

Towards real-time photorealistic 3D holography with deep neural networks

<https://doi.org/10.1038/s41586-020-03152-0>

Liang Shi^{1,2}✉, Beichen Li^{1,2}, Changil Kim^{1,2}, Petr Kellnhofer^{1,2} & Wojciech Matusik^{1,2}✉

Received: 22 April 2020

Accepted: 21 December 2020

Published online: 10 March 2021

 Check for updates

The ability to present three-dimensional (3D) scenes with continuous depth sensation has a profound impact on virtual and augmented reality, human–computer interaction, education and training. Computer-generated holography (CGH) enables high-spatio-angular-resolution 3D projection via numerical simulation of diffraction and interference¹. Yet, existing physically based methods fail to produce holograms with both per-pixel focal control and accurate occlusion^{2,3}. The computationally taxing Fresnel diffraction simulation further places an explicit trade-off between image quality and runtime, making dynamic holography impractical⁴. Here we demonstrate a deep-learning-based CGH pipeline capable of synthesizing a photorealistic colour 3D hologram from a single RGB-depth image in real time. Our convolutional neural network (CNN) is extremely memory efficient (below 620 kilobytes) and runs at 60 hertz for a resolution of $1,920 \times 1,080$ pixels on a single consumer-grade graphics processing unit. Leveraging low-power on-device artificial intelligence acceleration chips, our CNN also runs interactively on mobile (iPhone 11 Pro at 1.1 hertz) and edge (Google Edge TPU at 2.0 hertz) devices, promising real-time performance in future-generation virtual and augmented-reality mobile headsets. We enable this pipeline by introducing a large-scale CGH dataset (MIT-CGH-4K) with 4,000 pairs of RGB-depth images and corresponding 3D holograms. Our CNN is trained with differentiable wave-based loss functions⁵ and physically approximates Fresnel diffraction. With an anti-aliasing phase-only encoding method, we experimentally demonstrate speckle-free, natural-looking, high-resolution 3D holograms. Our learning-based approach and the Fresnel hologram dataset will help to unlock the full potential of holography and enable applications in metasurface design^{6,7}, optical and acoustic tweezer-based microscopic manipulation^{8–10}, holographic microscopy¹¹ and single-exposure volumetric 3D printing^{12,13}.

Holography is the process of encoding a light field¹⁴ as an interference pattern of variations in phase and amplitude. When properly lit, a hologram diffracts an incident light into an accurate reproduction of the original light field, producing a true-to-life recreation of the recorded three-dimensional (3D) objects¹. The reconstructed 3D scene presents accurate monocular and binocular depth cues, which are difficult to simultaneously achieve in traditional displays. Yet, creating photorealistic computer-generated holograms (CGHs) power-efficiently and in real time remains an unsolved challenge in computational physics. The primary challenge is the tremendous computational cost required to perform Fresnel diffraction simulation for every object point in a continuous 3D space. This remains true despite extensive efforts to design various digital scene representations^{3,15–18} and algorithms for the detection of light occlusions¹⁹.

The challenging task of efficient Fresnel diffraction simulation has been tackled by explicitly trading physical accuracy for computational speed. Hand-crafted numerical approximations based

on look-up tables of precomputed elemental fringes^{20–22}, multilayer depth discretization^{23–25}, holographic stereograms^{26–29}, wavefront recording plane (alternatively intermediate ray sampling planes)^{30,31} and horizontal/vertical-parallax-only modelling³² were introduced at a cost of compromised image quality. Harnessing rapid advances of graphics processing unit (GPU) computing, the non-approximative point-based method (PBM) recently produced colour and textured scenes with per-pixel focal control at a speed of seconds per frame². Yet, PBM simulates Fresnel diffraction independently for every scene point, and thus does not model occlusion. This prevents accurate recreation of complex 3D scenes, where the foreground will be severely contaminated by ringing artefacts due to the unoccluded background (Extended Data Fig. 1d). This lack of occlusion is partially addressed by light-field rendering^{3,29,33}. However, this approach incurs substantial rendering and data storage overhead, and the occlusion is only accurate within a small segment (holographic element) of the entire hologram. Adding a per-ray visibility test during

¹Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. ²Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge, MA, USA. [✉]e-mail: liangs@mit.edu; wojciech@mit.edu

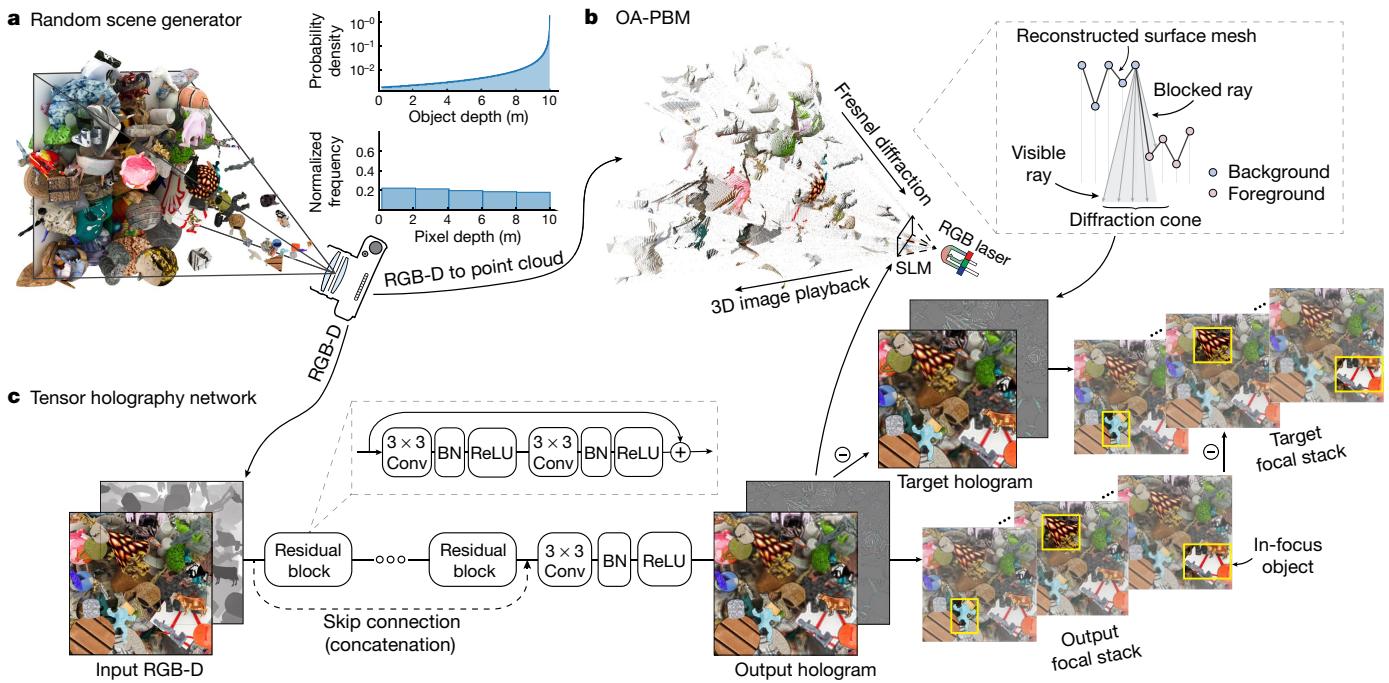


Fig. 1 | Tensor holography workflow for learning Fresnel holograms from RGB-D images. **a**, A custom ray-tracer renders an RGB-D image of a random scene. The meshes are distributed exponentially along the depth axis and the resulting pixel depth distribution is statistically uniform. **b**, An OA-PBM reconstructs a triangular surface mesh from the point cloud defined by the RGB-D image. During Fresnel diffraction simulation, wavefronts carried by the

occluded rays are excluded from the hologram calculation. **c**, A fully convolutional residual network synthesizes a Fresnel hologram from the same RGB-D image. The network is optimized against the target hologram using a data fidelity loss and a focal stack loss. BN, batch renormalization. The minus symbol indicates error minimization. The plus symbol denotes layer concatenation along the colour channel. Conv, convolution.

Fresnel diffraction simulation ideally resolves the problem, yet the additional cost of an occlusion test, access for neighbour points and conditional branching slow down the computation. This quality-speed trade-off is a trait shared by all existing physically based approaches and fundamentally limits the practical deployment of dynamic holographic displays.

We resolve this dilemma with a physics-guided deep-learning approach, dubbed tensor holography. Tensor holography avoids the explicit approximation of Fresnel diffraction and occlusion, but imposes underlying physics to train a convolutional neural network (CNN) as an efficient proxy for both. It exploits the fact that propagating a wave field to different distances is equivalent to convolving the same wave field with Fresnel zone plates of different frequencies. As the zone plates are radially symmetric and derived from a single basis function using different propagation distances, our network accurately approximates them through successive application of a set of learned 3×3 convolution kernels. This reduces diffraction simulation from spatially varying large kernel convolutions to a set of separable and spatially invariant convolutions, which runs orders of magnitude faster on GPUs and application-specific integrated circuits (ASICs) for accelerated CNN inference. Our network further leverages nonlinear activation (that is, ReLU or the rectified linear unit³⁴) in the CNN to handle occlusion. The nonlinear activation selectively distributes intermediate results produced through forward propagation, thus stopping the propagation of occluded wavefronts. We note that although the mathematical model of the CNN is appealing, the absence of a large-scale Fresnel hologram dataset and an effective training methodology impeded the development of any learning-based approach. Despite recent successful adoption of CNNs for phase retrieval^{35–37} and for recovering in-focus images or extended depth-of-field images from optically recorded digital holograms^{38–40}, Fresnel hologram synthesis, as an inverse problem, is more challenging

and demands a carefully tailored dataset and design of the CNN. So far, the potential suitability of CNNs for the hologram synthesis task has been demonstrated for only 2D images positioned at a fixed depth^{41,42} and for post compression⁴³.

Hologram dataset of tensor holography

To facilitate training CNNs for this task, we introduce a large-scale Fresnel hologram dataset, MIT-CGH-4K, consisting of 4,000 pairs of RGB-depth (RGB-D) images and corresponding 3D holograms. Our dataset is created with three important features to enable CNNs to learn photorealistic 3D holograms. First, the 3D scenes used for rendering the RGB-D images are constructed with high complexities and large variations in colour, geometry, shading, texture and occlusion to help the CNN generalize to both computer-generated and real-world captured RGB-D test inputs. This is achieved by a custom random scene generator (Fig. 1a), which assembles a scene by randomly sampling 200–250 triangle meshes with repetition from a pool of over 50 meshes and assigning each mesh a random texture from a pool of over 60,000 textures from publicly available texture synthesis datasets^{44,45} with augmentation (see Methods for more rendering details). Second, the pixel depth distribution of the resulting RGB-D images is statistically uniform across the entire view frustum. This is crucial for preventing the learned CNN from biasing towards any frequently occurring depths and producing poor results at those sparsely populated ones when a non-uniform pixel depth distribution occurs. To ensure this property, we derived a closed-form probability density function (PDF) for arranging triangle meshes along the depth axis (z axis):

$$f_\alpha(z) = \frac{\alpha}{C(z_{\text{far}} - z_{\text{near}})} \left(\frac{z_{\text{far}} - z}{z_{\text{far}} - z_{\text{near}}} \right)^{\frac{\alpha}{C}-1} \quad (z_{\text{near}} \leq z < z_{\text{far}}), \quad (1)$$

Article

where z_{near} and z_{far} are the distances from the camera to the near and far plane of the view frustum, C is the number of meshes in the scene and α is a scaling factor calibrated via experimentation. This PDF distributes meshes exponentially along the z axis (Fig. 1a, top) such that the pixel depth distribution in the resulting RGB-D images is statistically uniform (Fig. 1a, bottom; see Methods for derivation and comparison with existing RGB-D datasets). Here we set z_{near} and z_{far} to 0.15 m and 10 m, respectively, to accommodate a wide range of focal distances (approximately a 6.6-diopter range for the depth of field). Third, the holograms computed from the RGB-D images can precisely focus each pixel to the location defined by the depth image and properly handle occlusion. This is accomplished by our **occlusion-aware point-based method** (OA-PBM).

