Supplement for Holodepth: Programmable Depth-Varying Projection via Computer-Generated Holography

Dorian Chan[1], Matthew O'Toole[1], Sizhuo Ma[2], and Jian Wang[2⋆]

[1] Carnegie Mellon University
{dorianchan,mpotoole}@cmu.edu
[2] Snap Inc.
{sma,jwang4}@snap.com

References to the main paper are in blue. Sec. 1 shows more comparisons with other potential methods for projecting depth-varying content. Sec. 2 discusses the limits of depth variation with our proposed setup. Sec. 3 visualizes the effects of using a static element for étendue expansion in the projection of depth-varying content. Sec. 4 provides a proof for the equivalent stereo resolution from Sec. 4.2. Sec. 5 provides more details on the multilayer model we used to represent the lens array. Sec. 6 discusses the slight amount of light redistribution with depth when projecting depth-dependent content. Sec. 7, Sec. 8 and Sec. 9 describe the details of our implementation, including our calibration processes.

Technology	Programmability	HW complexity	Time-mult.	Light eff.
LCD/DLP [2, 23, 32]	Low (blur)	Low (1 SLM)	No	Low
Laser scanning [22, 29, 30]	Low (minimal blur)	Low (scanning mirror)	No	High
Focus-tunable lens [31]	Moderate (crosstalk)	Moderate (SLM+tun. lens)	Yes	Depends
Coded aperture [12, 15, 16]	Moderate (conv. with aper.)	Low (SLM+static ap.)	No	Low
Time-mult. coded ap. [13, 18]	High	High (2 SLMs)	Yes	Low
Holographic [3, 6–8, 19, 20, 27, 28]	Moderate (limited étendue)	Low (SLM+laser)	No	High
Holo. + lens array (ours, [4, 21] in NEDs)	High	Low (Holo.+lens arr.)	No	High

Table 1: High-level comparison of different potential depth-varying projection systems. A holographic approach with a lens array étendue-expander allows for depth-dependent content to be programmed on a single SLM, without needing time multiplexing or light loss. Thus, our work uses such a system to engineer a depth-varying projector.

⋆ Work partially completed while D. Chan was a Snap Research intern. Sample code shared online.

1 More comparisons of depth-varying projectors

More simulated comparisons between a holographic system and traditional approaches that could create a depth-varying projector are shown in Fig. 1. A holographic system with étendue expanded with a lens array outperforms a coded aperture projector, a high-speed projector focused at two depths with a focus-tunable lens, a naive holographic projector rescaled to the same field-of-view, and a similar holographic system with a random phase étendue expander.

In Fig. 2, we show more sophisticated projection systems. In Fig. 2(a), we show a modified version of a focus-tunable lens and high-speed projector system, where the patterns projected for each focal length are jointly optimized to match the target images. This slightly improves the projection quality, but performance still does not match the case of a holographic system with étendue expanded by a lens array. In Fig. 2(b), (c), and (d), we explore time-multiplexed versions of coded aperture projectors [18] and holographic systems. Time-multiplexing significantly improves the performance of a coded aperture approach as illustrated in Fig. 2(b). The improvement is less pronounced for holographic systems as shown in Fig. 2(c) and (d) — we attribute this to limited étendue even after expansion. While time multiplexing is a useful tool, as discussed in Sec. 1, it can be impractical in real applications thanks to increased bandwidth requirements and decreased framerates. Furthermore, a time-multiplexed coded aperture requires an additional SLM, increasing form factor and cost. Like standard coded aperture, projected patterns will also be much darker than a holographic approach thanks to the requisite blocking of light.

These various systems are summarized in Tab. 1. Our system does not require time-multiplexing or complex hardware, but is able to project depth-dependent content with a high level of programmability. It also inherits the light efficiency of holographic projectors [6–8, 28].

