Occlusion-Capable Optical See-Through Augmented Reality Display using a single Spatial Light Modulator

Brooke Krajancich

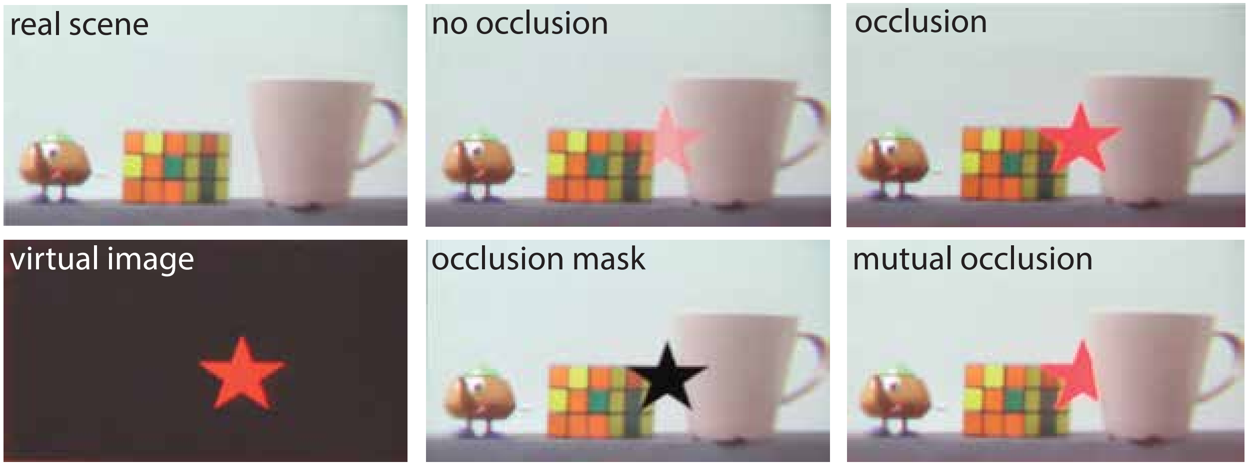

Fig. 1. Result of single SLM occlusion strategy on rendering a virtual image of a single color. Column 1: shows the real-scene as seen by the viewer and the virtual image displayed. Column 2: shows an emulation of a commercially available OST-AR set-up with no occlusion support and the occlusion mask generated in this work. Column 3: shows the real-world light blocking ability of our technique, both full and mutual object occlusion demonstrations.

Abstract—We propose an occlusion technique using a single spatial light modulator (SLM) for optical see-through augmented reality (OST-AR). Occlusion is a powerful visual cue that is important for depth perception and realism in OST-AR. Without proper occlusion, virtual objects rendered in front of bright real-world objects appear semi-transparent and non-realistic. The most successful previous work to overcome this has been to use an additional SLM to block incoming real-world light pixel-by-pixel. However, the extra hardware required does not lend itself easily to the compactness required for head-mounted displays. In this work, we propose an occlusion-capable OST-AR system, using a single SLM to both render the virtual image and block real-world light. We present results from simulations and build a proof-of-concept fixed-focus prototype using off-the-shelf components. We demonstrate occlusion capability, with small trade-offs in color fidelity.

Index Terms—Occlusion, see-through display, augmented reality

1 Introduction

Augmented Reality (AR) is being increasingly viewed as the next transformative technology, enabling new ways of accessing and perceiving digital information essential to our daily life. Optical see-through augmented reality (OST-AR) shows particular promise for widespread adoption, utlizing an optical combiner to merge virtual images with a real scene [13]. However, presents the challenge of correctly rendering mutual occlusion of real and virtual objects in space.

Occlusion is the light-blocking phenomenon that occurs when an object is between another object and a viewer, and acts as the most important depth cue for humans [10, 11]. Hence, when a virtual object is to be rendered in front of a real object, the virtual image must block light coming from the real-world. This is challenging, particularly when the real-world scene is bright, since typical optical combiners can only add light to the scene composition, not remove it.

