HoOPOE Supplementary Materials

S1 SCCGH Improvements

We detail SCCGH's loss function formulation in Eqn. 1 through Eqn. 4.

$$\text{Ach} = 0.299 \cdot I_{\lambda_r} + 0.587 \cdot I_{\lambda_g} + 0.114 \cdot I_{\lambda_b} \tag{1}$$

$$\text{RG} = I_{\lambda_r} - I_{\lambda_g} \tag{2}$$

$$\text{YV} = I_{\lambda_b} - (I_{\lambda_r} + I_{\lambda_g}) \tag{3}$$

$$\text{L} = \mathcal{L}(f(\widehat{\text{Ach}}), f(\text{Ach})) + \mathcal{L}(f(\widehat{\text{RG}}), f(\text{RG})) + \mathcal{L}(f(\widehat{\text{YV}}), f(\text{YV})) \tag{4}$$

In SCCGH's perceptual loss function, the authors first approximate cone-opponent responses in Eqn. 1 through Eqn. 3, for both the reference and test images. The cone-opponent channels of the reference image are represented as $\widehat{\text{Ach}}$ for the achromatic channel, $\widehat{\text{RG}}$ for the red-green channel, and $\widehat{\text{YV}}$ for the yellow-violet channel. The same channels are represented as Ach, RG, and YV respectively for the test image. MSE is then applied between the respective filtered channels of the reference and test images.

After cone-opponency calculation, the authors then filter out theoretically invisible signals by applying a low-pass filter (represented by $f(\cdot)$) to the achromatic and two chromatic channels. The low-pass filter for the achromatic channel is set to 75% of the width of the Fourier domain. The authors report that this effectively filters spatial frequencies above 30 cycles per degree. The chromatic channel low-pass filter is set to 45% of the Fourier domain, which eliminates signals above 18 cycles per degree.

We make two adjustments to their algorithm. First, we replace their color space conversion math in Eqn. 1 through Eqn. 3 with a more principled conversion to the DKL color space [Derrington et al. 1984]. The original color space conversion reported in SCCGH is not an accurate calculation of cone opponency. The luminance calculation in Eqn. 1 borrows constants from the luminance calculation for the BT.601 color space. However, the color space of the holographic display in their paper is much wider. Additionally, red-green and yellow-violet cone opponency responses can only be correctly calculated from a cone-response color space.

Second, we adjust the widths of the Fourier domain filters such that visual signals above 30 cycles per degree and 18 cycles per degree are filtered for the achromatic and chromatic channels, for our specific monitor and viewing distance.

S2 CSF Contrast Weighting

In existing CSF-based contrast weighting approaches [Mantiuk et al. 2021, 2024b; Tariq et al. 2023; Tursun et al. 2019], the contrast in Eqn. 5 is weighted by the contrast sensitivity function (CSF) as shown in Eqn. 5a:

$$C_w(p,s) = \frac{C(p,s)}{\text{CSF}(G(p,s+2), F(s))} \tag{5a}$$

$$= \frac{1}{G(p,s+2)\text{CSF}(G(p,s+2), F(s))} L(p,s) \tag{5b}$$

where $\text{CSF}(G(p,s+2), F(s))$ is the contrast sensitivity computed for pixel p at the Laplacian pyramid level s, parameterized by the local adapting luminance $G(p,s+2)$ and the dominant frequency $F(s)$ at layer s. Both CSF and F are defined for each color channel c, omitted here for simplicity.

Importantly, this contrast weighting can be reformulated to weight the absolute magnitude of the Laplacian layer, as shown in Eqn. 5b, yielding one possible specification of $w_{c,s} = 1/(G(p,s+2)\text{CSF}(G(p,s+2), F(s)))$.

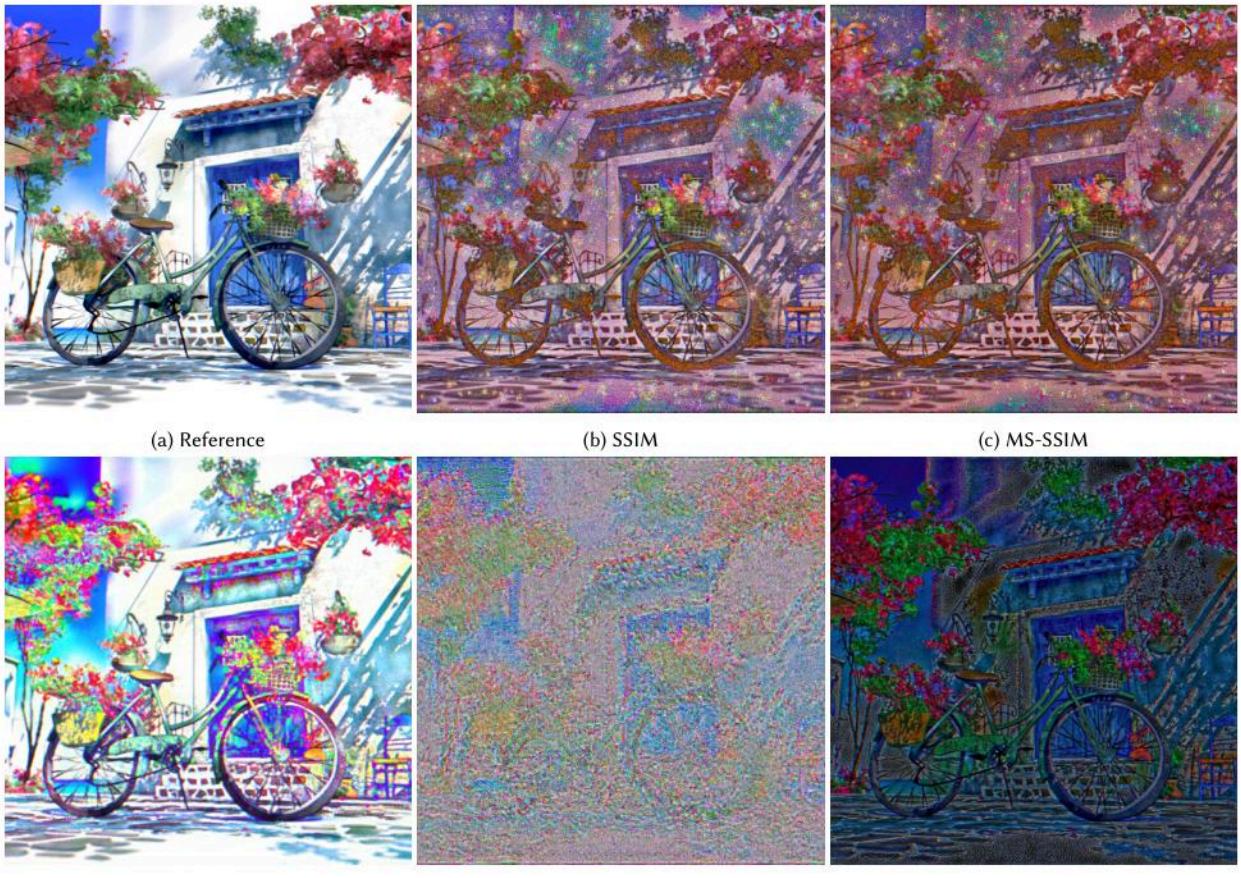
We do not use this approach for two main reasons. First, CSFs are typically defined in cone-opponent color spaces such as DKL. Because we operate in CIELAB space (Sec. 3.3), which is a non-linear transformation away from DKL, these CSF-based weights cannot be directly applied to our loss. Second, the CSF-weighted contrasts are usually subjected to additional processing (e.g., further weighting or pooling) using coefficients calibrated on particular artifacts. These coefficients may not be directly transferable to holography, and there are no large-scale subjective datasets available for a comprehensive recalibration.

S3 Excluded benchmarks

The following perceptual metrics were tested but excluded from the user study:

- MS-SSIM [Wang et al. 2003]
- SSIM [Wang et al. 2004]
- MILO [Cogalan et al. 2025]
- ColorVideoVDP [Mantiuk et al. 2024b]
- LPIPS (VGG) [Zhang et al. 2018]

MS-SSIM, SSIM, MILO, ColorVideoVDP, and LPIPS were excluded from the user study due to their poor performance. Fig. S1 showcases the performance of these five perceptual quality metrics as holography loss functions.



(a) Reference (b) SSIM (c) MS-SSIM

(d) CVVDP (e) MILO (f) LPIPS VGG

Fig. S1. Loss functions we excluded from the user study due to poor performance on the training set, even after Optuna hyperparameter tuning of learning rate and step decay rate. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

MSE in OKLAB and CAM16 UCS space were excluded as they were visually similar to MSE in LAB space. We opted to include MSE in LAB space for the ablation study, as HoOPOE is tabulated in LAB space. A comparison is shown in Fig. S2.



(a) MSE in LAB space (b) MSE in OKLAB space (c) MSE in CAM16 space

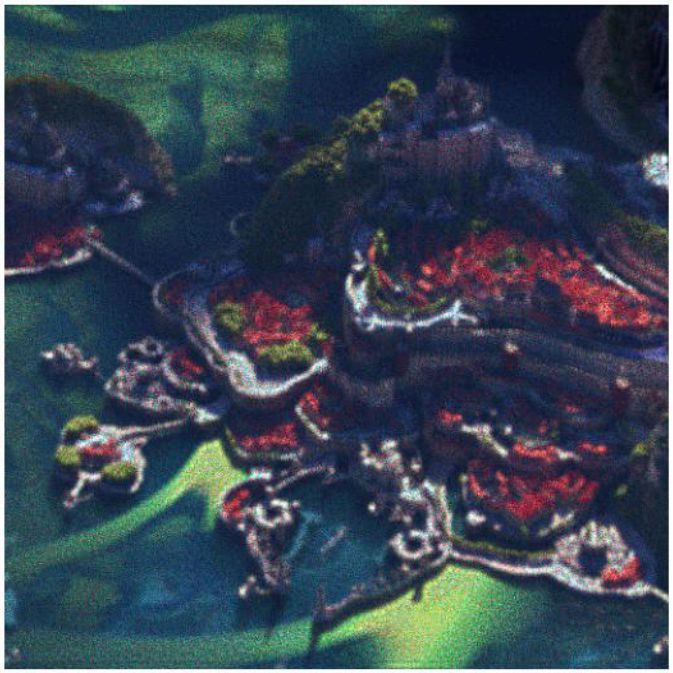
Fig. S2. MSE in LAB space vs OKLAB space vs CAM16 space. Trained on a focal stack with 5 images, at propagation distances of [0.5, 1.0, 1.5, 2.0, 2.5] cm. These three are single frame images. No temporal averaging is applied. These three losses perform perceptually similarly. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

S4 Incorporating Other Models of Visual Perception in HoOPOE

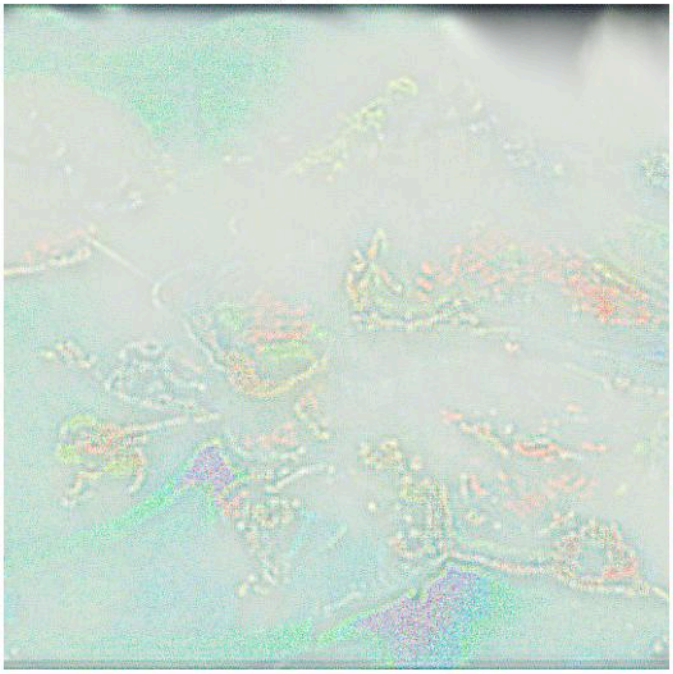
Prior to running our user studies, we also experimented with different variations of HoOPOE that accounted for visual masking, local contrast calculation, and models of contrast sensitivity. Below are examples of stimuli generated with each one of these HoOPOE variants. Due to their poor performance, they were excluded from further consideration. Copyright for the original 3D model shown below (licensed under CC-BY-SA) goes to Krynski [2022].



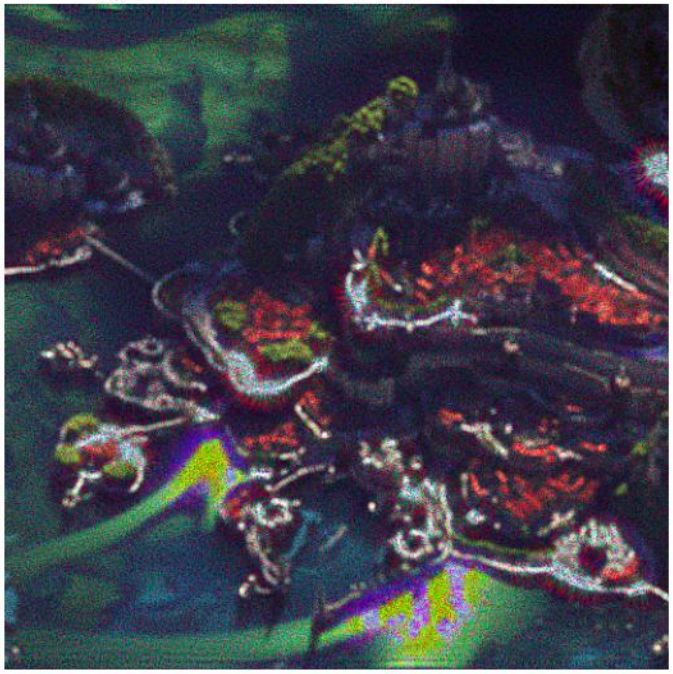
(a) Ground truth



(b) HoOPOE with local contrast calculation [Peli 1990]



(c) HoOPOE with CastleCSF-weighting [Ashraf et al. 2024]



(d) HoOPOE with visual masking

S5 Datasets Used

The following Blender demo scenes were used to construct the focal stack dataset:

- Blender 4.1 by Lynxdesign, CC-BY [Lynxsdesign 2024]
- Blender 4.0 by Gaku Tada, CC-BY-SA [Tada 2023]
- Blender 3.5 Cozy Kitchen by Nicole Morena, CC-BY-SA [Morena 2023]
- Blender 3.3 Scanlands by Piotr Krynski, CC-BY-SA [Krynski 2022]
- Blender 3.1 Secret Deer by Lorenzo Aiello, CC-BY-SA [Aiello 2022]

The focal stacks used are displayed in Fig. S4, Fig. S5, Fig. S6, Fig. S7, and Fig. S8.

For HoOPOE weight tuning, the Segment Anything 1B dataset was used. The following image IDs in particular were randomly selected as part of the training set:

- 45
- 2019
- 4105
- 5055
- 10468



(a) Plane 0 (closest)

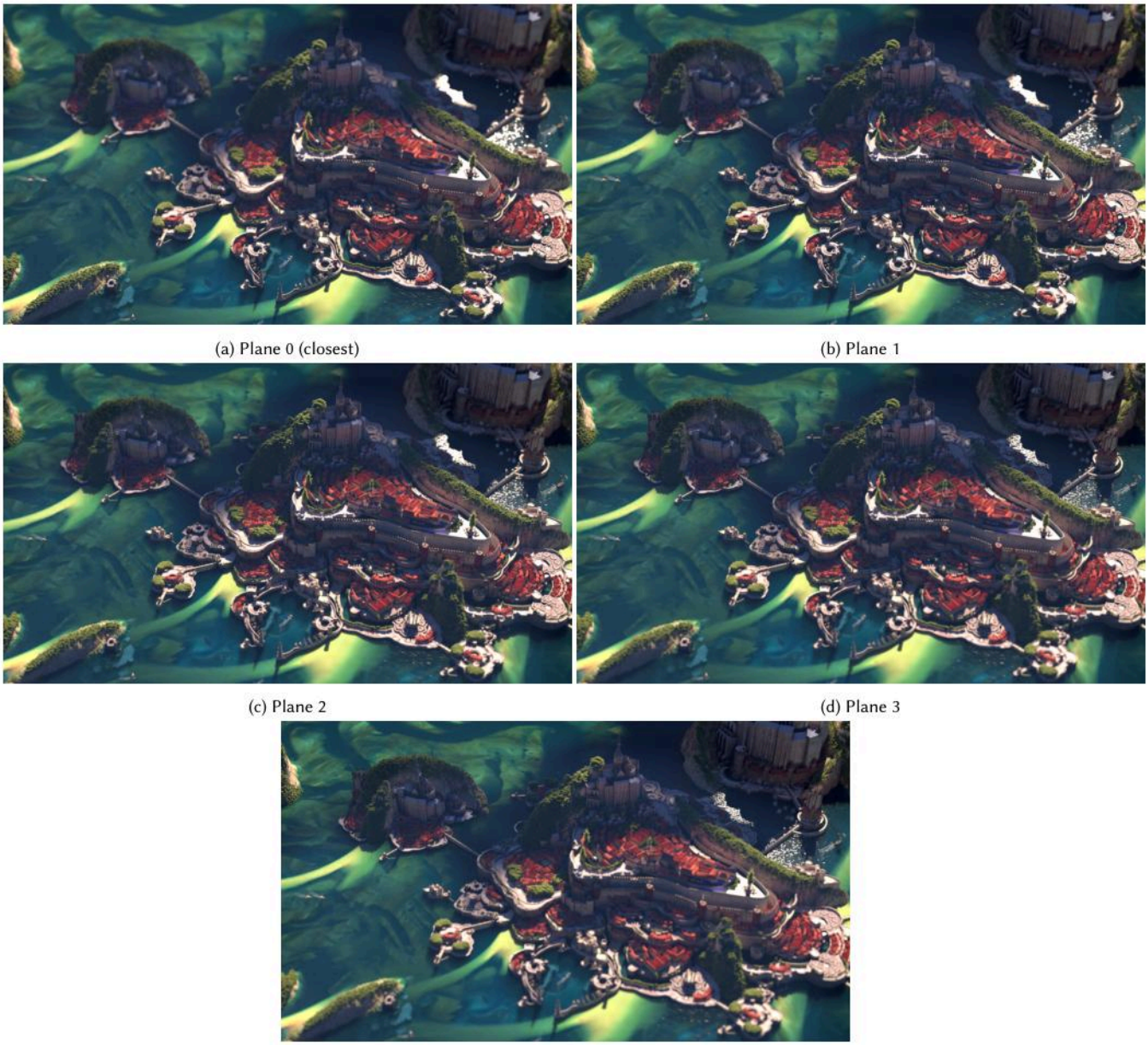
(b) Plane 1

(c) Plane 2

(d) Plane 3

(e) Plane 4 (furthest)

Fig. S4. 5-image focal stack. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].



(a) Plane 0 (closest)

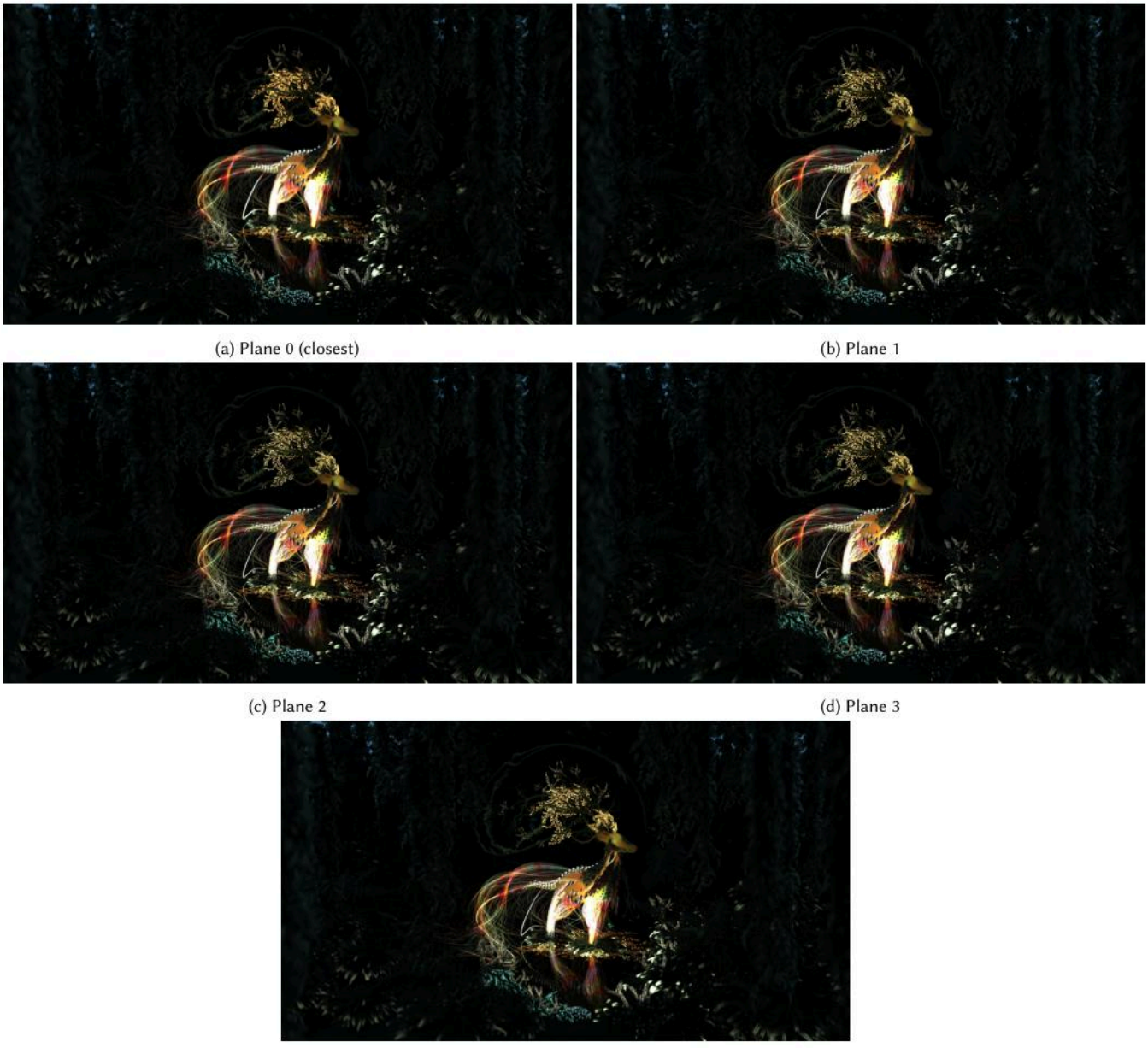
(b) Plane 1

(c) Plane 2

(d) Plane 3

(e) Plane 4 (furthest)

Fig. S5. 5-image focal stack. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].



(a) Plane 0 (closest)

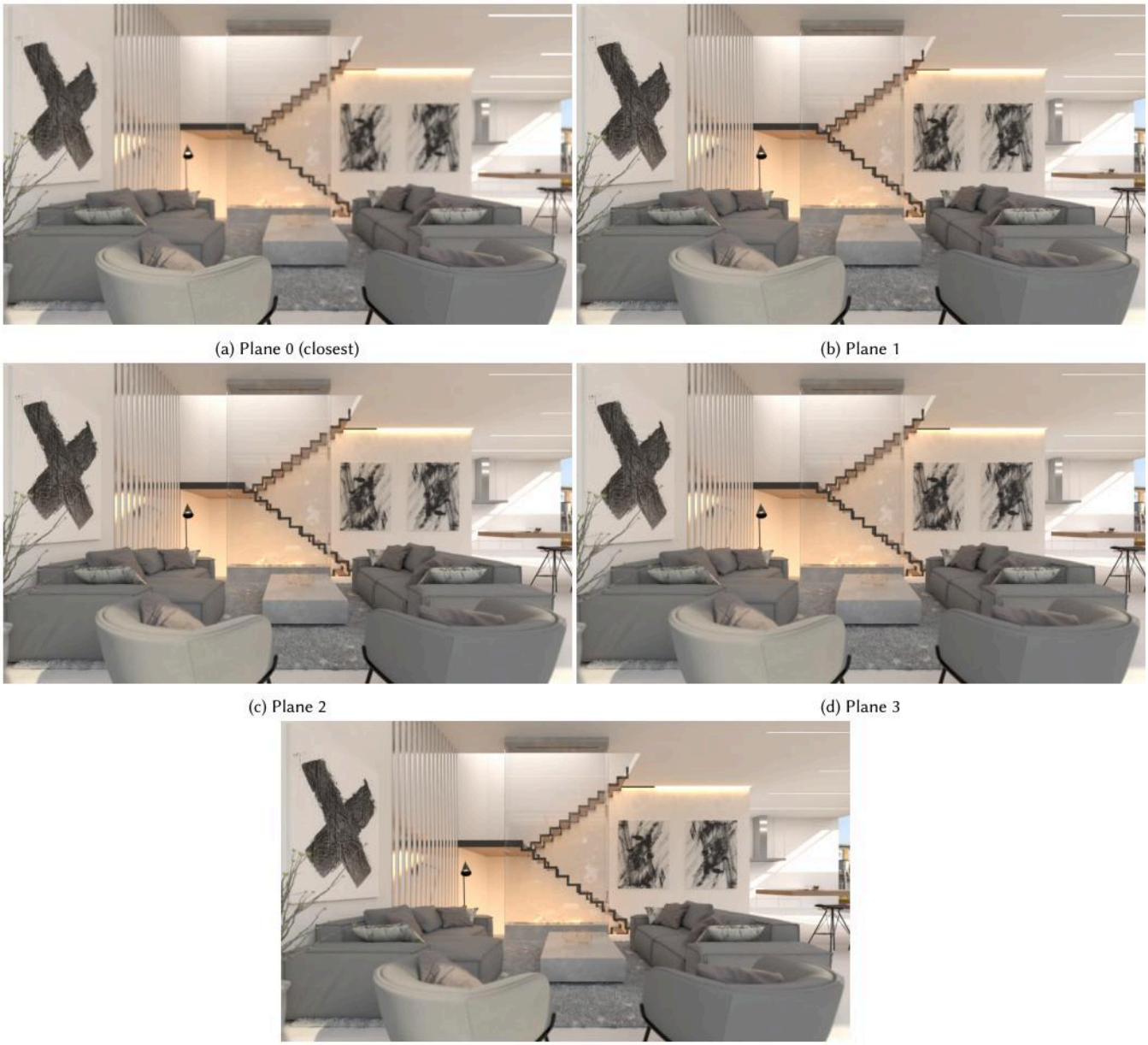
(b) Plane 1

(c) Plane 2

(d) Plane 3

(e) Plane 4 (furthest)

Fig. S6. 5-image focal stack. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Aiello [2022].



