

# Holographic Projector for Near-Eye Displays

Yingsi Qin

School of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh PA  
yingsiq@andrew.cmu.edu

## 1. Summary

We built a 3-dimensional holographic projector that can display monochromatic phase-only holograms with true per-pixel focal control for near-eye displays (*e.g.* VR and AR displays). We intended to build such a projector following the hardware and software pipeline discussed in [8]. Learning outcomes of the project includes, but are not limited to: (1) implementing principles of Frenel holography for multiple depths, (2) building an end-to-end hardware and software optical rendering pipeline that consists of SLM encoding, and (3) optics alignment. As future work, we can continue to expand the etendue limitation discussed by Maimone *et al.* [8], by optimizing and incorporating a static scattering mask as discussed in [4]. The code and additional results can be found in the Github repository<sup>1</sup>.

## 2. Background

3D near-eye displays require correct focus cues at each object point in the 3D scene. The three representative areas of 3D displays that are accommodation-supporting are light field displays, multifocal and varifocal displays, and holographic displays. Among them, holographic displays is the most appealing since it offers high resolution imagery, per-pixel focal control, vision correction, large field of view and eyebox, and compact form factors. Therefore, to explain why this project is interesting, we will first discuss other relevant fields of 3D displays, and then build from there.

### 2.1. Multifocal and Varifocal Displays

Multifocal displays reproduce a 3D image by creating multiple focal planes and displaying the virtual images at these different depths at the same time. It creates the focal planes via spatial multiplexing. However, since it displays the planes simultaneously, multifocal displays are restricted to poor occlusion cues and a very small eyebox. Varifocal displays also displays virtual images at different depths, but instead it uses one focal plane only and dynamically adjusts its depth based on the user gaze. It displays the images via time multiplexing. For example, a focus-tunable liquid lens

that can quickly switch between multiple states can be used to time-multiplex multiple focal planes [7]. However, since it uses only one focal plane, varifocal display suffers on latency (especially the gaze tracker) and requires additional rendering of defocus blur for different depths.

### 2.2. Light-Field Displays

Light field displays, on the other hand, express the wavefront of light using individually addressable ray bundles. The light ray angles can be modulated using a microlens array [5], multilayer liquid crystal displays (LCD) [2], or mirror scanners [3]. It thus reproduces the light field entering the eye. However, since the ray bundles are encodings of the wavefront at discrete locations, light field displays can suffer from resolution loss. When a larger size of the ray bundle is used for higher spatial resolution, the display then suffers from the trade-off of low angular resolution due to diffraction and a shallow depth range.

Although both multifocal/varifocal and light field displays can work on incoherent light, they both suffer from coarse wavefront control as they are based on ray optics. Their ability to produce accurate and natural focus cues is limited.

### 2.3. Why this project: Holographic Displays

Holographic displays, on the other hand, can fully replicate reality by reproducing the magnitude and phase of the light for all wavelengths of light emitted by a real scene.

Here is how it works. It reproduces the wavefront by modulating the phase and magnitude of all spatial points in the scene via a spatial light modulator (SLM). Specifically, it starts from encoding the wavefront emitted from a 3D object into a digital diffractive pattern, the computer-generated hologram (CGH), through numerical holographic superposition. Then optically reconstructs the object by (1) displaying the CGH on the SLM and (2) illuminating it with a coherent light source. The degree of freedom to control the accommodation and depth range is thus higher. It offers precise wavefront control, fine focus control, and no spatial and angular resolution trade-off [1].

