2084 Vol. 51, No. 8 / 15 April 2026 / Optics Letters Letter

Optics Letters

Computer-generated holography using the generalized Van Cittert–Zernike Schell propagator

MANUEL MONTOYA,[1,2,*] YUNFENG NIE,[3] AND DAVID BLINDER[1,2]

[1]Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussel, Belgium
[2]IMEC, Kapeldreef 75, B-3001 Leuven, Belgium
[3]Brussels Photonics (B-PHOT), Department of Applied Physics and Photonics, Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium
*manuel.montoya.zuluaga@vub.be

Received 19 February 2026; revised 23 March 2026; accepted 24 March 2026; posted 25 March 2026; published 6 April 2026

We introduce a computer-generated holography system based on the theory of partial spatial coherence, simplifying the required optical setups and improving computational efficiency. The method utilizes the generalized Van Cittert–Zernike Schell (GVS) model to calculate the forward propagation of light emitted by a common mobile phone screen and reflected by a phase-only spatial light modulator at various propagation distances. An inverse problem is then solved using stochastic gradient descent optimization to obtain the intensity and phase holograms that produce the desired intensity images at different depths. We conduct a two-layer optimization validated through numerical and optical experiments. Using the GVS model, we report speed improvements of 15× and higher-quality numerical reconstructions over the reference method of compressive propagation with random wavefronts. Furthermore, we achieve numerical and optical 3D focus–defocus effects by optimizing a focal stack with 20 depth layers.

https://doi.org/10.1364/OL.595741

Computer-generated holography (CGH) encompasses a range of numerical techniques that enable the reconstruction of arbitrary optical fields by generating interference patterns in a computer. It has been extensively adopted in the fields of optics and photonics [1], with applications in holographic microscopy and tomography [2], optical tweezers [3,4], modeling of imaging systems [5], and 3D display technology [6]. Traditionally, perfect spatially and temporally coherent light has been assumed in CGH studies [7], which is convenient because it allows leveraging the Fast Fourier Transform (FFT) algorithm to make efficient diffraction calculations [8]. However, coherent light presents inherent limitations for 3D displays and computational imaging. One significant limitation is speckle noise, which reduces image quality and can be hazardous to the human visual system [9]. Moreover, the use of highly coherent laser sources results in higher costs and greater complexity in both imaging and display systems.

In recent years, many studies have tried to overcome these issues by using light sources with lower coherence [10–15]. A notable example is the work by Horisaki *et al.* in which they demonstrated a CGH method using incoherent light emitted from a mobile phone screen and one spatial light modulator (SLM) [15]. This approach proved to be more cost-effective, less complex, and had better reproduction performance than in their previous studies. To achieve this, the authors solved an inverse problem using stochastic gradient descent (SGD) and compressive propagation with random wavefronts (CP-R) [13], obtaining promising results in a low-cost color 3D display system. Nonetheless, CP-R is an approximate method for modeling partial spatial coherence and requires computing M coherent propagations, thereby suffering from a speed-accuracy trade-off.

In this Letter, we propose a novel CGH method based on partial spatial coherence that is highly accurate and at least one order of magnitude faster than CP-R. We recently introduced the Generalized Van Cittert–Zernike Schell (GVS) propagator [16], which relies solely on the Fresnel and paraxial approximations and can be efficiently computed using a single forward propagation with multiple FFTs. Here, we use GVS and SGD to realize partially spatially coherent CGH. We demonstrate the effectiveness of the method through numerical and optical experiments, in which we reproduce two different color images at two different depths using a mobile phone screen and a phase-only SLM, as in [15]. Moreover, we perform a focal stack optimization with 20 planes to further evaluate the technique's 3D display capabilities, which we also validate through optical experiments.