(a) Plane 0 (closest)

(b) Plane 1

(c) Plane 2

(d) Plane 3

(e) Plane 4 (furthest)

Fig. S7. 5-image focal stack. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].



(a) Plane 0 (closest)

(b) Plane 1

(c) Plane 2

(d) Plane 3

(e) Plane 4 (furthest)

Fig. S8. 5-image focal stack. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Morena [2023].

S6 HoOPOE as a Video Quality Metric

Despite being designed as a perceptual loss function, HoOPOE performs comparatively well as a perceptual video quality metric. Below are the Spearman's Rank Correlation Coefficients for HoOPOE alongside other perceptual metrics and loss functions over the XR-DAVID dataset [Chapiro et al. 2024]. Note that, despite scoring lower than ColorVideoVDP [Mantiuk et al. 2024b] as a perceptual quality metric, HoOPOE outperforms ColorVideoVDP as an objective function.

Table S1. SRCC comparison across metrics for the XR-DAVID dataset [Chapiro et al. 2024]. SRCC values for ColorVideoVDP, FovVideoVDP, MS-SSIM, SSIM, PSNR in sRGB space, and LPIPS taken from benchmark performed by Mantiuk et al. [2024a].

Metric	SRCC
ColorVideoVDP [Mantiuk et al. 2024b]	0.8913
FovVideoVDP [Mantiuk et al. 2021]	0.6918
HoOPOE	**0.6771**
PSNR (sRGB)	0.6432
MS-SSIM [Wang et al. 2003]	0.5597
MSE LAB [International Commission on Illumination 2019]	0.5545
SCCGH [Markley et al. 2023]	0.4615
SSIM [Wang et al. 2004]	0.4252
LPIPS (VGG) [Zhang et al. 2018]	0.4074

S7 Generated Stimuli

The 5 generated stimuli benchmarked in the user study, simulated at different frame rates via time averaging, are shown in the figures below. Note that since we are simulating the appearance of the effect of temporal multiplexing via time averaging, the estimated appearance of these images is only an approximation of the perceived content when observing in person. Some visual artifacts like flicker, which are very objectionable when viewing in person, are obscured by these visualizations.

We optimize our hologram using a 3D focal stack consisting of 5 images. The focal stacks are visualized in Supplementary Material S5. The propagation distances used in this study ranged from 0.5 cm to 2.5 cm, with planes evenly spaced.

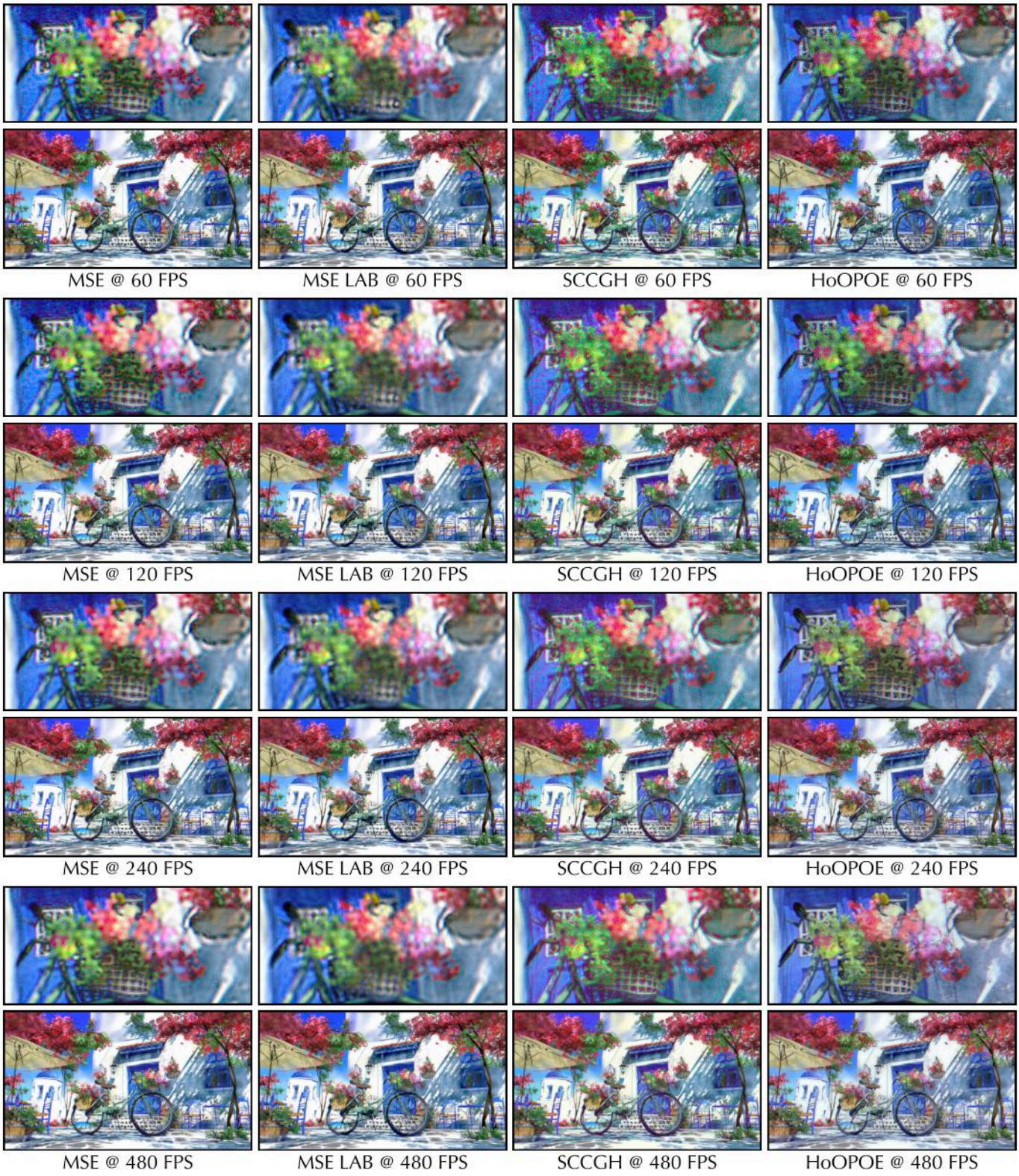


Fig. S9. Images generated by HoOPOE, MSE, MSE LAB, and SCCGH at various frame rates for stimulus 0. The ground truth of this stimulus is Fig. S4. Note that this rendering is only an approximation of real-life viewing. Some visual artifacts like flicker are invisible in this representation. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

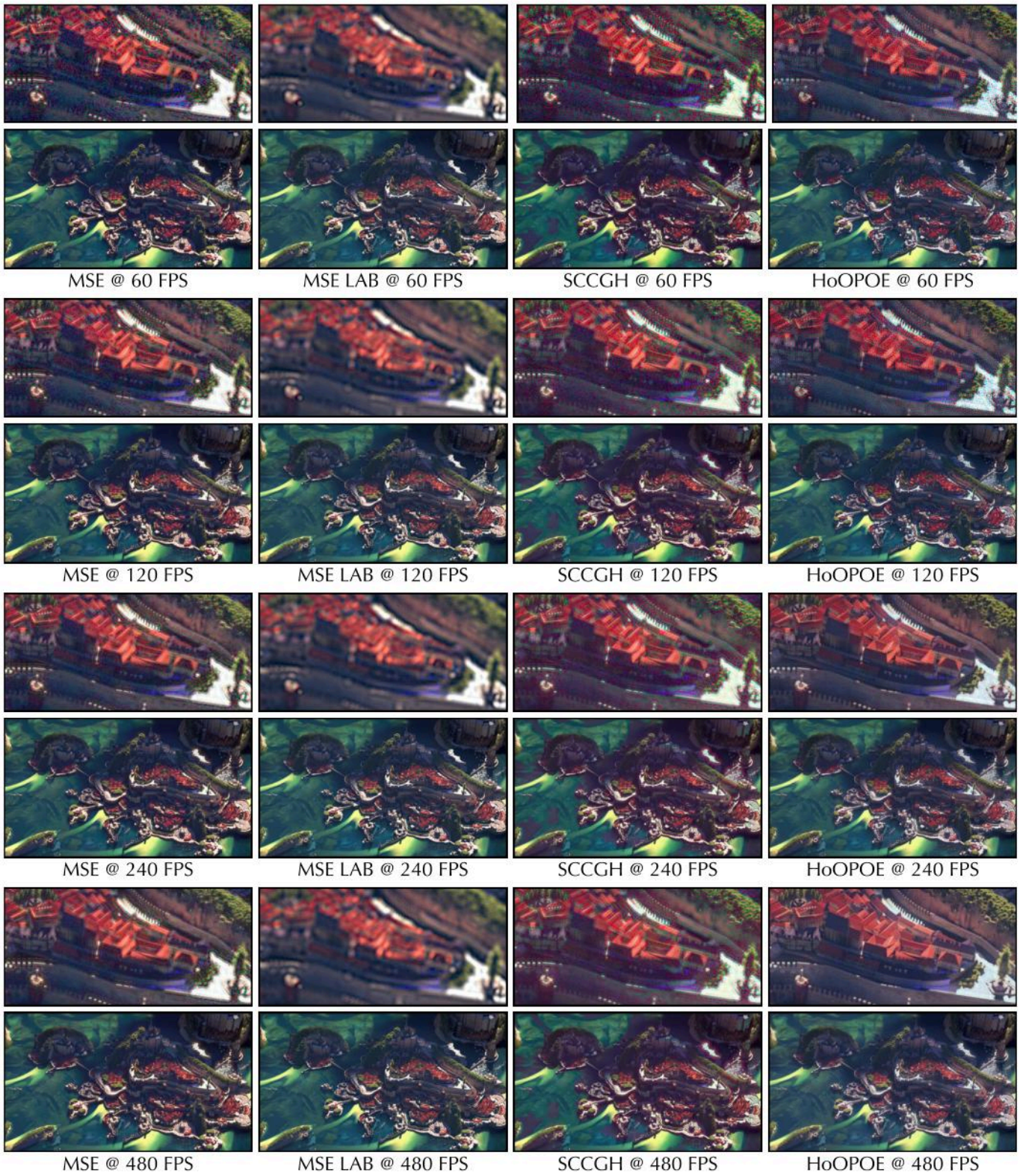


Fig. S10. Images generated by HoOPOE, MSE, MSE LAB, and SCCGH at various frame rates for stimulus 1. The ground truth of this stimulus is Fig. S5. Note that this rendering is only an approximation of real-life viewing. Some visual artifacts like flicker are invisible in this representation. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].

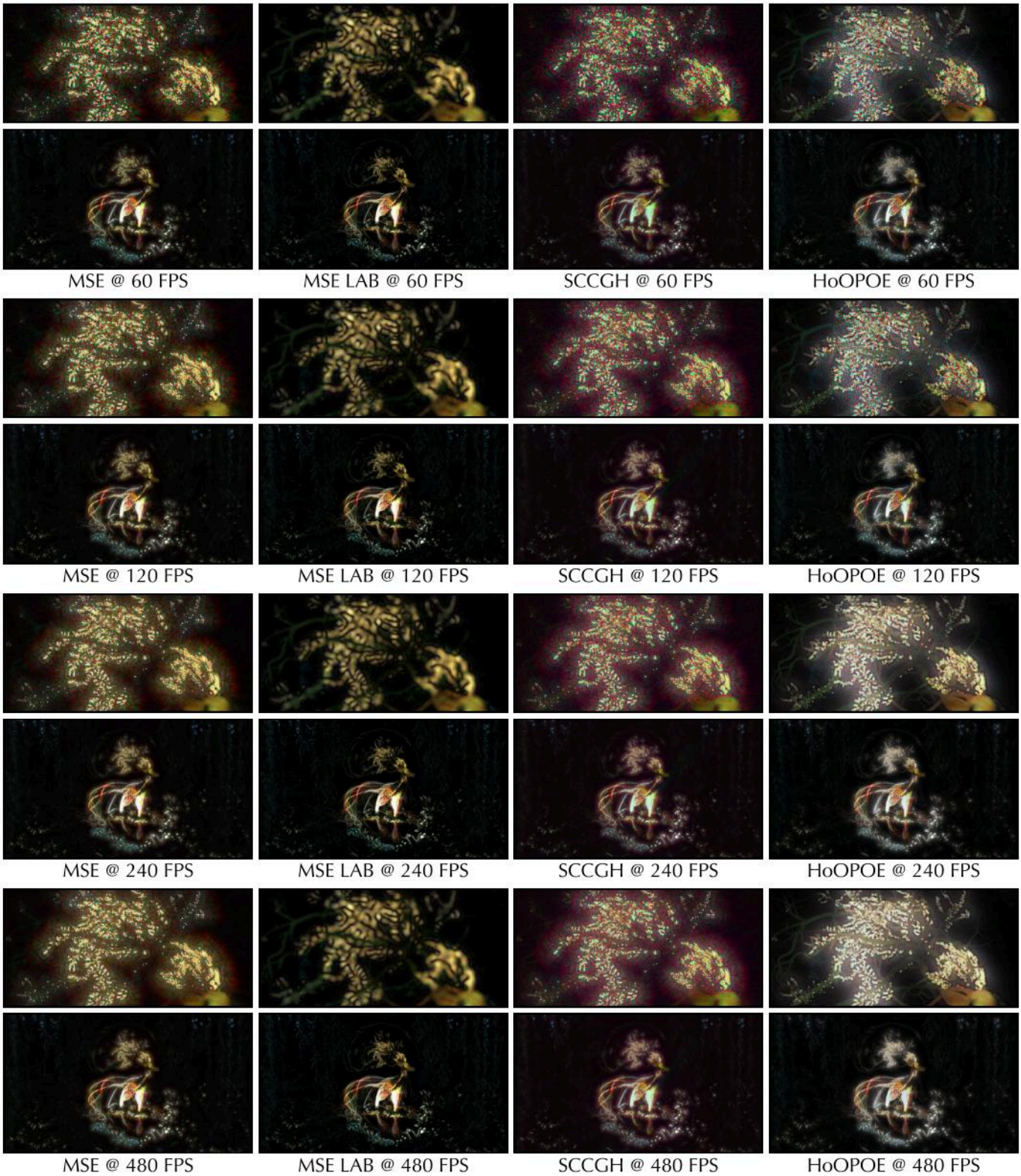


Fig. S11. Images generated by HoOPOE, MSE, MSE LAB, and SCCGH at various frame rates for stimulus 2. The ground truth of this stimulus is Fig. S6. Note that this rendering is only an approximation of real-life viewing. Some visual artifacts like flicker are invisible in this representation. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Aiello [2022].

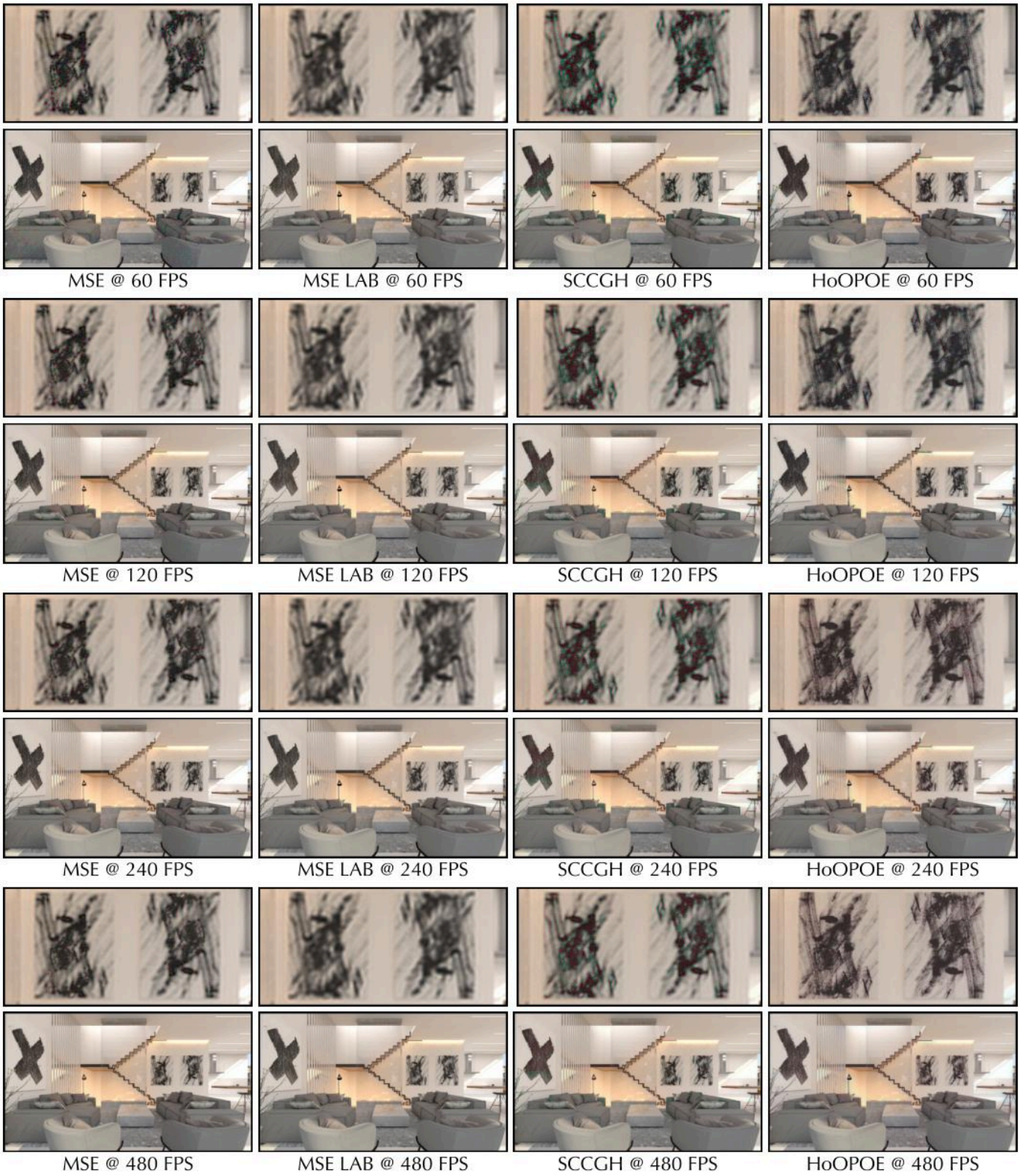


Fig. S12. Images generated by HoOPOE, MSE, MSE LAB, and SCCGH at various frame rates for stimulus 3. The ground truth of this stimulus is Fig. S7. Note that this rendering is only an approximation of real-life viewing. Some visual artifacts like flicker are invisible in this representation. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].

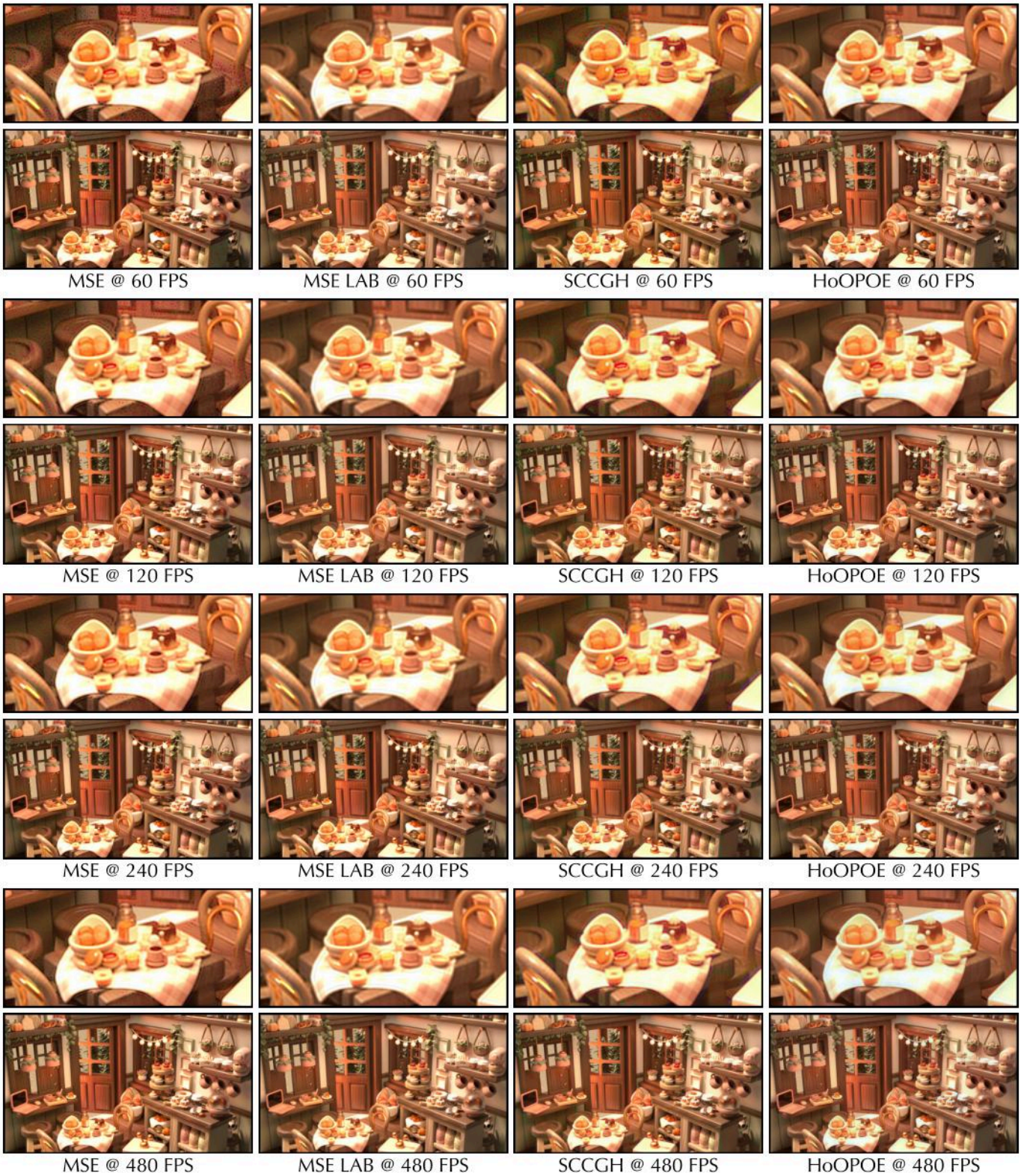


Fig. S13. Images generated by HoOPOE, MSE, MSE LAB, and SCCGH at various frame rates for stimulus 4. The ground truth of this stimulus is Fig. S8. Note that this rendering is only an approximation of real-life viewing. Some visual artifacts like flicker are invisible in this representation. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Morena [2023].

S8 JOD Values by Stimulus

We show the detailed JOD comparison between our method and the baselines for each of the stimuli used in the validation study described in Sec. 4.

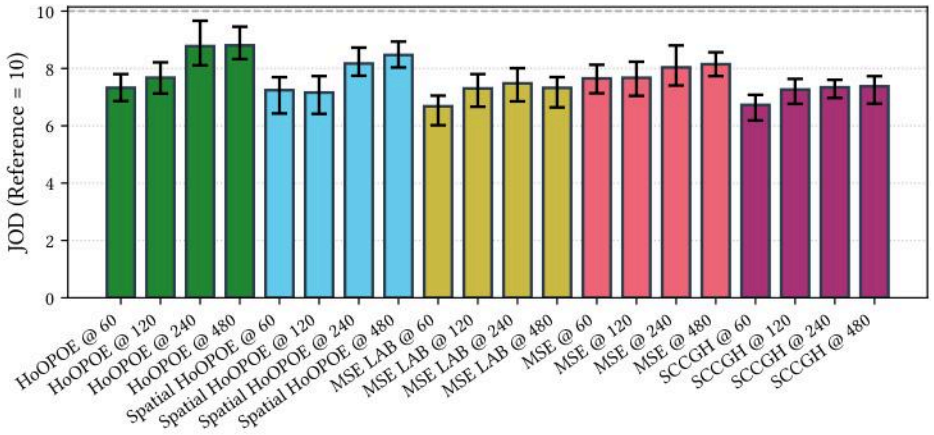


Fig. S14. JOD values for stimulus 0.

