Holocurtains: Programming Light Curtains via Binary Holography

Supplementary Materials

1. Overview

The supplementary materials include this document, a website, and code.

This document elaborates on Fourier-transform holography, the wavefront propagation operator, the calibration procedure, and our usage of existing assets. References to the main paper are highlighted in blue. Video results of our light curtains can also be viewed through our supplementary website *index.html*.

2. A Short Introduction to Fourier-Transform Holography

In our work, we leverage a concept known as Fourier-transform holography. It is well known in the wave optics literature that under coherent monochromatic illumination, an ideal lens forms the 2D Fourier transform of the wavefront at its back focal plane [4]. More concretely, let $U_d(x, y)$ be the wavefront at the plane a distance d in front of the lens. Then, the wavefront at the back focal plane will be given by:

$$U(s,t) = \frac{e^{i\frac{k}{2f}\left(1-\frac{d}{f}\right)\left(s^2+t^2\right)}}{i\lambda f} \iint U_d(x,y) e^{-i\frac{2\pi}{\lambda f}(xs+yt)} \mathrm{d}x\mathrm{d}y \tag{1}$$

$$= \frac{e^{i\frac{k}{2f}\left(1-\frac{d}{f}\right)\left(s^2+t^2\right)}}{j\lambda f} F_d\left(\frac{s}{\lambda f}, \frac{t}{\lambda f}\right) \tag{2}$$

where f is the focal length of the lens, F_d is the Fourier transform of U_d, λ is the wavelength of the illumination, and k is the corresponding wave number $k = \frac{2\pi}{\lambda}$. A camera used to capture this image plane will then simply capture the squared magnitude of a rescaled 2D Fourier Transform of the original signal:

$$I(s,t) = |U(s,t)|^2 \propto \left| F_d\left(\frac{s}{\lambda f}, \frac{t}{\lambda f}\right)\right|^2 \tag{3}$$

A system based on Fourier-transform holography uses a spatial light modulator (SLM) and a coherent light source to form the wavefront $U_d(x, y)$. The resulting Fourier transform at the back focal plane of the lens is then utilized for the desired application.

In our work, we leverage this Fourier transform relationship to redistribute light. To illustrate this concept, consider the case where the desired pattern is just a single point:

$$I(s,t) = \delta(s,t) \tag{4}$$

The corresponding wavefront that would generate this pattern would be given by:

$$U_d(x,y) = 1 \tag{5}$$

The entire wavefront at the plane d away from the lens will be refocused to form the point. In other words, the system is perfectly light efficient when forming this point.

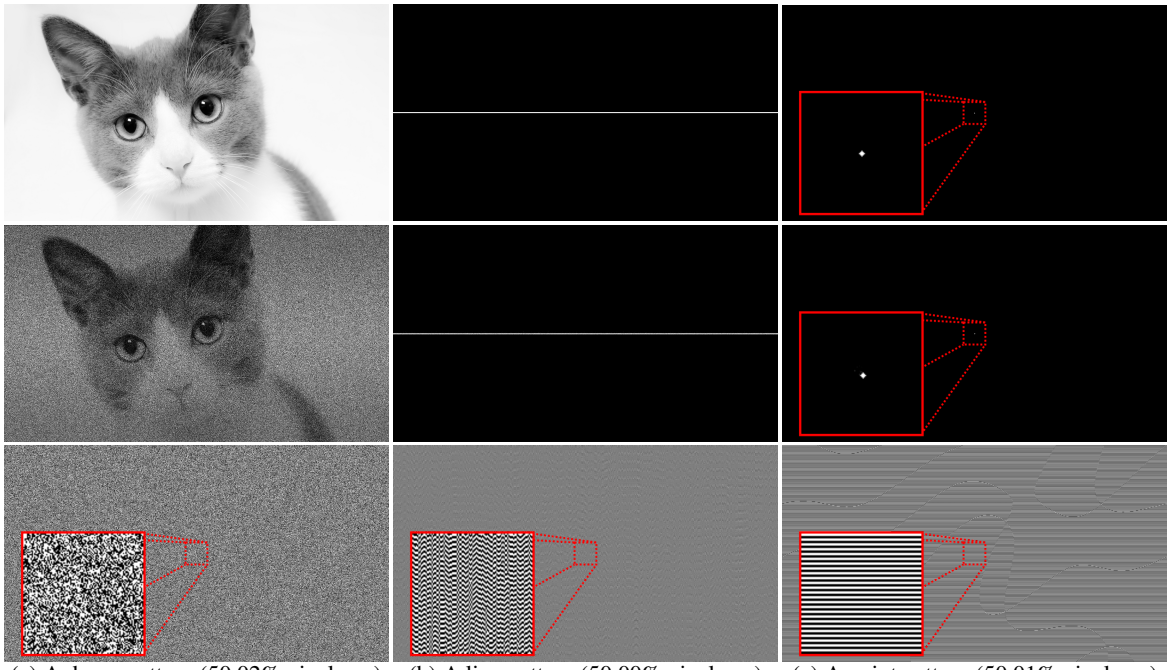
For more general patterns, to maximize light efficiency, the magnitude of the wavefront U_d must also be maximized for every point (x, y). In the case of a binary holographic system like ours, this translates directly to the number of pixels that are turned on to form a desired pattern. We find that this number typically hovers around 50%, as shown in Fig. 1.