In the proposed method, we optimize for four image channels (RGB and phase). We initialize the red, green, and blue channels with random data, and the phase channel to a zero matrix. The RGB channels form the intensity hologram displayed on the mobile phone screen, and the phase channel corresponds to the phase-only hologram displayed on the SLM. We assume a perfectly incoherent light source, and we use Angular Spectrum GVS (AS-GVS) [16] to compute the forward light propagation accounting for partial spatial coherence, as follows:

$$\hat{I}_{c,n} = \frac{1}{\pi(\lambda_c z_s z_n)^2} \boldsymbol{F}\left\{\boldsymbol{F}^{-1}\left\{\|\mathbf{ASM}[S_c]\|^2\right\}\mu_c\right\}, \qquad (1)$$

where $\hat{I}_{c,n}$ corresponds to the intensity at the sensor plane, computed independently for each color channel c and each depth layer n; λ_c is the wavelength associated with the current color channel, z_s is the distance between the phone screen and the

SLM, z_n is the distance between the SLM and the current depth layer, $\boldsymbol{F}$ denotes the Fourier transform, $\boldsymbol{F}^{-1}$ its inverse, and $\mathbf{ASM}[\,\cdot\,]$ the angular spectrum method, as implemented in [17]. S_c represents an operator that translates the real-valued phase-only hologram h_p for the SLM to a complex amplitude field, and μ_c is the complex coherence factor reaching the SLM from the intensity hologram h_c displayed on the screen.

The operator S_c for the phase-only hologram is expressed as:

$$S_c = \exp\left(2\pi j \Lambda_c h_p\right) \exp\left(\frac{j\pi}{\lambda_c z_s}(x^2+y^2)\right), \tag{2}$$

where j denotes the imaginary unit, x and y are the spatial coordinates on the SLM plane, and $\Lambda_c = \lambda_c/\lambda_{ref}$ represents the spectral dispersion scaling factor, with λ_{ref} indicating the reference wavelength for which the SLM is calibrated [18].

The complex coherence factor μ_c is given by:

$$\mu_c = \mathrm{interp}_{\mathrm{SLM}}\left[\frac{\boldsymbol{F}\{\mathrm{interp}_n[h_c]\}}{\max|\boldsymbol{F}\{\mathrm{interp}_n[h_c]\}|}\right], \tag{3}$$

where $|\cdot|$ denotes absolute value, max is the maximum value operator, interp_n is an operator for interpolating the intensity holograms to ensure consistent dimensions during the propagation process depending on each given z_n, and $\mathrm{interp}_{\mathrm{SLM}}$ is an operator for interpolating the resulting field to the size of h_p.

As explained in [15], our goal is to obtain h_{red}, h_{green}, h_{blue} and h_p that bring $\hat{I}_{c,n}$ closest to the target image $I_{c,n}$ for each color and depth layer. We can write this inverse problem as:

$$\underset{h_c,h_p}{\arg\min} \sum_{n=1}^{N}\sum_{c=1}^{3}\left\|\hat{I}_{c,n} - I_{c,n}\right\|^2, \tag{4}$$

which we solve using SGD with the Adam optimizer [19]. We implement forward propagation using PyTorch and let its autodiff capabilities track gradients during the optimization process. The initial holograms h_c and h_p are then updated within a fixed number of iterations, after which we obtain the final holograms to be displayed on the mobile phone screen and the SLM. We provide a Python repository with implementations in [20].

The optical setup for the experiments is shown in Fig. 1. We used an iPhone 12 Pro as the mobile phone screen (manufactured by Apple, pixel count: 2532×1170, pixel pitch: 55.2 µm). The light emitted from the iPhone passed through a polarizer, a beam splitter, and was reflected by a phase-modulation SLM (GAEA-2.1 LCOS SLM manufactured by HOLOEYE Photonics, pixel count: 3840×2160, pixel pitch: 3.74 µm) located at 25 cm from the screen. The reflected light propagated toward the target depth planes, located between 8 cm (layer 1) and 10 cm (layer N), where the target color images were displayed, and we recorded them using a color image sensor (45.7MP FX-Format BSI CMOS Sensor from a Nikon Z7 II camera, manufactured by Nikon, pixel count: 8256×5504, pixel pitch: 4.35 µm).

In the numerical propagation, we set $\lambda_1 = 620$ nm, $\lambda_2 = 532$ nm, and $\lambda_3 = 461$ nm, which we determined by measuring the spectrum of a white image displayed on the iPhone using a spectrometer (AvaSpec-Mini2048CL manufactured by Avantes), as shown in Fig. 1. We set $\lambda_{ref} = 520$ nm according to the calibration of the SLM, the intensity hologram pixel pitch $\Delta x_s = 55.2$ µm and pixel count $M_s = 512^2$, and the hologram pixel pitch $\Delta x_{\mathrm{SLM}} = 3.74$ µm and pixel count $M_{\mathrm{SLM}} = 2048^2$. We calculated the target number of pixels M_n for the operator interp_n, which was different for each depth layer n, as:

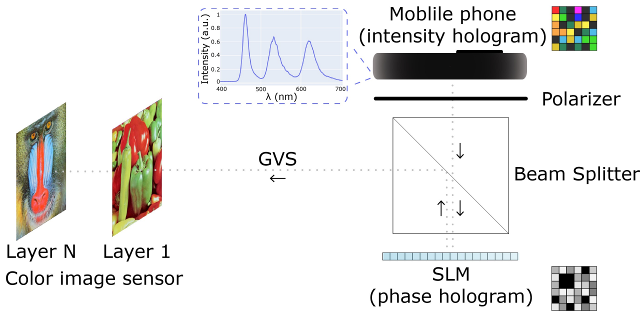

Fig. 1. Numerical and optical setup used in the experiments. The intensity hologram is displayed on an iPhone 12 Pro, and the phase-only hologram is displayed on the SLM. The forward propagation is computed using GVS [16].

$$M_n = \frac{\Delta x_s}{\Delta x_{\mathrm{SLM}}} \cdot \frac{z_s}{z_n} \cdot M_s. \tag{5}$$

To avoid aliasing effects from circular convolution, we applied zero padding in the spatial domain to each image before propagation. For numerical stability, we normalized all the computed intensities w.r.t. the first layer of each color channel, $\hat{I}_{c,n-norm} = \hat{I}_{c,n}/\max|\hat{I}_{c,1}|$. We achieved better results by narrowing the observation region, cropping the resulting images to 1024^2 before comparing with the target images. We set the learning rate of the Adam optimizer to 0.01, the number of iterations to 3000, and we ensured that the intensity images remained in the $[0, 1]$ interval by element-wise clipping at every iteration. We note that the physical conditions of the optical and numerical setups that we use here fall within the validity range of the paraxial and Fresnel approximations, as detailed in Supplement 1. This is a trade-off inherent in GVS that limits its generality in exchange for high computational efficiency. Care must be taken when minimizing propagation distances to achieve a more compact setup, or when modeling non-paraxial applications, in which approaches based on more general propagators, such as CP-R, can achieve better results.

Figure 2 shows the numerical and optical results of the proposed method when optimizing for two different color images at two different depths, and a comparison with the results of CP-R with FresnelCGH taken from Horisaki *et al.* [15]. We used the standard color images of *peppers* and *baboon* in the first and second layers. The peak signal-to-noise ratios (PSNRs) of the red, green, and blue channels between the target images and the numerical GVS reconstructions for the first layer were 23.0 dB, 21.2 dB, and 18.3 dB, respectively, achieving a luminance PSNR (Y-PSNR) of 22.1 dB. Those on the second layer were 21.0 dB, 20.2 dB, and 19.0 dB, respectively, achieving a Y-PSNR of 21.1 dB. In the numerical and optical GVS reconstructions, speckle noise and zeroth-order light are absent because we use partially coherent light. Moreover, the random noise observed in the numerical and optical CP-R results is absent in the GVS results, indicating a higher reconstruction quality. Possible reasons for discrepancies between the optical and numerical results include approximating single wavelengths per color, the SLM fill factor, SLM nonlinearity, and phase quantization. Visualization 1 shows a continuous transition of the optical GVS reproduction from the first to the second layer.

It is important to note here that the comparison presented in Fig. 2 is not entirely equivalent, as the resolutions of the