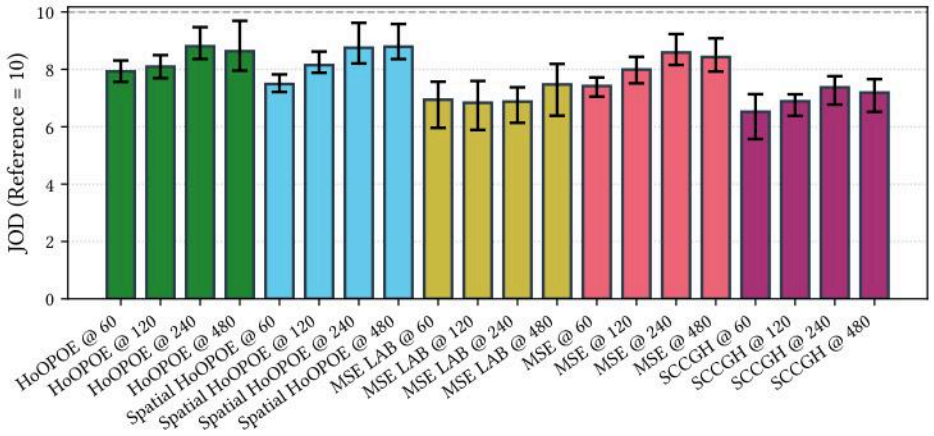


Fig. S15. JOD values for stimulus 1.

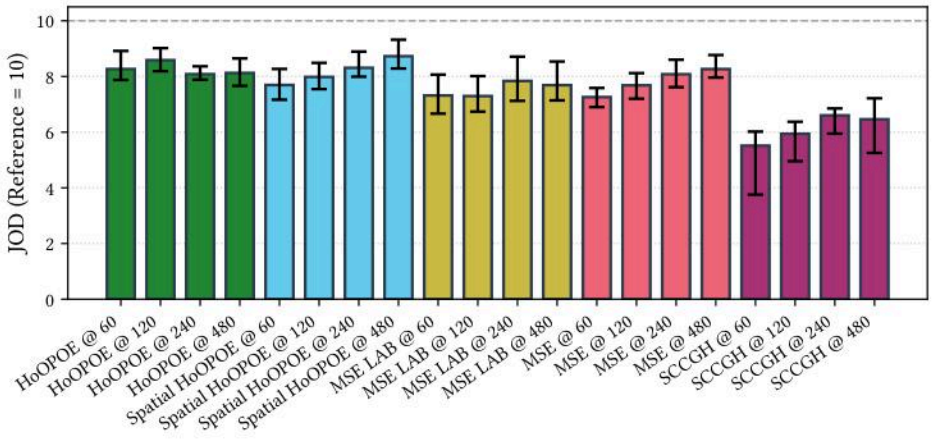


Fig. S16. JOD values for stimulus 2.

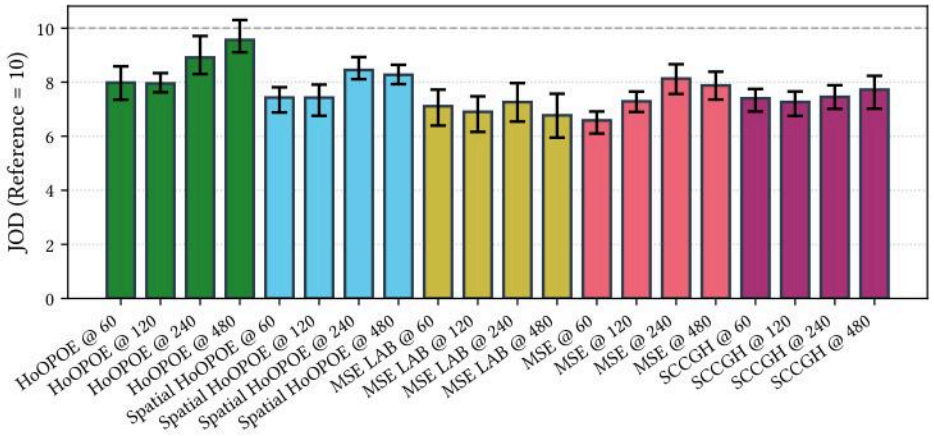


Fig. S17. JOD values for stimulus 3.

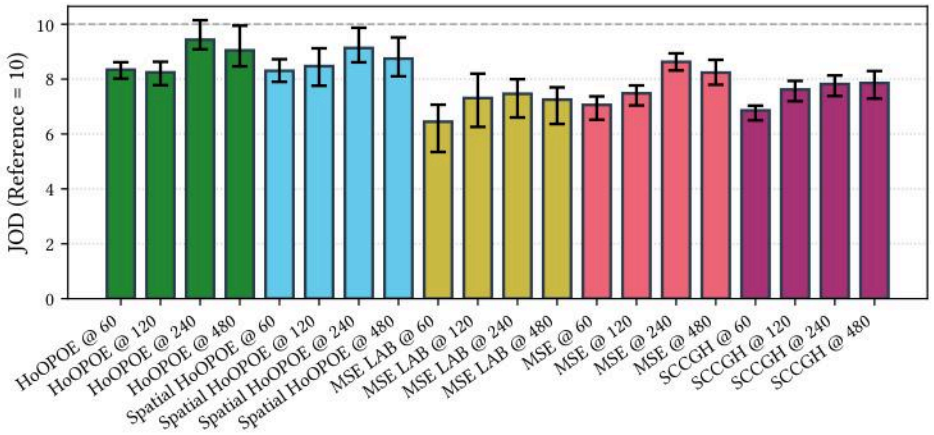


Fig. S18. JOD values for stimulus 4.

S9 Validation Study Significance

We show in Fig. S19 the full pairwise comparison results between all 21 conditions; see caption for details.

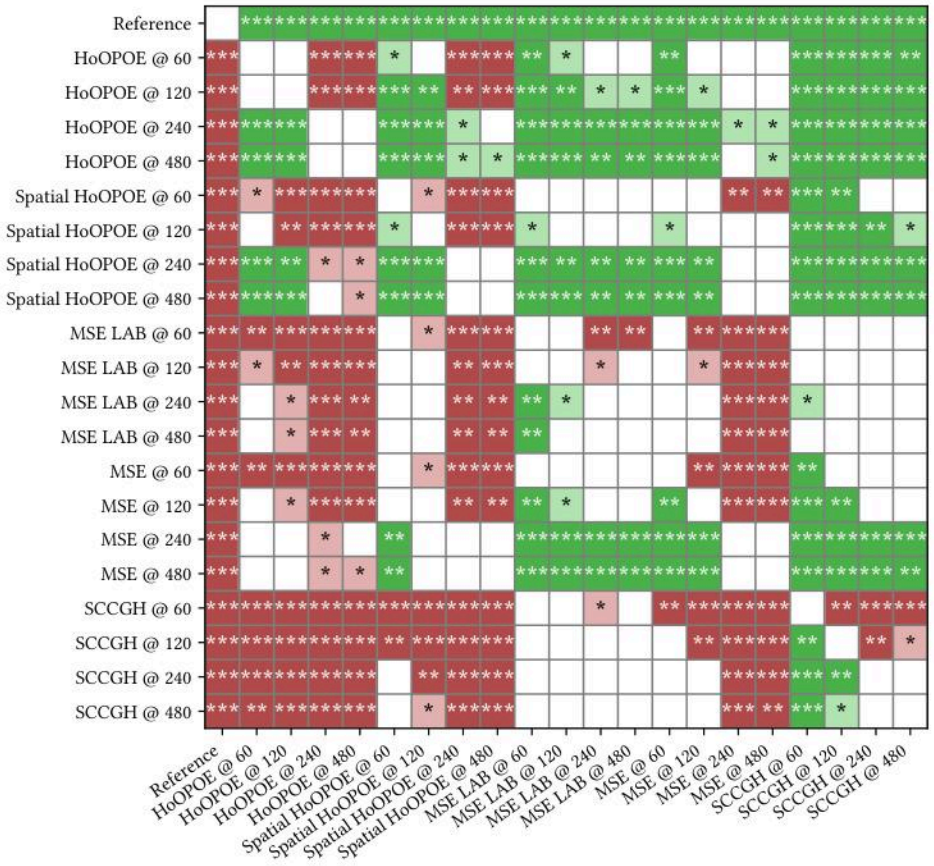


Fig. S19. 2D grid showing the statistical significance of each observed difference in JOD. Red cells indicate that the benchmark on the Y axis performed worse than the corresponding benchmark on the X axis. A single star indicates $p < 0.05$. Two stars indicate $p < 0.01$ and three stars indicate $p < 0.001$.

S10 Validation on a Different Model of ASM Propagation

We additionally benchmarked HoOPOE on a different ASM propagation model based on Matsushima and Shimobaba [2009]. We made the following assumptions for the CGH generation in this user study:

- Optimizing simultaneously for two distances between SLM and observer: 4 cm and 4.5 cm
- Phase-only SLM
- RGB laser wavelengths: 640 nm, 520 nm, 460 nm
- Pitch size: 6.4 μm
- SLM resolution: 512×512
- ROI: 512×512

Unlike the validation study setup described in Sec. 4, which utilized a 2IFC psychophysical paradigm, we instead used a 2AFC task. A visualization of the task is shown below in Fig. S20. Instead of viewing the three stimuli in separate intervals, all three stimuli are present on the screen simultaneously in this task. Therefore, the JOD values in this iteration of the study and the validation study discussed in the main paper are *not* comparable due to the change in psychophysical methodology.

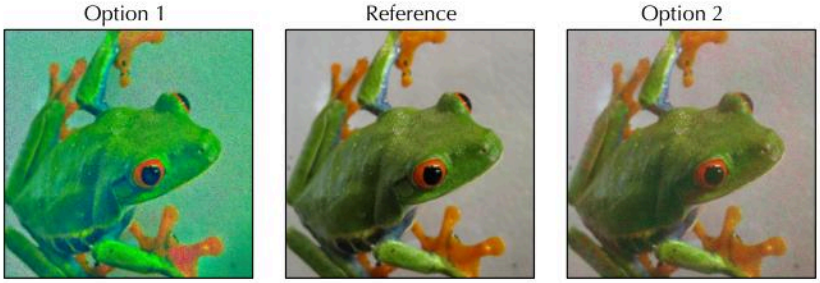


Fig. S20. An example of a 2AFC task.

The JOD plots of each loss function across frame rates and stimuli can be found in Fig. S21, Fig. S22, Fig. S23, Fig. S24, Fig. S25, and Fig. S26. The significance matrix for our psychophysical study on the different model of ASM propagation can be found in Fig. S27. Overall, HoOPOE again outperforms all baselines.

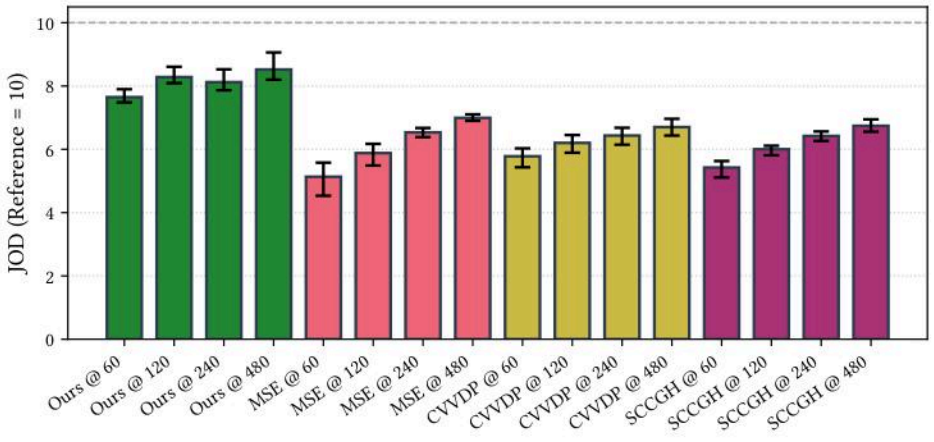


Fig. S21. JOD values for all stimuli.

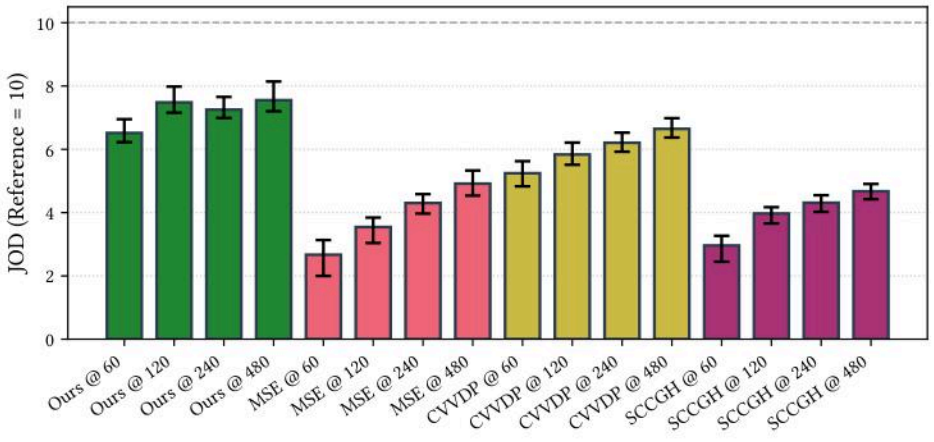


Fig. S22. JOD values for stimulus 0.

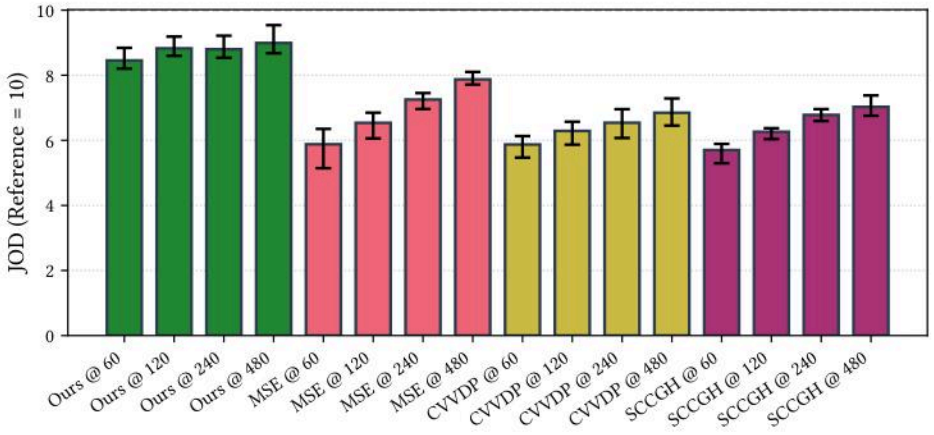


Fig. S23. JOD values for stimulus 1.

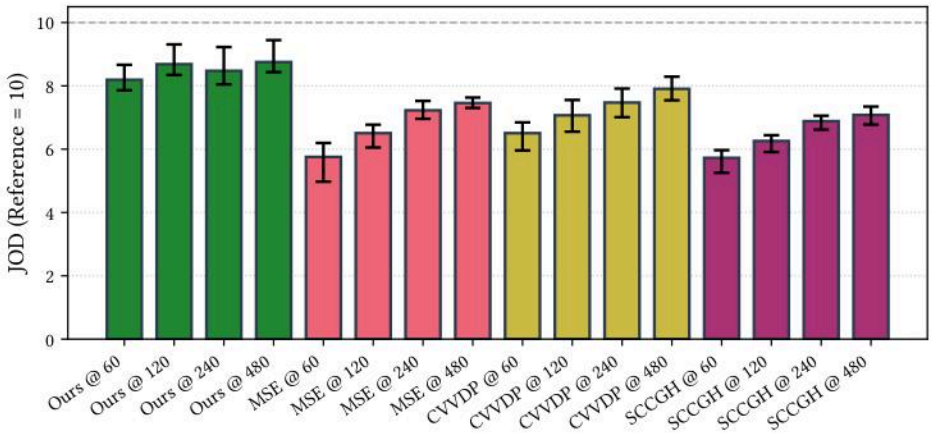


Fig. S24. JOD values for stimulus 2.

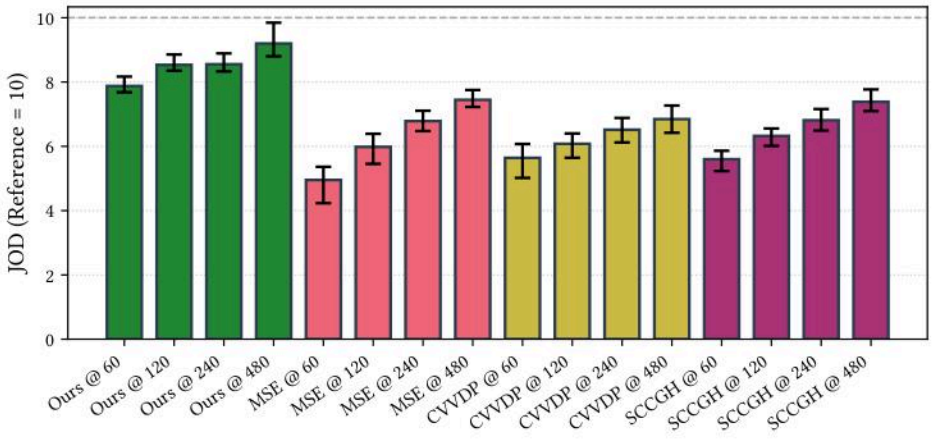


Fig. S25. JOD values for stimulus 3.

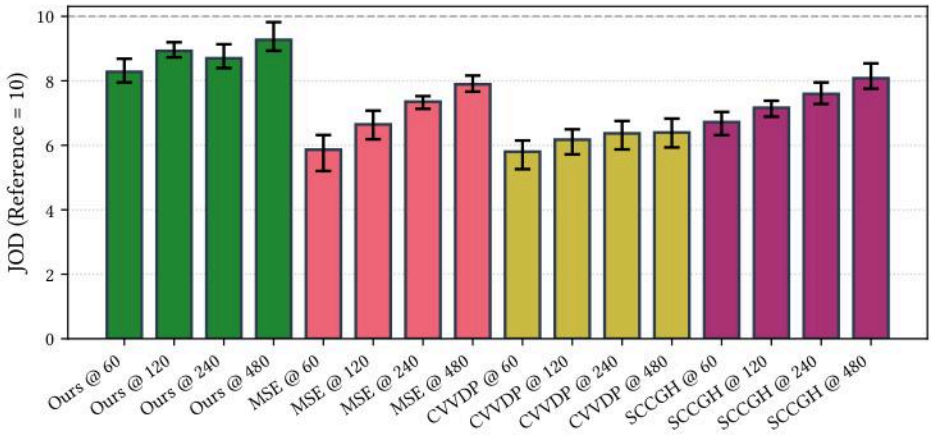


Fig. S26. JOD values for stimulus 4.

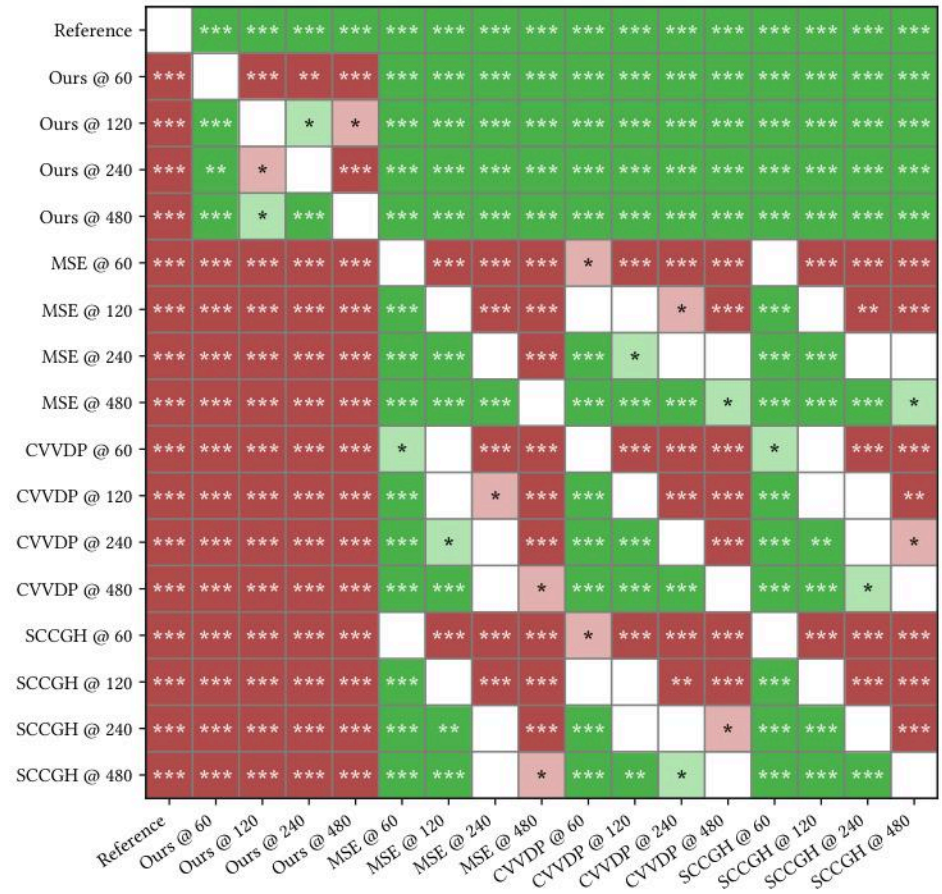


Fig. S27. 2D grid showing the statistical significance of each observed difference in JOD. Red cells indicate that the benchmark on the Y axis performed worse than the corresponding benchmark on the X axis. A single star indicates $p < 0.05$. Two stars indicate $p < 0.01$ and three stars indicate $p < 0.001$.

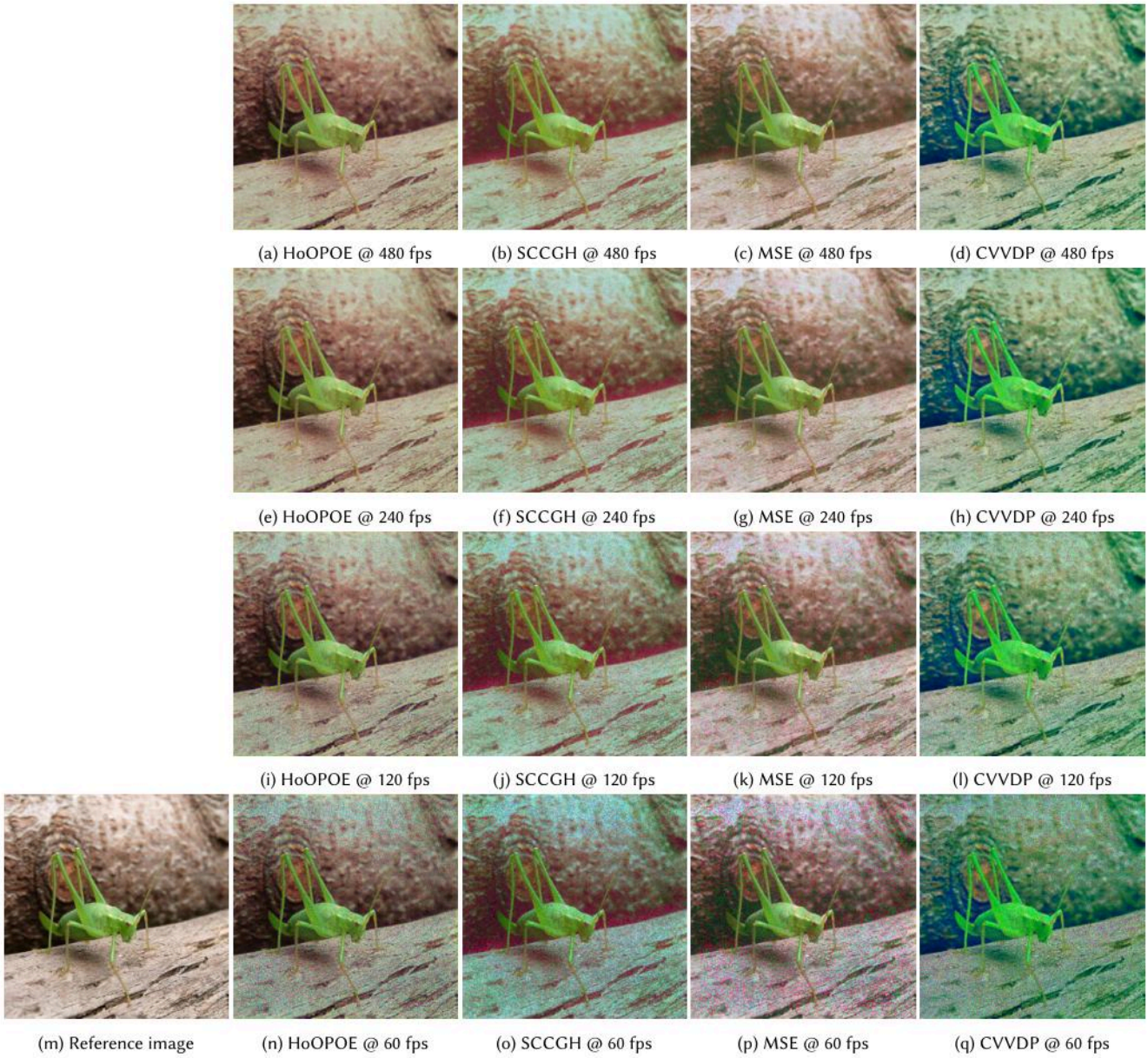
The 5 generated stimuli benchmarked in the different ASM propagator user study, simulated at different frame rates via time averaging, are shown in the figures below. Note that since we are simulating the appearance of the effect of temporal multiplexing via time averaging, the estimated appearance of these images is only an approximation of the perceived content when observing in person. The reference images for this study are sourced from ImageNet [Deng et al. 2009]. The reference image for stimulus 0 is in the public domain [AndrewH 2002].

