog converters (DACs). Arbitrary intensities can be generated with turn-on and turn-off times on the order of 300 ns (see Appendix A for details). We use a 10  $\mu$ s pulse during each binary frame. Our measurements show that the full-scale dynamic range of this module is approximately 115 dB; however, the present display does not use the full extent of this dynamic range, in part, to maintain color balance with slightly non-linear behaving hardware.

In order to display RGB HDR with 16 bit/color depth, our FPGA’s display process iterates among each bit of each primary color, writing the bit values to the DMD and emitting a synchronized pulse of appropriate color and brightness from the light synthesizer, updated for each of the 48 binary frames. In theory, one could interleave the colors and bitplanes in any order, but we use a color sequential order due to the FPGA board’s memory size and speed limitations. To maintain a low latency response to the user’s motion, the latency correction operation to realign the virtual imagery to match the physical world operates on a binary frame basis (15 302 Hz); thus the value of  $d$  varies in both time and image space.

### 3.2 Positional Intensity Compensation

If part of the physical scene, as viewed from the user’s perspective, were as bright as the display maximally supports, then the output content needs to be as bright; if the scene is locally dim, then the output must match it there as well.

The PC in our system generates 8 bit/color imagery, without knowledge of the sensed light conditions. To display it for the physical HDR scene, we must scale it appropriately. Supposing one sensed ambient light value represented the entire visible scene, we could



**Figure 5:** Panoramic photo of demonstration setup.

produce the HDR desired value ( $d_{16}$ ) using the standard range value ( $d_8$ ) and a scale factor based on the sensed and maximum supported light values ( $l_{\text{sensed}}$  and  $l_{\text{max}}$ ); this is shown in Equation 6.

$$d_{16} = 2^8 \times \frac{l_{\text{sensed}}}{l_{\text{max}}} \times d_8 \quad (6)$$

For example, if the sensed brightness of the scene were as high as the display is capable of producing, then the 8 bit input data would occupy the most significant eight bits of the 16 bit output. If, instead, the scene were half as bright as maximally supported, then the output data would be half of the maximally bright value.

In order to provide a more localized HDR match, we could replace  $l_{\text{sensed}}$  with a viewpoint (image-space) representation of the sensed light,  $l_{\text{sensed}}(x, y)$ , for each pixel. This requires measuring the lighting conditions of the physical scene with at least as much range as the display’s output. We were unable to obtain a camera with the necessary range of sensitivity, so we use live measurements from a horizontal array of four light sensors (pictured at the top right of Figure 4), each physically located and restricted to seeing approximately one-quarter of the display’s horizontal field-of-view. We currently use a piece-wise linear function (5 segments) and an exponential-decay averaging operation to interpolate sensor measurements in space (horizontally) and to smooth them in time. Each sensor updates its measurement at 40 Hz. In order to provide smooth transitions in time, we perform interpolation updates much faster (1514 Hz, limited by the FPGA’s CPU). While small measured differences in brightness between physical regions are not perceptually disturbing, significant variation in brightness occasionally produces human-perceptible discontinuities in the output along the sensor boundaries. In general, these issues were minor.

## 4 Demonstration of Results

To demonstrate that our system meets our objectives—high dynamic range color display, adaptive brightness, and low latency—we created test setups that allow us to use standard cameras (i.e., 8 bit/color) to capture what a user would see.

### 4.1 Demonstration Fixtures

We designed a physical environment (Figure 5) in which we could create areas of both bright light and dark shadow using a desk lamp (250 W-equivalent LED light bulb), two stand lights (each with two or three 100 W incandescent bulbs, two bright flashlights, and the ceiling fluorescent lights. Lighting conditions varied among demonstrations.

We augment the user’s view of a real desk (about 3 m from the user) with a pair of colorful 3D teapots (Figure 6), positioned so that they



**Figure 6:** A raw  $1920 \times 1080$  GPU screenshot used as input to the illumination algorithm.

exist in scene areas with different lighting conditions. Demonstration 1 uses static teapot images; Demonstrations 2 and 3 use rotating, Phong-shaded teapots. Note that the GPU lighting model is static: in no case is the GPU rendering modified by the real-time lighting measurements.

As a proxy for a user’s eye, we use a Point Grey CM3-U3-13Y3C camera with a Kowa LM6NCM (1/2 in, 6 mm, F/1.2) C-mount lens. The camera is rigidly mounted to the HMD apparatus (Figure 4, lower left). Filming with a standard camera would result in flicker (not perceived by human viewers) due to capturing variable numbers of MSB-intensity binary frames in each camera frame. To avoid this we trigger the camera in phase with the current color synthesis sequence, and we restrict the shutter duration to an integer multiple of synthesis sequence durations. The segments in the accompanying video<sup>1</sup> were recorded at 22.77 Hz for Demonstration 1 and 45.45 Hz for Demonstrations 2 and 3.