Maimone *et al.* [8] built such a holographic display

<sup>1</sup><https://github.com/yingsiqin/3DHologram>

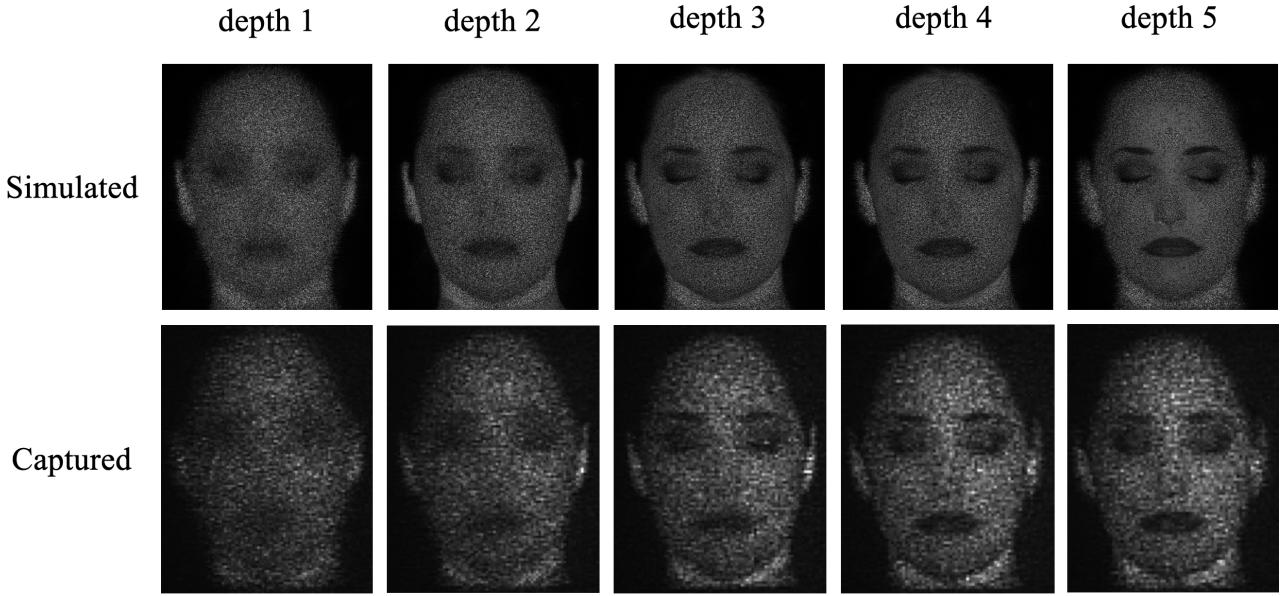


Figure 1. Comparison of our simulation and experimental results: the reconstructed 3D hologram views of the face image from Homework 5, at 5 different depths. We can see that the ears and face center go in and out of focus interchangeably due to them being at different depths. The range of depth is [0.020m, 0.021m]. We discuss this result and the causes of the noises in Section 5.

pipeline that can display a high-resolution image within a large field of view (FOV) while supporting accommodation cues and a large eyebox with a compact form factor.

#### 2.4. The project and the learning

Building such a holographic display pipeline includes software and hardware components. The *software* side includes MATLAB simulation, which consists of (1) generating the Fresnel zone plate, (2) forward propagation and backward propagation of light between the focus plane and SLM plane, (3) amplitude control with phase-only SLMs, for which we adopted Maimone *et al.*'s method, a variation of the double phase encoding method that expresses a complex field as the summation of two phase-only fields for multi-focal Fresnel holograms. The software side also includes using the camera control application Spinvview and adjusting camera parameters such as exposure time, exposure compensation, gain, and the frame rate per second. The *hardware* side, on the other hand, includes aligning the optical components (*e.g.* lenses, aperture irises, laser, camera, SLM, and the eye piece), adjusting the polarization of the laser, and performing all rendering and holographic calculations on a PC. More details will be discussed in Section 3.

Further studies into holographic displays include exit-pupil shifting and gaze-tracking [6]. Neural holography has recently emerged, which achieves unprecedented image

fidelity and real-time frame rates by building an algorithmic CGH framework that uses camera-in-the-loop training of deep neural networks [9]. For the scope of this project, we will focus on the building block - to build a holographic projector.

As explained above, the nature of holographic displays, the topic itself, is an appealing one for the future of 3D displays. While this project has a lot of theoretical overlap with the computational imaging research area, it requires substantial learning on holographic theories and implementing the equations. It also requires optical hardware implementation. We believe that such project would be provide a worthwhile and fun learning experience.

### 3. The Set Up

#### 3.1. The Hardware and Software Pipeline

We built a monochromatic holographic projector that allows per-pixel focal control. We built off from the hardware setup described in [8]. We used a HOLOEYE GAEA liquid crystal on silicon (LCOS) reflective phase-only SLM with a resolution of  $3840 \times 2160$  pixels, pixel pitch of  $3.74\mu m$ . Since it is a 4K resolution SLM, the high linear pixel density allows a wide FOV (but shallow DOF).

