

Towards Passing Visual Turing Test with Computational 3D Displays and Appearance Modeling

By

Liang Shi

S.B., Beihang University (2014)

S.M., Stanford University (2016)

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

© 2024 Liang Shi. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Liang Shi
Department of Electrical Engineering and Computer Science
May 17, 2024

Certified by: Wojciech Matusik
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejwski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Towards Passing Visual Turing Test with Computational 3D Displays and Appearance Modeling

by

Liang Shi

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2024, in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Abstract

In Neal Stephenson's 1992 science-fiction novel "Snow Crash," the term "metaverse" was coined to describe a virtually persistent world in which humans, as avatars, interact with each other and software agents just as they would in the physical world. This is no longer science fiction with the rapid development of virtual and augmented reality devices in the past decade. To stimulate future development and research, Meta Inc. (previously Facebook) defined a grand challenge termed visual Turing test in 2022, asking if the virtual experience can be created as vivid and realistic as the real-world such that a human observer can no longer distinguish. Passing this test requires significant technological advancements in areas such as display, rendering, acquisition, sensing, modeling, and simulation. The focus of this thesis is on developing next-generation computational 3D near-eye displays to deliver a more realistic and natural 3D experience, as well as devising inverse procedural material (PM) modeling and capture frameworks that allow non-expert users to bring real-world appearances into a digital twin in a few simple clicks.

Existing near-eye 3D displays rely on the stereoscopic principle by showing parallax images to deceive the brain into perceiving depth from 2D images. This deviates from the experience of how humans observe the real world, causing visual discomfort and vision degradation after prolonged use. Computer-generated holography (CGH) is widely regarded as the next-generation technology capable of utilizing light interference and diffraction to produce true 3D graphical projections. However, its adoption was fundamentally limited by the low image quality and vast computational cost. Similarly, PMs have recently emerged as the industry standard for representing physically-based, spatially-varying appearance due to their unlimited resolution, editability, and memory efficiency. Nonetheless, the extensive domain knowledge and labor required to master and author PMs have made it difficult for non-experts to create and operate them.

This thesis exploits recent advances in machine learning, optimization, rendering, and differentiable software infrastructures to develop algorithms and user-friendly tools for synthesizing, compressing, encoding/decoding, and deploying photorealistic 3D holograms, as well as for authoring, editing, and manipulating procedural materials. The ultimate objective is to pass the visual turning test and democratize the creation of a metaverse for

ordinary people.

At the very end of this thesis, an extrapolation of the Virtual Turing Test is applied for physical reproduction, and a novel 3D printing system that fabricates painting facsimiles faithfully to the originals will be discussed, as well as the potential of this technology to help culture preservation and education.

Thesis Supervisor: Wojciech Matusik

Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I would like to thank my Ph.D. advisor, Prof. Wojciech Matusik, for the unconditional trust he puts in me. Wojciech helped me build my confidence even prior to the beginning of the program, when he flew to the Bay area and visited me at Stanford. To this day, I can still hardly believe it, and the hour-long conversation we had at Peet's cafe on the Clark center's top floor convinced me that Wojciech is the professor I should work with and that I could accomplish remarkable things in his lab. During my PhD, Wojciech gave me full respect and freedom to explore my research interest even though nobody else in the lab was working on anything close. He encouraged me to think boldly, scientifically, and not be afraid of aiming high. He provided me with all the opportunities and exposure to earn funding support for my projects. He made space for me to set up a small optics lab with discretionary use of funding to purchase all necessary lab equipment. There are countless other things to mention, all of which smoothed out my PhD journey, for which I will always deeply appreciate. Meanwhile, his patience and understanding of the setbacks, hurdles, and delays in different projects significantly reduced my stress and gave me time to overcome all types of challenges.

Next, I want to extend my gratitude to my Master's advisor, Prof. Gordon Wetzstein, for introducing me to the realm of scientific research and coaching me on establishing a strong foundation and strategic approach to tackling difficult problems. His rigor, systematic thinking, craving for greatness, and pursuit of aesthetics have significantly influenced my research style down to the finest level. His boundless passion and resolute determination have been a constant source of inspiration and motivation for me, especially during difficult times when I feel overwhelmed. I am immensely grateful for his strong recommendation, which played a pivotal role in my acceptance into the PhD program at MIT. Without his endorsement, I would not have embarked on this exciting journey.

I would like to thank Prof. Cardinal Warde and Prof. Fredo Durand for serving on my thesis committee. Prof. Warde, who also serves on my RQE committee, has offered me in-depth technical advice on optics and physics, which significantly broadened my perspective and enlightened me with new solutions. Prof. Durand taught me how to strategically plan

the big picture of my research and better position myself in the academic job market to seize greater opportunities.

I would like to sincerely thank five exceptional mentors who have significantly impacted my professional development during my internships: Ward Lopes, Fu-Chung Huang, Milos Hasan, Kalyan Sunkavali, and Changwon Jang. During my first internship at NVIDIA, Ward provided invaluable guidance and hands-on experience in optics that laid the foundation for my subsequent PhD work in holographic display. Fu-Chung was instrumental in mentoring me through my first SIGGRAPH submission, from drafting the manuscript to solving complex parallel programming challenges. During my second internship at Adobe, Milos and Kalyan introduced me to the exciting and emerging new direction of differentiable procedural material modeling, which I found extremely fascinating and continued to work on in the later phase of my PhD. Finally, during my last internship at Meta, Changwon demonstrated me how to handle technical and logistic challenges calmly and efficiently amid a rapidly changing environment. Together, their mentorship not only helps me become a better researcher but also helps me become a better person with greater responsibility and eagerness to help others.

I would like to particularly thank my labmate Beichen Li, who has been an incredible collaborator to me. It's been amazing that he and I have collaborated on more than 5 papers and successfully commercialized our technology to Adobe. His self-discipline, outstanding technical skills, and meticulous execution have been the driving force behind the success of our projects. I wish him all the best for the rest of his PhD journey. I would like to thank Petr Kellnhofer for his extremely helpful guidance and mentorship during the arduous process of preparing and revising our Nature paper. His support and expertise were crucial in navigating the complex and challenging requirements from the reviewers and editors. I would like to thank Changil Kim and Mike Foshey for providing their valuable expertise in deep learning and 3D printing, which proved critical in the success of our fine art reproduction project. I would like to thank Donghun Ryu for joining the same research battlefield with me in the last one and a half years of my PhD. It is such a fruitful experience to mentor him, witness his growth, and work together to complete my final PhD project. I hope that he can carry on the momentum of our research direction in CDFG and pass it on

to future generations of PhDs, just as I passed the torch to him.

My PhD will also be incomplete without many important external collaborators. I would like to thank Prof. Kaan Akşit for kindly inviting me to collaborate on the Holo-MultiColor project with his talented student, Koray Kavaklı, and for offering me several opportunities to present my PhD works to a bigger audience. I would like to thank Prof. Szymon Rusinkiewicz for his insightful mentorship in the fine art reproduction project. I would like to thank Thomas J Lane for offering the opportunity to work on a coherent diffraction imaging project at SLAC National Laboratory during my time at Stanford.

I want to thank my best PhD comrades, Jie Xu and Yuanming Hu, for all the delightful moments we shared when the three of us used to sit in the corner of 32-322. Your smartness, optimistic outlook toward life and research, and charming personality made even the most challenging days of my early PhD life much more easier and manageable. The memories of making flying cups in the lab, flying UAVs in the gymnastics and running fields, and debugging the Raft system of 6.824 will always be cherished in my heart.

I would like to thank other current and past CDFG members for fostering a welcoming and supportive environment that has allowed me to thrive and develop: Mina Konaković Luković, Vahid Babaei, Timothy Erps, Tae-Hyun Oh, Wan Shou, Mélina Skouras, Yagiz Aksoy, Pitchaya Sitthi-amorn, Kiril Vidimče, Peter Yichen Chen, Bolei Deng, Bohan Wang, Javier Ramos, Kui Wu, Bo Zhu, Ruitao Su, Wenqiang Chen, Alexandre Kaspar, David Kim, James Minor, Andrew Spielberg, Subra Sundaram, Harrison Wang, Lara Zlokapa, Tao Du, Adriana Schulz, Michael Foshey, Yifei Li, Yiyue Luo, Pingchuan Ma, Liane Makatura, Yunsheng Tian, Allan Zhao, Minghao Guo, Yichen Li, Wil Norton, Megan Tjandrasuwita, Felicia Raton, Roger White, Kshama Ananthapura, Bryt Bradley, Yihui Li, Rupert Deng, Buttercup Foshey. I would also like to thank early members of Stanford Computational Imaging Group, who I have overlapped with: Robert Konrad, Shikhar Shrestha, Felix Heide, Isaac Kauvar, Julie Chang, Xuemei Hu, and Samuel Yang.

I would like to thank my tennis friends at both MIT and Stanford: Xinzhe Fu, Zhiheng Yang, Jiaquan xx, Shiyu Zhu, Yang Liu, Tianyu Li, Fangze Tang, Qiuyu Fu, Penglu Fang, Ji Lin, Jinye Zhang, Ruoxuan Xiong, Yang Yue, Xuexiu Han, who have not only helped me improve my skills but also brought me immense joy both on and off the court. Many

of them have become my best friends, and I sincerely hope all of them continue to relish playing tennis and excel in their careers.

I would like to thank my outside-of-school friends in Boston: Jonathan Wu, Xinyue Zhong, Zhaoyu Zheng, Qiuyi Hong, Fangyue Peng, Wang Zhang, Xia Xiao, Zeyu Wu, Yuzhe Yang, and Luxin Zhang for making my after-school life relaxing and exciting. I will always remember the countless days and nights we spent playing poker and board games, celebrating each other's birthdays and holidays, hosting animal parties and BBQs, and traveling east and west with many of you. I couldn't imagine how much tougher my PhD life would be without you around and I hope our friendship will last forever, no matter which corner we are in this world.

I owe many thanks to my parents in China, who have instilled in me a love of science and technology from a young age. Together with my parents-in-law, they have provided me and my wife immense physical and mental support during COVID-19 and at the time we had our first child.

Finally, I would like to convey my most sincere gratitude to my wife for her unwavering support throughout my academic research journey, dating back to our time at Stanford and continuing through my PhD at MIT. Her love, encouragement, and understanding have been a constant source of strength and motivation, and I could not have made it through without her.

Contents

1	Introduction	41
1.1	Thesis Overview	45
2	Related Works	47
2.1	Computational Near-Eye 3D Displays	47
2.2	Material Acquisition and Modeling	50
2.3	3D Printing for Appearance Reproduction	52
3	Light Field-based Holographic Displays	55
3.1	Preliminary	56
3.1.1	Light Field	56
3.1.2	Computer Generated Holography (CGH)	56
3.1.3	Connecting Light Field and CGH with Wigner Distribution Functions	59
3.2	Method	60
3.2.1	Holographic Light Field Rendering under Plane Wave Illumination	60
3.2.2	Spherical-Holographic Light Field Rendering	65
3.2.3	Calculating Object Wave for Displaying on SLM	68
3.2.4	Software Implementation	70
3.3	Theoretical Analysis	71
3.3.1	Sampling Outside the Maximum Diffraction Cone	71
3.3.2	Resolution Limits on the Depth of Field (DoF)	73
3.3.3	Elemental Image Resolution & Hogel Partitioning	80
3.3.4	Practical Design Trade-off	82

3.4	Experiment	85
3.4.1	Experimental Setup	85
3.4.2	Experimental Evaluation	87
3.5	Discussion	88
4	Deep Learning Holography	91
4.1	Tensor Holography: Real-Time 3D Holography with Deep Neural Networks	92
4.1.1	Motivation	92
4.1.2	Occlusion-Aware Point-Based Method (OA-PBM)	94
4.1.3	MIT-CGH-4K: the First Large-Scale Hologram Dataset	97
4.1.4	CNN Model Architecture, Training, Evaluation, and Comparisons	100
4.1.5	Anti-Aliasing Double Phase Method (AA-DPM)	108
4.1.6	Experimental Setup	110
4.1.7	Experimental Results	112
4.1.8	Discussion	116
4.2	Tensor Holography V2: End-to-end Learning of 3D Phase-only Holograms	116
4.2.1	Motivation	117
4.2.2	Layered Depth Images and Silhouette-Mask Layer-Based Method for Hologram Generation.	118
4.2.3	MIT-CGH-4K-V2	123
4.2.4	End-to-end Learning of 3D Phase-only Hologram	125
4.2.5	Compensating Hardware Imperfection	129
4.2.6	Software Implementation	132
4.2.7	Simulation Assessment	134
4.2.8	Experimental Setup	138
4.2.9	Experimental Results	139
4.2.10	Discussion	141
4.3	Neural Compression for Hologram Images and Videos	144
4.3.1	Motivation	145
4.3.2	High Fidelity Hologram Compression (HiFiHC)	147

4.3.3	Assessment	149
4.3.4	Discussion	157
5	Multi Color Holography	159
5.1	Motivation	160
5.2	Method	161
5.3	Experimental Setup	164
5.4	Simulation & Experimental Assessment	165
5.5	Discussion	174
6	Ergonomic-Centric Holography	177
6.1	Motivation	177
6.2	Method	181
6.2.1	Simulation of Incoherent Depth of Field Targets for 3D CGH	181
6.2.2	Optimization of Time-multiplexed 3D CGH with High-order Diffrac-	
	tions	181
6.2.3	Software Implementation	183
6.3	Experiment & Assessment	185
6.3.1	Experimental Setup	185
6.3.2	Unfiltered Mode Evaluation	186
6.3.3	Pupil-Mimicking Mode Evaluation	189
6.3.4	Ablation Study of Pupil Optimization Scheme	192
6.3.5	Evaluation of Downsampling Operation in High-Order Diffraction	
	Propagation	195
6.3.6	Evaluation of Number of Time-multiplexed Frames for Speckle	
	Reduction and Incoherent Defocus Reproduction	196
6.3.7	Reproduction of Intra-pupil Motion Parallax	197
6.3.8	Additional Phase Patterns & Focal Stacks	199
6.4	Limitations & Discussions	201

7	Differentiable Procedural Material Modeling and Capture	205
7.1	Preliminaries	206
7.1.1	Rendering Equation & Material Modeling	206
7.1.2	Procedural Materials	208
7.2	Motivation	210
7.3	DiffMat: A Differentiable Procedural Material Modeling Library	212
7.3.1	Design Overview	212
7.3.2	Generator Nodes	214
7.3.3	Filter Nodes	215
7.3.4	Automatic Graph Translator	216
7.4	End-to-end Procedural Material Capture	217
7.4.1	Image-based Parameter Optimization	217
7.4.2	Network-based Graph Selection and Initialization	220
7.5	Results	222
7.5.1	Material Capture from Synthetic Images	222
7.5.2	Material Capture from Real Data	224
7.5.3	Comparisons with Previous Work	224
7.5.4	Applications	229
7.5.5	Limitations & Discussion	232
8	3D Printing for Multispectral Fine Art Reproduction	235
8.1	Motivation	236
8.2	System Overview	238
8.3	Hardware & Fabrication & Ink Set	238
8.4	Data-driven Spectral Prediction & Layout Planning	243
8.4.1	Spectral Prediction Model	243
8.4.2	Layout Prediction Model	243
8.4.3	Spectral Vector Error Diffusion	247
8.4.4	Training Protocol	249
8.5	Experiments & Results	249

8.5.1	Model Evaluation	249
8.5.2	Painting Reproduction Evaluation	257
8.5.3	Run-time Performance	261
8.6	Discussion & Limitation	263
9	Conclusion & Future Work	265
9.1	Discussion	266
9.2	Future Work	270

List of Figures

3-1	Plane wave illuminated holographic display. Each display pixel diffracts wavefront into a spatially-uniform light cone. Rays are back-traced for each point emitter within the diffraction cone. The viewer can only observe the full extent of display behind the intersection of two marginal cones diffracted from the edge of the display.	58
3-2	We render the holographic light field with both color (top-left) and position (top-right) information. The per-SLM-pixel phase (bottom left) can be derived from the position, and the amplitude is obtained from both inputs. Although the hogel-to-elemental is a one-to-one mapping, but the mapping within block is a complete bipartite graph, resulting in heavy computation. .	61
3-3	Holographic light field rendering for plane wave illumination. The view frustum setup (top) covers the entire space behind the SLM and avoids geometry clipping, as opposed to convention frustum (bottom). The per-SLM-pixel perspective is obtained by aligning the diffraction cone in the far-clipping plane with the sliding window w_2 . Pixels rendered from the ambiguity region, as shown in the inset, needs a maximal diffraction culling.	63
3-4	Spherical wave driven holographic display. Each display pixel diffracts a light cone toward the focus. The entire display is observable by placing pupil in the diamond-shape viewing zone.	65

3-5	Holographic light field rendering for spherical wave illumination. The camera view frustum is spatially non-uniform and steered off-axis to capture the hogel's entire field of view. Calculating per-SLM-pixel view is similar to that in plane wave setup. Points in red triangle can be distinguished by examining the angle with respect to the central ray of diffraction cone (dashed green line).	66
3-6	Comparing performance under CPU and different GPU implementations, and different illuminations. Under plane wave method, the elemental image size is 128×128 (2048×1152 overall points cloud), and 160×160 (2560×1440 overall) under spherical wave method; both methods achieve 30 pixels per degree angular sampling on a 1920×1080 resolution SLM.	69
3-7	WDF analysis of aliasing caused by sampling outside maximum diffraction cone. (Left): The dashed lines denote replicas from discrete sampling. The display's space-bandwidth product in green rectangle is bounded by the spatial extent w_s and maximum spatial frequency $\pm 1/2\Delta p$. (Right): The visible field is the lateral extent of the sheared space-bandwidth product. Each constructively interfered replica in red dash line projects to a aliased point at displacement depicted the Nyquist sampling rate.	72
3-8	Photographed aliasing response (left) and simulated aliasing response (right) due to sampling outside maximum diffraction cone as discussed in Section 3.3.1. Multiple Fresnel zone plates are presented in the over-sampled diffraction pattern (red inset). At non-paraxial region, the aliased points no longer preserve a circular shape.	73

3-9	Spatial resolution limit. (First Row): <i>spatially-limited sampling outside the DoF</i> . The red bounding box marks the width of the main lobe as resolution limit, and the bounding box scales proportional to the distance from the point emitter to the display. The observed resolution limits (Eq. (3.26)) is the same for display with the same spatial extent but smaller pixel pitch (b,c); however, their DoF (Eq. (3.23)) are different. (Second Row): <i>frequency-limited sampling within the DoF</i> . Spatial resolution in (e,f,h) remains at Δp . However, when the display boundary crops the diffraction cone, reconstruction degrades as shown in (g) due to under-sampling.	74
3-10	WDF analysis of resolution limit outside the DoF. (Left): Frequency is minimally dilated to the main lobe of the Sinc function near the center, while the peripheral dilation (orange Sinc) has wider spread and weaker peak amplitude. (Middle): The yellow dash line marks the sheared minimally dilated frequency at the observed distance. (Right): The dash lines bound the region where Sinc response is always positive, and the projected circle bounds the resolution.	76
3-11	WDF analysis of resolution limit under frequency-limited sampling inside the DoF. (Left): Active sampling region is limited within the maximum diffraction cone projection. Frequency is minimally dilated to the main lobe of the Sinc function at x_0 . (Middle): The yellow dash line marks the sheared minimally dilated frequency at the observed distance. Within the DoF, minimum spatial dilation remains at Δp . (Right): The dash lines bound the region where Sinc response is always positive, and the projected WDF inside this region only interferes constructively.	78

3-12	Illustration of spatial resolution as a function of SLM width and pixel pitch.	
	For $(w_s, \Delta p)$ SLM (yellow top), red and blue point inside DoF are reproduced at Δp resolution. For $(w_s, \Delta p/2)$ SLM (dark blue top), resolution inside DoF is doubled (red point). Sampling of blue point becomes spatially-limited as DoF span halves, however, spatial resolution is still higher than $(w_s, \Delta p)$ case since active sampling area is the entire SLM. For both cases, the green point subtends to the entire SLM and are reproduced at same spatial resolution. For $(2w_s, \Delta p)$ SLM (yellow top+bottom), the expanded DoF encapsulates larger space with Δp resolution and produces higher resolution outside DoF, i.e. green point.	79
3-13	Comparisons of elemental image resolution w_l (blue inset). We simulate focusing on the bunny on a 1080p $8\mu m$ CGH. The angular sampling rate is (a): 6, (b): 18, (c): 30, and (d): 45 pixels per degree respectively. The reconstructions exhibit obvious aliasing on the top row (red inset) but are smoothed out greatly above 30 pixels per degree (c).	81
3-14	Design trade space between the pixel pitch Δp and the SLM width w_s assuming eye box $w_e = 10mm$ and $f\# = 1$. (Left) Maximum FoV increases with smaller pitch and wider SLM. The white dash line separates the above pitch bounded case from the lower lens bounded case (shaded region). (Right): Minimum Eye relief (focal length) decreases with pitch size (Eq. (3.12)) for a fixed eye box requirement, and the current off-the-shelf SLM is quite far to the lens-bounded region.	83
3-15	Illustration of (left): pitch-bounded and (right): lens-bounded display modes. When the spherical illumination is constrained by the $f\#$ (right), the diffraction cones emitted from the edges of the SLM shift away, and eye relief increases on a wider SLM.	83
3-16	Optical setup relaying amplitude modulation to phase modulation using field sequential colors. The eye box also filters higher order diffractions.	85

3-17	Optically captured hologram results at different focal distances. The color artifacts is caused by the laser filtering, alignment, and inprecise beam stop, but does not affect the correctness. We show the reference 2D image in (a). When the object is displayed on a hologram, we can support accommodation (b and c).	86
3-18	The proposed method (top row, photographed) provides an accurate rendering setup (Section 3.2.1) such that no geometry is clipped near the SLM nor aliased reconstruction shown in prior work (c and d). The captured results from a 1080p SLM closely resemble the simulations (red insets).	87
3-19	Comparisons of (a): the ideal amplitude and phase hologram with (b): phase-only hologram, (c): lower phase bit rate, and (d): lower amplitude resolution. We compare the simulated results (insets) using a 1080p SLM.	88
4-1	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 resulted pixel depth distribution is statistically uniform. b , An occlusion-aware point-based method 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.	94

4-2 Visualization of masked Fresnel zone plates computed by OA-PBM

and performance comparison of foreground occlusion. a, A depth im-

age cropped from a frame of Big Buck Bunny. Three regions with differ-

ent depth landscapes are highlighted in different colors. **b,** Masked Fresnel

zone plates computed for the center 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 PBM, OA-PBM, and Fresnel diffraction. The scene is a cropped mod-

ulation transfer function bar target with a step depth profile. PBM leaks a

considerable portion of the background into the foreground due to a lack

of occlusion handling. The artifacts are clearly visible in the original un-

magnified view. OA-PBM removes a significant portion of the artifacts and

the remaining artifacts are visually insignificant in the unmagnified view. **d,**

Comparison of focal stacks reconstructed by PBM and OA-PBM for the

Big Buck Bunny scene. The orange bounding boxes mark the background

leakage in the PBM reconstructions. 96

4-3 Samples of MIT-CGH-4K and comparison with the DeepFocus dataset.

a, The RGB-D image, amplitude, and phase of two samples from MIT-

CGH-4K. The RGB image records the amplitude of the scene (directly

visualized in sRGB space) and consists of large variations in color, texture,

shading, and occlusion. The pixel depth has a statistically uniform distri-

bution throughout the view frustum. The phase presents high-frequency

features at both occlusion boundaries and texture edges to accommodate

rapid depth and color changes. **b,** A sample RGB-D image from the Deep-

Focus dataset[258]. **c,** Histograms of pixel depth distribution computed for

MIT-CGH-4K and the DeepFocus dataset. 99

4-4	Schematic diagram of the midpoint hologram calculation.	a, A holo-
	graphic display magnified through a diverging point light source.	b, A
	holographic display unmagnified through the thin lens formula.	c, The tar-
	get hologram in this example is propagated to the center of the unmagnified	
	view frustum to produce the midpoint hologram. The width of the maxi-	
	mum sub-hologram is significantly reduced.	 101

4-5 Performance evaluation of occlusion-aware point-based method 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 the “Big Buck Bunny” (© by Blender Foundation).

The bottom right inset visualizes the depth image. **b**, Comparison of focal

stacks reconstructed at highlight regions in **a**. The CNN prediction is vi-

sually 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 image [139]. **d**, Per-

formance comparison of PBM, OA-PBM, and CNNs with various model

capacities. The default CNN model consists of 30 convolution layers / 24

filters per layer, the small and mini models has 15 and 8 convolution layers

respectively. The reduction of convolution layers gracefully degrades the

reconstructed image quality. The mini model runs in real time (60Hz). **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 (1st). The ablation of attention mask (2nd) dilutes CNN’s attention

to out-of-focus features and results in inferior performance. The ablation

of data loss (3rd) removes the regularization of phase information and leads

to poor generalization to unseen examples and large focal stack error. The

ablation of perceptual loss (4th) removes the guide of focal stacks and uni-

formly degrades the performance. **g**, Comparison of a ground truth Fresnel

zone plate and a CNN prediction (by a model with 30 layers / 120 filters

per layer) computed for a 6 mm distant point (propagated for another 20

mm for visualization). 103

4-6 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 patterns (USAF-1951 and RCA Indian-	
head).	105
4-7 Evaluation of tensor holography CNN on additional computer-rendered	
scenes. CNN prediction of amplitude and phase along with focused recon-	
structions for holograms of a, a living room scene from the DeepFocus	
dataset [258] and b, a night landscape scene from the Stanford light field	
dataset [200].	106
4-8 Evaluation of tensor holography CNN on real-world captured scenes.	
CNN prediction of amplitude and phase along with focused reconstructions	
for holograms of a, a statue scene and b, a mansion scene. Both scenes are	
from the ETH light field dataset [139].	107
4-9 Comparison of the original double phase method and anti-aliasing dou-	
ble phase method. Reconstruction of two real-world scenes from the en-	
coded 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 artifacts. (left) Double phase	
method. (right) anti-aliasing double phase method.	110
4-10 Holographic display prototype used for the experimental results shown	
in this Letter. The control box of the laser, Labjack DAQ, and camera are	
not visualized in the figure.	111

4-11 Experimental demonstration of 2D and 3D holographic projection. a,	
Scheme of our phase-only holographic display prototype. Only the green	
laser is visualized. b, (left) A flat (2D) target image (by Ana Blazic Pavlovic/Shut-	
terstock) for testing the spatial resolution of our prototype. (right) A photo-	
graph of the CNN predicted hologram (encoded with anti-aliasing double	
phase method) displayed on our prototype. The insets on the top right	
show the magnified bounding boxes. c, Photographs of our prototype pre-	
sending a real-world captured 3D couch scene in Fig. 4-5c. The left photo-	
graph is focused on the mouse toy and the right photograph is focused on	
the perpetual desk calendar. d, Photographs of our prototype presenting a	
computer-rendered 3D Big Buck Bunny scene in Fig. 4-5a. The left photo-	
graph is focused on the bunny’s eye and the right photograph is focused on	
the background tree leaves.	113
4-12 Additional experimental demonstration of 3D holographic projection	
(part 1). The RGB-D input can be found in Fig. 4-8.	114
4-13 Additional experimental demonstration of 3D holographic projection	
(part 2). The RGB-D inputs can be found in Fig. 4-8 for a , and Fig. 4-6	
for b	115
4-14 Methods comparison for occlusion modeling. The top 4 images consti-	
tute a 2-layer LDI that completely depicts the test scene. Different types of	
artifacts are only marked for the front focus images.	120
4-15 Comparison of LDI and voxel grid as 3D scene representation. a Ren-	
dering of an LDI records the intersections between a penetrating ray (col-	
ored differently after every new hit) and the scene objects (front-facing	
surfaces) at every spatial location. b First three layers of the LDI rendered	
for the scene in (a). The number of valid pixels rapidly decreases as the	
index of the layer increases. c Rendering of a voxel grid that partitions	
the 3D scene into equal-length slabs and generate one RGB image for each	
slab. The depth resolution is proportional to the number of slabs.	121

4-16 SM-LBM hologram quality as a function of LDI layers and depth discretization.	
a The addition of more LDI layers provides diminishing returns on the hologram fidelity and the improvement plateaus after 5 layers.	
b A depth resolution over 11 bits (2048) results in negligible difference to the image quality.	124
4-17 Overview of training procedure.	
In the first stage, the CNN is trained to reproduce the ground truth midpoint hologram with direct supervision and a dynamic focal stack loss. In the second stage, the CNN prediction is propagated to the target hologram plane, filtered by a second CNN, double phase encoded, Fourier filtered, and back propagated to the center of the 3D volume to obtain the post-encoding midpoint hologram. No ground truth phase-only hologram is provided, and the CNN is trained to discover an optimal solution with the dynamic focal stack loss between the post-encoding focal stack and the target focal stack and a regularization loss (see Equation (4.24)). The Fourier space amplitude is visualized in the logarithmic scale.	126
4-18 Black modulation test using the double phase encoding.	
Using the per-pixel lookup table calibrated via the proposed approach produces a more uniform image with a deeper black level over the one achieved by the factory-provided global lookup table.	130

4-19 RGB-D and LDI inputs used in this paper.	From top to bottom and left to right: 'Couch' and 'Mansion' from Kim et al.[139], a living room scene from Xiao et al.[258], a forest scene from Padmanaban et al.[200], 'Wanderer' and 'Tree Creature' by Daniel Bystedt, 'PartyTug 6:00AM' by Ian Hubert, 'Flower' from Mildenhall et al.[190], 'Big Buck Bunny' by (© 2008, Blender Foundation), and 'MIT-EECS-Logo' by the authors. 'Couch', 'Mansion', and 'Flower' are real-world captured, and the rest are rendered. 'Wanderer' and 'Tree Creature' are LDIs and the subsequent layers are visualized at the top corners. The yellow boxes mark the regions with obvious depth misalignment or inconsistency, the soft leave edges in the depth map of 'Flower' are not marked.	135
4-20 Performance evaluation of TensorHolo v1, TensorHolo v2-RGB-D, and TensorHolo v2-LDIs.	a Comparison of 3D holograms for computer-rendered scenes. The yellow inset marks the region of interest. b Comparison of 3D holograms for real-world captured scenes.	136
4-21 Performance evaluation of DDPM.	a , Comparison of simulated reconstruction using DDPM, AA-DPM, and BL-DPM for midpoint hologram. For AA-DPM and BL-DPM, two filter strengths are applied: the weaker one results in similar sharpness to DDPM; the stronger one results in an artifact-free image. b , Comparison of simulated reconstruction using DDPM, DDPM (without the pre-filtering CNN), and AA-DPM for target holograms with 6mm, 12mm, and 24mm offset to midpoint hologram. The filter strengths used for the boat and the couch scene are ($\sigma = 1.0, w = 5$) and ($\sigma = 0.8, w = 3$) respectively. They are the minimal strengths that result in artifact-free images for the midpoint hologram. c , Comparison of pre-encoding target hologram phase and its double-phase encoded phase-only hologram predicted by CNNs trained with and without the regularization loss. The hologram plane is offset by 6mm.	137

4-22 Schematic of the experimental holographic display prototype.	
cylindrical lens is used for the aberration-correction experiment (see Fig. 4-	
24). The control box of the laser and Labjack DAQ are not visualized.	139
4-23 Experimental demonstration of holographic 3D projections.	
Two computer-	
rendered (Top/Bottom) and one real-world captured (Middle) scenes are	
displayed and photographed for target holograms located at three different	
distances (marked on the left) to the midpoint hologram. The yellow box	
marks the foreground object of interest, and the blue box marks the back-	
ground object of interest. Readers are encouraged to zoom in to examine	
the image details.	140
4-24 Performance evaluation of vision aberration correction.	
a Top: A holo-	
gram calculated for normal vision. Middle: A hologram calculated using	
the proposed variant of SM-LBM that corrects a synthetic astigmatic vi-	
sion induced by a 200mm cylindrical lens. The calibration method is de-	
tailed in Methods. Bottom: The CNN trained on the astigmatism-corrected	
dataset jointly performs diffraction simulation and aberration correction.	
The PSNR and SSIM of the predicted hologram are marked on the top	
right corner. b Experimentally captured CNN-predicted holograms of a	
natural scene and a test scene without (Left) and with (Right) astigmatism	
correction. The yellow and the blue box marks the foreground and the	
background object of interest, respectively. The in-focus object in the cor-	
rected hologram exhibits horizontal blurs in the uncorrected hologram (i.e.,	
treefrog's tentacles, eye reflections, tree leaves, and test patterns).	142

4-25	HiFiHC pipeline for hologram image and video compression. For image compression, the encoder E encodes one latent code for the hologram's real and imaginary components. The latent code is quantized by Q , entropy coded with side information generated through P , decoded by G , and classified by D . For video compression, E takes an H.265 compressed frame with its associated residual and encodes a latent code only for reconstructing the residual. The reconstructed residual is added back to the H.265 frame.	146
4-26	Comparison of HiFiHC, HEIC, and BPG performance on hologram images. Readers are encouraged to zoom in and examine details. The second and the third row in each label mark the peak signal to noise ratio (PSNR) and structure similarity index (SSIM) for the hologram amplitude (first number) and the refocused DoF image (second number). Source images: PartyTug 6:00AM (left) by Ian Hubert, and Mansion (right) from Kim et al. [139]. More discussion to be added.	150
4-27	Comparison of HiFiHC and H.265 (at lower CRF) performance on hologram videos. Readers are encouraged to zoom in and examine details. In each inset, the top right and bottom left numbers mark the PSNR and SSIM for the refocused DoF image. The second row in the frame label marks the frame type and the bpp of the HiFiHC latent code. Source images: Big Buck Bunny (top) by Blender Foundation, and Horns (bottom) from Mildenhall et al. [190]. The H265 (lower CRF) results use CRF of 15 and 18 for Big Buck Bunny and Horns respectively, both of which yield a similar amount of additional bpp compared to HiFiHC.	151

4-28	Additional comparison of HiFiHC, HEIC, and BPG performance on hologram images. Readers are encouraged to zoom in and examine details. The second and the third row in each label mark the peak signal to noise ratio (PSNR) and structure similarity index (SSIM) for the hologram amplitude (first number) and the refocused DoF image (second number). Source images: Couch (top left) from Kim et al. [139], Mr. Elephant by Glenn Me- lenhorst, Robot (bottom left), and Home (bottom right) scene from Xiao et al. [257]	153
4-29	Additional comparison of HiFiHC and H.265 (at lower CRF) performance on hologram videos. Readers are encouraged to zoom in and examine details. The top and bottom numbers in each inset mark the PSNR and SSIM for the refocused DoF image. The second row in the frame label marks the frame type and the bpp of the HiFiHC latent code. Source images: Orchids (top) from [190], and Market_5 (bottom) from MPI Sintel dataset [34]. The H265 (lower CRF) results in the use CRF of 18 and 19 for Orchids and Market_5, respectively, both of which yield a similar amount of additional bpp compared to the HiFiHC.	154
4-30	Additional comparison of HiFiHC and H.265 (at lower CRF) performance on hologram videos. Readers are encouraged to zoom in and examine details. Each inset's top and bottom numbers mark the PSNR and SSIM for the refocused DoF image. The second row in the frame label marks the frame type and the bpp of the HiFiHC latent code. Source images: Market_6 (top) and Cave_4 (bottom) from MPI Sintel dataset [34]. The H265 (lower CRF) results use CRF of 20 and 20 for Market_6 and Cave_4, respectively, both of which yield a similar amount of additional bpp compared to the HiFiHC.	155

4-31	Performance comparison of hologram image compression (left) and video compression (right) under different offsets between the 3D volume and the hologram. The PSNRs are calculated for the refocused insets in Fig. 2, Fig. 4-28 (for image compression); Fig. 3, Fig. 4-29, Fig. 4-30 (for video compression). For both hologram image and video, the PSNR decreases as the offset increases.	156
4-32	Rate-distortion curves for image and video compression evaluated at the hologram plane and a stack of object planes. For object planes, five layers are evenly sampled through the object space and the mean PSNR is reported. The PSNRs are calculated for the scenes in Fig. 2, Fig. 4-28 (for image compression); Fig. 3, Fig. 4-29, Fig. 4-30, and additional test scenes from Xiao et al. [257] (for video compression).	156
5-1	Holographic display prototype used in assessing HoloMultiColor holograms.	164
5-2	$\times 1.63$ brightness optimized with dynamic scaling (100 ms exposure, source image: Midjourney).	165
5-3	Increasing peak brightness levels with HoloMultiColor. Photographs show that HoloMultiColor can enhance the peak brightness levels of the captures up to $\times 1.8$ without noticeable artifacts or distortions. In contrast, the conventional hologram fails to support beyond $\times 1.0$ (Source: MidJourney, Inc., 140 ms exposure).	166
5-4	Photographs showing HoloMultiColor generating higher brightness beyond $\times 2.0$ (source image: Midjourney, 50 ms exposure).	166
5-5	Increasing peak brightness levels with our multi-color holograms. All photographs are captured at a 140 ms exposure. Our multi-color holograms can enhance the peak brightness levels of the captures up to $\times 1.8$ without artifacts or distortions, whereas conventional holograms fail to support beyond $\times 1.0$ (140 ms exposure, source image: Midjourney).	167

5-6	Comparing red, green, and blue histograms of a target image with conventional and multi-color schemes for varying intensity levels (Same target as Fig. 5-5, $\times 1 - \times 1.8$ intensity, 140 ms - 240 ms exposure).	169
5-7	Using fewer subframes with HoloMultiColor. The first row shows photographs of HoloMultiColor results with a peak brightness increase for a grayscale content (50 ms exposure). The second row demonstrates the color reproduction quality increase for a full-color scene with the increasing number of subframes (200 ms exposure, source image: Midjourney.) . .	170
5-8	Given a target image (right) with $\times 1.0$ brightness and varying projection distances (from left to right), simulations of our multi-color holograms suggest an improvement in color reproduction capabilities when the projection distance prolongs, and may mean using fewer subframes to achieve the same image quality (source image: Midjourney).	172
5-9	3D scenes using our multi-color holograms. Each row shows a multi-plane scene generated by our multi-color scheme with three focus planes. The targeted brightness level is $\times 1.8$ (140 ms exposure, source image: DIV2K [4], Midjourney)	175
6-1	Comparison of the incoherent depth of field rendering. Lee et al. [153] apply binary masking and blending on an RGB-D input to process occlusion and accumulate incoherent wavefronts. Their method fails to conserve total energy (the forefront image is noticeably brighter than the eye/leaf focus), produces an attenuated background at occlusion boundaries due to the loss of background content information, and allows background light to pass through. The proposed method eliminates these artifacts using ray tracing and LDI representation. Source image: "Tree Creatures", reproduced under terms of the CC-BY-NC-SA license, Copyright 2018, by Daniel Bysted, published with the author's permission.	180

6-2	Visualization of fixed pupils and random pupils used in pupil optimization.	
	The fixed pupils are underlaid in faint orange in the visualization of random pupils.	
		182
6-3	Schematic rendering of the experimental setup. Only green laser is visualized. The control boxes of the SLM and laser are omitted.	185
6-4	Comparison of 3D CGH algorithms on experimental results captured under unfiltered mode. The camera focuses are marked by white rectangles (plane 4, 16, 28), and the numbers indicate the PSNR with respect to the target image. Source image: “PartyTug 6:00AM”, reproduced under terms of the CC-BY-SA license, Copyright 2020, by Ian Hubert, published with the author’s permission.	186
6-5	Additional experimental results captured in the unfiltered mode. The captured results are presented in 900 DPI resolution for close-up examination. The “Tree Creature” shows recording planes 9, 17, 28, the “Racing Car” shows recording planes 5, 15, 27.	187
6-6	Additional experimental results captured in the unfiltered mode. The captured results are presented in 900 DPI resolution for close-up examination. Source image (bottom): “Wanderer”, reproduced under terms of the CC-BY-SA license, Copyright 2018, by Daniel Bystedt, published with the author’s permission. The “Mr. Elephant” shows recording planes 4, 15, 29, the “Wanderer” shows recording planes 4, 10, 15.	188
6-7	Comparison of 3D CGH algorithms on experimental results captured at various pupil locations and sizes. The bottom right corners in column 1 images mark the pupil position and size used to capture the results in the respective row. The numbers indicate the PSNR with respect to the target image. Source image: “Mr. Elephant”, reproduced under terms of the CC-BY-ND license, Copyright 2017, by Glenn Melenhorst, published with the author’s permission.	189