Without proper occlusion, virtual objects rendered in OST-AR appear semi-transparent and less realistic (see Figure 2). This could even be dangerous, as it may induce misjudgments on users in mission-critical tasks [3]. Improving occlusion capability of OST-AR is important for improving the perceived physical accuracy of these systems, and thereby, improving effectiveness.

2 Previous work

Initial demonstrations of occlusion-capable OST-AR operated by selectively blocking rays from the see-through scene without focusing them. This was often implemented by selectively modifying the reflective properties of physical objects or by passing the light from the real scene through a single layer spatial light modulators (SLM) placed directly near the eye [7, 14] However, this concept assumes the eye is a pinhole aperture. In fact, it is practically impossible to use a single-layer SLM to block all the rays seen by the eye from an object without blocking the rays from other surrounding objects. Moving to multi-layer SLM's showed improvement [12, 15] but was still subject to major limitations such as the significantly degraded see-through view, limited accuracy of the occlusion mask, and low light efficiency.

The most successful demonstrations of occlusion capable OST-AR have been those that provide per-pixel occlusion by inserting a SLM in the real-world optical path (before the optical combiner) to selectively block real-world light pixel-by-pixel. Several fixed-focus displays using this concept have been demonstrated using LoCoS and LCD tech-

- *Brooke Krajancich is with Stanford University. E-mail: brookek@stanford.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

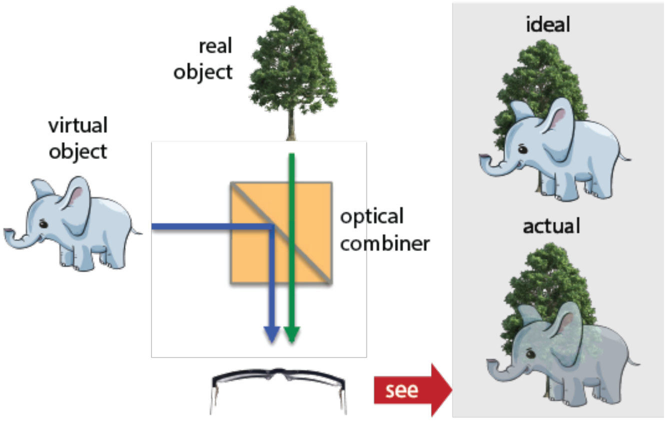

Fig. 2. Illustration showing combination of real-world and virtual images in typical OST-AR configurations and the lack of proper occlusion produced.

nology, both in bench-top [1,4,16] and early prototype head-mounted displays [8,9]. Although good occlusion capability was often shown, the alignment of multiple SLM's often made set-ups bulky or required freeform lenses which are often expensive and challenging to design and fabricate. Earlier this year, this concept was adapted in combination with controlled movement of the blocking SLM to demonstrate varifocal occlusion-capable OST-AR [5]. This showed good success in emulating occlusion for objects at a large range of distances from the viewer, however, required a system of eight lenses along with the two SLM's. No demonstrations to date show promise for working towards the compact form-factor required for commercially-relevant head-mounted displays.

3 This Work

While previous work has utilized separate SLM's for displaying virtual content and masking real-world light, by using a digital micro-mirror device (DMD), it may be possible to use the same SLM, for both digital image formation and real-world occlusion. Doing so could decrease optical complexity and reduce form-factor, which is important for head-mounted display applications.

This course project forms a continuation of a PhD rotation project, working with Nitish Padmanaban, Gordon Wetzstein and the Stanford Computational Imaging Group.

3.1 Digital micro-mirror devices (DMDs)

A DMD is a type of SLM that consists of a dense two-dimensional array of small micro-mechanical mirrors. These mirrors can be individually rotated at high speeds (up to 4000 Hz) ±10 - 12° to an ON or OFF state. In the ON state, light from an external source (such as an LED) is reflected towards a viewer, while in the OFF state light is directed elsewhere, typically towards a light dump. Brightness at each pixel can be adjusted by time-modulating these mirror flips between the ON and OFF states. Displaying an RGB image is typically achieved by dividing the image up into its red, green and blue channels and sequentially rendering each frame on the DMD synchronously with the corresponding LED color. As long as the combination of the three channels are displayed faster than the human flicker fusion threshold (approximately 60 Hz [2]), the human brain will blend the colors together to give the target RGB image.