In addition, instead of illuminating the SLM with a single optical fiber that was coupled to three laser diodes emitting at 448, 524, and 638 nm, we restricted to a monochro-

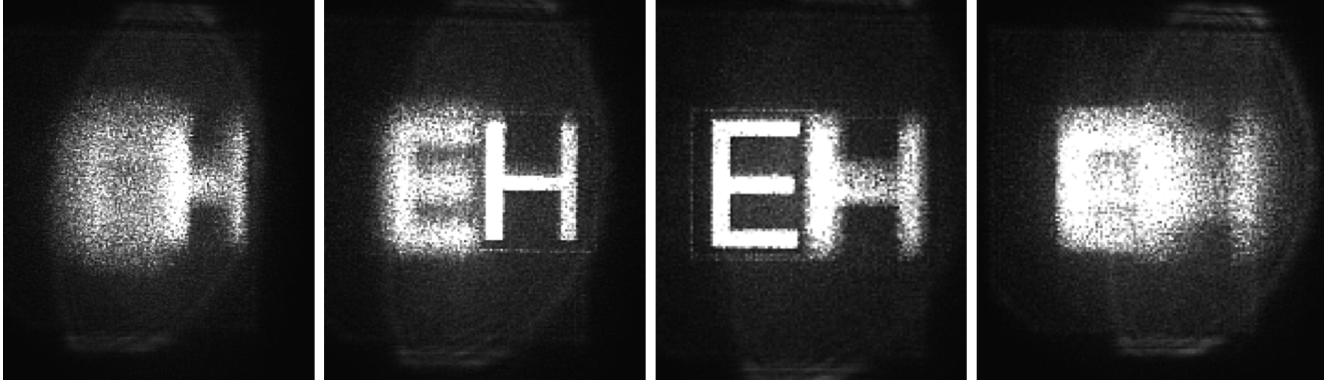


Figure 2. Comparison of our simulation and experimental results of the two-depth image: the reconstructed 3D hologram views of two alphabets, displayed at 4 different depths. We can see that the two alphabets go in and out of focus interchangeably at different depth planes. The two depths are [0.07m, 0.08m]. We discuss the this result and the causes of the noises in Section 5.

matic laser at 520 nm. We linearly polarized the laser light to match the requirement of the SLM. We connected the SLM to a PC to perform all rendering and holographic calculations. We then photographed the display using a Point Grey Blackfly monochromatic camera, different from Maimone *et al.* who used a Point Grey Blackfly S industrial color camera with a resolution of  $2448 \times 2048$  pixels. On the software side, MATLAB was used to perform calculation of the CGH, *Spinview* was used to control the camera, and HOLOEYE GAEA pattern generator was used to display our calculated CGH *only*. To evaluate our result, we used our own face depth map calculated from Homework 5 using calibrated photometric stereo. The picture of our setup is shown in Figure 3.

### 3.2. The code

There was not a publicly available codebase or any starter code from [8]. All of our code were written from scratch following the equations explained in the Methods section based on 3D CGH principles explained in the book [10]. The physical parameters defined in our code were determined after trial and errors. Our code are uploaded to the Github repository <sup>2</sup>.

## 4. Methods

We show in this section how we compute the phase-only hologram  $H$  which we display on SLM, how we encode the amplitude in phase-only terms, and how we simulate the reconstructed hologram views via backward propagation. In our implementation, we performed both forward and backward propagation between the SLM plane and the hologram plane, the former for computing the hologram  $H$ , and the latter for visualizing a simulated reconstruction, which we showed later in the results section.

<sup>2</sup><https://github.com/yingsiqin/3DHologram>

### 4.1. Generation of hologram: forward propagation

We compute the hologram using the Fresnel holography principle adopted by Maimone[8], depicted in Figure 4. A beam is focused at finite distances by a hologram, to several object points, which are formed by overlapping regions that are sub-holograms. Each sub-hologram is a phase-altering function like a refractive lens. The lens phase function, or sub-hologram,  $f_{\text{forward}}(\vec{p})$ , for each pixel, is defined as:

$$f_{\text{forward}}(\vec{p}) = e^{j(\phi_0 + \frac{2\pi\|\vec{p} - \vec{o}\|}{\lambda})}$$

where  $\vec{p}$  is the location on the SLM and  $\vec{o}$  is the location of the projected hologram in space.  $\phi_0$  is initialized to be a random phase for each pixel.