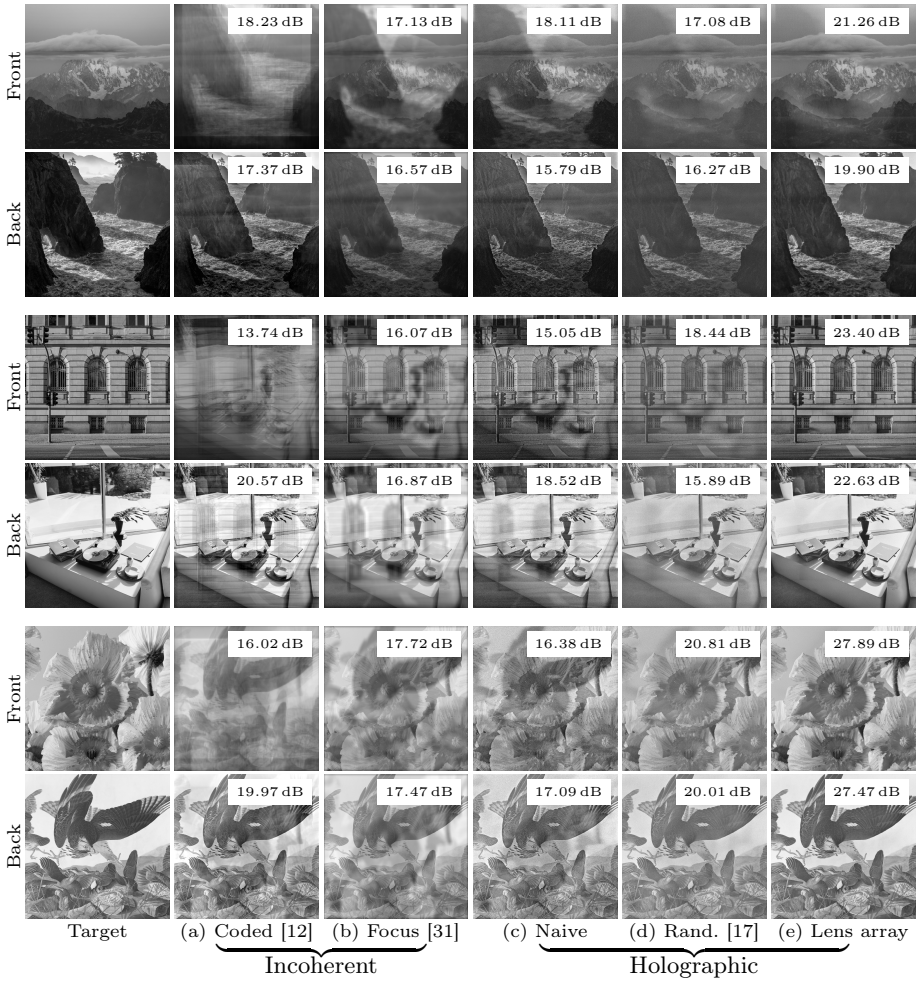


Fig. 1: More simulated comparisons of depth-varying projectors. Like Fig. 2, unique content is projected to two planes. Coded aperture projectors [12] **(a)**, focus-tunable lens/high-speed projectors [31] **(b)**, and naive holographic projectors **(c)** struggle with crosstalk. A holographic system with étendue expanded by a random phase mask [17] performs well visually, but low frequency errors significantly decrease PSNR. A holographic system with étendue expanded by a lens array provides the best results, and we leverage such a setup for our real world prototype.



Fig. 2: More complex depth-varying projectors. (a) We show an optimized version of a focus-tunable lens/high-speed projector, where the patterns projected for each distance are jointly optimized. This increases performance over Fig. 1(b), but still does not match Fig. 1(d). **(b)** We demonstrate a 4× time-multiplexed coded aperture projector [18], which significantly increases performance over the single aperture/display pattern case. **(c)**, **(d)** We demonstrate 4× time-multiplexed versions of a naive holographic system and an étendue-expanded version with the same field-of-view. Time multiplexing mildly improves performance for these holographic systems. In general, while time multiplexing is beneficial, there are practical challenges towards implementation, like increased bandwidth and lower framerates. Furthermore, time-multiplexed coded aperture requires an additional SLM, and will be significantly darker than a holographic approach thanks to the requisite light attenuation.

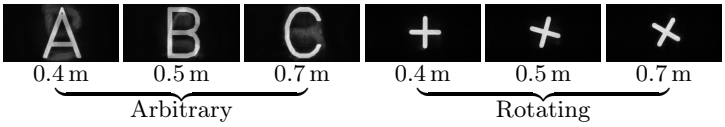


Fig. 3: Simulation of different depth-varying projections. Some patterns can be easier to resolve than other ones, like a rotating pattern [11].

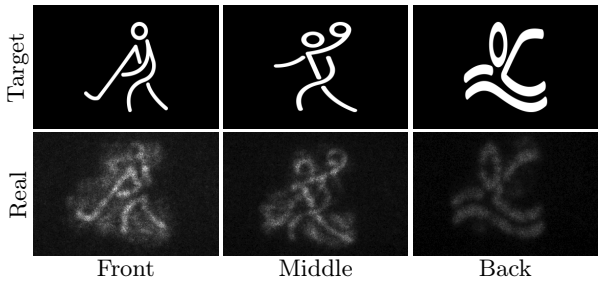


Fig. 4: Limits on depth variation. We tried forming unique content at three planes separated 2.5 mm apart before the projector lens. There is much more blur than a similar scene in Fig. 6(e).

2 Limits of holographic depth variation

In general, some depth-varying patterns are easier to form than others on our holographic system. As shown in Fig. 3, arbitrary content at different planes can be more difficult than specially structured patterns, like a rotating plus. Additionally, the closer together these planes are, the more difficult it is to clearly disambiguate the content at each one, as visualized in Fig. 4. To explore these limits, in Fig. 5, we project unique content at two planes as we vary the distance between them. As we move these two planes closer together, it is harder to find an SLM pattern that will project the correct content at both planes, and instead an intermixing of the content arises. We note, however, that our simulated setup has a somewhat higher depth resolution than our real setup. This suggests that closing the simulation-to-real gap, perhaps using a neural-augmented model [5, 9, 10, 14, 24, 26], could potentially improve the real depth resolution.

3 Static elements and étendue expansion

As described in the main paper, the usage of static étendue-expanding optics produces a structured defocus pattern thanks to the implicit spatial multiplexing of the SLM performed by a high-resolution optic. This structured defocus is illustrated in Fig. 6. Practically, when projecting natural images, this effect reduces the contrast of depth variation when compared to a larger SLM with equivalent étendue, as shown in Fig. 7.