3.2 A novel adaption

With the goal of using the same SLM to display the virtual image and block real-world light, we propose removing the light dump from a DMD and re-directing the optical path towards a real scene, as illustrated in Figure 3.

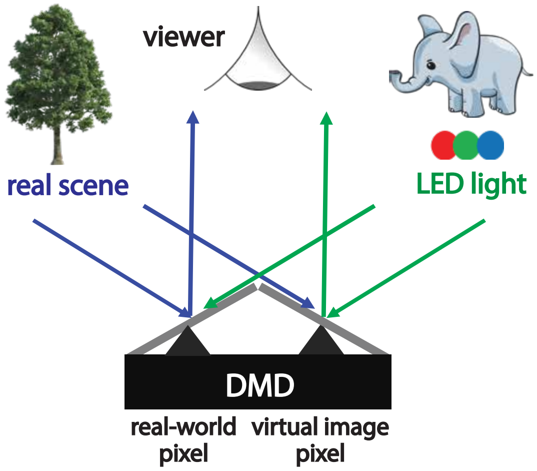

Fig. 3. Illustration of ON and OFF DMD pixels, directing virtual image and real-world light towards the viewer, respectively.

For a monochrome virtual images, this is simple since the LED can be set to a single color, the DMD pixels corresponding to the image turned ON and the remaining pixels turned OFF (to display the real scene). This is shown in Figure 1.

However, displaying RGB virtual images is more complex. To illustrate this, take for example rendering a virtual image with a green, blue and red pixel over green tree present in the real-world (illustrated in Figure 4). All of these colors are single channel and thus for perfect color fidelity, each pixel should not be exposed to the other color channels. However, the LED illuminates the entire DMD and the removal of the light dump means that in each of the LED color states, we must pick whether to flip the pixel to direct the LED or real-world. For example, when the LED flashes red, we can turn the red pixel towards it and the green pixel towards the real-world (since the tree is a good approximation of the desired color), but there is no ideal direction in which to direct the blue pixel, leading to color infidelity. We can then formulate an optimization problem to determine the series of DMD states and corresponding LED colors that minimizes this effect.

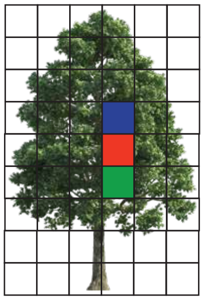

Fig. 4. Illustration of a difficult image composition to render with our technique.

4 Results

4.1 Bench-top prototype

In this work we designed, built and tested a proof-of-concept bench-top configuration using off-the-shelf components. Figure 5 shows a schematic of this design and Figure 6 shows photographs of the constructed set-up. Importantly, the LED and scene are incident on the DMD from ±24° from the normal, such that the mirrors direct light along the normal direction towards the camera. Two doublet lenses with focus lengths of 25mm and 50mm are used to de-magnify and focus the scene on to the DMD and collimate and magnify the DMD image onto the camera sensor, respectively. The relay lens is used to translate the DMD image such that the high powered doublet can fit geometrically.

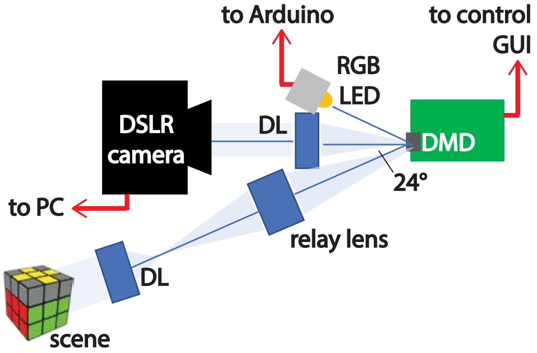

Fig. 5. Schematic of bench-top prototype demonstration. D. doublet lens.