We optimize our hologram to appear coherent from two viewing distances from the SLM. The stimulus generated for the near viewing plane is listed in Sec. S10.1. The stimulus for the far viewing plane is in Sec. S10.2. The near plane stimulus was used in the validation study.

S10.1 Near viewing plane (4 cm)



Fig. S28. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 0, for the near viewing plane. Original public domain image taken by AndrewH [AndrewH 2002].



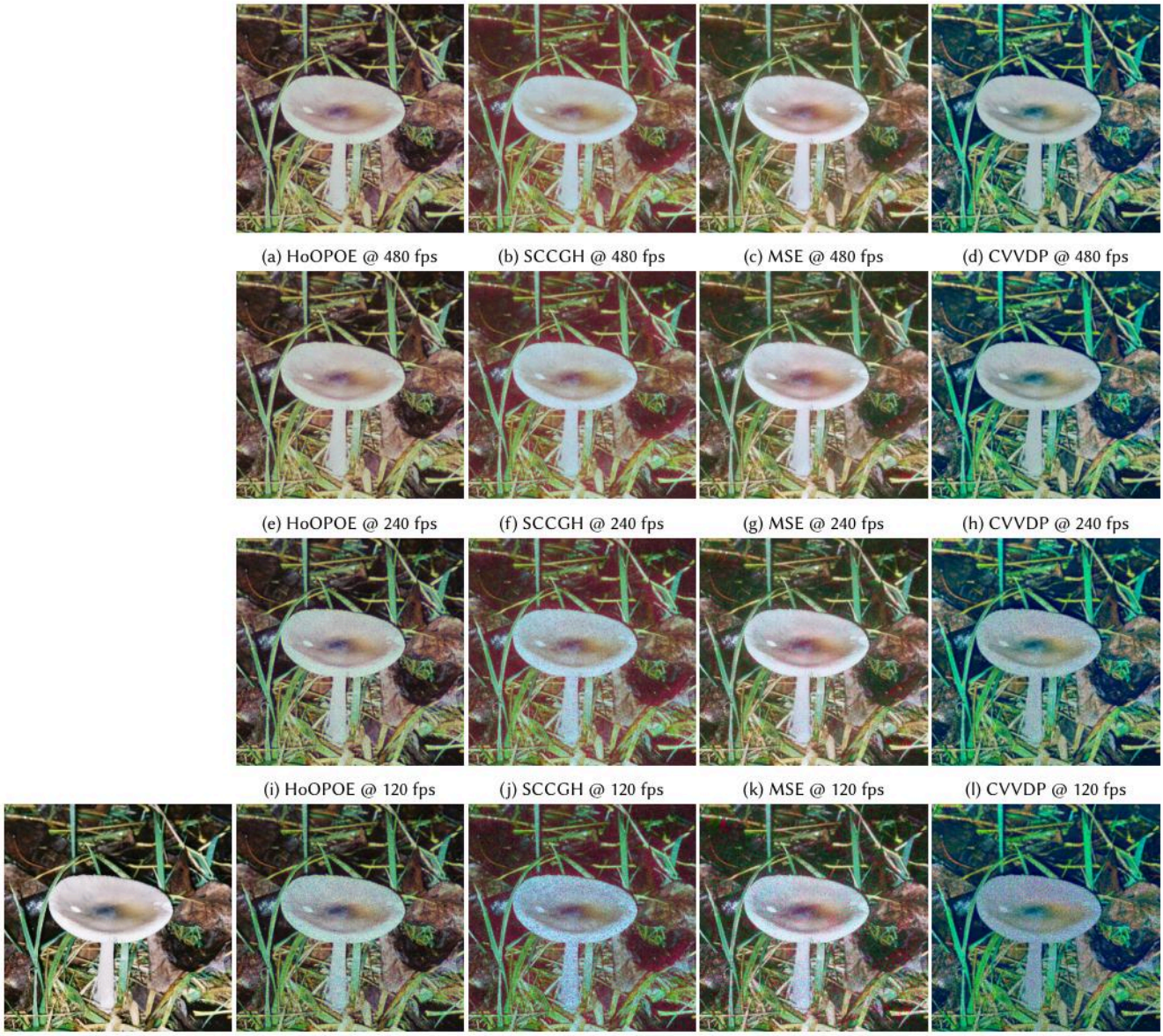
(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S29. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 1, for the near viewing plane



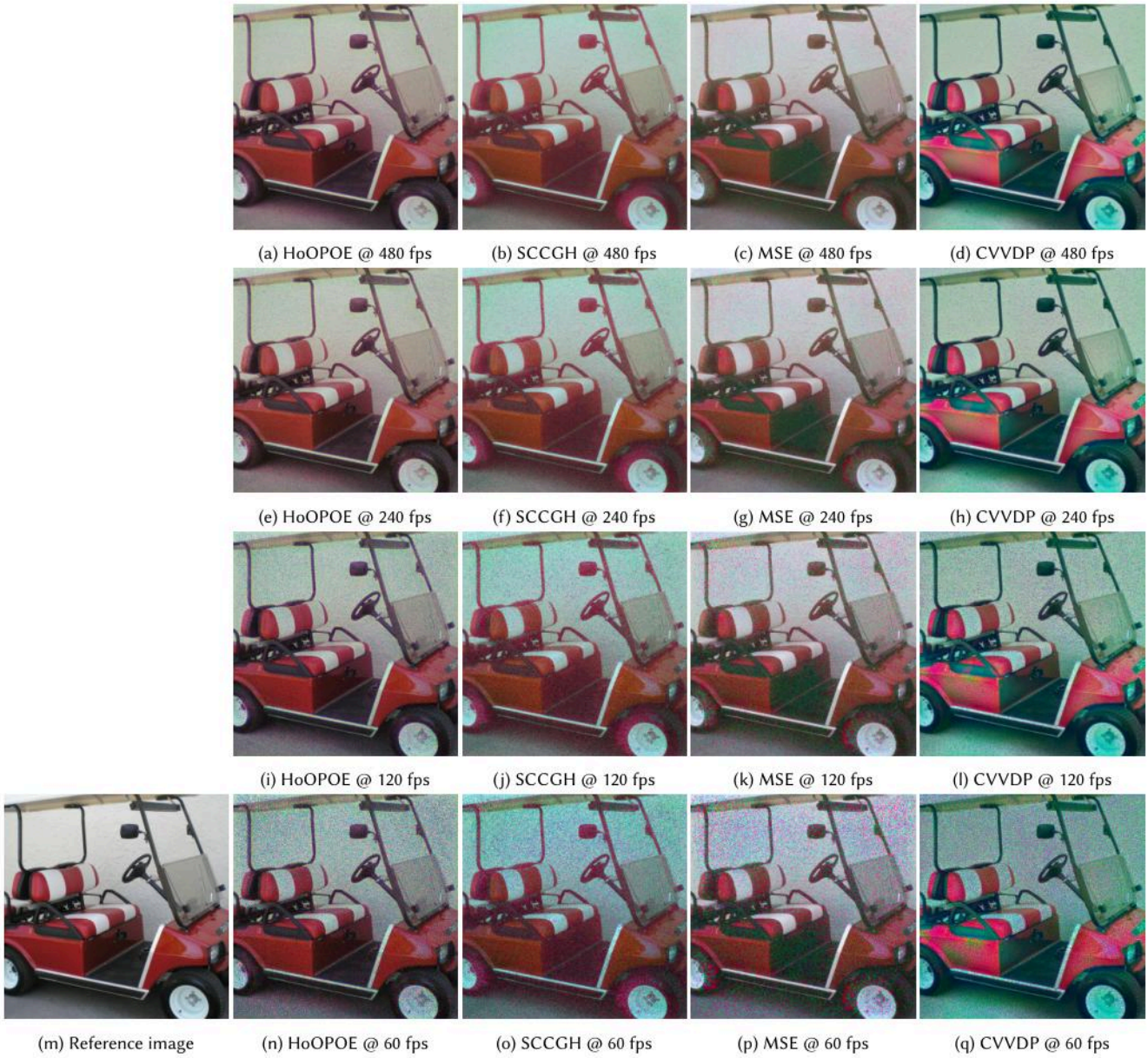
(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S30. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 2, for the near viewing plane



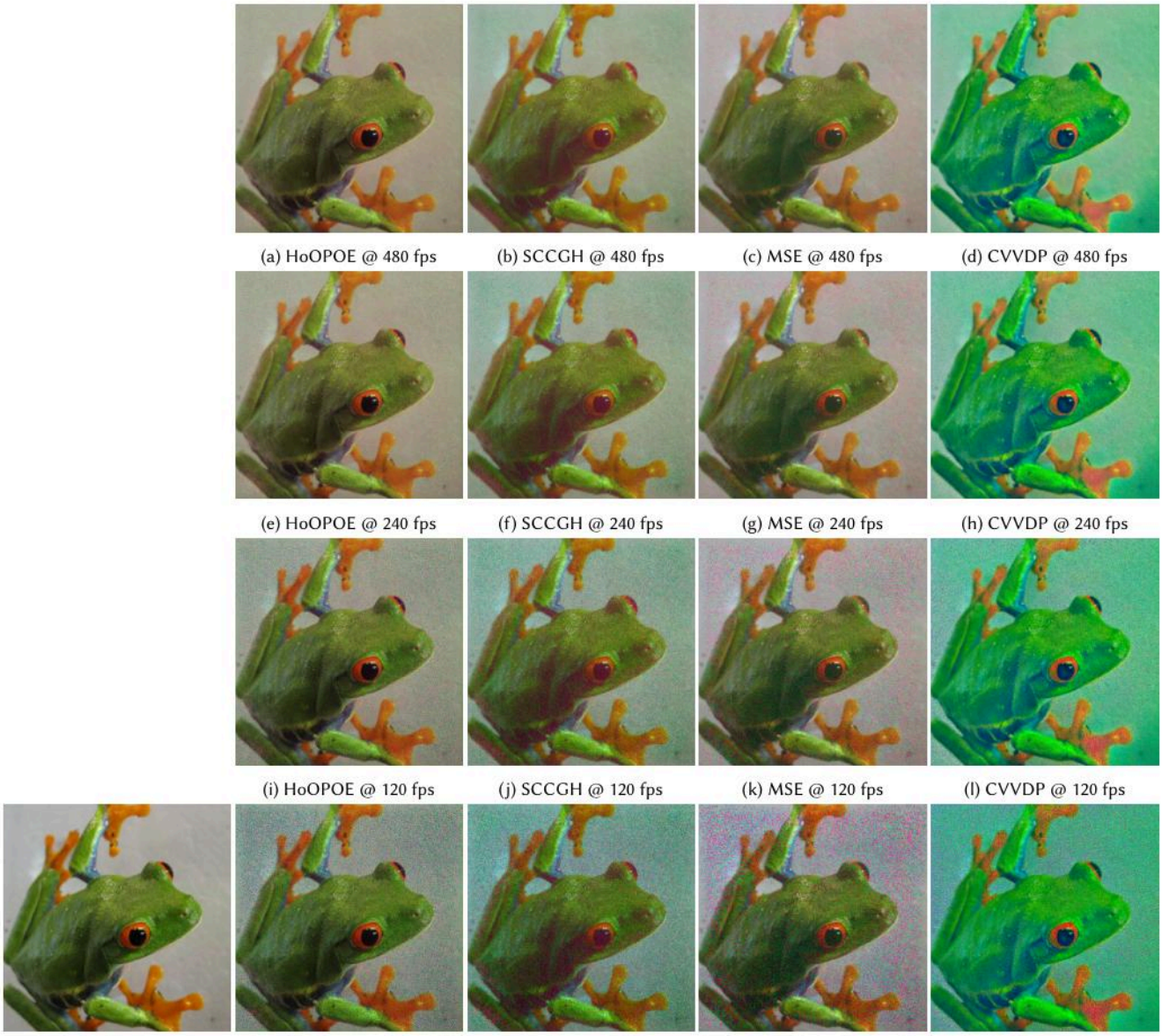
(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S31. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 3, for the near viewing plane



(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S32. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 4, for the near viewing plane

S10.2 Far viewing plane (4.5 cm)



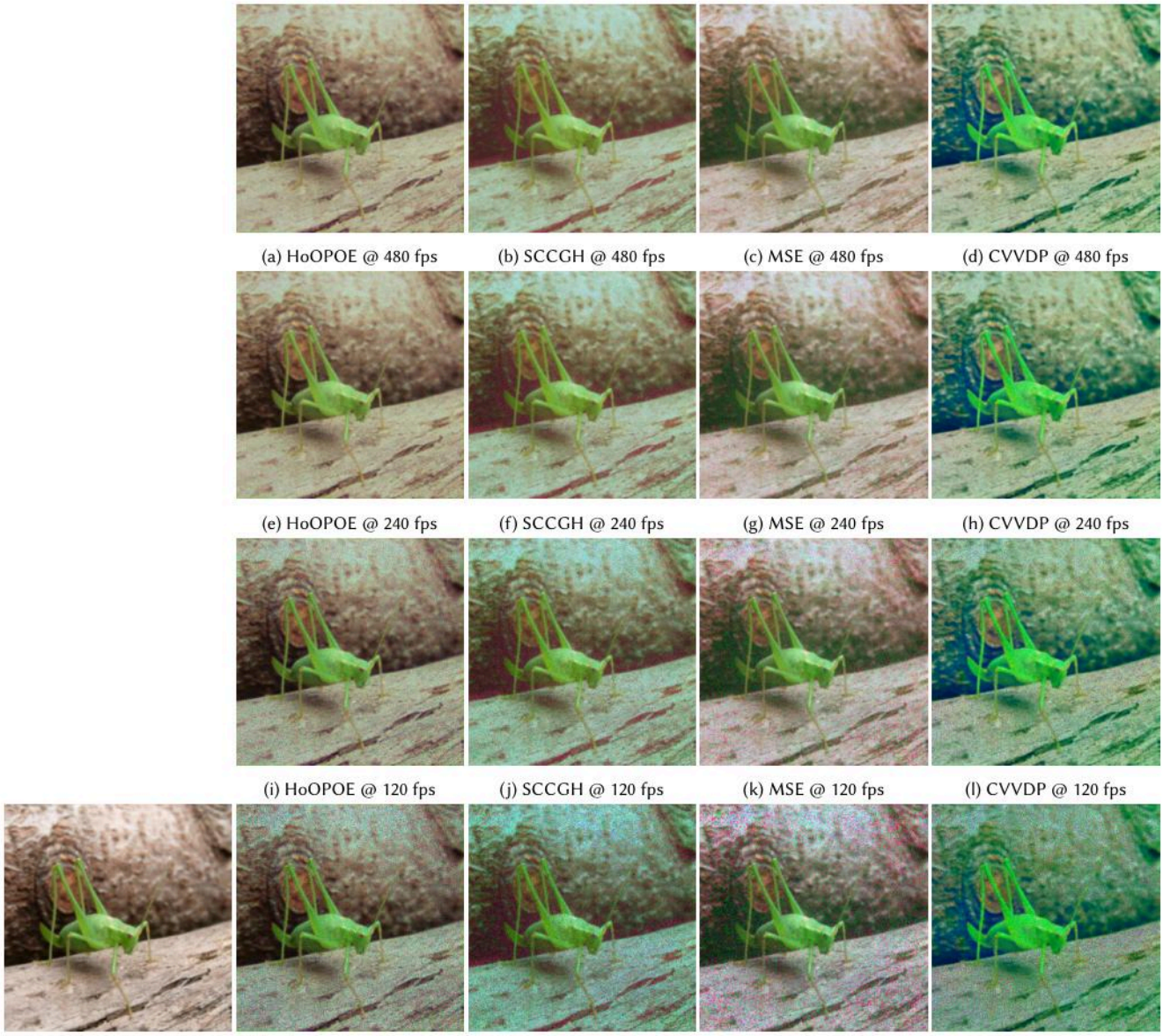
(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S33. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 0, for the far viewing plane. Original public domain image taken by AndrewH [AndrewH 2002].



(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S34. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 1, for the far viewing plane



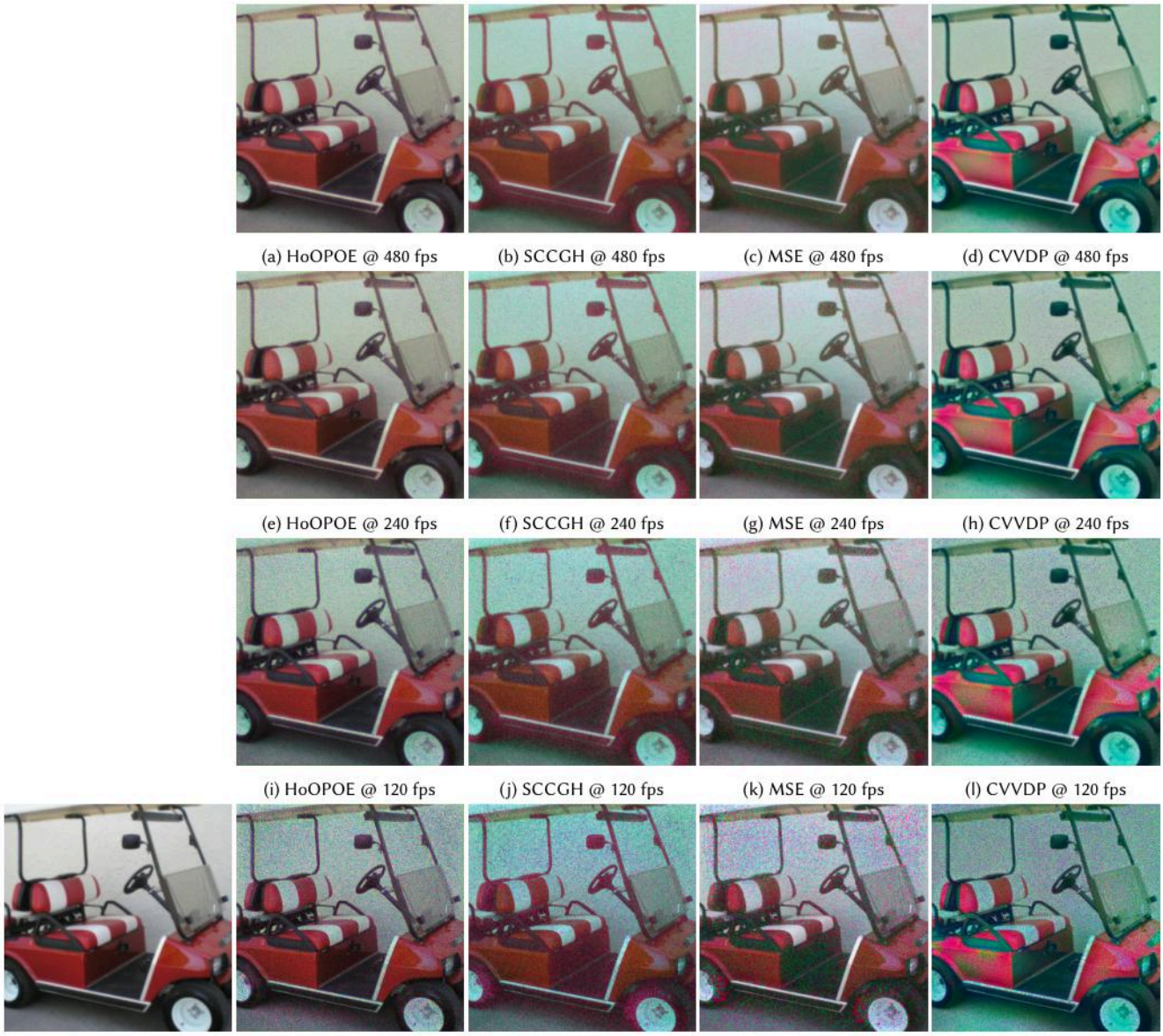
(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S35. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 2, for the far viewing plane



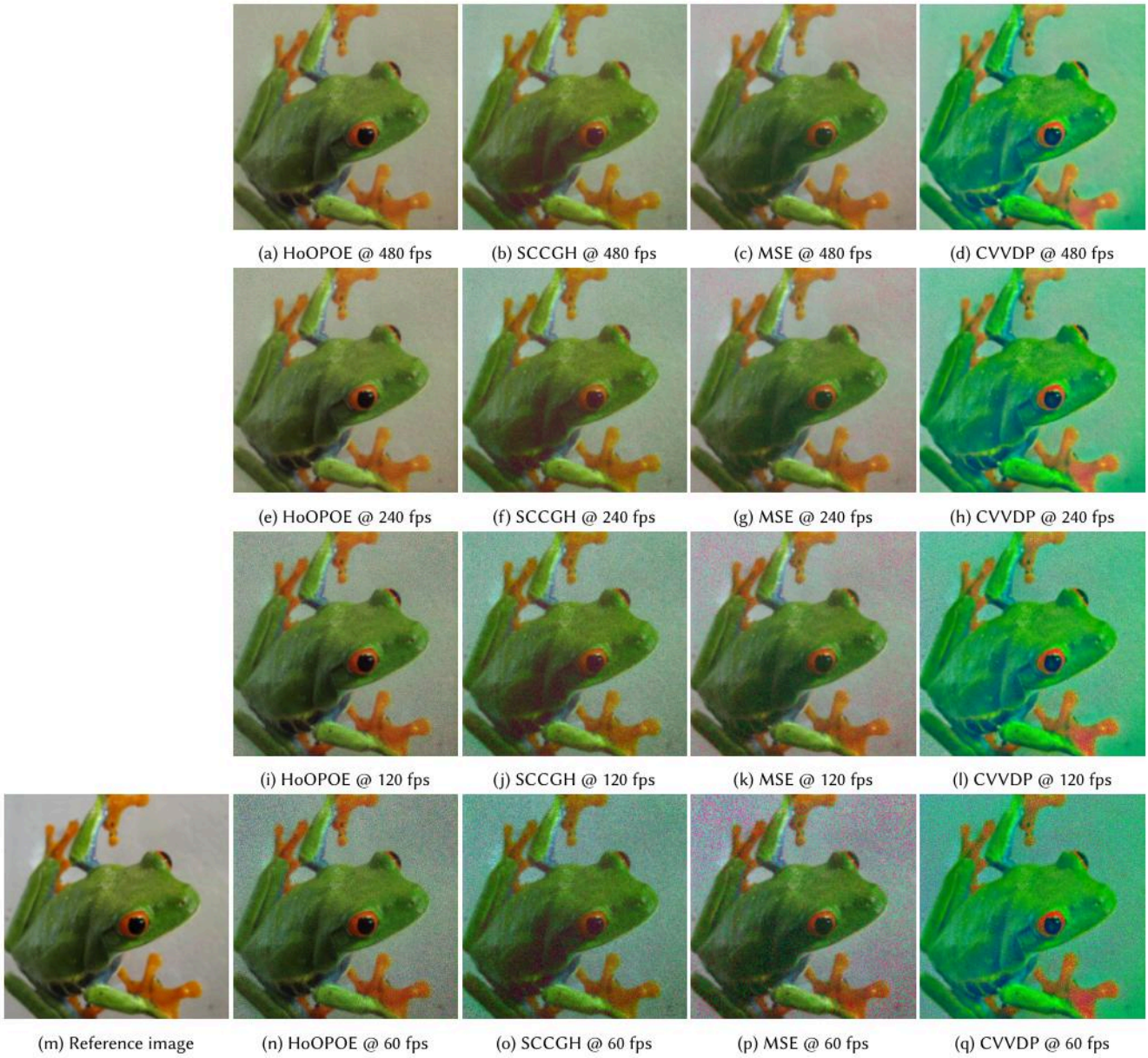
(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

Fig. S36. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 3, for the far viewing plane



(a) HoOPOE @ 480 fps (b) SCCGH @ 480 fps (c) MSE @ 480 fps (d) CVVDP @ 480 fps

(e) HoOPOE @ 240 fps (f) SCCGH @ 240 fps (g) MSE @ 240 fps (h) CVVDP @ 240 fps

(i) HoOPOE @ 120 fps (j) SCCGH @ 120 fps (k) MSE @ 120 fps (l) CVVDP @ 120 fps

(m) Reference image (n) HoOPOE @ 60 fps (o) SCCGH @ 60 fps (p) MSE @ 60 fps (q) CVVDP @ 60 fps

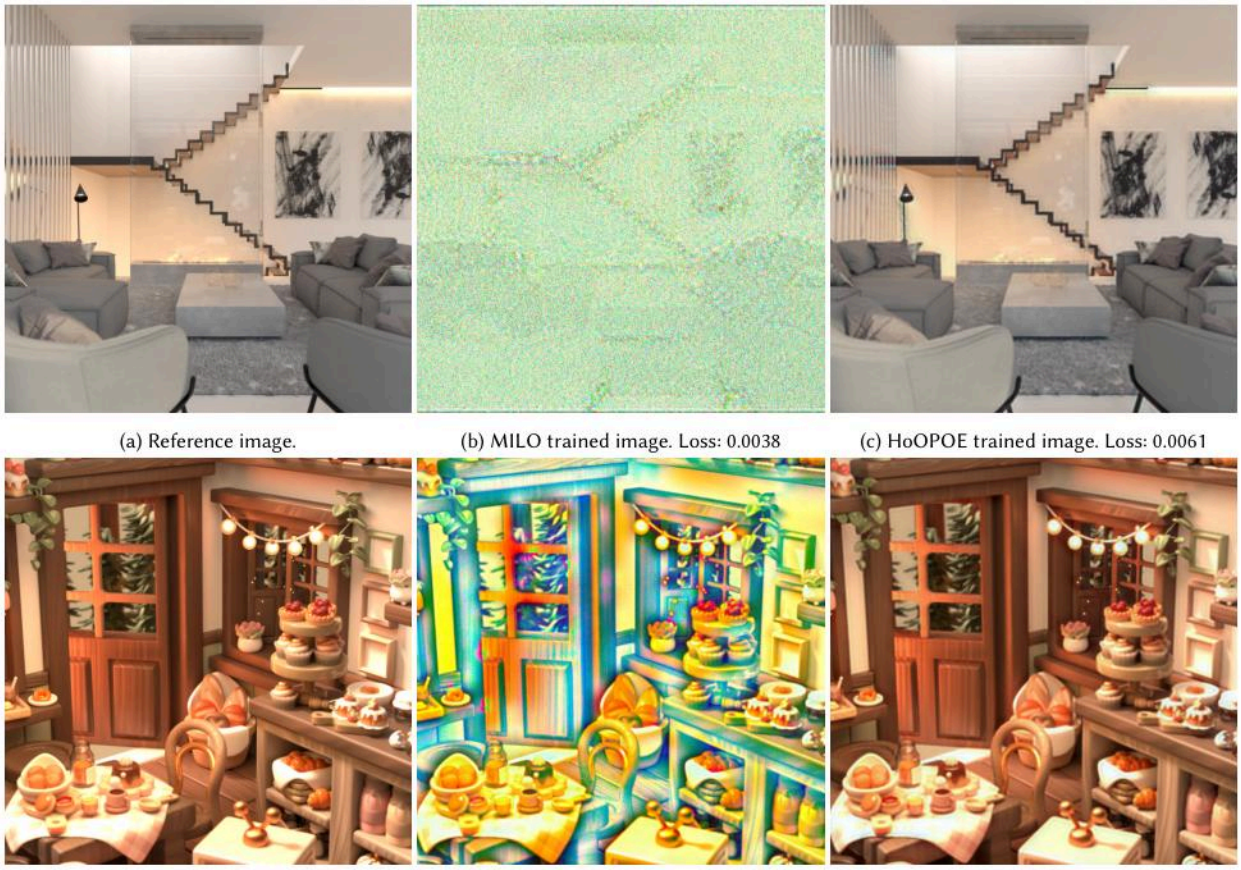
Fig. S37. Holograms generated by HoOPOE, SCCGH, CVVDP, and MSE simulated at variable frame rates for stimulus 4, for the far viewing plane

S11 Analyzing Adversarial Stimuli and Optimization Landscapes

This section presents more examples of adversarial stimuli and analyzes the optimization landscape of each loss function.