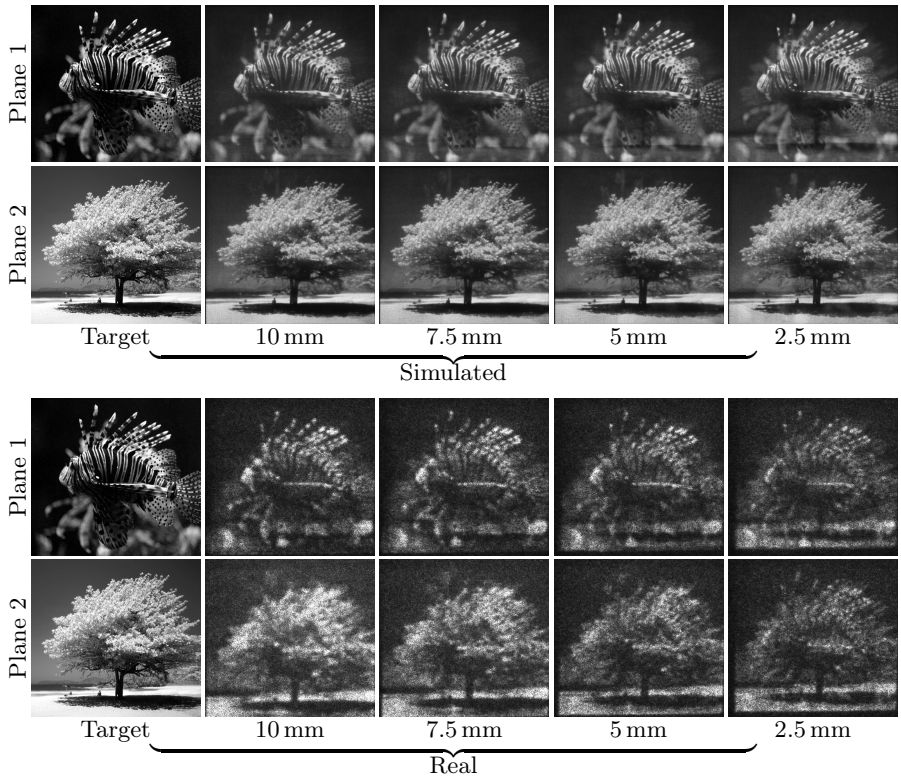


Fig. 5: Depth variation resolution visualization. We visualize the depth variation of two planes receiving different content in simulation and on our real setup. As the distance between the two planes decreases, the more the content intermixes and the harder it is to visually separate. The distances shown are before the projection lens.

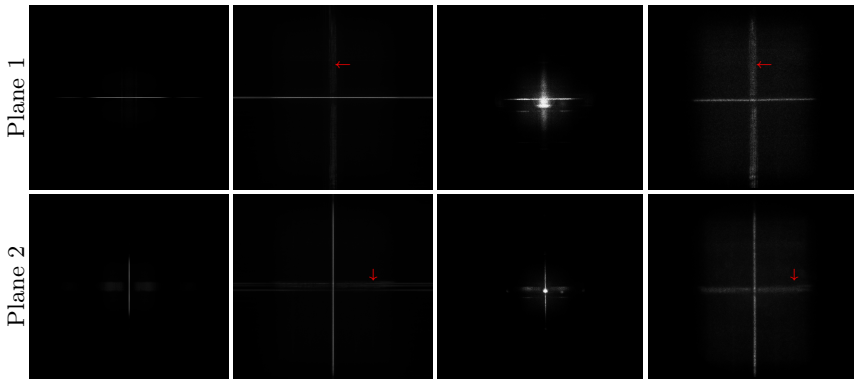


Fig. 6: Structured defocus from a static étendue expander. We visualize the defocus pattern created by a system with étendue expanded by a static element, like a lens array [4, 21] or phase mask [1, 17]. While the defocus pattern stays the same size, it becomes more structured, as shown by the faint lines that appear in **(b)** and **(d)**.