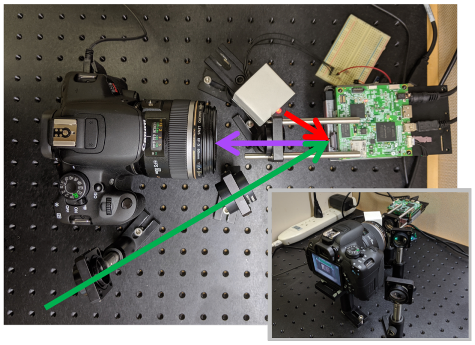

Fig. 6. Birds-eye photograph of the constructed bench-top prototype. The red and green arrows show the optical paths from the LED and real-world scene, respectively, and the purple arrows shows the path of light reflected by the DMD to the viewer (camera). The inset shows a view of the set-up from behind the camera.

The LED is connected to an Arduino, which is used to control the brightness of the red, green and blue channels at any time. The DMD mirror flips are programmed using a GUI interface provided by the manufacturer (Texus Instruments) by uploading a series of up to 96 1-bit BMP frames. The Arduino is connected to the input trigger of the DMD in order to synchronise LED and DMD states.

4.2 Image generation workflow

Determining the optimum DMD and LED states to display a particular virtual image depends upon the scene and position in which it is to be rendered. Thus, firstly, every DMD mirror is switched to transmit the real-world and a background calibration image is taken. A target image composition is created by importing this into MATLAB and overlaying the virtual image in the desired position. At this point, if real-objects in the scene are to appear in front of the virtual image, Adobe Illustrator can be used to erase that part of the virtual image to simulate depth map input and pixel-by-pixel selective rendering.

The target image is then passed to the optimization algorithm which produces a matrix of DMD states and and RGB LED values. This algorithm was developed by another student in previous work and thus is considered beyond the scope of this project. A short summary can be found in Appendix 7.1. Since the DMD GUI can hold a maximum of 96 DMD frames, the number of gray levels was taken to be 32 (32×3 = 96 for all channels combined).

Converting the DMD matrix into a sequence of 96 1-bit BMP files, these images are uploaded to the DMD GUI. Converting the LED states to 8-bit values, this sequence of 96 RGB values is passed to the Arduino as a .h file. Finally, a short Arduino program is used to synchronously trigger the corresponding LED and DMD state at 4000Hz, giving a frame rate of ~ 46 Hz. This is slightly under the human flicker fusion threshold, but is not problematic since the final image composition is captured by a DLSR camera.

A graphical illustration of this workflow is included in Appendix 7.2.

4.3 Preliminary demonstrations

Figure 7 shows two simulation and demonstration results, rendering RGB images (insets) of a cat and the Pokmon character, Pikachu.

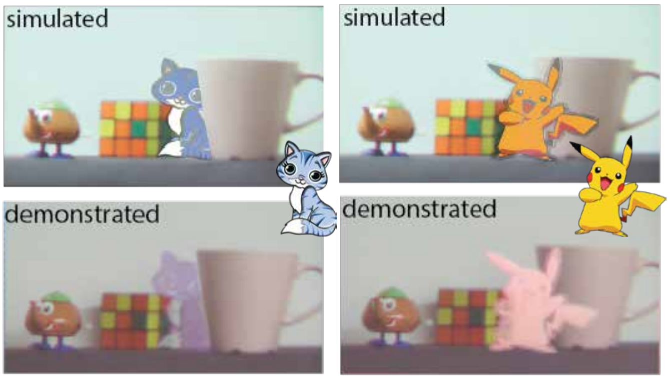

Fig. 7. Simulated and demonstrated renderings of two virtual images by our technique. The original images are overlaid to the right side of the result images.