Then the hologram is a summation of product the amplitude at each pixel with the lense phase function at that pixel:

$$H(\vec{p}) = \sum_{j \in s_p} a_j f_{\text{forward}}(\vec{p}_j)$$

This is essentially a convolution. In our implementation, for single-depth images, we computed this as the Inverse Fourier Transform of the product of the Fourier Transforms of  $f_{\text{forward}}(\vec{p})$  and  $I$  where  $I$  is the input intensity image. We name this forward propagation, from the SLM plane to the hologram volume.

For multi-depth images, for depth  $d \in [R]$ , we perform the above propagation for each  $I_d$ , which is the input intensity image having only intensities at pixels of depth  $d$  and zero elsewhere.

$$H_d = \text{IFFT} \left( \text{FFT}(f_{\text{forward}}(\vec{p})) \cdot \text{FFT}(I_d) \right)$$

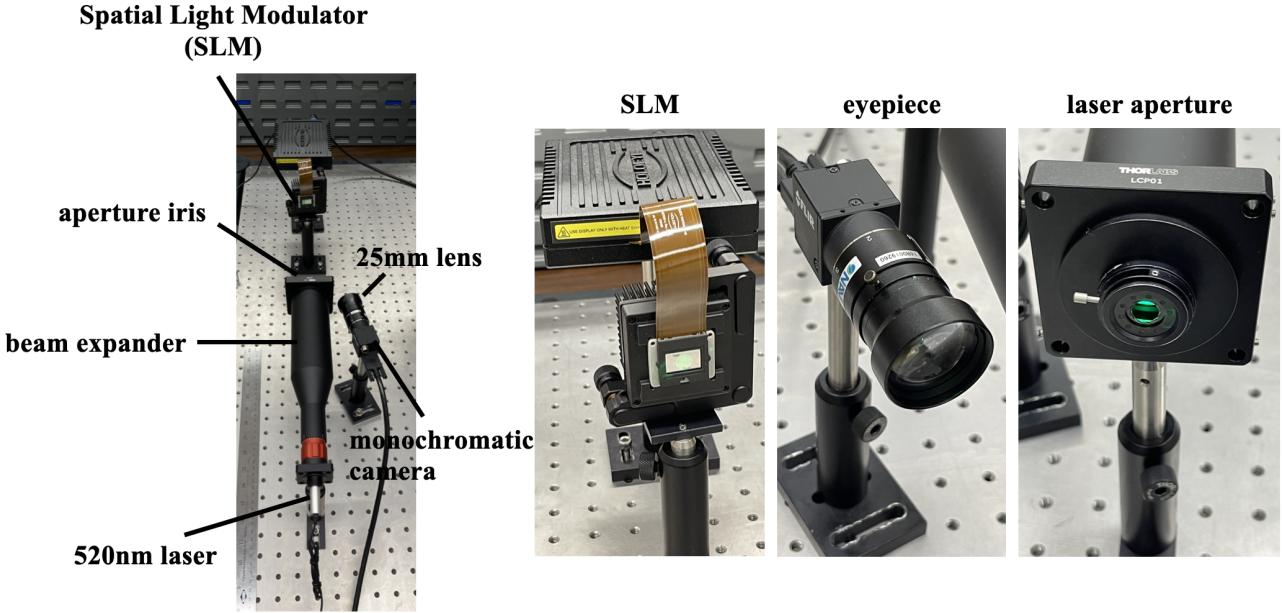


Figure 3. our optics setup

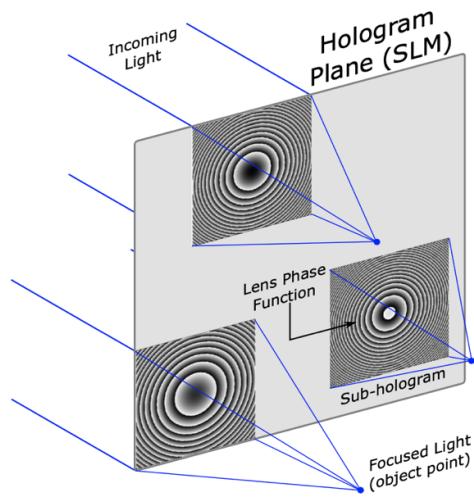


Figure 4. Taken from [8]: illustration of Fresnel holography.

