Optics Letters

High-speed computer-generated holography using an autoencoder-based deep neural network

JIACHEN WU, KEXUAN LIU, XIAOMENG SUI, AND LIANGCAI CAO*

State Key Laboratory of Precision Measurement Technology and Instruments, Department of Precision Instruments, Tsinghua University, Beijing 100084, China
**Corresponding author: clc@tsinghua.edu.cn*

Received 18 March 2021; revised 14 May 2021; accepted 17 May 2021; posted 18 May 2021 (Doc. ID 425485); published 14 June 2021

Learning-based computer-generated holography (CGH) provides a rapid hologram generation approach for holographic displays. Supervised training requires a large-scale dataset with target images and corresponding holograms. We propose an autoencoder-based neural network (holoencoder) for phase-only hologram generation. Physical diffraction propagation was incorporated into the autoencoder's decoding part. The holoencoder can automatically learn the latent encodings of phase-only holograms in an unsupervised manner. The proposed holoencoder was able to generate high-fidelity 4K resolution holograms in 0.15 s. The reconstruction results validate the good generalizability of the holoencoder, and the experiments show fewer speckles in the reconstructed image compared with the existing CGH algorithms.

https://doi.org/10.1364/OL.425485

Holographic display technology utilizes spatial light modulators (SLMs) to control the light field and achieve an arbitrary intensity distribution. The major challenge in computer-generated holograms (CGHs) is that the existing SLMs cannot realize simultaneous amplitude and phase modulations. Because phase modulation has higher optical efficiency and the reconstructions have no interference from the conjugate image, phase-only liquid crystal-based SLMs are widely used for holographic display and optical field generation.

Calculating the phase-only CGHs is an ill-posed problem because the number of solutions is uncertain. Common iterative projection algorithms such as the Gerchberg–Saxton (G–S) [1,2], or nonconvex optimization methods [3,4] require iterative procedures and are time-consuming. Several noniterative methods have been proposed to achieve a fast calculation of CGHs, such as spatial multiplexing [5], error diffusion [6,7], and phase encoding [8–12]. However, these methods involve complex amplitude modulation at the expense of spatial resolution and are prone to the presence of artifacts in the reconstructed images.

Recently, deep learning has become a powerful tool for solving such challenging problems owing to its versatile and efficient performance. Learning-based CGH methods can provide both high-speed and high-quality image reconstruction [13–15] and realize manifold types of holograms [16–18]. Early learning-based methods required an iterative method to generate holograms as the training set in advance [19], which has a high computational cost and is impractical for large datasets. Several methods have adopted a forward propagation model to create a dataset, which could avoid the iterative generation of holograms [20,21]. In contrast, unsupervised learning dispenses with the need for a labeled dataset, and the performance does not limit the quality of the labeled dataset, which is an appropriate training method for learning-based CGH methods. The existing unsupervised strategy compared the virtual reconstruction of the hologram with the target intensity pattern to achieve unsupervised learning [22]. However, the network maps the target amplitude to the complex field in the image plane without explicitly optimizing the phase of the hologram. This indirect training strategy may not obtain an optimal hologram.

In this Letter, we proposed a holoencoder, a neural network architecture for calculating CGHs based on an autoencoder. An autoencoder is an unsupervised neural network that originally learns how to efficiently compress and encode data. Because the recording and reconstruction of holograms can be considered as the encoding and decoding of images, the end-to-end model of the holographic display can be represented by an autoencoder. In contrast to general autoencoders, the decoding part of the holoencoder is a determined optical propagation model without learnable parameters. By incorporating the physical model into the training of the network, the holoencoder can achieve both searching for the hologram of the target image and learning the mapping from the target image to the hologram. The holoencoder is capable of generating high-fidelity 4K holograms in 0.15 s.

The architecture of the holoencoder is shown in Fig. 1. The encoder part of the holoencoder is implemented using the U-Net architecture, which has four downsampling and corresponding upsampling blocks. Each block is composed of two sets of batch normalization, nonlinearity (ReLU), and a convolutional layer stacked one above the other. The output layer of the U-Net is a tanh function that limits the phase value in the range $[-\pi, \pi]$.