The simulation of the cat indicates that the set-up is unable to replicate the light blue in the original image. However, the image would look feasible for a viewer who had not previously seen the original. The demonstrated image shows good correspondence with the simulation, but is more washed out in color. More investigation is required to determine the cause of this, but it suspected that the RGB levels of the LED are not properly balanced and their maximum intensity not matched to that of the real-world, as the simulation assumes.

The simulation of Pikachu also shows a similar behaviour, this time a noticeably more orange rendering which the red on the cheeks faded in intensity. Looking closely it can be seen that multiple border artifacts have been generated. More investigation and tuning of the optimization algorithm is needed to find the cause of this. The demonstrated image shows general correspondence with the simulation, but is again, more washed out in color, and this time fails to fully occlude the rubix cube. Again, LED issues are suspected.

5 Observations and on-going work

From the above results we can see that the DMD struggles when the real-world is bright, since it needs to 'remove' a significant amount of light. In order to remove light, it pixels need to spend time facing away from the real-world, but with the LED off. Since true black would require it to be off for the entire time, this presents an inherent trade-off. For the same reason, colors that are complementary to each other (such as the yellow of Pikachu and the green in the Rubix cube behind) are also difficult. The end result is a loss in color fidelity in exchange for light removal. On-going work will look at exploring gaze-contigent update to minimise color disparity in areas in which the viewer is directly looking. Since humans are more sensitive to brightness, rather than hue, it may be possible to adapt the optimization algorithm to weight this and/or integrate color weights based on perceptual significance.

Obvious white balance, color saturation and poor optical resolution effects will also be investigated.

REFERENCES

[1] O. Cakmakci and a. J. P. Rolland. A compact optical see-through head-worn display with occlusion support. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 16–25, Nov. 2004. doi: 10.1109/ISMAR.2004.2

[2] A. Eisen-Enosh, N. Farah, Z. Burgansky-Eliash, U. Polat, and Y. Mandel. Evaluation of Critical Flicker-Fusion Frequency Measurement Methods for the Investigation of Visual Temporal Resolution. *Scientific Reports*, 7(1):15621, Nov. 2017. doi: 10.1038/s41598-017-15034-z

[3] H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer. Augmented reality visualization for laparoscopic surgery. In W. M. Wells, A. Colchester, and S. Delp, eds., *Medical Image Computing and Computer-Assisted Intervention MICCAI98*, Lecture Notes in Computer Science, pp. 934–943. Springer Berlin Heidelberg, 1998.

[4] C. Gao, Y. Lin, and H. Hua. Occlusion capable optical see-through head-mounted display using freeform optics. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 281–282, Nov. 2012. doi: 10.1109/ISMAR.2012.6402574

[5] T. Hamasaki and Y. Itoh. Varifocal Occlusion for Optical See-Through Head-Mounted Displays using a Slide Occlusion Mask. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1961–1969, May 2019. doi: 10.1109/TVCG.2019.2899249

[6] N.-D. Ho. *Nonnegative matrix factorization algorithms and applications*. PhD Thesis, 2008.

[7] Hong Hua, Chunyu Gao, L. D. Brown, N. Ahuja, and J. P. Rolland. A testbed for precise registration, natural occlusion and interaction in an augmented environment using a head-mounted projective display (HMPD). In *Proceedings IEEE Virtual Reality 2002*, pp. 81–89, Mar. 2002. doi: 10.1109/VR.2002.996508

[8] K. Kiyokawa, M. Billinghurst, B. Campbell, and E. Woods. An occlusion capable optical see-through head mount display for supporting co-located collaboration. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 133–141, Oct. 2003. doi: 10.1109/ISMAR.2003.1240696

[9] K. Kiyokawa, Y. Kurata, and H. Ohno. An optical see-through display for mutual occlusion with a real-time stereovision system. *Computers & Graphics*, 25(5):765–779, Oct. 2001. doi: 10.1016/S0097-8493(01)00119-4

[10] E. Kruijff, J. E. Swan, and S. Feiner. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pp. 3–12, Oct. 2010. doi: 10.1109/ISMAR.2010.5643530