The OA-PBM augments the PBM with occlusion detection. Instead of processing each 3D point independently, the OA-PBM reconstructs a triangle surface mesh from the RGB-D image and performs ray casting from each vertex (point) to the hologram plane (Fig. 1b). Wavefronts carried by the rays intersecting the surface mesh are excluded from hologram computation to account for foreground occlusion. In practice, a point light source is often used to magnify the hologram for an extended field of view (Extended Data Fig. 3a); thus, the OA-PBM implements configurable illumination geometry to support ray casting towards spatially varying diffraction cones. Figure 2b visualizes a focal stack refocused from the OA-PBM-computed holograms, in which clean occlusion boundaries are formed and little to no background light leaks into the foreground (see Methods for a comparison with PBM results and OA-PBM implementation details).

Combining the random scene generator and the OA-PBM, we rendered our dataset at wavelengths of 450 nm, 520 nm and 638 nm to match the RGB lasers deployed in our experimental prototype. The MIT-CGH-4K dataset is also rendered for multiple spatial light modulator (SLM) resolutions (see Methods for details) and will be made publicly available.

Neural network of tensor holography

Our CNN model is a fully convolutional residual network. It receives a four-channel RGB-D image and predicts a colour hologram as a six-channel image (RGB amplitude and RGB phase), which can be used to drive three optically combined SLMs or one SLM in a time-multiplexed manner to achieve full-colour holography. The network has a skip connection that creates a direct feed of the input RGB-D image to the penultimate residual block and has no pooling layer for preserving high-frequency details (see Fig. 1c for a scheme of the network architecture; see Methods for performance analysis and comparisons with other architectures). Let W be the width of the maximum subhologram (Fresnel zone plate) produced by the farthest object points to the hologram. We note that the minimal receptive field aggregated from all convolution layers should match W to physically accurately predict the target hologram. Yet, W of the target hologram varies according to the relative position between the hologram plane and the 3D volume, and can often reach hundreds of pixels (see Methods for derivation), resulting in too many convolution layers and slowing down the inference speed. To address the issue, we apply a pre-processing step to compute an intermediate representation (midpoint hologram), which reduces the effective W and losslessly recovers the target hologram.

The midpoint hologram is an application of the wavefront recording plane³⁰. It propagates the target hologram to the centre of the view frustum to optimally minimize the distance to any scene point, thus reducing the effective W . The calculation follows the two steps shown in Extended Data Fig. 3. First, the diverging frustum V induced by the point light source is mathematically converted to an analogous collimated frustum V' using the thin-lens formula describing the magnification of the laser beam (see Methods for calculation details).

The change of representation simplifies the simulation of depth-of-field images perceived in V into free-space propagation of the target hologram to the remapped depth in V' . Let $H_{\text{target}} \in \mathbb{C}^{M \times N}$ be the target hologram (colour channel is omitted here), where \mathbb{C} denotes the set of complex numbers, and M and N are the number of pixels along the width and height of the hologram. Let d'_{near} and d'_{far} be the distances from the target hologram to the near and far clipping plane of V' . H_{target} is propagated for a distance of $d'_{\text{mid}} = (d'_{\text{near}} + d'_{\text{far}})/2$ to the centre of V' to form the midpoint hologram $H_{\text{mid}} \in \mathbb{C}^{M \times N}$. The angular spectrum method⁴⁷ (ASM) is employed to model the propagation of a wave field:

$$H_{\text{mid}}(m, n) = \text{ASM}(H_{\text{target}}, d'_{\text{mid}}) \\ = F^{-1}\left\{ F\{H_{\text{target}}\} e^{i2\pi d'_{\text{mid}} \sqrt{\lambda^{-2} - (m/L_w)^2 - (n/L_h)^2}} \right\}. \quad (2)$$

Here, F and F^{-1} are the Fourier and inverse Fourier transform operators, respectively; L_w and L_h are the physical width and height of the hologram, respectively; λ is the wavelength; $m = -M/2, \dots, M/2 - 1$ and $n = -N/2, \dots, N/2 - 1$. Replacing the target hologram with the midpoint hologram reduces W by a factor of $d'_{\text{far}}/\Delta d'$, where $\Delta d' = (d'_{\text{far}} - d'_{\text{near}})/2$. The reduction is a result of eliminating the free-space propagation shared by all the points, and the target hologram can be exactly recovered by propagating the midpoint hologram back for a distance $-d'_{\text{mid}}$. In our rendering configuration, where the collimated frustum V' has a 6-mm optical path length, using the midpoint hologram as the CNN's learning objective minimizes the convolution layers to 30.

We introduce two wave-based loss functions to train the CNN to accurately approximate the midpoint hologram and learn Fresnel diffraction. The first loss function serves as a data fidelity measure and computes the phase-corrected ℓ_2 distance between the predicted hologram $\tilde{H}_{\text{mid}} = \tilde{A}_{\text{mid}} e^{i\tilde{\phi}_{\text{mid}}} \in \mathbb{C}^{M \times N}$ and the ground-truth midpoint hologram $H_{\text{mid}} = A_{\text{mid}} e^{i\phi_{\text{mid}}}$:

$$l_{\text{data}} = \left\| \tilde{A}_{\text{mid}} - A_{\text{mid}} e^{i(\delta(\tilde{\phi}_{\text{mid}}, \phi_{\text{mid}}) - \hat{\delta}(\tilde{\phi}_{\text{mid}}, \phi_{\text{mid}}))} \right\|, \quad (3)$$

where \tilde{A}_{mid} and $\tilde{\phi}_{\text{mid}}$ are the amplitude and phase of the predicted hologram, A_{mid} and ϕ_{mid} are the amplitude and phase of the ground truth hologram, $\delta(\tilde{\phi}_{\text{mid}}, \phi_{\text{mid}}) = \text{atan}2[\sin(\tilde{\phi}_{\text{mid}} - \phi_{\text{mid}}), \cos(\tilde{\phi}_{\text{mid}} - \phi_{\text{mid}})]$, $\hat{\cdot}$ denotes the mean and $\|\cdot\|_p$ denotes the ℓ_p vector norm applied on a vectorized matrix output. The phase correction computes the signed shortest angular distance in the polar coordinates and subtracts the global phase offset, which exerts no impact on the intensity of the reconstructed 3D image.

The second loss function measures the perceptual quality of the reconstructed 3D scene observed by a viewer. As ASM-based wave propagation is a differentiable operation, the loss is modelled as a combination of the ℓ_1 distance and total variation of a dynamic focal stack, reconstructed at two sets of focal distances that vary per training iteration

$$l_{\text{pcp}}(t) = \frac{\text{Dynamic focal stack}}{\sum_{d'_t \in \{D_t^{\text{fix}} \cup D_t^{\text{float}}\}} \left\| e^{\beta(2\Delta d' - (d'_t - D'_t))} \right\|} \\ \frac{\text{Attention mask}}{\frac{\text{Image difference}}{(|\text{ASM}(\tilde{H}_{\text{mid}}, d'_t)|^2 - |\text{ASM}(H_{\text{mid}}, d'_t)|^2)} \\ + \frac{\nabla|\text{ASM}(\tilde{H}_{\text{mid}}, d'_t)|^2 - \nabla|\text{ASM}(H_{\text{mid}}, d'_t)|^2}{\text{Total variation difference}} \right\|}. \quad (4)$$

Here, $|\cdot|^2$ denotes element-wise squared absolute value; ∇ denotes the total variation operator; t is the training iteration; $D'_t \in \mathbb{R}^{M \times N}$ is the depth channel (remapped to V') of the input RGB-D image, where \mathbb{R} denotes the set of real numbers; β is a user-defined attention scale; D_t^{fix} and D_t^{float} are two sets of dynamic focal distances calculated

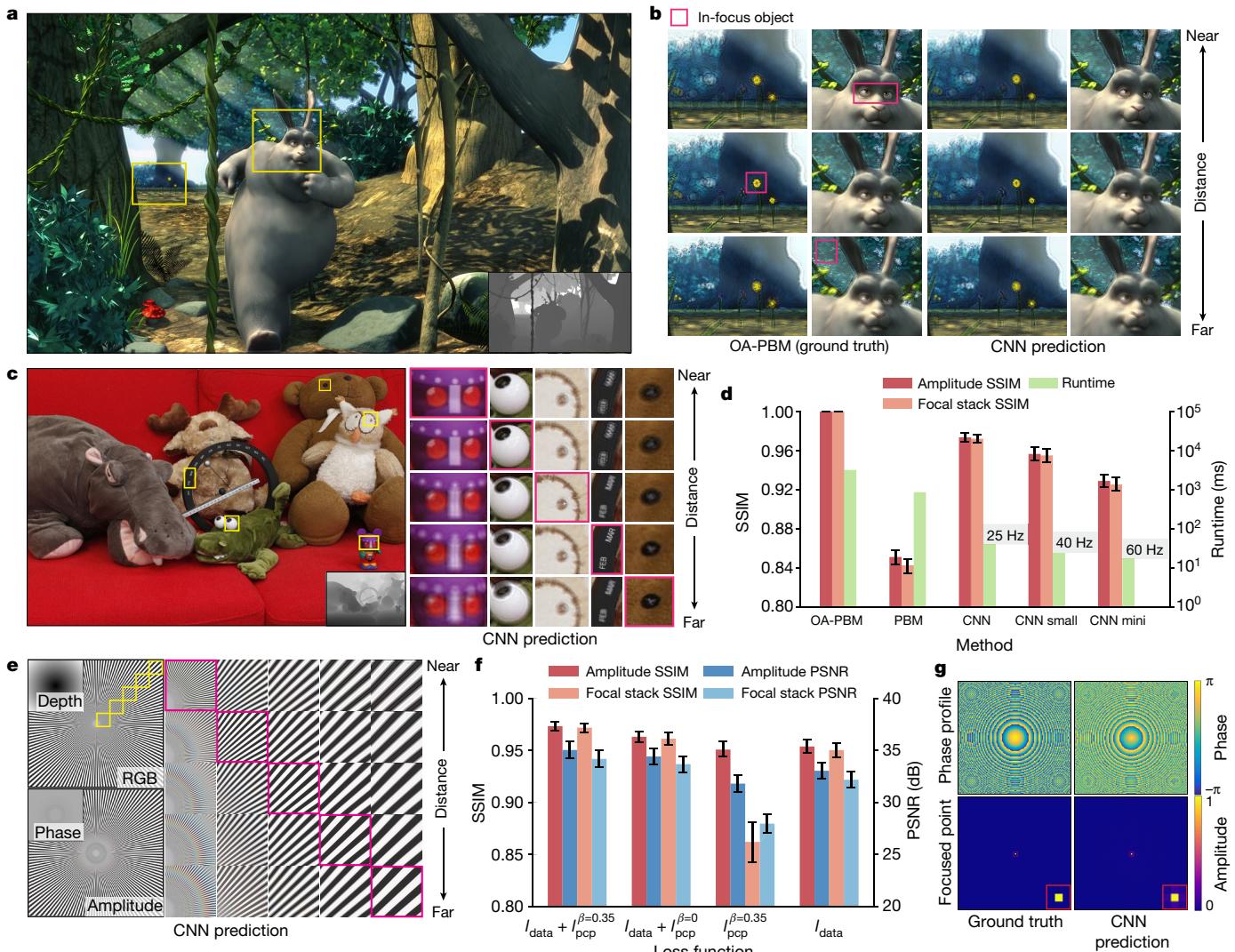


Fig. 2 | Performance evaluation of the OA-PBM and tensor holography CNN. **a**, A simulated depth-of-field image refocused from a CNN predicted hologram. The bunny's eye is in focus. The input RGB-D image is from *Big Buck Bunny*. The bottom right inset visualizes the depth image. **b**, Comparison of focal stacks reconstructed at highlighted regions in **a**. The CNN prediction is visually similar to the OA-PBM ground truth. **c**, A simulated depth-of-field image and focal stack (the magnified insets) reconstructed from the CNN predicted hologram of a real-world captured RGB-D image⁴⁶. **d**, Performance comparison of the PBM, OA-PBM and CNNs with various model capacities. The default CNN model consists of 30 convolution layers and 24 filters per layer, the small and mini models have 15 and 8 convolution layers, respectively. The reduction of convolution layers gracefully degrades the reconstructed image quality. The mini model runs in real time (60 Hz). The error bars are the standard deviation. SSIM, structural similarity index measure. **e**, A CNN predicted hologram and reconstructed depth-of-field images (the magnified insets) of a star test