A Fresnel transform method was adopted for the decoding part of the holoencoder. A spherical phase $\varphi(x, y)$ was first added to the hologram. Then, the intensity of the spectrum of the field in the hologram plane is the reconstructed image. The process can be formulated as

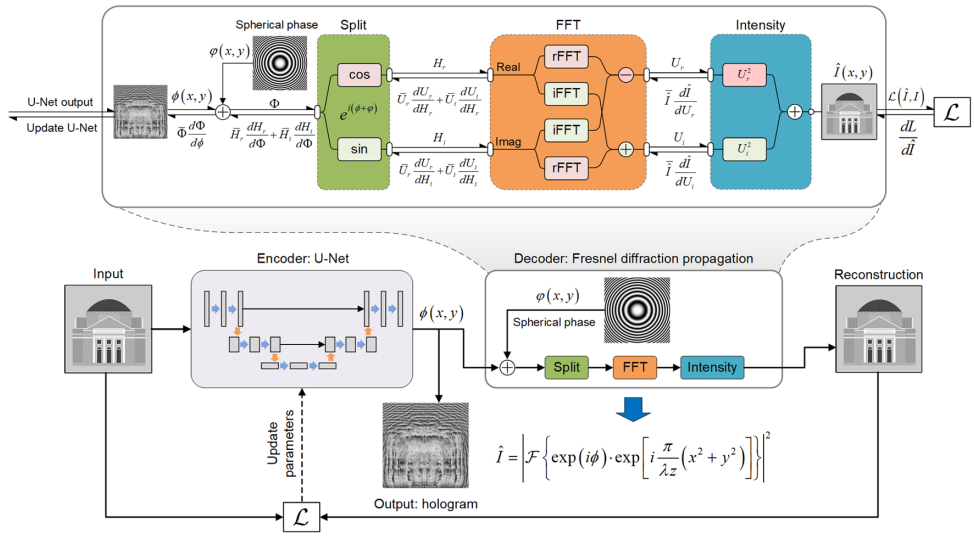

Fig. 1. Architecture of the holoencoder. A target image is first input into a U-Net, which outputs the predicted phase. Then, the phase is input into the decoder which realizes the Fresnel diffraction propagation in the network form. During the training period, the *autodiff* technique is used to evaluate the derivatives of each elementary arithmetic node. The formulas below the reverse arrows indicate the derivatives of loss function with respect to the variables above the forward arrows. Note the variable with a bar $\bar{v}$ denoted the derivative dL/dv.

$$I(x, y) = \left| \mathcal{F}\left\{ \exp\left[i\phi(x_o, y_o) \right] \exp\left[i\frac{\pi}{\lambda z}(x_o^2 + y_o^2) \right] \right\} \right|^2, \quad \textbf{(1)}$$

where $\phi(x_o, y_o)$ is the output of the U-Net, indicating the phase-only hologram, and $\mathcal{F}\{\cdot\}$ denotes the Fourier transform. Note that the exponential coefficient of the Fresnel diffraction formula is ignored because it does not affect the intensity distribution. Compared with the angular spectrum method, the Fresnel transform method involves a single Fourier transform and does not require a zero-padding operation, which saves many computational resources. To deal with the backpropagation of complex numbers in a neural network, the phase is first split into real and imaginary parts of the complex amplitude using Euler's formula. Then, the Fourier transform is decomposed into two individual transformations for the real part "rFFT" and imaginary part "iFFT." Finally, the reconstructed intensity is the sum of the squares of the real and imaginary parts. By utilizing automatic differentiation, the loss can be propagated backward to the encoder part, and the learnable parameters of the U-Net can be updated during the training process.

We used a combination of negative Pearson correlation coefficient (NPCC) and perceptual loss [23] to train the holoencoder. An optional regularizer on the reconstructed phase patterns can enforce constraints on the phase variations.

$$\mathcal{L}(\hat{I}, I) = \mathcal{L}_{\text{NPCC}}(\hat{I}, I) + \tau \mathcal{L}_{\text{percep}}(\hat{I}, I) + \gamma \|\nabla \Phi\|_p, \quad \textbf{(2)}$$

$$\mathcal{L}_{\text{NPCC}}(X, Y) = (-1) \times \frac{\sum_i^n (X_i - \bar{X})(Y_i - \bar{Y})}{\left\{ \sum_i^n (X_i - \bar{X})^2 \sum_i^n (Y_i - \bar{Y})^2 \right\}^{1/2}}, \quad \textbf{(3)}$$

$$\mathcal{L}_{\text{percep}}(X, Y) = \|P(X) - P(Y)\|_2^2, \quad \textbf{(4)}$$

where $I = \sqrt{|I|} \cdot \exp(i\Phi)$, and $P(\cdot)$ represent a transform to a perceptual feature space. τ and γ are the relative weights of the perceptual loss component and regularization term, respectively. The NPCC loss guarantees linear amplification