Then after obtaining the corresponding hologram  $H_d$ , we sum all  $H_d$  for all depths and the summation

$$H = \sum_{d \in \mathbb{R}} H_d$$

which is our generated multi-depth hologram. Per-pixel focal control is thus allowed as different pixels can be focused at different depths.

#### 4.2. Simulating the reconstructed hologram views: backward propagation

To simulate the reconstructed hologram view at depth  $d$ , we perform backward propagation, by taking

$$I_d = \text{IFFT} \left( \text{FFT} (f_{\text{backward}}(\vec{p})) \cdot \text{FFT} (H) \right)$$

where

$$f_{\text{backward}}(\vec{p}) = e^{-j(\phi_0 + \frac{2\pi \|\vec{p} - \vec{\sigma}\|}{\lambda})}$$

Then we have  $I_d$  is the simulated propagated hologram view at depth  $d$ . Examples of these simulations are shown in Figure 2 and Figure 1.

#### 4.3. Encoding the Amplitude

**Double phase encoding.** Since we are working with a phase-only SLM, we need a method to encode the amplitude in the phase. We adopt the same method used by [8]. We can encode any complex value  $c$  with amplitude  $a \in [0, 1]$  (our input intensity image is normalized) and phase  $p$  into the sum of two values  $c = ca + cb$ :

$$c = ae^{ip}$$

$$p_a = p - \cos^{-1} a, p_b = p + \cos^{-1} a$$

$$c_a = 0.5e^{ip_a}, c_b = 0.5e^{ip_b}$$

Thus we convert the complex-valued hologram  $H$  into summation of the following phase-only holograms:

$$P_a(\vec{p}) = \angle H(\vec{p}) - \cos^{-1} |H(\vec{p})|$$

$$P_b(\vec{p}) = \angle H(\vec{p}) + \cos^{-1} |H(\vec{p})|$$

Note that we require two phase values and thus two pixels for each complex value in  $H$ . We expect to see loss in the resolution.

Note that we require two phase values and thus two pixels for each complex value in  $H$ . We expect to see loss in the resolution.

**Direct removal of intensity.** Alternatively, we also experimented with directly removing the amplitude of the complex-valued hologram  $H = Ae^{j\theta}$  treating  $A = 1$  across the image and encoding the angle directly using euler's equation. The phase-only hologram  $HP$  is computed as:

$$HP = \cos(\theta) + j * \sin(\theta)$$

We observe a significant decrease in the noise using double phase method than direct removal of intensity. One can reproduce our result by following the README in our Github repository<sup>3</sup>.

## 5. Results and Discussion

**Simulation and Experimental Results.** Here we show two examples of the results. The first example, the input image is shown in Figure 5 and the simulated and experimental results are shown in Figure 2. This scene has two depths. The input image has the left half at one depth and the right half at another other depth. We can see that from left to right, the letter E and H get focused and de-focused interchangeably. It is apparent that the two depth planes are reconstructed. The two depths are  $[0.07m, 0.08m]$ . The second example, as shown in Figure 6 and Figure 1, has 13 depths. We only show a finite range of depths for demonstration. Per-pixel focal control is possible by using input image where each pixel is at a different depth. We can see that from depth 1 to depth 5, different parts of the face are focused and de-focused. It is apparent that the ears and the face center are at different depths. When the face center is focused, the ears go out of focused, and vice versa. The range of depth is  $[0.020m, 0.021m]$ . Experimentally, we found a high DC term in the propagated hologram. Theoretically, we can apply a linear phase ramp to the hologram and thus shift it away from the DC term and use an aperture to crop the DC term out. However, by applying a linear phase ramp, we are also skewing the shape of our 3D hologram volume. Therefore, instead of applying a linear phase ramp, we opted to capture the first order

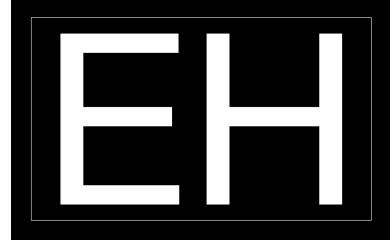


Figure 5. Input intensity image for two depths only



Figure 6. Input face image: intensity on the left, depth map on the right. The depth map is discretized into 13 depths.

diffraction, which were not impacted as much by the strong DC output of the laser.