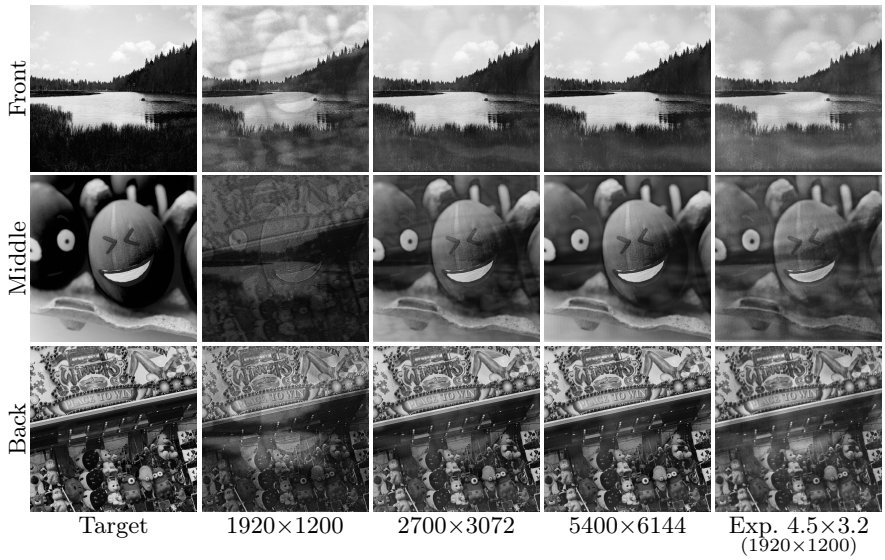
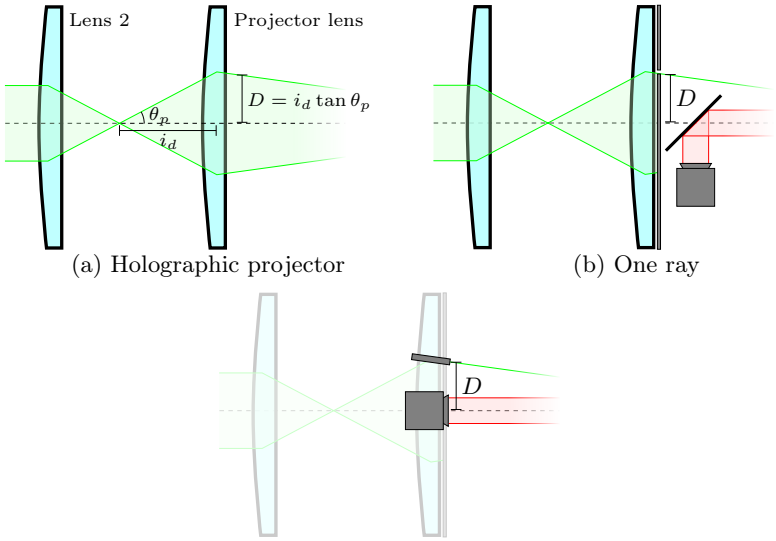


Fig. 7: Comparing depth variation of the étendue-expanded system with larger SLMs. The structured defocus introduced by the static element reduces the quality of depth variation. In our system, a lens array expands the étendue of a 8 µm pixel pitch SLM by 4.5×3.2, creating the same field-of-view as a 1.78 µm×2.5 µm pitch SLM. Ideally, for the same fixed field-of-view, the depth variation of a system with étendue expanded by 4.5×3.2 should be similar to an SLM with 4.5×3.2 times the pixels, *e.g.*, 1920×1200 → 5400×6144. However, its depth variation is more similar to that of a SLM with 2700×3072 pixels.



(a) Holographic projector (b) One ray

(c) Equivalent stereo baseline

Fig. 8: Holographic depth variation resolution visualization. We can coarsely estimate the effective resolution of our system using geometric optics approximations, following a depth-from-defocus argument [25]. **(a)** shows the geometry of the holographic projector. For some point d away from the system, we can calculate its effective aperture as $D = i_d \tan\theta_p$ with $\frac{1}{i_d} = \frac{1}{f_p} - \frac{1}{d}$ and $\sin\theta_p = \frac{\delta N}{2f_2}$, where f_2 is the focal length of Lens 2, f_p the focal length of the projector lens, δ is the SLM pixel pitch and N is the number of SLM pixels along the largest dimension. As shown in **(b)**, consider the case where no light leaves the system except for the light ray that corresponds to this maximum diffraction angle θ_p. Then, as illustrated in **(c)**, the system is roughly equivalent to a stereo system with baseline D.

4 Visualizing the depth resolution of a holographic depth-varying pattern

Using system parameters and geometric optics approximations, we can approximately determine the resolution of our baseline-free holographic depth variation cue, as illustrated in Fig. 8. First, we can compute the resolution of the wavefront after Lens 2 with focal length f_2 as $\frac{\lambda f_2}{\delta N}$, producing maximum diffraction angle $\sin\theta_p = \frac{\delta N}{2f_2}$. This wavefront is then projected into the scene by a projection lens with focal length f_p. Then, for a point p_d that is a distance d from this projection lens, we can calculate the radius of the effective aperture from which this point receives light as $i_d \tan\theta_p$, where $\frac{1}{i_d} = \frac{1}{f_p} - \frac{1}{d}$. Now, following a similar argument to depth-from-defocus [25], consider the case where the depth-varying pattern is selected to be a single light ray that intersects the perimeter of this effective aperture at a_d and passes through p_d. This ray can be effectively replaced with a single collimated laser source at a_d (Fig. 8(c)). From this point-of-view, the effective depth resolution of our system should be roughly equivalent to a stereo

system with baseline $i_d \tan\theta_p$ as described in Sec. 4.2 ("Depth cue comparison"). For example, for a point that is 0.5 m away, the effective baseline should be ≈9 mm.

In general, this derivation ignores the holographic elements of our system which could change its sensitivity. However, it provides a general guideline towards the depth resolution of a holographic depth-varying projector.