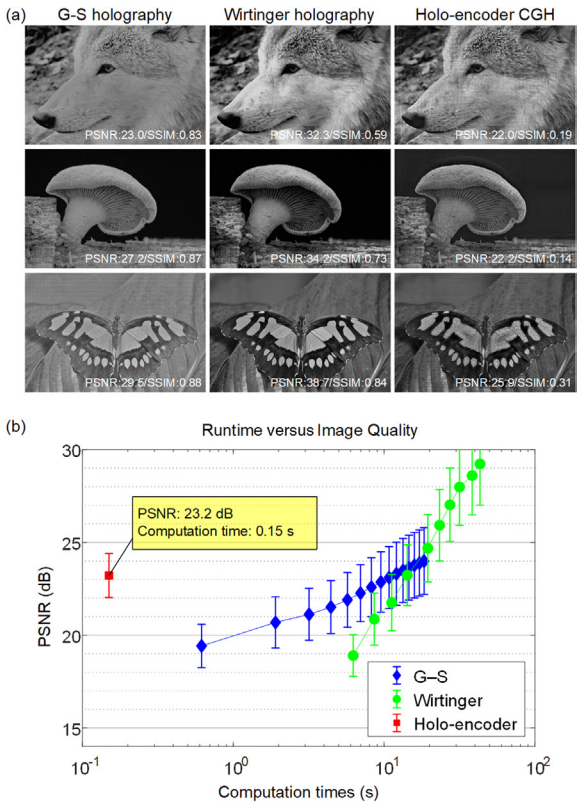

Fig. 2. (a) Numerical reconstructions for G–S holography, Wirtinger holography, and holoencoder CGH. The network is trained on 800 2 K resolution samples, and then tested on 100 samples. The three images are selected from the validation set. (b) Evaluation of algorithm running time and reconstruction quality. The markers indicate the mean values for the 100 samples, and the length for the bar represent the standard deviation.

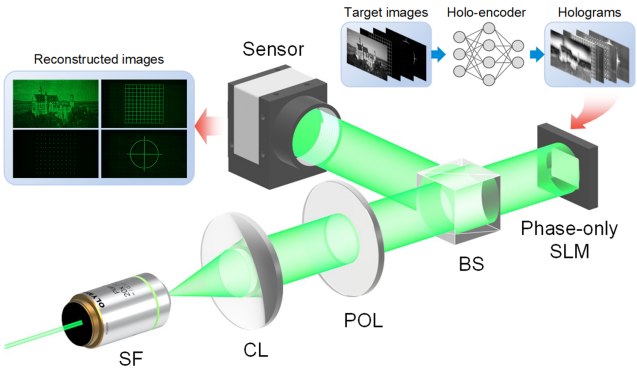

Fig. 3. Experimental arrangement. SF, spatial filter; CL, collimating lens; POL, linear polarizer; BS, beam splitter; SLM, spatial light modulator.

and bias-free reconstruction, which increases the convergence probability. The phase regularizer can enforce constraints on phase variations. When $p = 1$, the regularizer is equivalent to the total variation regularizer, which produces a smooth phase in the reconstructed field, and the speckle can be reduced.

In the numerical simulation, the pixel pitch of the hologram is 3.74 μm and the resolution is 3840 × 2160. The illumination wavelength is 532 nm, and the propagation distance is 160 mm. The network is trained for 10 epochs using an Adam optimizer. The training images are obtained from the DIV2K dataset [24]. In contrast to the conventional training process, an additional 10 iterations are involved in training for each batch. The additional iterations make the hologram closer to the optimal solution and enable the encoding part of the network to learn more precise features. The trained network model is shown in Code 1, Ref. [25] for reproducibility.