In contrast, a phase SLM has fixed magnitude $|U_d(x, y)| = 1$ and controllable phase, making such a system 100% light efficient. However, as mentioned in the main paper, such devices are often severely limited in frame rate to about $60\,\mathrm{Hz}$. In contrast, binary amplitude DMDs can be controlled at rates up to $30\,\mathrm{kHz}$, making them the natural choice for fast structured light applications like light curtains.

In practice, as shown in Fig. 2, our system only requires $50\,\mathrm{mW}$ of light to generate a curve of similar brightness to that of a Sony VPL HS51A LCD projector with a $135\,\mathrm{W}$ bulb—which is $2700\times$ brighter than our laser source. This exemplifies how a binary holographic projector is much more light efficient than a normal projector when creating sparse patterns, making them especially suitable for applications like light curtains.

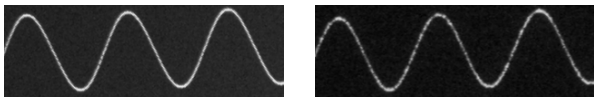
3. Generalized Propagation Operator

In Fourier-transform holography, we model wavefront propagation between two parallel planes (*i.e.*, the Fourier and image planes) using a Fourier transform operator. An



(a) A dense pattern (50.02% pixels on) (b) A line pattern (50.00% pixels on) (c) A point pattern (50.01% pixels on)

Figure 1. Examples of binary holograms generated through the modified Gerchberg-Saxton optimization procedure. **Row 1**: The target patterns. **Row 2**: Simulated reconstructions. Scaled such that the white level is 25% of the maximum value, and the black level is 2.5% of the maximum value. **Row 3**: The recovered binary patterns. Note that approximately 50% of the DMD pixels are turned on in the recovered binary patterns, indicating that only half of the light is blocked by the DMD.



Regular projector Holo. projector [Ours]

Figure 2. Comparing the brightness of a curve generated by a LCD projector with a 135 W bulb versus our system at 50 mW.

objective lens then relays the wavefront $U(s,t)$ to another parallel plane in the scene, scaling the projector pattern in the process. Similar to a normal projector, we can manually adjust the focus of the objective lens to form a sharp image at a particular distance.

Alternatively, scalar diffraction theory, like Kirchhoff diffraction, can also be used to propagate a wavefront from the image plane to any point $\mathbf{x} \in \mathcal{R}^3$ in the scene:

$$U_{scene}(\mathbf{x}) = \mathcal{P}_{diffraction}\{U(s,t)\}(\mathbf{x}) \tag{6}$$

For example, a generalized propagation operator could be used in conjunction with the Gerchberg-Saxton algorithm to generate a hologram at a different plane, *without* needing to physically adjust the objective lens. In the case of a light curtain, we could also optimize the hologram for a 2D manifold R in 3D space, *e.g.*, the region of space imaged by a row of camera pixels. If R is planar, this can be done using techniques based on rotating the angular spectrum [5, 10]. However, these propagation operators can be computationally expensive, especially in the context of an iterative Gerchberg-Saxton algorithm which requires evaluating the propagation operator multiple times. We therefore choose to use the simpler Fourier-based image formation model, which works well in practice.

4. Light Curtain Calibration Procedures

4.1. Geometric Calibration

To compute the intrinsic and extrinsic parameters of the projector-camera system, we use the projector-camera calibration procedure proposed by Moreno *et al*. [6], with two minor differences. First, in place of a checkerboard pattern, we opt to use an inverted circleboard pattern (white circles on a black background) printed onto a planar calibration target. Second, instead of a conventional projector, we rely on our holographic projector to illuminate the calibration target with a sequence of Gray code patterns. Decoding the corresponding Gray code images produces a dense set of correspondences between camera and projector pixels.