5 A multilayer lens array model

The vast majority of recent work [1, 17, 21] proposes using a single thin-layer model to represent static étendue-expanding optics, *e.g.*, $\mathcal{M}_{\text{thin}}(U) = A_{\text{thin}} \cdot U$. However, our lens array is not a thin element, and has a finite thickness. In our work, we try to represent such elements with a more generic model composed of a first modulation, a convolution, and a second modulation, *e.g.* $\mathcal{M}_{\text{ours}}(U) = A_2 \cdot \mathcal{P}^{\text{opt}}_{z_{\text{thickness}}}(A_1 \cdot U)$. Intuitively, these 3 components can be viewed as light entering the array, light propagating within the array, and light exiting the array, akin to the ABCD matrix model for thick lenses. Rather than trying to explicitly define A_1, $\mathcal{P}^{\text{opt}}_{z_{\text{thickness}}}$ and A_2, we optimize for them as part of the calibration in Eq. (8), where the array is illuminated with many different patterns by the SLM and a camera captures how the array manipulates these patterns. As shown in Tab. 2, this generalized model better captures the effects of the lens array than the thin element model over our validation set. We visualize the calibrated parameters for both models in Fig. 10 and Fig. 11.

Model	PSNR
$\mathcal{M}_{\text{thin}}(U) = A_{\text{thin}} \cdot U$	30.28
$\mathcal{M}_{\text{ours}}(U) = A_2 \cdot \mathcal{P}^{\text{opt}}_{z_{\text{thickness}}}(A_1 \cdot U)$	**31.74**

Table 2: Lens array model PSNR over validation set.

6 Light redistribution along depth

The light redistribution property of the Fourier relationship described in Sec. 3.1 describes "in-plane" redistribution — in other words, light from "off" pixels in the same plane are redirected in (x, y) to "on" pixels. Propagating content to different planes, as in our work, can also lead to another dimension of redistribution over depth z. Intuitively, light used to form nearer points can also be used to form farther points, and vice versa. This level of redistribution is determined by the size of the sub-hologram cone, which is defined as the extent to which the light from a single point can diffract [8]. The size of this cone can be calculated by:

$$L = d \tan \sin^{-1} \frac{\lambda}{2\delta}, \tag{1}$$

where d is the propagated distance, L is the radius of this cone at d, and δ is the effective pitch (as determined by the SLM pixel size or étendue-expanded resolution). The larger the difference in depth and the smaller the effective pitch, the more drastic this depth-wise redistribution is.

We explore this experimentally on our setup in Fig. 9. We tried projecting two non-overlapping patterns p_1 and p_2 in three different cases. First, we combined both patterns into a single texture which we projected at a single depth plane, once at z_1 and once at z_2. Then, we used Eq. (4) to find an SLM pattern that would project p_1 at z_1 and p_2 at z_2. Theoretically, assuming an ideal system, when the patterns are projected at unique planes, they should be brighter than the corresponding regions in the combined texture thanks to this depth redistribution. As can be seen in Fig. 9, while the effect is only slight thanks to the small difference in depth between z_1 and z_2, this does hold on our prototype. This increase could potentially lead to a larger effective range of a holographic structured light system, especially for scenes with both close and far objects and significant ambient light.

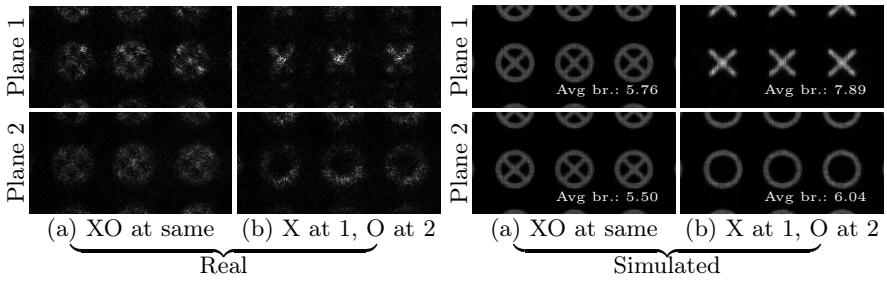


Fig. 9: Light redistribution along depth. In addition to spatial light redistribution [6–8,28], our system also allows a degree of light redistribution along depth — light used to form closer patterns can also be used to form farther patterns. To demonstrate this, an overlayed 'X' and 'O' are projected at plane 1 in the top row of **(a)**, and at plane 2 in the bottom row. In **(b)**, the 'X' is focused at plane 1, while the 'O' is focused at plane 2. When projected at different planes, the 'X' and 'O' are slightly brighter than when they are projected to the same plane. The two planes are separated by 1 cm before the projection lens.

7 Calibration details

7.1 Calibrating the lens array

For calibration with Eq. (8), we projected content onto a blank white wall, which we imaged with a camera. To map the captured images to the projected content, we first projected Gray codes without the lens array in the system. Because the extent of the projected pattern without the lens array should match to a central

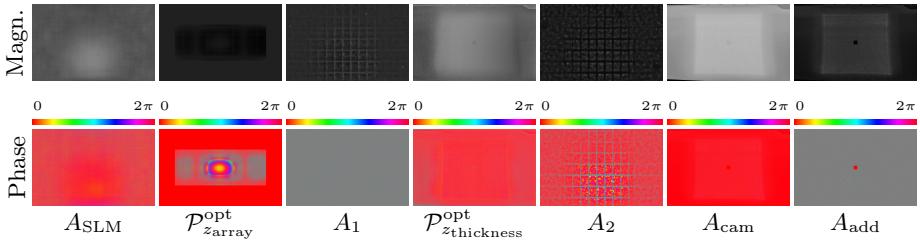


Fig. 10: Calibrated projector parameters. We visualize the learned modulations from Eq. (8). Note that the central blue square in the phase for A_{add} is due to a manual cropping of the DC of the projected pattern.