The numerical reconstructions are presented in Fig. 2. All the algorithms were run on the same workstation with a Xeon CPU E5-2650 (2.20 GHz) and 128 GB of RAM. In particular, an NVIDIA Quadro GV100 GPU was used for the prediction of the holoencoder. We compared the holoencoder with two iterative algorithms: G–S holography and Wirtinger holography [4]. For iterative methods, the reconstruction quality increases with the computation time. See Visualization 1 for the dynamic iterative process. Wirtinger holography can achieve rare high-quality numerical reconstruction, but the computation time is the longest. The holoencoder breaks the trade-off between computation time and reconstruction quality and provides both short computation time and moderate reconstruction quality. Note that the deviation of PSNR and SSIM is caused by artifacts such as ringing and stripes. The SSIM is sensitive to structures that do not exist in the original image. These artifacts are visually unnoticeable but significantly reduce the SSIM value. For the generation of 4K resolution holograms, the holoencoder only takes 0.15 s and achieves an average reconstruction quality of 23.2 dB PSNR. However, the numerical reconstruction is the discrete sampling of the reconstruction intensity, which does not represent the practical display results. The turbulent phase in the reconstructed field could result in speckle reconstruction, as shown in the subsequent experimental results.

The experimental setup is illustrated in Fig. 3. The SLM used was Holoeye GAEA-2. The experimental configuration had the same parameters as the numerical simulation. No lens is needed in the display optical path except for the collimating lens, which is suitable for naked-eye display.

The experimental results are shown in Fig. 4. Different types of data were used to test the generalizability of the proposed network. Three binary patterns of *grid*, *dot matrix*, and *cross* were reconstructed, as shown in Fig. 4(a). These patterns can be reconstructed without distortion. The results show that the holoencoder has great potential for the design of diffracted optical elements. We then compared the reconstruction quality of G–S holography, Wirtinger holography, and holoencoder CGH for grayscale images, as shown in Fig. 4(b). Although the iterative algorithm has a good appearance in numerical reconstruction, the speckles in the physical reconstruction degrade the image quality. Owing to the regularization term in the loss function, the holoencoder CGH shows less speckle noise in the reconstructed intensity. The phase distributions in the reconstructed plane shown in the fourth row of Fig. 4(b) clearly illustrate that the holoencoder could generate a smoother phase distribution compared with the two other methods.

To the best of our knowledge, this is the first study in which a neural network was used to generate 4K resolution holograms, and the content of holographic display could be an arbitrary image rather than a simple character set. Owing to modern

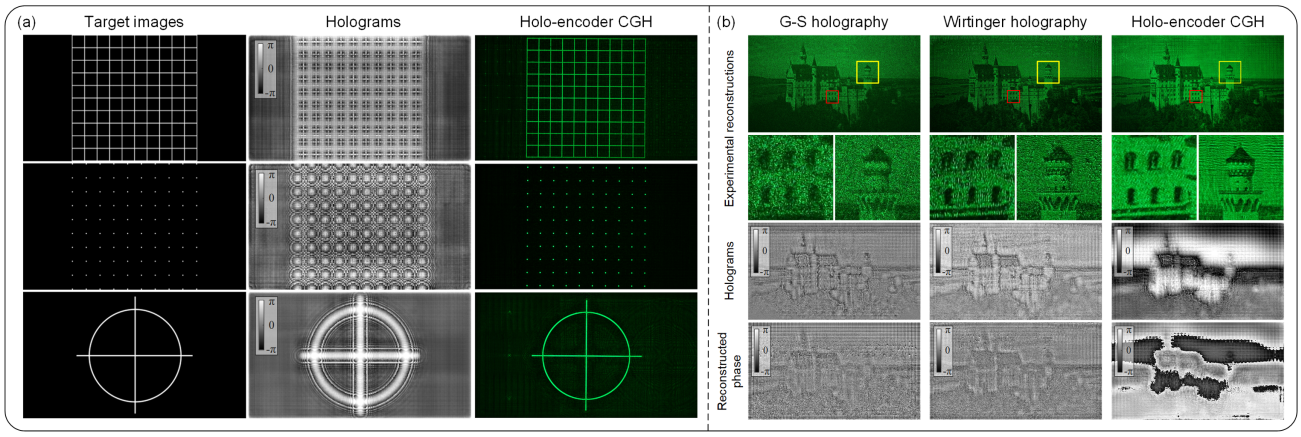

Fig. 4. Experimental results. (a) Holoencoder CGH for binary patterns. (b) Comparison of grayscale image reconstruction quality. Holoencoder CGH could achieve less speckle noise than G–S and Wirtinger methods. The second row is the close-up of red and yellow boxes in the first row. The third row shows the corresponding holograms. The fourth row shows the phase distributions in the reconstruction plane.