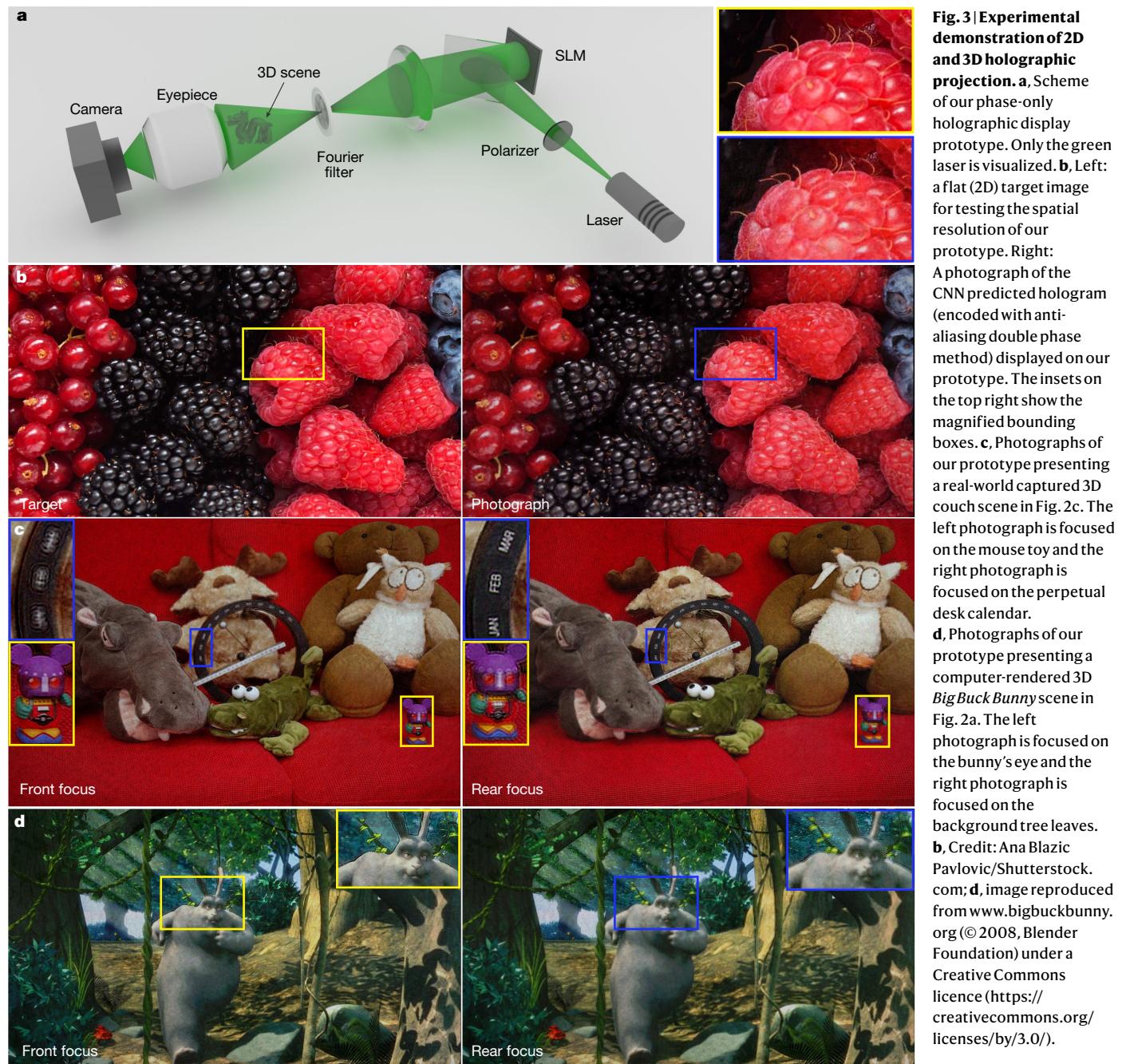
pattern. Line pairs of varying frequencies are sharply reconstructed at different depths, and the wavelength-dependent light dispersion is accurately reproduced. **f**, Ablation study of the full loss function (first). The ablation of attention mask (second) dilutes the CNN's attention to out-of-focus features and results in inferior performance. The ablation of data loss (third) removes the regularization of phase information and leads to poor generalization to unseen examples and large focal stack error. The ablation of perceptual loss (fourth) removes the guide of focal stacks and uniformly degrades the performance. The error bars are the standard deviation. PSNR, peak signal-to-noise ratio. **g**, Comparison of a ground truth Fresnel zone plate and a CNN prediction (by a model with 30 layers and 120 filters per layer) computed for a 6-mm distant point (propagated for another 20 mm for visualization).

b, c, Images reproduced from www.bigbuckbunny.org (© 2008, Blender Foundation) under a Creative Commons licence (<https://creativecommons.org/licenses/by/3.0/>).

as follows: (1) V is equally partitioned into T depth bins, (2) D_t^{fix} picks the top- k_{fix} bins from the histogram of D_t' and D_t^{float} randomly picks k_{float} bins among the rest, and (3) a depth is uniformly sampled from each selected bin. Here, D_t^{fix} guarantees the dominant content locations in the current RGB-D image are always optimized, while D_t^{float} ensures sparsely populated locations are randomly explored. The random sampling within each bin prevents overfitting to stationary depths, enabling the CNN to learn true 3D holograms. The attention mask directs the CNN to focus on reconstructing in-focus features in each depth-of-field image. Figure 2f validates the

effectiveness of each training loss component through an ablation study.

Our CNN was trained on a NVIDIA Tesla V100 GPU for 84 h (see Methods for model parameters and training details). The trained model generalizes well to computer-generated (Fig. 2a, Extended Data Fig. 5), real-world captured (Fig. 2c, Extended Data Fig. 6) RGB-D inputs, and standard test patterns (Fig. 2e, Extended Data Fig. 4). The simulated focal sweep of CNN-predicted 3D holograms can be found in Supplementary Videos 1, 2, 6. Compared with the reference OA-PBM holograms, the CNN predictions are both perceptually similar (Fig. 2b) and



numerically close (Fig. 2d, f). Evaluated on a single distance-point target, the output from a CNN with sufficient model capacity faithfully approximates a Fresnel zone plate (Fig. 2g), under the low-rank solution space restricted by a set of successively applied 3×3 convolution kernels. When all algorithms are implemented on a GPU with the CNN in NVIDIA TensorRT, and the OA-PBM and PBM in NVIDIA CUDA, the mini CNN achieves more than two orders of magnitude speed-up (Fig. 2d) over the OA-PBM and runs in real time (60 Hz) on a single NVIDIA Titan RTX GPU. As our end-to-end learning pipeline completely avoids logically complex ray–triangle intersection operations, it runs efficiently on low-power ASICs for accelerated CNN inference. In Supplementary Video 5, we demonstrate interactive mobile hologram computation on an iPhone 11 Pro, leveraging the A13 Bionic chip’s neural engine. Our model has an extremely low memory footprint of only 617 KB at Float32 precision and 315 KB at Float16 precision. At Int8 precision, it runs at 2 Hz on a single Google Edge TPU. All reported runtime performance is evaluated on inputs with a resolution of $1,920 \times 1,080$ pixels.

Display prototype of tensor holography

We have built a phase-only holographic display prototype (see Fig. 3a for a scheme and Extended Data Fig. 8 for a version of the physical setup) to experimentally validate our CNN. The prototype uses a HOLOEYE PLUTO-2-VIS-014 reflective SLM with a resolution of $1,920 \times 1,080$ pixels and a pixel pitch of $8 \mu\text{m}$ (see Methods for prototype details). The colour image is obtained field sequentially⁴⁸. To encode a CNN-predicted complex hologram into a phase-only hologram, we introduce an anti-aliasing double phase method (AA-DPM), which produces artefact-free 3D images around high-frequency objects and occlusion boundaries (see Methods for algorithm details and comparison with the original double phase method (DPM)^{49,50}). In Fig. 3b, we demonstrate speckle-free, high-resolution and high-contrast 2D projection, where the fluff of the berries can be found to be sharply reconstructed. In Fig. 3c, d, we show 3D holograms photographed for the couch scene and the *Big Buck Bunny* scene with focus set to the front and rear objects. Additional

photographs of real-world, computer-rendered and test scenes can be found in Extended Data Figs. 9, 10, where the image details closely match the simulation. Demonstration of real-time computation and focal sweep of 3D holograms can be found in Supplementary Videos 3, 4.

Discussion

Our results present evidence of using CNNs for real-time, photorealistic 3D CGH synthesis from a single RGB-D image, a task that was traditionally considered to be beyond the capabilities of existing computational devices. Our multi-resolution, large-scale Fresnel hologram dataset, created by the tailored random scene generator and the OA-PBM, will enable a wide range of conventional image-related applications to be transferred to holography: examples include super-resolution, compression, semantic editing of holograms and foveation-guided holographic rendering. Ultimately, it provides a test-bed for both commercial and academic research fields that will benefit from real-time, high-resolution CGH, for example, consumer holographic displays for virtual and augmented reality, hologram-based single-shot volumetric 3D printing, optical trapping with substantially increased foci and real-time simulation for holographic microscopy. Tensor holography itself can be further improved by directly learning phase-only holograms to discover an optimal encoding, avoiding explicit complex-to-phase-only conversion. In addition, though the RGB-D input is inexpensive to compute and memory efficient, it provides accurate 3D depiction from only a single perspective. Thus, extending our pipeline to support true volumetric 3D input (voxel grid, dense light fields and general point cloud) could expedite the synthesis of holograms that support view-dependent effects and observation under large baseline movement (see Methods for expanded discussion). Finally, the rapid development of ASICs will soon make high-frame-rate tensor holography viable on mobile devices, enabling untethered real 3D viewing experiences and substantially lowering the cost and barrier to entry for holographic content creation.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41586-020-03152-0>.