**The generated 3D Holograms.** We show the generated phase-only holograms for the alphabets example in Figure 9 and for the face example in Figure 8. More computed examples can be found in our Github repository. Each hologram is scaled to the resolution of our HOLOEYE GAEA SLM which has a pixel resolution of  $3840 \times 2160$  pixels. Note that when displaying the holograms during the experiments, we iteratively adjusted our setup, therefore, constant translation were applied to the holograms on the SLM display. In addition, constant phase ramps were applied to revert the tilt on the SLM in our setup. One can replicate our result by changing the physical parameters in our code, such as the wavelength of laser, resolution of the SLM, and SLM pixel pitch, etc.

**Iterative propagation.** We also experimented with iteratively propagating between the SLM and hologram depth planes for more than just one time. In the single-depth case, we would just propagate the entire image iteratively for forward and backward propagation since it only has one hologram focus plane. However, for multi-depth input images, we perform iterative propagation by forward propagating from SLM to hologram depth planes separately for each depth plane, then summing the resulting holograms, and backward propagating the summed hologram to

<sup>3</sup><https://github.com/yingsiqin/3DHologram>

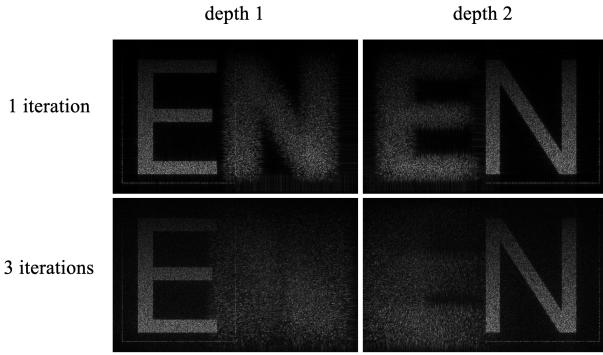


Figure 7. Simulated reconstruction results for iterative propagation. The intensity is intentionally scaled vertically to observe the behaviors on different intensities.

different depths, resulting in separate intensity images. For the subsequent iteration, we then forward propagate these resulting intensity images separately to different holograms again, which will be summed subsequently as well. We show the simulation results comparing 1 and 3 iterations in Figure 7. We observe that higher iterations of propagation create more blurs in the out-of-focus area. This was not an expected behavior, which was likely due to the tilt of the SLM in our setup – the camera image plane is not perpendicular to the optical axis which the hologram is propagated on.

**Noise.** We observe a lot of noise in our results. This can be caused by:

1. SLM pixel pitch restriction on sub-hologram size
2. the random phase offset term in the Hologram
3. the double phase method where we use two SLM pixels to encode one pixel of the input image

## 6. Future work

As future work, we hope to (1) enable full color display, (2) complete a wide FOV virtual reality display, and (3) expand the etendue by exploring the use of an additional scattering mask recently published by Kuo *et al.* [4].

For (1): This will require coupling the single optical fiber to three laser diodes emitting three wavelengths, and programming an Arduino device to control the laser power so we can synchronize the three color lasers to the display.

For (2): Having a holographic projector unit is just one of the components of the display system. For near-eye displays, an eyepiece is needed to magnify the image and relay it to the eye. Therefore, we can complete a 70° wide FOV VR display by adding a suitable eyepiece (Maimone

used Meade Series 5000 21 mm MWA) and following their benchtop optical design.

For (3): Up to this point, the limit of etendue, based on the number of pixels in the SLMs, creates a trade-off between the FOV and the eyebox size. We will explore scattering-based étendue expansion to break this trade-off by following Kuo *et al.* to optimize a static scattering mask and place it before focusing the coherent light into an image.