To minimize the effect of speckle artifacts in the projection patterns, we compute 4 binary holograms of the same Gray code pattern, using different initializations of Gerchberg-Saxton. We then capture and average the corresponding images to obtain despeckled measurements.

4.2. Timing Calibration

The light curtain system requires accurate time synchronization between the DMD and rolling-shutter camera. A hardware trigger starts the camera's exposure when the DMD displays the first pattern of a sequence. We must then determine an appropriate pixel clock for the camera, and exposure time for the DMD patterns.

We first use an LED blinking at a known, fixed rate to calibrate the inter-row delay of the rolling-shutter camera for different pixel clock values. Then, we position the camera in front of the DMD, and illuminate the DMD with a bright point light source; the light reflects towards the camera only if the DMD pixels are turned "on". We calibrate the DMD timings by displaying a sequence of patterns where all the pixels are turned "off", except for two patterns at known indices where all the pixels are turned "on". By examining where the bright rows occur in the rolling-shutter capture, we can determine both the real pattern exposure time of the DMD as well as the delay between the start of the DMD pattern sequence and the start of the rolling-shutter frame for some desired DMD pattern exposure time.

In practice, because the resolution of the inter-row delay on the camera and the pattern exposure time on the DMD are limited, not all values of pattern exposure times will correspond to a pixel clock value that matches well (e.g., for some pixel clock value, a pattern cannot be exactly displayed at row r, only $r+1$ or $r-1$ for two sequential values of the pattern exposure time). Thus, we performed an exhaustive search of different pixel clock values and different pattern exposure times to find the best matching pair, which we found to be a pattern exposure time of $132\,\mu$s and a pixel clock of $40\,$MHz. We use these parameters for all of the experiments shown in this work.

4.3. DMD Phase-Distortion Calibration

4.3.1 Technical Description

Sec. 4.3 described our phase-distortion calibration method at a high level. Here, we describe our approach in more mathematical detail.

Let m, n be the dimensions of the DMD, and $q \ll m$ and $r \ll n$. Let $P(x, y)$ be some function with non-zero values only in range $(-\frac{q}{2}, \frac{q}{2}) \times (-\frac{r}{2}, \frac{r}{2})$. Then, we can write a shifted version of $P(x, y)$ as follows:

$$P_{(a,b)}(x, y) = P(x - a, y - b) \tag{7}$$

where a and b represent a shift of the pattern such that the $q \times r$ patch is centered over point (a, b). The 2D Fourier transform of $P_{(a,b)}$ will be given by:

$$\mathcal{F}\{P_{(a,b)}\}(s, t) = \mathcal{F}\{P\}(s, t)e^{-i2\pi(as+bt)} \tag{8}$$

Consider the case where there are no distortions at the DMD. Then, if we display $P_{(a,b)}$ at the DMD, by Eq. (3), the pattern formed at the image plane will simply be a rescaled 2D Fourier Transform of P:

$$I_{\text{ideal}}(s, t) \propto \left| \mathcal{F}\{P\}\left(\frac{s}{\lambda f}, \frac{t}{\lambda f}\right) e^{i\frac{2\pi}{\lambda f}(as+bt)} \right|^2 \tag{9}$$

$$\propto \left| \mathcal{F}\{P\}\left(\frac{s}{\lambda f}, \frac{t}{\lambda f}\right) \right|^2 \tag{10}$$

Note that the created pattern is invariant to the exact values of a and b; shifts in the Fourier plane translate to phase modulation in the image plane, which is dropped in the intensity calculation.