**Demonstration 1: Full Dynamic Range** For this demonstration, we optimized the camera exposure for the shadow half of the scene by fully opening the aperture (Figure 1(a)). To show the maximal range of the display without losing detail, we disabled brightness adaptation by disabling the system’s light sensors. We also locked the display’s 16 bit brightness scaling parameters, setting these to utilize both the upper and lower 8 bit subranges in each half of the display; referring to Equation 6, we set the scale factor value of  $l_{\text{sensed}}/l_{\text{max}}$  to 1 for the left half and to  $1/2^8$  for the right half. Since it is not normally possible to capture the full range of the display with a standard camera, we physically placed a neutral density filter in front of the camera so that it covers the bright (left) half of the scene (Figure 1(b)) and the whole image (Figure 1(c)). Note that teapots’ details remain visible in both halves of the scene.

**Demonstration 2: Dynamic Scene Brightness** To demonstrate dynamic range under changing lighting conditions, we varied the lighting across the physical scene while recording. To do this, we 1) moved a pair of bright flashlight beams around the scene (Figure 7) and 2) manipulated the head of a bright desk lamp (Figure 8). Our system’s light sensors detect the changing lighting conditions, and the system responds by modifying the displayed brightness of the virtual scene elements. For recording, the user’s proxy camera’s exposure was optimized for the brightest condition in the demonstration. As a result, when illumination is very low, the teapots can dim to zero.



**Figure 7:** The scene illuminated by bright flashlights. The right teapot (closer to the brighter light) appears brighter than the left.

**Demonstration 3: Low Latency Verification** To verify that our display maintains registration between the virtual and physical worlds, we lit the scene evenly and rotated the HMD back-and-forth by hand by approximately  $15^\circ$  (smaller than our display’s  $30^\circ$  field-of-view, though it keeps both teapots in view) at varying frequencies. A protractor and a metronome helped us generate consistent motion. We captured frames of a checkerboard (Figure 9) and the teapot scene (video) both with latency compensation enabled and with it disabled.

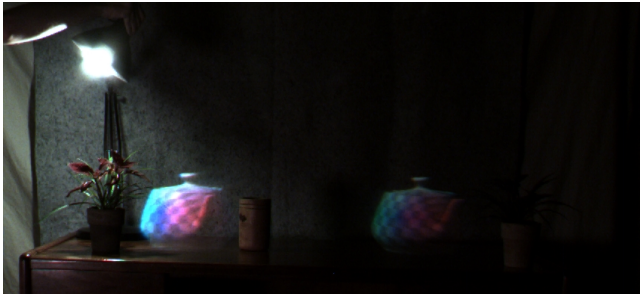
## 4.2 Latency Analysis

We examine end-to-end video latency (the time it takes for a change in appearance generated by the PC to become visible on the display) and motion latency (the time it takes for a viewpoint change to become visible on the display). Video latency affects the color or internal animation appearance of the displayed virtual imagery, while the motion latency affects its registration (alignment) to real scene objects. Causes of latency in the system described here vary in well defined ways from the latencies described of the previous similar system [Lincoln et al. 2016].

The video latency in our new system is higher due to the triple buffering pipeline stage (Figure 3(d)) which stores the color frames in off-chip RAM. Reading and writing to off-chip memory adds about 1 ms to the video latency.

The end-to-end motion-to-photon latency of our new system is not affected by the additional buffering, as the view-direction driven region selection (Figure 3(h)) occurs after the buffering. For every binary frame (15 302 Hz), delay is introduced by waiting to activate the color LED until the DMD’s micromirrors have stabilized after updating the binary image. This new delay is relative to the start of binary pixel transmission. The illuminator used in Lincoln et al. [2016] was constantly on, so its end-to-end latency only required a quarter of the frame to be processed by the DMD before becoming observable. The current system must wait for the entire binary frame to be loaded and all of the mirrors to be flipped and stabilized before it can activate the LED, making the output observable. Based on the DMD command and mirror stabilization protocol specifications, we estimate that this adds  $62 \mu\text{s}$  of latency for the transmission step and  $4.5 \mu\text{s}$  for the mirrors to stabilize, resulting in an estimated average total motion-to-photon latency of about  $124 \mu\text{s}$ .