6-8	Additional experimental results captured in the pupil-mimicking mode. The captured results are presented in 900 DPI resolution for close-up examination. Source image (top): “The Junk Shop”, reproduced under terms of the CC-BY license, Copyright 2019, by Alex Treviño/ original concept by Anaïs Maamar.	190
6-9	Additional experimental results captured in the pupil-mimicking mode. The captured results are presented in 900 DPI resolution for close-up examination.	191
6-10	Ablation study of pupil-mimicking optimization. Here, we compare mixed pupil optimization with using only random pupils or fixed pupils at the same number of total pupils.	193
6-11	Comparison of simulation and experimental results obtained using coherent integration and incoherent integration.	195
6-12	Comparison of experimentally captured EC-H holograms using different numbers of time-multiplexed frames.	197
6-13	Experimentally captured vertical motion parallax for the “Racing Car”. The horizontal white line and small white bars help perceive the motion parallax.	198
6-14	Experimentally captured horizontal motion parallax. The bounding box on the right side of the image shows the occlusion of the racing track by the foreground free as the viewpoint moves. The left label marks the pupil off-set along the x-axis. Source image: “Racing Car”, reproduced under terms of the CC-BY-SA license, Copyright 2016, by Pokedstudio, published with the author’s permission.	198
6-15	Visualization of exemplary focal stack images and holograms optimized for unfiltered and pupil-mimicking modes for scenes used in this paper.	200
6-16	Simulated reconstruction of the USAF-1951 target under unfiltered and pupil-mimicking modes.	202
6-17	Simulated reconstruction at different propagation distances with numbers indicating PSNR with respect to the target image.	202

6-18	Simulated reconstructions at pupils shifted far away from the eye box center.	203
7-1	An exemplar leather material graph created using substance 3D designer.	
	The graph starts with several generator nodes that provide initial patterns	
	for subsequent filter nodes to manipulate. The graph is internally parti-	
	tioned into subgraphs that create different visual attributes of the leather	
	pattern. It outputs four material maps that define the parameters of an	
	SVBRDF.	209
7-2	The code structure of DiffMat	213
7-3	Overview of MATch’s fully differentiable procedural material design pipeline.	
	Given an input target image (potentially a flash photograph captured using	
	a cellphone), we compute an input texture descriptor (flattened Gram ma-	
	trix) using a pre-trained VGG network. For each procedural material in	
	our library, we train a parameter prediction network by appending fully	
	connected layers to this texture descriptor. We use these models to predict	
	node parameters of the target image for every material graph, and generate	
	the corresponding SVBRDFs using the differentiable material graphs trans-	
	lated by DiffMat. We render images from these SVBRDFs, compute their	
	corresponding texture descriptors and select the top-3 material graphs with	
	the lowest ℓ_1 texture descriptor difference with respect to the input. Finally,	
	we refine the previously predicted node parameters using a gradient-based	
	optimization with our differential node graphs to improve the match with	
	the target image. The material graph (along with the refined node param-	
	eters) that produces the closest matched result is output as the final result.	
	Here, we visualize this pipeline with a real, captured wood material sample.	218

7-4	Prediction and optimization results for synthetic materials. For a majority of synthetic examples, optimization alone is sufficient to reproduce the target materials. Compared to Hu et al. [113] (our reimplementation on our data and SVBRDF model), our network prediction produces more accurate color, fine features, and glossiness with the help of rendering loss. The prediction result is further improved by the optimization step to precisely match the target material.	223
7-5	Optimization results for real-world materials. Top: At each row, we show multiple real-world materials matched by one single procedural material graph using the same default initial parameters. The * symbol marks the examples that are optimized against a target material of a different material type. Middle: Additional examples on unmatched material and graphs. Bottom: Additional examples on matched material and graphs.	225
7-6	Prediction plus optimization results for real-world materials. Left: Speed of convergence comparison between optimizing default and predicted node parameters. Right: Comparison of optimization results on challenging target materials. Optimizing from the default parameters stuck in the local minimum, while optimizing from the predicted parameters produces a significantly closer reproduction.	226
7-7	Comparison of our method to Hu et al. [113] network prediction (without non-differentiable style transfer step) for real-world captured materials. For our results, we show optimization over exposed parameter only, exposed parameters with the rest of independent node parameters, and all node parameters. Our method overall produces a better match through optimizing a significantly larger design space and iteratively update the appearance to match the target. In contrast, the prediction by Hu et al. network is one-shot and can't not be iterative refined through back-propagation.	226

7-8	Comparison of our method to single-image per-pixel SVBRDF capture methods of Deschaintre et al. [60] and Gao et al. [80] illustrate the key difference to our method. While the per-pixel methods are able to produce texture patterns more closely aligned to the exact target photo, they suffer from various artifacts such as blurring and color fringing. Our method, predicting node graph parameters instead of pixels directly, is much more robust to these problems, though of course at the cost of not matching the target texture patterns exactly.	228
7-9	Material editing. In each row, we show an optimized real-world material and variants generated by manually adjusting the optimized node parameters. Note that all the variants are generated from the same group of noise patterns, yet display a wide range of appearances.	229
7-10	High resolution procedural materials optimized for real-world photograph inputs. Readers are encouraged to zoom in and examine details.	230
7-11	Results of exposed parameter transfer between material graphs. The labeled arrow between each image pair indicates the name of the exposed parameter. We direct readers to the Supplementary Videos for transition animations when varying exposed parameters from 0 to 1.	231
7-12	Failure cases due to unmodeled objects (substrate behind the metal grate), limited expressiveness of overly simple graph structures (subgraphs that produce metallic of the metal grate, normal of the scratched metal, and the bed sheet example), and inability to optimize the generator nodes to match the input pattern (albedo of the scratched metal, the bed sheet example).	233
8-1	<i>Left:</i> A photograph of one of our printed sample patches. Each color square is 1 mm $\times$ 1 mm. <i>Right:</i> Our spectral acquisition setup.	238

8-2	Our ink library. <i>Left</i> : The pigments and concentrations used in our inks, arranged in the order from top layer to bottom. <i>Right</i> : The spectra of saturated color of our inks printed with 30 layers. The color of each line corresponds to the color of an ink. The higher white line represents the Transparent White and the lower represents the Opaque White.	241
8-3	Illustration of our network structure. $\mathbf{F}$ denotes our spectral prediction model (§8.4.1), which is used to estimate the spectrum of the predicted layout by our layout prediction model $\mathbf{B}$ (§8.4.2) and provides the means to measure the quality of it. Q represents our soft quantization layer (§8.4.3). ψ denotes the spectrum and ξ the layer-layout vector. Both $\mathbf{F}$ and $\mathbf{B}$ are implemented as fully-connected feed-forward neural networks.	244
8-4	(a) Comparison of test spectra from our contoning ink stack dataset predicted by our spectral prediction model $\mathbf{F}$ and by the 4-flux method. (b) Projection of predicted spectra onto CIELAB space (under D65) with the dot volume proportional to the spectral error E_{spec}	251
8-5	Comparison of our 10-ink gamut with two other printing methods and sRGB (under D65). The figure shows a^*b^* slices at different L^* values. . .	253
8-6	Comparisons of different quantization methods. (a) ground truth; (b) spectral vector error diffusion with brute-force enumeration; (c) spectral vector error diffusion with nearest-integer rounding; (d) per-pixel brute-force enumeration; (e) per-pixel nearest-integer rounding; (f) per-pixel nearest-integer rounding <i>without</i> using our soft quantization layer. The results in (b–e) were created using the network <i>with</i> the soft quantization layer. Each row at bottom corresponds to a rectangular region in the painting at top. Paintings ©Azadeh Asadi.	254
8-7	Comparison of historical pigment spectra reproduced by the proposed neural network and the nearest neighbor search. The white dots mark the challenging spectra that are difficult for our ink set.	256

8-8	Reproduced paintings using our method. We show 4x2 sets of our printed reproductions. For each set, we show our printed reproduction with CMYK inks (top row), the original painting (middle row), and our printed reproductions with 10 inks (bottom row). Each painting is lit by four different light sources (halogen (2750K), Phillips Hue LED (Warm), fluorescent (5000K), and Phillips Hue LED (Cool)), which are shown so ordered. The spectrum of each light is illustrated in the supplementary material Figure A.1. Paintings ©Azadeh Asadi.	259
8-9	Simulated comparisons under D65 illumination of our method to the CMYK contoning. Our reproduction (a) of three paintings are compared against the original painting (b) and the CMYK contining method (c) of [14]. Our method provides consistently more faithful reproductions to the original. Paintings ©Azadeh Asadi.	260
8-10	Comparisons of the printed results and measured spectral reflectance between <i>our</i> CMYK reproduction, 10-ink reproduction and the original painting. In the lily example (top), the reconstruction of green and purple hues are extensively examined, which are known to be difficult for CMYK. In the bottom example, we examine a wide varity of both saturated and desaturated colors. In both results, 10 inks result consistently outperforms the CMYK both perceptually and in terms of spectral RMSE. Paintings ©Azadeh Asadi.	262

List of Tables

4.1	Runtime performance of TensorHolo V2 RGB-D CNNs on GPU and iPhone 13 Pro. The numbers are reported for the CGH synthesis networks with a different number of convolution layers. When the target hologram plane offsets from the midpoint hologram, the addition of the pre-filtering CNN adds 4.8 ms for an TITAN RTX, and 60 ms for an iPhone 13 Pro.	133
4.2	Methods comparison on the validation set. The unbracketed result is calculated for the amplitude map of the predicted hologram, the bracketed result is calculated by averaging the results of a focal stack with 15 most frequent depths and 5 random depths for each image.	134
5.1	Image quality evaluation of conventional and HoloMultiColor schemes for various levels of peak brightness. Blue color indicates values for HoloMultiColor.	173
5.2	Ablation Study for HoloMultiColor. We remove only one component (not multiple) from the HoloMultiColor pipeline at each study and report image quality metrics. Without the “-” component refers to the complete model. .	174
8.1	The accuracy of different spectral reflectance prediction models, in terms of the spectral error E_{spec} (Eq. 8.1) and the CIEDE2000 (ΔE_{00}) [174] under three illuminants. We report the mean, standard deviation, median, and maximum. The lowest errors are denoted in boldface.	250

8.2 The accuracy of our layer prediction model trained with different objec-

tives, measured in the spectrum error E_{spec} (Eq. 8.1) and the CIEDE2000

ΔE_{00} under three illuminants. We report the mean and standard deviation.

The lowest errors are shown boldface and the second lowest underlined. . . 253

Chapter 1

Introduction

In 1950, Alan Turing introduced the Turing test to evaluate a computer's capacity to convincingly mimic human behavior. Similarly, Meta introduced the term visual Turing test in 2022, alongside leading academic teams, to assess whether visuals presented in VR headsets can be distinguished from reality. Unfortunately, this test remains beyond the reach of current VR technology. Despite VR's ability to immerse users in virtual environments with remarkable realism, it has yet to prompt observers to question the authenticity of what they see, pondering whether it's real or virtual. Achieving this goal requires solving challenges spans over multiple disciplines: optics, graphics, sensors, compute, wireless communication, and many to name. For that reason, National Academy of Engineering (NAE) has identified "enhancing VR" as one of the fourteen grand challenges for engineering in the 21st century. Augmented Reality (AR) represents a form of VR where users encounter synthetic virtual world imagery superimposed and integrated with their real-world surroundings. It's widely anticipated that VR, alongside AR, will emerge as the next mass-market platform, potentially supplanting PCs, laptops, and mobile phones. Ultimately, we envision them to become persistent interface that travels with users, providing spatial and context-aware 3D information that is hardly distinguishable from the real world. To take a step towards passing the visual Turing test and providing an ideal immersive experience, this thesis focuses on two main areas: near-eye displays and inverse material modeling & capture. Below, we identify some key challenges in these two areas.

Near-eye displays serves as the direct interface between users and virtual content within

a headset, primarily determining the comfort, immersion, realism of the visual experience. Over the last few years, microdisplays for near-eye applications have been improving at a rapid pace, with significantly improved resolution, color gamut, brightness, and refresh rate. However, many crucial but under-investigated aspects, some of which we discuss below, still lack an ideal solution, setting roadblocks for the ultimate goal of passing the visual Turing test.

Vergence-Accommodation Conflict (VAC) VAC is a phenomenon that occurs in current stereoscopic AR/VR headsets. It arises due to the disparity between the depth cues provided by the eyes' vergence (i.e., the inward or outward rotation of the eyeballs to focus on objects at different distances) and accommodation (i.e., the adjustment of the lens inside the eye to focus on objects at different distances). In natural viewing situations, the eyes adjust both their vergence and accommodation simultaneously to maintain clear and single binocular vision. However, in stereoscopic AR/VR headsets, the images created by 2D microdisplays have fixed focal distances, causing a discrepancy between the depth cues provided by the vergence (which is based on the disparity between the left and right images) and accommodation (which is based on the fixed focal distance of the display). As a result, users may experience eyestrain because their eyes are attempting to converge or diverge to focus at a different distance than the display is presenting. This conflict can further lead to visual discomfort, fatigue, and even headaches, especially during prolonged use [107, 150, 228].

Form Factor Current AR/VR headsets are still bulky, heavy, and uncomfortable to wear for extended periods. The weight and size of the headset can cause discomfort, neck strain, and headaches, especially during prolonged use. A headset with an eyeglasses-like (or perhaps even a contact lens-like) form factor appears the most appealing: it covers much of the viewer's field of view, is non-encumbering and already used by much of the population, is socially acceptable, and can easily be taken on and off. However, existing consumer products such as XREAL Light, Google Glass, North Focals, and Vuzix Blade offer very limited field of view, resolution, compelling image quality, and absence of support for

accommodation cues and vision correction capability. In research community, attempts have been made to create headsets of such form factors using holographic optics, directional backlight, and polarization-based optical folding, yet with wide field-of-view, such design remains challenging to incorporate accommodation cues and vision correction capability [176]. Lanman et al. [151] constructed a light field near-eye display with accommodation support, but at a significantly lowered spatial resolution due to spatial angular resolution trade-off inherited this technology.

Vision Aberration For users who rely on corrective lenses in their daily lives, such as those who are near-sighted or far-sighted, using existing headsets still necessitates additional corrective lenses. This not only increases the weight, cost, form factor, and complexity of the headset design but also hinders the sharing of devices among users with varying vision conditions and often results in poor fit on the head. Furthermore, when creating eyeglasses-like form factor headsets, where no additional corrective lenses can be easily attached, correcting vision aberration would require complete customization of the display optics, which is costly and impractical. Even with the use of corrective lenses, they may introduce optical distortions or aberrations that affect the clarity and quality of the virtual experience.

Dynamic holographic displays offer a promising solution to collectively these challenges by in theory providing a wide field of view, incorporating accommodation cues, allowing for glass-free vision correction, and potentially achieving a form factor similar to ordinary eyeglasses, ensuring comfortable wear throughout the day. However, the widespread adoption of holographic displays has been impeded by issues such as low image quality and substantial computational costs, especially at higher resolutions, and large form factor required by optical filtering. Consequently, there is an urgent need for the development of computational methods and systems specifically tailored to solve these challenges for holographic display to enable realistic 3D visualization in real-time and slim form factor. Additional design parameters such as energy efficiency, brightness, and ergonomics should also be carefully considered and jointly optimized.

Reproducing real scenes in virtual environments stands as another key task in AR/VR

and more broadly for the graphics industry. Achieving an immersive and convincing experience hinges on accurately recreating the scene’s appearance, which has a direct impact on the perceived realism. This task is inherently challenging due to the intricate interplay between light, material reflectance, and geometry in shaping the appearance of real-world objects. While techniques such as multiview stereo algorithms or depth scanners can capture scene geometry, recovering a numerical representation of materials from photographs or measurements of a surface is at the heart of inverse material modeling & capture algorithms.

Traditionally, achieving accurate material appearance capture entails densely sampling light and view directions within a controlled acquisition environment. This typically involves sophisticated setups such as robot arms, spherical gantries, or light domes. Such advanced configurations are necessary because various combinations of reflectances, geometries, or lighting conditions can result in the same observed image. However, these setups are costly, time-intensive, and impractical for casual users. Inverse material modeling and capture aim to overcome these challenges by enabling users to capture material appearance from a single or a few images, without the need for specialized equipment or expertise. This is often achieved by incorporating prior assumptions about the space of plausible material solutions to guide reconstruction and facilitate reduced measurement. Recently, deep learning has emerged as a powerful approach for automatically learning effective priors from data. Methods have been developed to estimate spatially-varying bidirectional reflectance functions (SVBRDFs) from just a few or even a single picture of the material. However, these methods produce digitized materials in raw pixels that are difficult to scale in resolution, ensure seamless tiling, and create meaningful material variations through editing.

Alternatively, procedural programming offers a parametric representation for users to customize their materials. Modern material authoring tools (Blender, Substance 3D Designer, etc.) implement a graph-like interface where the output of this design process is a procedural material which users can edit freely, synthesize to arbitrary spatial resolution, and store compactly. However, even though procedural material modeling is powerful and user-friendly, it remains time-consuming to hand-craft a high quality material even for ex-

pert artists from a reference material photograph. While it is possible to consider such a program as a custom form of neural networks with human-designed operators and parameters, there lacks a computational framework and software infrastructure to leverage it as a prior and differentially optimize it to guide the inverse material modeling and capture process. Method that solely treat it as a prediction problem using deep neural networks fail to produce and generalize to complex material photographs [114]. As a result, obtaining a high-quality procedural material from a single image remains a challenging task prior to the works discussed in this thesis.

Lastly, we consider an extrapolation of the visual Turing test to physical reproduction. The particular example this thesis focuses on is the reproduction of paintings. The ability to accurately reproduce paintings is crucial for culture preservation, education, and research. However, traditional color reproduction of paintings carried out by 2D printers fails to faithfully capture the wide gamut and spectral reflectance of the original paintings, leading to unfaithful reproduction with diminished value for many of the applications mentioned above. Recent development of multi-material 3D printers offers a potential solution to this problem by enabling physical reproduction with significantly expanded number of color primaries and flexibility to explore 3D deposition of materials. Nonetheless, creating a spectral-aware color reproduction pipeline capable of accurately reproducing the spectral reflectance of paintings remains an unresolved challenge prior to the work discussed in this thesis.

1.1 Thesis Overview

In this thesis, we introduce our solutions for the aforementioned challenges. Chapter 2 overviews the prior works related to the several topics covered in this dissertation. Chapter 3 describes a novel implementation of light field near-eye 3D displays based on the holography principle [222]. The system comprises a specialized light field rendering algorithm tailored for hologram calculation, meticulously designed to eliminate aliasing in both oversampling, undersampling, and avoid geometry clipping. Through rigorous theoretical analysis, we established the limitations regarding aliasing, sampling, spatial resolution,

and design parameters critical for utilizing holograms to faithfully reproduce light fields at resolution much beyond prior instantiations. A see-through full-color near-eye display prototype is demonstrated with complex modulation capabilities, supporting continuous focus cues and intra-pupil occlusion. Chapter 4 presents the first deep learning-based techniques for 3D hologram generation, first-time achieving real-time computational speeds (i.e., 60Hz for 1080p content) for future AR/VR applications with high image quality [224, 225]. Alongside, we demonstrate interactive computation on mobile devices, the ability to embed vision correction into trained network, as well as neural compression algorithms specifically designed for holographic data. Chapter 5 introduces the first optimization-based framework for generating multi-color holograms capable of operating under simultaneous illumination by multiple color primaries with co-optimized intensities [137]. Without requiring alterations to display hardware, this method enables a more efficient utilization of each color primary, enhancing perceived brightness by dedicating additional "on" time in every subframe. Chapter 6 introduces the first ergonomic-centric hologram optimization method aimed at systematically enhancing the realism, comfort, and immersion of holographic displays—areas that were only addressed individually in previous works [227]. Our experimental results showcase a simultaneous improvement in perceived experimental image quality and expansion in eyebox size without necessity a Fourier filter, paving the way for reducing display form factor to eyeglass-like. Chapter 7 devises a by-example inverse procedural material modeling system for capturing SVBRDF of a near-planar surface from a single flash image without leveraging user expertise. The system comprises the first differentiable procedural material modeling library, designed to translate Adobe Substance 3D materials into differentiable computational graphs; as well as, a gradient-based optimization framework that optimizes the procedural graph parameters to minimize the difference between the rendered and captured images [223, 159]. Chapter 8 introduces a deep-learning-based data-driven workflow for accurate physical reproduction of painting appearances, capturing their spectral reflectance with precision through the utilization of a multi-material 3D printer [221]. Chapter 9 concludes the dissertation with a discussion on the contribution and future work.

Chapter 2

Related Works

2.1 Computational Near-Eye 3D Displays

Here we provide a brief overview of computational near-eye 3D displays that aimed to support accommodation cues, slim form factor, and vision aberration correction. A more comprehensive review of different optical designs can be found in [119, 146, 35, 41].

Holographic Displays Holographic displays create three-dimensional images by diffracting coherent light waves to form interference patterns that match target intensities [22]. Computer-generated holography (CGH), the process of obtaining holograms, numerically simulates the physical process of light diffraction and interference as if recording an optical hologram. The typical simulation process involves computing the interference pattern of object waves from a 3D scene modeled as self-illuminating point sources with a reference wave [254, 175, 45], where the free space propagation can be modeled using Rayleigh-Sommerfeld diffraction integrals (see Sec. 3.1.2 for more detailed discussion). It is a time-consuming process for high resolution holograms that requires minutes of computation time on a CPU and many seconds on a consumer-level GPU for a single frame [272, 175]. Nevertheless, the process allows integration of computational vision correction kernels into the wavefront calculation, enabling the display to correct for the viewer’s vision aberrations without necessitating corrective glasses [175].

Displaying a CGH necessitates spatial light modulators (SLM) to modulate the phase

and/or amplitude of light waves, yet a single SLM can only modulate one of the properties, and typically a phase-only modulator is preferred due to its higher diffraction efficiency. When a single SLM is used, phase-only constraints need to be imposed on the hologram. Several direct methods enforce phase-only constraints, such as double phase encoding [111, 175] and iterative methods like Gerchberg-Saxton [84, 157] or the hybrid input-output methods [75, 21]. More recently, gradient-based optimization methods [38, 274, 49, 48] and deep-learning-based methods [204, 39, 110, 109] for phase retrieval have gained interest due to their promise of producing high-quality holograms. While direct methods are faster, they generally offer lower image quality and/or reduced brightness compared to iterative methods. To optimize computational complexity or to better handle complex physical effects (spatially-varying appearance, shading, occlusion), researchers have also explored other input scene representations such as polygon [156, 181, 182], light field [170, 261, 200, 236], or layer primitives [270, 43] with associated physical simulation algorithms for CGH generation. Meanwhile, leveraging holographic optical elements (HOEs) or waveguides can simplify the optical design, and enable eyeglass-like form factor holographic display with wide field of views [175, 130]. These designs typically require a tradeoff between field of view, eyebox size, and image quality, where we show better computational algorithms could mitigate their tradeoffs.

Light Field Displays By taking a geometric optics approximation of wavefront, light field displays create 3D images by reproducing intensity and direction of light rays emitted from an underlying 3D scene. Lanman and Luebke [151] first applied integral imaging [165] principle create VR near-eye light field displays using microlens array, with an AR variant later developed by Hua and Javidi [120]. These prototypes depict near-correct retinal blur and intra-pupil parallax, but exhibit low resolution due to explicit tradeoff between angular and spatial resolution. Maimone et al. [177] and Huang et al. [121] introduced computational near-eye light field displays based on amplitude-only SLM stacks with light field factorization, achieving image resolutions that far exceed microlens-based approaches. However, the multi-SLM stack approach ultimately confronts practical resolution limits due to diffraction and compression artifacts. In Chapter 3, we demonstrate

light field reproduction based on holographic principle that overcomes these limitations by leveraging the diffraction and interference. Huang et al. [123] also demonstrated that vision correction can be encoded into a light field display by predistorting the image presented, yet the capability was restricted to correct 2D images unlike the 3D correction that was demonstrated for holographic displays.

Multifocal and Varifocal Displays Varifocal displays dynamically adjust the focal plane of virtual images and computationally blur the scene objects not at the current depth based on calculated depth of field. Such systems require an eye tracker to determine the viewer's distance and a variable focusing element to adjust focal distance. Shiwa et al. [230] initially demonstrated this concept using actuated lenses on an optical bench. Sugihara et al. [239] developed the first varifocal HMD, where the display itself translated instead of a lens. Subsequent advancements by Liu et al. [166] and Konrad et al. [144] showcased varifocal displays utilizing electronically tunable lenses. More recently, Dunn et al. [66] and Padmanaban et al. [199] introduced varifocal displays integrated with eye tracking capabilities to facilitate gaze-contingent depth of field rendering. However, it's important to note that varifocal displays still provide a single focal plane at any given moment, and computational depth of field cannot fully replicate natural depth of field cues.

Multifocal displays use multiple display planes to generate approximate focus cues without eye tracking. Neil et al [76] demonstrated the first multi-focal near-eye display, with analysis by Rolland et al. [212] showing 2.0 diopter depth of field requires up to 27 planes. Such systems were typically constructed with ferroelectric liquid crystal (FLC) SLMs [76, 169] and by time-multiplexing with high-speed switchable lenses [166]. Akeley et al. [8] demonstrated this concept on a large format display, while Hu and Hua [112] extended the idea to near-eye displays by using deformable mirrors to support high resolution imagery. Matsuda et al. [180] deformed a plane to a non-planar surface using a phase-only spatial light modulator to enable depiction of more accurate depth of field with fewer multiplexed frames, yet suffers from the challenging of reproducing sharp depth discontinuities and color reproduction. Unlike light field or holographic displays, multifocal displays are limited in capability to produce a large number of focal planes and they often

use more complex optical assemblies that are difficult to miniaturize and extend to wide field of views.

2.2 Material Acquisition and Modeling

Materials define the way the light is scattered and absorbed on and within geometries, and is commonly represented by the Spatially-Varying Bidirectional Reflectance Distribution Function (SVBRDF) which defines for each incident lighting angle and outgoing view angle the amount of energy that is reflected for each considered wavelength at a spatial location (see Sec. 7.1.1 for detailed background discussion and mathematical treatment). Despite advancements in material authoring tools, achieving realism remains challenging, even for skilled artists. One common challenge is faithfully replicating real-world physical materials. Below, we briefly explore methods for acquiring and digitally modeling materials.

Material Capture with Specialized Hardware Physically accurate SVBRDF capture requires dense spatial and angular sampling to disambiguate material parameters. Achieving adequate sampling at large scales often necessitates a specialized captured environment that includes multiple lights and cameras. Many prior works have devised custom capture setups, including the use of a robot arm [56], matrix light pattern (LCD, DLP projector) [6], spherical gantry [152, 267, 108], light dome [87, 86, 196], light cube [134], spherical harmonic illumination [248, 85], LED array [252, 12, 91], or moving linear light source [208, 81]. While these methods generally exhibit satisfactory reconstruction accuracy, the entailed hardware costs, human labor in environment setup, operation complexity, and acquisition time can be prohibitive for amateur users and casual purposes.

Prior-based Mobile Material Capture The rapid advancement in cell phone imaging has ushered in a new era of lightweight mobile appearance capture solely based on casually captured flash photographs. Aittala et al. [7] first demonstrated such a system using a pair of flash/no-flash photographs, while a contemporary work by Riviere et al. [211] used

a flash video captured over a hemisphere. However, the scarcity of captured views and lighting variations in these methods demand domain-specific priors to constrain the results within a plausible solution space. These priors may require the results to be a linear combination of certain basis materials [283, 255, 208, 124, 259, 195], exhibit self-similarity (containing stationary spatial structures) [7], be illuminated under sparse incident lighting [64], or exist within a low-dimensional manifold [65]. Unfortunately, these assumptions do not always hold true for various material families, which often compromises the complexity and fidelity of the reconstructed SVBRDFs.

Learning-based Mobile Material Capture The revolutionary knowledge distillation and generalization capabilities of convolutional neural networks (CNNs) make them an attractive alternative to hand-crafted priors. Li et al. [162] trained a CNN-based SVBRDF predictor utilizing a small labeled dataset of measured SVBRDFs with self-augmentation. Ye et al. [264] extended this method to enable training with unlabelled data. Later on, the introduction of differentiable rendering layers and large-scale synthetic SVBRDF datasets allowed for per-pixel albedo, normal, and roughness predictions [60, 61, 163], as well as the joint recovery of 3D shape [164]. Alternatively, CNNs were trained to generate a low-dimensional latent space as a neural prior, enabling the optimization of more plausible SVBRDFs [80, 101, 103]. Zhou et al. [282] employed adversarial training to enhance the realism of the reconstructed SVBRDFs. Guo et al. [98] introduced highlight-aware convolution to handle saturated specular highlights.

Inverse Procedural Material Capture Another category of lightweight material capture optimizes a procedural appearance model to match materials against user-specified examples. Early attempts used hand-crafted templates and rules to extract structural and local information from the source examples [154, 63, 155, 67]. The effectiveness of these methods primarily depends on the quality of the templates, and their applications are often restricted to certain material categories. For more stochastic and isotropic texture examples, procedural noises are employed to directly match their appearances by optimizing the power spectrum in both the image and frequency domains [79, 149, 99]. Example-based

noises such as local random phase noises [88], Texton noise [78], and histogram-preserving blending operator [102] were proposed to accommodate more appearance variations. Hu et al. [116] expands upon this concept by modeling procedural noises using hierarchical, multi-frequency texture decomposition. However, the expressiveness and general usability of the resulting materials are limited due to an insufficient set of noise patterns, filters, and decomposition models [116].

The advances in node-based procedural material authoring tools (e.g., Adobe Substance 3D Designer, Blender) facilitated the creation of complex procedural materials. Simultaneously, these tools fostered an online community where artists can freely share and reuse high-quality assets. Such a phenomenon has inspired a new line of methods that leverage artist-designed production-grade graphs to match photographed material appearances. In the inaugural work by Hu et al. [113], it predicts the exposed parameters (artist-defined high-level meta-parameters) in a material graph using neural networks. In Chapter 7, we present the first differentiable procedural material modeling library (DiffMat) for S3D and demonstrate its capability to optimize both exposed and low-level node parameters for procedural material capture from a single flash photograph [223, 159]. This library has been used in a variety of follow-up applications, including material capture from indoor 3D photos [265, 262] and unconditional/conditional material graph generation [97, 115].

2.3 3D Printing for Appearance Reproduction

3D printing, also known as additive manufacturing, has revolutionized numerous industries by enabling the creation of complex three-dimensional objects directly from digital designs. While its applications span from prototyping to production, one particularly intriguing facet is its capability for appearance reproduction. In this context, 3D printing serves as a powerful tool for replicating not just the geometric form of an object, but also its surface texture, color, and finish. Whether recreating historical artifacts, architectural models, or intricate works of art, the ability of 3D printing to faithfully reproduce appearance opens up new avenues for preservation, replication, and customization. In Chapter 8, we present a novel 3D printing system that fabricates the spectral color of painting facsim-

iles faithfully to the original. In the following, we briefly review the state-of-the-art color 3D printing for appearance reproduction.

Color 3D Printing While the use of multiple, custom inks is prevalent in 2D printing, 3D printing has recently begun to accommodate color printing. Although the effort to introduce multiple colors in 3D printing has been started with the fused deposition modeling (FDM) technology [207, 104], the best quality for fine art reproduction is currently achieved by resin-based inkjet 3D printers. Brunton et al. [32] introduced an error diffusion algorithm for the surfaces of 3D printed geometry. In order to eliminate the halftoning artifacts, Babaei et al. [14] proposed a *contoning* scheme in which different inks are layered to form the print; our printing workflow builds on this idea. Recently, Elek et al. [69] proposed a color reproduction workflow that preserves the texture by simulating the crosstalk between neighboring voxels using Monte Carlo path tracing. It is also possible to incorporate translucency in color printing pipelines, owing to clear resins, and thereby creating spatially-varying translucency and color using a 3D printer [31]. Yet another approach to decorating a surface is to transfer a previously printed texture to the 3D object, using methods such as hydrographic printing [278, 201] and thermoforming [218, 277]. All these methods, however, focus on colorimetric reproduction using CMYK inks. Compared to them, our work in Chapter 8 aims at spectral reproduction and uses a significantly larger number of inks to achieve a larger color gamut.

Chapter 3

Light Field-based Holographic Displays

A light field (LF) is a vector field that describes the intensity and direction of light rays flowing through every point in space. When the wavelength of light is very small compared to the objects which light is passing, a LF can accurately represent the underlying scene appearance, enabling the rendering of arbitrary viewpoint with control of focus and depth of field. In computer graphics, LFs are critical tools for making static 3D prints and dynamic 3D displays. To date, great success has been achieved in demonstrating decent image qualities and portability of such displays for AR and VR applications as discussed in Chapter 2, yet their spatio-angular resolution is fundamentally limited in current two implementations. Specifically, microlens-based design that operates on integral imaging principle necessitate a direct trade-off between spatial and angular resolution. Multilayer-LCD-based design suffers a fundamental upperbound on the spatial and angular resolution as LF's ray-based modeling deviates from reality due to diffraction when the pixel pitch becomes comparable to the wavelength of light. The limitation of both designs prevents creation of an ideal light field display with high angular and spatial resolution.

Holography offers a solution to this challenge by elevating the ray model to the wave model and directly utilizing diffraction to form 3D images. As straightforward as it may sound, conversion from a light field to a hologram indeed requires careful deliberation to achieve optimal image quality. In this chapter, we first review the background and connection of light field and holography. We then propose a method to efficiently convert light field to holograms that avoids aliasing in both over-sampling, under-sampling and geom-

etry clipping found in previous work. We analyze theoretical limits on sampling, spatial resolution, and discuss the trade-off space for designing light field near-eye displays with significantly higher resolution using CGH. Finally, we build a complex-modulation see-through color display prototype supporting continuous focus cues and an expanded field of view to verify our rendering and calculation results.

3.1 Preliminary

3.1.1 Light Field

A light field models the discrete ray space-angle distribution in geometric optics and can be efficiently rendered using graphics pipelines. Mathematically, a 4D light field L can be represented as a function of two spatial dimensions (x, y) and two angular dimensions (θ, ϕ) , as well as wavelength λ . It can be denoted as $L(x, y, \theta, \phi, \lambda)$, where the function specifies the radiance of light passing through a point (x, y) from direction (θ, ϕ) and at wavelength λ . In practice, λ is often interpreted in terms of RGB color channels. The light field can be captured by a camera array, a microlens array, or a plenoptic camera, and can be rendered to generate images with arbitrary viewpoint, focus, and depth of field.

3.1.2 Computer Generated Holography (CGH)

CGH simulates the physical processes of an optical hologram recording, and reconstruction, using numerical methods. The goal is turn an illumination wave E_{ref} into a desired object wave E_{obj} . Here, we review the method to compute a hologram for scenes modeled as point light emitters, which is the most commonly used. Given a scene modeled as 3D point cloud, the object wave at the hologram plane can be modeled as a superposition of spherical waves from point emitters at the location of object points and propagating towards the hologram plane where the SLM is located. Let $\mathbf{o} \in \mathbb{R}^3$ be (the location of) an object point, $S_{\mathbf{o}}$ be the set of SLM pixels within the extent of $\mathbf{o}$'s sub-hologram, $\mathbf{p} \in \mathbb{R}^3$ be (the

location of) an SLM pixel within S_o , the wavefront contributed from $\mathbf{o}$ to $\mathbf{p}$ is given by

$$h_o(\mathbf{p}) = \frac{a_o}{w_o} e^{i\left(\phi_o + \frac{2\pi\|\mathbf{p}-\mathbf{o}\|_2}{\lambda}\right)} \quad (3.1)$$

where a_o is the amplitude associated with $\mathbf{o}$, $w_o = \sqrt{a_o^2 / \sum_{j \in S_{\text{slm}}} [j \in S_o]}$ is an amplitude attenuation factor for energy conservation ($[\cdot]$ denotes Iverson bracket), and ϕ_o is the initial phase associated with $\mathbf{o}$. The existence of S_o arises from that fact that the spatial frequency of the object wave casted on the SLM plane increases from the normal incident angle to the oblique angles, due to increasing variation phase variation as a consequence of optical path difference change. Due to the finite spatial frequency of the SLM, phase alteration beyond the maximum spatial frequency of the SLM cannot be reproduced and is thus clipped. The object wave is then calculated by summing sub-holograms contributed from all object points within the scene:

$$E_{\text{obj}}(\mathbf{p}) = \sum_{j \in S_p} h_j(\mathbf{p}) \quad (3.2)$$

Note that the resulting object wave is complex-valued.

A CGH changes the wavefront multiplicatively. For CGH illuminated by a *plane wave* (collimated beam) with a unit constant amplitude, the CGH to display is exactly E_{obj} . For more general illumination pattern, the CGH to display is given by $E_{\text{obj}}E_{\text{ref}}^*$, where $*$ denote complex conjugate. In such case, the maximum half-cone diffraction angle between the incident angle θ_{in} of light and the outgoing angle θ_{out} is characterized by the grating equation:

$$\frac{\lambda}{2\Delta p} = |\sin \theta_{\text{in}} - \sin \theta_{\text{out}}|, \quad (3.3)$$

where Δp is the SLM pixel pitch size. A typical $8\mu\text{m}$ pitch SLM only maximally deflects 3.81° on a 532nm green light, whereas a traditional optically-recorded hologram with nanometer-scale diffracting pattern can achieve a maximum angle greater than 100° . The small diffraction angle of the SLM impacts the practical application to near eye displays under collimated plane wave illumination commonly used for holograms, as shown in Fig. [3-1](#). To observe the full extent of a SLM of diameter w_s , the minimal distance d_e^{min} from the eye to the SLM, called *eye relief*, is determined by the marginal rays deflected

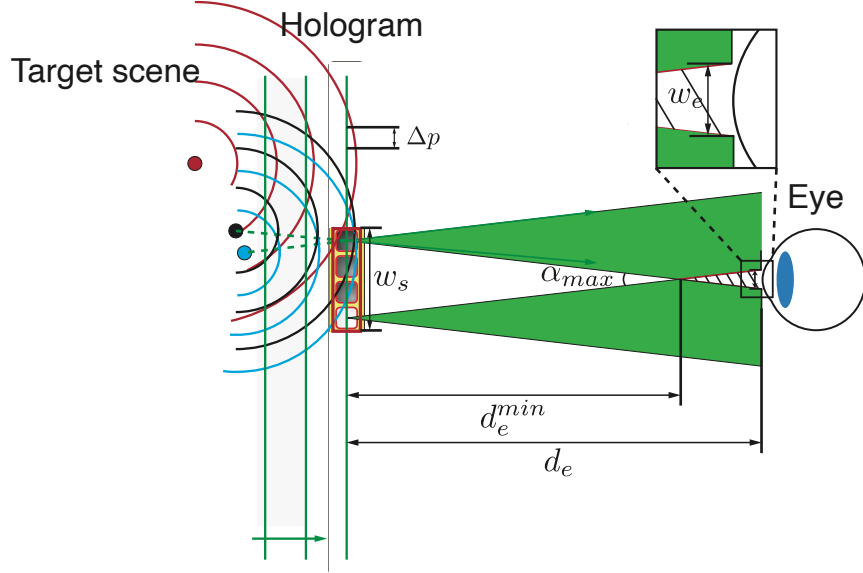


Figure 3-1: Plane wave illuminated holographic display. Each display pixel diffracts wavefront into a spatially-uniform light cone. Rays are back-traced for each point emitter within the diffraction cone. The viewer can only observe the full extent of display behind the intersection of two marginal cones diffracted from the edge of the display.

from the edges of the SLM, and is given by:

$$d_e^{min} = \frac{w_s}{2 \tan \left(\sin^{-1} \left(\frac{\lambda}{2\Delta p} \right) \right)}. \quad (3.4)$$

The maximum angle α_{max} subtended by the visible SLM gives the maximum field of view (FoV):

$$\alpha_{max} = 2 \tan^{-1} \left(\frac{w_s}{2d_e^{min}} \right) = 2 \sin^{-1} \left(\frac{\lambda}{2\Delta p} \right). \quad (3.5)$$

The red region allowing the eye to move freely and still be able to see the entire hologram is called the *eye box*. With eye relief $d_e > d_e^{min}$, the eye box can be obtained as

$$w_e = \frac{d_e - d_e^{min}}{d_e^{min}} w_s, \quad (3.6)$$

The prescribed $8\mu m$ pitch SLM with 1080p resolution requires a minimum $265mm$ eye relief and allows only 3.81° field of view. Further, Equation (3.5) shows that the FoV does not scale with the SLM diameter. This suggests that CGH under plane wave is not a good candidate for a near-eye display. In Sec. 3.2.2 we will show that using a spherical

illumination beam overcomes these difficulties.