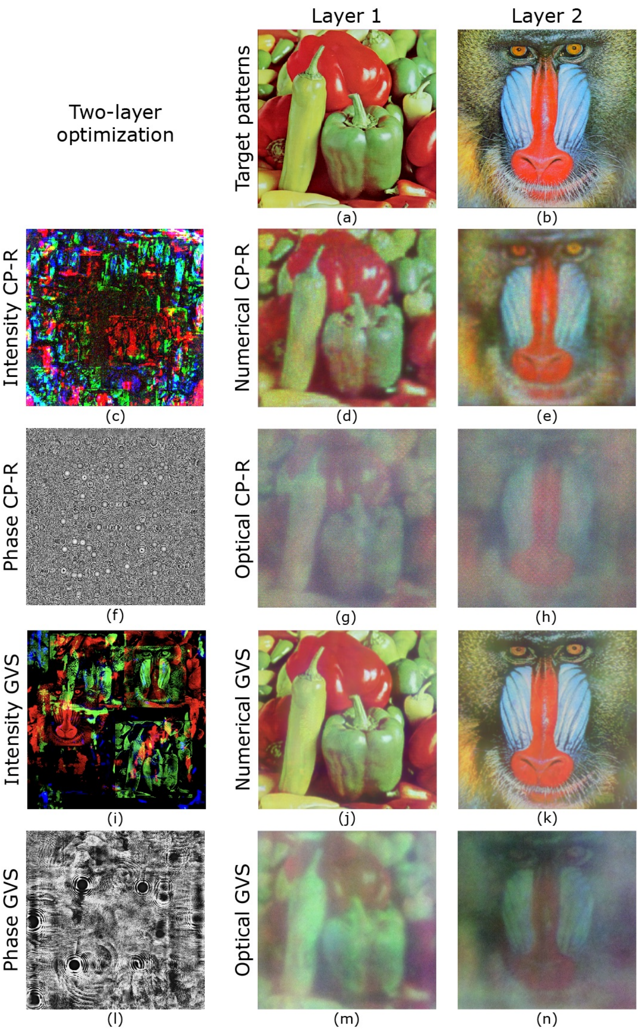

Fig. 2. Two-layer optimization using GVS and CP-R Fresnel-CGH. (a) and (b) Target color images, (c) and (f) intensity and phase-only holograms obtained with CP-R (resolutions of 224^2 and 1272×1024, respectively), (d) and (e) numerical reproductions from CP-R, (g) and (h) optical reproductions from CP-R, (i) and (l) intensity and phase-only holograms obtained with GVS (resolutions of 512^2 and 2048^2, respectively), (j) and (k) numerical reproductions from GVS, (m) and (n) optical reproductions from GVS. Adapted with permission from [15].

SLM and the mobile phone screen used for the CP-R experiment were lower than those for the GVS experiment. Also, the SLM used in GVS had a smaller pixel pitch, and the propagation distances between the optical elements differed. For fair comparison with CP-R, we ran the numerical GVS experiment using the same optical parameters as those reported for FresnelCGH in [15]. Table 1 presents the PSNRs obtained with GVS and those reported in the original work, along with the optimization execution times for the two methods implemented in PyTorch. GVS ran for 500 iterations, and CP-R ran for 1500 iterations, due to GVS's better convergence rate during optimization. We conducted these experiments using an NVIDIA A100 GPU. These results demonstrate that our method outperforms CP-R in this experiment, achieving a 15× speedup and a higher numerical PSNR in most cases. This speedup agrees with the theoretical computational complexity per iteration of the two methods,

Table 1. Numerical Comparison between CP-R[a] and GVS[b]

	CP-R	GVS
PSNR L1, R | G | B (dB)	19.6 | 18.9 | 18.9	**21.7**[c] | **19.9** | **22.4**
Y-PSNR L1 (dB)	19.4	**22.0**
PSNR L2, R | G | B (dB)	18.0 | **17.7** | 16.1	**19.6** | 17.3 | **17.4**
Y-PSNR L2 (dB)	18.7	**18.8**
Execution time (s)	537	**35**

[a]Reference method that requires M propagations [15].
[b]Proposed method that requires one propagation [16]. L1, layer 1 (Peppers); L2, layer 2 (Baboon).
[c]Bold values indicate higher PSNR and lower execution time.

where GVS has $O(N^2 \log N)$ and CP-R has $O(MN^2 \log N)$ [16]. We also note that the numerical reconstructions were generated using the same GVS model, so potential discrepancies with a more general propagation model are not considered here.

Thanks to GVS's computational efficiency, we were able to perform a focal stack optimization to further test the display's 3D capabilities. In the following experiment, we used an image featuring three different characters and a depth mask with distinct depths for each character. For each depth, we propagated the image portion corresponding to the depth mask using coherent angular spectrum to obtain a complex-amplitude hologram for each color channel. Next, we backpropagated each hologram using AS-GVS at 20 different depth layers, obtaining color images for optimization with reduced speckle noise stemming from coherent propagation. In this case, we used a learning rate of 0.05 and 200 iterations. Figure 3(a) presents the original image containing the three characters A, Z, and 9, which is propagated according to the depth mask shown in Fig. 3(d). Figures 3(b) and 3(e) show the resulting target color images at depth layers 1 and 20. After focal stack optimization, we obtained intensity and phase holograms for display on the screen and the SLM, as shown in Figs. 3(c) and 3(f). The numerical reconstructions in the front, middle, and back planes are shown in Figs. 3(g)–3(i), and the optical reconstructions in the front, middle, and back planes are shown in Figs. 3(j)–3(l). These results demonstrate the proposed method's ability to achieve 3D focus–defocus effects with an incoherent light source, simplifying the optical setup.