<sup>1</sup><https://www.youtube.com/watch?v=HZPQUSbVFLY>

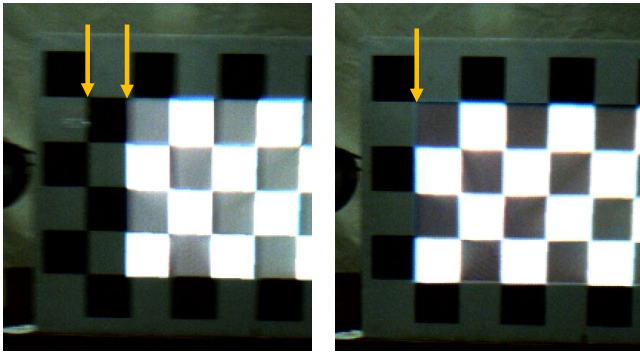


(a) Desk lamp tilted left. The left teapot appears significantly brighter than its right, more distant counterpart.



(b) Desk lamp tilted right. The left teapot is slightly brighter than its right, more distant counterpart.

**Figure 8:** The scene is illuminated by a desk lamp (Demonstration 2). The camera exposure is optimized for the brightest possible condition without over-saturating. Note the variation in brightness of the teapots caused by placement and lighting condition.



(a) Latency compensation disabled. The virtual checkerboard lags by one complete checker (about 12 cm). (b) Latency compensation enabled. The two checkerboards are nearly aligned.

**Figure 9:** The physical and virtual scenes each consist of matched checkerboards (Demonstration 3). In both images, the user’s view is turning right at about  $16^\circ/s$ . Note the significant difference in alignment, caused by latency, between the two images.

## 5 Future Work

**Extending Dynamic Range** In the current system, the illuminating LEDs are turned on for a constant duration for each pulse. Given that the illuminator can turn on and off in under  $1\ \mu s$ , we could activate it for shorter times, producing a higher bit depth at darker intensities, perhaps at up to 20 bit/color.

**Improving Adaptive Scene Sensing and Rendering** Building a higher resolution, 2D array of sensors with higher sampling rates and putting the array in the optical path would improve measurement of the ambient scene brightness, reducing interpolation artifacts occurring at sensor boundaries. An ability to collect full scene illumination parameters would enable relighting the virtual objects to match the real scene.

**Reducing Latency** Using a custom ASIC or an FPGA with larger on-chip memory would eliminate the video latency due to the off-chip memory reading/writing.

## 6 Conclusions

The work presented here represents a proof-of-concept. We have demonstrated the value of adapting the display brightness of the augmenting virtual elements to the lighting of the real scene. In particular, we adapt the display of the augmentation to the brightness of the ambient light to prevent the augmenting objects from “disappearing” in very bright light and from overwhelming very dark portions of the real scene. In addition, due to the speed of the components in the system, we can, in real-time, compensate for user head motion and dynamic lighting.

Initially, we were motivated to develop a low latency, color, optical see-through, head-mounted display because the world is colorful and dynamic. However, the naïve approach to doing color added unacceptable amounts of latency, so we changed direction and adopted the novel solution described here. Following the useful principle of overbuilding for flexibility in an experimental prototype system, the system as designed and built used components allowing 16 bit color—enabling a high dynamic range display. Once we had that capability at the display, it remained to develop the system as a whole, in particular the module sensing the brightness of the real scene.

There is substantial work to be done before a system such as described here can be deployed. In addition to the items noted in Section 5, the system needs size and weight reduction, improved algorithms, and sufficiently fast and accurate tracking of the users head. The benefit has been demonstrated, but there are substantial research and development challenges remaining.

## Acknowledgements

The authors wish to thank Kurtis Keller and Jim Mahaney for engineering advice and support and to thank Fay Alexander, Steve Partin, Brett Piper, and Hope Woodhouse for sitting in for photos.

This research was supported in part by NSF grant CHS IIS-1423059: Minimal-Latency Tracking and Display for Head-Worn Augmented Reality Systems. It was also supported by the BeingTogether Centre, a collaboration between Nanyang Technological University (NTU) Singapore and University of North Carolina (UNC) at Chapel Hill. The BeingTogether Centre is supported by the National Research Foundation, Prime Minister’s Office, Singapore, under its International Research Centres in Singapore Funding Initiative.