3.1.3 Connecting Light Field and CGH with Wigner Distribution Functions

Considering a 2D light field $L(x, \theta, \lambda)$, the angle θ corresponds to the normal direction of the wavefront and under paraxial approximation, it can be related to the spatial-frequency $u = \theta/\lambda$ of a Fourier transformed scalar wave function $F(E(x))$, also denoted as the angular spectrum, where F denotes the Fourier transform operator. Similar to light field, Wigner Distribution Function (WDF) [9, 20, 19] can jointly present both space and spatial-frequency of the scalar function, and is obtained by Fourier transforming the mutual intensity function of a scalar wave field $E(x)$:

$$W(x, u) = \int \left\langle E\left(x + \frac{f_u}{2}\right) E^*\left(x - \frac{f_u}{2}\right) \right\rangle e^{-i2\pi f_u u} df_u, \quad (3.7)$$

where $\langle \cdot \rangle$ is a time-averaging operator for incoherent illumination. Note that Wigner distribution is real but not necessarily positive due to destructive interference; however, projecting the Wigner distribution along either axis results in a non-negative function. In particular, we obtain the intensity distribution by projecting along the spatial-frequency:

$$\int_{-\infty}^{+\infty} W(x, u) du = |E(x)|^2. \quad (3.8)$$

Finally, a real-valued WDF that matches the light field can be further obtained by convolving the WDF of $E(x)$ with the inverted WDF of a finite aperture function [279].

The definition of Wigner distribution, unlike light field, reveals an important property of hologram: the angular spectrum is inherently encoded into the spatial pattern of the scalar function $E(x)$, and no spatial resolution is sacrificed. Despite the similarity and relation, reversing a light field to a scalar function $E(x)$ is challenging: even though the inverse Fourier transform of the Wigner distribution $F^{-1}(W(x/2, u))$ gives the scalar function up to a constant scalar $E(x)E^*(0)$ [52], obtaining the Wigner distribution by deconvolving an arbitrary light field with the inverted WDF of a local aperture is impractical and does

not always yield a valid Wigner distribution [128]. Even with valid WDFs, inverting high resolution (i.e. 1080p) WDF requires storing and processing trillions of samples that are not practically feasible. Ye et al. [263] approximate the nonnegative WDF via iterative global optimization. However, they sacrifice the spatial and angular resolution by partitioning the scalar field into independent segments.

3.2 Method

In the proposed CGH rendering pipeline, we start with polygon-based light field rendering such that occlusion is handled through the graphics pipeline using z-buffer. The output is a multiview holographic light field which contains color and position information intended for CGH calculation, as shown in Fig. 3-2 (top). We describe an improved rendering setup under plane wave illumination, which avoids aliasing and geometry clipping inherited in the previously proposed rendering setup [273]. We then extend the proposed algorithm to support spherical wave illumination, which enables near-eye CGHs. For both rendering setups, the point based method with local partitioning is used for CGH calculation, which allows for per-pixel depth reconstruction and parallel computation on GPU at interactive speed in 1080p image quality for the first time.

3.2.1 Holographic Light Field Rendering under Plane Wave Illumination

Calculating a full parallax hologram demands a densely sampled 4D light field [121, 177], where a local view (elemental image, see Fig. 3-2 (top red bounding box)), which records depth-sorted visible scene points, is rendered for each SLM pixel from its spatial location and based on the diffraction geometry. The same 4D light field can also be rendered through a dense angular view sampling at the SLM resolution, an approach that is often more preferred due to fewer rendering passes. Nevertheless, even under the small diffraction angle of existing SLMs, rendering a 1080p spatial resolution light field at an angular resolution equivalent to human visual acuity is computationally impractical for real-time graphics.

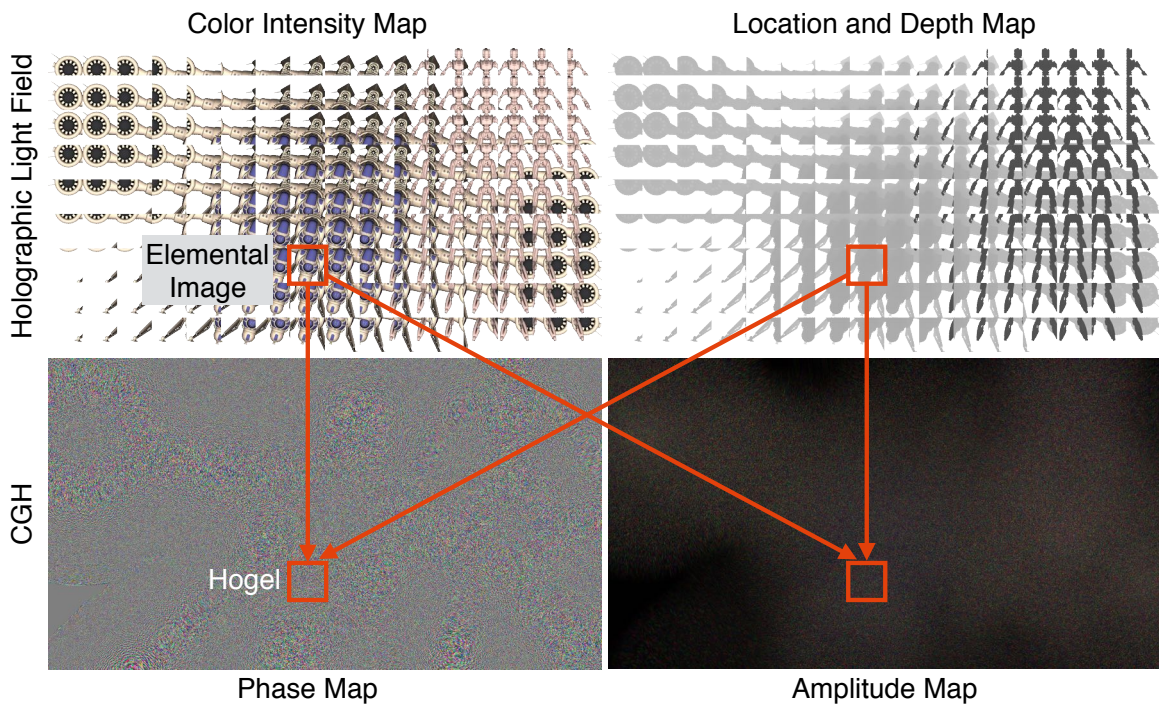


Figure 3-2: We render the holographic light field with both color (top-left) and position (top-right) information. The per-SLM-pixel phase (bottom left) can be derived from the position, and the amplitude is obtained from both inputs. Although the hogel-to-elemental is a one-to-one mapping, but the mapping within block is a complete bipartite graph, resulting in heavy computation.

Assuming Lambertian surfaces, rendering a dense 4D light field generates highly similar views for adjacent SLM pixels and results in large redundancies: a single recording of each object point, regardless of the visibility to each SLM pixel, is sufficient to determine the point’s wavefront over the entire SLM. Leveraging this observation, a hologram can be spatially partitioned into abutting grids, called holographic elements (hogels), as shown in Fig. 3-2 (bottom red bounding box), such that only a single representative view, an elemental image, is rendered and used in calculation for each hogel, assuming all captured points are visible to all pixels in the hogel. The reduction in parallax introduces an error in occlusion bounded by the hogel size w_h (about 1mm in our prototype) within the eye box; this approximation substantially reduces the number of rendering pass, and is widely adopted in holographic stereogram calculations [261, 133]. Under plane wave illumination, the previous work renders a hogel’s view by setting a virtual camera directly on the SLM surface and at the hogel’s center, as shown in Figure 5 (bottom). The view frustum, as shown by the pink cone, has a field of view equaling the maximum diffraction angle defined by the SLM pixel. However, this scheme cannot provide an accurate per-pixel wavefront accumulation as it only renders the object points visible to the hogel’s center pixel and also extends poorly to spherical wave illumination. Consider the hogel’s bottom SLM pixel, the rendering of objects from the top red parallelogram is mistakenly incorporated, and that from the bottom yellow parallelogram is incorrectly excluded. Meanwhile, to prevent geometry clipping, the separation w_h between adjacent view frustums sets an implicit depth limit

$$d_{min} = \frac{w_h}{2 \tan \left(\sin^{-1} \left(\frac{\lambda}{2\Delta p} \right) \right)}, \quad (3.9)$$

for which objects placed inside will be clipped and not presented in the final hologram; artifacts of such are shown in Sec. 3.4.2. We propose a unified framework, holographic light field rendering, in the following, to support accurate view estimation under various illuminations.

To gather the part of 3D scene visible to the entire hogel, we shift the virtual camera in the z direction, as shown in Figure 5 (top), to the intersection of the two marginal rays, defined by the diffraction cone associated with the hogel’s top and bottom pixel. The lateral

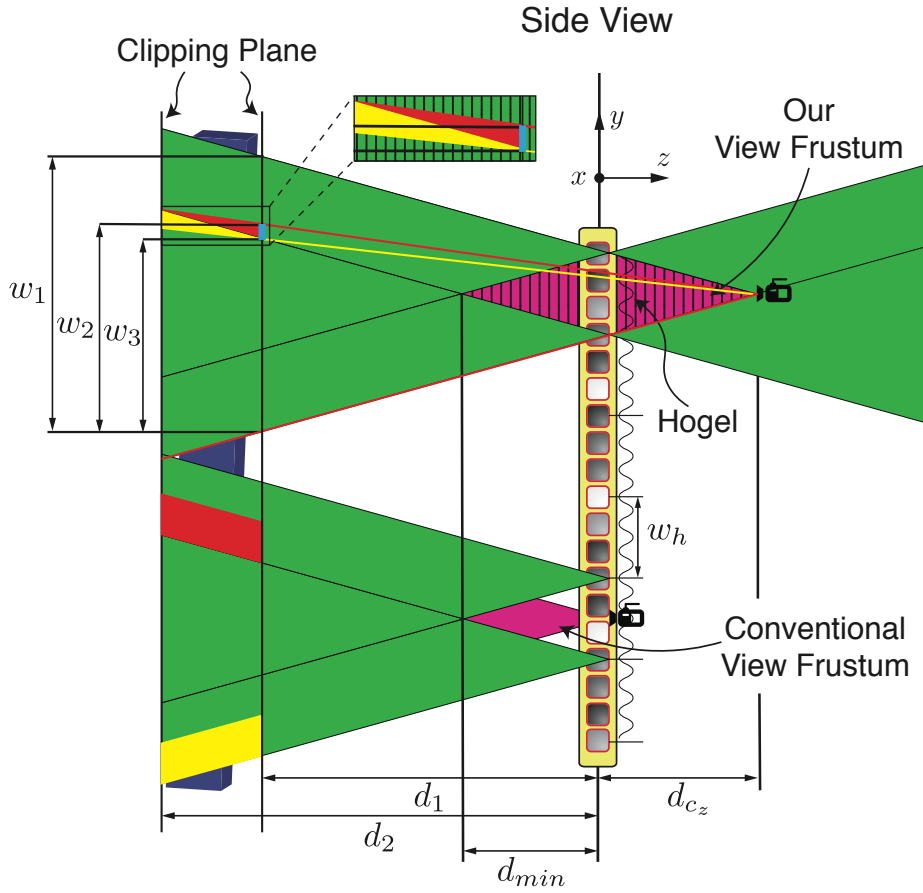


Figure 3-3: Holographic light field rendering for plane wave illumination. The view frustum setup (top) covers the entire space behind the SLM and avoids geometry clipping, as opposed to convention frustum (bottom). The per-SLM-pixel perspective is obtained by aligning the diffraction cone in the far-clipping plane with the sliding window w_2 . Pixels rendered from the ambiguity region, as shown in the inset, needs a maximal diffraction culling.

offset to the SLM is given by $d_{c_z} = d_{min}$. The offset camera projects all the scene points, visible to the pixels in the hogel, to the area $w_1 = 2 \sin^{-1}(\lambda/2\Delta p)(d_1 + d_{c_z})$ for further processing. Again, for the hogel's bottom pixel, we incorrectly incorporate the top portion of the scene points when using the entire view w_1 .

To address this problem, a conservative sliding window considers only the valid fraction in the rendered hogel view during the final wave calculation. This sliding window, $w_3 = \left(1 - \frac{d_{c_z}}{d_1 + d_{c_z}}\right) w_1$, on the front clipping plane is defined by projecting a cone of maximum diffraction angle from the SLM pixel under consideration (the bottom hogel pixel in fig. 5 (top)) to the scene object. Since we approximate the projection geometry by replacing the per-pixel rendering with the per-hogel rendering, the 3D projection to the offset virtual camera also introduces ambiguity segments, as shown in the inset of Figure 5. For the bottom hogel pixel, an example is shown in the blue segment, where both the valid yellow triangle (should be considered) and the invalid red triangle (should be excluded) are projected to the same location. During the final wave calculation, we disambiguate this region by testing whether or not the angle to the SLM pixel under consideration is smaller than the maximum diffraction angle. To test the visibility, we include the ambiguity segment by extending the sliding window to the following:

$$w_2 = \left(1 - \frac{d_{c_z}}{d_2 + d_{c_z}}\right) w_1, \quad (3.10)$$

Arranging a camera array under the proposed configuration allows for unrestricted disposition of scene objects. The lateral separation between camera and SLM ensures that adjacent camera views overlap immediately in front of the SLM, and the tiled frustum array fully covers the field of view of the entire hologram, allowing near clipping plane to be set at arbitrary depth in front of SLM. The configuration works well for rendering under plane wave diffraction geometry. For rendering under spherical wave illumination, the prior discussion still applies, but the camera geometry has to consider the limited eye box and the expanded field of view as discussed in Sec. [3.2.2](#).

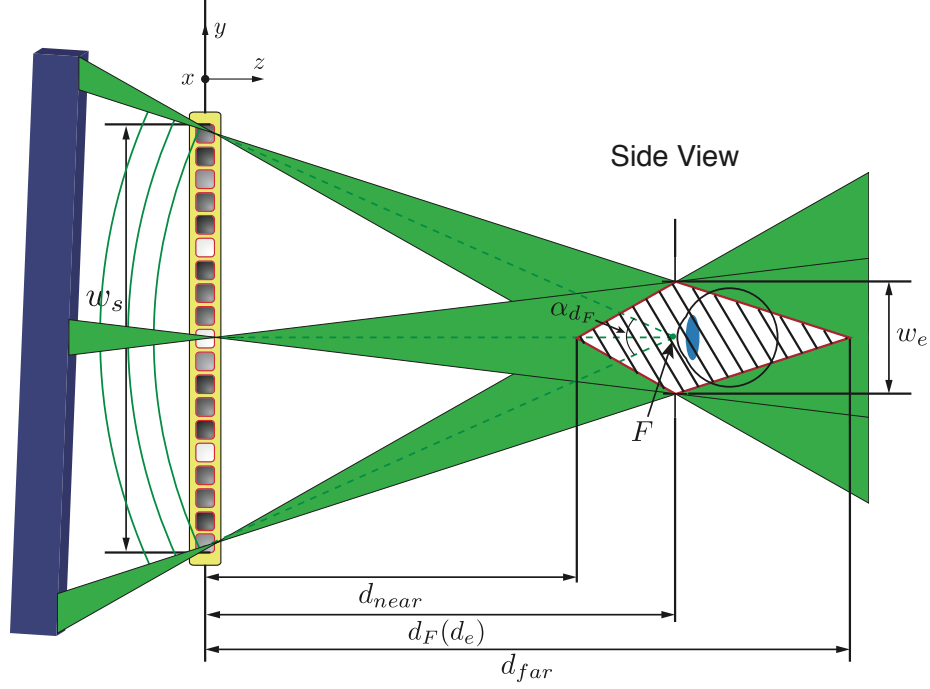


Figure 3-4: Spherical wave driven holographic display. Each display pixel diffracts a light cone toward the focus. The entire display is observable by placing pupil in the diamond-shape viewing zone.

3.2.2 Spherical-Holographic Light Field Rendering

CGH using SLM with plane wave illumination gives narrow field of view and requires long eye relief, as discussed in Sec. 3.2.1. Similar to Qu et al. [206] applying a diverging source to expand the image field for projection, we use a converging beam to create a light field near the pupil with a wider field of view and shorter eye relief, as shown in Fig. 3-4.

We denote the focus of converging wavefront F , and the distance from the focus to SLM d_F . Unlike in plane wave illumination, the eye box is confined in between the two depths according to Eq. (3.3):

$$d_{\{far,near\}} = \frac{w_s}{2 \tan \left(\sin^{-1} \left(\sin \left(\tan^{-1} \left(\frac{w_s}{2d_F} \right) \right) \pm \frac{\lambda}{2\Delta p} \right) \right)}. \quad (3.11)$$

When eye is located at focus (eye relief $d_e = d_F$), the resulted field of view α_{d_F} and eye

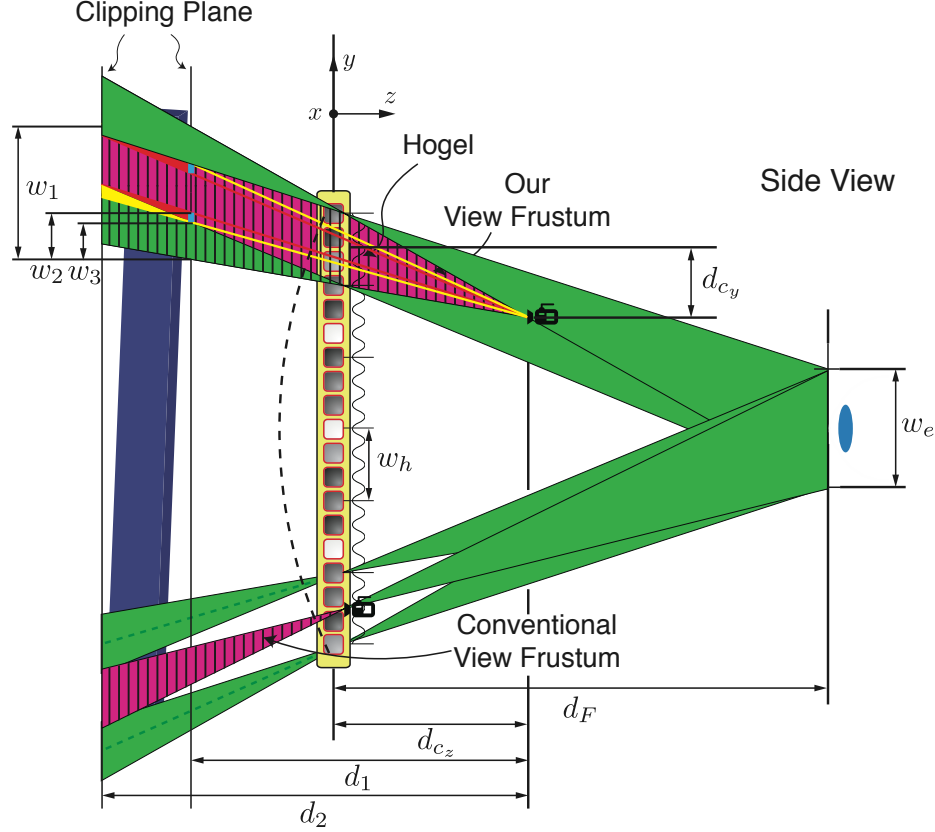


Figure 3-5: Holographic light field rendering for spherical wave illumination. The camera view frustum is spatially non-uniform and steered off-axis to capture the hogel's entire field of view. Calculating per-SLM-pixel view is similar to that in plane wave setup. Points in red triangle can be distinguished by examining the angle with respect to the central ray of diffraction cone (dashed green line).

box $w_{e(d_F)}$ follows

$$\alpha_{d_F} = 2 \tan^{-1} \left(\frac{w_s}{2d_e} \right), \quad w_{e(d_F)} = 2 \tan \left(\sin^{-1} \left(\frac{\lambda}{2\Delta p} \right) \right) d_F. \quad (3.12)$$

Equation (3.12) shows the significance of using converged spherical wave illumination, that (1) the eye relief is solely determined by the focal length of the converging wavefront, (2) the field of view only depends on the spatial extent of the SLM w_s and the eye relief d_e , which is constrained by the demand for larger eye box, and (3) smaller pixel pitch results in a larger eye box with fixed eye relief. These relations offer more flexibilities in designing near-eye displays: using a shorter focal length wavefront allows for a wider field of view but a smaller eye box [175]; in this paper, we require a minimal 10mm eye box coverage.

Under spherical illumination, the viewing frustums undergo spatially-varying transforms: the wavefront curvature introduces an off-axis rotation to the local incident ray direction of the diffraction cone, as shown in Fig. 3-5 (top). The diverged viewing cones collectively widen the field of view of hogel. In the conventional setup Fig. 3-5 (bottom), the viewing frustums are completely disjoint to the diffraction cones and fail to provide a properly generalized view.

Extending our proposed camera arrangement to spherical illumination sets the virtual camera at the intersection of the marginal rays restricted by the eye box (Eq. (3.12)) and skews the field of view. The separation between camera and SLM is given by

$$d_{c_z} = \frac{d_F w_h}{w_e(d_F) + w_h}. \quad (3.13)$$

The offset between camera and hogel center along x - and y -axis depends on each hogel's position to the eye box. Assuming $2M + 1$ by $2N + 1$ partitioning of the SLM along x and y , the displacement from the (m, n) -th hogel center to its virtual camera is given by

$$d_{c_x} = \frac{m w_h d_{c_z}}{d_F}, \quad d_{c_y} = \frac{n w_h d_{c_z}}{d_F}. \quad (3.14)$$

In camera space, we can define the off-axis projection matrix as

$$P_{\{m,n\}} = \begin{bmatrix} \frac{2d_{c_z}}{w_h} & 0 & \frac{2d_{c_x}}{w_h} & 0 \\ 0 & \frac{2d_{c_z}}{w_h} & \frac{2d_{c_y}}{w_h} & 0 \\ 0 & 0 & -\frac{d_2+d_1}{d_2-d_1} & -\frac{2d_2d_1}{d_2-d_1} \\ 0 & 0 & -1 & 0 \end{bmatrix}. \quad (3.15)$$

The sliding window w_2 inside each elemental image to disambiguate the projected pixels is

$$w_2 = \left(1 - \frac{d_{c_z}(d_2 + d_F - d_{c_z})}{d_2 d_F} \right) w_1, \quad (3.16)$$

and the fraction of error-free segment in the sliding view

$$\frac{w_3}{w_2} = 1 - \frac{(d_2 - d_1)d_{c_z}}{d_1(d_2 - d_{c_z})}. \quad (3.17)$$

is used to derive the hogel size required to obtain an acceptable sampling error, as will be discussed in Sec. [3.3.3](#).

We emphasize two extreme choices for hogel size. If the hogel reduces to a single pixel, the holographic light field rendering expands to full light field rendering [\[121\]](#). On the other hand, if the hogel extend to the entire SLM, holographic light field recedes to a single map of points cloud rendered from the nearest distance where SLM is fully observable to the viewer. For ultra-short focal length wavefront, this location is effectively at the focal point, and Maimone et al. [\[175\]](#) also adopts similar setting.

3.2.3 Calculating Object Wave for Displaying on SLM

In holographic light field, the one-to-one mapping between SLM hogel and its visible elemental image, as shown in Fig. [3-2](#), facilitates parallel computation using point-based method through Fresnel integral.

Let $\mathbf{p} \in \mathbb{R}^3$ denotes a SLM pixel in the (m, n) -th hogel at a displacement $(\Delta x, \Delta y)$ to the hogel center. In its associated camera space under spherical illumination, $\mathbf{p}$'s spatial coordinate is given by $(\Delta x + d_{c_x}, \Delta y + d_{c_y}, -d_{c_z})$. Let $\mathbf{p}$'s estimated view be a sliding window of $k \times k$ pixels and define $\mathbf{o}_j$ be the j -th elemental pixel whose rendered point locates at $(x_{\mathbf{o}_j}, y_{\mathbf{o}_j}, z_{\mathbf{o}_j})$ with amplitude $A_{\mathbf{o}_j}$ and initial phase $\phi_{\mathbf{o}_j}$. The Euclidean distance between $\mathbf{p}$ and $\mathbf{o}_j$ is given by

$$r(\mathbf{p}, \mathbf{o}_j) = \sqrt{(x_{\mathbf{o}_j} - \Delta x - d_{c_x})^2 + (y_{\mathbf{o}_j} - \Delta y - d_{c_y})^2 + (z_{\mathbf{o}_j} + d_{c_z})^2}. \quad (3.18)$$

Since the CGH converts an illuminating reference wave $E_{\text{ref}}(x)$ to the object wave $E_{\text{obj}}(x)$, we need to interfere or pre-multiply the object wave with the conjugate reference wave—a reference wave that reversely propagates to the SLM. Conjugate spherical wave also has to be evaluated in a per-pixel basis that requires the distance from $\mathbf{p}$ to the focus of spherical

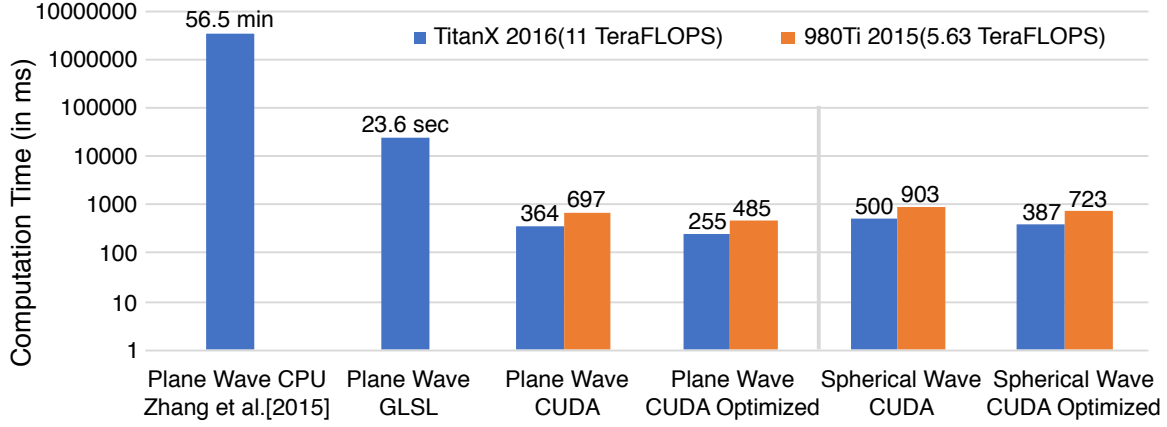


Figure 3-6: Comparing performance under CPU and different GPU implementations, and different illuminations. Under plane wave method, the elemental image size is 128×128 (2048×1152 overall points cloud), and 160×160 (2560×1440 overall) under spherical wave method; both methods achieve 30 pixels per degree angular sampling on a 1920×1080 resolution SLM.

wavefront $\mathbf{F}$ at the origin:

$$r(\mathbf{p}, \mathbf{F}) = \sqrt{d_F^2 + (mw_h + \Delta x)^2 + (nw_h + \Delta y)^2}. \quad (3.19)$$

As a result, the on-axis spherical illuminating wave establishes the fringe of the SLM at location p :

$$E(p) = \underbrace{\left(\sum_{j=1}^{k^2} \frac{A_{o_j}}{w_{o_j}} e^{i2\pi \frac{r(\mathbf{p}, \mathbf{o}_j)}{\lambda} + \phi_{o_j}} \right)}_{\text{Object Wave } E_{\text{obj}}(x)} \underbrace{\left(\frac{A_F}{w_F} e^{-i2\pi \frac{r(\mathbf{p}, \mathbf{F})}{\lambda}} \right)}_{\text{Conjugate Reference Wave } E_{\text{ref}}^*(x)}, \quad (3.20)$$

where A_F denotes the amplitude of conjugate spherical wave, w_{o_j} and w_F are the normalizing factors for the object wave and the conjugate reference wave respectively. Note that the CGH under plane wave illumination is simply the object wave $E_{\text{obj}}(x)$ in the equation, where $r(\mathbf{p}, \mathbf{o}_j)$ is reformulated according to plane wave rendering geometry. For paraxial region, Eq. (3.20) can be simplified by Fresnel approximation to remove the square root operation. In this work, we aim for an accurate quadruple sum.

3.2.4 Software Implementation

We render the holographic light field on a PC with an Intel core i7 3.3GHz CPU and 16GB of RAM, and a NVIDIA TitanX GPU. The multiview holographic light field is rendered with Simultaneous Multi-projection to the geometry shader stage. The elemental image resolution bounds the rendering cost. Unlike the full resolution light field [121], the view size is bounded by the relatively small diffraction cone. For 4K SLM, the overall rendering of resolution 5376×3024 (16×9 hogels) takes less than 20 ms.

We use CUDA-OpenGL Inter-Op to calculate the CGH, and the pseudo kernel code at each SLM pixel is outlined in Algorithm 1, where P_{ws} , P_{wh} , P_{w_1} , and P_{w_2} denote the number of pixels in the SLM, the hogel, the elemental image, and the sliding window respectively. The heavy computations in the loop (line 5-12) hides the latency of the repeated texture fetching (line 6 and line 9). The major performance bottleneck comes from three parts: the exact distance from point to SLM pixel to obtain the phase (line 8), generating the complex number of the wave involving trigonometric operations (line 11), and the conditional branching (line 10). We re-formulate the maximum diffraction comparison to avoid division and minimize the branching code, and we have further parallelized the compute-looping (line-5) using prefix-sum algorithm and using on-chip shared memory to avoid the memory bank conflicts, and obtained additional 40% performance compared to an un-optimized GPU CUDA code, $100\times$ to a GLSL shader implementation of the same algorithm, and more than $13000\times$ to a C++ implementation on CPU [273]. Maimone et al. [175] show similar performance to our GLSL code for offline computation of the point based method. We profiled the code to be heavily compute-bounded (nearly 11 trillion floating point operation).

Figure 3-6 evaluates the performance of the software implementations under different illuminations. The runtime is obtained at 30 pixels per visual degree angular sampling rate, similar to commercial VR/AR devices. Specifically, the CPU based plane wave method by Zhang et al. [273] takes nearly an hour to compute a 3-color-channel 1080p hologram, where an optimized CUDA implementation takes only 250ms, achieving more than 5 orders of magnitude speed up. The performance is compute-bounded, thus a Pascal Titan X with

Algorithm 1 Complex Field CGH Calculation with Spherical Wave Illumination

```
1:  $\mathbf{i} \leftarrow \text{pixel index in SLM}$ 
2:  $\phi_{E_{\text{ref}}^*(x)} \leftarrow \text{length} \left( \left\{ \left( \mathbf{i} - \frac{P_{ws}}{2} \right) \cdot \Delta p, D_F \right\} \right) / \lambda \cdot 2\pi$ 
3:  $\mathbf{i}_{Hogel} \leftarrow \text{mod}(\mathbf{i}, P_{wh})$ 
4:  $\mathbf{i}_{Elem} \leftarrow \lfloor \mathbf{i} / P_{wh} \rfloor \cdot P_{w1} + \mathbf{i}_{Hogel} / P_{wh} \cdot (P_{w1} - P_{w2})$ 
5: for Pixel index in sliding window  $\mathbf{j} = 0$  to  $P_{w2} \times P_{w2}$  do
6:    $\mathbf{q} \leftarrow \text{LightFieldPositionTex}[\mathbf{i}_{Elem} + \mathbf{j}]$ 
7:    $\Delta \mathbf{q} \leftarrow \left\{ \left( \mathbf{i} - \frac{P_{wh}}{2} \right) * \Delta p + \mathbf{d}_{c_{\{x,y\}}}, -d_{c_z} \right\}$ 
8:    $\phi \leftarrow \text{Length}(\mathbf{q} - \Delta \mathbf{q}) / \lambda \cdot 2\pi + \phi_{E_{\text{ref}}^*(x)} + \phi_{\text{Init}}$ 
9:    $A \leftarrow \text{LightFieldColorTex}[\mathbf{i}_{Elem} + \mathbf{j}] \cdot (1/r)$ 
10:  if Angle < MaxDiffractionAngle then
11:     $\mathbb{E}(\mathbf{i}) \leftarrow \mathbb{E}(\mathbf{i}) + A \cdot e^{i\phi}$ 
```

11 TeraFLOPS nearly doubles the frame rate using a GeForce 980Ti with 5.63 TeraFLOPS.

3.3 Theoretical Analysis

In this section, we provide theoretical analysis on the sampling, resolution limits of light field-based holographic display, and its design space trade-off.

3.3.1 Sampling Outside the Maximum Diffraction Cone

In Sec. [3.2.1](#), a sliding window w_2 limits the wavefront sampling; we mathematically show that sampling wavefront of any point outside the half maximum diffraction cone dictated by highest spatial frequency $u^{max} = |1/2\Delta p|$ results in aliasing. Assuming paraxial region where WDF description is accurate, we analyze a single spherical point emitter on a 1D hologram. Consider the emitter at (x_0, z_0) , its wavefront at $z = 0$ and WDF (Eq. [\(3.7\)](#)) are:

$$E(x) = e^{i\pi \frac{(x-x_0)^2}{z_0\lambda}}, \quad W_{z=0}(x, u) = \delta \left(\frac{x-x_0}{z_0\lambda} - u \right). \quad (3.21)$$

Sampling the wavefront at discrete interval Δp on the SLM generates infinite replicas (Fig. [3-7](#) (left)) of doubly sampled WDF $W_{z=0}(x, u)$ along u -axis at a displacement of

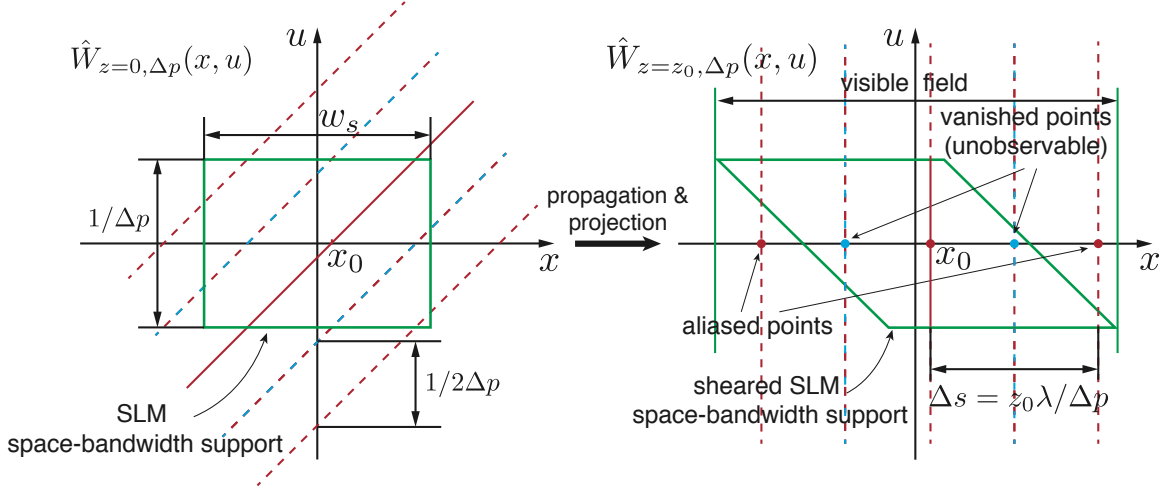


Figure 3-7: WDF analysis of aliasing caused by sampling outside maximum diffraction cone. (Left): The dashed lines denote replicas from discrete sampling. The display's space-bandwidth product in green rectangle is bounded by the spatial extent w_s and maximum spatial frequency $\pm 1/2\Delta p$. (Right): The visible field is the lateral extent of the sheared space-bandwidth product. Each constructively interfered replica in red dash line projects to a aliased point at displacement depicted the Nyquist sampling rate.

$1/2\Delta p$ [237]:

$$\hat{W}_{z=0,\Delta p}(x,u) = \frac{1}{2\Delta p} \left(\sum_k \sum_l \delta(x - k\Delta p) W_{z=0}(x, u - l/2\Delta p) + (-1)^k \delta(x - (k + 1/2)\Delta p) W_{z=0}(x, u - l/2\Delta p) \right). \quad (3.22)$$

The space-bandwidth product of the SLM, shown as green rectangle in the Wigner distribution chart, bounds the reproducible wavefront. To observe the point emitter, we propagate the wavefront back to z_0 ; the WDF, similar to light field, is sheared along x -axis [93]. The intensity distribution at z_0 is obtained by projecting along the u -axis (Eq. (3.8)) from the visible WDF enclosed by the sheared space-bandwidth support, as shown in Fig. 3-7 (right). For odd k -th replicas, no spatial point is observed because alternating signs (dash line interleaved with red and blue) destructively interfere; for even k -th replicas, aliased points are generated within the field of view. Nyquist theorem depicts the displacement between adjacent aliased emitters is jointly characterized by both the maximal spatial frequency and the depth of the target emitter, and is given by $\Delta s = z_0 \lambda / \Delta p$. To avoid aliasing, the even k -th replicas should be outside the sheared space-bandwidth support. From the

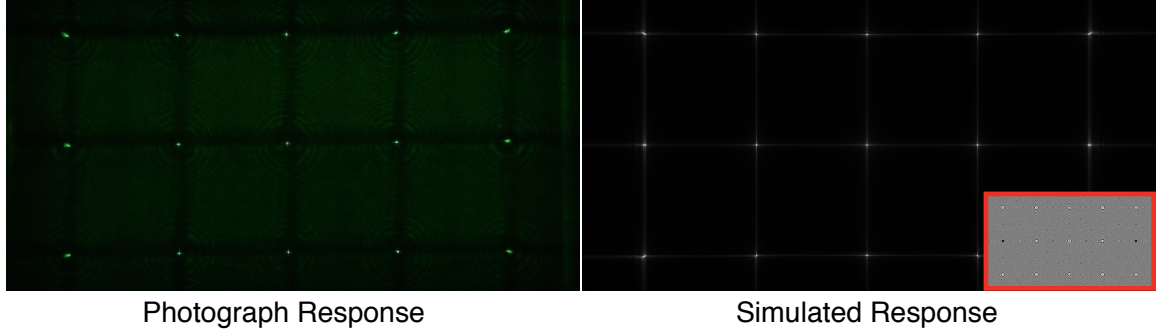


Figure 3-8: Photographed aliasing response (left) and simulated aliasing response (right) due to sampling outside maximum diffraction cone as discussed in Sec. 3.3.1. Multiple Fresnel zone plates are presented in the over-sampled diffraction pattern (red inset). At non-paraxial region, the aliased points no longer preserve a circular shape.

analysis, if the point emitter is located within a minimal distance to the SLM

$$z_0 \leq z_{min} \equiv (w_s + |2x_0|)\Delta p/\lambda, \quad (3.23)$$

we need to avoid sampling outside the maximum diffraction cone limit using a sliding window w_2 ; artifact due to oversampling is shown in the Fig. 3-8. **Equation (3.23) also characterizes the *Depth of Field*.** In Sec. 3.3.2, we will discuss the limits of reconstructed resolution inside and outside z_{min} .

3.3.2 Resolution Limits on the Depth of Field (DoF)

We show that, like light field displays, holographic displays also have limited spatial resolution outside the depth of field, beyond which spatial resolution exhibits a depth-dependent degradation. The effect is caused by the finite extent of the SLM under anti-aliased (Sec. 3.3.1) discrete sampling of a continuous wavefront. We distinguish two cases: *spatially-limited* sampling when a point emitter is outside z_{min} (Eq. (3.23)) and *frequency-limited* if inside.