Next, we performed focal stack optimization using a continuous depth map. We took a picture of two small decorative objects: a blue fish and a golden cat; we used a free tool to estimate a depth map from the RGB image using generative AI [21]. We set M_{SLM} to 2600×2160, and we cropped the resulting intensities to 940×842. The rest of the process was the same as in the previous experiment, using AS-GVS and 20 different depth layers. Figure 4(a) shows the original captured image, which is propagated according to the continuous depth map presented in Fig. 4(b). After running the optimization process, we obtained the intensity and phase holograms shown in Figs. 4(c) and 4(d). Figures 4(e)–4(h) present four different color images from the focal stack, and Figs. 4(i)–4(l) present the numerical reconstructions for the corresponding depths. These numerical results show the 3D focusing–defocusing effect, with different parts of the cat and fish objects being focused at different depth layers. Visualization 2 shows a transition of the GVS numerical reconstructions for all layers. Optical reconstructions are provided in Supplement 1.

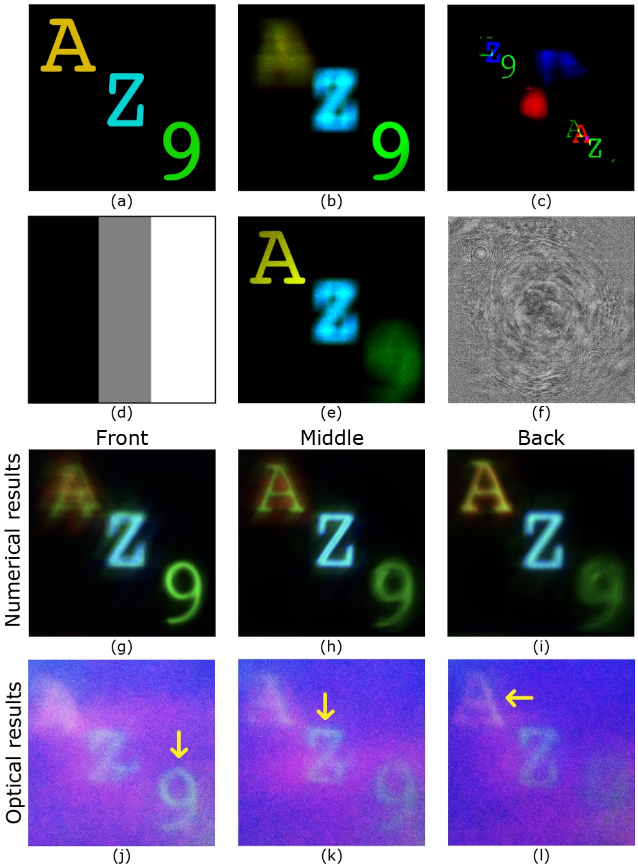

Fig. 3. Focal stack optimization featuring the characters A, Z, and 9 with 20 depth layers. (a) Original image, (d) depth mask, (b) and (e) layers 1 and 20 of the target focal stack, (c) optimized intensity hologram, (f) optimized phase hologram, (g)–(i) numerical results at different depths, and (j)–(l) optical results at different depths. Yellow arrows indicate the character in focus.

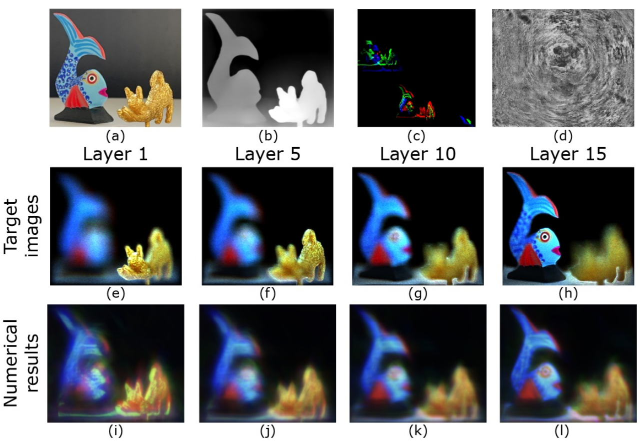

Fig. 4. Focal stack optimization featuring the cat and fish objects with 20 depth layers. (a) Original image, (b) AI-generated depth mask, (c) optimized intensity hologram, (d) optimized phase hologram, (e)–(h) target focal stack at different depth layers, and (i)–(l) numerical results at different depth layers.