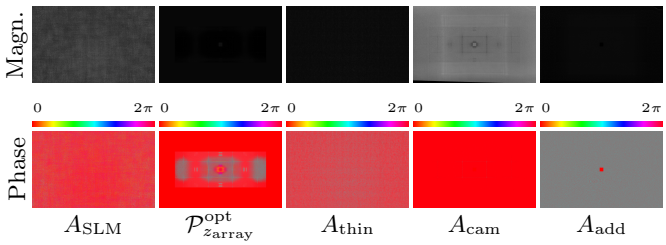


Fig. 11: Calibrated projector parameters with thin mask approximation. We visualize the learned modulations from a version of Eq. (8) that approximates the lens array with a thin phase mask: $\mathcal{M}_{\text{thin}}(U) = A_{\text{thin}} \cdot U$. We observe that the magnitude of solely A_{thin} is not as distinct compared to the learned magnitudes of A_1 and A_2 in Fig. 10, and the phase of A_{thin} is much noisier than that of A_1.

rectangle of the system with the étendue-expanding optic, we can then calculate a homography from the Gray codes mapping captured images to the étendue-expanded projections.

With a mapping, we can now run the optimization process described in Sec. 3.2. We manually cropped out a small neighborhood around the DC spot to reduce its impact on the optimization following Chan *et al.* [6]. We first tried taking two high resolution natural image datasets, and computing SLM patterns for each image using the naive projector model given in Eq. (3). We then captured the corresponding projector output, and ran the optimization given in Eq. (8) with these pattern/capture pairs. However, we found that this resulted in subpar performance — the model did not well replicate the real images, the optimization was not able to recover the structure of the microlenses, and it did not generalize well to new patterns.

We hypothesize that this unimpressive output was because the captured images had very little structure beyond hard-to-model, low-contrast speckle. To address this issue, we observed that each lens in the lens array, when combined with Lens 2 in Fig. 3, effectively formed an afocal relay system that mapped content at the front focal plane of the lens array to the back focal plane of Lens 2. The combination of the n lenses in the lens array results in n overlayed relay

systems at the back focal plane. Thus, if the lenses in the lens array have focal length f_{lens}, we can create calibration images with more structure by instead using Eq. (2) with a propagation distance of $z_{\text{array}} - f_{\text{lens}}$ instead of Eq. 3. We also augmented our dataset with sparse images, composed of randomly placed letters and lines. With these modifications, after 500,000 iterations with a batch size of 1, our optimization from Eq. (8) produced the masks shown in Fig. 10. We also experimented with a thin-layer model $\mathcal{M}_{\text{thin}}(U) = A_{\text{thin}} \cdot U$, as discussed in Sec. 5. As visualized in Fig. 11, the learned magnitude of A_{thin} is much less distinct than that of A_1 and A_2 in Fig. 10, and its phase is noisier than that of A_1.

7.2 Calibrating depth estimation

When recovering depth using Eq. (9), we need some ground-truth projected pattern for every depth z considered for the desired depth resolution. An ideal solution would be to use the model calibrated in Eq. (8), and simulate the expected projected pattern for every depth z. However, in practice, there is a fairly significant simulation-to-real gap when projecting content thanks to the presence of speckle and blur, and as a result, directly using the simulated patterns yielded suboptimal results.

To avoid these issues, we aim to robustly calibrate the depth-varying pattern for every possible configuration of the projected patterns, where different regions of the pattern may be projected to different planes. To start, our chosen projected patterns consist of repeated 'X's and 'O's projected to different planes z_i and z_j — we term each individual 'X' and 'O' as $v_{\mathbf{x}}$, where the depth-varying pattern is centered about pixel $\mathbf{x}$. For simplicity, we constrain the pixels in a local neighborhood around each $v_{\mathbf{x}}$ to the same planes z_i and z_j. For each individual $v_{\mathbf{x}}$, we solve Eq. (4) for every possible configuration of planes $z_i, z_j, \forall i, j$. We denote each computed pattern as $p_{\mathbf{x},(z_i,z_j)}$, which we store in a dictionary for later use. To synthesize a future pattern p_n where each $v_{\mathbf{x}}$ is projected to depths $(z_{i_{\mathbf{x},n}}, z_{j_{\mathbf{x},n}})$, we simply add the associated patterns together:

$$p_n = \sum_{\mathbf{x}} p_{\mathbf{x},(z_{i_{\mathbf{x},n}}, z_{j_{\mathbf{x},n}})} \tag{2}$$

Then, we aim to calibrate the response of every $p_{\mathbf{x},(z_i,z_j)}$. To do so, we first sum together all the patterns that correspond to depth planes z_i, z_j:

$$p_{\text{calib},(z_i,z_j)} = \sum_{\mathbf{x}} p_{\mathbf{x},(z_i,z_j)}. \tag{3}$$

We then project each $p_{\text{calib},(z_i,z_j)}$, and capture images of the corresponding depth pattern by sweeping a board through the scene, and capturing images at known depths (we spaced out each capture by 0.635 cm after our projection lens). For each $\mathbf{x}$, we crop the surrounding neighborhood from these captures, and set it as the real depth-varying pattern for $p_{\mathbf{x},(z_i,z_j)}$ — we call this $c_{\mathbf{x},(z_i,z_j)}$.