Spatially-limited Sampling When $z_0 \geq z_{min}$, the SLM is smaller than the projected maximum diffraction cone, and all pixels sample the wavefront, the sampled wavefront is expressed as:

$$E_{w_s}(x, 0) = e^{i\pi \frac{(x-x_0)^2}{z_0\lambda}} \Pi\left(\frac{x}{w_s}\right), \quad (3.24)$$

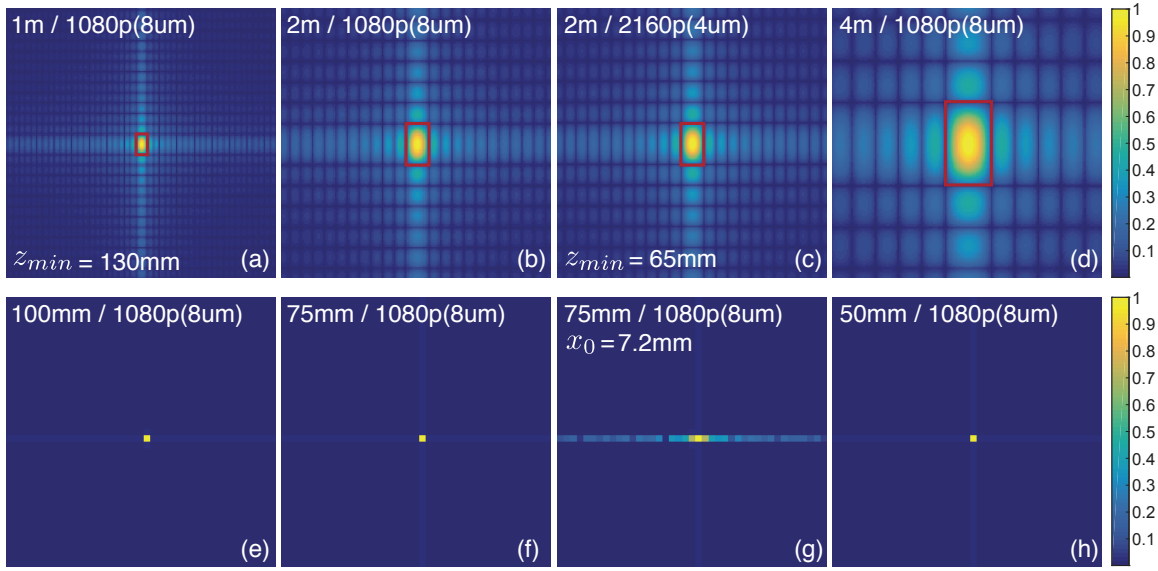


Figure 3-9: Spatial resolution limit. (First Row): *spatially-limited sampling outside the DoF*. The red bounding box marks the width of the main lobe as resolution limit, and the bounding box scales proportional to the distance from the point emitter to the display. The observed resolution limits (Eq. (3.26)) is the same for display with the same spatial extent but smaller pixel pitch (b,c); however, their DoF (Eq. (3.23)) are different. (Second Row): *frequency-limited sampling within the DoF*. Spatial resolution in (e,f,h) remains at Δp . However, when the display boundary crops the diffraction cone, reconstruction degrades as shown in (g) due to under-sampling.

where $\square(\cdot)$ is a rectangular function modeling the spatial extent of the SLM. The corresponding WDF can be derived as

$$\begin{aligned} W_{w_s, z=0}(x, u) &= 2w_s \left(1 - \frac{|2x|}{w_s}\right) \text{Sinc} \left(w_s \left(1 - \frac{|2x|}{w_s}\right) \left(\frac{x-x_0}{z_0\lambda} - u \right) \right) \\ &= \delta \left(\frac{x-x_0}{z_0\lambda} - u \right) \otimes_u 2w_s \left(1 - \frac{|2x|}{w_s}\right) \text{Sinc} \left(2w_s \left(1 - \frac{|2x|}{w_s}\right) u \right), \end{aligned}$$

which is a convolution along u -axis between the WDF of the continuous wavefront and the WDF of the rectangular function. The convolution dilates the frequency response from a Dirac delta function to a spatially varying Sinc function, as shown in Fig. 3-10 (left). Because Sinc degrades rapidly after its first zero-crossing, we characterize the frequency dilation by the width of the main lobe:

$$\Delta u = \frac{1}{w_s \left(1 - \frac{|2x|}{w_s}\right)}, \quad x \in (-w_s/2, w_s/2). \quad (3.25)$$

At $x = 0$, the minimal frequency dilation $\Delta u_{min} = 1/w_s$ is obtained and centers around $u = \frac{-x_0}{z_0\lambda}$. To observe the point emitter, we propagate the WDF back to $z = z_0$ and the frequency dilation shears to a spatial dilation. In Fig. 3-10 (middle), the yellow dash line marks the sheared $W_{w_s, z=0}(x = 0, u)$ and bounds the minimal spatial dilation

$$\Delta x_{min} = z_0\lambda \Delta u_{min} = \frac{z_0\lambda}{w_s} \equiv \frac{z_0\lambda}{(P_{w_s}\Delta p)}. \quad (3.26)$$

Calculating the exact intensity distribution at $z = z_0$ requires integrating the shared spatially varying Sinc along u -axis (Eq. (3.8)). However, we notice that for $x \in (x_0 - \frac{1}{2}\Delta x_{min}, x_0 + \frac{1}{2}\Delta x_{min})$, as shown in the region defined by two dash lines in Fig. 3-10 (right), the WDF is always positive and every sheared main lobes from different spatial locations are at least Δx_{min} wide. Thus, the spatial resolution of the point is upper-bounded by Δx_{min} ; this width agrees with the closed-form solution $\text{Sinc}^2 \left(\frac{(x-x_0)}{\Delta x_{min}} \right)$ obtained by Kizacki [145] and shows a depth-dependent degradation.

To validate, we reconstruct a point emitter at distances $1m$, $2m$, $4m$ to a 1920×1080 SLM of $8\mu m$ pixel pitch, as shown in Fig. 3-9 (a, b, and d), in simulation [184], and the

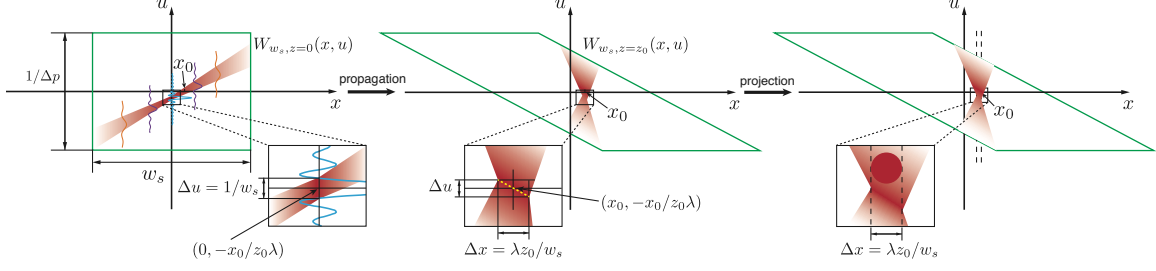


Figure 3-10: WDF analysis of resolution limit outside the DoF. (Left): Frequency is minimally dilated to the main lobe of the Sinc function near the center, while the peripheral dilation (orange Sinc) has wider spread and weaker peak amplitude. (Middle): The yellow dash line marks the sheared minimally dilated frequency at the observed distance. (Right): The dash lines bound the region where Sinc response is always positive, and the projected circle bounds the resolution.

extent of reconstructed points scales proportionally to the distance. Doubling the SLM resolution while halving the pitch results in the same degraded spatial dilation (Eq. (3.26)) but the DoF also reduces. A point reconstructed at $2m$ to a 3840×2160 SLM of $4\mu m$ pixel pitch SLM (c) has the same dilation as that from a 1920×1080 SLM of $8\mu m$ pixel pitch SLM (b). Increasing display size w_s , e.g. by tiling, expand the DoF and avoid resolution degradation.

Frequency-limited Sampling For object within the DoF of the the display $z_0 < z_{min}$, the SLM is wider than the maximum diffraction cone projection and the anti-aliased sampling limits SLM active region inside the boundary of the projected maximum diffraction cone and the SLM extent

$$x_l = \max\left(x_0 - \frac{z_0\lambda}{2\Delta p}, -\frac{w_s}{2}\right), \quad x_r = \min\left(x_0 + \frac{z_0\lambda}{2\Delta p}, \frac{w_s}{2}\right), \quad (3.27)$$

as shown in Fig. 3-11 (left). For simplicity, we denote $x_c = (x_l + x_r)/2$ and $w'_s = (x_r - x_l)$.

The continuous wavefront can be subsequently modeled as

$$E_{w'_s}(x, 0) = e^{i\pi \frac{(x-x_0)^2}{z_0\lambda}} \Pi\left(\frac{x-x_c}{w'_s}\right). \quad (3.28)$$

and the corresponding WDF can be derived as

$$\begin{aligned}
W_{w'_s, z=0}(x, u) &= 2w'_s \left(1 - \frac{2|x-x_c|}{w'_s}\right) \text{Sinc} \left(2w'_s \left(1 - \frac{2|x-x_c|}{w'_s}\right) \left(\frac{x-x_0}{z_0\lambda} - u\right)\right) \\
&= \delta \left(\frac{x-x_0}{z_0\lambda} - u\right) \otimes_u 2w'_s \left(1 - \frac{2|x-x_c|}{w'_s}\right) \text{Sinc} \left(2w'_s \left(1 - \frac{2|x-x_c|}{w'_s}\right) u\right),
\end{aligned} \tag{3.29}$$

from which the frequency dilation is characterized as the width of main lobe

$$\Delta u = \frac{1}{w'_s \left(1 - \frac{2|x-x_c|}{w'_s}\right)}, \quad x \in (x_l, x_r). \tag{3.30}$$

We note that the minimal frequency dilation $\Delta u_{min} = 1/w'_s$ is achieved at $x = x_c$, which depends on both (x_0, z_0) and w_s (Fig. 3-11 left). The minimal spatial dilation is similarly obtained through x -axis shearing

$$\Delta x_{min} = z_0\lambda\Delta u_{min} = z_0\lambda/w'_s, \tag{3.31}$$

and substituting the active sampling region $w'_s = z_0\frac{\lambda}{\Delta p}$ in Eq. (3.31) gives

$$\Delta x_{min} = \Delta p. \tag{3.32}$$

This reveals a key observation: **the spatial resolution remains constant and equals to the size of the pixel pitch when the distance is within the depth of field.** Note that this assumes the active region is solely limited by (x_0, z_0) (Eq. (3.27)), namely only by the maximum diffraction cone. Additional cropping due to SLM extent reduces the spatial resolution (Fig. 3-9 (g)).

To validate the analysis, we reconstruct a point emitter at distances 100mm, 75mm, 50mm to the SLM, all within the DoF of a 1920×1080 SLM of 8um pixel pitch, and all points are of single pixel wide (Fig. 3-9 (e, f, and h)). When the point is laterally offsetted, the spatial extent of the SLM causes undersampling, and an under-resolved point

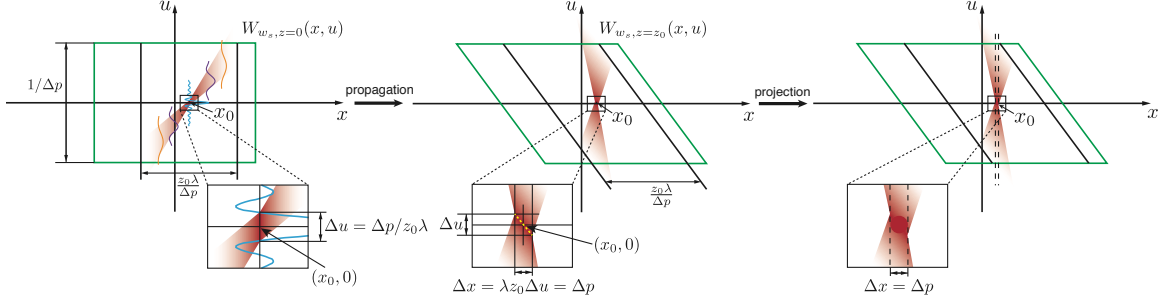


Figure 3-11: WDF analysis of resolution limit under frequency-limited sampling inside the DoF. (Left): Active sampling region is limited within the maximum diffraction cone projection. Frequency is minimally dilated to the main lobe of the Sinc function at x_0 . (Middle): The yellow dash line marks the sheared minimally dilated frequency at the observed distance. Within the DoF, minimum spatial dilation remains at Δp . (Right): The dash lines bound the region where Sinc response is always positive, and the projected WDF inside this region only interferes constructively.

is reconstructed as a line in (g). To conclude, we show an overall resolution distribution as a function of SLM width and pixel pitch in Fig. 3-12

Resolution Limits under Spherical Illumination The above analysis assumes plane wave illumination. Placing a lens of focal length d_F behind the SLM magnifies the optical setup and creates a converging spherical illumination. With the magnification, the depth of field under spherical illumination is also magnified:

$$\overline{z_{min}} = \frac{d_F}{d_F - z_{min}} z_{min}. \quad (3.33)$$

We still distinguish two cases: when the depth of field under plane wave is outside the converging focal length ($z_{min} \geq d_F$), the CGH under spherical illumination has infinite depth of field. On the other hand, the reconstructed point between the depth of field and the focal length ($z_{min} \leq z_0 \leq d_F$) has a magnified point spread at the virtual image plane. We conclude the reconstructed resolution as following:

$$\overline{\Delta x} = \begin{cases} \frac{d_F}{d_F - z_0} \Delta p, & \text{if } z_0 \leq \min(d_F, z_{min}) \\ \frac{d_F}{d_F - z_0} \Delta x_{min}, & \text{otherwise} \end{cases} \quad (3.34)$$

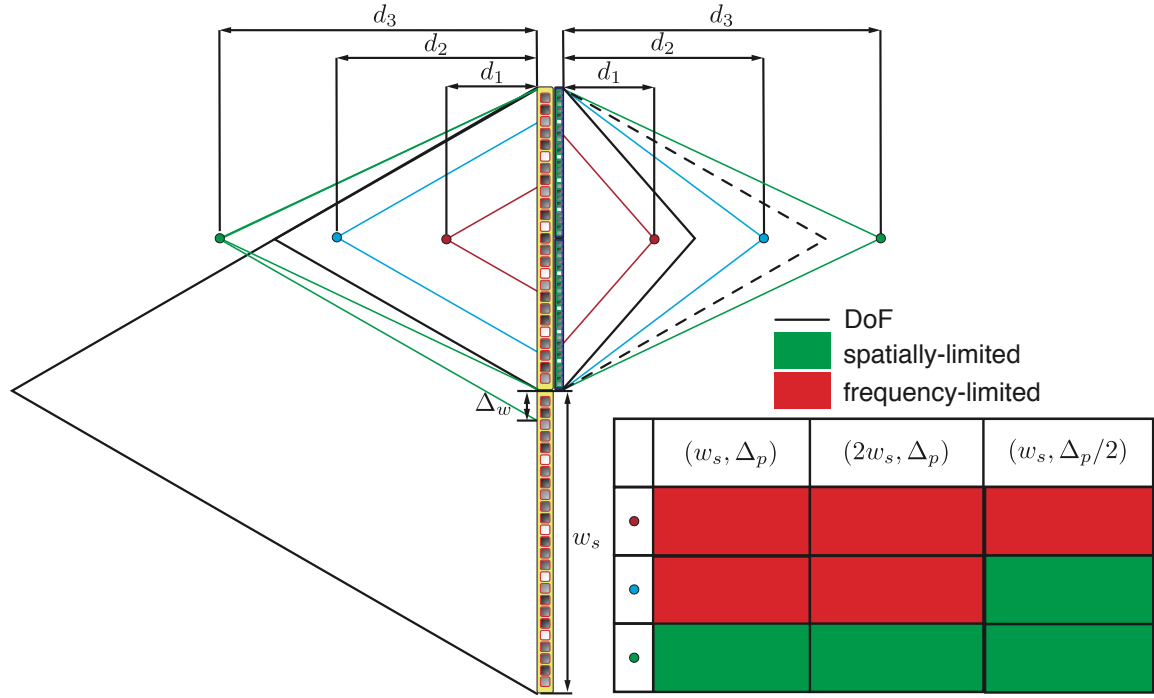


Figure 3-12: Illustration of spatial resolution as a function of SLM width and pixel pitch. For $(w_s, \Delta p)$ SLM (yellow top), red and blue point inside DoF are reproduced at Δp resolution. For $(w_s, \Delta p/2)$ SLM (dark blue top), resolution inside DoF is doubled (red point). Sampling of blue point becomes spatially-limited as DoF span halves, however, spatial resolution is still higher than $(w_s, \Delta p)$ case since active sampling area is the entire SLM. For both cases, the green point subtends to the entire SLM and are reproduced at same spatial resolution. For $(2w_s, \Delta p)$ SLM (yellow top+bottom), the expanded DoF encapsulates larger space with Δp resolution and produces higher resolution outside DoF, i.e. green point.

Alternatively, spherical wave illuminated scene geometry can be viewed as miniaturized setup under plane wave illumination. We consider a magnifying glass placed right after the SLM with focal length d_F and the pupil locates at the back focal plane. In specific, scene reconstructed under spherical wave illumination can be considered as a miniature scene lying in between SLM and front focal plane reconstructed under plane wave illumination. Let Δx_{min} be the spatial resolution at z_0 in front of SLM under plane wave illumination. When $z_0 \leq d_F$, a virtual image of spatial resolution $\frac{d_F}{d_F - z_0} \Delta x_{min}$ is created at $\frac{d_F z_0}{d_F - z_0}$, and the angular resolution is obtained from the angle subtended by the spatial resolution of the virtual image to the pupil:

$$\Delta\alpha = \left(\frac{d_F}{d_F - z_0} \Delta x_{min} \right) / \left(\frac{d_F z_0}{d_F - z_0} + d_F \right) = \Delta x_{min} / d_F. \quad (3.35)$$

When $z_0 \leq z_{min}$, we have $\Delta x_{min} = \Delta p$ and the maximum angular resolution is obtained as $\Delta\alpha_{max} = \Delta p / d_F$. If $z_{min} > d_F$, all scene points will be reconstructed at highest angular resolution $\Delta\alpha_{max} = \Delta p / d_F$, i.e., infinite depth of field.

3.3.3 Elemental Image Resolution & Hogel Partitioning

The elemental image resolution gives the angular sampling rate under spherical wave. We derived the maximum angular resolution $\Delta\alpha_{max} = \Delta p / d_F$ (Eq. (3.35)); human visual acuity or the angular resolution determine the required upper bound on the angular sampling rate $\beta = \max((1/60)^\circ, \Delta\alpha_{max})$. For a hogel partition of width w_h , the maximum hogel viewing cone φ is given by:

$$\varphi = 2 \tan^{-1} \left(\frac{w_h}{2d_{c_z}} \right) = 2 \tan^{-1} \left(\frac{w_{e(d_F)} + w_h}{2d_F} \right), \quad (3.36)$$

and the maximum elemental image resolution is obtained as:

$$k = \varphi / \beta. \quad (3.37)$$

For a 10mm eye box, a 3840×2160 SLM of 4um pixel pitch achieves an angular sampling $4 \times$ the visual acuity, which is too expensive and impractical to render for interactive

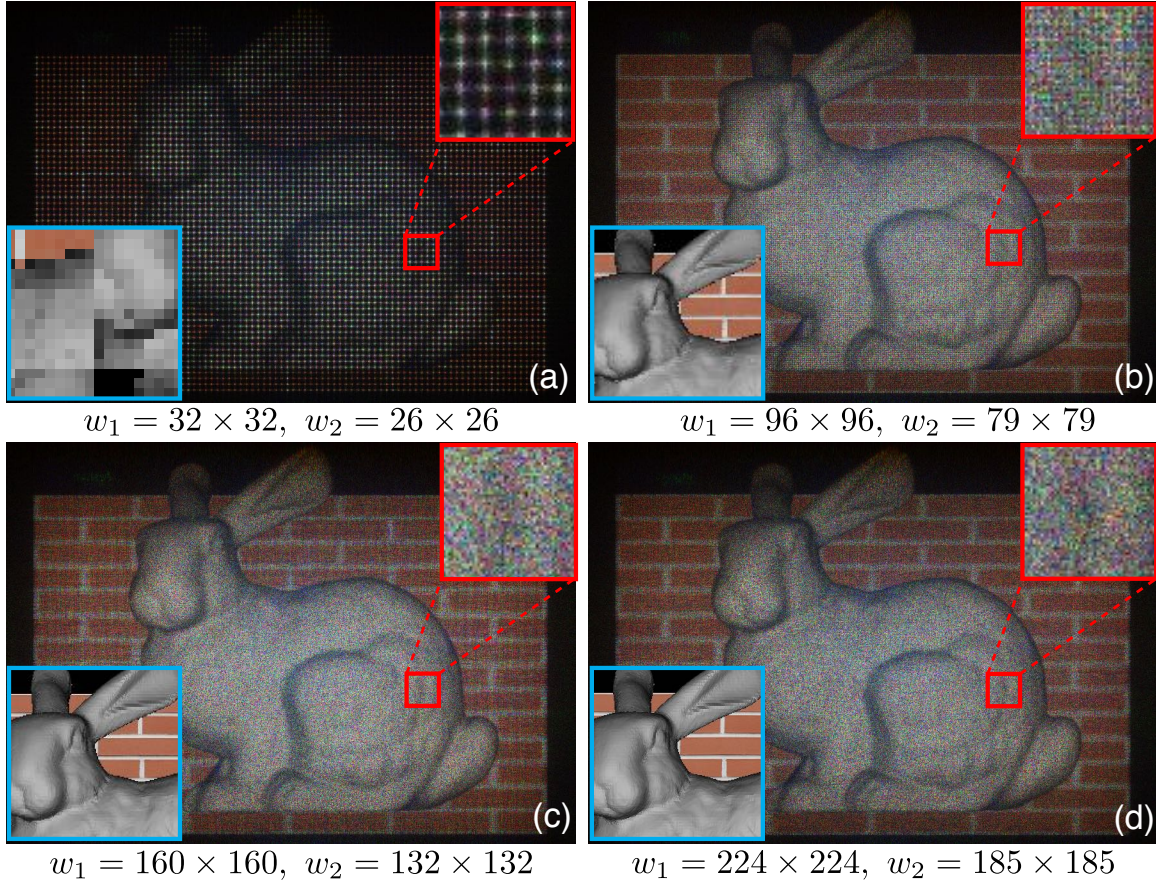


Figure 3-13: Comparisons of elemental image resolution w_1 (blue inset). We simulate focusing on the bunny on a 1080p $8\mu m$ CGH. The angular sampling rate is (a): 6, (b): 18, (c): 30, and (d): 45 pixels per degree respectively. The reconstructions exhibit obvious aliasing on the top row (red inset) but are smoothed out greatly above 30 pixels per degree (c).

applications. In Fig. 3-13, we find the threshold above 30 pixels per degree gives good approximation without strong perceivable aliasing.

Hogel Partition Choosing a small hogel size and dense partitioning increases the number of views need to be rendered, but reduces the ambiguity region (Eq. (3.17)) and produces accurate perspectives for intra-ocular occlusion. To balance, we evaluate the hogel size w_h based on the ratio between the error-free segment w_3 and the approximated view sliding window w_2 . Choosing $w_h \approx 1mm$ in our prototypes reduces the ambiguity region to less than 0.16% for a two-dimensional view and gives a 16×9 hogel partitioning. Larger pixel pitch requires denser hogel partitioning.

3.3.4 Practical Design Trade-off

Using spherical illumination opens a design trade space not possible to plane wave illuminated CGH: we decouple the requirement to increase the FoV with the same eye relief without simultaneously increasing the display panel and decreasing the pixel pitch (Eq. (3.12)).

We show the design space in Fig. 3-14 for spherical wave illuminated holographic displays according to Eq. (3.12), assuming a minimum eye box $w_e = 10mm$, wavelength $\lambda = 532nm$, and a f -number limited to one; the f -number characterizes the ratio of the converging focal length of the lens to the SLM width ($f\# = \frac{d_F}{w_s}$). Small f -number or small eye box creates even wider FoV as shown in [175]. We plot off-the-shelf SLMs in the figure; SLMs with pitch $\leq 3\mu m$ or 8K resolution exists [26].

For a given pixel pitch Δp and eye box w_e , we also derive the bound below which FoV and eye relief are solely limited by maximum diffraction angle. The bound for the SLM width is give by

$$\overline{w_s} = \left(\frac{\tan \left(\sin^{-1} \left(\sin \left(\tan^{-1} (f\#/2) \right) + \frac{\lambda}{2\Delta p} \right) \right)}{\tan \left(\sin^{-1} \left(\frac{\lambda}{2\Delta p} \right) \right)} - 1 \right) w_e. \quad (3.38)$$

As shown in Fig. 3-14, the SLM width in white dash line separates the pitch-bounded case from the the lens-bounded case (shaded region below dash line).

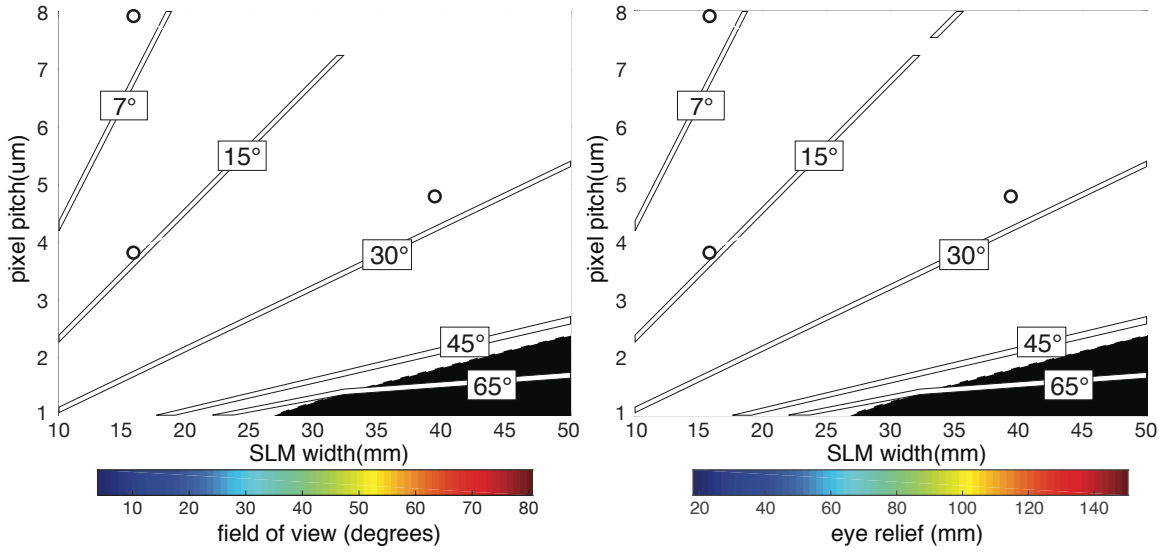


Figure 3-14: Design trade space between the pixel pitch Δp and the SLM width w_s assuming eye box $w_e = 10mm$ and $f\# = 1$. (Left) Maximum FoV increases with smaller pitch and wider SLM. The white dash line separates the above pitch bounded case from the lower lens bounded case(shaded region). (Right): Minimum Eye relief(focal length) decreases with pitch size (Eq. (3.12)) for a fixed eye box requirement, and the current off-the-shelf SLM is quite far to the lens-bounded region.

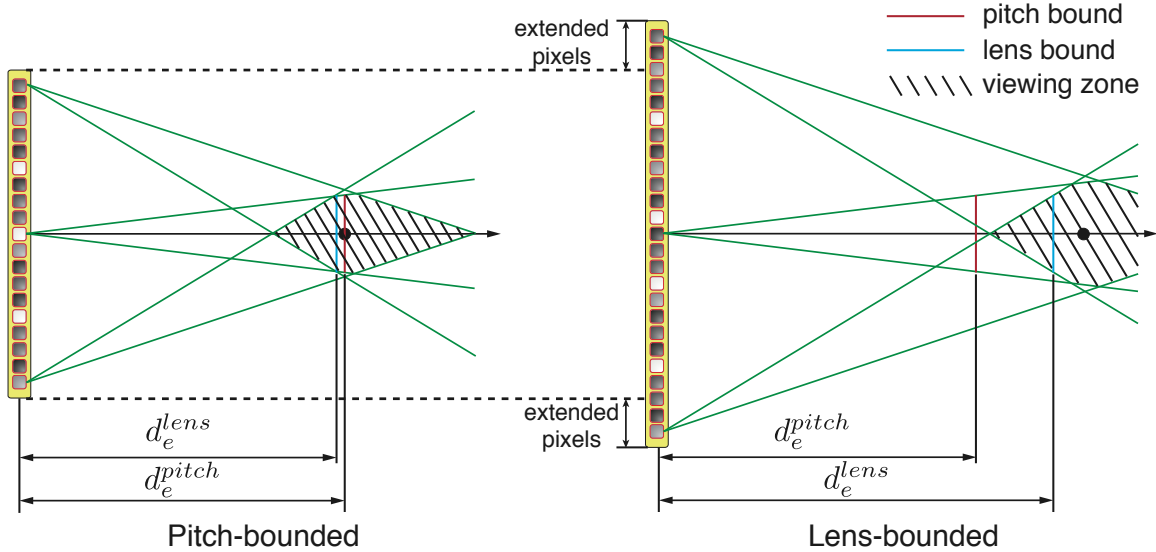


Figure 3-15: Illustration of (left): pitch-bounded and (right): lens-bounded display modes. When the spherical illumination is constrained by the $f\#$ (right), the diffraction cones emitted from the edges of the SLM shift away, and eye relief increases on a wider SLM.

To understand the bounds, we distinguish the two different cases when the maximum FoV and shortest eye relief are bounded by either the pitch pixel (maximum diffraction angle, Fig. 3-15 (left)) or the lens ($f\#$, Fig. 3-15 (right)). Given a required eye box w_e , the minimum eye relief d_e^{pitch} enforced by the central cone is solely determined by the pixel pitch:

$$d_e^{pitch} = \frac{w_e}{2 \tan \left(\sin^{-1} \left(\frac{\lambda}{2\Delta p} \right) \right)}. \quad (3.39)$$

On the other hand, the minimum lens-bounded eye relief d_e^{lens} enforced by the peripheral cones is jointly determined by the SLM width w_s , focal length d_F and pixel pitch Δp according to Eq. (3.3) and similar triangle relation:

$$d_e^{lens} = \frac{w_s + w_e}{2 \tan \left(\sin^{-1} \left(\sin \left(\tan^{-1} \left(\frac{w_s}{2d_F} \right) \right) + \frac{\lambda}{2\Delta p} \right) \right)}. \quad (3.40)$$

Equation (3.40) makes two assumptions: the eye box w_e is bounded by the longitude width of the rhombus bound of the two margin cones, and f -number is not limited to one. It is easily shown that when $d_e^{pitch} = d_F$ and the focal length d_F is valid ($f\# \geq 1$), the eye box w_e is always feasible and $d_e^{lens} \leq d_e^{pitch}$ is always satisfied, and thus is pitch-bounded (Fig. 3-15 (left)).

A wider SLM (Fig. 3-15 (right)) with the same pixel pitch has the same pitch-bounded eye relief d_e^{pitch} , but the lens-bounded eye relief extends to d_e^{lens} due to the $f\# \geq 1$ limitation. Substituting the $f\#$ in Eq. (3.40) and equating Eqs. (3.39) and (3.40) gives the maximum pitch-bounded SLM width $\overline{w_s}$ (Eq. (3.38)).

A compound microscopic pupil-forming design or a holographic optical element (HOE) based design allows for a higher angular magnification in trade for a smaller eye box. With such technique, tiling three GAEA SLMs allows for a 66.7° FoV and $35mm$ eye relief with a $5mm$ eye box. In summary, smaller pitch or larger SLM improves the visual experience under spherical illumination.

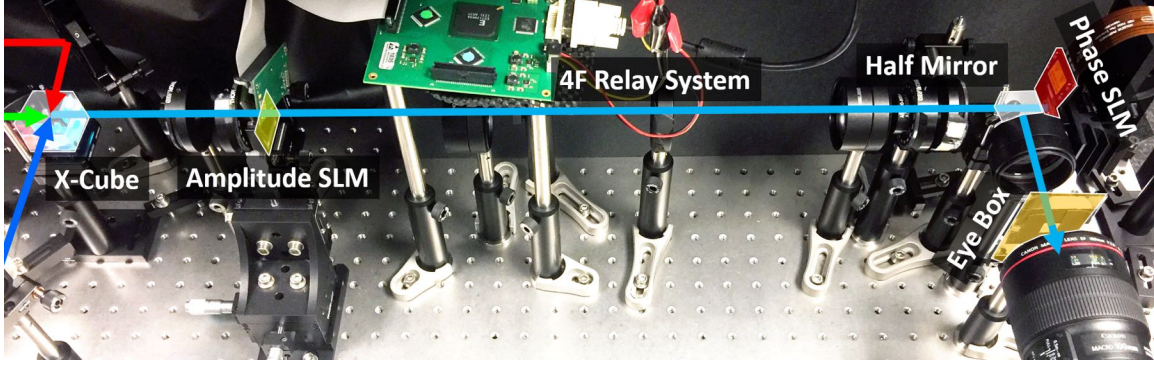


Figure 3-16: Optical setup relaying amplitude modulation to phase modulation using field sequential colors. The eye box also filters higher order diffractions.

3.4 Experiment

3.4.1 Experimental Setup

Our benchtop optical setup is shown in Fig. 3-16 with three incoming narrow band RGB laser at wavelength $642nm$, $515nm$, and $442nm$ respectively. We modulate the RGB beams in field-sequential color, and relay the amplitude SLM to the phase SLM via a 4F system. The amplitude SLM has 1920×1080 transmissive $8.5\mu m$ pixel made by bbs bild und lichtsysteme (part# LCD L3C07U-85G13). The phase SLM is 4094×2464 resolution $3.74\mu m$ pitch Liquid Crystal on Silicon (LCoS) by HoloEye (part# GAEA). The hologram is computed at 4K resolution, but the amplitude is down-sample to 1080p using nearest neighbor filtering for display. We capture the photograph with a Canon EOS 5D MIII DSLR camera with a $100mm$ f/2.8 lens.

A Thorlab $75mm$ achromatic lens is placed at $50mm$ after the phase SLM and before the eye box. The lens effectively converts the plane wave illumination to a converging illumination focused at the back focal point centered at the eye box. Our rendering and calculation are adjusted with respect to the magnified virtual SLM and the converging spherical illumination.

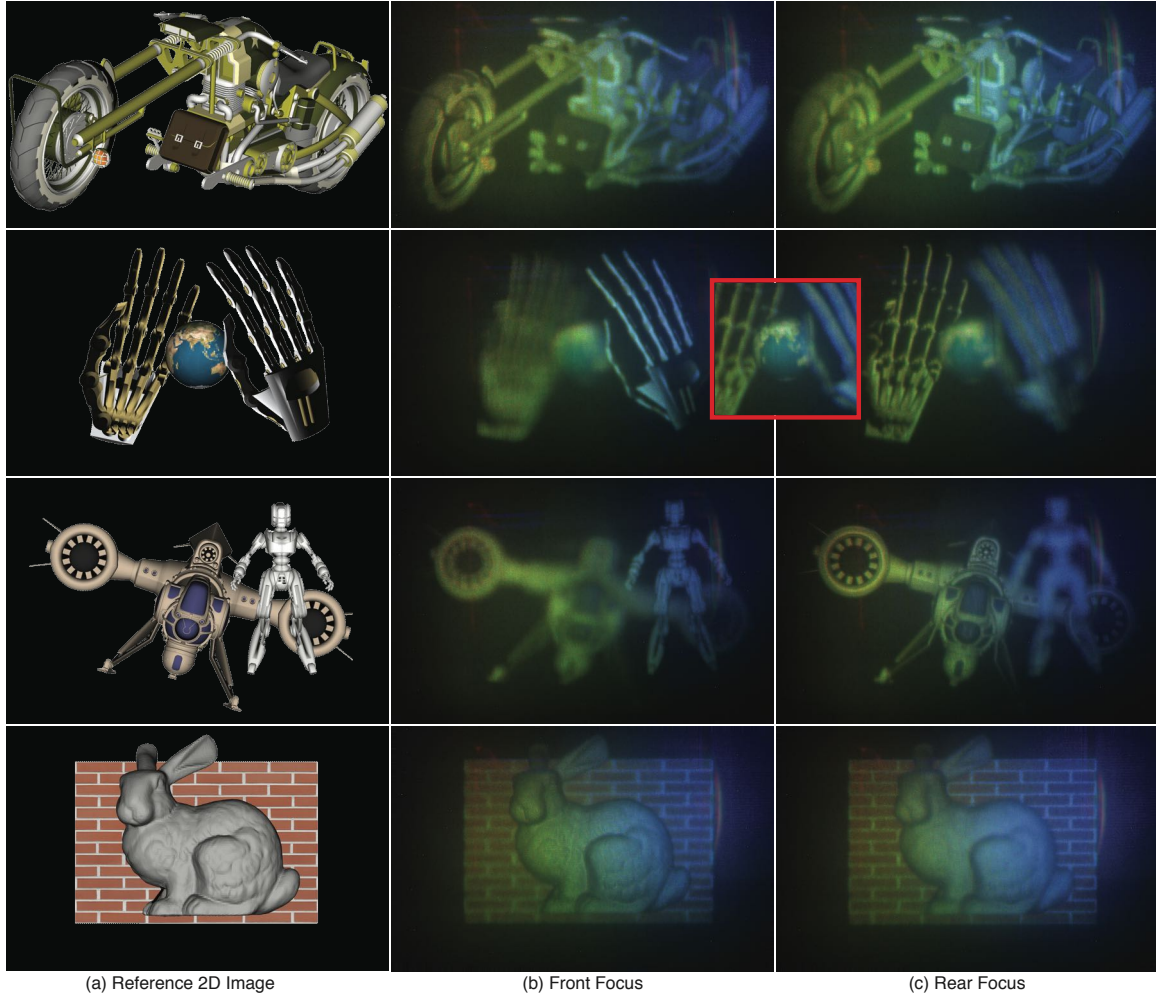


Figure 3-17: Optically captured hologram results at different focal distances. The color artifacts is caused by the laser filtering, alignment, and inprecise beam stop, but does not affect the correctness. We show the reference 2D image in (a). When the object is displayed on a hologram, we can support accommodation (b and c).

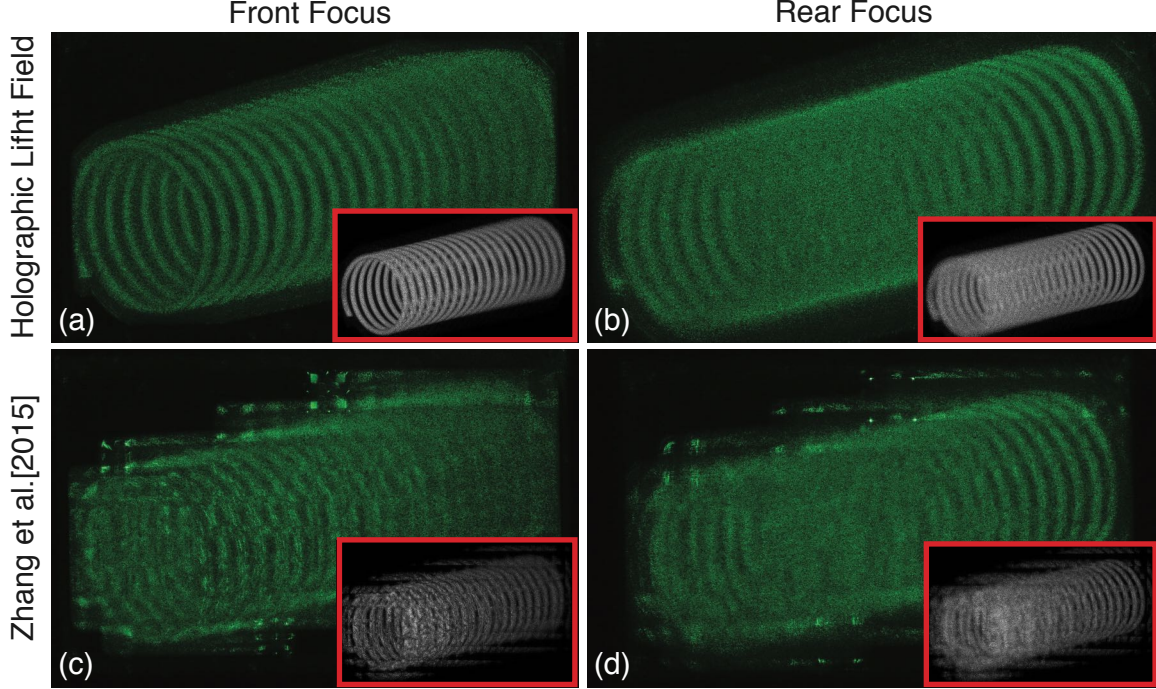


Figure 3-18: The proposed method (top row, photographed) provides an accurate rendering setup (Sec. 3.2.1) such that no geometry is clipped near the SLM nor aliased reconstruction shown in prior work (c and d). The captured results from a 1080p SLM closely resemble the simulations (red insets).