1. Benton, S. A., Bove, J. & Michael, V. *Holographic Imaging* (John Wiley & Sons, 2008).
2. Maimone, A., Georgiou, A. & Kollar, J. S. Holographic near-eye displays for virtual and augmented reality. *ACM Trans. Graph.* **36**, 85:1–85:16 (2017).
3. Shi, L., Huang, F.-C., Lopes, W., Matusik, W. & Luebke, D. Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3D computer graphics. *ACM Trans. Graph.* **36**, 236:1–236:17 (2017).
4. Tsang, P. W. M., Poon, T.-C. & Wu, Y. M. Review of fast methods for point-based computer-generated holography [Invited]. *Photon. Res.* **6**, 837–846 (2018).
5. Sitzmann, V. et al. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Trans. Graph.* **37**, 114:1–114:13 (2018).
6. Lee, G.-Y. et al. Metasurface eyepiece for augmented reality. *Nat. Commun.* **9**, 4562 (2018).
7. Hu, Y. et al. 3d-integrated metasurfaces for full-colour holography. *Light Sci. Appl.* **8**, 86 (2019).
8. Melde, K., Mark, A. G., Qiu, T. & Fischer, P. Holograms for acoustics. *Nature* **537**, 518–522 (2016).
9. Smalley, D. et al. A photophoretic-trap volumetric display. *Nature* **553**, 486–490 (2018).
10. Hirayama, R., Plasencia, D. M., Masuda, N. & Subramanian, S. A volumetric display for visual, tactile and audio presentation using acoustic trapping. *Nature* **575**, 320–323 (2019).
11. Rivenson, Y., Wu, Y. & Ozcan, A. Deep learning in holography and coherent imaging. *Light Sci. Appl.* **8**, 85 (2019).
12. Shusteff, M. et al. One-step volumetric additive manufacturing of complex polymer structures. *Sci. Adv.* **3**, eaao5496 (2017).
13. Kelly, B. E. et al. Volumetric additive manufacturing via tomographic reconstruction. *Science* **363**, 1075–1079 (2019).
14. Levoy, M. & Hanrahan, P. Light field rendering. In Proc. 23rd Annual Conference on Computer Graphics and Interactive Techniques 31–42 (ACM, 1996).
15. Waters, J. P. Holographic image synthesis utilizing theoretical methods. *Appl. Phys. Lett.* **9**, 405–407 (1966).
16. Leseberg, D. & Frère, C. Computer-generated holograms of 3-D objects composed of tilted planar segments. *Appl. Opt.* **27**, 3020–3024 (1988).
17. Tommasi, T. & Bianco, B. Computer-generated holograms of tilted planes by a spatial frequency approach. *J. Opt. Soc. Am. A* **10**, 299–305 (1993).
18. Matsushima, K. & Nakahara, S. Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method. *Appl. Opt.* **48**, H54–H63 (2009).
19. Symeonidou, A., Blinder, D., Munteanu, A. & Schelkens, P. Computer-generated holograms by multiple wavefront recording plane method with occlusion culling. *Opt. Express* **23**, 22149–22161 (2015).
20. Luente, M. E. Interactive computation of holograms using a look-up table. *J. Electron. Imaging* **2**, 28–35 (1993).
21. Luente, M. & Galyean, T. A. Rendering interactive holographic images. In Proc. 22nd Annual Conference on Computer Graphics and Interactive Techniques, 387–394 (ACM, 1995).
22. Luente, M. Interactive three-dimensional holographic displays: seeing the future in depth. *Comput. Graph.* **31**, 63–67 (1997).
23. Chen, J.-S. & Chu, D. P. Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications. *Opt. Express* **23**, 18143–18155 (2015).
24. Zhao, Y., Cao, L., Zhang, H., Kong, D. & Jin, G. Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method. *Opt. Express* **23**, 25440–25449 (2015).
25. Makey, G. et al. Breaking crosstalk limits to dynamic holography using orthogonality of high-dimensional random vectors. *Nat. Photon.* **13**, 251–256 (2019).
26. Yamaguchi, M., Hoshino, H., Honda, T. & Ohyama, N. In *Practical Holography VII: Imaging and Materials* Vol. 1914 (ed. Benton, S. A.) 25–31 (SPIE, 1993).
27. Barabas, J., Jolly, S., Smalley, D. E. & Bove, V. M. Jr In *Practical Holography XXV: Materials and Applications* Vol. 7957 (ed. Bjelkhagen, H. I.) 13–19 (SPIE, 2011).
28. Zhang, H., Zhao, Y., Cao, L. & Jin, G. Fully computed holographic stereogram based algorithm for computer-generated holograms with accurate depth cues. *Opt. Express* **23**, 3901–3913 (2015).
29. Padmanaban, N., Peng, Y. & Wetzstein, G. Holographic near-eye displays based on overlap-add stereograms. *ACM Trans. Graph.* **38**, 214:1–214:13 (2019).
30. Shimobaba, T., Masuda, N. & Ito, T. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane. *Opt. Lett.* **34**, 3133–3135 (2009).
31. Wakunami, K. & Yamaguchi, M. Calculation for computer generated hologram using ray-sampling plane. *Opt. Express* **19**, 9086–9101 (2011).
32. Häussler, R. et al. Large real-time holographic 3D displays: enabling components and results. *Appl. Opt.* **56**, F45–F52 (2017).
33. Hamann, S., Shi, L., Solgaard, O. & Wetzstein, G. Time-multiplexed light field synthesis via factored Wigner distribution function. *Opt. Lett.* **43**, 599–602 (2018).
34. Nair, V. & Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In Proc. International Conference on International Conference on Machine Learning (ICML) 807–814 (Omnipress, 2010).
35. Sinha, A., Lee, J., Li, S. & Barbastathis, G. Lensless computational imaging through deep learning. *Optica* **4**, 1117–1125 (2017).
36. Metzler, C. et al. prdeep: robust phase retrieval with a flexible deep network. In Proc. International Conference on International Conference on Machine Learning (ICML) 3501–3510 (JMLR, 2018).
37. Ebypbos, M. H., Caira, N. W., Chakravarthula, P., Atisa, M. & Pégard, N. C. In *Optics and the Brain* BTu2C-2 (Optical Society of America, 2020).
38. Rivenson, Y., Zhang, Y., Günaydin, H., Teng, D. & Ozcan, A. Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light Sci. Appl.* **7**, 17141 (2018).
39. Ren, Z., Xu, Z. & Lam, E. Y. Learning-based nonparametric autofocusing for digital holography. *Optica* **5**, 337–344 (2018).
40. Wu, Y. et al. Extended depth-of-field in holographic imaging using deep-learning-based autofocusing and phase recovery. *Optica* **5**, 704–710 (2018).
41. Horisaki, R., Takagi, R. & Tanida, J. Deep-learning-generated holography. *Appl. Opt.* **57**, 3859–3863 (2018).
42. Peng, Y., Choi, S., Padmanaban, N. & Wetzstein, G. Neural holography with camera-in-the-loop training. *ACM Trans. Graph.* **39**, 185:1–185:14 (2020).
43. Jiao, S. et al. Compression of phase-only holograms with JPEG standard and deep learning. *Appl. Sci.* **8**, 1258 (2018).
44. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S. & Vedaldi, A. Describing textures in the wild. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 3606–3613 (IEEE, 2014).
45. Dai, D., Riemenschneider, H. & Gool, L. V. The synthesizability of texture examples. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 3027–3034 (IEEE, 2014).
46. Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A. & Gross, M. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.* **32**, 73:1–73:12 (2013).
47. Matsushima, K. & Shimobaba, T. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Opt. Express* **17**, 19662–19673 (2009).
48. Shimobaba, T. & Ito, T. A color holographic reconstruction system by time division multiplexing with reference lights of laser. *Opt. Rev.* **10**, 339–341 (2003).
49. Hsueh, C. K. & Sawchuk, A. A. Computer-generated double-phase holograms. *Appl. Opt.* **17**, 3874–3883 (1978).
50. Mendoza-Yero, O., Minguez-Vega, G. & Lancis, J. Encoding complex fields by using a phase-only optical element. *Opt. Lett.* **39**, 1740–1743 (2014).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2021

Article

Methods

OA-PBM

The OA-PBM assumes a general holographic display setting, where the RGB-D image is rendered with perspective projection and the hologram is illuminated by a point source of light co-located with the camera. This includes the support of collimated illumination, a special case where the point light source is located at infinity and the rendering projection is orthographic. During ray casting, every object point defined by the RGB-D image produces a subhologram at the hologram plane. The maximum spatial extent of a subhologram is dictated by the grating equation

$$\Delta p(\sin\theta_m - \sin\theta_i) = \pm\lambda, \quad (5)$$

where Δp is the SLM pixel pitch, θ_i is the light incidence angle from the point light source to a hologram pixel, θ_m is the maximum outgoing angle from the same hologram pixel and λ is the wavelength. Let $\mathbf{o} \in \mathbb{R}^3$ be (the location of) an object point defined by the RGB-D image, S_o be the set of SLM pixels within the extent of the subhologram of \mathbf{o} , $\mathbf{p} \in \mathbb{R}^3$ be (the location of) an SLM pixel in S_o , $\mathbf{l} \in \mathbb{R}^3$ be (the location of) the point light source and S_{slm} be the set of all SLM pixels, the wavefront contributed from \mathbf{o} to \mathbf{p} under the illumination of \mathbf{l} is given by

$$h_o(\mathbf{p}) = \frac{a}{w_o} e^{i\left(\phi_o + \frac{2\pi(\|\mathbf{p}-\mathbf{o}\|_2 + \|\mathbf{p}-\mathbf{l}\|_2)}{\lambda}\right)}, \quad (6)$$

where a is the amplitude associated with \mathbf{o} , $w_o = \sqrt{a^2 / \sum_{j \in S_{\text{slm}}} [\mathbf{j} \in S_{\text{slm}}]}$ is an amplitude attenuation factor for energy conversation (where \mathbf{j} is a dummy variable that denotes an SLM pixel in S_o and $[\cdot]$ denotes Iverson bracket) and ϕ_o is the initial phase associated with \mathbf{o} . The initialization of ϕ_o uses the position-dependent formula by Maimone et al.² instead of random initialization to allow different Fresnel zone kernels to cancel out at the hologram plane and achieve a smooth phase profile. We emphasize that this deterministic phase initialization method is critical to the success of CNN training, as it ensures the complex holograms generated for the entire dataset are statistically consistent and bear repetitive features that can be learned by a CNN.

The OA-PBM models occlusion by multiplying $h_o(\mathbf{p})$ with a binary visibility mask $v_o(\mathbf{p})$. The value of $v_o(\mathbf{p})$ is set to 0 if ray \mathbf{op} intersects the piece-wise linear surface (triangular surface mesh) built from the RGB-D image. In practice, this ray-triangle intersection test can be accelerated with space tracing by only testing the set of triangles Q_{op} that may lie on the path of \mathbf{op} . Let \mathbf{p}_{oi} be the SLM pixel intersecting \mathbf{ol} (pixel at the subhologram centre of \mathbf{o}), the set Q_{op} only consists of triangles whose vertices' x-y coordinate indices are on the path of line segment \mathbf{pp}_{oi} , and

$$v_o(\mathbf{p}) = \neg \left[\bigvee_{q \in Q_{\text{op}}} \mathbf{op} \text{ intersects } q \right], \quad (7)$$

where q is a dummy variable that denotes a triangle on the path of Q_{op} . Finally, the target hologram H_{target} is obtained by summing subholograms contributed from all object points

$$H_{\text{target}}(\mathbf{p}) = \sum_{j \in S_p} v_j(\mathbf{p}) h_j(\mathbf{p}), \quad (8)$$

where S_p is the set of object points whose subholograms are defined at \mathbf{p} . Extended Data Fig. 1b visualizes the masked Fresnel zone plate computed for different depth landscapes. Compared with the PBM, the OA-PBM considerably reduces background leakage (Extended Data Fig. 1d). It is important to note that the OA-PBM is still a first-order approximation of the Fresnel diffraction, and the hologram quality

could be further improved by modelling wavefronts from secondary point sources stimulated at the occlusion boundaries based on the Huygens-Fresnel principle. While theoretically possible, in practice the number of triggered rays grows exponentially with respect to the number of occlusions, and both the computation and memory cost becomes intractable for complex scenes and provides only minor improvement (see Extended Data Fig. 1c for a comparison study of an elementary case).

Random scene generator

The random scene generator is implemented using the NVIDIA OptiX-ray-tracing library with the NVIDIA AI-Accelerated denoiser turned on to maximize customizability and performance. During the construction of a scene, we limit the random scaling of mesh such that the longest side of the mesh's bounding box falls within 0.1 times to 0.35 times the screen space height. This prevents a single mesh from being negligibly small or overwhelmingly large. We also distribute meshes according to equation (1) to produce a statistically uniform pixel depth distribution in the rendered depth image. To show the derivation of the probability density function $f(z)$, we start from an elementary case where only a single pixel is to be rendered. Let a series of mutually independent and identically distributed random variables z_1, z_2, \dots, z_C denote the depths of all C' meshes in the camera's line of sight. The measured depth of this pixel z_d is dictated by the closest mesh to the camera, namely $z_d = \min\{z_1, z_2, \dots, z_C\}$. For any $z \in [z_{\text{near}}, z_{\text{far}}]$

$$z_d \geq z \iff \min\{z_1, z_2, \dots, z_C\} \geq z \iff \bigwedge_{i=1}^{C'} z_i \geq z, \quad (9)$$

where i is a dummy variable that iterates from 1 to C' . From a probabilistic perspective

$$\Pr(z_d \geq z) = \Pr\left(\bigwedge_{i=1}^{C'} z_i \geq z\right) = \prod_{i=1}^{C'} \Pr(z_i \geq z) = [\Pr(z_1 \geq z)]^{C'}. \quad (10)$$

When z_d obeys a uniform distribution over $[z_{\text{near}}, z_{\text{far}}]$, $\Pr(z_d \geq z) = (z_{\text{far}} - z)/(z_{\text{far}} - z_{\text{near}})$. Meanwhile, $\Pr(z_1 \geq z) = \int_z^{z_{\text{far}}} f(t) dt$. Thus, equation (10) can be rewritten into the following form for every $z \in [z_{\text{near}}, z_{\text{far}}]$

$$\int_z^{z_{\text{far}}} f(t) dt = [\Pr(z_d \geq z)]^{\frac{1}{C'}} = \left(\frac{z_{\text{far}} - z}{z_{\text{far}} - z_{\text{near}}} \right)^{\frac{1}{C'}-1}. \quad (11)$$

Differentiating both the leftmost and the rightmost side with respect to z

$$f(z) = \frac{1}{C'(z_{\text{far}} - z_{\text{near}})} \left(\frac{z_{\text{far}} - z}{z_{\text{far}} - z_{\text{near}}} \right)^{\frac{1}{C'}-1} \quad (12)$$

gives a closed-form solution to the PDF associated with z_1, z_2, \dots, z_C .