S11.1 More Examples of Adversarial Stimuli

Here we show more examples of adversarial stimuli that challenge perceptual metrics like CVVDP and MILO as loss functions for CGH. MILO and CVVDP perform well on visual quality datasets [Cogalan et al. 2025; Mantiuk et al. 2024b]. However, when used as an optimization target, these metrics discover adversarial stimuli that minimize their loss value but look visually objectionable. In the examples below, MILO and CVVDP recognize their stimuli as being superior, despite being visually degraded.



(a) Reference image. (b) MILO trained image. Loss: 0.0038 (c) HoOPOE trained image. Loss: 0.0061

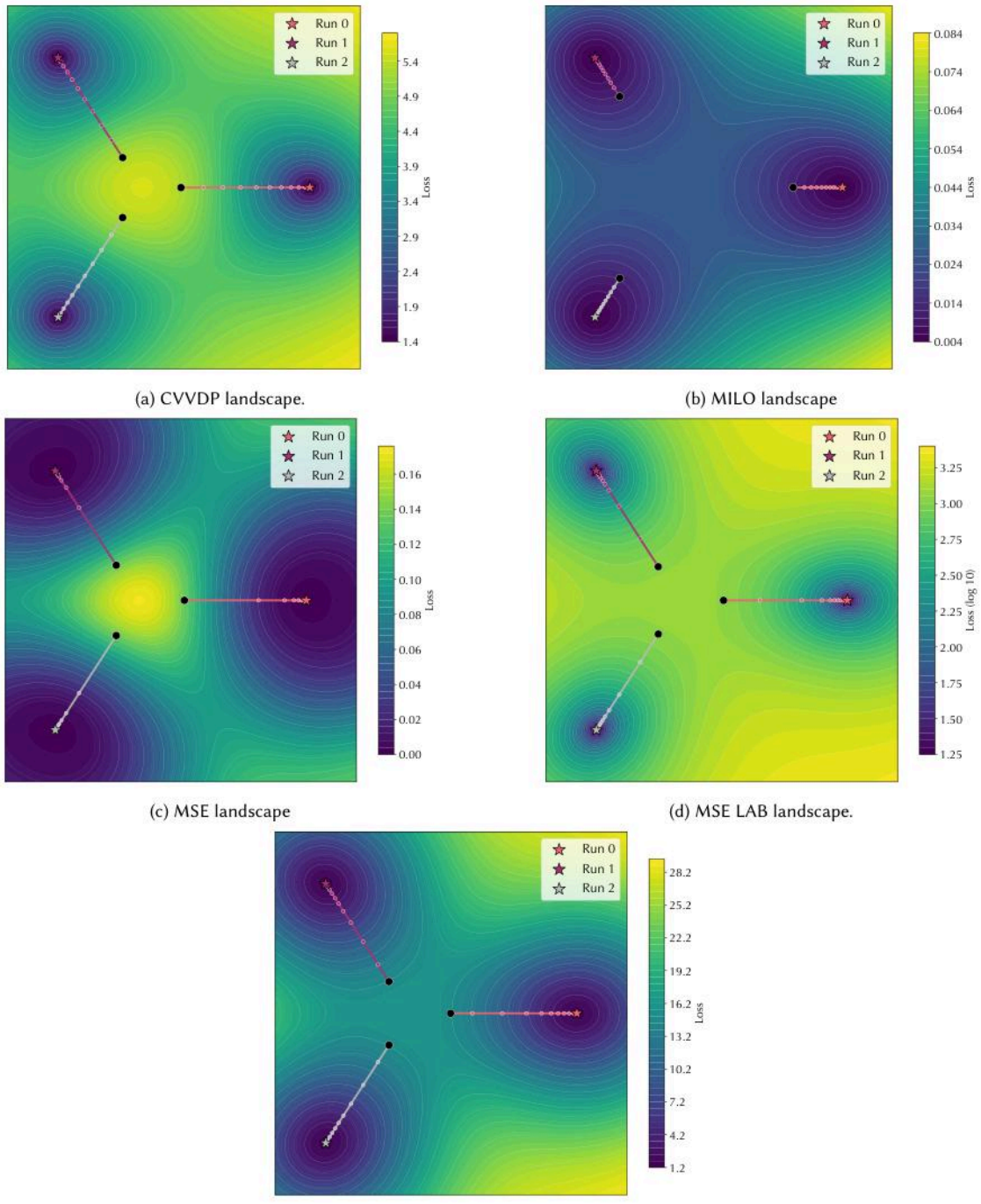
(d) Reference image. (e) CVVDP trained image. JOD: 9.03 (f) HoOPOE trained image. JOD: 7.44

Fig. S38. Adversarial examples of CVVDP and MILO. Copyright for the original 3D models goes to Lynxsdesign [2024] (licensed under CC-BY) and Morena [2023] (licensed under CC-BY-SA).

S11.2 Optimization Landscape Analysis

We also analyze the optimization landscape of various quality metrics/loss functions. Since the optimization space is extremely high-dimensional (i.e., the number of SLM pixels × the number of discrete phases an SLM pixel can take), we visualize the landscape using the following approach. For each loss function, we randomly initialize three holograms, and then train a corresponding image using each initialization. This gives us three end points, which uniquely define a plane in the high-dimensional optimization space. It is on this plane we visualize the landscape.

Fig. S39 visualizes the reduced-dimensional optimization landscapes of CVVDP, MILO, MSE, MSE LAB, and HoOPOE, where the colorbars indicate the magnitude of each loss. Unsurprisingly, all landscapes are highly non-convex (with respect to the SLM pixel space): depending on the initialization, the optimization may converge to different basins (i.e., local minima). However, for each loss function, these basins are effectively "equally good," in the sense that they attain nearly identical loss values. Consequently, if the quality assessment provided by a loss



(a) CVVDP landscape.

(b) MILO landscape

(c) MSE landscape

(d) MSE LAB landscape.

(e) HoOPOE landscape

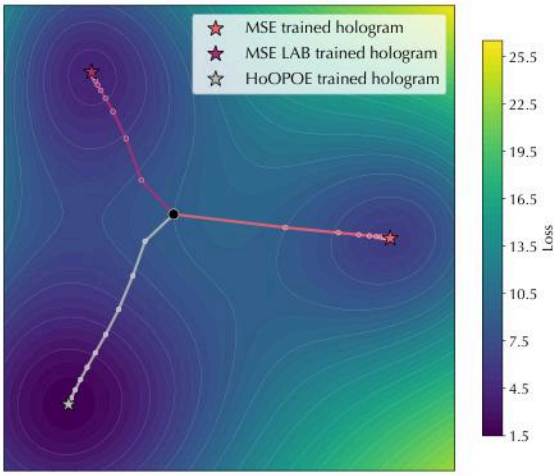
Fig. S39. Optimization landscape of each loss function with respect to the SLM pixel space. Each loss is trained with three random initializations. We superimpose on the landscape the optimization trajectory under each initialization.

function is reliable, convergence to a particular local minimum should not significantly affect the final perceptual quality, as the resulting images ought to be of comparable quality.

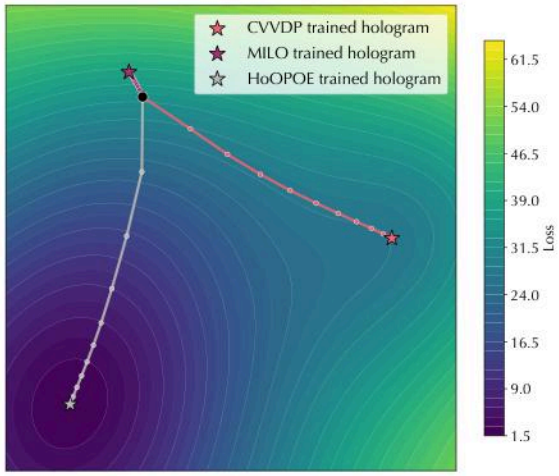
The results here are perhaps not all that surprising in light of prior work on learning theory [Allen-Zhu et al. 2019; Du et al. 2019]. When optimizing an over-parameterized forward model using gradient descent—as is the case here given the large number of free parameters in the SLM pixel space—the optimization landscape in the neighborhood of a random initialization is often effectively smooth and locally close to convex. Critically, under such conditions, local minima tend to achieve objective values that are nearly as good as the global minimum, up to a small error bound. Therefore, the ineffectiveness of existing loss functions for CGH is likely attributed *not* to their non-convexity, but rather to their misjudgment of CGH content's quality.

To further illustrate the differences in quality assessment among loss functions, we compare holograms optimized under different losses using the *same* HoOPOE landscape. In Fig. S40a, we initialize three holograms identically and optimize them separately using HoOPOE, MSE, and MSE LAB, respectively. We then visualize their optimization trajectories within the HoOPOE landscape, where the dimensionality reduction is performed in the same manner as before. Fig. S40b shows a similar experiment, except that the three holograms are optimized using HoOPOE, CVVDP, and MILO, respectively.

In both figures, the holograms optimized using HoOPOE converge to the deepest basins, which are, naturally, assigned the highest quality by HoOPOE itself. Holograms trained under other loss functions end up at shallower points in the HoOPOE landscape, indicating lower quality as quantified by HoOPOE. Notably, holograms trained by MSE and MSE LAB are judged as having higher quality by HoOPOE than those trained by CVVDP and MILO, confirming the subjective results before (Supplementary Material S3).



(a) Training trajectories of MSE, MSE LAB, and HoOPOE from the same initialization, superimposed over the HoOPOE landscape.



(b) Training trajectories of CVVDP, MILO, and HoOPOE from the same initialization, superimposed over the HoOPOE landscape.

Fig. S40. Holograms trained under various loss functions plotted in the HoOPOE landscape.

S12 Hardware Study

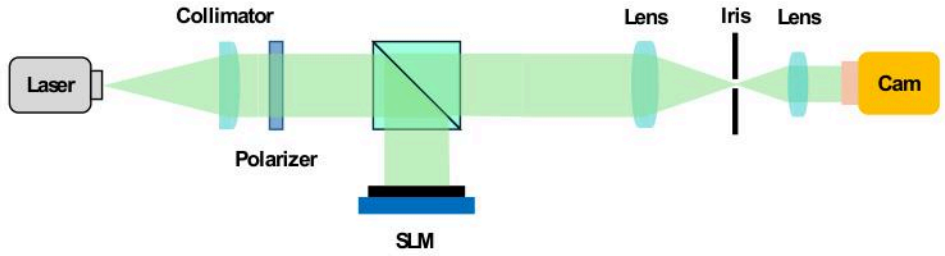


Fig. S41. Holographic display prototype illustration

We use a benchtop holographic display prototype consisting of a fiber-coupled laser module, a collimating lens, a linear polarizer, a phase-only spatial light modulator, a 4f relay, and a color camera, as illustrated in Fig. S41. The illumination source is a FISBA ReadyBeam fiber-coupled module with three optically aligned laser diodes; the measured wavelengths are 638, 520, and 450 nm. The modulator is a HOLOEYE PLUTO LCoS panel with a resolution of 1920×1080, a pixel pitch of 8.0 μm, and a bit depth of 8 bits per pixel. The field reflected from the SLM is relayed through a 4f system (f_1 = 150 mm, f_2 = 75 mm). Color images are captured with a ZWO ASI533MC Pro CMOS sensor.

We adopt the camera-calibrated wave propagation model from [Choi et al. 2022], which wraps a band-limited angular-spectrum propagator between two convolutional refinement networks and carries four learnable physical correction terms — amplitude and phase residuals at the SLM and Fourier planes — together with a learned look-up table for phase quantization. To keep the forward model fully interpretable and to enable a fair comparison across loss functions during phase optimization, we drop the two refinement networks and retain only the SLM- and Fourier-plane corrections together with the look-up table, which jointly compensate for hardware imperfections. The model is trained per channel on 1200 phase–intensity pairs captured on the prototype, with the training phases generated under four loss functions (MSE, MSE LAB, HoOPOE, and SCCGH).

Additional images of our camera captures can be found in Fig. S42, Fig. S43, and Fig. S44.

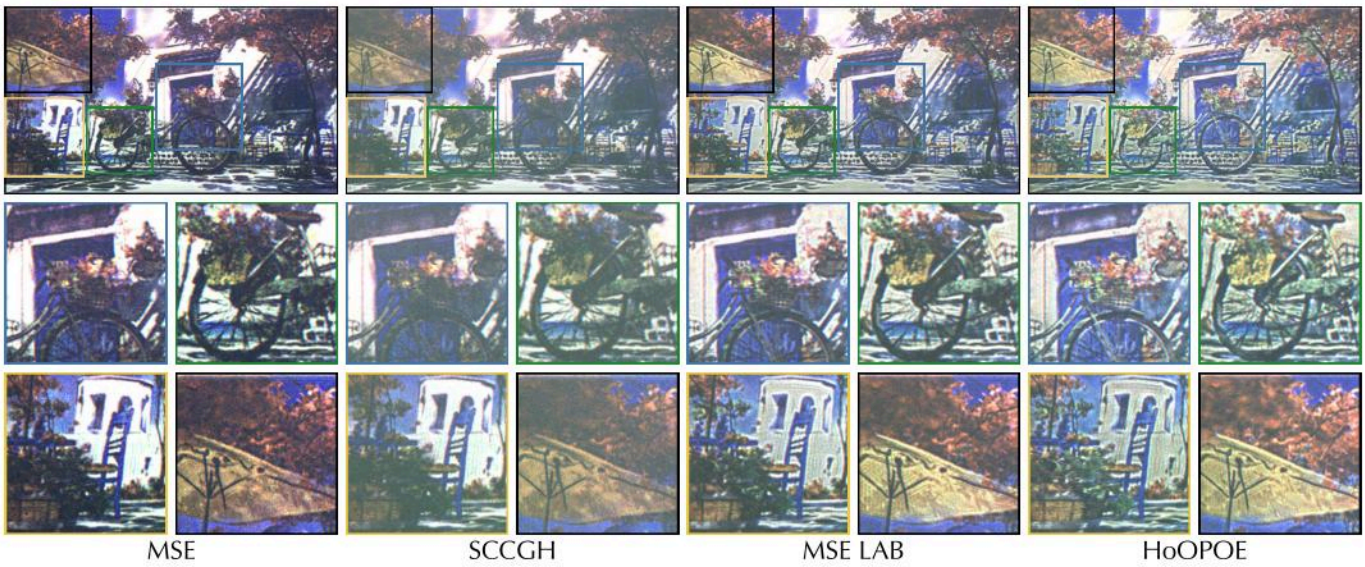


Fig. S42. Real-life camera captures for MSE, MSE LAB, SCCGH, and HoOPOE for stimulus 0. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

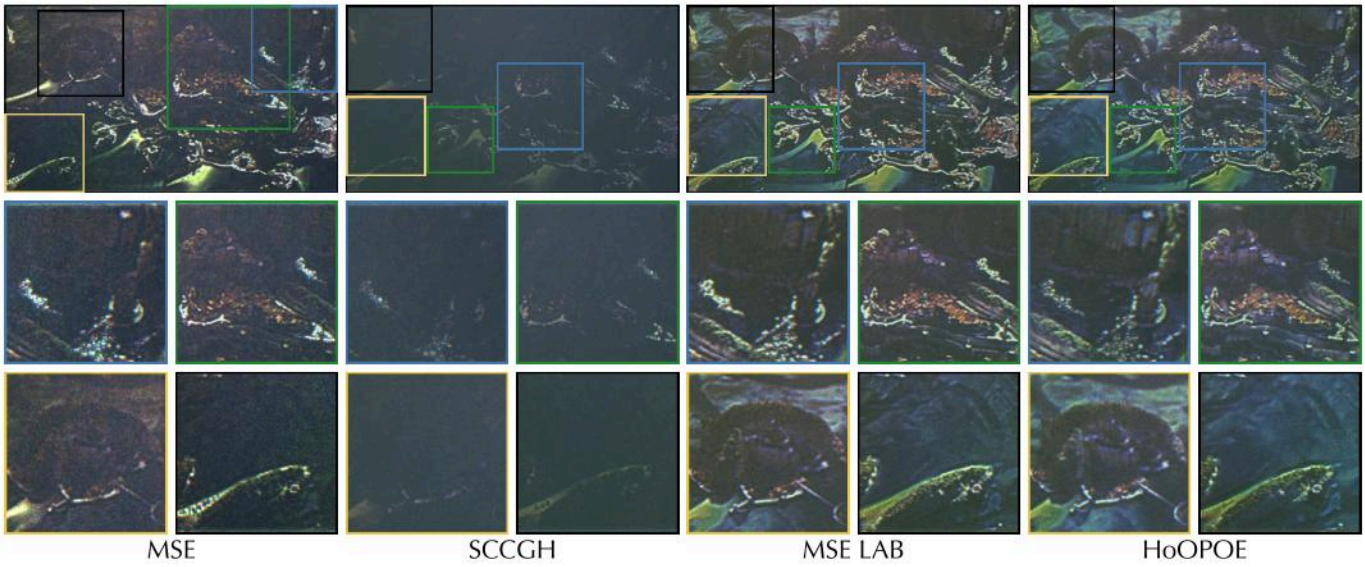


Fig. S43. Real-life camera captures for MSE, MSE LAB, SCCGH, and HoOPOE for stimulus 1. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].

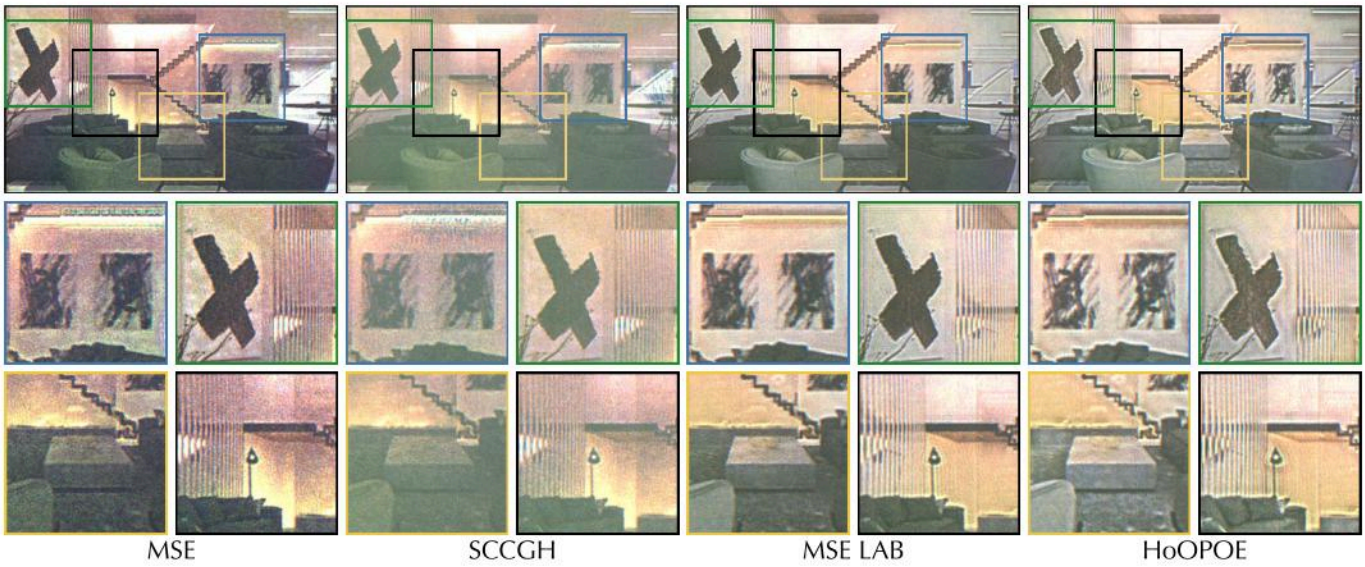


Fig. S44. Real-life camera captures for MSE, MSE LAB, SCCGH, and HoOPOE for stimulus 3. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].

S13 Generalizability of HoOPOE to Different Hardware Configurations and Viewing Conditions

In order to demonstrate the generalizability of HoOPOE, we also trained HoOPOE weights on two different SLM display setups and propagation distances. The details of the two simulated SLM displays are listed below:
SLM 1:

- Pitch size: 4.5 μm
- Propagation distances: 0.25 to 1.25 cm
- SLM resolution: 1920×1080

SLM 2:

- Pitch size: 12.5 μm
- Propagation distances: 2 cm to 10 cm
- SLM resolution: 1272×1024

The below holograms were trained on the same focal stack consisting of 5 images. Only 1 frame is generated here. No temporal multiplexing is applied in this section. No ROI cropping was utilized. HoOPOE remains superior to other loss functions, even under these different conditions.

Results for SLM 1 can be found in Fig. S45, Fig. S46, Fig. S47, Fig. S48, Fig. S49. Results for SLM 2 can be found in Fig. S50, Fig. S51, Fig. S52, Fig. S53, Fig. S54.



Fig. S45. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 0. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

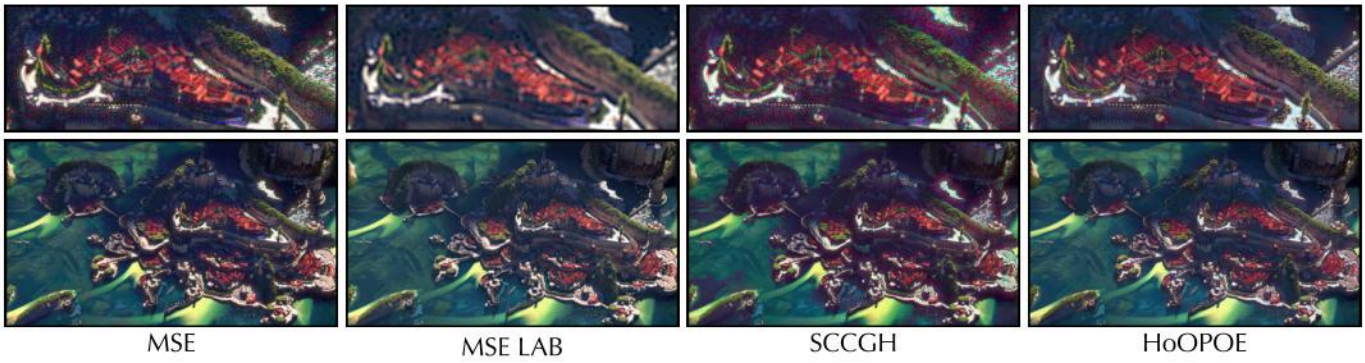


Fig. S46. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 1. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].