3.4.2 Experimental Evaluation

We verify our rendering and CGH calculation using the abovementioned optical setup; photographed results are shown in Fig. 3-17. Color artifacts are due to laser filtering, alignment, and imprecise beam stop, and do not affect the correctness. Our method reconstructs per-pixel depth that allows for continuous focusing, as shown in the inset for the second row example. High frequencies in geometry and texture are well preserved as shown in the third and fourth row examples.

The proposed method avoids geometry clipping (Fig. 3-3, bottom) and aliasing due to over- and under-sampling outside the central viewing frustum of the hogel in prior work. We visualize the comparison in Fig. 3-18. Strong distortion occurs to the front of the spring due to geometry clipping (c). While the rear geometry reconstructs well, over- and under-sampling still introduce additional aliased points (d), as analyze in Sec. 3.3.1.

With diffuse object (i.i.d random phases), the resulted hologram amplitude becomes

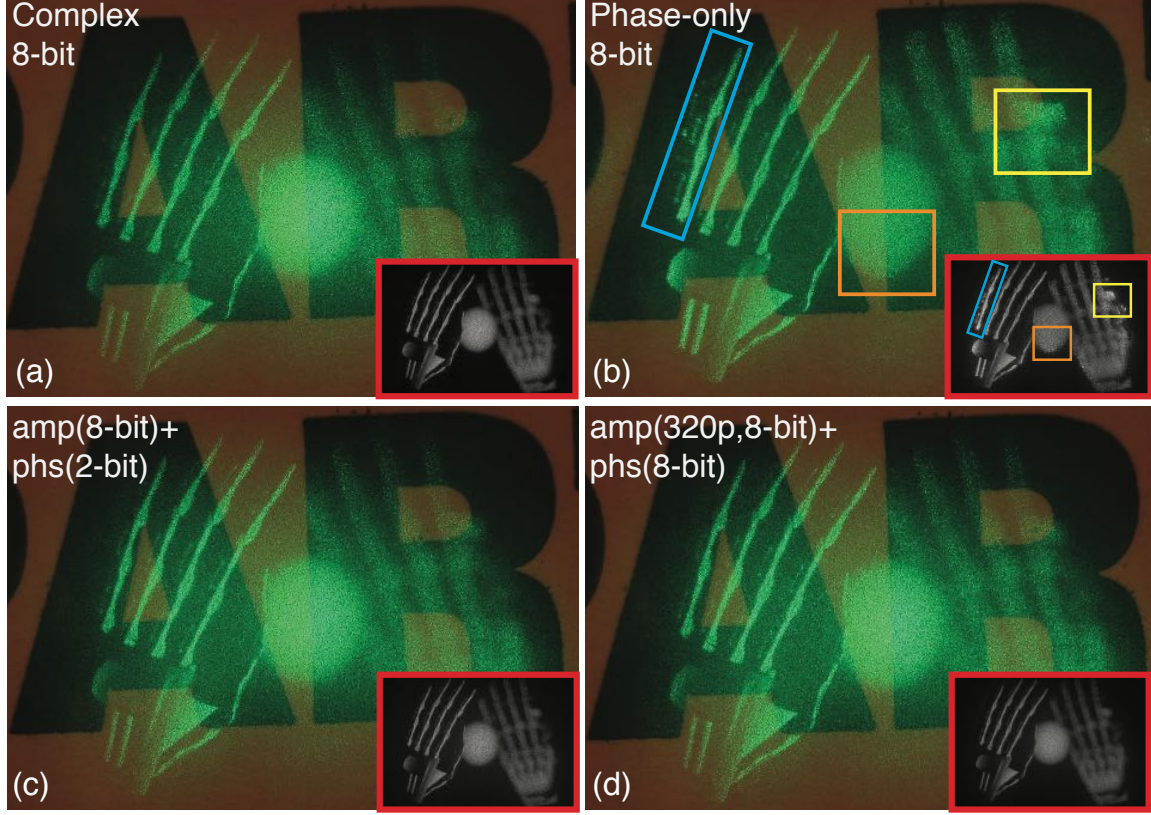


Figure 3-19: Comparisons of (a): the ideal amplitude and phase hologram with (b): phase-only hologram, (c): lower phase bit rate, and (d): lower amplitude resolution. We compare the simulated results (insets) using a 1080p SLM.

relatively uniform and most information is transferred to the phase [92]. Since most prior work only use phase, we also reconstruct with a phase-only hologram, as shown in Fig. 3-19(b), and it closely resembles the reference result (a) with minor artifacts marked in blue, orange and yellow. Adding a low-resolution amplitude mitigate these artifacts significantly (d). The bit-level of phase precision can also be reduced [280] for other purposes, e.g. faster refresh or more pixels, without noticeably sacrificing the image quality. The image reconstructs well with a slight contrast loss even with 2-bit quantization (c).

3.5 Discussion

Plane wave-illuminated CGH encounters limitations when applied to near-eye displays; we advocate using spherical wave to break many constraints. We show an accurate ren-

dering algorithm by offsetting the camera and using a sliding window to provide a pixel-accurate rendering perspective, and we extend the algorithm to spherical wave illuminated rendering. Computing the Fresnel integral on GPU provides over five orders of magnitude performance compared to previous work, making interactive CGH possible. We derive theoretical resolution limits and trade-off analysis for designing a light field-based holographic near-eye display in-depth. In this work, we show the first practical near-eye CGH for high resolution at interactive rates, modeling intra-pupil occlusion and providing continuous focus cue with per-pixel depth. Below we discuss limitations and future work.

Dense Angular Sampling Holograms are capable of showing highly view-dependent effects like occlusion, specular, or mirror surfaces; these are critical in VR or AR to perceive distances and depths [268, 245]. Our rendering considers sparse views at hogel width intervals, and is imprecise for highly specular material. Spectral analysis on material could help determine the increased sampling needed and possibly support non-uniform sampling density and hogel partitioning.

Simultaneous Amplitude-Phase Modeling In this work, we use both amplitude and phase SLMs. Alternatively, the double phase or super pixel methods enable simultaneous of amplitude and phase modulation by sacrificing spatial resolution. In Chapter 4, we discuss the application of double phase method for high-quality hologram generation using a single phase SLM.

Correcting Individual Aberrations Wearing a corrective lens inside the near-eye display is cumbersome. Light field displays can incorporate personalized inverse aberrations [151, 123]. In Chapter 4, we discuss how to incorporate vision aberration correction into a holographic display system.

Wavelength Mixing Most CGH use field sequential colors, but multiplexing multiple wavelengths has been shown [90, 158]. Content-adaptive optimization for the entire hologram can increase the frame rate two to three times. In Chapter 5, we devise such a

framework for generating multi-wavelength holograms for high dynamic range and high framerate holographic projection.

Chapter 4

Deep Learning Holography

Creating photorealistic CGHs power-efficiently and in real time remains an unsolved challenge in computational graphics. The primary challenge is the tremendous computational cost required to perform Fresnel diffraction simulation for every object point in a continuous 3D space. Although the introduction of General-Purpose Graphical Processing Units (GPGPUs) and Field Programmable Gate Arrays (FPGAs) has reduced computation time to several seconds with parallel computing (as was discussed in the previous chapter), achieving real-time applications would still require more than an order of magnitude increase in processing speed.

In this chapter, we discuss how deep learning, a technology that shares surprisingly similar highs and lows, hopes and disappointments with holography, may help bridge this performance gap when creating CGHs for virtual and augmented reality applications. In particular, we make a key observation that if CGHs can be computed in real-time, at any particular moment in time, the active CGH only needs to reflect the current point of view accurately. As a result, scene representations that are significantly more data and rendering efficient than conventional light fields as input for CGH computation. Based on that, we investigate methods to enhance physically-based CGH simulation techniques and devise large-scale datasets to expedite the training of neural network surrogate models, vastly accelerating computation speeds. Additionally, we introduce strategies for efficiently converting complex holograms to phase-only holograms and correcting users' vision aberrations, empowering neural networks to perform these tasks effectively. Moreover, in Sec. [4.3](#), we

present a novel neural compression algorithm tailored for hologram images and videos, enabling efficient encoding and streaming capabilities.

4.1 Tensor Holography: Real-Time 3D Holography with Deep Neural Networks

4.1.1 Motivation

Holograms are renowned for their high information capacity, making them highly versatile for encoding data that ranges from simple 2D images to complex 3D images with full parallax (i.e., both vertical and horizontal). In practice, the amount of data required to load into a hologram depends on its intended use. For instance, when a non-refreshable (static) hologram is used as a 3D visual artwork installation, it needs to support viewing from various angles by multiple individuals, with each person perceiving the correct perspective. Computing a CGH that fulfills this purpose would require explicitly modeling and ensuring all possible perspectives match, which would incur extremely high rendering costs.

For near-eye display applications, the required information is considerably relaxed as each user has a dedicated display for their virtual graphics, and multiview support can be dropped. The headset enclosure/frame and the dimension of the microdisplay also limit the users' views, restricting them to see only a portion of the entire virtual environment. Thus, existing stereoscopic near-eye displays use inertial measurement units (IMU) to track user motion and only render virtual content within the users' field of view in real-time.

To integrate CGH support, this pipeline would only require an additional step of converting the instantaneous 3D view a user sees (e.g., represented by a single RGB+depth image) into a CGH that encodes the same 3D view. If reducing the required information leads to a significantly faster CGH calculation, the augmented rendering pipeline may still maintain real-time performance, holding the promise to be integrated into the next-generation headsets. However, in practice, even with parallelization of CGH calculation on the state-of-the-art GPU (as discussed in the previous chapter), the most efficient physically-based algorithms is still more than an order away from achieving real-time results.

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[173, 172, 171], multi-layer depth discretization[44, 281, 178], holographic stereograms[261, 18, 272, 200], wavefront recording plane (alternatively intermediate ray sampling planes)[229, 250] were introduced at a cost of compromised image quality. As a result, finding ways to significantly accelerate the calculation process without compromising the precision of CGH has become one of the most sought-after challenges.

To tackle this challenge from an entirely new perspective, we make another crucial observation that the 3D CGH calculation is essentially a spatially-varying large kernel convolution operation. This can be effectively approximated at a significantly reduced cost by applying a series of small convolution kernels. This observation naturally leads to the application of convolutional neural networks (CNN), which operate on a very similar principle. To accurately reflect the reality, CGH calculation also needs to account for occlusions of wavefronts by foreground objects. Physically-based methods implement this task by detecting scene occlusions through ray tracing and checking for intersections during forward propagation (see Sec. 4.1.2) or by dividing the scene into depth layers and performing layer-by-layer spatial masking based on active pixel locations (see Sec. 4.2.2). However, both approaches add substantial computation overhead. Fortunately, the non-linear activation layers included in CNNs can handle this task efficiently without incurring additional computation burden.

The above analysis sketches the preliminary evidence that CNNs can replace physical simulations as surrogate models and provide savings on computation cost. However, fulfilling this promise requires careful design of companion physically-based algorithms, datasets, CNN architectures, and training procedures. In the following sections, we introduce our deep-learning-based solution for CGH rendering for near-eye displays, dubbed *Tensor Holography*. The project code can be found at:

https://github.com/liangs111/tensor_holography

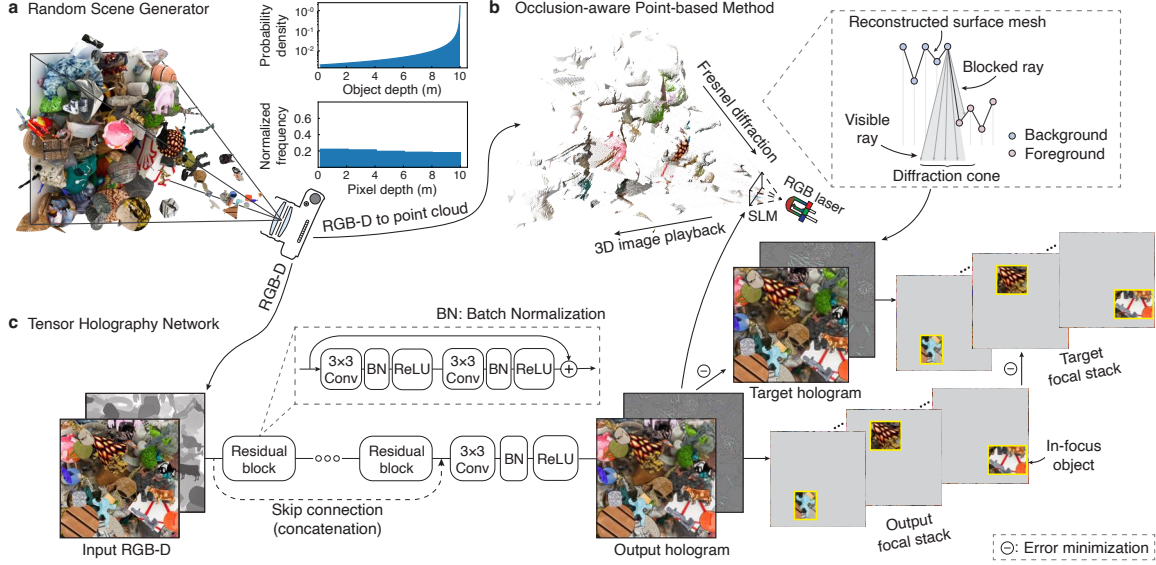


Figure 4-1: **Tensor holography workflow for learning Fresnel holograms from RGB-D images.** **a**, A custom ray-tracer renders an RGB-D image of a random scene. The meshes are distributed exponentially along the depth axis and the resulted pixel depth distribution is statistically uniform. **b**, An occlusion-aware point-based method 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.

4.1.2 Occlusion-Aware Point-Based Method (OA-PBM)

The performance of data-driven methods heavily relies on the quality of training data. Consequently, we first devise an accurate physical simulation method to generate photorealistic holograms. To calculate 3D (Fresnel) holograms, the vanilla point-based method (PBM) treats each 3D point as an independent, self-emitting spherical source without considering their occlusion relationships. In the prior chapter, this problem is largely resolved from the scene representation step through light field rendering to capture the correct visibility of the scene from different perspectives. In contrast, if PBM was directly applied to a point cloud specified by a single RGB-D image, artifacts due to penetrating background points would be visible in the reconstructed hologram. This is because the wavefront emitted by distant point sources may be partially obstructed by nearer ones (see Fig. 4-2d top row), but naive PBM does not account for this occlusion.

We introduce occlusion-aware point-based method (OA-PBM) to handle occlusion at the calculation step by augmenting the vanilla PBM with occlusion detection. Instead of processing each 3D point independently, OA-PBM reconstructs a triangle surface mesh from the RGB-D image and performs ray casting from each vertex (point) to the hologram plane (see Fig. 4-1b). Wavefronts carried by the rays intersecting the surface mesh are excluded from hologram computation to account for foreground occlusion. Consider the same notation and definitions used in Eq. (3.1), 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 $\vec{o\mathbf{p}}$ 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 $Q_{\vec{o\mathbf{p}}}$ that may lie on the path of $\vec{o\mathbf{p}}$. Let $\mathbf{p}_{ol}$ be the SLM pixel intersecting $\vec{o\mathbf{l}}$ (pixel at $\mathbf{o}$'s sub-hologram center), the set $Q_{\vec{o\mathbf{p}}}$ only consists of triangles whose vertices' xy coordinate indices are on the path of line segment $\overleftrightarrow{\mathbf{p}\mathbf{p}_{ol}}$, and

$$v_o(\mathbf{p}) = \neg \left[\bigvee_{q \in Q_{\vec{o\mathbf{p}}}} \vec{o\mathbf{p}} \text{ intersects } q \right] \quad (4.1)$$

Finally, the object wave at the hologram plane H_{target} is obtained by summing sub-holograms contributed from all object points

$$E_{\text{obj}}(\mathbf{p}) = \sum_{j \in S_p} v_j(\mathbf{p}) h_j(\mathbf{p}) \quad (4.2)$$

where S_p is the set of object points whose sub-holograms are defined at $\mathbf{p}$, and the final target hologram is given by $H_{\text{target}} = E_{\text{obj}} E_{\text{ref}}^*$. Figure 4-2b visualizes the masked Fresnel zone plate computed for different depth landscapes. Comparing to PBM, OA-PBM significantly reduces background leakage (see Fig. 4-2d). It is important to note that OA-PBM is still a first-order approximation of the Fresnel diffraction, and the hologram quality could be further improved by modeling 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. In Sec. 4.2.2, we address this challenge with silhouette-mask layer-based method.

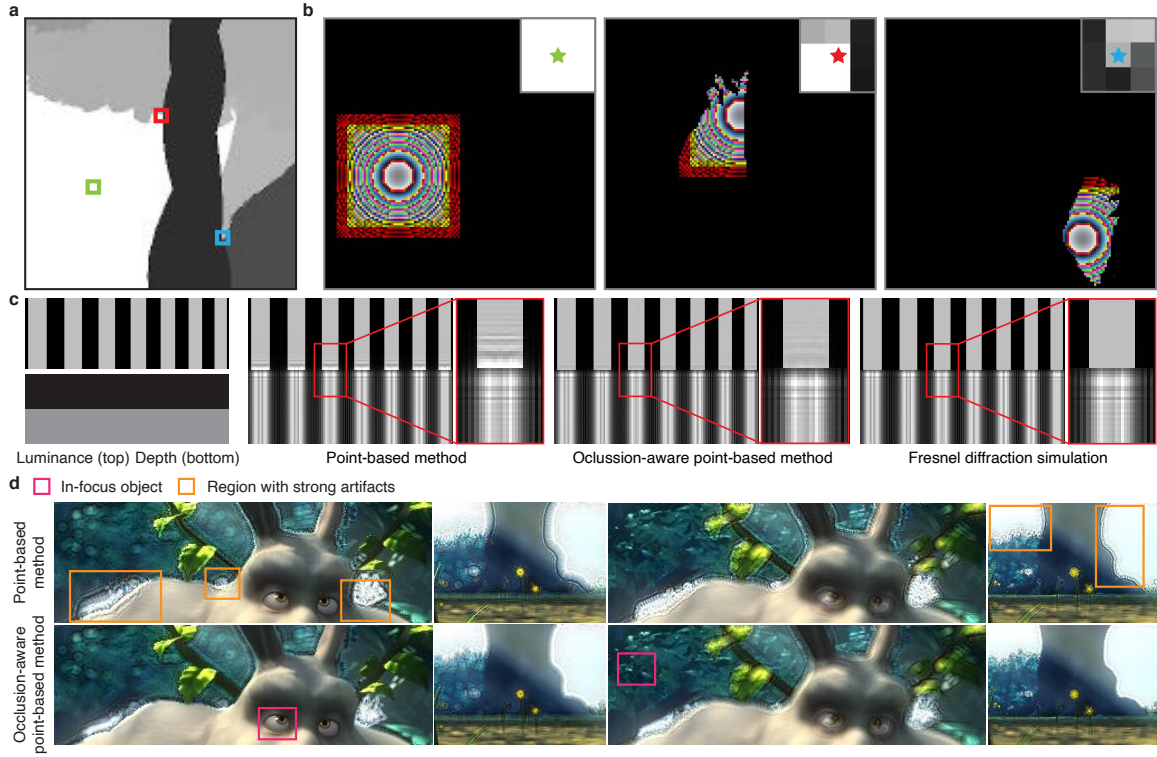


Figure 4-2: Visualization of masked Fresnel zone plates computed by OA-PBM and performance comparison of foreground occlusion. **a**, A depth image cropped from a frame of Big Buck Bunny. Three regions with different depth landscapes are highlighted in different colors. **b**, Masked Fresnel zone plates computed for the center 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 PBM, OA-PBM, and Fresnel diffraction. The scene is a cropped modulation transfer function bar target with a step depth profile. PBM leaks a considerable portion of the background into the foreground due to a lack of occlusion handling. The artifacts are clearly visible in the original unmagnified view. OA-PBM removes a significant portion of the artifacts and the remaining artifacts are visually insignificant in the unmagnified view. **d**, Comparison of focal stacks reconstructed by PBM and OA-PBM for the Big Buck Bunny scene. The orange bounding boxes mark the background leakage in the PBM reconstructions.

4.1.3 MIT-CGH-4K: the First Large-Scale Hologram Dataset

To facilitate training CNNs as neural wave propagation surrogates, we introduce the first large-scale 3D (Fresnel) hologram dataset, MIT-CGH-4K, consisting of 4000 pairs of RGB-D images and corresponding 3D holograms. Our dataset is created with two 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 color, geometry, shading, texture, and occlusion to help the CNN generalize to both computer-rendered and real-world captured RGB-D test inputs. This is achieved by a custom random scene generator (see Fig. 4-1a), 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 60000 textures from publicly available texture synthesis datasets [51, 55] with augmentation. Second, the pixel depth distribution of the resulted 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) (Eq. (4.7)) for arranging triangle meshes along the depth axis (z -axis). We provide the derivation below.

Starting 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 $z \in [z_{\text{near}}, z_{\text{far}}]$,

$$z_d \geq z \Leftrightarrow \min\{z_1, z_2, \dots, z_{C'}\} \geq z \Leftrightarrow \bigwedge_{i=1}^{C'} z_i \geq z \quad (4.3)$$

From a probabilistic perspective,

$$\Pr(z_d \geq z) = \Pr\left(\bigwedge_{i=1}^{C'} z_i \geq z\right) = \prod_{i=1}^{C'} \Pr(z_i \geq z) = (\Pr(z_i \geq z))^{C'} \quad (4.4)$$

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, $\Pr(z_i \geq z) = \int_z^{z_{\text{far}}} f(t) dt$. Thus, Eq. (4.4) can be rewritten into the following form for every $z \in [z_{\text{near}}, z_{\text{far}})$,

$$\int_z^{z_{\text{far}}} f(t) dt = [\Pr(z_d \geq z)]^{\frac{1}{C'}} = \left(\frac{z_{\text{far}} - z}{z_{\text{far}} - z_{\text{near}}} \right)^{\frac{1}{C'}} \quad (4.5)$$

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

$$f(z) = \frac{1}{C'(z_{\text{far}} - z_{\text{near}})} \left(\frac{z_{\text{far}} - z}{z_{\text{far}} - z_{\text{near}}} \right)^{\frac{1}{C'} - 1} \quad (4.6)$$

gives a closed-form solution to the PDF associated to $z_1, z_2, \dots, z_{C'}$.

Although it is required by definition that $C' \in \mathbb{Z}^+$, Eq. (4.6) 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 since 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 modeled by using a scaling factor α such that $C' = C/\alpha$. Eq. (4.7) is thus obtained by applying this equation to (4.6). Based on experimentation, we find setting $\alpha = 50$ results in a sufficiently statistically uniform pixel depth distribution for $200 \leq C \leq 250$. Fig. 4-3 shows a comparison of the resulted 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.

$$f_{\alpha}(z) = \frac{\alpha}{C(z_{\text{far}} - z_{\text{near}})} \left(\frac{z_{\text{far}} - z}{z_{\text{far}} - z_{\text{near}}} \right)^{\frac{\alpha}{C} - 1} \quad (z_{\text{near}} \leq z < z_{\text{far}}), \quad (4.7)$$

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 (see Fig. 4-1a top plot) such that the pixel depth distribution in the resulted RGB-D images is statistically uniform (see Fig. 4-1a bottom plot). Here, we set z_{near} and z_{far} to 0.15 m and 10 m to

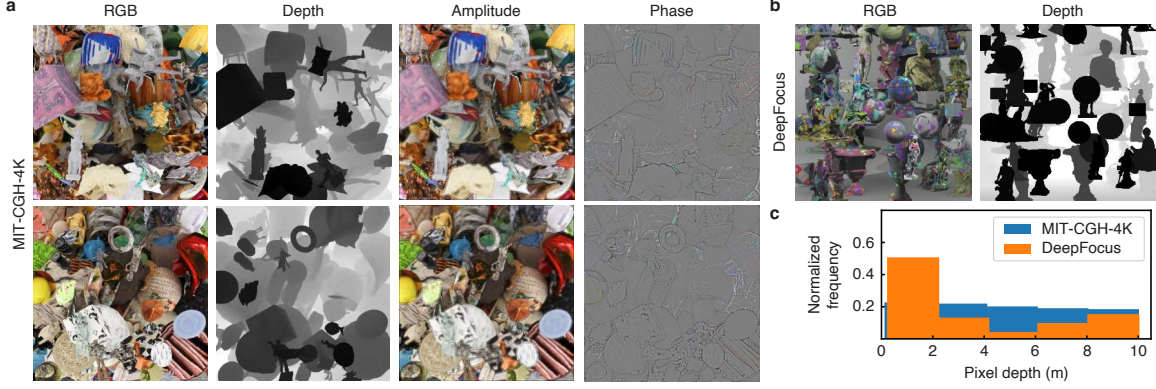


Figure 4-3: **Samples of MIT-CGH-4K and comparison with the DeepFocus dataset.** **a**, The RGB-D image, amplitude, and phase of two samples from MIT-CGH-4K. The RGB image records the amplitude of the scene (directly visualized in sRGB space) and consists of large variations in color, texture, shading, 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 color changes. **b**, A sample RGB-D image from the DeepFocus dataset[258]. **c**, Histograms of pixel depth distribution computed for MIT-CGH-4K and the DeepFocus dataset.

accommodate a wide range of focal distances (approximately a 6.6 diopter range for the depth of field).

The random scene generator is implemented using NVIDIA OptiX ray-tracing library with NVIDIA AI-Accelerated denoiser turned on to maximize customizability and performance. During the construction of scene, we limit the random scaling of mesh such that the longest side of mesh’s bounding box fall within $0.1\times - 0.35\times$ of the screen space height. This prevents a single mesh to be negligibly small or overwhelmingly large. We generated 4000 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 $8\text{ }\mu\text{m}$ and $16\text{ }\mu\text{m}$ pixel pitch SLMs. The image resolution was chosen to be 384×384 and 192×192 pixels respectively to match the physical size of the resultant holograms and enable training on commonly available GPUs. Note that since 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.

4.1.4 CNN Model Architecture, Training, Evaluation, and Comparisons

Our CNN model is similar to DeepFocus[258], a fully convolutional neural network designed for synthesizing image content for varifocal, multifocal, and light field head-mounted displays. It receives a 4-channel RGB-D image and predicts a color hologram as a 6-channel image (RGB amplitude & RGB phase), which can be used to drive three optically combined SLMs or one SLM in a time-multiplexed manner to achieve full color 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. 4-1c for a scheme of the network architecture).

All convolution layers use 3×3 convolution filters. The number of minimally required convolution layers depends on the maximal spatial extent of the sub-hologram W . Quantitatively, successive application of x convolution layers results an effective $3 + (x - 1) \times 2$ convolution. Solving for the maximum sub-hologram width $W = 3 + (x - 1) \times 2$ yields $\lceil (W - 3) / 2 + 1 \rceil$ minimally required convolution layers. 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, 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 plane [229]. It propagates the target hologram to the center 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 Fig. 4-4. 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. 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 (color channel is omitted here), and $d'_{\text{near}}, d'_{\text{far}}$ be the distances from the target hologram to the near

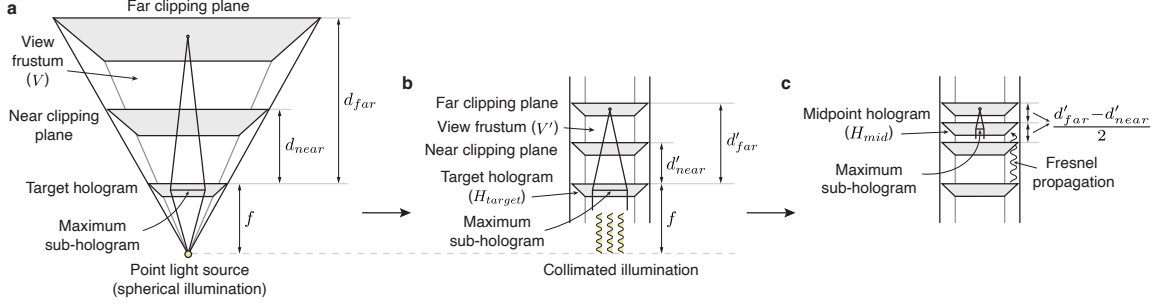


Figure 4-4: **Schematic diagram of the midpoint hologram calculation.** **a**, A holographic display magnified through a diverging point light source. **b**, A holographic display unmagnified through the thin lens formula. **c**, The target hologram in this example is propagated to the center of the unmagnified view frustum to produce the midpoint hologram. The width of the maximum sub-hologram is significantly reduced.

and far clipping plane of V' . H_{target} is propagated for a distance of $d'_{mid} = (d'_{near} + d'_{far})/2$ to the center of V' to form the midpoint hologram $H_{mid} \in \mathbb{C}^{M \times N}$. The angular spectrum method [185] (ASM) is employed to model the propagation of a wave field:

$$H_{mid}(m, n) = \text{ASM}(H_{target}, d'_{mid}) = F^{-1} \left\{ F \{ H_{target} \} e^{i2\pi d'_{mid} \sqrt{\lambda^{-2} - (m/L_w)^2 - (n/L_h)^2}} \right\}. \quad (4.8)$$

Here, F , F^{-1} are the Fourier and inverse Fourier transform operator; L_w , L_h are the physical width and height of the hologram; λ is the wavelength; $m = -M/2, \dots, M/2 - 1$ and $n = -N/2, \dots, N/2 - 1$, where M and N are the number of pixels along the width and height of the hologram. Replacing the target hologram with the midpoint hologram reduces W by a factor of $d'_{far}/\Delta d'$, where $\Delta d' = (d'_{far} - d'_{near})/2$. 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 $-d'_{mid}$. In our rendering configuration, where the collimated frustum V' has a 6 mm optical path length, using midpoint hologram as CNN's learning objective minimizes the convolution layers to 30.

We introduce two wave-based loss functions to train 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 output 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 $H_{\text{mid}} = A_{\text{mid}} e^{i\phi_{\text{mid}}}$:

$$l_{\text{data}} = \left\| \tilde{A}_{\text{mid}} - A_{\text{mid}} e^{i \overbrace{(\delta(\tilde{\phi}_{\text{mid}}, \phi_{\text{mid}}) - \bar{\delta}(\tilde{\phi}_{\text{mid}}, \phi_{\text{mid}}))}^{\text{corrected phase difference}}} \right\|_2, \quad (4.9)$$

where $\delta(\tilde{\phi}_{\text{mid}}, \phi_{\text{mid}}) = \text{atan2}(\sin(\tilde{\phi}_{\text{mid}} - \phi_{\text{mid}}), \cos(\tilde{\phi}_{\text{mid}} - \phi_{\text{mid}}))$, $\bar{\cdot}$ 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. Since ASM-based wave propagation is a differentiable operation, the loss is modeled 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,

$$l_{\text{pcp}}(t) = \underbrace{\sum_{d'_t \in \{D_t^{\text{fix}} \cup D_t^{\text{float}}\}}}_{\text{dynamic focal stack}} \left\| \underbrace{e^{\beta(2\Delta d' - (d'_t - D'_t))}}_{\text{attention mask}} \left(\underbrace{|\text{ASM}(\tilde{H}_{\text{mid}}, d'_t)|^2 - |\text{ASM}(H_{\text{mid}}, d'_t)|^2}_{\text{image difference}} \right. \right. \\ \left. \left. + \underbrace{\nabla |\text{ASM}(\tilde{H}_{\text{mid}}, d'_t)|^2 - \nabla |\text{ASM}(H_{\text{mid}}, d'_t)|^2}_{\text{total variation}} \right) \right\|_1. \quad (4.10)$$

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; β is a user-defined attention scale; $D_t^{\text{fix}}, D_t^{\text{float}}$ are two sets of dynamic focal distances calculated as follows: (i) V' is equally partitioned into T depth bins, (ii) D_t^{fix} picks the top- k_{fix} bins from D'_t 's histogram and D_t^{float} randomly picks k_{float} bins among the rest, (iii) 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 CNN to learn true 3D holograms. The attention mask directs CNN to focus on reconstructing in-focus features in each depth-of-field image. Figure 4-5f validates the effectiveness of each training loss component through an ablation study.

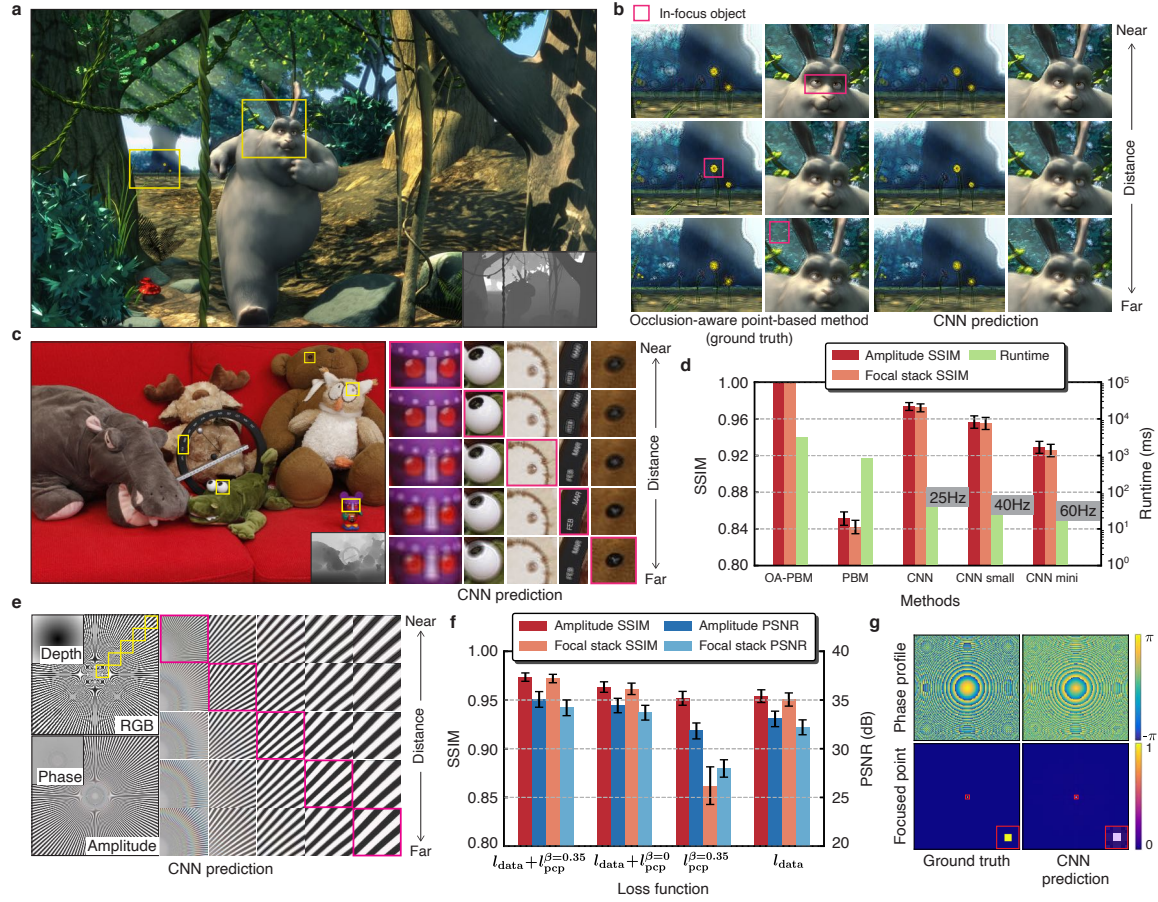


Figure 4-5: Performance evaluation of occlusion-aware point-based method and tensor holography CNN. **a**, A simulated depth-of-field image refocused from a CNN predicted hologram. The bunny's eye is in focus. The input RGB-D image is from the "Big Buck Bunny" (© by Blender Foundation). The bottom right inset visualizes the depth image. **b**, Comparison of focal stacks reconstructed at highlight 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 image [139]. **d**, Performance comparison of PBM, OA-PBM, and CNNs with various model capacities. The default CNN model consists of 30 convolution layers / 24 filters per layer, the small and mini models has 15 and 8 convolution layers respectively. The reduction of convolution layers gracefully degrades the reconstructed image quality. The mini model runs in real time (60Hz). **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 (1st). The ablation of attention mask (2nd) dilutes CNN's attention to out-of-focus features and results in inferior performance. The ablation of data loss (3rd) removes the regularization of phase information and leads to poor generalization to unseen examples and large focal stack error. The ablation of perceptual loss (4th) removes the guide of focal stacks and uniformly degrades the performance. **g**, Comparison of a ground truth Fresnel zone plate and a CNN prediction (by a model with 30 layers / 120 filters per layer) computed for a 6 mm distant point (propagated for another 20 mm for visualization).

Our CNN is trained on 384×384 pixels 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 1000 epochs using Adam[142] optimizer at a constant learning rate of 1×10^{-4} . The dataset is partitioned into 3800, 100, and 100 samples for training, testing, and validation. Training takes 84 hours on a single NVIDIA Tesla V100 GPU. Fig. 4-6 quantitatively compares the performance of our CNN with U-Net[213] and Dilated-Net[266], 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. 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 run time performance is evaluated on 1920×1080 pixels resolution inputs.