Although it is required by definition that $C' \in \mathbb{Z}^+$, where \mathbb{Z}^+ denotes the set of positive integers, equation (12) extrapolates to any positive real number no less than 1 for C' . In practice, calculating an average C' for the entire frame is non-trivial, as meshes of varying shapes and sizes are placed at random x-y positions and scaled stochastically. Nevertheless, C' is typically much smaller than the total number of meshes C , and well modelled by using a scaling factor α such that $C' = C/\alpha$. Equation (1) is thus obtained by applying this equation to equation (12). On the basis of experimentation, we find setting $\alpha = 50$ results in a sufficiently statistically uniform pixel depth distribution for $200 \leq C \leq 250$. Extended Data Fig. 2 shows a comparison of the resulting RGB-D images and histograms of pixel depth between our dataset and the DeepFocus dataset. The depth distribution of the DeepFocus dataset is unevenly biased to the front and rear end of the view frustum. This is due to both unoptimized object depth

distribution and sparse scene coverage that leads to overly exposed backgrounds.

We generated 4,000 random scenes using the random scene generator. To support application of important image processing and rendering algorithms such as super-resolution and foveation-guided rendering to holography, we rendered holograms for both 8 μm and 16 μm pixel pitch SLMs. The image resolution was chosen to be 384 \times 384 pixels and 192 \times 192 pixels, respectively, to match the physical size of the resultant holograms and enable training on commonly available GPUs. We note that as the CNN is fully convolutional, as long as the pixel pitch remains the same, the trained model can be used to infer RGB-D inputs of an arbitrary spatial resolution at test time.

Finally, we acknowledge that an RGB-D image records only the 3D scene perceived from the observer's current viewpoint, and it is not a complete description of the 3D scene with both occluded and non-occluded objects. Therefore, it is not an ideal input for creating holograms that are intended to remain static, but being viewed by an untracked viewer for motion parallax under large baseline movement or simultaneously by multiple persons. However, with real-time performance first enabled by our CNN on RGB-D input, this limitation is not a concern for interactive applications and particularly with eye position tracked, as new holograms can be computed on-demand on the basis of the updated scene, viewpoint or user input to provide an experience as though the volumetric 3D scene was simultaneously reconstructed. This is especially true for virtual and augmented-reality headsets, where six-degrees-of-freedom positional tracking has become omnipresent, and we can always deliver the correct viewpoint of a complex 3D scene for a moving user by updating the holograms to reflect the change of view.

However, the low rendering cost and memory overhead of RGB-D representation is a key attribute that enables practical real-time applications. Volumetric 3D representations (dense point cloud, voxel grid, light fields) at the same spatial resolution generally consume orders of magnitude more data. The increased rendering, memory, input/output and data streaming cost alone have made them much less practical for real-time applications with current graphics hardware (that is, a 1080P light field video with only 8 \times 8 views is already four times the data of an 8-K video), not including proportionally increased hologram computation cost, which dominates the total cost. The additional points (objects) offered by these representations, however, are either occluded or out of the frame of the current viewpoint. Consequently, they contribute little to no wavefront to the perceived 3D image of the current view. Beyond computer graphics, the RGB-D image is readily available with low-cost RGB-D sensors such as Microsoft Kinect or integrated sensors of modern mobile phones. This further facilitates utilization of real-world captured data, whereas high-resolution full 3D scanning of real-world-sized environments is much less accessible and requires specialized high-cost imaging devices. Thus, the RGB-D representation strikes a balance between image quality and practicality for interactive applications.

CNN model architecture, training, evaluation and comparisons

Our network architecture consists of only residual blocks and a skip connection from the input to the penultimate residual block. The architecture is similar to DeepFocus⁵¹, a fully convolutional neural network designed for synthesizing image content for varifocal, multifocal and light field head-mounted displays. Yet, our architecture ablates its volume-preserving interleaving and de-interleaving layer. The interleaving layer reduces the spatial dimension of an input tensor through rearranging non-overlapped spatial blocks into the depth channel, and the de-interleaving layer reverts the operation. A high interleaving rate reduces the network capacity and trades lower image quality for faster runtime. In practice, we compared three different network miniaturization methods in Extended Data Fig. 4b: (1) reduce the number of convolution layers; (2) use a high interleaving rate; and (3) reduce

the number of filters per convolution layer. At equal runtime, approach 1 (using fewer convolution layers) produces the highest image quality for our task; approach 3 results in the lowest image quality because the CNN model contains the lowest number of filters (240 filters for approach 3 compared with 360 or 1,440 filters for approaches 1 and 2, respectively), while approach 2 is inferior to approach 1 mainly because neighbouring pixels are scattered across channels, making a reasoning of their interactions much more difficult. This is particularly harmful when the CNN has to learn how different Fresnel zone kernels should cancel out to produce a smooth phase distribution. Given this observation, we ablate the interleaving and de-interleaving layers in favour of both performance and model simplicity.

All convolution layers in our network use 3×3 convolution filters. The number of minimally required convolution layers depends on the maximal spatial extent of the subhologram. Quantitatively, successive application of x convolution layers results an effective $3 + (x - 1) \times 2$ convolution. Solving for the maximum subhologram width $W = 3 + (x - 1) \times 2$ yields $[(W - 3)]/2 + 1$ minimally required convolution layers. In Extended Data Fig. 3, we demonstrate the calculation of the midpoint hologram, which reduces the effective maximum subhologram size through relocating the hologram plane. First, the holographic display magnified by the point light source is unmagnified to its collimated illumination counterpart. The original view frustum V and the unmagnified view frustum V' are related by the thin-lens equation $1/d' = 1/d + 1/f$, where f , d and d' are the distance between the point light source and the hologram, the hologram and a point in V , and the hologram and the same point mapped to V' respectively. Then, the target hologram is propagated to the centre of the unmagnified view frustum V'' following equation (2). As the resulting midpoint hologram depends on only the thickness of the 3D volume, it leads to a substantial reduction of W if the relative distance between the hologram plane and the 3D volume is far. For example, in our rendering setting, we assume a 30-mm eyepiece magnifies a collimated frustum between 24 mm and 30 mm away, effectively resulting in a magnified frustum that covers from 0.15 m to infinity for an observer that is one focal length behind the eyepiece. If the hologram plane is co-located with the eyepiece (30 mm to the far clipping plane), using the midpoint to substitute the target hologram reduces the maximum subhologram width by ten times from 600 pixels to 60 pixels, resulting in 30 convolution layers as minimally required. In practice, we find using fewer convolution layers than the theoretical minimum only moderately degrades the image quality (Fig. 2d). This is because the use of the phase initialization of Maimone et al.² allows the target phase pattern to be mostly occupied by low-frequency features and absent from Fresnel-zone-plate-like high-frequency patterns. Thus, even with reduced effective convolution kernel size, such features are still sufficiently easy to reproduce.

We reiterate that the midpoint hologram is an application of the wavefront recording plane (WRP)³⁰ as a pre-processing step. In physical-based methods, the WRP is introduced as an intermediate ray-sampling plane placed either inside⁵² or outside^{30,53} the point cloud to reduce the wave propagation distance and thus the subhologram size during Fresnel diffraction integration. Application of multiple WRPs was also combined with the use of precomputed propagation kernels to further accelerate the runtime at the price of sacrificing accurate per-pixel focal control^{19,54}. For fairness, the GPU runtimes reported for the OA-PBM and PBM baseline in Fig. 2d have been accelerated by putting the WRP to a plane that corresponds to the centre of the collimated frustum.

Our CNN is trained on a 384 \times 384-pixel RGB-D image and hologram pairs. We use a batch size of 2, ReLu activation, attention scale $\beta = 0.35$, number of depth bins $T = 200$, number of dynamic focal stack $k_{\text{fix}} = 15$ and $k_{\text{float}} = 5$ for the training. We train the CNN for 1,000 epochs using the Adam⁵⁵ optimizer at a constant learning rate of 1×10^{-4} . The dataset is partitioned into 3,800, 100 and 100 samples for training, testing and validation. Extended Data Fig. 4a quantitatively compares the

Article

performance of our CNN with U-Net⁵⁶ and Dilated-Net⁵⁷, both of which are popular CNN architectures for image synthesis tasks. When the capacity of the other two models is configured for the same inference time, our network achieves the highest performance. The superiority comes from the more consistent and repetitive architecture of our CNN. Specifically, it avoids the use of pooling and transposed convolution layers to contract and expand the spatial dimension of intermediate tensors, thus the high-frequency features of Fresnel zone kernels are more easily constructed and preserved during forward propagation.

In Extended Data Fig. 4c, we evaluate our CNN on two additional standard pattern (USAF-1951 and RCA Indian-head) variants made by the authors. The CNN-predicted holograms can reproduce a few-pixel-wide patterns as shown by the magnified in-focus insets. In Extended Data Figs. 5, 6, we show four additional complex scenes (two computer rendered and two real-world captured) and the CNN predicted holograms.

AA-DPM

The double phase method encodes an amplitude-normalized complex hologram $Ae^{i\phi} \in \mathbb{C}^{M \times N}$ ($0 \leq A \leq 1$) into a sum of two phase-only holograms at half of the normalized maximum amplitude:

$$Ae^{i\phi} = 0.5e^{i(\phi - \cos^{-1}A)} + 0.5e^{i(\phi + \cos^{-1}A)}. \quad (13)$$

There are many different methods to merge decomposed two phase-only holograms into a single phase-only hologram. The original DPM⁵⁰ uses a checkerboard mask to select interleaving phase values from the two phase-only holograms. Maimone et al.² first discard every other pixel of the input complex hologram along one spatial axis and then arrange the decomposed two phase values along the same axis in a checkerboard pattern. The latter method produces visually comparable results, but reduces the complexity of the hologram calculation by half via avoiding calculation at unused locations. Nevertheless, for complex 3D scenes, they produce severe artefacts around high-frequency objects and occlusion boundaries (Extended Data Fig. 7, left). This is because the high-frequency phase alterations presented at these regions become under-sampled due to the interleaving sampling pattern and disposal of every other pixel. Although these artefacts can be partially suppressed by closing the aperture and cutting the high-frequency signal in the Fourier domain, this leads to substantial blurring. Although sampling is inevitable, we borrow techniques employed in traditional image subsampling to holographic content and introduce an AA-DPM. Specifically, we first convolve the complex hologram by a Gaussian kernel $G_{W_G}(\sigma)$ to obtain a low-pass-filtered complex hologram $\bar{A}e^{i\bar{\phi}} \in \mathbb{C}^{M \times N}$:

$$\bar{A}e^{i\bar{\phi}} = Ae^{i\phi} * G_{W_G}(\sigma), \quad (14)$$

where $*$ denotes a 2D convolution operator, W_G is the width of the 2D Gaussian kernel and σ is the standard deviation of the Gaussian distribution. In practice, we find setting W_G no greater than 5 and σ between 0.5 and 1.5 is generally sufficient for both the rendered and captured 3D scenes used in this paper, while the exact σ can be fine-tuned based on the image statistics of content. For flat 2D images, σ can be further tuned down to achieve sharper results. The slightly blurred $\bar{A}e^{i\bar{\phi}}$ avoids aliasing during sampling and allows the Fourier filter (aperture) to be opened wide, thus resulting in a sharp and artefact-free 3D image. We also add a global phase offset to $\bar{A}e^{i\bar{\phi}}$ to centre the mean phase around half of the full phase-shift range of the SLM (3π in our case). This avoids phase warping and results in smooth phase distribution². Finally, let $P_1 \in \mathbb{C}^{M \times N}$ and $P_2 \in \mathbb{C}^{M \times N}$ be the two phase-only holograms decomposed from $\bar{A}e^{i\bar{\phi}}$ using equation (13), the final phase-only hologram $P \in \mathbb{C}^{M \times N}$ is calculated by arranging P_1 and P_2 in a checkerboard pattern

$$P(m, n) = \begin{cases} P_1(m, n) & \text{if } m + n \text{ is odd} \\ P_2(m, n) & \text{if } m + n \text{ is even} \end{cases} \quad (0 \leq m \leq M - 1, 0 \leq n \leq N - 1). \quad (15)$$

This alternating sampling pattern yields a high-frequency, phase-only hologram, which can diffract light as effectively as a random hologram, but without producing speckle noise. Extended Data Fig. 7 compares the depth-of-field images simulated for the AA-DPM and DPM, where the AA-DPM produces artefact-free images in regions with high-spatial-frequency details and around occlusion boundaries. The AA-DPM can be efficiently implemented on a GPU as two gather operations, which takes less than 1 ms to convert a $1,920 \times 1,080$ -pixel complex hologram on a single NVIDIA TITAN RTX GPU.