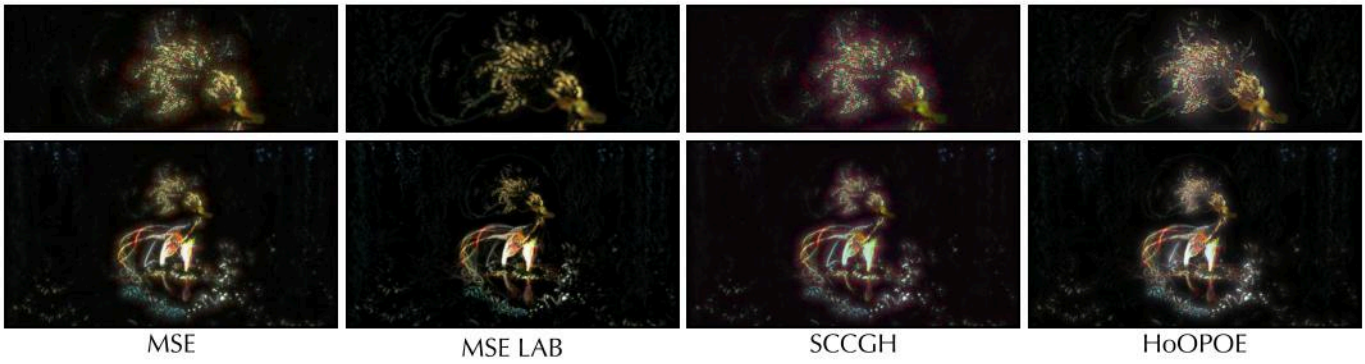


Fig. S47. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 2. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Aiello [2022].

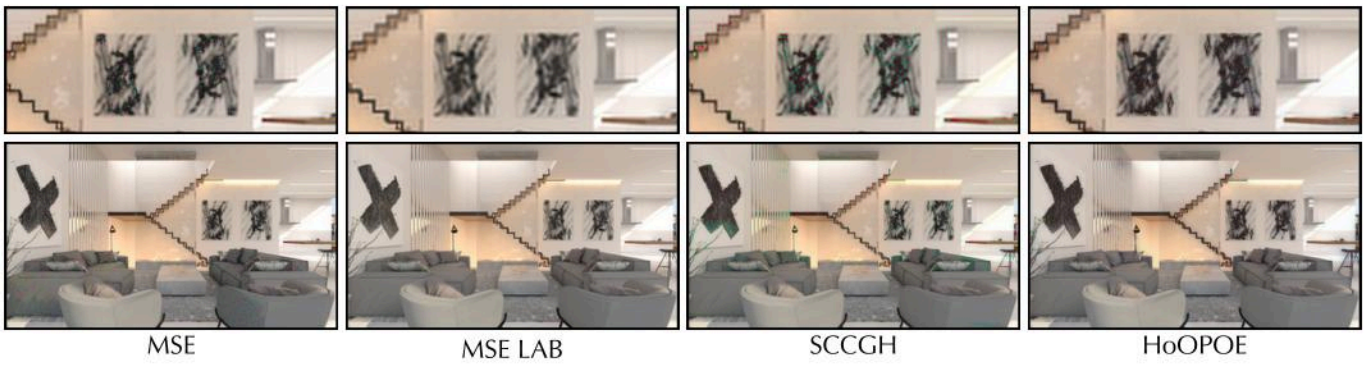


Fig. S48. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 3. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].



Fig. S49. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 4. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Morena [2023].

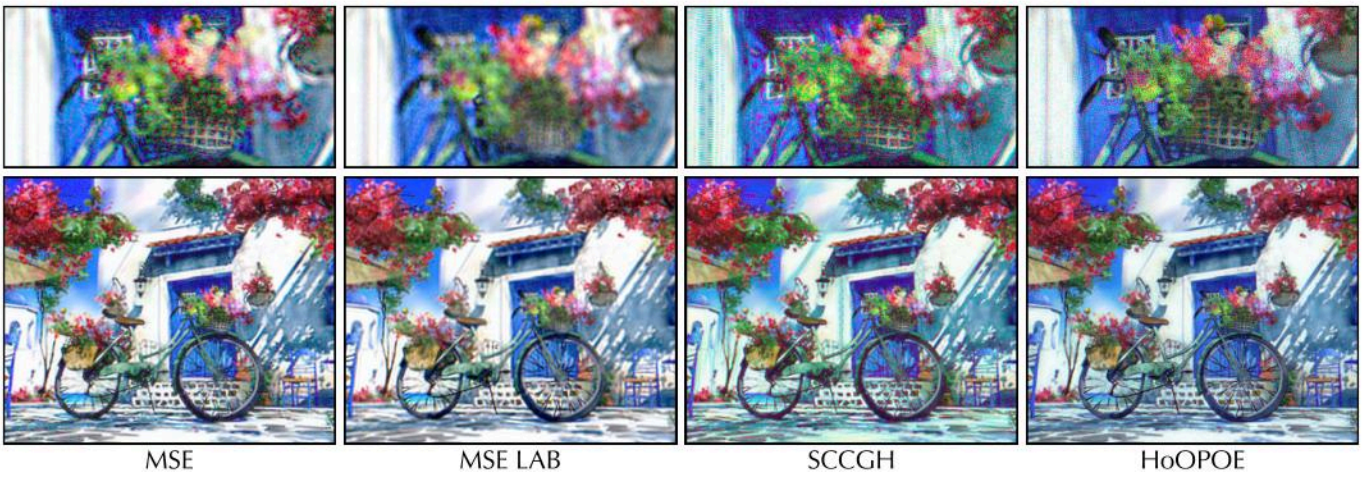


Fig. S50. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 0. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

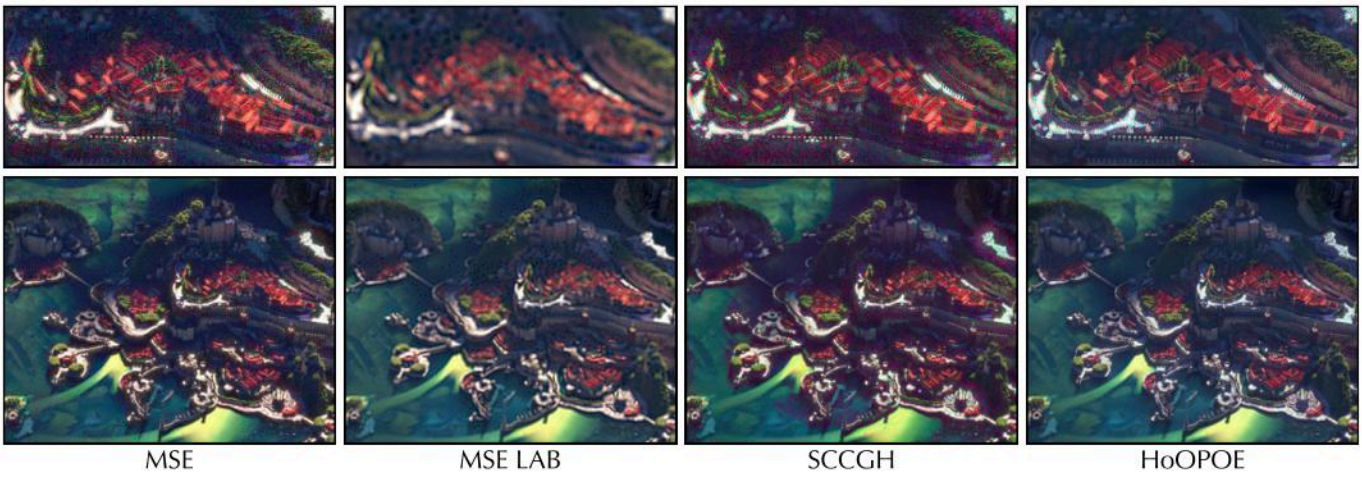


Fig. S51. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 1. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].

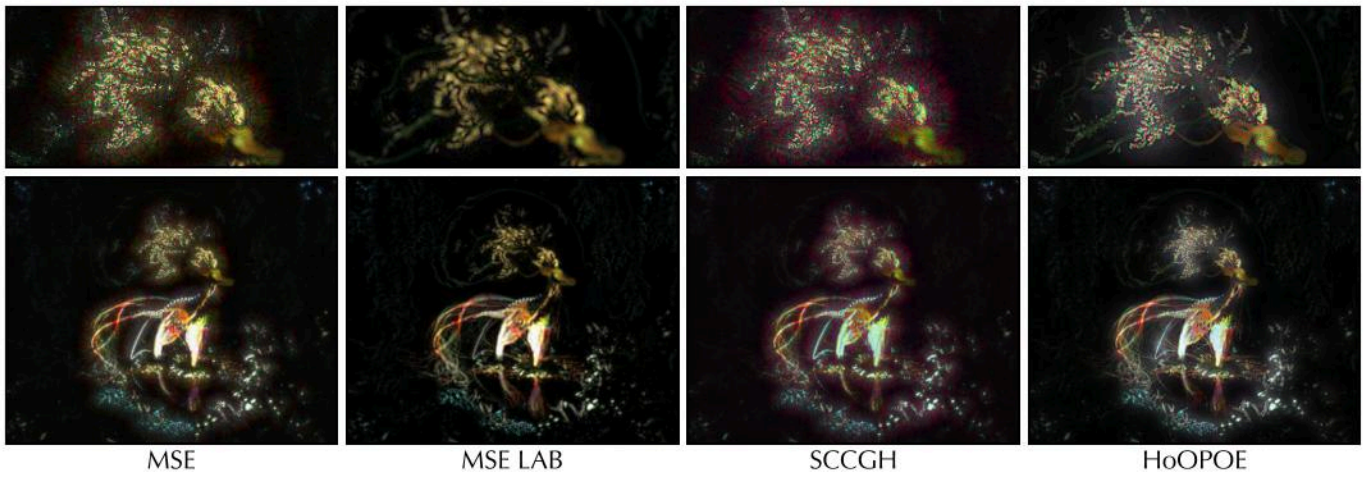


Fig. S52. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 2. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Aiello [2022].

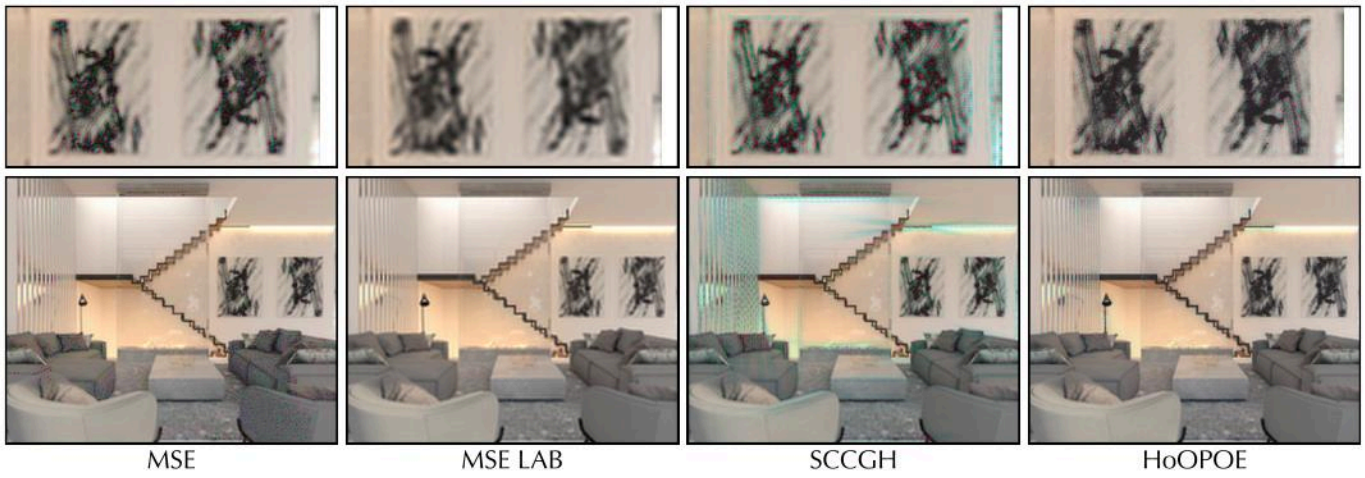


Fig. S53. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 3. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].



Fig. S54. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 4. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Morena [2023].

S14 Generalizability of HoOPOE Weights to Different Hardware Configurations and Viewing Conditions

In order to demonstrate that HoOPOE weights remain generalizable, even across different hardware setups, we additionally re-generate holographic images using the SLM 1 and SLM 2 hardware setups described in Supplementary Material S13 *without* re-tuning HoOPOE weights. Visual stimuli for SLM 1 can be found in Fig. S59, Fig. S60, Fig. S61, Fig. S62, and Fig. S63. Stimuli for SLM 2 can be found in Fig. S64, Fig. S65, Fig. S66, Fig. S67, and Fig. S68.

We conduct a user study on the stimuli for the two alternative SLM setups, using the same methodology described in Sec. 4.2 of the paper. JOD values for SLM 1 are in Fig. S55. JOD values for SLM 2 are in Fig. S56. The significance table for SLM 1 is plotted in Fig. S57, and SLM 2's in Fig. S58. SCCGH was excluded from this round of tests due to time constraints and lackluster visual results. Users still (statistically significantly) prefer HoOPOE-generated stimuli over baseline, even without the re-tuning of HoOPOE weights.

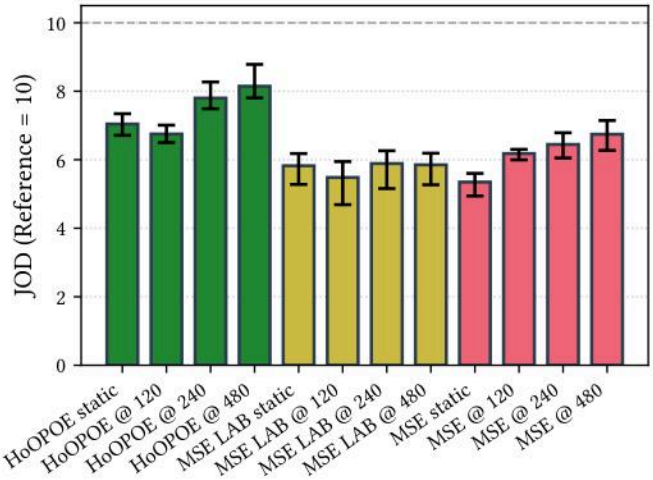


Fig. S55. User study derived JOD values for stimuli generated using SLM 1 (parameters described in Supplementary Material S13). HoOPOE, MSE, and MSE LAB benchmarked. Three frame rates (120 fps, 240 fps, 480 fps) tested, alongside static (non-temporally multiplexed) images. N = 18.

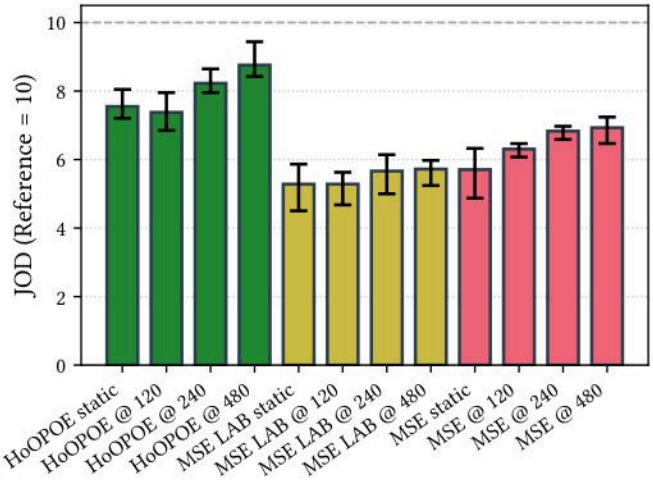


Fig. S56. User study derived JOD values for stimuli generated using SLM 2 (parameters described in Supplementary Material S13). HoOPOE, MSE, and MSE LAB benchmarked. Three frame rates (120 fps, 240 fps, 480 fps) tested, alongside static (non-temporally multiplexed) images. N = 18.

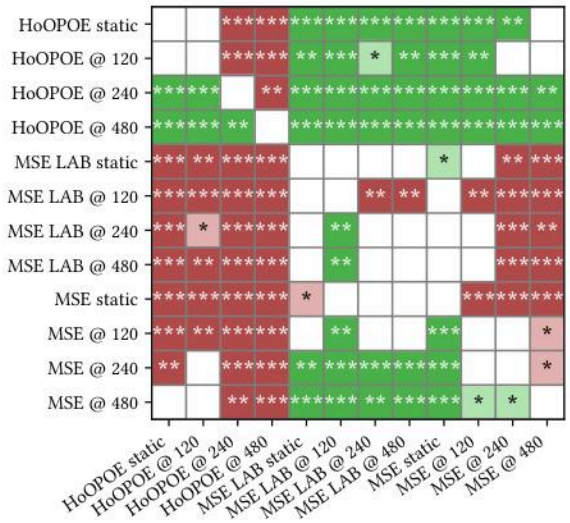


Fig. S57. 2D grid showing the statistical significance of each observed difference in JOD in Fig. S55 (SLM 1). Red cells indicate that the benchmark on the Y axis performed worse than the corresponding benchmark on the X axis. Green indicates the opposite. A single star indicates $p < 0.05$. Two stars indicate $p < 0.01$ and three stars indicate $p < 0.001$.

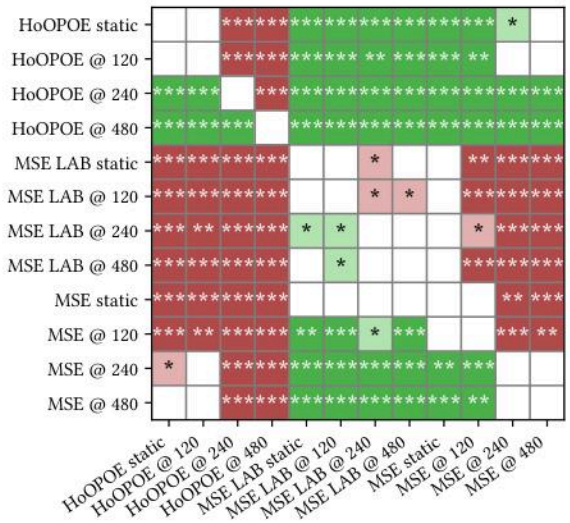


Fig. S58. 2D grid showing the statistical significance of each observed difference in JOD in Fig. S56 (SLM 2). Red cells indicate that the benchmark on the Y axis performed worse than the corresponding benchmark on the X axis. Green indicates the opposite. A single star indicates $p < 0.05$. Two stars indicate $p < 0.01$ and three stars indicate $p < 0.001$.



Fig. S59. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 0, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

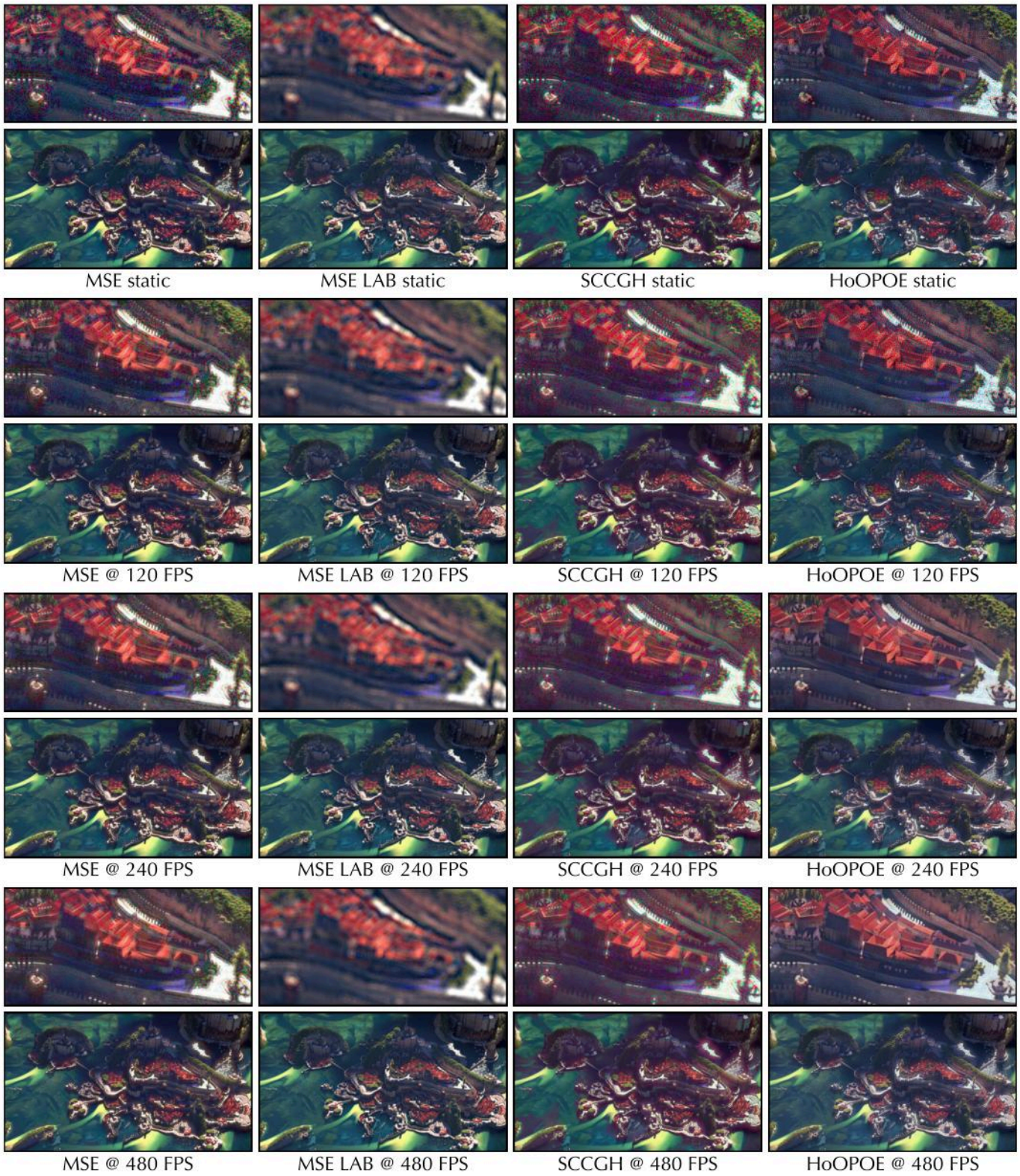


Fig. S60. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 1, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].

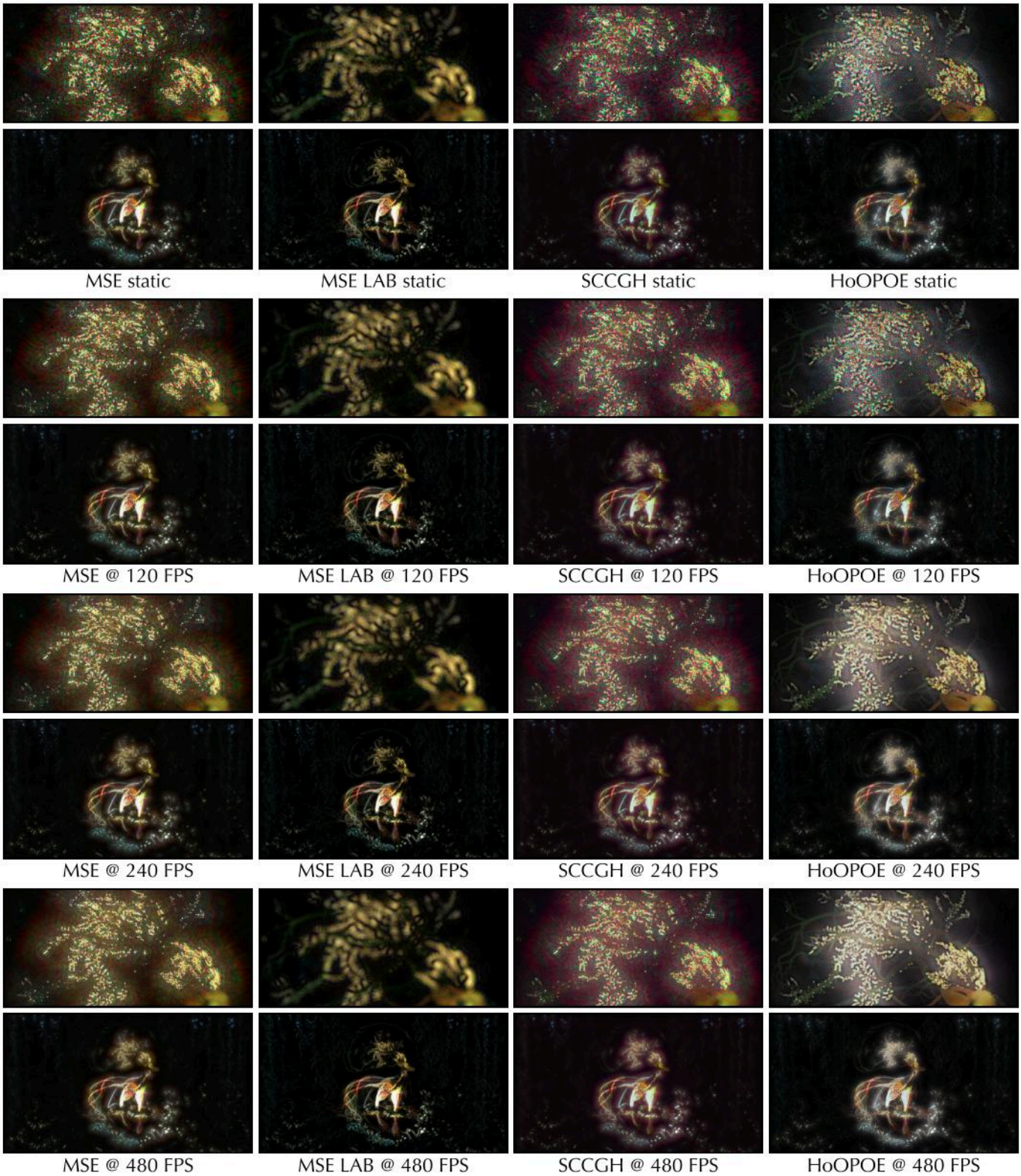


Fig. S61. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 2, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Aiello [2022].