The trained model generalizes well to computer-rendered (see Fig. 4-5a and Fig. 4-7), real-world captured (see Fig. 4-5c and Fig. 4-8) RGB-D inputs, and standard test patterns (see Fig. 4-5e and Fig. 4-6). Compared to the reference OA-PBM holograms, the CNN predictions are both perceptually similar (see Fig. 4-5b) and numerically close (see Fig. 4-5d/f). Evaluated on a single distance point target, the output from a CNN with sufficient model capacity faithfully approximates a Fresnel zone plate (see Fig. 4-5g), under the low-rank solution space restricted by a set of successively applied 3×3 convolution kernels. When all algorithms are implemented on GPU with the CNN in NVIDIA TensorRT, and OA-PBM and PBM in NVIDIA CUDA, the mini CNN achieves more than two orders of magnitude speed-up (see Fig. 4-5d) over OA-PBM and runs in real time (60 Hz) on a single NVIDIA Titan RTX GPU. Since 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. For example, leveraging the neural engine of iPhone 11

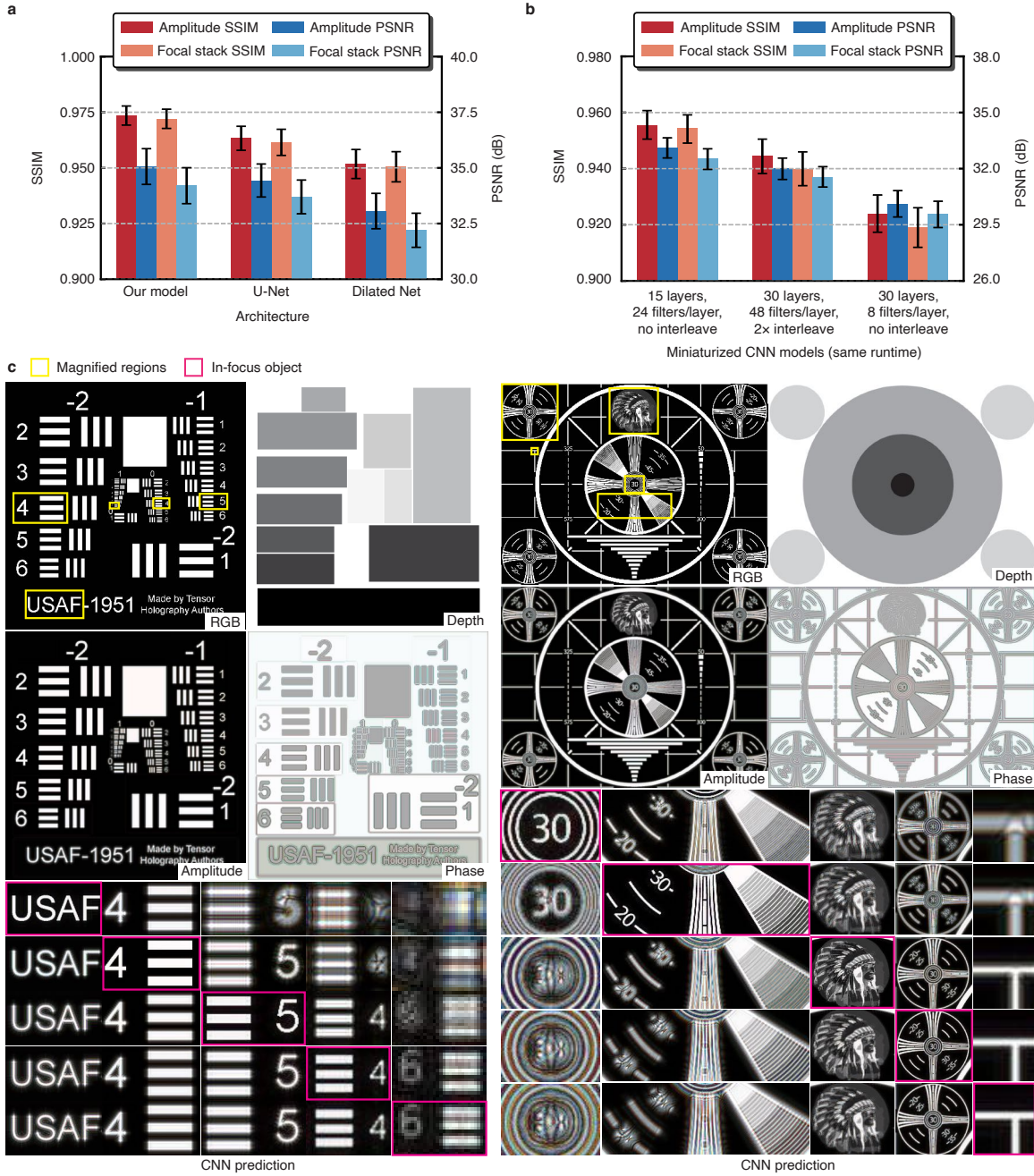


Figure 4-6: **Evaluation of tensor holography CNN on model architecture and test patterns.** **a**, Performance comparison of different CNN architectures. **b**, Performance comparison of different CNN miniaturization methods. **c**, CNN prediction of two standard test patterns (USAF-1951 and RCA Indian-head).



Figure 4-7: **Evaluation of tensor holography CNN on additional computer-rendered scenes.** CNN prediction of amplitude and phase along with focused reconstructions for holograms of **a**, a living room scene from the DeepFocus dataset [258] and **b**, a night landscape scene from the Stanford light field dataset [200].

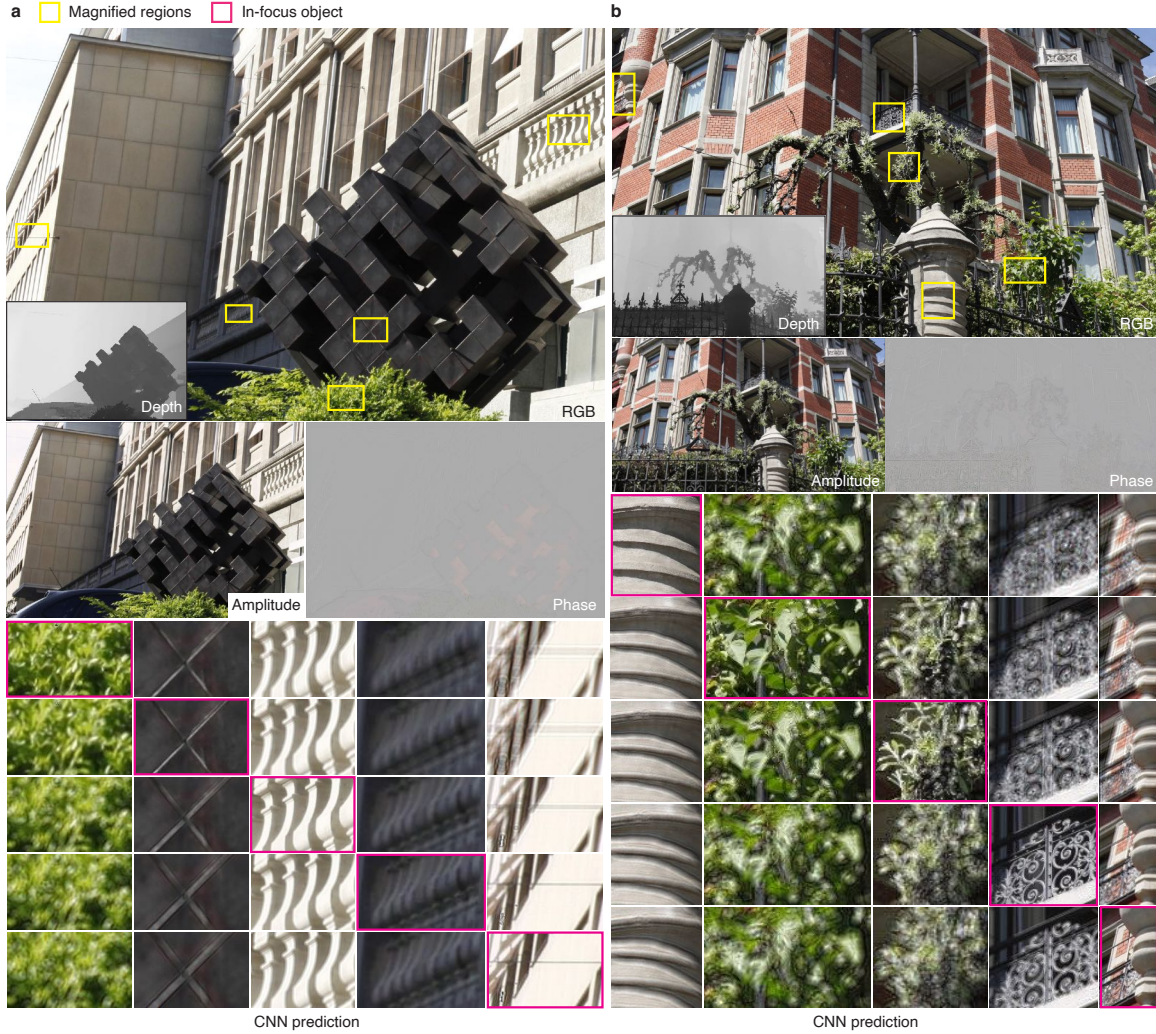


Figure 4-8: **Evaluation of tensor holography CNN on real-world captured scenes.** CNN prediction of amplitude and phase along with focused reconstructions for holograms of **a**, a statue scene and **b**, a mansion scene. Both scenes are from the ETH light field dataset [139].

Pro’s A13 Bionic chip, the mobile inference runs at a speed of 1.3Hz.

In Fig. 4-6c, we evaluate our CNN on two additional standard patterns (USAF-1951 and RCA Indian-head). The CNN predicted holograms are able to reproduce a few pixel wide textures inside the bottom-right magnified in-focus regions. In Fig. 4-7 and Fig. 4-8, we show 4 additional complex scenes (2 computer-rendered and 2 real-world captured) and the CNN predicted holograms.

4.1.5 Anti-Aliasing Double Phase Method (AA-DPM)

The proposed network predicts a complex-valued hologram, which requires a phase-only conversion to be able to physically displayed by a phase-only SLM. Double phase method encodes an amplitude-normalized complex hologram $Ae^{i\phi} \in \mathbb{C}^{M \times N}$ ($0 \leq A \leq 1$) into sum of two phase-only holograms at half of the normalized maximum amplitude:

$$Ae^{i\phi} = 0.5e^{i(\phi - \cos^{-1} A)} + 0.5e^{i(\phi + \cos^{-1} A)} \quad (4.11)$$

There are many different methods to merge decomposed two phase-only holograms into a single phase-only hologram. The original double phase method [187] uses a checkerboard mask to select interleaving phase values from the two phase-only holograms. Maimone et al. [175] 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 artifacts around high frequency objects and occlusion boundaries (see Fig. 4-9 (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 artifacts can be partially suppressed by closing the aperture and cutting high frequency signal in the Fourier domain, that leads to significant blurring. While sampling is inevitable, we borrow techniques employed in traditional image sub-sampling to holographic content and introduce an anti-aliasing double phase method. Specifically, we first convolve the complex

hologram by a Gaussian kernel $G_{W_G}(\sigma)$ to obtain a low-pass filtered complex hologram $\bar{A}e^{i\bar{\phi}} \in \mathbb{C}^{M \times N}$:

$$\bar{A}e^{i\bar{\phi}} = Ae^{i\phi} * G_{W_G}(\sigma) \quad (4.12)$$

where $*$ denotes 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 in between 0.5 and 1 are generally sufficient for both 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 artifact-free 3D image. We also add a global phase offset to $\bar{A}e^{i\bar{\phi}}$ to center the mean phase around half of the full phase shift range of SLM (3π in our case). This avoids phase wrapping and results in smooth phase distribution [175]. 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 Eq. (4.11), 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

$$P(m, n) = \begin{cases} P_1(m, n) & \text{if } m+n \text{ is odd} \\ P_2(m, n) & \text{if } m+n \text{ is even} \end{cases} \quad (-M/2 \leq m \leq M/2 - 1, -N/2 \leq n \leq N/2 - 1). \quad (4.13)$$

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. Fig. 4-9 compares the depth-of-field images simulated for AA-DPM and DPM, where AA-DPM produces artifacts-free images in regions with high spatial frequency details and around occlusion boundaries. AA-DPM can be efficiently implemented on GPU as two gather operations, which takes less than 1 ms to convert a 1920×1080 pixels complex hologram on a single NVIDIA TITAN RTX GPU.

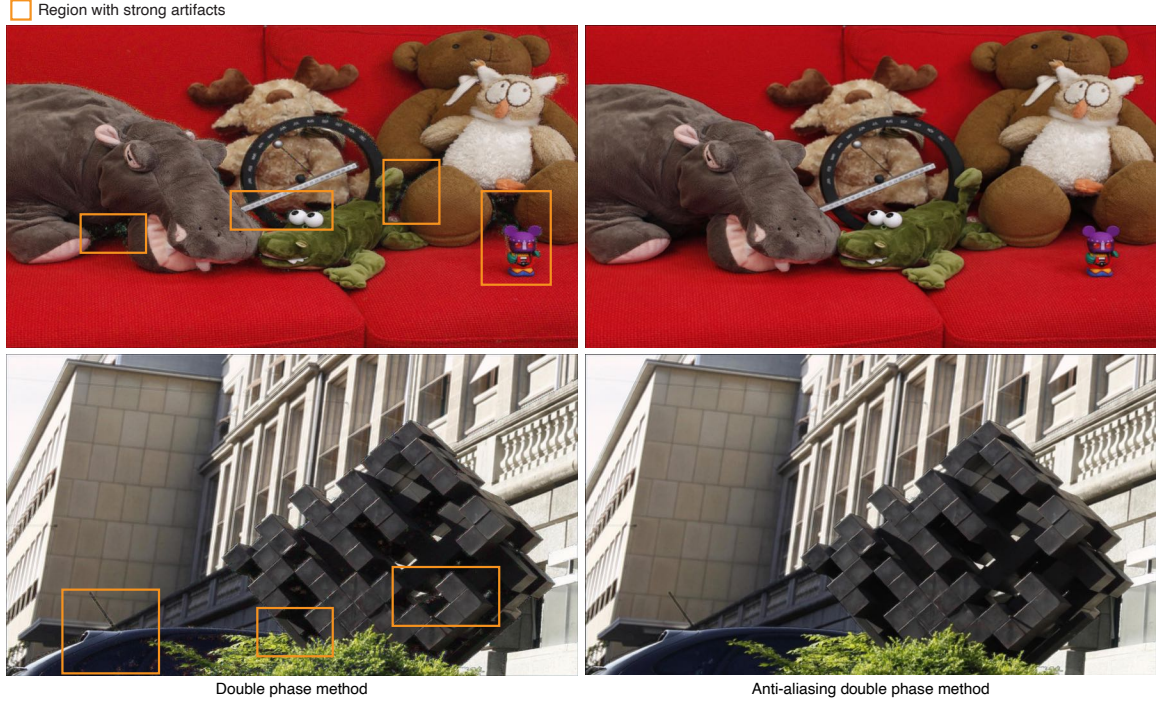


Figure 4-9: **Comparison of the original double phase method and anti-aliasing double phase method.** Reconstruction of two real-world scenes from the encoded phase-only holograms. The couch scene is focused on the mouse toy and the statue scene is focused on the black statue. Orange bounding boxes highlight regions with strong high-frequency artifacts. (left) Double phase method. (right) anti-aliasing double phase method.

4.1.6 Experimental Setup

We designed a holographic display prototype to experimentally validate the proposed algorithm. Our display prototype (see Fig. 4-10) uses a Fisba RGBeam fiber-coupled laser and a single HOLOEYE PLUTO-2-VIS-014 liquid crystal on silicon (LCOS) reflective phase-only SLM with a resolution of 1920×1080 pixels and $8 \mu\text{m}$ pitch. 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 Thorlabs 30 mm and 60 mm cage system and components. The fiber-coupled laser is mounted using an FC/PC 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 beamsplitter 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

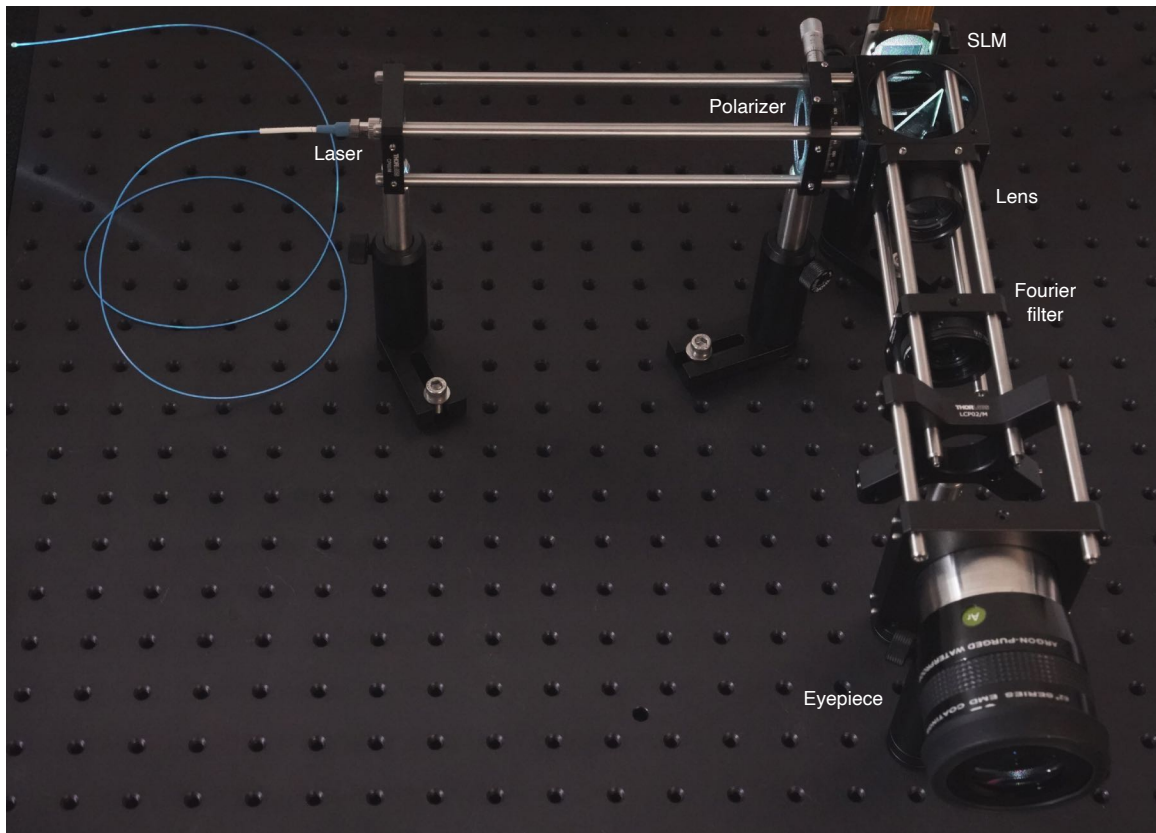


Figure 4-10: **Holographic display prototype used for the experimental results shown in this Letter.** The control box of the laser, Labjack DAQ, and camera are not visualized in the figure.

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. A Sony A7 Mark III mirrorless camera with a resolution of 6000×4000 pixels and a Sony 16-35 mm f/2.8 GM lens is paired to photograph and videograph the display. Color 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 color-matched phase-only holograms. Each hologram is quantized to 8-bit to match the bit depth of the SLM. For results shown in Fig. 4-11b, Fig. 4-12, and Fig. 4-13a, we used a Meade Series 5000 21 mm MWA eyepiece. For results shown in Fig. 4-11c/d and Fig. 4-13b, we used an Explore Scientific 32 mm eyepiece. The photograph was captured by exposing each color channel for 1 s. The long exposure time improves the signal-to-noise ratio and color accuracy. Video results showing real-time computation and continuous 3D refocus can be found at: <http://cgh.csail.mit.edu>. No post sharpening, denoising, or de-speckling 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.[175].

4.1.7 Experimental Results

We experimentally validate both 2D and 3D holograms. In Fig. 4-11b, 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. 4-11c/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 Fig. 4-12 and 4-13, where the image details closely match the simulation.

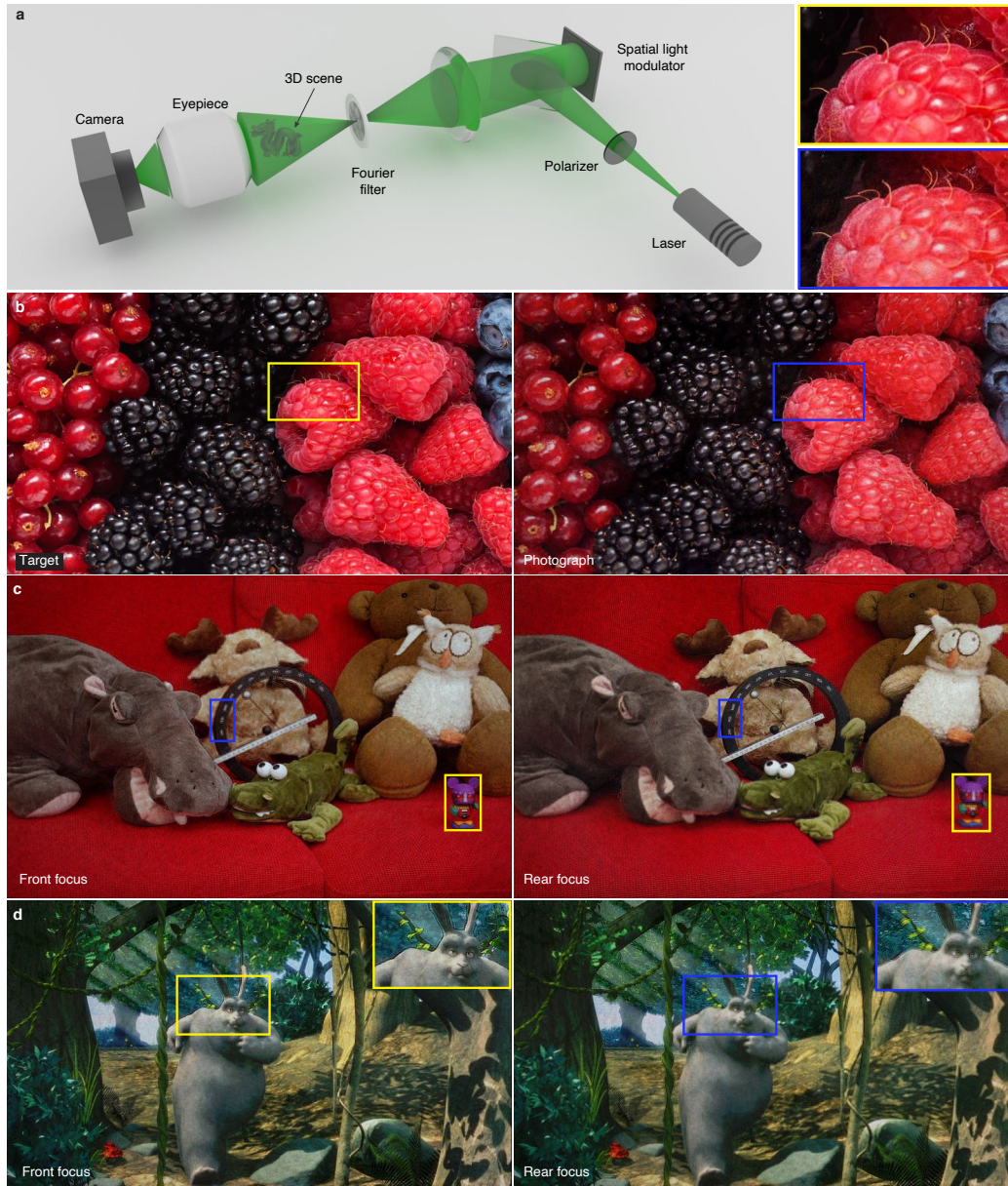


Figure 4-11: **Experimental demonstration of 2D and 3D holographic projection.** **a**, Scheme of our phase-only holographic display prototype. Only the green laser is visualized. **b**, (left) A flat (2D) target image (by Ana Blazic Pavlovic/Shutterstock) for testing the spatial resolution of our prototype. (right) A photograph of the CNN predicted hologram (encoded with anti-aliasing double phase method) displayed on our prototype. The insets on the top right show the magnified bounding boxes. **c**, Photographs of our prototype presenting a real-world captured 3D couch scene in Fig. 4-5c. The left photograph is focused on the mouse toy and the right photograph is focused on the perpetual desk calendar. **d**, Photographs of our prototype presenting a computer-rendered 3D Big Buck Bunny scene in Fig. 4-5a. The left photograph is focused on the bunny’s eye and the right photograph is focused on the background tree leaves.

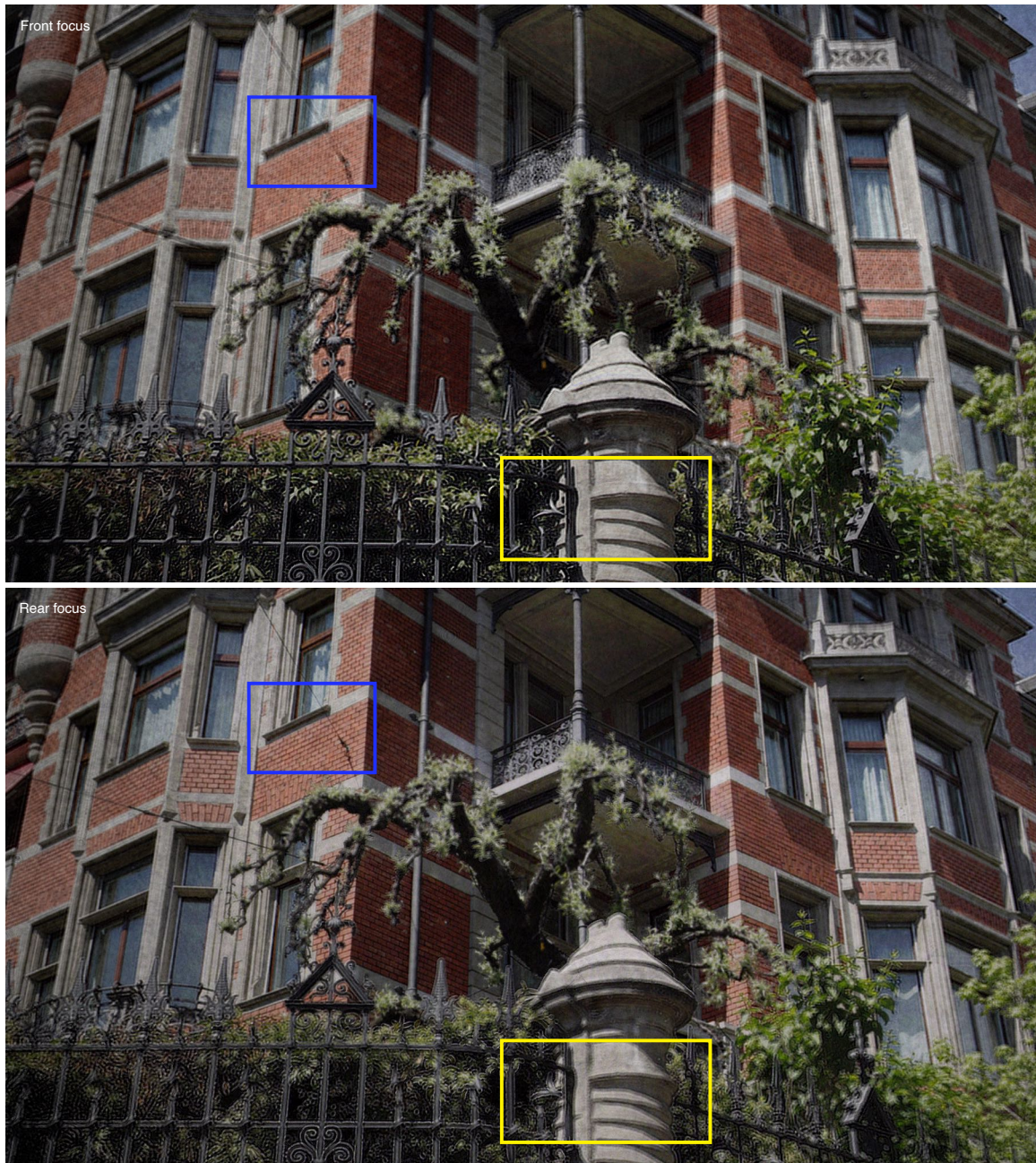


Figure 4-12: **Additional experimental demonstration of 3D holographic projection (part 1).** The RGB-D input can be found in Fig. [4-8](#).

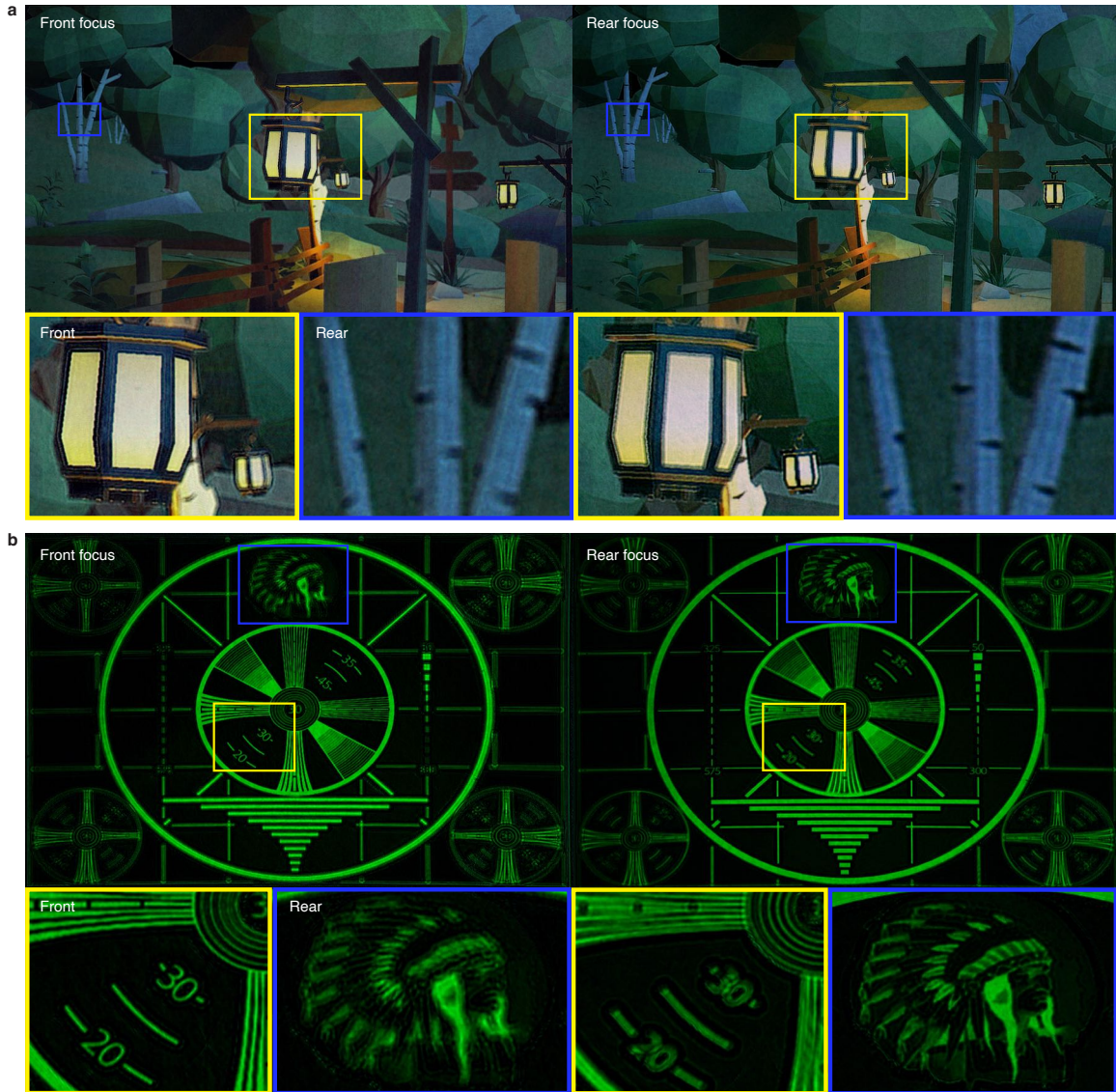


Figure 4-13: **Additional experimental demonstration of 3D holographic projection (part 2).** The RGB-D inputs can be found in Fig. 4-8 for a, and Fig. 4-6 for b.

4.1.8 Discussion

In this work, we present the first evidence of using convolutional neural networks for real-time photorealistic 3D CGH synthesis from a single RGB-D image, a task of which 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 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 testbed for both commercial and academic research fields that will benefit from real-time, high-resolution CGH, e.g., consumer holographic displays for AR/VR, hologram-based single-shot volumetric 3D printing, optical trapping with significantly 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 cheap to compute and memory-efficient, it only provides accurate 3D depiction from 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. Finally, the rapid development of ASICs will hopefully make high framerate tensor holography viable on mobile devices, enabling untethered real 3D viewing experiences and significantly lowering the cost and barrier to entry for holographic content creation.

4.2 Tensor Holography V2: End-to-end Learning of 3D Phase-only Holograms

4.2.1 Motivation

The method proposed in the earlier section suffers from a few key limitations. First, the RGB-D scene representation has intrinsic limitations in depicting the complete scene geometry as it records the frontmost surface points. The lack of wavefront diffracted from the occluded points causes visible dark seams or brightness attenuation at the background side of the occlusion boundaries (see Fig. 4-2d bottom row). Second, the AA-DPM algorithm for phase-only encoding requires manual tuning to find the optimal parameters. For video or live streaming inputs where manual tuning is infeasible, highly conservative filtering is required to keep the entire sequence artifacts free. Meanwhile, the spatially-invariant low-pass filtering does not adapt to the content and may incur unnecessary blur. In addition, the minimal filtering strength grows with the sub-hologram width, leading to degraded performance when the 3D frustum is placed far against the hologram plane. In practice, this limits the image quality when designing compact holographic near-eye displays, where it is ideal for reducing the physical distance between the SLM and the eyepiece by computationally shifting the 3D image tens of millimeters behind the hologram. Third, the generated hologram does not account for the different users' vision aberrations such that vision correction glasses are still required for them to see the image clearly.

In this chapter, we present a series of techniques aimed at overcoming the aforementioned limitations. Our methods encompass several innovations, including the pioneering use of a layered depth image [220] as a data-efficient 3D representation for hologram synthesis. Additionally, we introduce a novel hologram dataset computed using the silhouette-mask layer-based method [270] with ultra-dense layer partitioning (10000 layers), effectively eliminating remaining occlusion artifacts. To enhance performance further, we propose a two-stage supervised+unsupervised training protocol, leveraging the advantages of both simulation-based and optimization-based approaches. Moreover, we devise a method to integrate aberration correction into the 3D hologram learning pipeline. The resulting system, named Tensor Holography V2, is capable of synthesizing photorealistic 3D phase-only holograms end-to-end without the need for manual parameter adjustment. It exhibits robustness to variations in image statistics, depth misalignment in real-world captured inputs,

and variations in distance configurations between the hologram plane and the 3D volume. Furthermore, we introduce a calibration process to tailor holograms for end-users with different vision aberrations, demonstrating the neural surrogate’s ability to adapt generation to the statistics of customized holograms. Through experimental validation, we showcase the high image quality of holographic 3D projection and the effectiveness of aberration correction. Lastly, we discuss the innovative SLM calibration procedures utilized to achieve these demonstrated results.

4.2.2 Layered Depth Images and Silhouette-Mask Layer-Based Method for Hologram Generation.

Occlusion modeling for 3D holograms is critical to perceived realism. This task has been approached at the scene representation stage by rasterizing or ray-tracing an RGB-D image (with a depth map) to only record the frontmost visible surface [106, 175]; rendering light fields further accounts for view-dependent effects [200, 222]. It has also been approached at the hologram computation stage via ray casting for point-based methods [125] and silhouettes for FFT-based methods [182, 183]. The combination of both approaches is used to prune the computation of non-line-of-sight objects and remove the wavefront of visible background objects occluded by the foreground [224].

Figure 4-14 examines the effectiveness of each approach using a simple test scene consisting of two vertical bars with the front bar perfectly occluding the lower half of the rear bar (see Fig. 4-14 Top). We assign different amplitudes to the front and rear bars and repeat the bar pair horizontally for visualization. The complete scene is represented by an LDI with two layers: the first layer for the front bar plus the top half of the rear bar and the second layer for the bottom half of the rear bar. The rasterized scene is represented by the LDI’s first layer. We tested 6 pairs of input and CGH algorithms:

1. No occlusion: apply the point-based method to the complete scene.
2. Occlusion only from input: apply the point-based method to the rasterized scene.
3. Geometric occlusion detection (input+computation): apply the occlusion-aware point-based method to the rasterized scene.

4. Geometric occlusion detection (computation): apply the occlusion-aware point-based method to the complete scene.
5. Wave-based occlusion detection (input+computation): apply the silhouette-mask layer-based method to the rasterized scene.
6. Wave-based occlusion detection (computation): apply the silhouette-mask layer-based method to the complete scene.

Fig. 4-14 visualizes the result and we conclude 4 types of artifacts:

1. Leakage of the wavefront from occluded points.
2. Leakage of the wavefront from visible points.
3. Loss of wavefront from occluded points.
4. Inaccurate wavefront due to geometric optics modeling.

Modeling occlusion at the scene representation stage (setting 2) avoids type 1 artifacts and reduces the input data size. Applying either type of occlusion detection at the computation stage (setting 3 or 5) removes type 2 artifacts. If occlusion detection is wave-based (setting 5), type 4 artifacts are also removed, resulting in an artifact-free foreground reproduction. However, type 3 artifacts persist (for setting 2,3,5) due to the loss of occluded points and their wavefronts, causing loss of amplitude or attenuation at the background side of the occlusion boundaries, which are the dominating artifacts in TensorHolo V1. Retaining the complete scene and applying wave-based occlusion detection (setting 6) avoid all types of artifacts with the defocus response of the background bars matching the ones in setting 1. However, this mode incurs a higher data and computational cost.

Common representations that fully or largely serve this need include: 1) general point clouds; 2) meshes; 3) voxel grids; 4) light fields. Among them, general point clouds and meshes are CNN-unfriendly due to their scene-dependent feature lengths. Voxel grids operate in an explicit tradeoff: as the depth resolution increases, the memory cost also increases linearly. Thus, it does not scale with high-resolution input, and the CNN's first convolution layer bottlenecks the performance due to a vast number of input channels. For light fields,

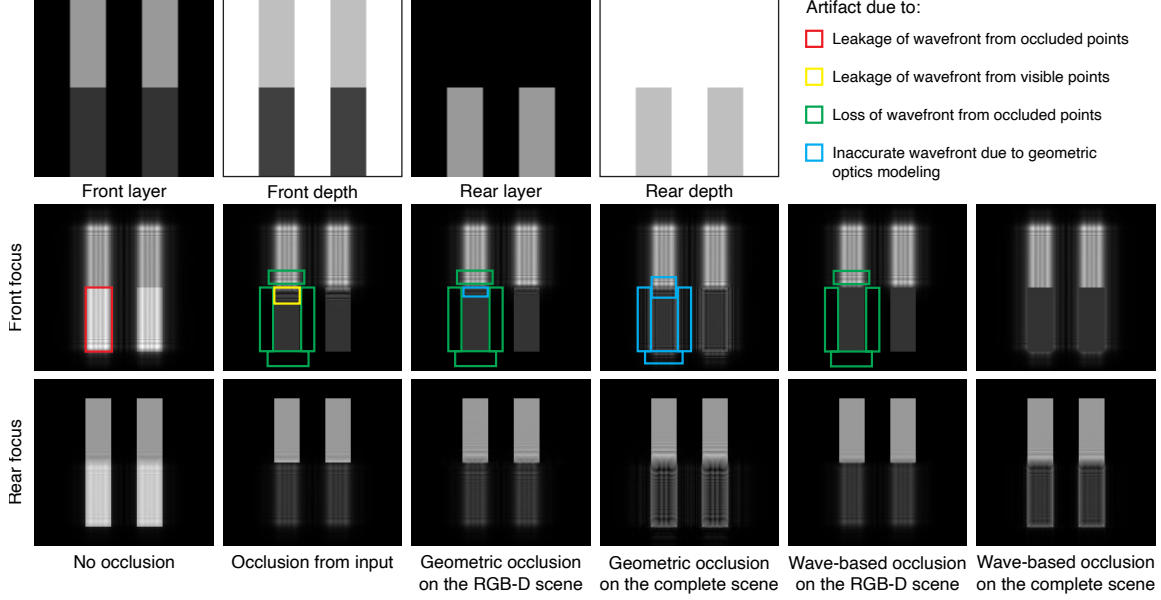


Figure 4-14: **Methods comparison for occlusion modeling.** The top 4 images constitute a 2-layer LDI that completely depicts the test scene. Different types of artifacts are only marked for the front focus images.

if pixels are treated as points [222, 273], it is inefficient as one point can be recorded multiple times in different views, and avoiding double-counting requires duplication detection; alternatively, if pixels are treated as rays [200], generating high-quality sub-holograms will require an extremely dense angular sampling rate.