Now, let's say we have some capture i_p, captured with pattern p_p where each $v_{\mathbf{x}}$ is projected to $(z_{i_{\mathbf{x},p}}, z_{j_{\mathbf{x},p}})$. We synthesize a ground-truth projected pattern c_p by combining our $c_{\mathbf{x},(z_i,z_j)}$:

$$c_p(\mathbf{x}) = c_{\mathbf{x},(z_{i_{\mathbf{x},p}},z_{j_{\mathbf{x},p}})}, \tag{4}$$

using which we solve Eq. (9) with. In summary, intuitively, we precomputed the patterns for all the possible depths each $v_{\mathbf{x}}$ can be projected to, using which we can synthesize new patterns with. We then calibrate the real depth-varying pattern for each of these possible $v_{\mathbf{x}}$'s, and then stitch up these calibrations into a final "ground-truth" depth-varying pattern.

8 Software implementation details

Our code base is implemented in PyTorch — we share code samples at `https://github.com/dorianchan/holodepth`. Minimizing Eq. (4) for two target planes typically takes about 11 minutes for 2000 iterations on our NVIDIA RTX 3090. For our loss function, we used an augmented version of mean-squared error:

$$\mathcal{L}(G, I) = \frac{1}{N} \sum_{\mathbf{x}} \left(s \cdot \Pi(G(\mathbf{x})) - I(\mathbf{x})\right)^2, \tag{5}$$

where G is the simulated output, I is the target, N is the number of pixels in the image, s is a learnable scale term that is fixed for all images [24], and $\Pi(\cdot)$ denotes an area downsampling that resizes G to the same resolution as I.

Inspired by Chao *et al.* [8], we also tried an energy-preserving version of Eq. (5):

$$\mathcal{L}_{\text{energy}}(G, I) = \frac{1}{N} \sum_{\mathbf{x}} \left(\Pi\left(\frac{1}{S_G} G(\mathbf{x})\right) - \frac{1}{S_I} I(\mathbf{x})\right)^2, \tag{6}$$

where $S_G = \sum_{\mathbf{x}} G(\mathbf{x})$ is the sum of the values in the simulated output and $S_I = \sum_{\mathbf{x}} I(\mathbf{x})$ is the sum of the values in the target. Intuitively, dividing by S_I and S_G effectively normalizes I and G, with $\frac{1}{S_I} I(\mathbf{x})$ representing the ideal energy-preserved signal. However, while we found that this worked well without the lens array, with it sparse patterns typically became extremely blurred in the real system, even though simulated results seemed reasonable. We attribute this to miscalibration in the system that Eq. (5) typically avoids by not trying to maximize light efficiency.

9 Implementation of other methods

For the experiments in Fig. 2, the front plane was at 0.4 m and the back plane was at 0.7 m, while in Fig. 1 and Fig. 2 the front plane was at 0.5 m. To determine the parameters of a coded aperture system, we used the field-of-view of our holographic system to estimate the focal length of a projection lens, given a

typical 0.47 inch DMD chip. Calculating a focal length of roughly 50 mm and assuming an f-number of 1, we simulated an aperture with diameter 50 mm. We selected the in-focus plane to be the back plane of a target depth-varying pattern, and modeled the pattern at the front plane with a convolution between the in-focus pattern with the aperture pattern [12], both of which are grayscale. We ran a gradient descent procedure to optimize both the display pattern and the aperture pattern, minimizing mean-squared error. We ran the same process for the time-multiplexed version in Fig. 2(b), except with multiple aperture and display patterns.

For the focus-tunable lens system, we set the aperture to diameter 10 mm following Xu *et al.* [31]. For the optimized version described in Fig. 2(a), we ran a gradient descent procedure to optimize the patterns displayed when the lens is focused at the front or back planes, accounting for the resulting blur at the other plane.

References

1. Baek, S.H., Tseng, E., Maimone, A., Matsuda, N., Kuo, G., Fu, Q., Heidrich, W., Lanman, D., Heide, F.: Neural etendue expander for ultra-wide-angle high-fidelity holographic display. arXiv preprint arXiv:2109.08123 (2021)
2. Brown, M.S., Song, P., Cham, T.J.: Image pre-conditioning for out-of-focus projector blur. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 1956–1963. IEEE (2006)
3. Buckley, E.: Holographic laser projection. Journal of display technology **7**(3), 135–140 (2011)
4. Chae, M., Bang, K., Yoo, D., Jeong, Y.: Étendue expansion in holographic near eye displays through sparse eye-box generation using lens array eyepiece. ACM Transactions on Graphics (TOG) **42**(4), 1–13 (2023)
5. Chakravarthula, P., Tseng, E., Srivastava, T., Fuchs, H., Heide, F.: Learned hardware-in-the-loop phase retrieval for holographic near-eye displays. ACM Transactions on Graphics (TOG) **39**(6), 1–18 (2020)
6. Chan, D., Narasimhan, S.G., O'Toole, M.: Holocurtains: Programming light curtains via binary holography. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17886–17895 (2022)
7. Chan, D., O'Toole, M.: Light-efficient holographic illumination for continuous-wave time-of-flight imaging. In: ACM SIGGRAPH ASIA 2023 (2023)
8. Chao, B., Gopakumar, M., Choi, S., Wetzstein, G.: High-brightness holographic projection. Optics Letters **48**(15), 4041–4044 (2023)
9. Choi, S., Gopakumar, M., Peng, Y., Kim, J., O'Toole, M., Wetzstein, G.: Time-multiplexed neural holography: a flexible framework for holographic near-eye displays with fast heavily-quantized spatial light modulators. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–9 (2022)
10. Choi, S., Gopakumar, M., Peng, Y., Kim, J., Wetzstein, G.: Neural 3d holography: learning accurate wave propagation models for 3d holographic virtual and augmented reality displays. ACM Transactions on Graphics (TOG) **40**(6), 1–12 (2021)
11. Greengard, A., Schechner, Y.Y., Piestun, R.: Depth from diffracted rotation. Optics letters **31**(2), 181–183 (2006)