Holographic display prototype

Our display prototype (Extended Data Fig. 8) uses a Fisba RGBeam fibre-coupled laser and a single HOLOEYE PLUTO-2-VIS-014 liquid-crystal-on-silicon reflective phase-only SLM with a resolution of $1,920 \times 1,080$ pixels and a pitch of $8 \mu\text{m}$. The laser consists of three precisely aligned diodes operating at 450 nm, 520 nm and 638 nm, and provides per-diode power control. The prototype is constructed and aligned using a Thorlabs 30-mm and 60-mm cage system and components. The fibre-coupled laser is mounted using a ferrule connector/physical contact adaptor, placed at a distance that results in an ideal diverging beam (adjustable based on the desired field of view) and linearly polarized to the x axis (horizontal) to match the incident polarization required by the SLM. A plate beam splitter mounted on a 30-mm cage cube platform splits the beam and directs it towards the SLM. After SLM modulation, the reconstructed aerial 3D image is imaged by an achromatic doublet with a 60-mm focal length. An aperture stop is placed about one focal length behind the doublet (the Fourier plane) to block higher-order diffractions. The radius of its opening is set to match the extent of the blue beam's first-order diffraction. We emphasize that this should be the maximum radius as opening it further includes second-order diffraction from the blue beam. A 30-mm to 60-mm cage plate adaptor is then used to widen the optical path and an eyepiece is mounted to create the final retinal image.

In this work, a Sony A7 Mark III mirrorless camera with a resolution of $6,000 \times 4,000$ pixels and a Sony 16–35 mm f/2.8 GM lens is paired to photograph and record video of the display (except Supplementary Video 4). Colour reconstruction is obtained field sequentially with a maximum frame rate of 20 Hz that is limited by the SLM's 60-Hz refresh rate. A Labjack U3 USB DAQ is deployed to send field sequential signals and synchronize the display of colour-matched phase-only holograms. Each hologram is quantized to 8 bits to match the bit depth of the SLM. For the results shown in Fig. 3b, Extended Data Figs. 9, 10a, we used a Meade Series 5000 21-mm MWA eyepiece. For the results shown in Fig. 3c, d, Supplementary Videos 3, 4, Extended Data Fig. 10b, we used an Explore Scientific 32-mm eyepiece. The photograph was captured by exposing each colour channel for 1 s. The long exposure time improves the signal-to-noise ratio and colour accuracy. Supplementary Video 3 was captured at 4 K/30 Hz and downsampled to 1080P. Supplementary Video 4 was captured by a Panasonic GH5 mirrorless camera with a Lumix 10–25 mm f/1.7 lens at 4 K/60 Hz (a colour frame rate of 20 Hz) and downsampled to 1080P. No post sharpening, denoising or despeckling was applied to the captured videos and photographs. Finally, our setup can be further miniaturized to an eyeglass form factor as demonstrated by Maimone et al.².

Data availability

Our hologram dataset (MIT-CGH-4K) and the trained CNN model will be made publicly available (on GitHub) along with the paper.

Code availability

The code to evaluate the trained CNN model will be made publicly available (on GitHub) along with the paper. Additional codes are available from the corresponding authors upon reasonable request.

51. Xiao, L., Kaplanyan, A., Fix, A., Chapman, M. & Lanman, D. DeepFocus: learned image synthesis for computational displays. *ACM Trans. Graph.* **37**, 200:1–200:13 (2018).
52. Wang, Y., Sang, X., Chen, Z., Li, H. & Zhao, L. Real-time photorealistic computer-generated holograms based on backward ray tracing and wavefront recording planes. *Opt. Commun.* **429**, 12–17 (2018).
53. Hasegawa, N., Shimobaba, T., Kakue, T. & Ito, T. Acceleration of hologram generation by optimizing the arrangement of wavefront recording planes. *Appl. Opt.* **56**, A97–A103 (2017).
54. Sifatul Islam, M. et al. Max-depth-range technique for faster full-color hologram generation. *Appl. Opt.* **59**, 3156–3164 (2020).
55. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015).
56. Ronneberger, O., Fischer, P. & Brox, T. U-net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* 234–241 (Springer, 2015).
57. Yu, F., Koltun, V. & Funkhouser, T. Dilated residual networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 472–480 (IEEE, 2017).

Acknowledgements We thank K. Aoyama and S. Wen (from Sony) for discussions; J. Minor, T. Du, M. Foshey, L. Makatura, W. Shou and T. Erps from MIT for improving/editing the manuscript; R. White for the administration of the project; X. Ju for the design of iPhone demo;

and P. Ma for providing an iPhone 11 Pro for the mobile demo. We acknowledge funding from Sony Research Award Program.

Author contributions L.S. conceived the idea, implemented the proposed framework, built the display prototype, performed experimental validation, and conducted the iPhone and Edge TPU demo. B.L. performed the pipeline evaluation and made the Supplementary Videos. B.L., C.K. and P.K. were involved in the design of the proposed framework. L.S. and P.K. led the writing and revision of the manuscript. W.M. supervised the work. All authors discussed ideas and results, and contributed to the manuscript.

Competing interests The authors declare no competing interests.

Additional information

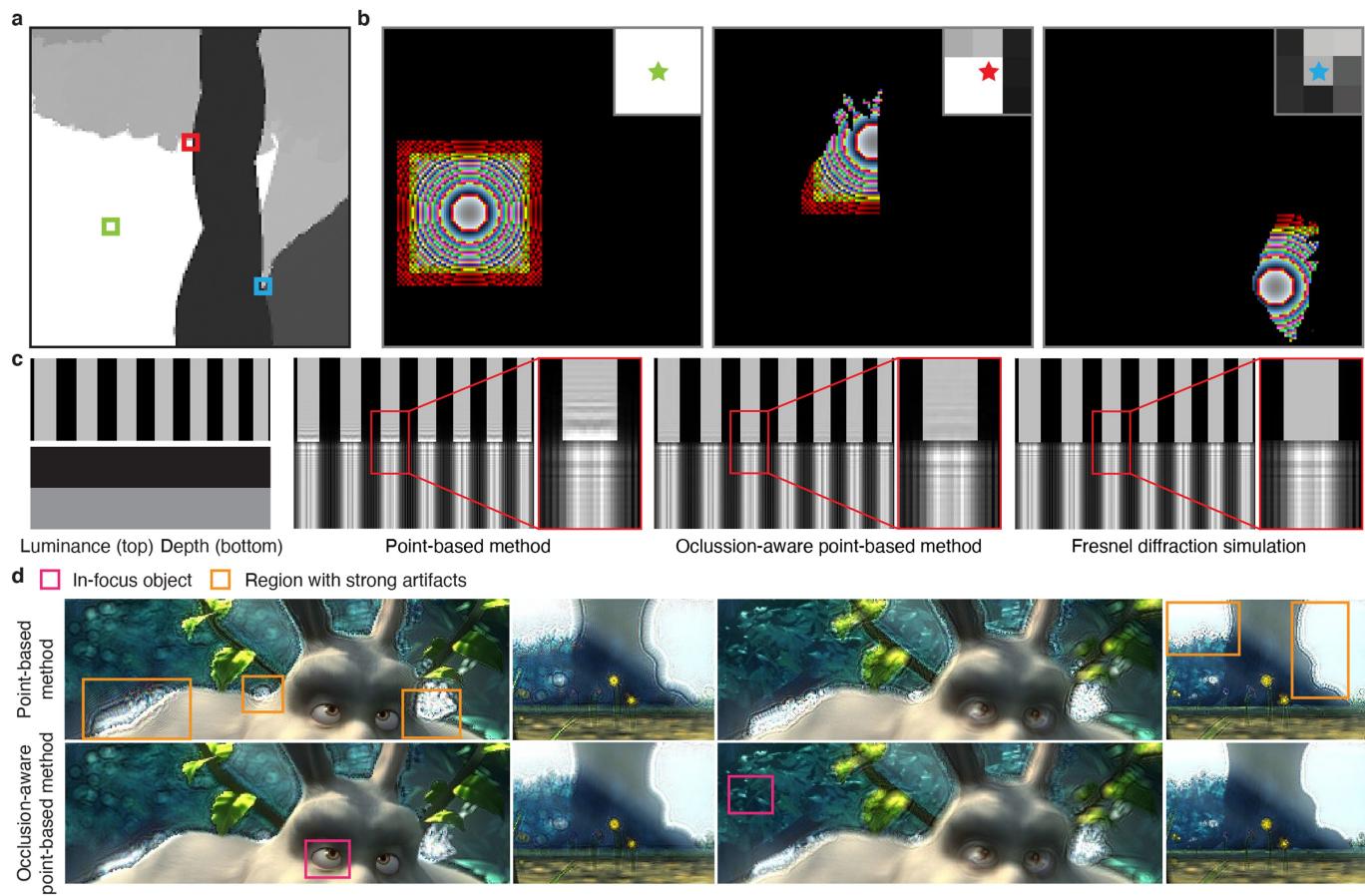
Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41586-020-03152-0>.

Correspondence and requests for materials should be addressed to L.S. or W.M.

Peer review information *Nature* thanks Tomoyoshi Shimobaba and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

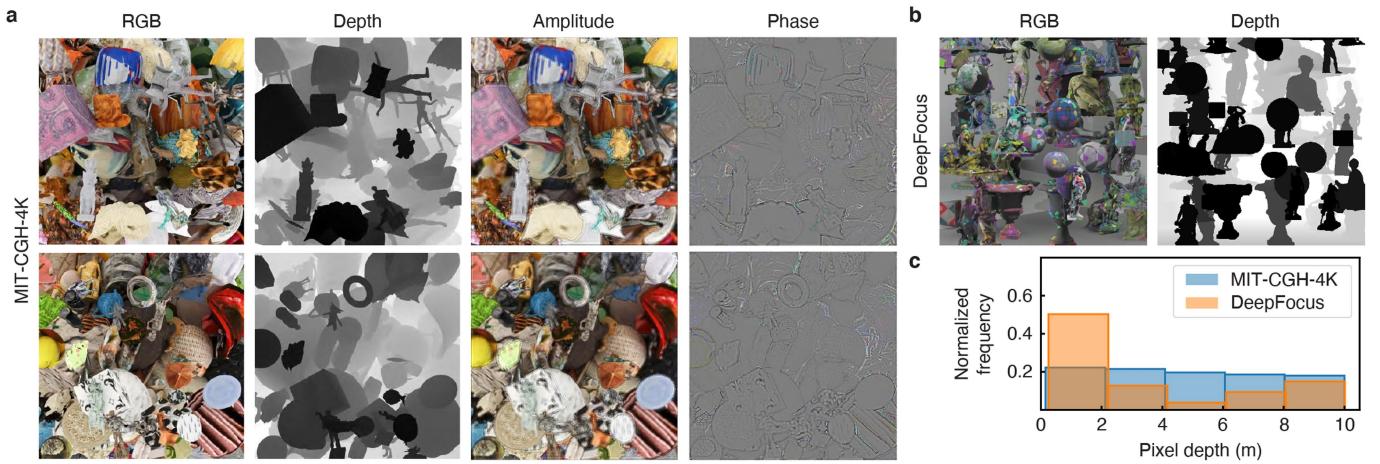
Reprints and permissions information is available at <http://www.nature.com/reprints>.