## References

- [1] Chenliang Chang, Kiseung Bang, Gordon Wetzstein, Byoungho Lee, and Liang Gao. Toward the next-generation vr/ar optics: a review of holographic near-eye displays from a human-centric perspective. *Optica*, 7(11):1563–1578, Nov 2020. 1
- [2] F. Huang, K. Chen, and G. Wetzstein. The Light Field Stereoscope: Immersive Computer Graphics via Factored Near-Eye Light Field Displays with Focus Cues. *ACM Trans. Graph. (SIGGRAPH)*, (4), 2015. 1
- [3] Changwon Jang, Kiseung Bang, Seokil Moon, Jonghyun Kim, Seungjae Lee, and Byoungho Lee. Retinal 3d: Augmented reality near-eye display via pupil-tracked light field projection on retina. *ACM Trans. Graph.*, 36(6), Nov. 2017. 1
- [4] Grace Kuo, Laura Waller, Ren Ng, and Andrew Maimone. High resolution étendue expansion for holographic displays. *ACM Trans. Graph.*, 39(4), July 2020. 1, 6
- [5] Douglas Lanman and David Luebke. Near-eye light field displays. *ACM Trans. Graph.*, 32(6), Nov. 2013. 1
- [6] Byoungho Lee, Youngjin Jo, Dongheon Yoo, and Juhyun Lee. Recent progresses of near-eye display for AR and VR. In Ettore Stella, editor, *Multimodal Sensing and Artificial Intelligence: Technologies and Applications II*, volume 11785, pages 1 – 6. International Society for Optics and Photonics, SPIE, 2021. 2
- [7] Gordon D. Love, David M. Hoffman, Philip J.W. Hands, James Gao, Andrew K. Kirby, and Martin S. Banks. High-speed switchable lens enables the development of a volumetric stereoscopic display. *Opt. Express*, 17(18):15716–15725, Aug 2009. 1
- [8] Andrew Maimone, Andreas Georgiou, and Joel S. Kollin. Holographic near-eye displays for virtual and augmented reality. *ACM Trans. Graph.*, 36(4), July 2017. 1, 2, 3, 4
- [9] Liang Shi, Beichen Li, Changil Kim, Petr Kellnhofer, and Wojciech Matusik. Towards real-time photorealistic 3d holography with deep neural networks. *Nature*, 592, Mar. 2021. 2
- [10] Peter Wai Ming Tsang. *Computer-Generated Phase-Only Holograms for 3D Displays: A Matlab Approach*. Cambridge University Press, 2021. 3