machine-learning techniques, high-speed generation of large holograms can be achieved. It is desired that under the collaborative optimization of both the network structure and computing platform, the computing speed could be further improved. The proposed method is applicable to real-time 3D displays, augmented reality displays [26], and the design of diffraction optical elements.

Funding. National Natural Science Foundation of China (61775117, 62035003).

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

REFERENCES

1. R. W. Gerchberg, Optik **35**, 237 (1972).
2. Y. Peng, X. Dun, Q. Sun, and W. Heidrich, ACM Trans. Graph. **36**, 191 (2017).
3. J. Zhang, N. Pégard, J. Zhong, H. Adesnik, and L. Waller, Optica **4**, 1306 (2017).
4. P. Chakravarthula, Y. Peng, J. Kollin, H. Fuchs, and F. Heide, ACM Trans. Graph. **38**, 213 (2019).
5. A. Jesacher, C. Maurer, A. Schwaighofer, S. Bernet, and M. Ritsch-Marte, Opt. Express **16**, 2597 (2008).
6. P. W. M. Tsang and T.-C. Poon, Opt. Express **21**, 23680 (2013).
7. H. Pang, J. Wang, M. Zhang, A. Cao, L. Shi, and Q. Deng, Opt. Express **25**, 14323 (2017).
8. O. Mendoza-Yero, G. Mínguez-Vega, and J. Lancis, Opt. Lett. **39**, 1740 (2014).
9. Y. Qi, C. Chang, and J. Xia, Opt. Express **24**, 30368 (2016).
10. A. Maimone, A. Georgiou, and J. S. Kollin, ACM Trans. Graph. **36**, 85 (2017).
11. X. Sui, Z. He, G. Jin, D. Chu, and L. Cao, Opt. Express **29**, 2597 (2021).
12. X. Li, J. Liu, J. Jia, Y. Pan, and Y. Wang, Opt. Express **21**, 20577 (2013).
13. Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, ACM Trans. Graph. **39**, 185 (2020).
14. S. Choi, J. Kim, Y. Peng, and G. Wetzstein, Optica **8**, 143 (2021).
15. L. Shi, B. Li, C. Kim, P. Kellnhofer, and W. Matusik, Nature **591**, 234 (2021).
16. H. Goi, K. Komuro, and T. Nomura, Appl. Opt. **59**, 7103 (2020).
17. J. Lee, J. Jeong, J. Cho, D. Yoo, B. Lee, and B. Lee, Opt. Express **28**, 27137 (2020).
18. S. Jiao, Z. Jin, C. Chang, C. Zhou, W. Zou, and X. Li, Appl. Sci. **8**, 1258 (2018).
19. S. Yamauchi, Y. Chen, and Z. Nakao, in *2nd International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No. 98EX111)* (1998), Vol. **223**, pp. 220–223.
20. A. Khan, Z. Zhijiang, Y. Yu, M. A. Khan, K. Yan, and K. Aziz, Complexity **2021**, 6662161 (2021).
21. R. Horisaki, R. Takagi, and J. Tanida, Appl. Opt. **57**, 3859 (2018).
22. M. H. Eybposh, N. W. Caira, M. Atisa, P. Chakravarthula, and N. C. Pégard, Opt. Express **28**, 26636 (2020).
23. J. Johnson, A. Alahi, and L. Fei-Fei, in *European Conference on Computer Vision* (Springer, 2016), pp. 694–711.
24. E. Agustsson and R. Timofte, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), pp. 126–135.
25. J. Wu and L. Cao, "High-speed computer-generated holography (CGH) using autoencoder-based deep neural network," GitHub (2021) [accessed 14 May 2021], https://github.com/THUHoloLab/Holo-encoder.
26. A. Markman, X. Shen, H. Hua, and B. Javidi, Opt. Lett. **41**, 297 (2016).