Here, we advocate a layered depth image (LDI) as an efficient 3D scene representation. LDI is a classical image-based rendering technique originally developed for novel view synthesis [220]. Unlike an RGB-D image that stores a single pixel at each spatial coordinate, it stores a sequence of RGB-D images along the line of sight originated from each spatial coordinate. Its first pixel records the first front-facing surface intersecting with the line of sight, and the second pixel records the second intersection assuming the line of sight pierces through the scene, and so forth till a maximum hit (layer) count is met (see Fig. 4-15a for an illustration). LDI has several unique advantages for holographic rendering applications. First, it is highly configurable such that if only a single layer is rendered, an LDI is equivalent to an RGB-D image; if all layers are exhaustively rendered, it losslessly encodes the entire scene for computing a physically correct hologram. In either case or for a limited number of layers, any point in the scene is only recorded once or discarded. Sec-

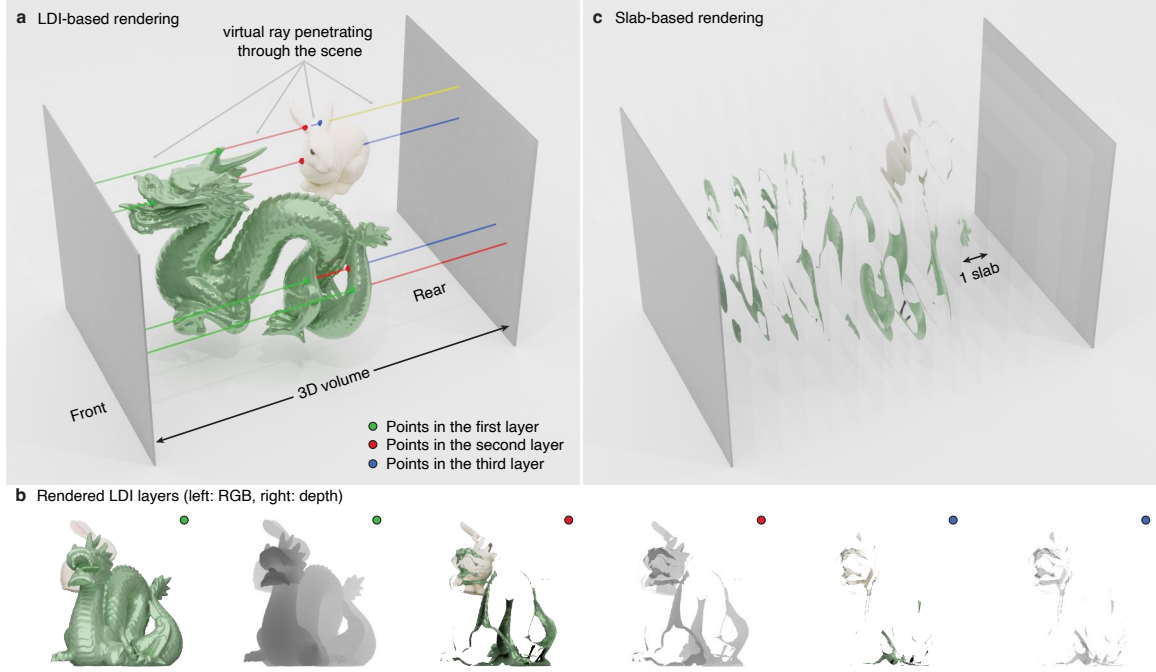


Figure 4-15: **Comparison of LDI and voxel grid as 3D scene representation.** **a** Rendering of an LDI records the intersections between a penetrating ray (colored differently after every new hit) and the scene objects (front-facing surfaces) at every spatial location. **b** First three layers of the LDI rendered for the scene in (a). The number of valid pixels rapidly decreases as the index of the layer increases. **c** Rendering of a voxel grid that partitions the 3D scene into equal-length slabs and generate one RGB image for each slab. The depth resolution is proportional to the number of slabs.

ond, unlike a voxel grid, an LDI records the exact depth for every hit, decoupling the depth resolution with the number of LDI layers. Third, the sequence of pixels is recorded in a depth-sorted manner with wavefront from further layers providing a diminishing contribution to the hologram due to the occlusion of the frontal layers. Consequently, we find that a few LDI layers (i.e., 5) are sufficient to produce a close-to-ground-truth hologram, and thus it is highly data-efficient. Fourth, once the number of layers is set, its feature-length is fixed and independent of scene complexity; thus, it is CNN-friendly. Finally, it can be efficiently rendered by the existing real-time graphics pipeline via depth peeling (see Methods for details).

To compute a 3D hologram from an LDI, ray casting can be performed from each point’s belonging mesh at the recorded 3D location, yet using geometric occlusion for diffraction calculation has been shown to cause minor artifacts. Because runtime is ul-

timately unimportant for dataset synthesis, we propose using the silhouette-mask layer-based method [270] (SM-LBM) with ultra-dense depth partition to avoid the mixed-use of geometric and wave optics models. SM-LBM was originally proposed to receive a voxel grid input generated by slab-based rendering (see Fig. 4-15b), which does not scale with increasing depth resolution. Using SM-LBM with LDI is straightforward. Any non-zero pixel in an LDI defines a valid 3D point before depth quantization. When the number of depth layers N is determined, each point is projected to its nearest plane, and a silhouette is set at the same spatial location. Denote the complex amplitude distribution of the N -th layer $L_N \in \mathbb{C}^{R_x \times R_y}$ as

$$L_N = A_N \exp\left(i \frac{2\pi z_N}{\lambda}\right). \quad (4.14)$$

Here, z_N is the signed distance from the N -th layer to the hologram plane, where a negative distance denotes a layer behind the hologram plane and vice versa, $A_N \in \mathbb{R}^{R_x \times R_y}$ is the amplitude of the layer, R_x , and R_y are the spatial resolution along the x and y axis. The exponential term defines the layer's initial phase following Maimone et al.'s formula to induce a smooth and roughly zero-mean phase profile at the hologram plane [175]. We use the angular spectrum method to propagate L_N to the location of $N-1$ -th layer

$$\begin{aligned} C_{N-1} &= \text{ASM}(L_N, d_l) \\ &= F^{-1} \left(F(L_N) \odot \exp \left(i 2\pi d_l \sqrt{\lambda^{-2} - \mu^2 - \nu^2} \right) \right), \end{aligned} \quad (4.15)$$

where $\mu \in \mathbb{R}^{R_x \times R_y}$ and $\nu \in \mathbb{R}^{R_x \times R_y}$ are the spatial frequencies along the x and y directions, d_l is the layer thickness (positive), $\odot$ denotes Hadamard element-wise product, F and F^{-1} are the 2D Fourier transform and inverse Fourier transform operator. C_{N-1} is multiplied by the binary silhouette mask at the $N-1$ layer

$$M_{N-1}(x, y) = \begin{cases} 0, & A_{N-1}(x, y) > 0 \\ 1, & A_{N-1}(x, y) = 0 \end{cases}, \quad (4.16)$$

and the complex amplitude at the $N-1$ layer is updated by adding the masked complex field

$$L_{N-1} = C_{N-1}M_{N-1} + L_{N-1}, \quad (4.17)$$

By iterating this process until reaching the first layer, the final complex hologram is obtained by propagating the updated first layer to the hologram plane.

We further augment SM-LBM with aberration correction at the cost of computational efficiency. Reconsidering the forward propagation of N -th layer L_N , we only process the occlusion of frontal layer without adding their content, namely removing the second addition term in Eq. (4.17). After processing the occlusion of all frontal layers, we propagate the resulted wavefront back to the starting location of L_N to obtain an occlusion-processed L'_N . We then perform aberration correction in the frequency domain

$$L'_{N_C} = F^{-1} \left(F(L'_N) \odot \Phi_{z_N} \right), \quad (4.18)$$

where $\Phi_{z_N} \in \mathbb{C}^{R_x \times R_y}$ is a depth-dependent global aberration correction kernel in the Fourier domain. We detail the process to obtain Φ_{z_N} and an extension to model depth-dependent spatially-varying aberration correction in Methods. Finally, L'_{N_C} is propagated to the target hologram plane. This procedure is repeated independently for the content at every discretized depth (i.e., from layer 1 to N) and integrating the results of all procedures at the target plane forms the final hologram. Note that the required total number of propagation operations increases to $N^2/2$ compared with N in the standard SM-LBM. This prevents N from being extremely high if aberration correction is needed, but the computational resource is limited.

4.2.3 MIT-CGH-4K-V2

SM-LBM and its aberration-correction variant are slow due to sequential occlusion processing. To improve the performance, we generate a new dataset with LDIs and SM-LBM holograms and train a CNN to accelerate inference. Generating this dataset requires setting three critical parameters: the depth of the 3D volume, the number of layers used by LDIs,

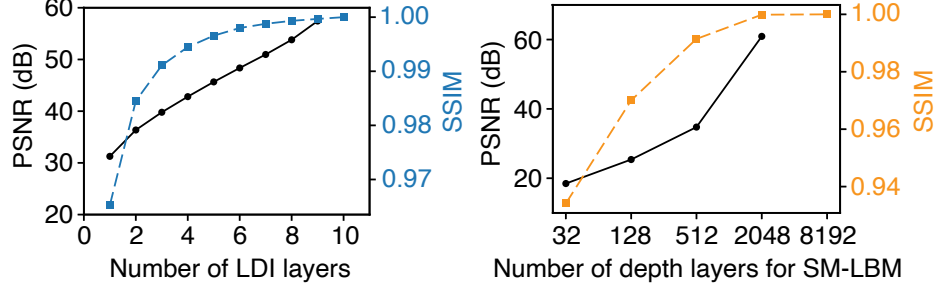


Figure 4-16: **SM-LBM hologram quality as a function of LDI layers and depth discretization.** **a** The addition of more LDI layers provides diminishing returns on the hologram fidelity and the improvement plateaus after 5 layers. **b** A depth resolution over 11 bits (2048) results in negligible difference to the image quality.

and the number of layers (depth resolution) used by SM-LBM.

We set the 3D volume depth to be 6 mm under collimated illumination to facilitate quantitative comparison with the publicly available TensorHolo V1 network [224], and similarly for the random scene configuration. To determine the number of layers for LDIs, we compute the mean peak signal to noise ratio (PSNR) and the mean structure similarity index [253] (SSIM) for the amplitude maps of the holograms computed from LDIs with $N = 1, 2, \dots, 9$ layers against the ones computed from LDIs with $N = 10$ layers (after which we observe few valid pixels) over 10 random scenes. The mean SSIM plateaus after $N = 5$ (see Fig. 4-16a), reflecting a diminishing improvement with more layers. Thus, we choose $N = 5$ for this work, but more layers can be used for higher accuracy. Similarly, to determine the number of layers for SM-LBM, we compute the holograms using 2^{N_d} layers for $N_d = 5, 7, 9, 11$ and compare the mean PSNR and the mean SSIM of these holograms against the ones computed from $N_d = 13$ over 10 random scenes. The mean SSIM plateaus after $N_d = 11$ (see Figure 4-16b), indicating negligible improvement for the 3D image quality. Nevertheless, we use a partition of 10000 layers (13.28-bit depth) as a showcase, which has not been demonstrated previously. We rendered MIT-CGH-4K-V2, a new hologram dataset with 4000 pairs of LDIs and holograms with 3800 for training, 100 for validation, and 100 for testing at 384×384 pixels similar to TensorHolo V1.

4.2.4 End-to-end Learning of 3D Phase-only Hologram

In TensorHolo V1, the CNN directly maps the input RGB-D image to the midpoint hologram, a wavefront recording plane set to the center of the 3D volume that minimizes the sub-hologram width, and phase-only encoding is performed separately using anti-aliasing double phase encoding (AA-DPM). However, both AA-DPM and an analogous band-limiting double phase method (BL-DPM) [240] require manual tuning to find the optimal low-pass filtering parameters that produce an artifacts-free image with the minimal artificial blur. For video or live streaming inputs where manual tuning is infeasible, highly conservative filtering is required to keep the entire sequence artifacts-free. Meanwhile, the low-pass filtering in both methods are either spatially-invariant or frequency-invariant; thus they do not adapt to the content and may incur unnecessary filtering. In addition, we observe the minimal filtering strength of both methods grows with the sub-hologram width, leading to degraded performance when the 3D frustum is placed far against the hologram plane. In practice, this limits the image quality when designing compact holographic near-eye displays, where it is ideal for reducing the physical distance between the SLM and the eyepiece by computationally shifting the 3D image tens of millimeters behind the hologram.

Although a CNN can be trained to directly predict an unconstrained 3D phase-only hologram using unsupervised learning by only forcing the focal stack to match the one produced by the target complex hologram [73], ablation studies have shown that removing the supervision of ground truth complex hologram noticeably degrades the image quality [224] and enforcing the phase-only constraint can only worsen the performance. Moreover, direct synthesis of phase-only holograms prevents using midpoint holograms to reduce computational costs since learning an unconstrained midpoint phase-only hologram does not guarantee a uniform amplitude at the target hologram plane.

We propose a two-stage supervised+unsupervised training to overcome these challenges (see Fig. 4-17 for visualization). The key insight is to keep using the double phase principle to perform phase-only encoding for retaining the advantage of learning the midpoint hologram, while embedding the encoding process into the end-to-end training

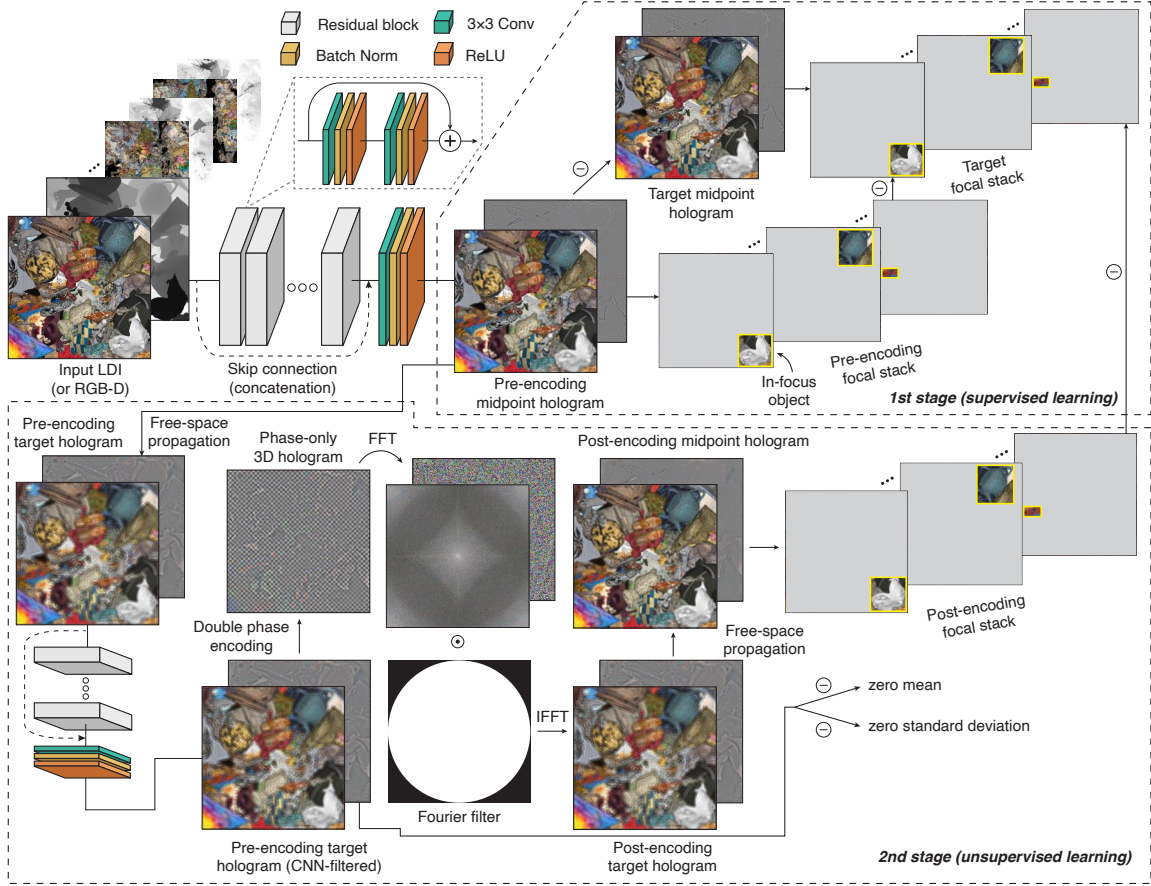


Figure 4-17: **Overview of training procedure.** In the first stage, the CNN is trained to reproduce the ground truth midpoint hologram with direct supervision and a dynamic focal stack loss. In the second stage, the CNN prediction is propagated to the target hologram plane, filtered by a second CNN, double phase encoded, Fourier filtered, and back propagated to the center of the 3D volume to obtain the post-encoding midpoint hologram. No ground truth phase-only hologram is provided, and the CNN is trained to discover an optimal solution with the dynamic focal stack loss between the post-encoding focal stack and the target focal stack and a regularization loss (see Equation (4.24)). The Fourier space amplitude is visualized in the logarithmic scale.

pipeline and relegating the CNNs to discover the optimal pre-encoding complex hologram through unsupervised training. We detail the training process below and refer to this neural phase-only conversion method as the deep double phase method (DDPM).

The first stage supervised training is identical to TensorHolo V1 despite that we train two versions of CNNs in V2. Both are trained to predict the target midpoint hologram computed from the LDI input, but one receives the full LDI, and the other receives only the first layer of the LDI, the same RGB-D input as the V1 network receives. The latter CNN has an additional job of inferring the occluded points close to the depth boundaries and filling in their missing wavefront. It is particularly useful for reducing the rendering overhead and reproducing real-world scenes captured as RGB-D images, where physically capturing a pixel-aligned LDI is nearly impossible.

Once the CNN excels at this task, we initialize the second stage unsupervised training by applying a chain of operations to the network-predicted midpoint hologram $\tilde{H}_{\text{mid}} = \tilde{A}_{\text{mid}} e^{i\tilde{\phi}_{\text{mid}}}$. First, it is propagated to the target hologram plane and pre-processed by a second CNN to yield the pre-encoding target hologram prediction

$$\begin{aligned}\tilde{H}_{\text{tgt-pre}} &= \tilde{a}_{\text{tgt-pre}} \tilde{A}_{\text{tgt-pre}} e^{i\tilde{\phi}_{\text{tgt-pre}}} \\ &= \text{CNN}_{\text{filter}} \left(\text{ASM}(\tilde{H}_{\text{mid}}, d_{\text{offset}}) \exp \left(i \frac{2\pi d_{\text{offset}}}{\lambda} \right) \right)\end{aligned}\quad (4.19)$$

where d_{offset} is the signed distance from the midpoint hologram to the target hologram plane, $\tilde{A}_{\text{tgt-pre}}$ is the normalized amplitude, and $\tilde{a}_{\text{tgt-pre}}$ is the scale multiplier. The second CNN serves as a content-adaptive filter to replace the Gaussian blur in AA-DPM. The exponential phase correction term ensures that the phase after propagation is still roughly centered at 0 for all color channels. It is also critical to the success of AA-DPM, which minimizes phase wrapping. Next, the standard double phase encoding is applied to obtain a phase-only hologram

$$P(x, y) = \begin{cases} 0.5 \tilde{a}_{\text{tgt-pre}} e^{i(\tilde{\phi}_{\text{tgt-pre}}(x, y) - \cos^{-1} \tilde{A}_{\text{tgt-pre}}(x, y))}, & x + y \text{ is odd} \\ 0.5 \tilde{a}_{\text{tgt-pre}} e^{i(\tilde{\phi}_{\text{tgt-pre}}(x, y) + \cos^{-1} \tilde{A}_{\text{tgt-pre}}(x, y))}, & x + y \text{ is even} \end{cases}, \quad (4.20)$$

and no pre-blurring is applied in contrast to AA-DPM. Third, the phase-only hologram is filtered in the Fourier space to obtain the post-encoding target hologram prediction

$$\tilde{H}_{\text{tgt-post}} = F^{-1}(F(P) \odot M_{\text{Fourier}}), \quad (4.21)$$

where M_{Fourier} models a circular aperture in the Fourier plane

$$M_{\text{Fourier}}(x, y) = \begin{cases} 1, & x^2 + y^2 \leq r^2 \\ 0, & x^2 + y^2 > r^2 \end{cases}, \quad (4.22)$$

Here, r is the radius of the aperture in the pixel space. We set it to half of the image resolution, letting the entire first-order diffraction pass through the physical aperture. Finally, the post-encoding target hologram prediction is propagated back to yield the post-encoding midpoint hologram

$$\tilde{H}_{\text{mid-post}} = \text{ASM}(\tilde{H}_{\text{tgt-post}}, -d_{\text{offset}}). \quad (4.23)$$

By appending these operations, the second stage unsupervised training fine-tunes the CNN prediction using the dynamic focal stack loss calculated between the post-encoding midpoint hologram and the ground truth midpoint hologram, plus a regularization loss on the pre-encoding target hologram phase

$$l_{\text{tgt-pre}} = \sqrt{\frac{\sum (\tilde{\phi}_{\text{tgt-pre}} - \overline{\tilde{\phi}_{\text{tgt-pre}}})^2}{R_x R_y}} + \overline{\tilde{\phi}_{\text{tgt-pre}}}, \quad (4.24)$$

where $\bar{\cdot}$ denotes the mean operation. The regularization loss encourages the pre-encoding target hologram phase to be zero mean and to exhibit a small standard deviation. This term minimizes phase wrapping during the double phase encoding, which may not affect the simulated image quality but degrade the experimental result. Without this loss, the unregulated phase exhibits a large standard deviation and shifts away from zero mean, leading to non-negligible phase wrapping, especially when the maximum phase modulation is limited to 2π (see Fig. 4-21c).

In the second training stage, direct supervision from the ground truth midpoint hologram is intentionally ablated. This expands the solution space by allowing the CNNs to freely explore the neural filtering to optimally match the ground truth focal stack, which a user ultimately sees. It also facilitates regularization on the pre-encoding target hologram

phase to better handle hardware limitations (i.e., limited range of phase modulation). In practice, the resulting prediction of the post-midpoint hologram phase visually differs from the ground truth as high-frequency details are attenuated or altered in a spatially-varying and content-adaptive manner to avoid speckle noise. With direct supervision that encourages retention of high-frequency details, we find it negatively impacts speckle elimination.

Collectively, the proposed two-stage training first excels at reproducing the ground truth complex 3D holograms at all levels of detail, then fine-tunes a display-specific CNN for fully automatic speckle-free 3D phase-only hologram synthesis. The second training stage takes fewer iterations to converge; therefore, upon the completion of the first training stage, it is efficient to optimize multiple CNNs for different display configurations. The training process is detailed in Methods.

4.2.5 Compensating Hardware Imperfection

Hardware imperfection can cause experimental results to deviate from the idealized simulations [204, 39]. Here we discuss methods to compensate for three sources of error: laser source intensity variation as a Gaussian beam, SLM’s non-linear voltage-to-phase response, and optical aberrations.

To calibrate the laser source intensity variation, we substitute the SLM with a diffuser and capture the reflected beam as a scaling map for adjusting the target amplitude. A 5×5 median filter is applied to the measurements to avoid pepper noise caused by dust on the optical elements. A Gaussian mixture model can be used to fit an analytical model of the resulting scaling map if needed [204].

For an imprecisely calibrated SLM, the non-linear voltage-to-phase response can severely reduce display contrast, especially for double-phase encoded hologram, since achieving deep black requires offsetting the checkerboard grating accurately by 1π . In many cases, the pixel response is also spatially non-uniform; thus, using a global look-up table is often inadequate (see Fig. 4-18). Existing calibration methods operate on the change of interference fringe offset (interferometry-based) or the change of near/far-field diffraction pattern

(diffraction-based), but they cannot produce a per-pixel look-up table (LUT) [160]. We propose a simple calibration procedure that uses double phase encoding to accomplish this goal. Specifically, for every 2-by-2 pixels, we keep the top right and bottom left pixels at 0 as a reference and increase the top left and bottom right pixels jointly from 0 to 255. Without modifying the display layout, we set the camera focus on the SLM and captured the intensity change for the entire frame. If the phase modulation range for the operating wavelength is greater equal than 2π , the intensity of the captured image will decrease to the minimum at 1π offset, return to the maximum at 2π offset, and repeat this pattern for every 2π cycle. Denote the k -th captured image I_k , the absolute angular difference in the polar coordinate between a reference pixel and an active pixel set to k is

$$\theta_k(x, y) = 2 \cos^{-1} \left(\frac{\sqrt{I_k}(x, y) - \sqrt{I_{\min}}(x, y)}{\sqrt{I_{\max}}(x, y) - \sqrt{I_{\min}}(x, y)} \right), \quad (4.25)$$

where $I_{\min}(x, y)$ and $I_{\max}(x, y)$ are the minimal and maximal intensities measured at location (x, y) when sweeping from 0 to 255. Let $k_{\min}(x, y)$ be the frame id associated with the minimal measurement at (x, y) , the phase difference is given by

$$\phi_k(x, y) = \begin{cases} \theta_k(x, y), & k \leq k_{\min}(x, y) \\ 2\pi - \theta_k(x, y), & k > k_{\min}(x, y) \end{cases}. \quad (4.26)$$

Experimentally, we take high-resolution measurements (24 megapixels) of the SLM response, downsample to the SLM resolution, perform the aforementioned calculations, and fit a linear generalized additive model (GAM) with monotonic increasing

constraint to obtain a smoothed phase curve for producing a per-pixel LUT. For simplicity, the LUT is directly loaded into the GPU memory for fast inference. To reduce memory con-

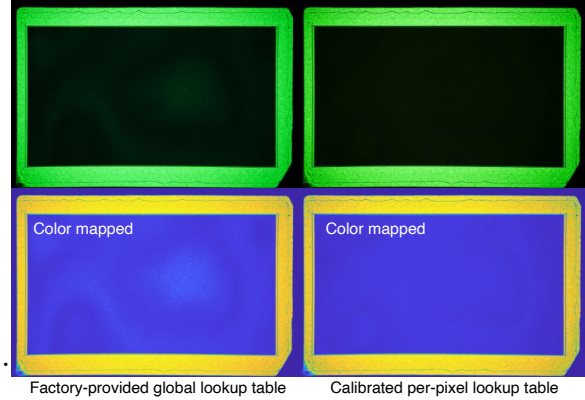


Figure 4-18: **Black modulation test using the double phase encoding.** Using the per-pixel lookup table calibrated via the proposed approach produces a more uniform image with a deeper black level over the one achieved by the factory-provided global lookup table.

sumption, a multi-layer perceptron can be learned and applied as a 1×1 convolution [204]. This in-situ calibration procedure eliminates potential model mismatch between a separate calibration setup and the display setup. The ability to accurately address phase differences results in more accurate color reproduction, i.e., producing deep black by accurately addressing 1π phase offset (see Fig. 4-18)

The optical aberrations are corrected using a variant of Maimone et al. [175]. Let $\phi'_d \in \mathbb{C}^{R_x \times R_y}$ (zero-padded to the frame resolution) be an ideal sub-hologram that focus plane wave to a signed distance d , we similarly use 5 Zernike polynomials:

$$Z_3(\rho, \theta) = a_{3_d}(2\rho^2 - 1) \quad \text{focus} \quad (4.27)$$

$$Z_4(\rho, \theta) = a_{4_d}(\rho^2 \cos 2\theta) \quad \text{vertical astigmatism} \quad (4.28)$$

$$Z_5(\rho, \theta) = a_{5_d}(\rho^2 \sin 2\theta) \quad \text{oblique astigmatism} \quad (4.29)$$

$$Z_6(\rho, \theta) = a_{6_d}((3\rho^2 - 2)\rho \cos \theta) \quad \text{horizontal coma} \quad (4.30)$$

$$Z_7(\rho, \theta) = a_{7_d}((3\rho^2 - 2)\rho \sin \theta) \quad \text{vertical coma} \quad (4.31)$$

$$\phi_d(\rho, \theta) = \phi'_d(\rho, \theta) e^{i \sum_j (Z_j(\rho, \theta))} \quad \text{corrected sub-hologram} \quad (4.32)$$

to model system aberrations, where a_{j_d} are Zernike coefficients, ρ is the normalized polar radius, and θ is the azimuthal angle. We perform a user calibration to adjust coefficients a_{j_d} until the camera images a tightly focused spot at d from the corrected sub-hologram ϕ_d . Once the calibration completes, we propagate ϕ_d to its focal plane to obtain the point spread function and compute the corrected amplitude transfer function as $\Phi_d = \text{ATF}_d = F(\text{PSF}_d) = F(\text{ASM}(\phi_d, d))$, which we use in Eq. (4.18) to perform frequency-domain aberration correction for the occlusion-processed layer. Note that this calibration procedure can be performed for different focal distances, and parameters can be piecewise linearly interpolated [175].

For compact eyeglass-like setups (i.e., Maimone et al.'s compact AR prototype), the same procedure can be followed by determining the unmagnified image location and the optical power of the diffractive lens while calibrating the kernel parameters to correct the system and eye aberrations jointly. Using a diffractive lens can yet cause strong aberrations that require spatially-varying aberration correction. In this case, we can calibrate the

display at multiple points (i.e., 15 points) and update the above procedure by convolving a spatially varying $\text{PSF}_d(x,y)$ calculated by interpolating the nearest measured parameters. Note that this operation can only be performed in the spatial domain but not in the Fourier domain. However, GPUs can accelerate this process, and speed is ultimately not critical for dataset generation. On the learning side, the CNN needs to receive an additional two-channel image that records the normalized xy coordinates to learn aberration correction in a spatially-varying manner. While this advanced task is exciting with a clear solution path, we defer it to future work.

4.2.6 Software Implementation

Algorithm 2 Depth peeling for LDI rendering

N : the number of layers in the output LDI
 X, Y : spatial resolution of the output LDI
 C : an RGB image buffer
 D_0, D_1 : two depth buffers
 F_{xy} : the set of all fragments defined at location (x, y)

```

1:  $D_1 \leftarrow 0$ 
2: for  $i \leftarrow 0$  to  $N - 1$  do
3:    $\text{CLEAR}(C)$ 
4:    $D_{i\%2} \leftarrow \infty$ 
5:   parfor  $x \leftarrow 0$  to  $X - 1$ ,  $y \leftarrow 0$  to  $Y - 1$  do ▷ shader
6:     for  $f$  in  $F_{xy}$  do
7:       if  $f.\text{frontfacing}$  and  $f.\text{depth} > D_{(i+1)\%2}(x, y)$  and  $f.\text{depth} <$ 
          $D_{i\%2}(x, y)$  then
8:          $C(x, y) = f.\text{color}$ 
9:          $D_{i\%2}(x, y) = f.\text{depth}$ 
10:    $\text{SAVE}(C, D_{i\%2})$  ▷ save  $i$ -th layer of LDI

```

We efficiently render LDI using depth peeling [72]. It is a rendering method originally developed for order-independent transparency, and an LDI is an intermediate product of the algorithm. To generate a N -layer LDI, depth peeling runs the rendering pipeline N times and simultaneously maintains two depth (Z) buffers. One works conventionally, while the other remains a constant at each rendering pass and sets the minimum distance at which a fragment can be drawn without being discarded. For each pass, the previous pass's conven-

Table 4.1: Runtime performance of TensorHolo V2 RGB-D CNNs on GPU and iPhone 13 Pro. The numbers are reported for the CGH synthesis networks with a different number of convolution layers. When the target hologram plane offsets from the midpoint hologram, the addition of the pre-filtering CNN adds 4.8 ms for an TITAN RTX, and 60 ms for an iPhone 13 Pro.

	30 layers	15 layers	8 layers
NVIDIA TITAN RTX	40ms	25ms	17ms
iPhone 13 Pro	478ms	297ms	209ms

tional Z-buffer is used as the current minimal Z-buffer so that the content right behind the previous pass is rendered. The pseudocode of depth peeling is outlined in Algorithm 2.

The CNNs are implemented and trained using TensorFlow 1.15 on an NVIDIA RTX 8000 GPU with Adam optimizer. The hologram synthesis CNN uses the same residual network architecture as the V1 CNN, which consists of 30 convolution layers with $24\ 3\times 3$ kernels per layer (see [224] for the discussion of design choice). The pre-filtering CNN uses the same architecture, but with only 8 convolution layers and $8\ 3\times 3$ kernels per layer. When the target hologram coincides with the midpoint hologram, the pre-filtering CNN can be omitted. The learning rate is 0.0001 with an exponential decay rate of $\beta_1 = 0.9$ for the first moment and $\beta_1 = 0.99$ for the second moment. The first stage training runs for 3000 epochs. The second stage training first pre-trains the pre-filtering CNN 50 epochs for identity mapping and then 1000 epochs jointly with the hologram synthesis CNN. The pre-training accelerates the convergence and yields better results. Both versions of CNN use a batch size of 2, $w_{\text{data}} = 1.0$, $w_{\text{pcp}} = 1.0$, $w_{\text{tgt-pre}} = 0.07$, where $w_{\text{data}}, w_{\text{pcp}}, w_{\text{tgt-pre}}$ are the weights for the data fidelity loss, the dynamic focal stack loss, and the regularization loss. Other parameters remain the same as TensorHolo V1. Table 4.1 lists the runtime performance of TensorHolo V2 RGB-D CNNs on an NVIDIA TITAN RTX GPU and an iPhone 13 Pro. In particular, the speed improvement on the mobile device is following Moore’s law (1.1 Hz to 5 Hz from iPhone 11 Pro to 13 Pro), which could promise real-time performance in a horizon of 5 years if this trend continues. Other dedicated ASICs for CNN, such as Graphcore IPU, Google TPU, and Groq TSP, may also enable efficient edge and cloud hologram computation and streaming for personal entertainment and enterprise

Table 4.2: **Methods comparison on the validation set.** The unbracketed result is calculated for the amplitude map of the predicted hologram, the bracketed result is calculated by averaging the results of a focal stack with 15 most frequent depths and 5 random depths for each image.

Methods	PSNR (dB)	SSIM
Maimone et al. [175]	21.6 (21.3)	0.816 (0.808)
Shi et al. [224]	23.6 (23.2)	0.830 (0.821)
Ours (RGB-D)	29.4 (28.9)	0.945 (0.942)
Ours (LDI)	29.6 (29.1)	0.947 (0.944)

devices.

4.2.7 Simulation Assessment

We qualitatively and quantitatively evaluate TensorHolo V2 CNNs in simulation. Figure 4-20 compares the depth-of-field images reconstructed by the complex hologram prediction of the V1 CNN, the V2 CNNs Stage 1, and the ground truth. The rendered LDIs and real-world captured RGB-D inputs can be found in Fig. 4-19. The V2 RGBD-CNN largely removes the artifacts at the occlusion boundaries, compensates the missing wavefront with a plausible estimation of the occluded pixels, and is robust to depth misalignment in the real-world captured inputs (see Fig. 4-20b). When the foreground of the depth boundary has extraordinary pixels, such as the specular white dot in the yellow inset of the frog scene (see Fig. 4-20a Row 2 Column 3), the V2 RGBD-CNN can mis-extrapolate the occlusion and produce an inaccurately defocused background. The V2 LDI-CNN eliminates this error with the LDI input (see Fig. 4-20a Row 2 Column 4). On the challenging validation set, V2 RGBD-CNN and V2 LDI-CNN perform similarly, both outperforming the V1 CNN and Maimone et al. by a large margin (see Table 4.2). DDPM robustly performs the phase-only conversion at detail-rich, high-frequency, high-contrast, and low-amplitude regions with full automation (see Fig. 4-21a and video results at <http://cgh-v2.csail.mit.edu>). In all examples, DDPM simultaneously achieves high resolution and speckle-free reconstruction. In contrast, AA-DPM and BL-DPM have to trade between resolution and speckle noise via different filter strengths. Benefiting from the display-specific training, DDPM maintains

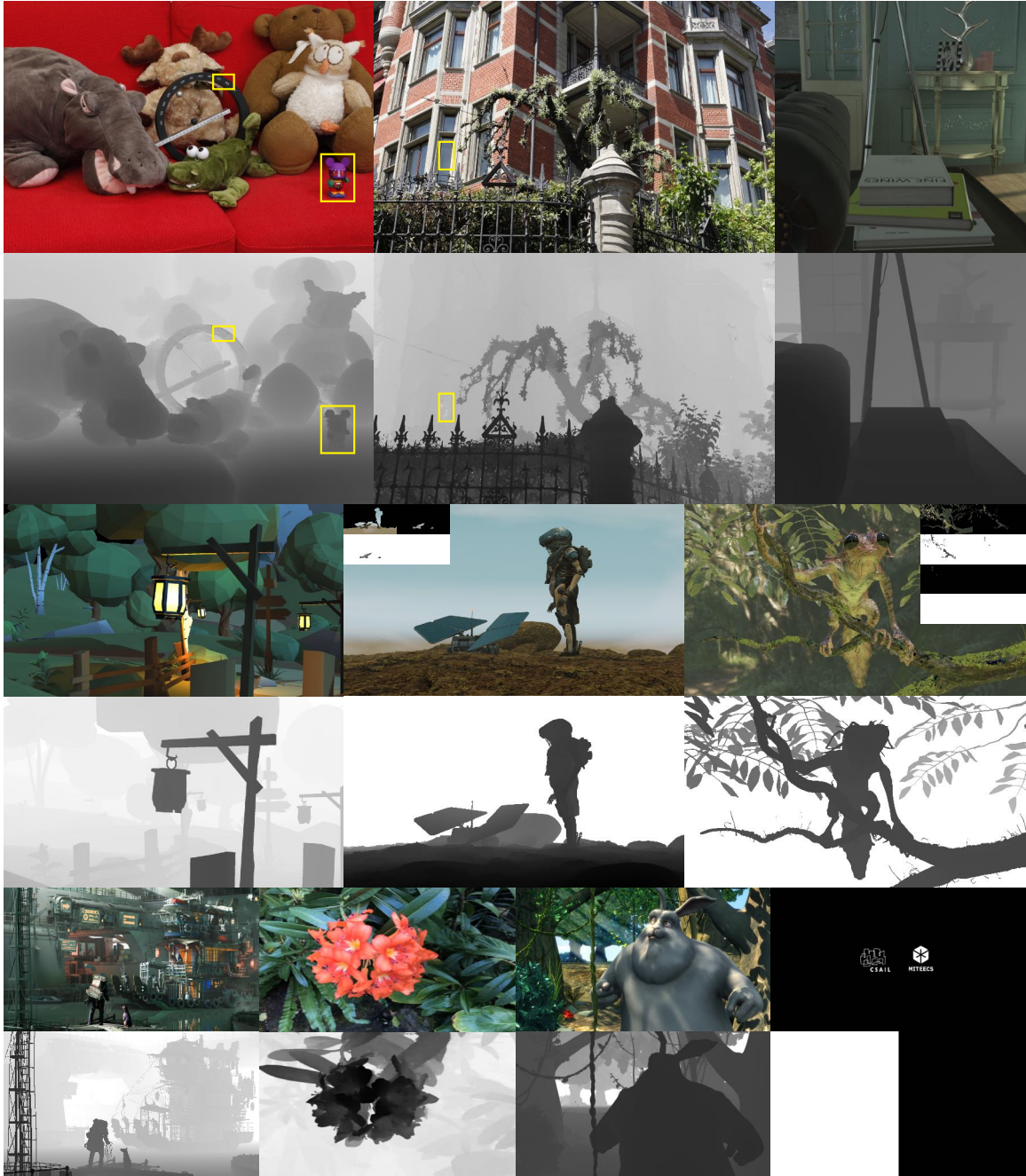


Figure 4-19: **RGB-D and LDI inputs used in this paper.** From top to bottom and left to right: 'Couch' and 'Mansion' from Kim et al. [139], a living room scene from Xiao et al. [258], a forest scene from Padmanaban et al. [200], 'Wanderer' and 'Tree Creature' by Daniel Bystedt, 'PartyTug 6:00AM' by Ian Hubert, 'Flower' from Mildenhall et al. [190], 'Big Buck Bunny' by (© 2008, Blender Foundation), and 'MIT-EECS-Logo' by the authors. 'Couch', 'Mansion', and 'Flower' are real-world captured, and the rest are rendered. 'Wanderer' and 'Tree Creature' are LDIs and the subsequent layers are visualized at the top corners. The yellow boxes mark the regions with obvious depth misalignment or inconsistency, the soft leave edges in the depth map of 'Flower' are not marked.