[11] M. A. Livingston, J. E. Swan, J. L. Gabbard, T. H. Hollerer, D. Hix, S. J. Julier, Y. Baillot, and D. Brown. Resolving multiple occluded layers in augmented reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 56–65, Oct. 2003. doi: 10.1109/ISMAR.2003.1240688

[12] A. Maimone and H. Fuchs. Computational augmented reality eyeglasses. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 29–38, Oct. 2013. doi: 10.1109/ISMAR.2013.6671761

[13] J. Rolland and H. Fuchs. Optical Versus Video See-Through Head-Mounted Displays in Medical Visualization. *Presence*, 9:287–309, June 2000. doi: 10.1162/105474600566808

[14] E. W. Tatham. Getting the Best of Both Real and Virtual Worlds, Sept. 1999.

[15] G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar. Tensor displays: compressive light field synthesis using multilayer displays with directional backlighting. *ACM Transactions on Graphics*, 31:1–11, July 2012. doi: 10.1145/2185520.2335431

[16] A. Wilson and H. Hua. Design and prototype of an augmented reality display with per-pixel mutual occlusion capability. *Optics Express*, 25(24):30539–30549, Nov. 2017. doi: 10.1364/OE.25.030539

6 APPENDICES

6.1 The optimization problem

We can define the DMD pixel states as $\mathbf{D} \in \left\{0, g^{-1}\right\}^{wh \times 3g}$, with the first dimension for pixels and the seconds for time (g gray levels times 3 channels). The LED states can be modelled as $\mathbf{L} \in [0,1]^{3g \times 3}$, assuming that we have an RGB LED that can vary in brightness. The final target image and incident real-world light is given by $\mathbf{T}, \mathbf{R} \in [0,1]^{wh \times 3}$. We model the cross-talk between the LED and real-world light by:

$$\frac{1}{3}\mathbf{R} \odot (\mathbf{D}\mathbf{1}_{3g \times 3}),$$

This gives us an image formation model of:

$$\mathbf{T} = \mathbf{D}\mathbf{L} + \mathbf{R} - \frac{1}{3}\mathbf{R} \odot (\mathbf{D}\mathbf{1}_{3g \times 3}), \tag{1}$$

Define $\mathbf{T}' = \mathbf{T} - \mathbf{R}$ and $\mathbf{R}' = \mathbf{R}/3$, then the loss function for this problem is:

$$J(\mathbf{D}, \mathbf{L}) = \frac{1}{2}\|\mathbf{T}' - \mathbf{D}\mathbf{L} + \mathbf{R}' \odot (\mathbf{D}\mathbf{1}_{3g \times 3})\|_{\mathrm{F}}^{2}. \tag{2}$$

We can then use the rank-one residue update method [6] to define the following residue matrix:

$$\mathbf{T}'_t \triangleq \mathbf{T}' - \sum_{i \neq t} \left(\mathbf{d}_i {\ell_i}^{\mathrm{T}} - \mathbf{R}' \odot (\mathbf{d}_i \mathbf{1}_3{}^{\mathrm{T}})\right) \qquad t = 1 \ldots 3g, \tag{3}$$

where $\mathbf{d}_i$ and ${\ell_i}^{\mathrm{T}}$ are the ith column and row of $\mathbf{D}$ and $\mathbf{L}$, respectively, we get the set of loss functions:

$$J_t(\mathbf{D}, \mathbf{L}) = \left\|\mathbf{T}'_t - \mathbf{d}_t {\ell_t}^{\mathrm{T}} + \mathbf{R}' \odot (\mathbf{d}_t \mathbf{1}_3{}^{\mathrm{T}})\right\|_{\mathrm{F}}^{2} \qquad t = 1 \ldots 3g. \tag{4}$$