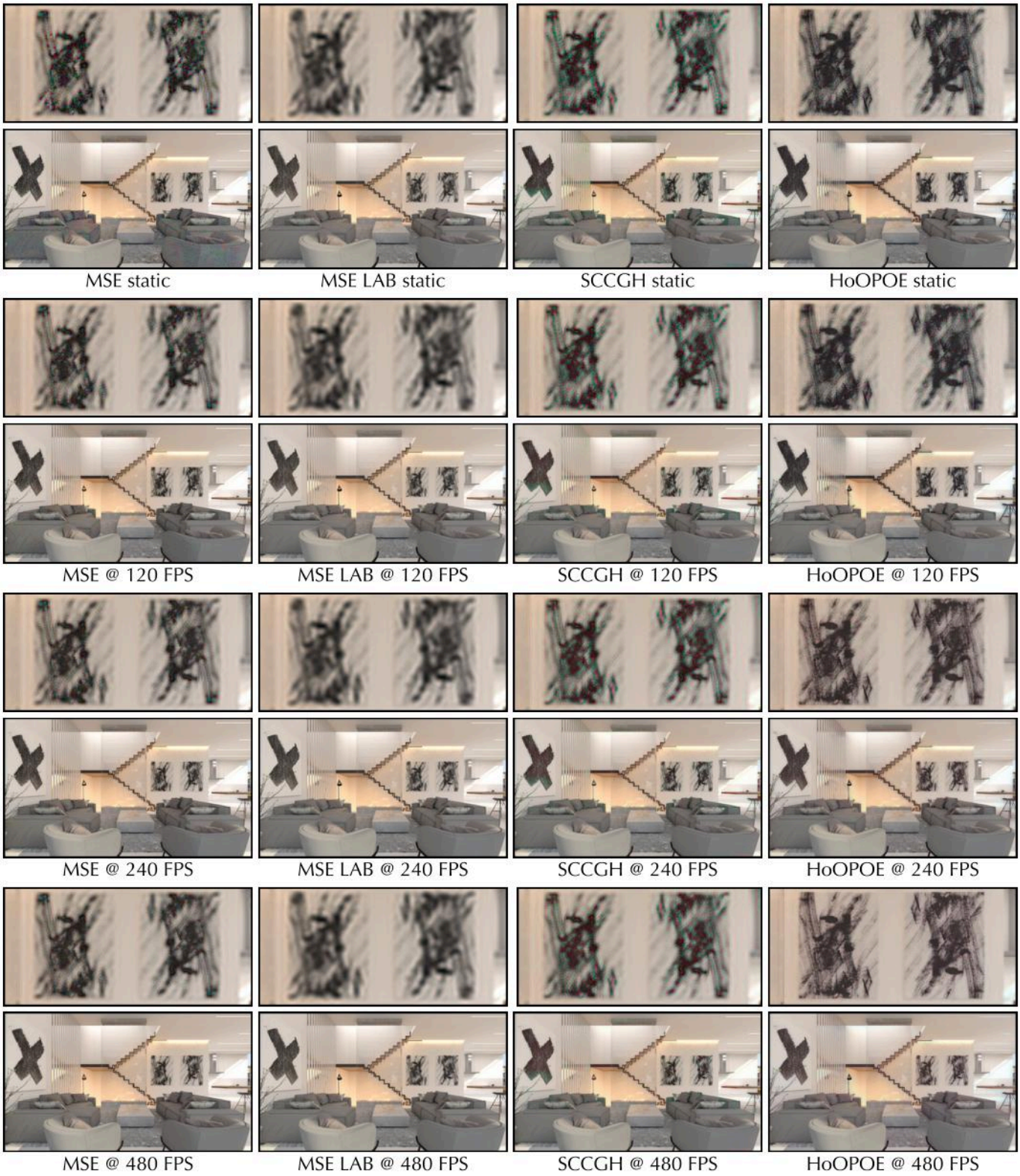


Fig. S62. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 3, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].

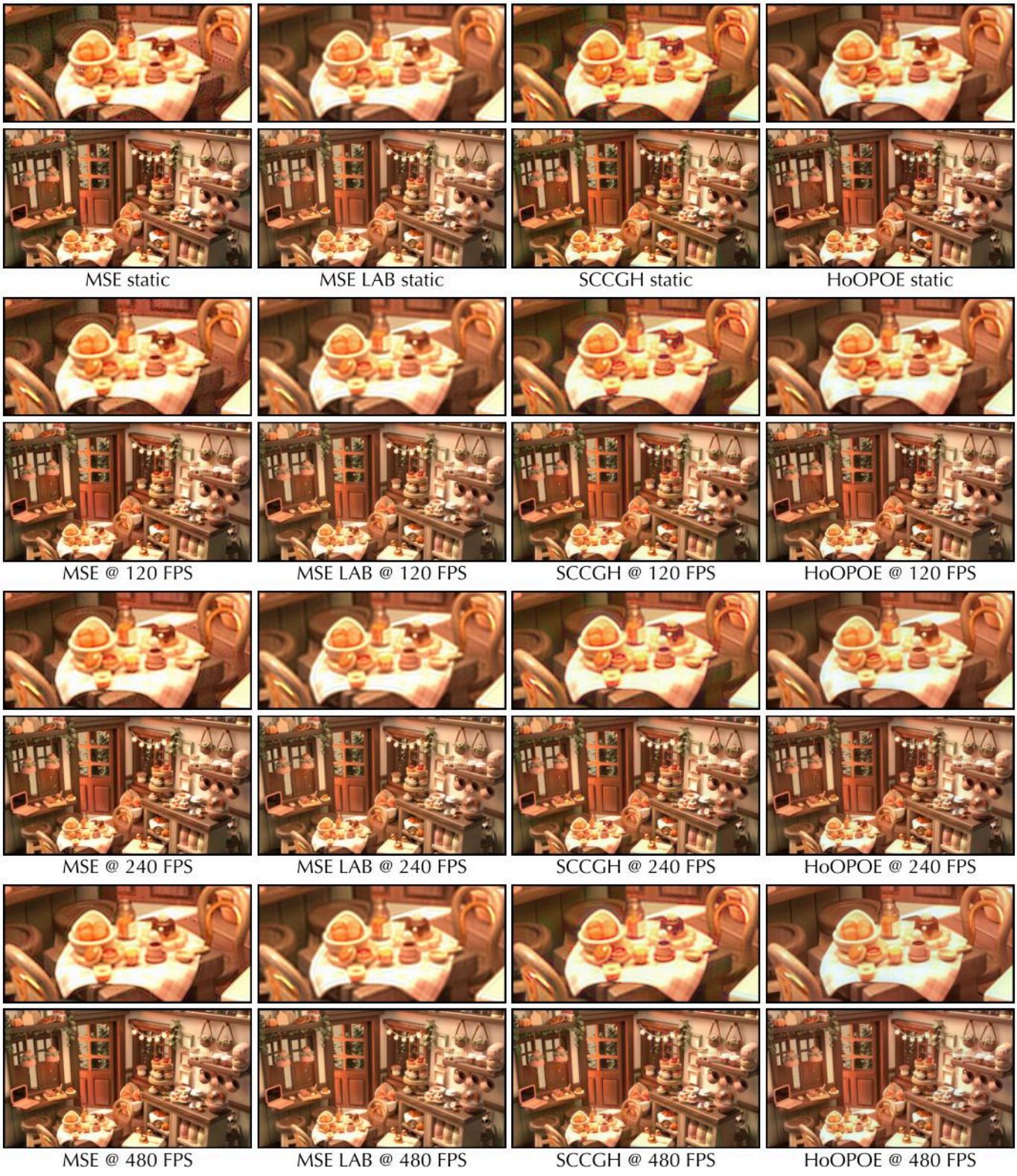


Fig. S63. Comparison of different loss functions for a 5 plane focal stack with pitch size of 4.5 μm and propagation distances of 0.25 to 1.25 cm for stimulus 4, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Morena [2023].

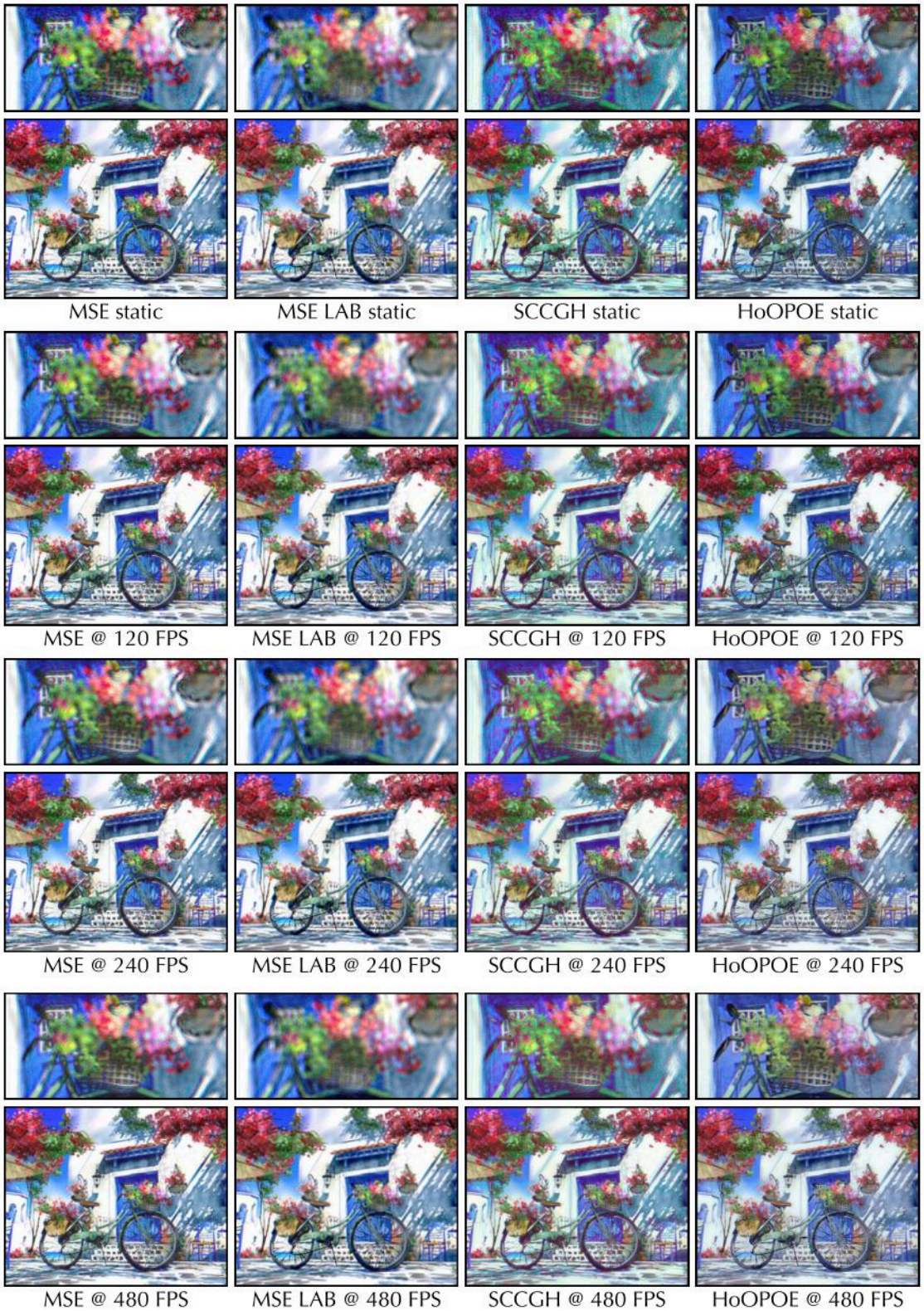


Fig. S64. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 0, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Tada [2023].

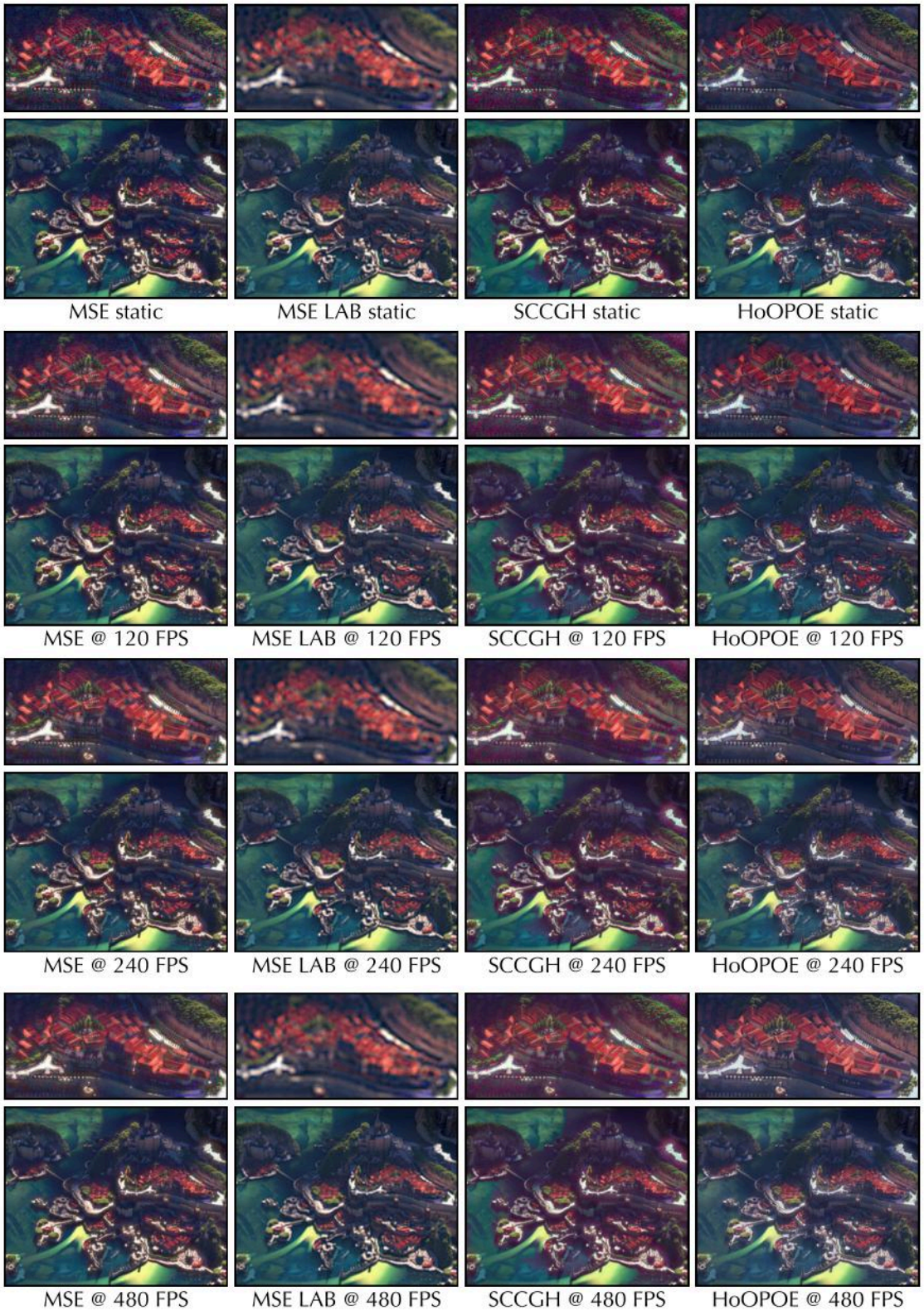


Fig. S65. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 1, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].

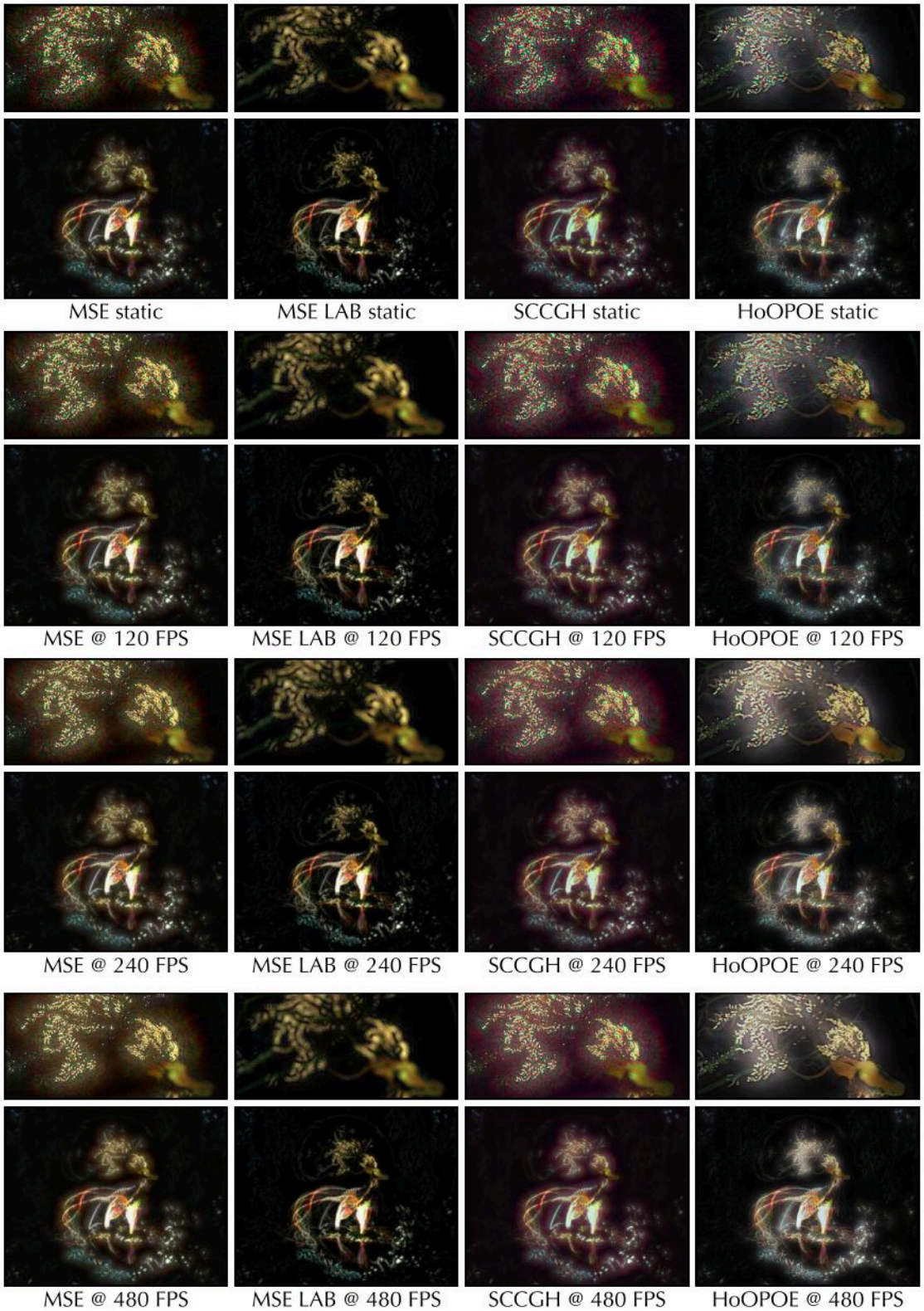


Fig. S66. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 2, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Aiello [2022].



Fig. S67. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 3, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY) goes to Lynxsdesign [2024].



Fig. S68. Comparison of different loss functions for a 5 plane focal stack with pitch size of 12.5 μm and propagation distances of 2 to 10 cm for stimulus 4, at various frame rates. Note that the topmost row (the "static" row) represents stimuli that are not temporally multiplexed (animated) at all. Note that the effect of temporal multiplexing is approximated here by averaging 2/4/8 frames for the 120/240/480 FPS stimuli. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Morena [2023].

S15 Benchmarking HoOPOE without Temporal Multiplexing

In order to measure how HoOPOE performs against the competition in the *absence* of any temporal multiplexing, we additionally run the user study described in Sec. 4. However, in this iteration of the experiment, we display only a single static frame on the monitor. Only image data is presented to the participant. The JOD values obtained are plotted in Fig. S69, and the corresponding statistical significances of these JOD differences are tabulated in Fig. S70.

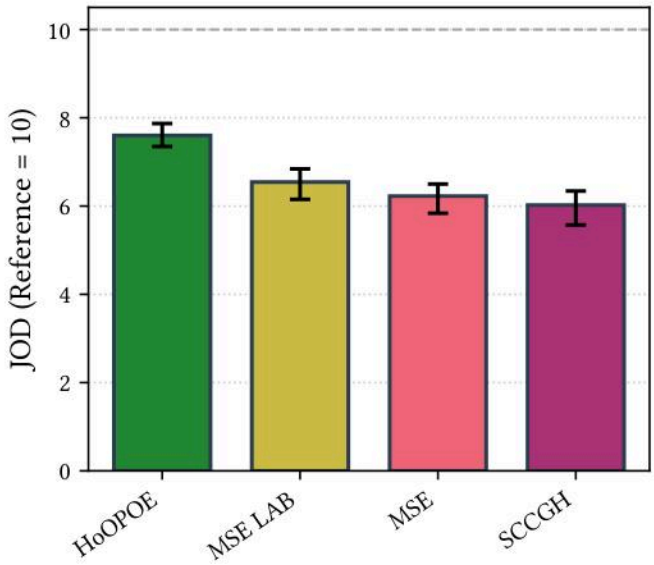


Fig. S69. User study derived JOD values for static (non-temporally multiplexed) stimuli. HoOPOE, MSE, MSE LAB, and SCCGH benchmarked. N = 18.

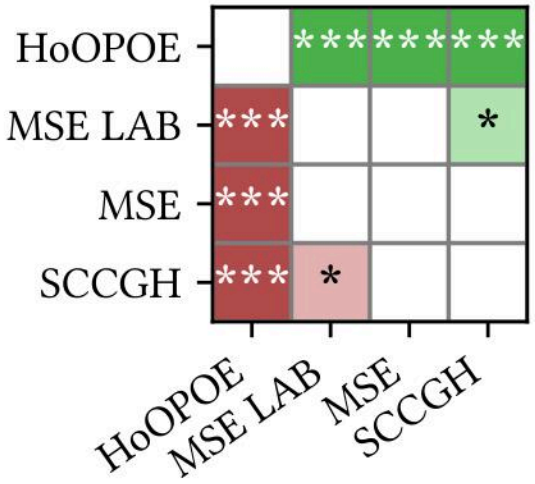


Fig. S70. 2D grid showing the statistical significance of each observed difference in JOD in Fig. S69. Red cells indicate that the benchmark on the Y axis performed worse than the corresponding benchmark on the X axis. Green indicates the opposite. A single star indicates $p < 0.05$. Two stars indicate $p < 0.01$ and three stars indicate $p < 0.001$.

S16 Generated SLM patterns

SLM patterns for stimulus 1 (ground truth at Fig. S5) for MSE, MSE LAB, SCCGH, and HoOPOE, alongside the associated reconstructions, are shown in Fig. S71. The optimization/SLM settings used are described in Sec. 4.1. SLM patterns for the real-life renders for stimulus 4 (ground truth at Fig. S8) for MSE, MSE LAB, SCCGH, and HoOPOE are shown in Fig. S72. Details about the CGH process for the real-life captures can be found in Supplementary Material S12.