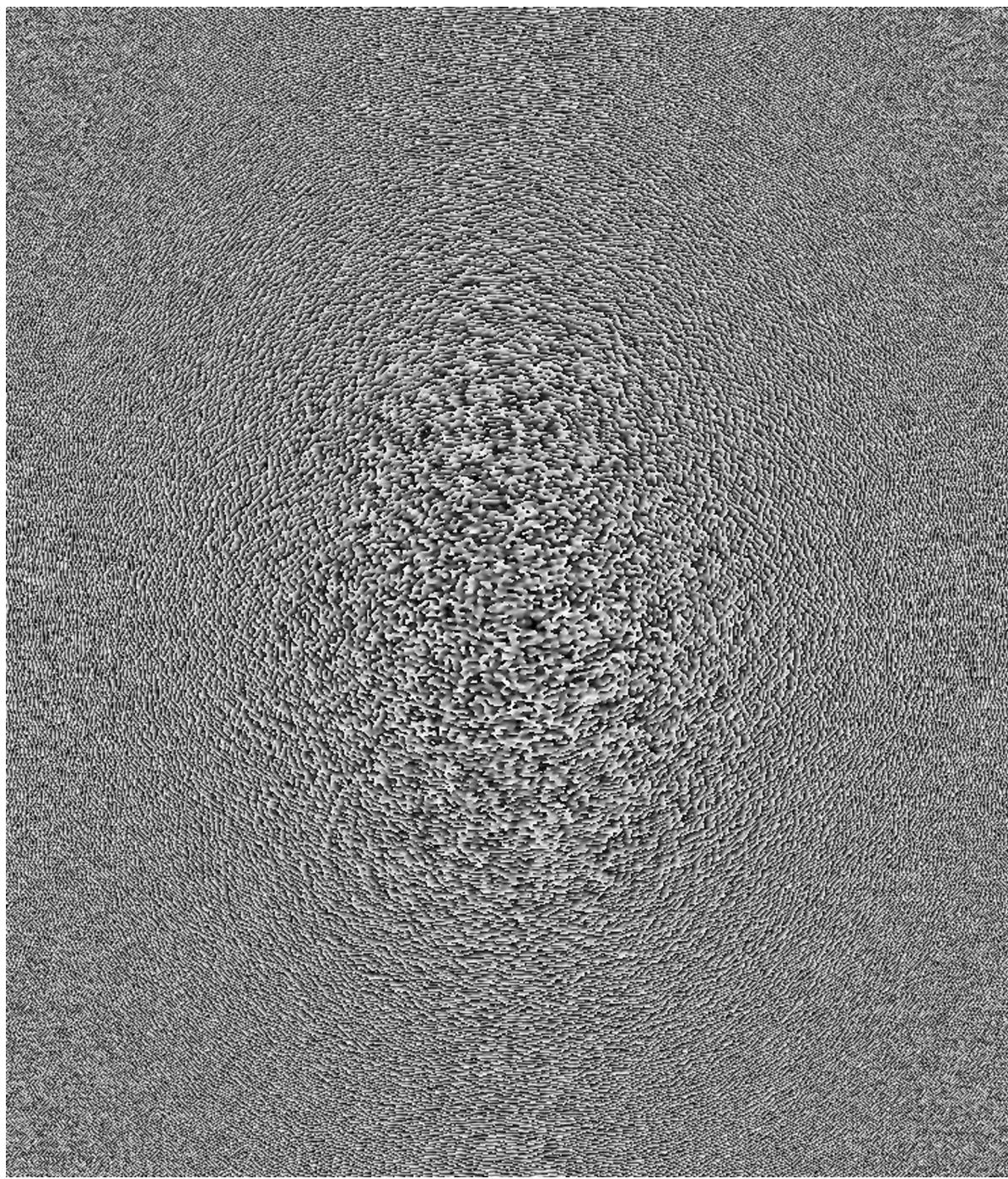


Figure 8. The Computed Generated Hologram (CGH) for example 2 in Figure 1.

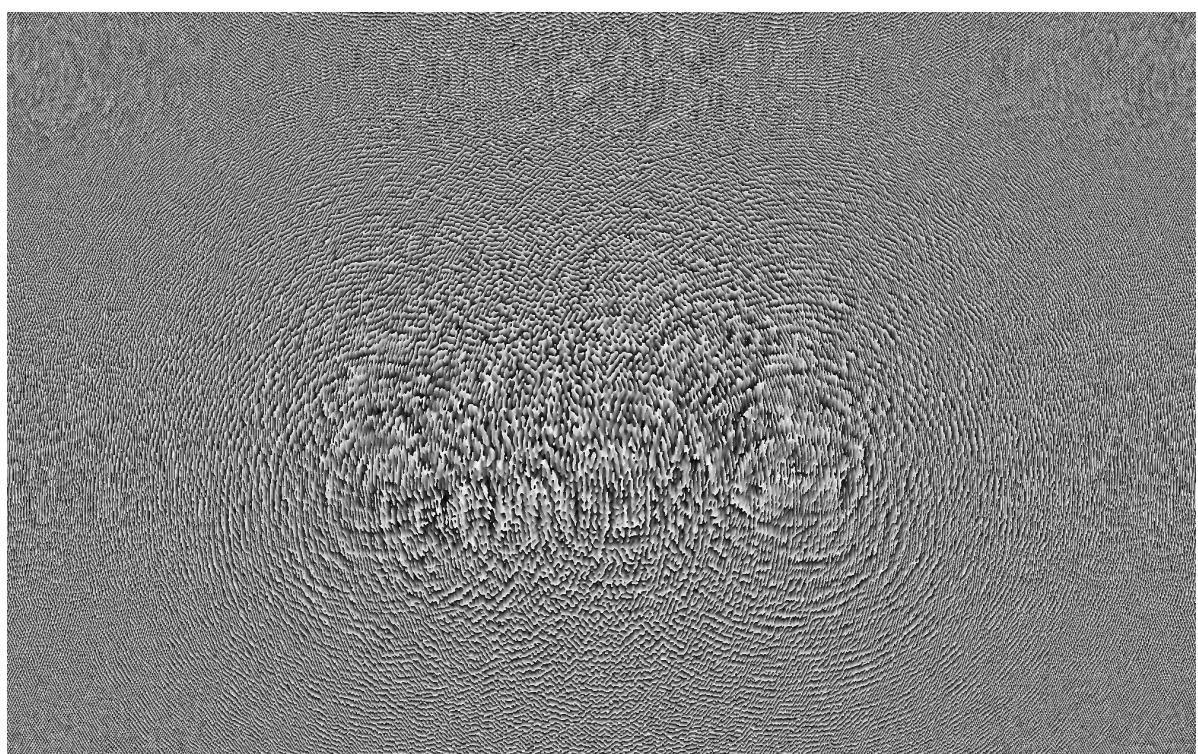


Figure 9. The CGH for example 1 in Figure 9.