Article



Extended Data Fig. 1 | Visualization of masked Fresnel zone plates computed by OA-PBM and performance comparison of foreground occlusion. **a**, A depth image cropped from a frame of *Big Buck Bunny*. Three regions with different depth landscapes are highlighted in different colours. **b**, Masked Fresnel zone plates computed for the centre pixel of each highlighted region. Three pixels are propagated for the same distance for ease of comparison. The flat depth landscape around the green pixel results in a non-occluded Fresnel zone plate. The masked Fresnel zone plates of red and blue pixels contain sharp cutoffs at their long-distance separated occlusion boundaries, and freeform shapes at occlusion boundaries with moderate distance separation and varying depth distribution. **c**, Comparison of

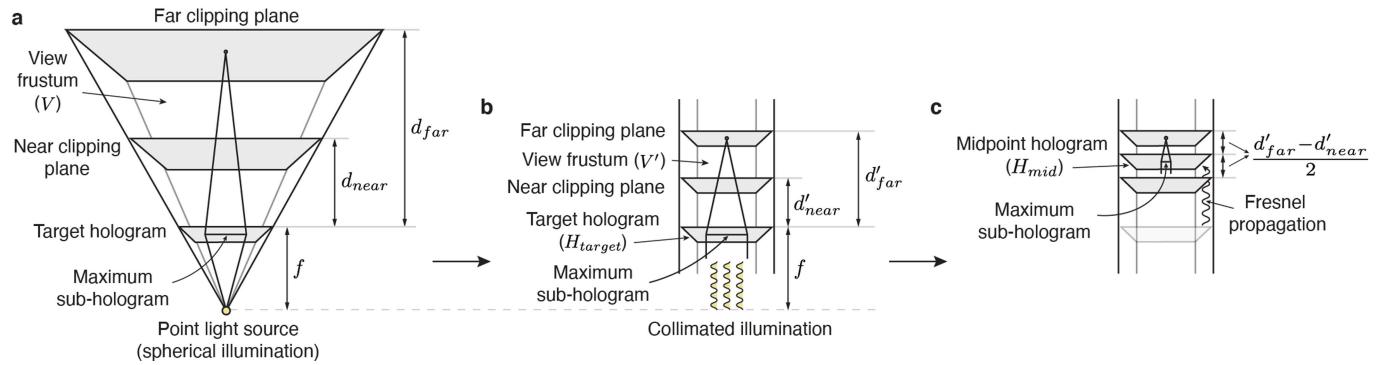
foreground reconstruction by the PBM, OA-PBM and Fresnel diffraction. The scene is a cropped modulation transfer function bar target with a step depth profile. The PBM leaks a considerable portion of the background into the foreground due to a lack of occlusion handling. The artefacts are clearly visible in the original unmagnified view. The OA-PBM removes a considerable portion of the artefacts and the remaining artefacts are visually inconsequential in the unmagnified view. **d**, Comparison of focal stacks reconstructed by the PBM and OA-PBM for the *Big Buck Bunny*. The orange bounding boxes mark the background leakage in the PBM reconstructions. **a, d**, Images reproduced from www.bigbuckbunny.org (© 2008, Blender Foundation) under a Creative Commons licence (<https://creativecommons.org/licenses/by/3.0/>).



Extended Data Fig. 2 | Samples of the MIT-CGH-4K dataset and comparison with the DeepFocus dataset. **a**, The RGB-D image, amplitude and phase of two samples from the MIT-CGH-4K dataset. The RGB image records the amplitude of the scene (directly visualized in sRGB space) and consists of large variations in colour, texture, shading and occlusion. The pixel depth has a statistically uniform distribution throughout the view frustum. The phase presents

high-frequency features at both occlusion boundaries and texture edges to accommodate rapid depth and colour changes. **b**, A sample RGB-D image from the DeepFocus dataset⁵¹. **c**, Histograms of pixel depth distribution computed for the MIT-CGH-4K dataset and the DeepFocus dataset. **b**, Image reproduced from ‘3D Scans from Louvre Museum’ by Benjamin Bardou under a Creative Commons licence (<https://creativecommons.org/licenses/by-nc/4.0/>).

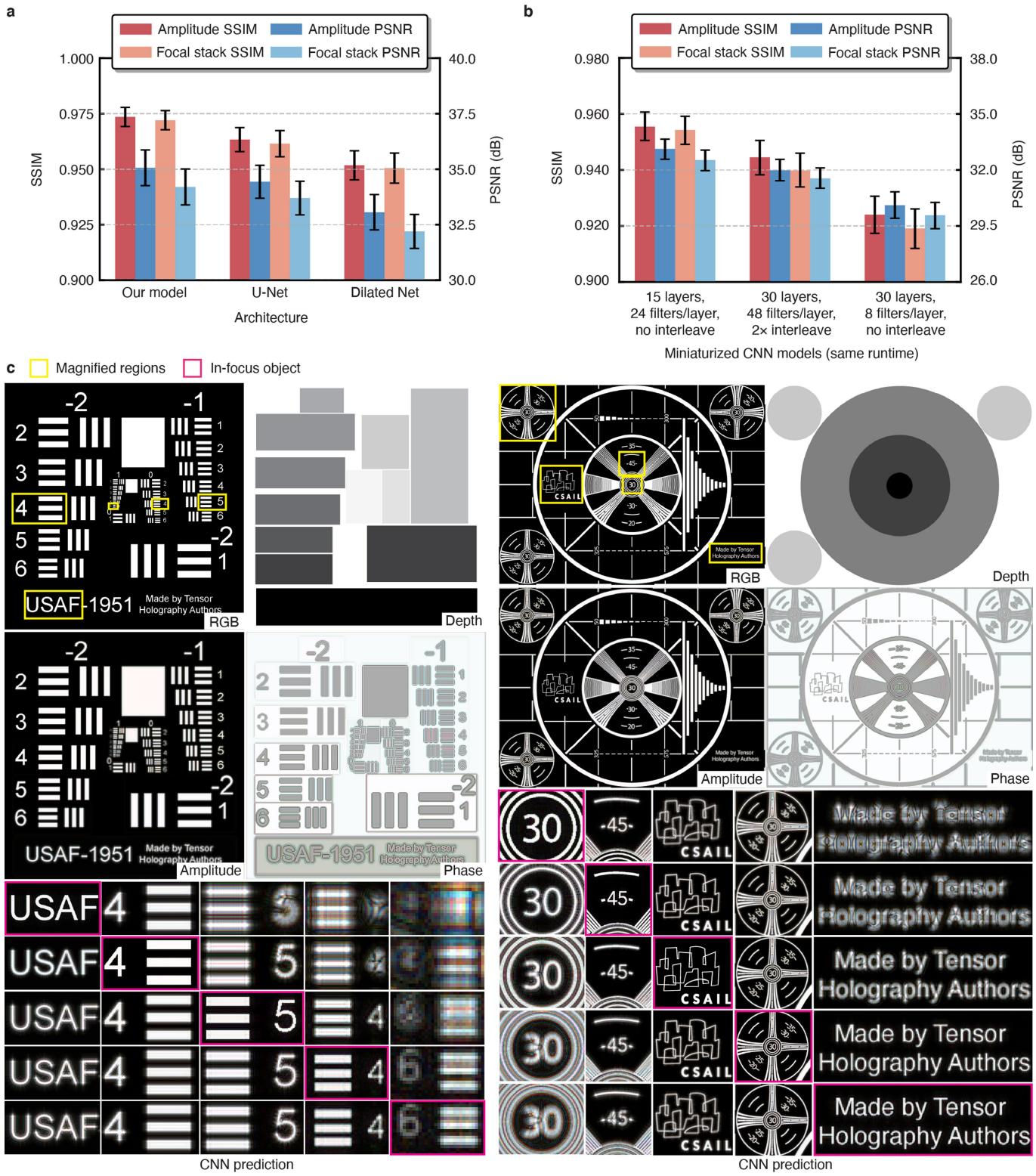
Article



Extended Data Fig. 3 | Schematic of the midpoint hologram calculation.

a, A holographic display magnified through a diverging point light source.
b, A holographic display unmagnified through the thin-lens formula.

c, The target hologram in this example is propagated to the centre of the unmagnified view frustum to produce the midpoint hologram. The width of the maximum subhologram is considerably reduced.



Extended Data Fig. 4 | Evaluation of tensor holography CNN on model architecture and test patterns. **a**, Performance comparison of different CNN architectures. **b**, Performance comparison of different CNN miniaturization

methods. **c**, CNN prediction of two standard test pattern (USAF-1951 and RCA Indian-head) variants made by the authors.

Article



Extended Data Fig. 5 | Evaluation of tensor holography CNN on additional computer-rendered scenes. **a, b**, CNN prediction of amplitude and phase along with focused reconstructions for holograms of a living room scene from the DeepFocus dataset⁵¹ (**a**) and a night landscape scene from the Stanford light field dataset²⁹ (**b**). **a**, Certain still images from 'ArchVizPRO Vol. 2' were used to

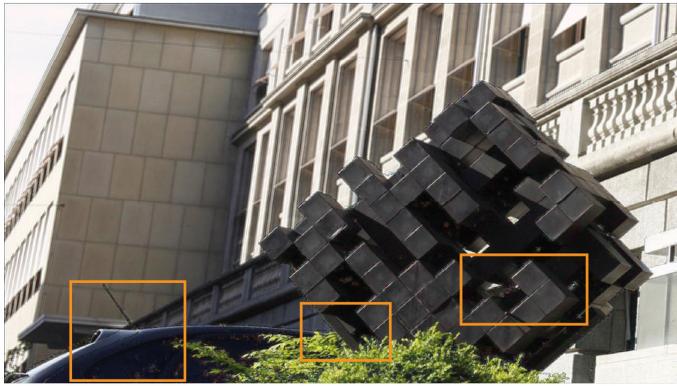
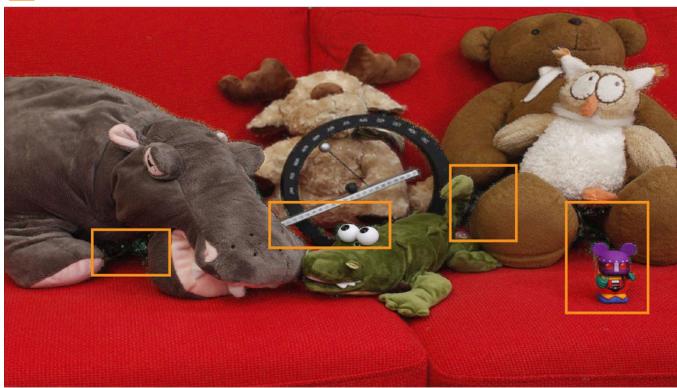
render new images for inclusion in this publication with the permission of the copyright holder (© Corridori Ruggero 2018), under a Creative Commons licence (<https://creativecommons.org/licenses/by-nc/4.0/>). Panel **b** reproduced with permission from ref.²⁹, ACM.



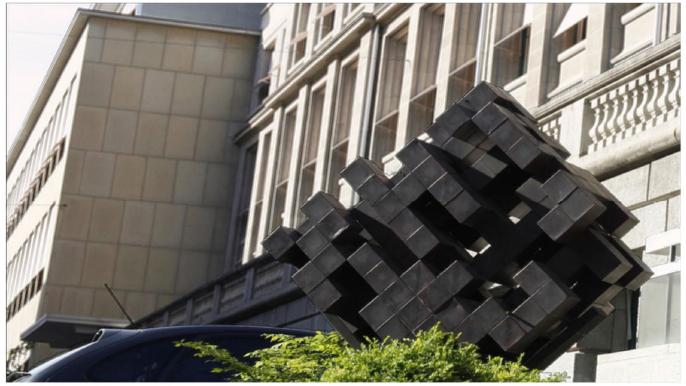
Extended Data Fig. 6 | Evaluation of tensor holography CNN on real-world captured scenes. **a, b,** CNN prediction of amplitude and phase along with focused reconstructions for holograms of a statue scene (**a**) and a mansion scene (**b**). Both scenes are from the ETH light field dataset⁴⁶.