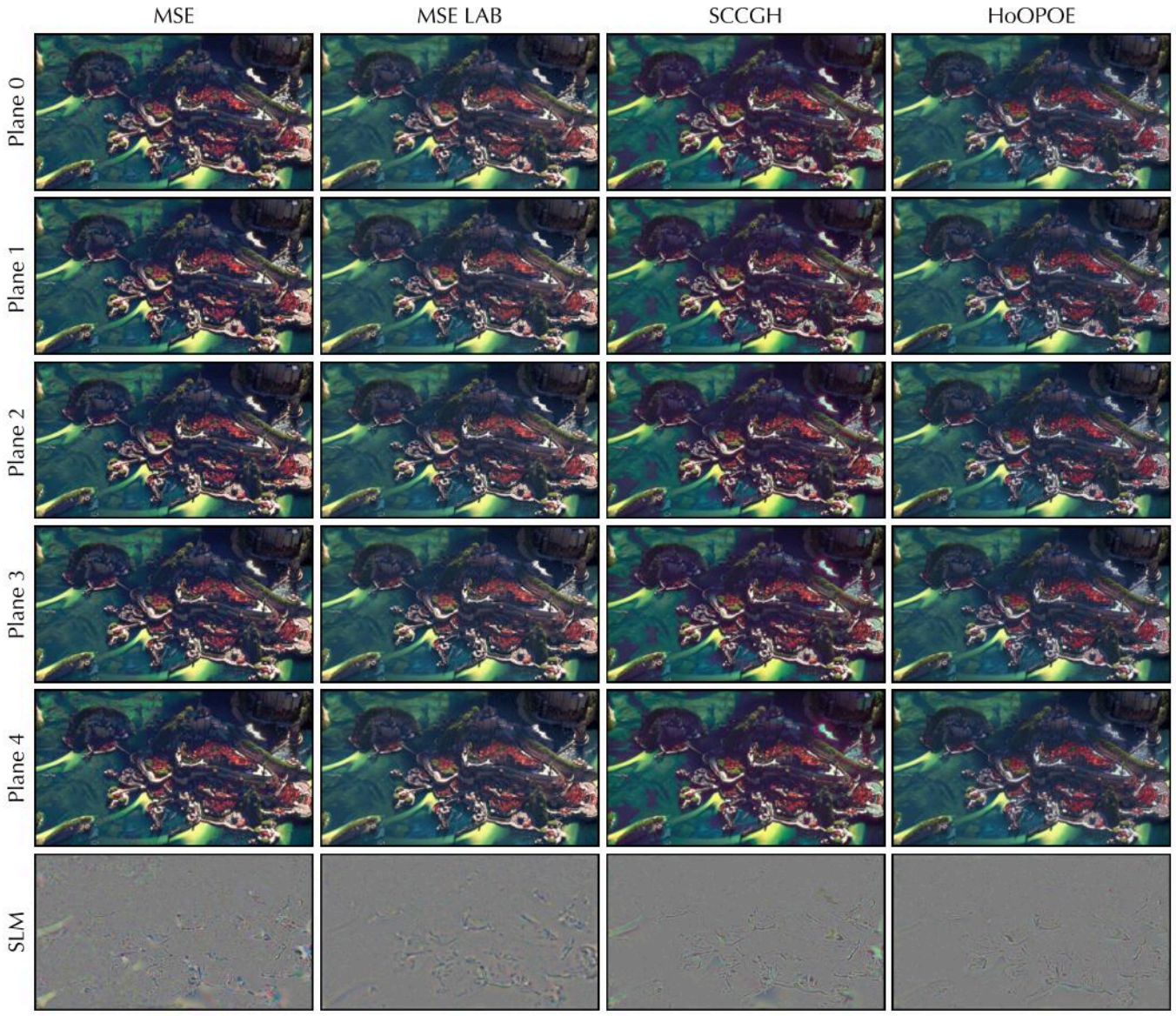
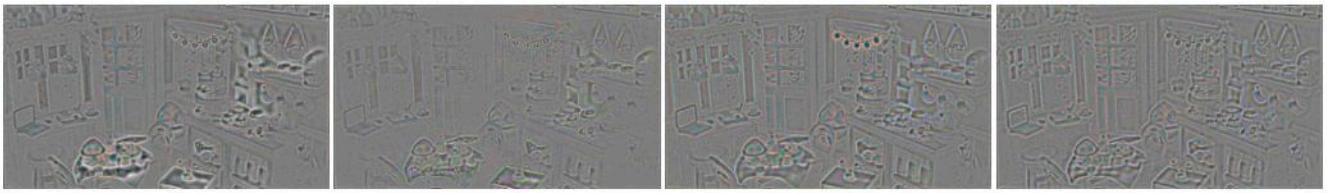


Fig. S71. SLM patterns for stimulus 1, alongside associated reconstructions, organized by plane, for MSE, MSE LAB, SCCGH, and HoOPOE. Copyright for the original 3D model (licensed under CC-BY-SA) goes to Krynski [2022].



(a) MSE (b) SCCGH (c) MSE LAB (d) HoOPOE

Fig. S72. SLM patterns for the hardware captures in Fig. 6. Details about the CGH process for the real-life captures can be found in Supplementary Material S12.

S17 HoOPOE on Video Data

We demonstrate that HoOPOE is also applicable to video reference data, as well as data generated from different scene representations. We utilize the bicycle Gaussian point cloud from Kerbl et al. [2023] to generate a 32 frame reference video (with 5 planes). Sec. 4.1 was utilized for the generation of the holographic video. Results are visualized in Fig. S73.



Fig. S73. Four frames (middle plane) from a holographic video, generated using HoOPOE, alongside associated SLM patterns per frame. All five planes per frame rendered. Original point cloud file from Kerbl et al. [2023].

S18 Runtime

We benchmarked several loss functions' latency and memory overhead for training eight frames of a 1920×1080 SLM pattern for 500 iterations. Each loss function was benchmarked 10 times. Benchmarking was implemented on a 40GB A100 GPU. The results are displayed in the table below. HoOPOE remains relatively lightweight to compute: Spatial HoOPOE is only 36% more time-consuming than MSE, while HoOPOE with temporal weighting requires 63% more compute time. Memory overhead is minimal. HoOPOE with temporal weighting requires 17% more memory than MSE.

Table S2. Runtime and memory overhead comparison across various loss functions. Standard deviations for latency provided.

Loss Function	Latency (s)	Memory Overhead (GB)
MSE	57.65 ± 0.043	7.26
MSE LAB	72.09 ± 0.094	7.48
MSE CAM16	87.09 ± 0.094	10.02
CVVDP	178.02 ± 1.009	16.53
SCCGH	74.12 ± 0.064	7.68
Spatial HoOPOE	78.48 ± 0.045	7.35
HoOPOE	93.89 ± 0.048	8.52

S19 Optical See Through UI Examples

HoOPOE also generalizes beyond the data demonstrated in the paper to other content. For instance, HoOPOE outperforms other loss functions for a proprietary optical-see-through (OST) UI dataset. We are not allowed to redistribute these renders, so we evaluate two mockup images instead. The focal stacks are depicted in Fig. S74 and Fig. S75. We train holograms under MSE, MSE LAB, SCCGH, and HoOPOE to replicate these focal stacks. We use propagation distances between 0.5 and 2.5 cm, evenly spaced. No temporal multiplexing is applied here – all holographic renders correspond to a single frame. We find that HoOPOE preserves text legibility much better than baseline loss functions. Comparisons can be found in Fig. S76 and Fig. S77.



(a) Plane 0 (closest)

(b) Plane 1



(c) Plane 2

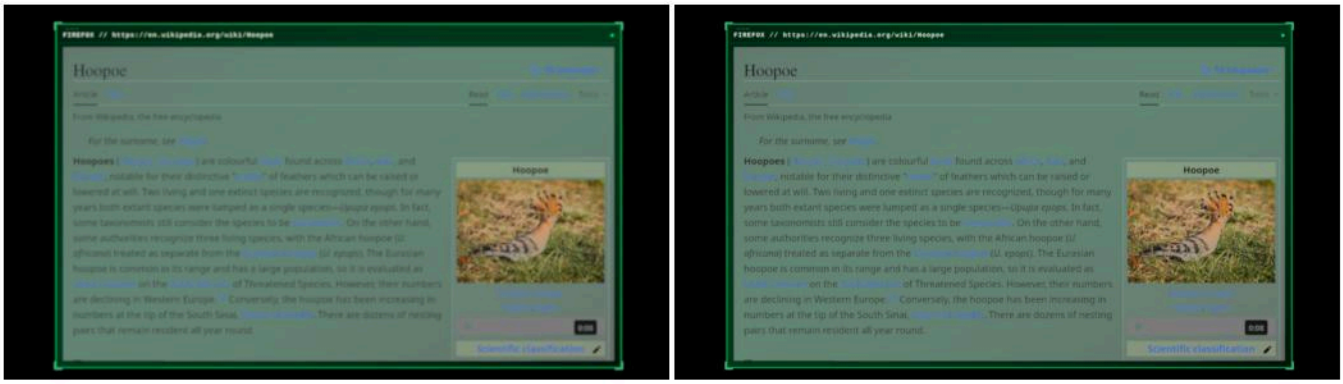


(d) Plane 3



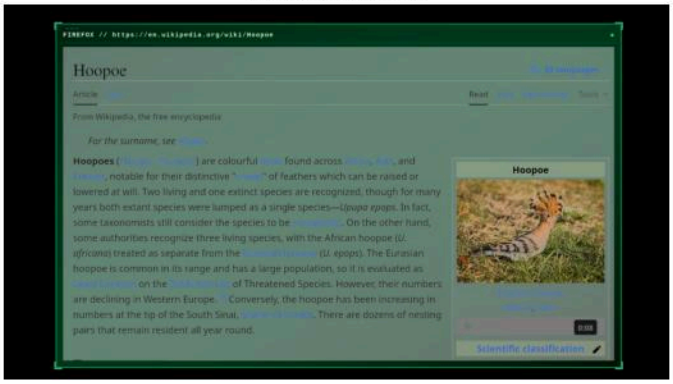
(e) Plane 4 (furthest)

Fig. S74. Focal stack for UI stimulus 0.

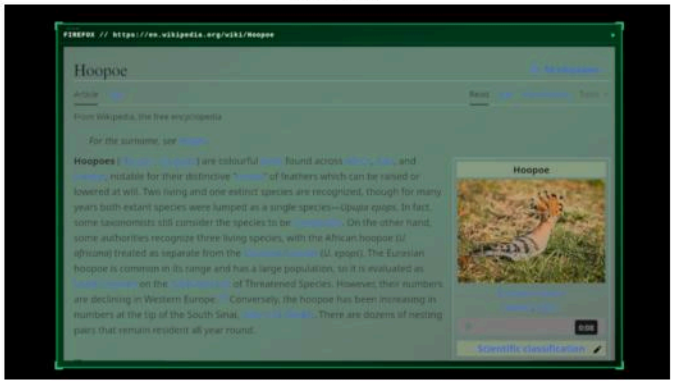


(a) Plane 0 (closest)

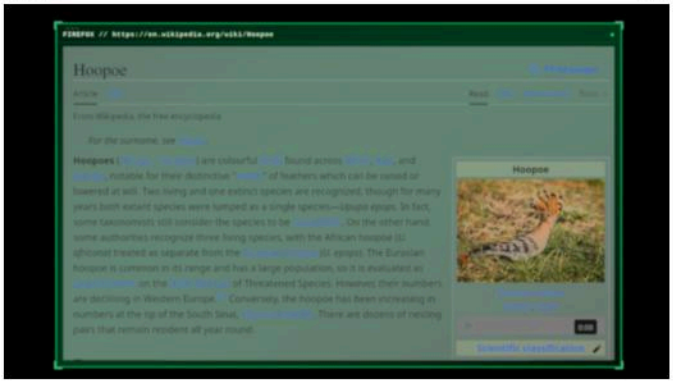
(b) Plane 1



(c) Plane 2



(d) Plane 3



(e) Plane 4 (furthest)

Fig. S75. Focal stack for UI stimulus 1. Original image taken from Wikipedia [2026]



Fig. S76. Comparison of images generated for UI stimulus 0. Rendered images simulated at a propagation distance of 1.5 cm. HoOPOE visibly reduces chromatic artifacts around text.

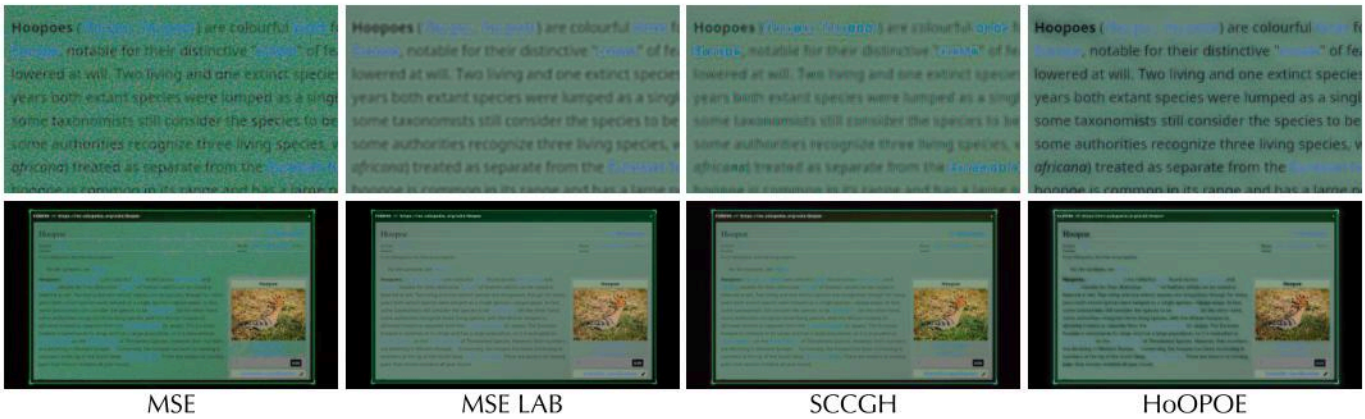


Fig. S77. Comparison of images generated for UI stimulus 1. Rendered images simulated at a propagation distance of 1.5 cm. HoOPOE significantly increases the legibility of text in the UI element.

S20 ASM Propagation Formulation

For the validation study in Sec. 4, we use the ASM propagation model (`propagation_ASM`) from the repository of Peng et al. [2020], modified to include a circular Fourier-plane aperture of $F_{ap} = 0.5$ to exclude high-frequency noise, as implemented in [Choi et al. 2021].

$$f(\phi) = \int\int \mathcal{F}(e^{i\phi(x,y)} u_{src}(x,y)) \mathcal{H}(f_x, f_y) e^{i2\pi(f_x x + f_y y)} df_x df_y \tag{6}$$

The above equation is sourced from section 3 of Peng et al. [2020]. In this equation, $\mathcal{F}$ represents the Fourier transform. $\phi(x,y)$ represents the phase delay at a specific point on the SLM. $u_{src}(x,y)$ is the coherent light source's illumination field at the same point. f_x and f_y represent spatial frequency along the x and y axes. However, unlike Peng et al. [2020]'s formulation, our formulation of the transfer function $\mathcal{H}$ differs. We integrate a circular Fourier-plane aperture of 0.5 to exclude high-frequency noise (Choi et al. [2021]).

$$\mathcal{H}\left(f_x, f_y\right) = \begin{cases} e^{i2\pi z\sqrt{\lambda^{-2} - f_x^2 - f_y^2}} & \text{if } |f_x| < f_x^{\max}, |f_y| < f_y^{\max}, \text{ and } f_x^2 + f_y^2 \leq \frac{F_{ap}^2}{2\Delta^2}, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

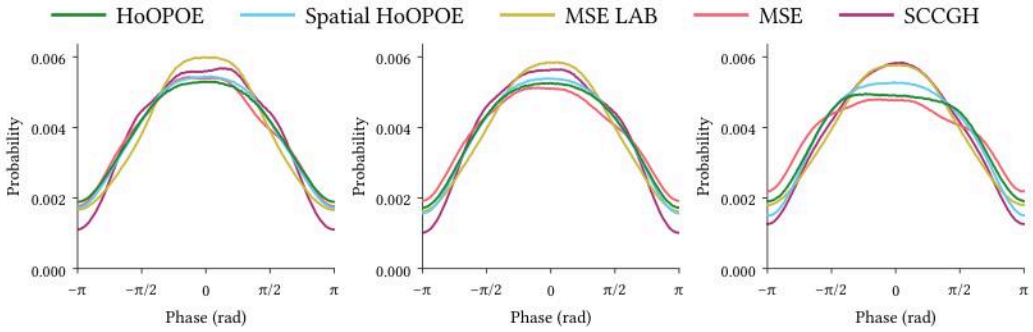
Here, λ is the laser wavelength and $f_x^{\max} = \left(\lambda\sqrt{1 + (2z/S_x)^2}\right)^{-1}$ with $S_x = 2N\Delta$ the padded window width (similar equations for $f_y^{\max}$ and S_y). Δ is the pitch of the SLM. z is the propagation distance. We use $F_{ap} = 0.5$. N is the horizontal pixel count of the SLM. Note that the band limit of Matsushima and Shimobaba [2009] is included in Peng et al. [2020]'s `propagation_ASM` code.

S21 Smoothness vs Randomness Analysis of SLM Pixel Phases

The phase distributions of holograms generated for our validation user study (Sec. 4) can be found in Fig. S78. The phase distribution of holograms generated for the hardware captures (Supplementary Material S12) can be found in Fig. S79. We also tabulate the Rayleigh distance (described in Chu et al. [2025]) of the optimized phase patterns to quantify phase randomness. We used the default settings for the `rayleigh_distance_loss` function in their repository (window = 11, sigma = 1.5, $c1 = 0.01^2$, $c2 = 0.03^2$). We input the SLM-plane field into the `rayleigh_distance_loss` function. Higher Rayleigh distances correspond to less random (more smooth) holograms. The standard deviation of the phase distribution and the Rayleigh distance are shown in Tbl. S3 (validation) and Tbl. S4 (hardware).

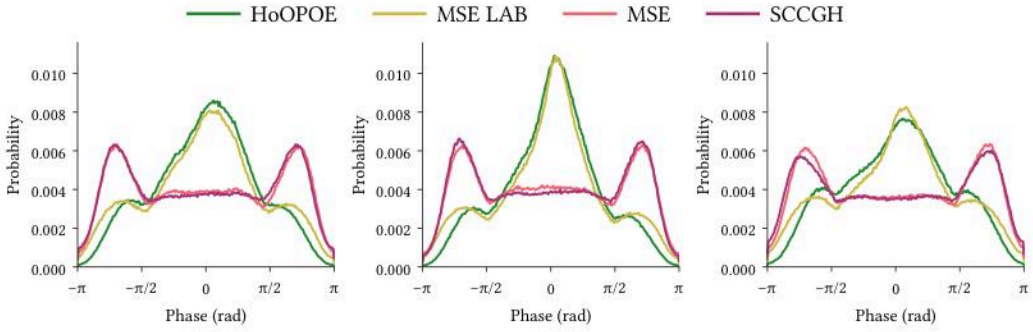
We observe that for the simulated reconstruction SLM patterns generated for Sec. 4, the phase distributions across all benchmarked loss functions diverge from the initialization of -0.3π to 0.3π. For the simulated results, there seems to be no correlation between visual quality and the phase distribution of the hologram. In our user study (Sec. 4.3), HoOPOE at 480 fps statistically significantly outperforms both MSE and MSE LAB at the same frame rate. However, the phase variance for HoOPOE ($\sigma = 1.56$) sits in between that of MSE ($\sigma = 1.58$) and MSE LAB ($\sigma = 1.48$). Additionally, phase randomness does not seem to be a good predictor of perceptual quality for the simulated reconstructions either. The Rayleigh distance of HoOPOE is lower than MSE LAB and higher than MSE, but HoOPOE outperforms both MSE and MSE LAB in user study results by a statistically significant margin.

However, when optimizing SLM patterns with the hardware capture pipeline described in Supplementary Material S12, the phase distribution of HoOPOE ($\sigma = 1.17$) and MSE LAB ($\sigma = 1.27$) are notably less diverged from the initialization compared to MSE ($\sigma = 1.60$) and SCCGH ($\sigma = 1.66$). HoOPOE and MSE LAB trained SLM patterns are also less random compared to MSE and SCCGH, as seen in the difference in Rayleigh distance values in Tbl. S4. Note that all holograms trained for the real-life captures started from the same uniformly random initialization of phase values from -0.3π to 0.3π. The comparative smoothness of MSE LAB and HoOPOE's SLM patterns likely reduced the impact of crosstalk on the captured images' quality at the cost of a more concentrated eyebox.



(a) Red ($\lambda = 638$ nm) (b) Green ($\lambda = 520$ nm) (c) Blue ($\lambda = 450$ nm)

Fig. S78. The probability distribution function for phase values (wrapped between $-\pi$ to π) for holograms generated for Sec. 4. Note that the HoOPOE distribution graphed above is for the 480 fps weight set. Statistics tabulated across all scenes and frames.



(a) Red ($\lambda = 638$ nm) (b) Green ($\lambda = 520$ nm) (c) Blue ($\lambda = 450$ nm)

Fig. S79. The probability distribution function for phase values (wrapped between $-\pi$ to π) for holograms generated for Supplementary Material S12. Statistics tabulated across all scenes and frames.

Table S3. Phase statistics of generated holograms for the validation study (Sec. 4). Phase σ represents the standard deviation of the phase values of the SLM patterns. The Rayleigh Distance is a measurement of randomness proposed by Chu et al. [2025]. Higher values correspond to less randomness. Statistics tabulated over all scenes, frames, and channels. Note that HoOPOE represents the 480 fps weight set.

Loss	Phase σ	Average Rayleigh Distance
Spatial HoOPOE	1.521	6.278
HoOPOE	1.556	6.052
MSE LAB	1.484	6.479
MSE	1.585	5.990
SCCGH	1.457	6.495

Table S4. Phase statistics of generated holograms for the hardware captures (Supplementary Material S12). Phase σ represents the standard deviation of the phase values of the SLM patterns. The Rayleigh Distance is a measurement of randomness proposed by Chu et al. [2025]. Higher values correspond to less randomness. Statistics tabulated over all scenes and channels.

Loss	Phase σ	Average Rayleigh Distance
HoOPOE	1.172	7.424
MSE LAB	1.267	7.311
MSE	1.601	6.052
SCCGH	1.660	6.309

References

Lorenzo Aiello. 2022. Blender 3.1 - Secret Deer. https://www.blender.org/download/demo/splash/blender-3.1-splash.zip

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A Convergence Theory for Deep Learning via Over-Parameterization. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 242–252. https://proceedings.mlr.press/v97/allen-zhu19a.html

AndrewH. 2002. Malahide Castle1.jpg. https://commons.wikimedia.org/wiki/File:Malahide_Castle1.jpg

Maliha Ashraf, Rafał K Mantiuk, Alexandre Chapiro, and Sophie Wuerger. 2024. castleCSF—A contrast sensitivity function of color, area, spatiotemporal frequency, luminance and eccentricity. *Journal of vision* 24, 4 (2024), 5–5.

Alexandre Chapiro, Yuta Asano, Rafal Mantiuk, Param Hanji, and Maliha Ashraf. 2024. XR-DAVID: XR Display Artifact Video Dataset. (2024). doi:10.17863/CAM.108210

Suyeon Choi, Manu Gopakumar, Yifan Peng, Jonghyun Kim, Matthew O'Toole, and Gordon Wetzstein. 2022. Time-multiplexed Neural Holography: A Flexible Framework for Holographic Near-eye Displays with Fast Heavily-quantized Spatial Light Modulators. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 32, 9 pages. doi:10.1145/3528233.3530734

Suyeon Choi, Manu Gopakumar, Yifan Peng, Jonghyun Kim, and Gordon Wetzstein. 2021. Neural 3D Holography: Learning Accurate Wave Propagation Models for 3D Holographic Virtual and Augmented Reality Displays. *ACM Trans. Graph. (SIGGRAPH Asia)* (2021).

Victor Chu, Felix Heide, Alberto Redo-Sanchez, Douglas Lanman, Oscar Pueyo-Ciutad, Florian Schiffers, Ethan Tseng, Oliver Cossairt, Grace Kuo, and Nathan Matsuda. 2025. *Artifact-resilient real-time holography.* Technical Report.

Ugur Cogalan, Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Colin Groth. 2025. MILO: A Lightweight Perceptual Quality Metric for Image and Latent-Space Optimization. *ACM Transactions on Graphics (TOG)* 44, 6 (2025), 1–11.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. doi:10.1109/CVPR.2009.5206848

Andrew M Derrington, John Krauskopf, and Peter Lennie. 1984. Chromatic mechanisms in lateral geniculate nucleus of macaque. *The Journal of physiology* 357, 1 (1984), 241–265.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. 2019. Gradient Descent Provably Optimizes Over-parameterized Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1eK3i09YQ

International Commission on Illumination. 2019. ISO/CIE 11664-4:2019. https://www.iso.org/standard/74166.html

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Piotr Krynski. 2022. Blender 3.3 LTS - Scanlands. https://www.blender.org/download/demo/splash/blender-3.3-splash.blend

Lynxsdesign. 2024. Blender 4.1 - Lynxdesign. https://www.blender.org/download/demo/splash/blender-4.1-splash.blend

Rafał Mantiuk, Param Hanji, Maliha Ashraf, Yuta Asano, and Alexandre Chapiro. 2024a. Metric comparison. https://www.cl/research/rainbow/projects/colorvideovdp/reports/benchmark/

Rafał K Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. 2021. Fovvideovdp: A visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–19.

Rafal K. Mantiuk, Param Hanji, Maliha Ashraf, Yuta Asano, and Alexandre Chapiro. 2024b. ColorVideoVDP: A visual difference predictor for image, video and display distortions. *ACM Trans. Graph.* 43, 4, Article 129 (July 2024), 20 pages. doi:10.1145/3658144

Eric Markley, Nathan Matsuda, Florian Schiffers, Oliver Cossairt, and Grace Kuo. 2023. Simultaneous Color Computer Generated Holography. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 22, 11 pages. doi:10.1145/3610548.3618250

Kyoji Matsushima and Tomoyoshi Shimobaba. 2009. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Optics express* 17, 22 (2009), 19662–19673.

Nicole Morena. 2023. Blender 3.5 - Cozy Kitchen. https://www.blender.org/download/demo/splash/blender-3.5-splash.blend

Eli Peli. 1990. Contrast in complex images. *Journal of the Optical Society of America A* 7, 10 (1990), 2032–2040.

Yifan Peng, Suyeon Choi, Nitish Padmanaban, and Gordon Wetzstein. 2020. Neural holography with camera-in-the-loop training. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.

Gaku Tada. 2023. Blender 4.0 - Gaku. https://www.blender.org/download/demo/splash/blender-4.0-splash.blend

Taimoor Tariq, Nathan Matsuda, Eric Penner, Jerry Jia, Douglas Lanman, Ajit Ninan, and Alexandre Chapiro. 2023. Perceptually adaptive real-time tone mapping. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.

Okan Tarhan Tursun, Elena Arabadzhiyska-Koleva, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk. 2019. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. doi:10.1109/TIP.2003.819861

Z. Wang, E.P. Simoncelli, and A.C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. 1398–1402 Vol.2. doi:10.1109/ACSSC.2003.1292216

Wikipedia. 2026. Hoopoe. https://en.wikipedia.org/w/index.php Page Version ID: 1350762486.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 586–595. doi:10.1109/CVPR.2018.00068