The optimal update for $\mathbf{d}_t$ is then found to be:

$$d^{*}_{t,i} \leftarrow \begin{cases} 1 & \text{if } x_{t,i} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{where} \tag{5}$$

$$\mathbf{x}_t = 2(\mathbf{W} \odot \mathbf{T}'_t \odot (\mathbf{1}_{wh}{\ell_t}^{\mathrm{T}} - \mathbf{R}'))\mathbf{1}_3 - \left(\mathbf{W} \odot (\mathbf{1}_{wh}{\ell_t}^{\mathrm{T}} - \mathbf{R}')^2\right)\mathbf{1}_3, \tag{6}$$

where $\mathbf{W}$ is a weighting matrix for which pixels are most important and $(\cdot)^2$ is elementwise. Define two quantities:

$$\mathbf{y}_t = {\mathbf{d}_t}^{\mathrm{T}}(\mathbf{W} \odot (\mathbf{T}'_t + (\mathbf{R}' \odot (\mathbf{d}_t \mathbf{1}_3{}^{\mathrm{T}})))), \quad \text{and} \quad \mathbf{z}_t = {\mathbf{d}_t}^{\mathrm{T}}(\mathbf{W} \odot \mathbf{d}_t \mathbf{1}_3{}^{\mathrm{T}}). \tag{7}$$

Then the optimal LED update is given by:

$$\ell_{t,i} \leftarrow \min\left\{\frac{[y_{t,i}]_+}{z_{t,i}}, 1\right\}. \tag{8}$$

6.2 Workflow schematic

See next page.

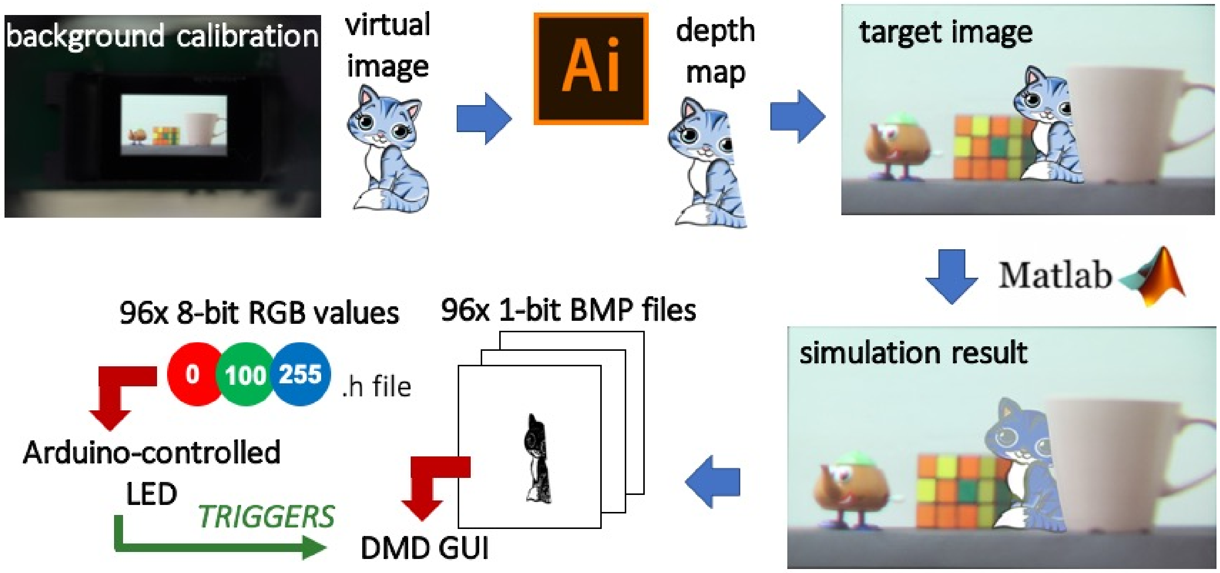

Fig. 8. Image generation workflow. First an image of the real-scene is taken and Adobe Illustrator used to place a chosen virtual image within the scene. The optimization algorithm is then run on this target image in MATLAB, and a simulation of a best-case-scenario generated. Converting the optimum DMD states into 1-bit BMP files, these are loaded onto the DMD via GUI software. The optimum LED states are fed to the RGB LED via an Aduino, which synchronously triggers the DMD to change frame.