Article

□ Region with strong artifacts



Double phase method

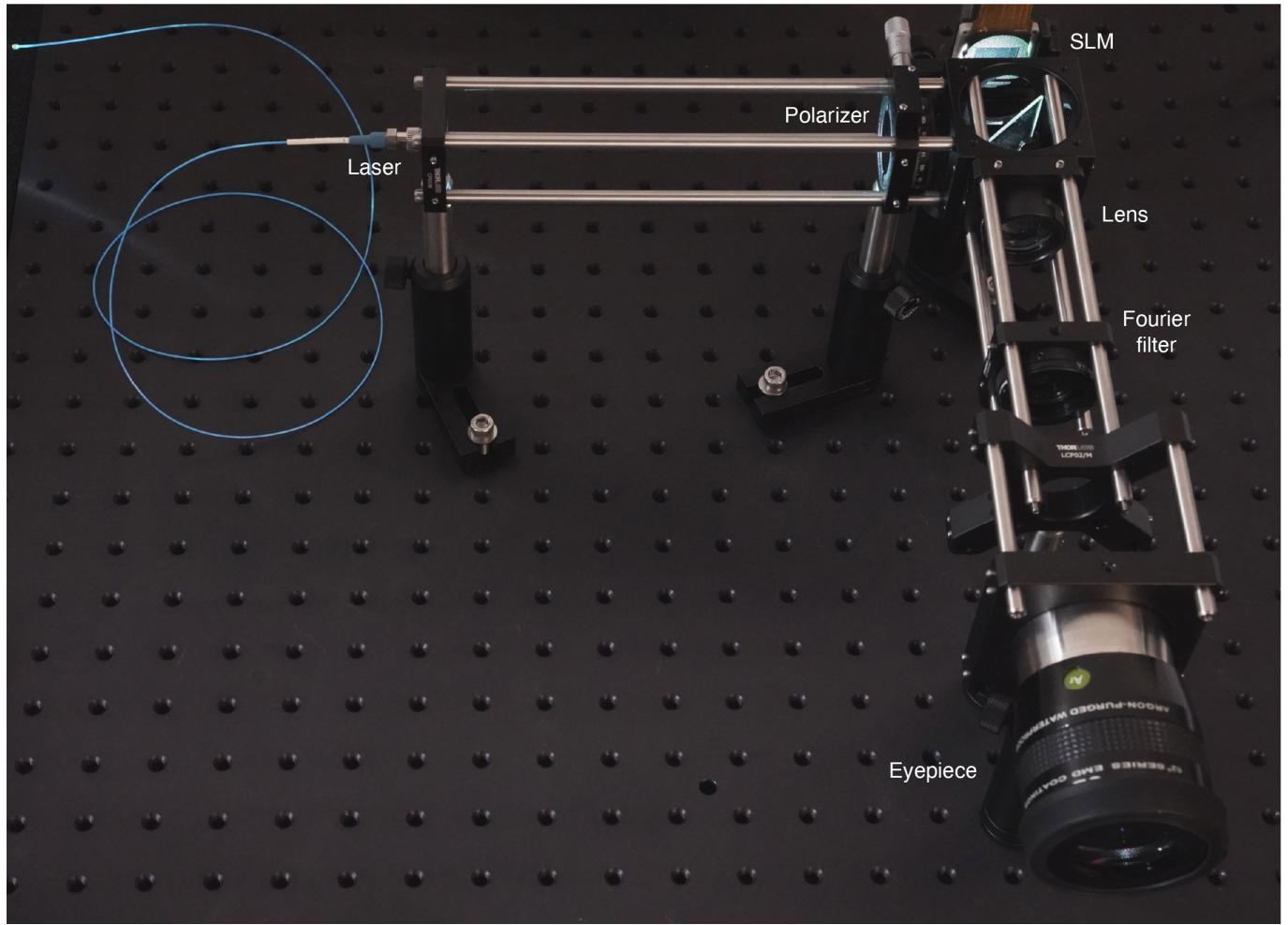


Anti-aliasing double phase method

Extended Data Fig. 7 | Comparison of the original DPM and the AA-DPM.

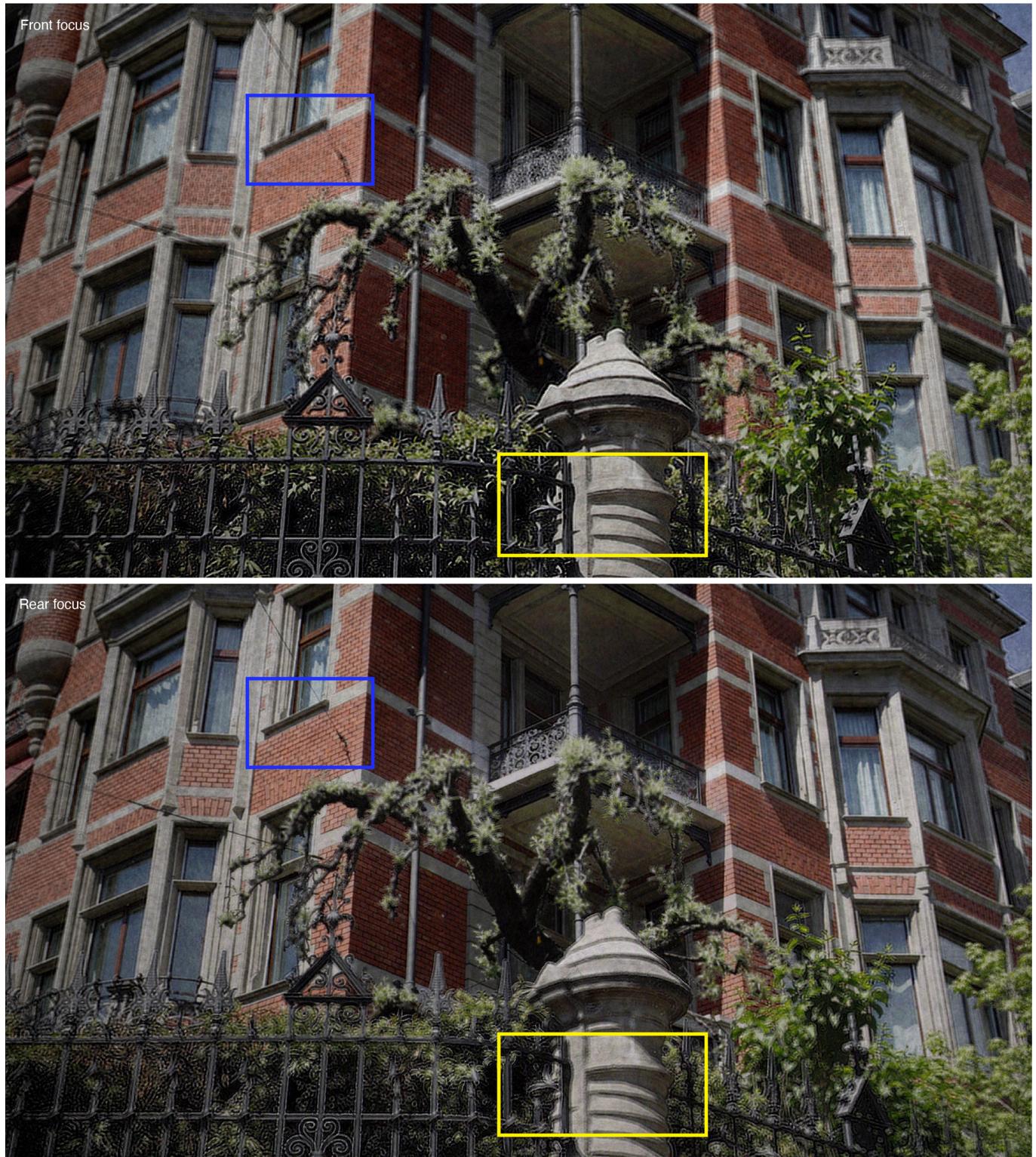
Reconstruction of two real-world scenes from the encoded phase-only holograms. The couch scene is focused on the mouse toy and the statue scene

is focused on the black statue. Orange bounding boxes highlight regions with strong high-frequency artefacts. Left: DPM. Right: AA-DPM.



Extended Data Fig. 8 | Holographic display prototype used for the experimental results shown in this paper. The control box of the laser, Labjack DAQ and camera are not visualized in the figure.

Article

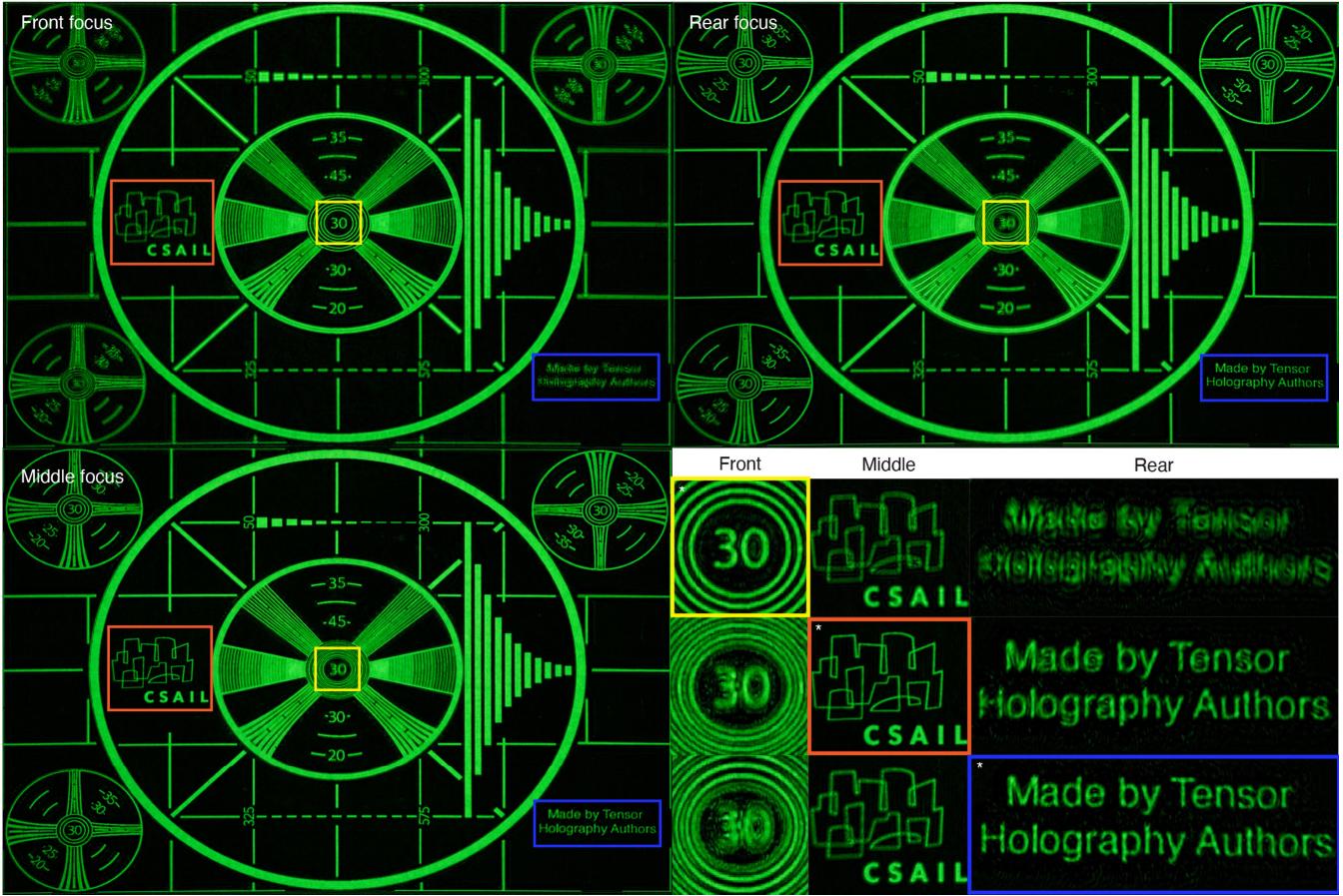


Extended Data Fig. 9 | Additional experimental demonstration of 3D holographic projection (part 1). The RGB-D input can be found in Extended Data Fig. 6.

a *: In-focus object



b



Extended Data Fig. 10 | Additional experimental demonstration of 3D holographic projection (part 2). The RGB-D inputs can be found in Extended Data Fig. 6 for **a**, and Extended Data Fig. 4 for **b**. Panel **a** reproduced with permission from ref.²⁹, ACM.