12. Grosse, M., Wetzstein, G., Grundhöfer, A., Bimber, O.: Coded aperture projection. ACM Transactions on Graphics (TOG) **29**(3), 1–12 (2010)
13. Hirsch, M., Wetzstein, G., Raskar, R.: A compressive light field projection system. ACM Transactions on Graphics (TOG) **33**(4), 1–12 (2014)
14. Kavaklı, K., Urey, H., Akşit, K.: Learned holographic light transport. Applied Optics **61**(5), B50–B55 (2022)
15. Kawasaki, H., Horita, Y., Masuyama, H., Ono, S., Kimura, M., Takane, Y.: Optimized aperture for estimating depth from projector's defocus. In: 2013 International Conference on 3D Vision (3DV). pp. 135–142. IEEE (2013)
16. Kawasaki, H., Ono, S., Horita, Y., Shiba, Y., Furukawa, R., Hiura, S.: Active one-shot scan for wide depth range using a light field projector based on coded aperture. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 3568–3576 (2015)
17. Kuo, G., Waller, L., Ng, R., Maimone, A.: High resolution étendue expansion for holographic displays. ACM Transactions on Graphics (TOG) **39**(4), 66–1 (2020)
18. Ma, C., Suo, J., Dai, Q., Raskar, R., Wetzstein, G.: High-rank coded aperture projection for extended depth of field. In: IEEE International Conference on Computational Photography (ICCP). pp. 1–9. IEEE (2013)
19. Makowski, M., Ducin, I., Kakarenko, K., Suszek, J., Sypek, M., Kolodziejczyk, A.: Simple holographic projection in color. Optics Express **20**(22), 25130–25136 (2012)
20. Makowski, M., Ducin, I., Sypek, M., Siemion, A., Siemion, A., Suszek, J., Kolodziejczyk, A.: Color image projection based on fourier holograms. Optics Letters **35**(8), 1227–1229 (2010)
21. Monin, S., Sankaranarayanan, A.C., Levin, A.: Analyzing phase masks for wide étendue holographic displays. In: 2022 IEEE International Conference on Computational Photography (ICCP). pp. 1–12. IEEE (2022)
22. O'Toole, M., Achar, S., Narasimhan, S.G., Kutulakos, K.N.: Homogeneous codes for energy-efficient illumination and imaging. ACM Transactions on Graphics (ToG) **34**(4), 1–13 (2015)
23. Oyamada, Y., Saito, H.: Focal pre-correction of projected image for deblurring screen image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8. IEEE (2007)
24. Peng, Y., Choi, S., Padmanaban, N., Wetzstein, G.: Neural holography with camera-in-the-loop training. ACM Transactions on Graphics (TOG) **39**(6), 1–14 (2020)
25. Schechner, Y.Y., Kiryati, N.: Depth from defocus vs. stereo: How different really are they? International Journal of Computer Vision (IJCV) **39**, 141–162 (2000)
26. Shi, L., Li, B., Kim, C., Kellnhofer, P., Matusik, W.: Towards real-time photorealistic 3d holography with deep neural networks. Nature **591**(7849), 234–239 (2021)
27. Shimobaba, T., Kakue, T., Ito, T.: Real-time and low speckle holographic projection. In: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). pp. 732–741. IEEE (2015)
28. Tilmon, B., Sun, Z., Koppal, S.J., Wu, Y., Evangelidis, G., Zahreddine, R., Krishnan, G., Ma, S., Wang, J.: Energy-efficient adaptive 3d sensing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5054–5063 (2023)
29. Tsuji, M., Kubo, H., Jayasuriya, S., Funatomi, T., Mukaigawa, Y.: Touch sensing for a projected screen using slope disparity gating. IEEE Access **9**, 106005–106013 (2021)

30. Ueda, T., Kubo, H., Jayasuriya, S., Funatomi, T., Mukaigawa, Y.: Slope disparity gating using a synchronized projector-camera system. In: 2019 IEEE International Conference on Computational Photography (ICCP). pp. 1–9. IEEE (2019)
31. Xu, H., Wang, L., Tabata, S., Watanabe, Y., Ishikawa, M.: Extended depth-of-field projection method using a high-speed projector with a synchronized oscillating variable-focus lens. Applied Optics **60**(13), 3917–3924 (2021)
32. Zhang, L., Nayar, S.: Projection defocus analysis for scene capture and image display. In: ACM SIGGRAPH 2006 Papers, pp. 907–915 (2006)