To conclude, this work introduces a new CGH method that accounts for partial spatial coherence by using the GVS framework in a simple optical system. We performed SGD optimization to obtain an intensity RGB hologram and a phase hologram, which were displayed on a mobile phone screen and on a phase-only SLM. We demonstrated the effectiveness of our method in a series of numerical and optical experiments. First, we conducted a two-layer optimization, demonstrating accurate RGB numerical and optical reproduction at two different depth layers. We also compared the proposed method with the CP-R FresnelCGH method from Horisaki *et al.* [15], achieving 15× speedup and a higher numerical PSNR for most color channels and depth layers. This efficiency enabled us to perform more complex experiments, such as focal stack optimization with 20 depth layers, demonstrating our method's ability to achieve 3D focus–defocus effects in both numerical and optical reproductions.

As future work, we aim to enhance the display's performance by developing a more accurate physical model that accounts for temporal coherence and by implementing more sophisticated optimization algorithms better suited to CGH. We also expect that the introduction of deep learning algorithms will reduce the computation time required to generate holograms while maintaining high-quality reproductions. We believe that this work represents a significant step toward enabling low-cost, high-quality holographic displays for next-generation augmented and virtual reality applications.

Funding. Research Foundation - Flanders (12ZQ223N, G089424N, G0A3O24N, G098426N).

Disclosures. The authors declare no conflicts of interest.

Data availability. Experimental data may be obtained from the authors upon request. For numerical implementations, see [20].

Supplemental document. See Supplement 1 for supporting content.

REFERENCES

1. K. Matsushima, *Introduction to Computer Holography: Creating Computer-Generated Holograms as the Ultimate 3D Image* (Springer Nature, 2020).
2. M. K. Kim, *Digital Holographic Microscopy, Vol. 162 of Springer Series in Optical Sciences* (Springer, 2011).
3. P. S. Salter and M. J. Booth, Light. Sci. Appl. **8**, 110 (2019).
4. Y. Yang, Y. Ren, M. Chen, *et al.*, Adv. Photonics **3**, 034001 (2021).
5. K. Khare, *Fourier Optics and Computational Imaging* (John Wiley & Sons Ltd., 2016).
6. P.-A. Blanche, Light Adv. Manuf. **2**, 28 (2021).
7. D. Blinder, T. Birnbaum, T. Ito, *et al.*, Light Adv. Manuf. **3**, 35 (2022).
8. J. W. Goodman, *Statistical Optics*, 2nd ed. (Wiley, 2015).
9. S. Lee, D. Kim, S.-W. Nam, *et al.*, Sci. Rep. **10**, 18832 (2020).
10. Y. Peng, S. Choi, J. Kim, *et al.*, Sci. Adv. **7**, eabg5040 (2021).
11. Z. Zhao, J. Duan, and J. Liu, Opt. Commun. **507**, 127604 (2022).
12. D. Pi, J. Liu, and Y. Wang, Light Sci. Appl. **11**, 231 (2022).
13. R. Horisaki, T. Aoki, Y. Nishizaki, *et al.*, Opt. Lett. **47**, 613 (2022).
14. R. Suda, M. Naruse, and R. Horisaki, Opt. Lett. **47**, 3844 (2022).
15. O. Shigematsu, M. Naruse, and R. Horisaki, Opt. Lett. **49**, 1876 (2024).
16. M. Montoya, M. J. Lopera, Y. Nie, *et al.*, Opt. Express **33**, 46313 (2025).
17. T. Shimobaba, J. Weng, T. Sakurai, *et al.*, Comput. Phys. Commun. **183**, 1124 (2012).
18. H. Wang and R. Piestun, Optica **5**, 1220 (2018).
19. D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *3rd International Conference on Learning Representations* (2015).
20. M. Montoya, "Partially coherent propagator in Python," GitHub (2025) https://github.com/mmonto952/gvs-propagator.
21. ArtificialStudio, "Free image depth map," (2025) https://app.artificial-studio.ai/tools/image-depth-map-generator.