Now, consider the case where there is distortion $D_{(a,b)}$ that can be approximated in a $q \times r$ region around point (a, b) as a linear phase ramp:

$$D_{(a,b)}(x, y) \approx e^{i2\pi\left(u_{(a,b)}(x-a)+v_{(a,b)}(y-b)\right)} \tag{11}$$

where $u_{(a,b)}$ and $v_{(a,b)}$ are the slopes of the phase ramp. Then, if pattern $P_{(a,b)}$ is displayed, the corresponding wavefront at the DMD will be given by $P_{(a,b)}D_{(a,b)}$. By the Fourier shift theorem, the resultant intensity captured by a camera at the image plane will be shifted by the slope of the phase ramp:

$$I_{(a,b)}(s, t) \approx I_{\text{ideal}}\left(s - u_{(a,b)}, t - v_{(a,b)}\right) \tag{12}$$

This equation suggests a simple method to calibrate for the phase distortions of the SLM. If I_{ideal} can be found, then to estimate the phase distortion around point (a, b), the resulting projected image $I_{a,b}$ can be simply cross-correlated with I_{ideal} to determine the shift. The peak of this cross correlation will give an estimate of the gradient of the phase distortion $\left(u_{(a,b)}, v_{(a,b)}\right)$. These gradients can then be recombined using a Poisson solve to recover a map of the phase distortion [9].

In practice, selecting q and r requires care. A larger patch may not be accurately represented by our simple linear phase ramp distortion model. At the same time, a smaller q and r yields both darker and lower frequency intensity patterns, which may reduce the accuracy of the cross-correlation matching process. In our case, we set q, r to be one-tenth of the size of the DMD. In addition, because capturing an image for every possible (a, b) location is infeasible, we sample a 19×19 grid on the DMD, and interpolate the resulting gradient values to the full resolution.

4.3.2 Practical Considerations

This approach requires that a sensor images exactly the Fourier Transform of the wavefront at the DMD. In practice, this may not be the case, as the sensor may image just a subset of the resulting interference pattern, or the interference pattern may be rotated or warped. To account for

these issues, we first projected a series of Gray codes to determine correspondences between pixels in the simulated interference pattern and sensor pixels in the real interference pattern. Using these correspondences, we calculated a homography to warp the captured interference patterns to match the simulated ones, after which we followed the procedure outlined in Sec. 4.3. Like in Sec. 4.1, we average over multiple Gerchberg-Saxton instantiations to despeckle the Gray code measurements.

5. Existing Assets and Code

The mesh used for the bunny model in Fig. 10 was taken from the Stanford 3D Scanning Repository [11]. The teapot [1] in Fig. 10 and the toy cow [2] in Fig. 1 are under CC0 licenses. The cat image from Fig. 1 is also under a CC0 license [7]. We created all other meshes used throughout the paper. We use the Open3D [12] and trimesh [3] packages for mesh processing. As described in the main paper, we used the Pycrafter 6500 package [8] to interface with our DLP LightCrafter 6500 EVM.

6. Code Submission

Our supplementary code includes the binary Gerchberg-Saxton algorithm, and our calibration procedure for recovering the DMD phase pattern. You can try out a simple example of binary Gerchberg-Saxton by running */code/gs/test.py* (our Python code requires NumPy and Matplotlib). We also include MATLAB code and data for our DMD phase calibration in */code/distort/*. Although the remainder of our code is specific to our hardware system, we are happy to provide other components upon request.

References

[1] Nik Clark. Utah teapot (solid). https://www.thingiverse.com/thing:852078. 4

[2] Keenan Crane. Spot. https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/. 4

[3] Dawson-Haggerty et al. trimesh. https://trimsh.org/. 4

[4] Joseph W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill physical and quantum electronics series. W. H. Freeman, 2005. 1

[5] Kyoji Matsushima, Hagen Schimmel, and Frank Wyrowski. Fast calculation method for optical diffraction on tilted planes by use of the angular spectrum of plane waves. *JOSA A*, 20(9):1755–1762, 2003. 2

[6] Daniel Moreno and Gabriel Taubin. Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 464–471. IEEE, 2012. 2

[7] Pixabay. Grey and white short fur cat. https://www.pexels.com/photo/grey-and-white-short-fur-cat-104827/. 4

[8] Paolo Pozzi, Dean Wilding, Oleg Soloviev, Hans Verstraete, Laurens Bliek, Gleb Vdovin, and Michel Verhaegen. High speed wavefront sensorless aberration correction in digital micromirror based confocal microscopy. *Optics Express*, 25(2):949–959, 2017. 4

[9] Tal Simchony, Rama Chellappa, and Min Shao. Direct analytical methods for solving poisson equations in computer vision problems. *IEEE transactions on pattern analysis and machine intelligence*, 12(5):435–446, 1990. 3

[10] Johannes Stock, Norman Girma Worku, and Herbert Gross. Coherent field propagation between tilted planes. *JOSA A*, 34(10):1849–1855, 2017. 2

[11] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, 1994. 4

[12] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 4