## References

- GREER, T., SPJUT, J., LUEBKE, D., AND WHITTED, T. 2016. 8-3: Hybrid modulation for near zero display latency. *SID Symposium Digest of Technical Papers* 47, 1, 76–78.
- HOROWITZ, P., AND HILL, W. 2015. *The Art of Electronics*, 3rd ed. Cambridge University Press, April.
- ITOH, Y., ORLOSKY, J., HUBER, M., KIYOKAWA, K., AND KLINKER, G. 2016. OST Rift: Temporally consistent augmented reality with a consumer optical see-through head-mounted display. In *2016 IEEE Virtual Reality (VR)*, 189–190.
- LINCOLN, P., BLATE, A., SINGH, M., WHITTED, T., STATE, A., LASTRA, A., AND FUCHS, H. 2016. From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. *IEEE Transactions on Visualization and Computer Graphics* 22, 4 (April), 1367–1376.
- MANN, S., LO, R. C. H., OVTCHAROV, K., GU, S., DAI, D., NGAN, C., AND AI, T. 2012. Realtime HDR (high dynamic range) video for eyetap wearable computers, FPGA-based seeing aids, and glasseyes (EyeTaps). In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1–6.
- PASMAN, W., VAN DER SCHAAF, A., LAGENDIJK, R., AND JANSEN, F. 1999. Low latency rendering for mobile augmented reality. In *Proceedings of the ASCII*, vol. 99, 15–17.
- PAVLOVYCH, A., AND STUERZLINGER, W. 2005. A high-dynamic range projection system. *Proc. SPIE* 5969, 59692O–59692O–8.
- REGAN, M., AND POSE, R. 1994. Priority rendering with a virtual reality address recalculation pipeline. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '94, 155–162.
- SEETZEN, H., WHITEHEAD, L. A., AND WARD, G. 2003. 54.2: A high dynamic range display using low and high resolution modulators. *SID Symposium Digest of Technical Papers* 34, 1, 1450–1453.
- WETZSTEIN, G., HEIDRICH, W., AND LUEBKE, D. 2010. Optical Image Processing Using Light Modulation Displays. *Computer Graphics Forum* 29, 1934–1944.
- ZHENG, F., WHITTED, T., LASTRA, A., LINCOLN, P., STATE, A., MAIMONE, A., AND FUCHS, H. 2014. Minimizing latency for augmented reality displays: Frames considered harmful. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, 195–200.

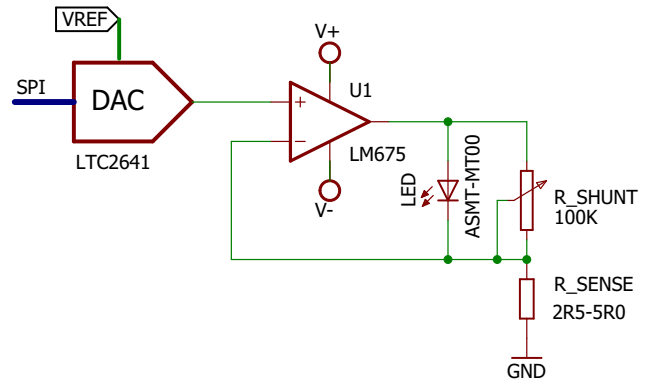


Figure 10: LED Driver Schematic.

## A LED Illuminator: Theory of Operation

The light output of an LED is (almost linearly) proportional to the current through the LED. The circuit of the illuminator is a high-speed, digitally programmable current source, akin to the classic op amp current source [Horowitz and Hill 2015] that leverages this proportionality. A simplified schematic of one channel of the illuminator is shown in Figure 10. A digital-to-analog converter (DAC), controlled by the display driver FPGA, impresses a voltage,  $V_{SET}$ , upon the non-inverting input of the op amp U1. Current flows from the output of the op amp, through the LED, and then through a sense resistor (R\_SENSE) to ground. Since the voltage drop across R\_SENSE is proportional to the current through the diode, this voltage drop is impressed upon the inverting input of the op amp. With this feedback network, the op amp drives its output such that the voltage drop across R\_SENSE is equal to  $V_{SET}$ .

In practice, the op amp's input offset voltage (on the order of 1 mV in our case) and the DAC's zero-code offset (on the order of a few tens of microvolts) cause a small current output even when  $V_{SET}$  is equal to zero, which would cause the LED to illuminate at a significant intensity. There are a number of ways to compensate for these offsets and cause the LED to turn completely off when  $V_{SET} = 0$ . Instead, we chose to leverage these offsets to improve performance: to prevent the diode from illuminating in this condition, resistor R\_SHUNT is placed in parallel with the diode to provide an alternate current path. The value of R\_SHUNT is tuned such that when  $V_{SET} = 0$ , the diode is forward-biased but emits essentially no light. By holding the diode in this condition, we effectively eliminate the LED's junction capacitance, permitting the 300 ns turn-on and turn-off times. Additionally, the typical voltage swing at the LED's anode, required to transition to full-scale intensity, is reduced from several volts to less than one volt (this varies by diode).