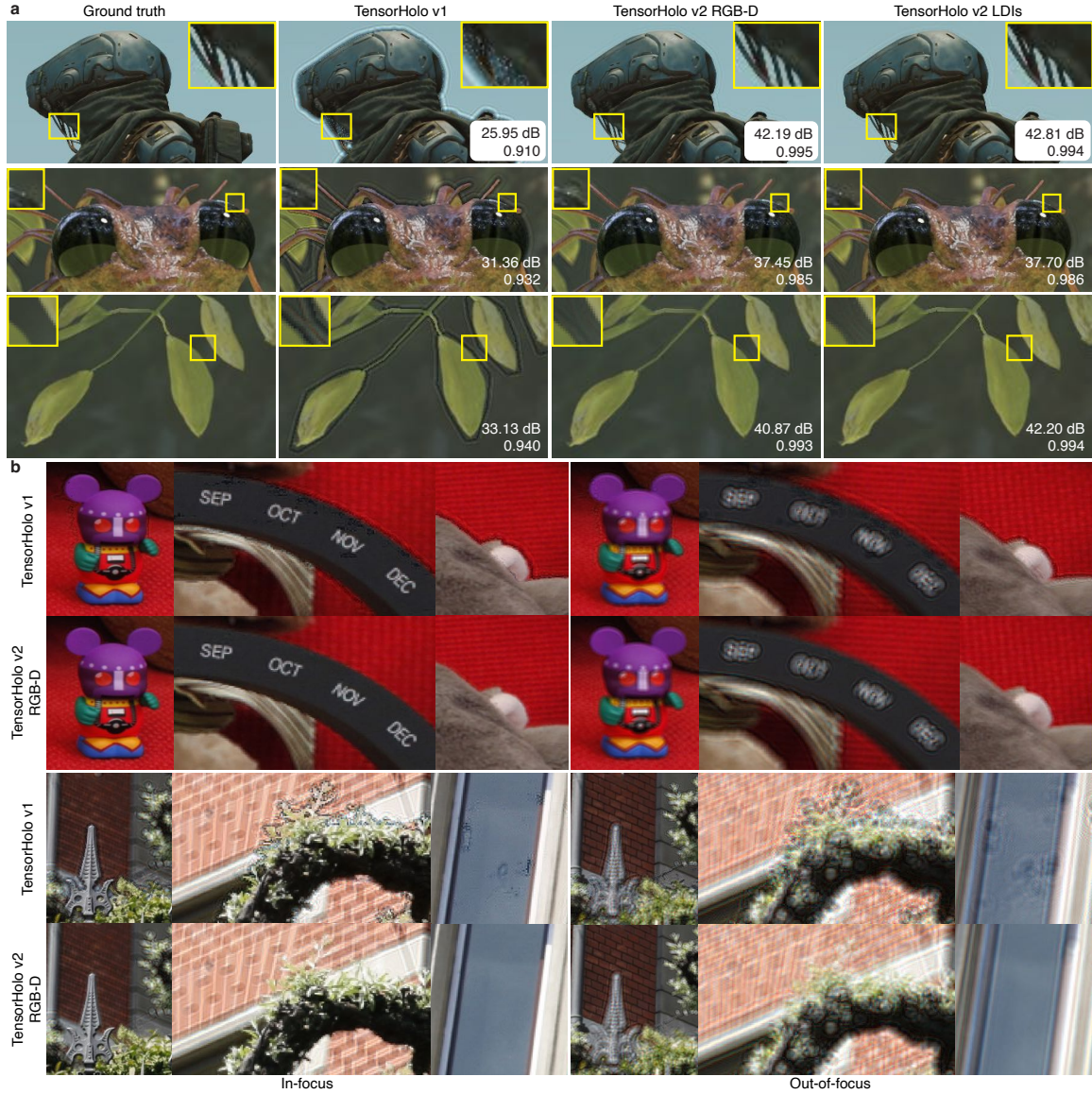


Figure 4-20: **Performance evaluation of TensorHolo v1, TensorHolo v2-RGB-D, and TensorHolo v2-LDIs.** **a** Comparison of 3D holograms for computer-rendered scenes. The yellow inset marks the region of interest. **b** Comparison of 3D holograms for real-world captured scenes.

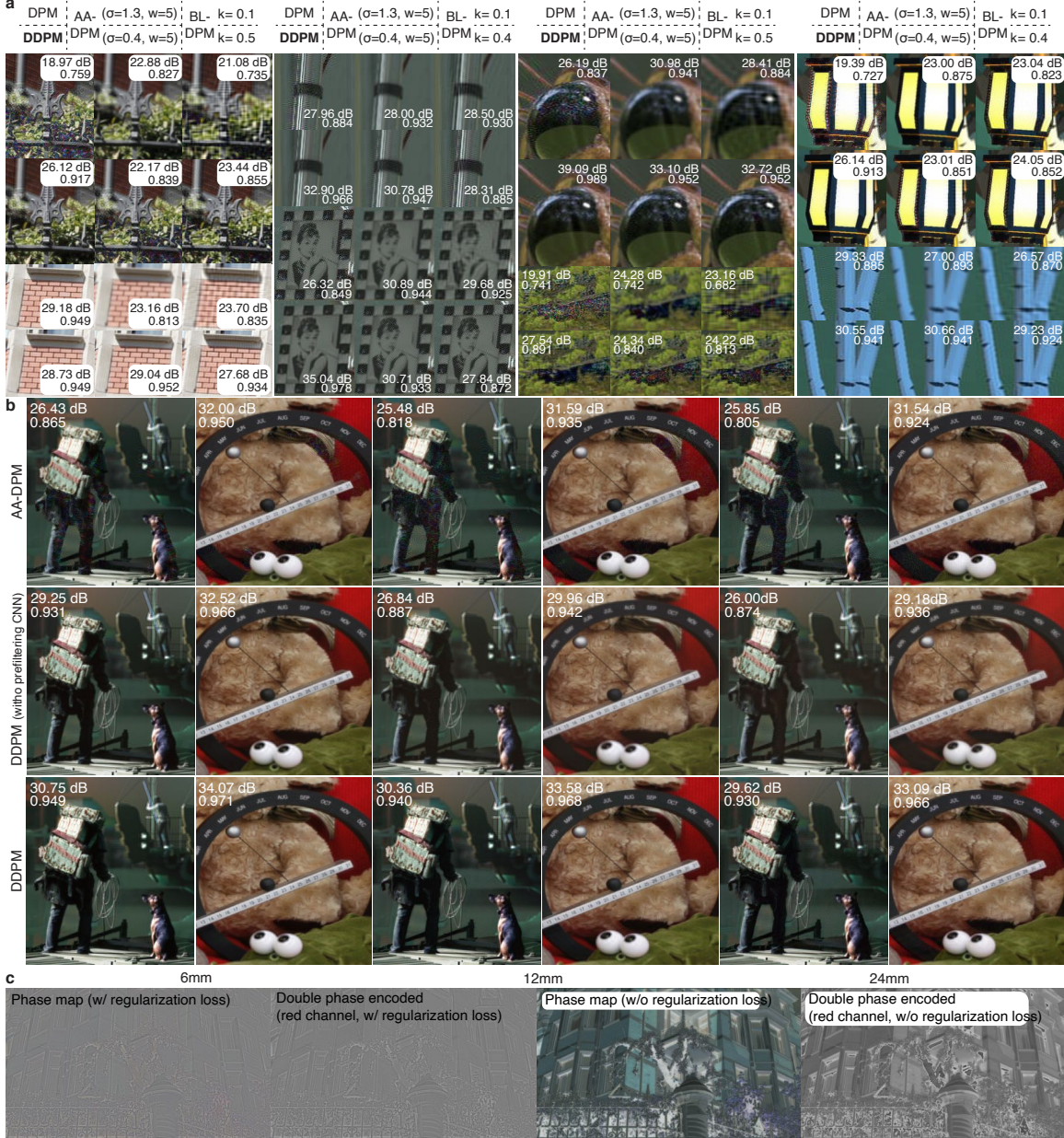


Figure 4-21: Performance evaluation of DDPM. **a**, Comparison of simulated reconstruction using DDPM, AA-DPM, and BL-DPM for midpoint hologram. For AA-DPM and BL-DPM, two filter strengths are applied: the weaker one results in similar sharpness to DDPM; the stronger one results in an artifact-free image. **b**, Comparison of simulated reconstruction using DDPM, DDPM (without the pre-filtering CNN), and AA-DPM for target holograms with 6mm, 12mm, and 24mm offset to midpoint hologram. The filter strengths used for the boat and the couch scene are ($\sigma = 1.0, w = 5$) and ($\sigma = 0.8, w = 3$) respectively. They are the minimal strengths that result in artifact-free images for the midpoint hologram. **c**, Comparison of pre-encoding target hologram phase and its double-phase encoded phase-only hologram predicted by CNNs trained with and without the regularization loss. The hologram plane is offset by 6mm.

the image resolution when hologram planes are offset against the 3D volume (see Fig. 4-21b Bottom), the addition of the pre-filtering CNN preserves the image contrast (see the human limbs and perpetual calendar in Fig. 4-21b Middle). In opposite, using AA-DPM with filter strength optimized for the midpoint hologram is inadequate and produces ringing artifacts at distant hologram planes (see Fig. 4-21b Top). Successful DDPM training benefits from the regularization loss to minimize phase wrapping (see Fig. 4-21c).

4.2.8 Experimental Setup

A benchtop holographic display is constructed and calibrated for experimental verification (see Fig. 4-22). The setup uses a HOLOEYE PLUTO (VIS-014) phase-only LCoS with a resolution of 1920×1080 pixels and a pixel pitch of 8 μm . This SLM provides a refresh rate of 60 Hz (monochrome) with a bit depth of 8 bits. The laser is a FISBA RGBeam single-mode fiber-coupled module with three optically aligned laser diodes at 638, 520, and 450 nm wavelengths. The diverging beam emitted by the laser is collimated by a 300mm achromatic doublet (Thorlabs AC254-300-A-ML) and polarized (Thorlabs LPVISE100-A) to match the SLM's function polarization direction. The beam is directed to the SLM by a beamsplitter (Thorlabs BSW10R), and the SLM is mounted on a linear translation stage (Thorlabs XRN25P/M). When displaying holograms with different relative positions to the 3D volumes, we adjust the linear translation stage to keep the position of 3D volumes stationary and thus avoid modifying the following imaging optics. The modulated wavefront is imaged by a 125 mm achromat (Thorlabs AC254-125-A-ML) and magnified by a Meade Series 5000 21 mm MWA eyepiece. An aperture is placed at the Fourier plane to block excessive light diffracted by the grating structure and higher-order diffractions. A SONY A7M3 mirrorless full-frame camera paired with a 16-35mm f/2.8 GM lens is used to photograph the results. A Labjack U3 USB DAQ is used to send field sequential signals and synchronize the display of color-matched phase-only holograms.

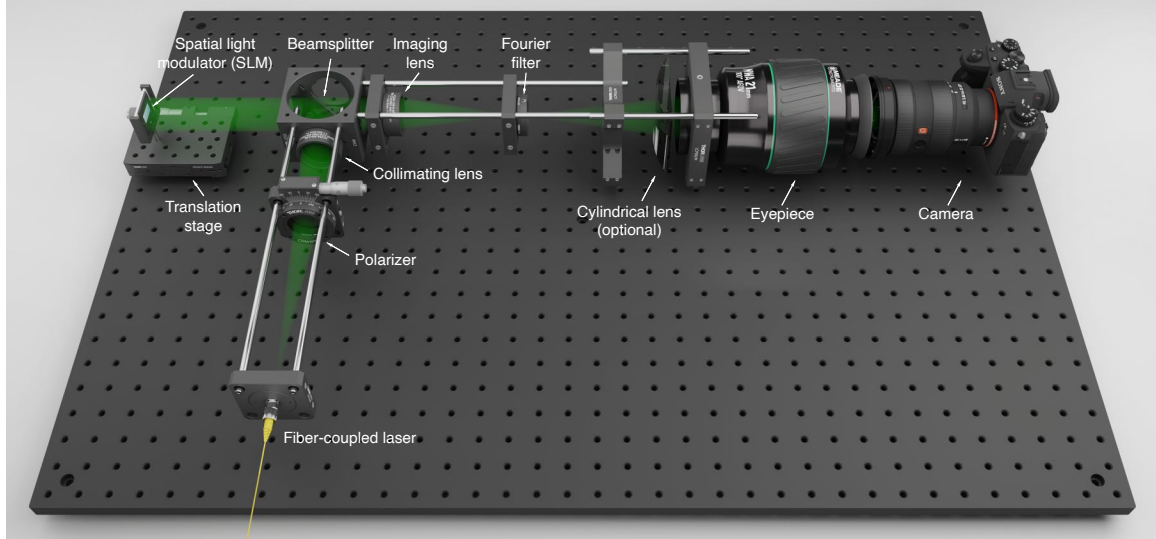


Figure 4-22: **Schematic of the experimental holographic display prototype.** The cylindrical lens is used for the aberration-correction experiment (see Fig. 4-24). The control box of the laser and Labjack DAQ are not visualized.

4.2.9 Experimental Results

We photographed 3D holograms of both rendered and real-world scenes generated by the V2 RGBD-CNN (see Fig. 4-23). The hologram planes are offset differently from the 3D scenes to evaluate the DDPM performance.

Compared to the video result of TensorHolo V1, the Big Buck Bunny result (see Fig. 4-23 top and video result at <http://cgh-v2.csail.mit.edu> for continuous 3D refocusing) no longer present visible dark seams around foreground objects such as the bunny body and tree vines. The mansion result (Fig. 4-23 middle) validates TensorHolo V2’s robustness to depth misalignment. Unlike Fig. 9 of TensorHolo V1, artifacts on the blue window, fence, and shrubs are eliminated, and the resulting 3D image is more naturally defocused. Reconstructed at a further distance, the boat result (Fig. 4-23 bottom) reproduces smooth color gradients of the wall and the water surface owing to accurate phase control from a per-pixel voltage-to-phase look-up table (see Methods for details). The human and dog in the foreground are absent from speckle noise that would otherwise appear if AA-DPM was used (see Fig. 4-21b). In all results, the V2 RGBD-CNN consistently produces realistic depth boundaries using a single RGB-D input. Video results at <http://cgh-v2.csail.mit.edu> further demonstrate real-time computation and playback of 3D holograms calculated for



Figure 4-23: **Experimental demonstration of holographic 3D projections.** Two computer-rendered (Top/Bottom) and one real-world captured (Middle) scenes are displayed and photographed for target holograms located at three different distances (marked on the left) to the midpoint hologram. The yellow box marks the foreground object of interest, and the blue box marks the background object of interest. Readers are encouraged to zoom in to examine the image details.

cellphone-captured RGB-D videos, where the V2 RGBD-CNN robustly handles depth misalignment. To demonstrate aberration correction, we place a cylindrical lens with a focal length of 200 mm in front of the eyepiece to cause astigmatism (see Fig. 4-22). The cylinder axis is placed vertically, causing the originally in-focus object to primarily exhibiting horizontal blurs. We performed system identification (see Methods for details) and generated an aberration-corrected dataset with 256 (8-bit) depth levels. The trained network predicts holograms that compensate for aberration for both the random scene in the validation set (see Fig. 4-24a), natural scenes, and test patterns (see Fig. 4-24b). The captured front focus image in the tree frog scene closely resembles the simulation of an unaberrated system in Fig. 4-21a.

4.2.10 Discussion

TensorHolo V2 makes a step towards the end-to-end synthesis of 3D phase-only holograms. It is fully automatic, robust to rendered and misaligned real-world inputs, produces realistic depth boundaries, and corrects vision aberrations. By reusing the minimalistic CNN architecture in TensorHolo V1, it runs in real-time on a consumer-grade GPU and 5 FPS on an iPhone 13 Pro (see Methods for runtime performance details), promising real-time mobile performance in future-generation AR/VR headsets and glasses.

To further produce a more visually engaging 3D experience and reduce the computational complexity, many extensions to the current method are worth investigating. One is foveation-guided holographic rendering [42, 36, 40, 251]. For stereoscopic and light-field AR/VR displays, foveated rendering lowers the image quality at the peripheral vision to reduce rendering cost [96, 202, 241]. In the context of holographic rendering, a 2D hologram with synthetic blur can be used for peripheral vision instead of a true 3D hologram. As eye trackers become widely available in next-generation head-mounted displays, this can be a powerful computation-only approach to improve the rendering performance.

Another direction to be reiterated here is to support view-dependent effects. Although an LDI provides sufficient scene information from one perspective, view-dependent effects are not explicitly modeled since disoccluded regions or out-of-view objects will become

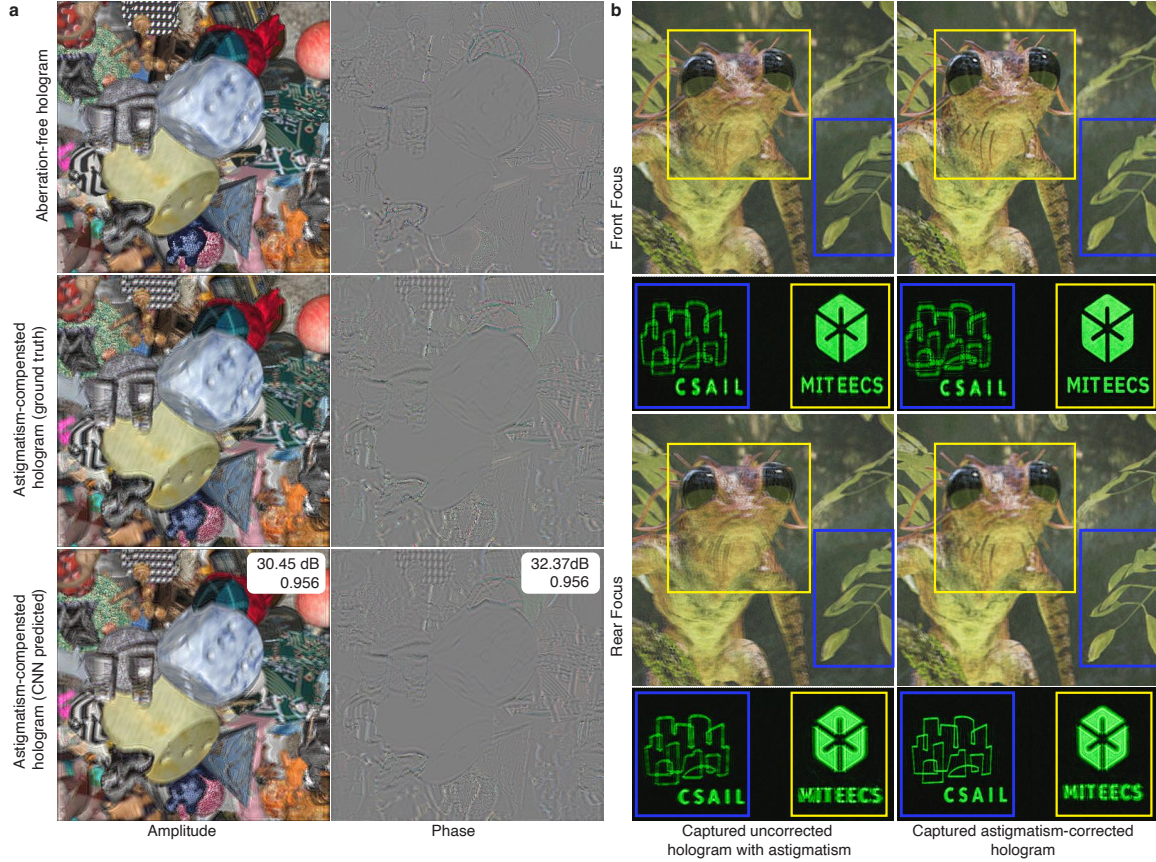


Figure 4-24: Performance evaluation of vision aberration correction. **a** Top: A hologram calculated for normal vision. Middle: A hologram calculated using the proposed variant of SM-LBM that corrects a synthetic astigmatic vision induced by a 200mm cylindrical lens. The calibration method is detailed in Methods. Bottom: The CNN trained on the astigmatism-corrected dataset jointly performs diffraction simulation and aberration correction. The PSNR and SSIM of the predicted hologram are marked on the top right corner. **b** Experimentally captured CNN-predicted holograms of a natural scene and a test scene without (Left) and with (Right) astigmatism correction. The yellow and the blue box marks the foreground and the background object of interest, respectively. The in-focus object in the corrected hologram exhibits horizontal blurs in the uncorrected hologram (i.e., treefrog’s tentacles, eye reflections, tree leaves, and test patterns).

visible from other views, as well as occlusion of currently visible points. However, such localized 3D experience may be sufficient for head-mounted displays since holograms can be dynamically updated to reflect the changed user’s viewpoint as the rendering of LDI is efficient. Yet, a hologram that supports a view-dependent effect is beneficial when fabricating ultra-high-resolution static copies. To design a scene representation sufficient for modeling view-dependent effect, one could render multiple LDIs from evenly sampled viewpoints within an arc or a line of view space, and compact LDIs into a master LDI plus side information of disocclusion, new occlusion, and out of view scene through point cloud duplication detection. One could also replace LDI with recently emerged neural scene representation (i.e., NeRF [191]), which uses a coordinate-based neural network to compactly memorize the entire 3D volume. For hologram rendering, Zhang et al. [271] showed that Fourier domain segmentation and stitching of multiple holograms rendered from different viewports could provide a continuous view-dependent effect for large baseline movement. Co-designing a learning-based system to incorporate both new input representations and new rendering schemes can further unleash the power of holograms when higher resolution SLMs become available.

The current system also requires a Fourier filter to shield higher-order diffractions for producing a clean image. Changing the image formation model to model higher-order diffractions explicitly can potentially remove the need for optical filtering [94] and increase the method compatibility to enable more flexible display designs. Meanwhile, the current ground truth focal stack is rendered under the assumption of a coherent imaging model, the real-world depth of field yet follows the incoherent imaging model. This can create a mismatch of defocus blur when overlaying virtual objects with real-world scenes. Using an incoherent rendering model to generate the target focal stack while matching it under the constraint of the coherent propagation model may offer a more realistic 3D perception without breaking holography’s principle of operation. This can be further combined with temporal time-multiplexing to improve the image quality [153]. We discuss these improvement in Chapter 6.

The current system uses Maimone et al.’s phase initialization to suppress speckle noise; however, the artificially-enforced smooth phase distribution could computationally con-

strain the angular spectrum and lead to reduced eyebox. A potential way to maximize the eyebox is to add pupil modeling in the Fourier plane during the unsupervised training stage. Because the pre-encoding phase is not forced to match the target hologram, the filtering of eccentric pupils can encourage a broad angular spectrum bandwidth to maintain image quality for different pupil locations while keeping the speckle low.

When miniaturizing a holographic display into an eyeglass-like form factor, one must deal with spatially-varying optical aberrations[175] inevitably. A successful demonstration of a learning system that produces aberration-corrected content for an eyeglass-like prototype will be necessary for industry adoption of holography.

Finally, the proposed system provides a limited immersive 3D experience due to the SLM’s low space-bandwidth product. Consequently, the development of SLMs with higher pixel density, faster refresh rate, higher bit depths, and possibly joint amplitude and phase modulation are imperative. Designing compact combining optics (i.e., holographic optical element (HoE) lens, geometric phase lens, and waveguides) with less aberration/more substantial optical power and novel optical path folding structures are also critical to the success of miniaturizing the display form factor [176, 216].

4.3 Neural Compression for Hologram Images and Videos

As of the 2020s, over 80% of internet traffic is dedicated to transmitting images, videos, and live streams, reflecting a shift in user preferences from text-based to visual content. Should the promises of holography be fulfilled in the near future, we anticipate a further shift from pixels that simply depict the 2D slice of 3D space to pixels that record intangible wavefronts that directly encoding the underlying 3D environment. While the formality of the content remains as 2D images, the existing data compression algorithms may no longer be suitable for CGHs, as they possess distinct characteristics from natural images. Naively applying these algorithms may lead to low compression rates and large compressed files, resulting in significantly increased ingress and egress costs, especially for high-resolution content (e.g., 4K and 8K) tailored for virtual and augmented reality applications.

In the meantime, designing new compression schemes for CGHs by following the ex-

isting compression routines of block-based segmentation, frequency space encoding, and dropping high-frequency components can be sub-optimal, as high-frequency interference fringes in CGHs could be even more critical than other components for the presentation of 3D images. Since CGHs encode the underlying 3D images in a non-WYSIWYG manner, hand-crafting compression algorithms becomes challenging due to the difficulty of inferring the function of seemingly random interference patterns. With the rise of cloud-based streaming as the new standard for content consumption, this chapter devises a neural network-based framework for hologram encoding and decoding that yields higher compression efficiency. This framework also distributes computation between the cloud server and the edge, with the goal of reducing both data transmission and computation on both ends.

4.3.1 Motivation

As shown in previous sections, a CGH represents a 2D slice of the optical wavefront as a complex-valued image, which differs from the domain where the conventional image and video lie. A CGH also necessitates high-frequency interference fringes to produce a realistic depth of field effect. In contrast, conventional image and video codecs discard visually insignificant high-frequency details to achieve a higher compression rate. These unmatched design choices lead to suboptimal compression performance when applying existing codecs to CGHs. For practical virtual and augmented reality applications, the ultra-high-resolution CGH (16K+) needed to simultaneously support a wide field of view, and a large eye box could further magnify codec inefficiency and impose a significant challenge. There have been many efforts to tackle CGH compression. Quantization and transform coding are the two main compression techniques. Quantization schemes employ reduction of bit level [247], non-uniform coding [231], and vector quantization [246, 260], while transform coding uses Discrete Cosine Transform (DCT) [219, 209], Discrete Wavelet Transform (DWT) [232, 46], and their variants [25, 68] to find compact basis. Disposing of the hologram amplitude further produces a phase-only hologram. Alternative phase representation [269], phase difference representation, and neural networks [143] can en-

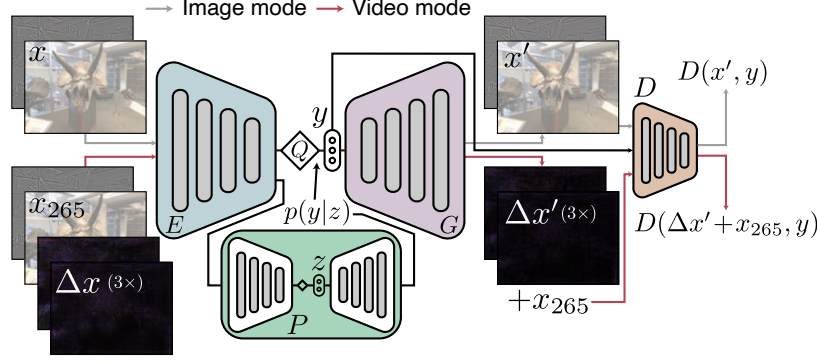


Figure 4-25: HiFiHC pipeline for hologram image and video compression. For image compression, the encoder E encodes one latent code for the hologram’s real and imaginary components. The latent code is quantized by Q , entropy coded with side information generated through P , decoded by G , and classified by D . For video compression, E takes an H.265 compressed frame with its associated residual and encodes a latent code only for reconstructing the residual. The reconstructed residual is added back to the H.265 frame.

code such holograms more efficiently. JPEG Pleno Holography provides a uniform compression standard for digital holography, including common test conditions for proponent methods [194]. A majority of these works are designed for random phase holograms with noise-like statistics that enables a wide field of view, while recent works [175, 224] suggest that smooth phase holograms with natural-image-like portfolios may induce less severe speckle and facilitate rapid phase-only encoding [175, 224, 111, 10]. Despite the pros and cons when applying either type of hologram to different applications, compression algorithms specialized for smooth-phase holograms are not well exploited.

Inspired by recent advances in deep optics [235], neural compression for conventional image/video [16, 192, 188], and large-scale hologram datasets [224], we propose High Fidelity Hologram Compression (HiFiHC), a learning-based end-to-end compression method for smooth-phase complex hologram images and videos targeted for near-eye displays with eye-piece, where hologram has limited depth range. For hologram images, an encoder network is learned to compress the hologram into a low-dimensional latent code, and a decoder network reconstructs the hologram. For hologram videos, High Efficiency Video Coding (HEVC, a.k.a. H.265) with a high constant rate factor compresses the low and mid-frequency content, while an encoder-decoder network compresses the high-frequency residual critical for the 3D image formation with assistance of motion vectors in the H.265

video.

4.3.2 High Fidelity Hologram Compression (HiFiHC)

Figure 4-25 shows the pipeline of HiFiHC. The architecture of HiFiHC repurposes High Fidelity Compression (HiFiC) [188], a network designed for conventional image compression, adjusted for hologram input, dataset, and to incorporate domain-specific loss. Specifically, it consists of an encoder E , a generator G , and a discriminator D . HiFiHC takes a 6-channel tensor input $x \in \mathbb{R}^{6 \times R_x \times R_y}$ created by concatenating the real and imaginary part of the complex hologram, where R_x and R_y are the spatial resolution of the hologram. The encoder produces a quantized latent $y = E(x)$, which is further decoded by the generator G to obtain a lossy reconstruction $x' = G(y)$. Using a probability model $\Pr$ and an entropy coding algorithm (e.g. arithmetic coding [179]), the latent y can be stored losslessly using bitrate $r(y) = -\log(\Pr(y))$. The discriminator produces a scalar value $D(y, x)$ and $D(y, x')$, indicating the probabilities that x and x' are real or fake (synthetic), respectively.

We train HiFiHC as a conditional generative adversarial network (GAN), where the the (E, G) tries to “fool” D into believing its lossy reconstructions are real, while D aims to classify the lossy reconstructions as fake and the uncompressed inputs as true. The loss function for training (E, G) is given by

$$\begin{aligned} \mathcal{L}_{(E,G)} = & w_r r(y) + w_{holo} \|x - x'\|_1 + w_{fs} d_{fs}(x, x') \\ & - w_D \log(D(x', y)). \end{aligned} \quad (4.33)$$

Here w_r, w_{holo}, w_{fs} , and w_D are the hyper-parameters controlling the trade-off between the terms, d_{fs} defines a dynamic focal stack loss following Shi et al. [224] that encourages the focal stack reconstructed from the compressed hologram to match the one from the uncompressed input.

The loss function for training D is

$$\mathcal{L}_D = -\log(1 - D(x', y)) - \log(D(x, y)), \quad (4.34)$$

which encourages the uncompressed hologram to be classified as 1 (True) and the compressed hologram as 0 (Fake).

For hologram video compression, compressing each frame to an individual latent prevents exploitation of temporary redundancy. Instead, we compress amplitude and phase into two regular videos using H.265 with a constant rate factor (CRF) of 23, which is insufficient for preserving fine interference fringe to recover the 3D scene details. Nevertheless, Predicted frames (P-frames) and Bidirectional predicted frames (B-frames) in each video encode the motion vectors, which can be leveraged during the residual learning. Denote the H.265 compressed frame x_{265} (converted to real+imaginary representation) and the residual $\Delta x = x - x_{265}$. We train a HiFiHC network to predict Δx with a 12-channel tensor input created by concatenating Δx and x_{265} along the channel dimension. Let $\Delta x'$ be the compressed residual, Δy be the latent of Δx , we train HiFiHC using an updated loss for (E, G)

$$\begin{aligned}\mathcal{L}_{\Delta(E,G)} = & w_{\Delta r} r(\Delta y) + w_{\Delta hol} \|\Delta x - \Delta x'\|_1 \\ & + w_{\Delta fs} d_{\Delta fs}(\Delta x + x_{265}, \Delta x' + x_{265}) \\ & - w_{\Delta D} \log(D(\Delta x' + x_{265}, \Delta y))\end{aligned}\tag{4.35}$$

and an updated loss for D

$$\mathcal{L}_D = -\log(1 - D(\Delta x' + x_{265}, \Delta y)) - \log(D(\Delta x + x_{265}, \Delta y)).\tag{4.36}$$

When encoding the residual frames, we consider one group of pictures (GOP) as a batch of frames. The residual of the GOP's Intra-coded frame (I-frame) is first compressed and denoted as $\Delta x'_I$. Let x_P be the anchored P-frame, and x_{265_P} be the H.265 compressed frame. We compute the motion-compensated P-frame residual as

$$\Delta x_P = x_P - (x_{265_P} + \text{warp}(\Delta x'_I, M_{I \rightarrow P})),\tag{4.37}$$

where $M_{I \rightarrow P}$ is the motion vector from the I-frame to the P-frame. Let $\Delta x'_P$ be the compressed residual and $\overline{\Delta x_P} = \Delta x'_P + \text{warp}(\Delta x'_I, M_{I \rightarrow P})$, the motion-compensated residual of

an anchored B-frame is given by

$$\Delta x_B = x_B - (x_{265_B} + \text{warp}(\Delta x'_I, M_{I \rightarrow B}) + \text{warp}(\overline{\Delta x_P}, M_{P \rightarrow B})), \quad (4.38)$$

where $M_{I \rightarrow B}$, $M_{P \rightarrow B}$ are the motion vector from the I-frame to the B-frame and the P-frame to the B-frame, respectively.

We train both versions of HiFiHC on the MIT-CGH-4K V2 dataset [226]. For the video version, we fine-tune the model using holograms rendered from the video sequence used in Xiao et al. [257]. To be used with a practical holographic near-eye display, we configure holograms in the dataset to have a 15mm propagation distance from the hologram to the 3D content. Still, we also evaluate HiFiHC performance at other propagation distances. Following HiFiC, we use the hyper-prior model [16] for Pr, and we pretrain (E, G) for 1 million iterations using $\mathcal{L}_{(E,G)}$ (without the last term) before training the full conditional GAN for another 1 million iterations. At evaluation, we use mv-extractor [27] to extract the motion vector.

4.3.3 Assessment

For hologram images, we evaluate HiFiHC against High Efficient Image File Format (HEIC) [2] and Better Portable Graphics (BPG) [1]. We omit JPG since it showed far worse performance. Figure 4-26 shows a comparison of a rendered boat scene and a real-world mansion scene evaluated by a model trained to yield results at a compression rate of ~ 0.3 bit per pixel (bpp). In the less cluttered boat scene, ripple-like interference fringes surrounding the subjects (i.e., rope, boatman, dog) are visible and relatively isolated. HiFiHC faithfully preserves mid and high-frequency fringe details amid smeared subject content due to defocusing. In contrast, the mid-frequency fringes (i.e., around the dog and dock man) are largely obviated in HEIC and BPG results. The structure of high-frequency fringes is broken due to the block-based compression pattern, and the contrast of rainbow-like intensity alternation is reduced. Consequently, the refocused depth of field images exhibits reduced sharpness and increased ringing artifacts. The highlighted mansion fence inset possesses heavily intermingled features due to interlaced depths. The interaction between randomly-

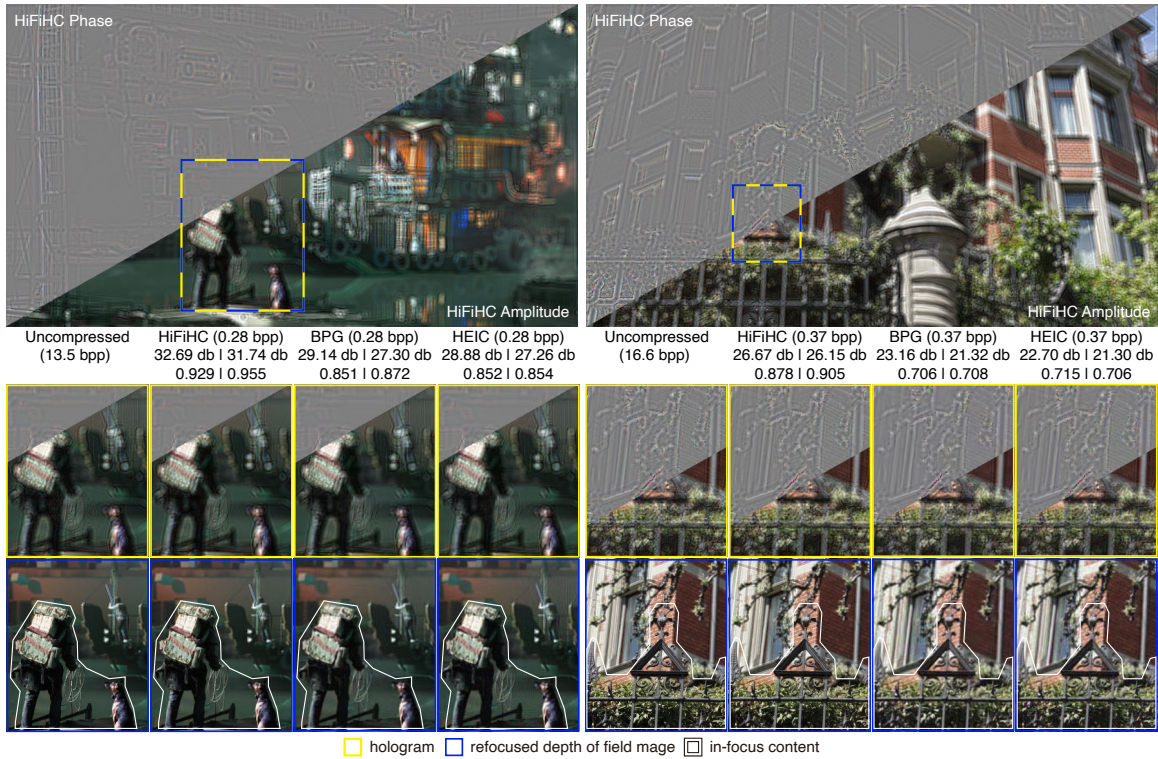


Figure 4-26: Comparison of HiFiHC, HEIC, and BPG performance on hologram images. Readers are encouraged to zoom in and examine details. The second and the third row in each label mark the peak signal to noise ratio (PSNR) and structure similarity index (SSIM) for the hologram amplitude (first number) and the refocused DoF image (second number). Source images: PartyTug 6:00AM (left) by Ian Hubert, and Mansion (right) from Kim et al. [139]. More discussion to be added.

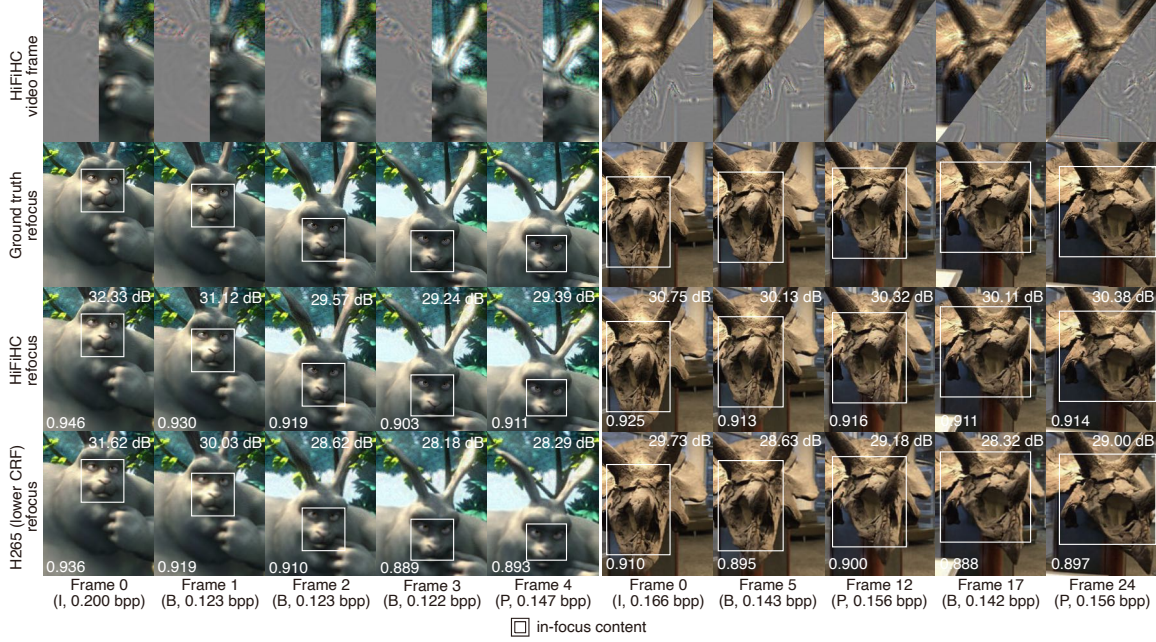


Figure 4-27: Comparison of HiFiHC and H.265 (at lower CRF) performance on hologram videos. Readers are encouraged to zoom in and examine details. In each inset, the top right and bottom left numbers mark the PSNR and SSIM for the refocused DoF image. The second row in the frame label marks the frame type and the bpp of the HiFiHC latent code. Source images: Big Buck Bunny (top) by Blender Foundation, and Horns (bottom) from Mildenhall et al. [190]. The H265 (lower CRF) results use CRF of 15 and 18 for Big Buck Bunny and Horns respectively, both of which yield a similar amount of additional bpp compared to HiFiHC.

oriented shrub leaves makes the aggregated fringes away from contouring shapes. Despite challenging, HiFiHC reasons such interaction through the dynamic focal stack loss and retains the dominating features for producing a sharp foreground after refocusing. Instead, BPG and HEIC significantly underperform, especially when evaluated in the image domain and the refocused results appear nearly uniformly blurred. Evaluation of additional image scenes, offset distances, and rate-distortion curves are visualized in Fig. 4-28/4-31/4-32).

For hologram videos, we evaluate HiFiHC against H.265 videos encoded at a lower CRF that incurs the same amount of additional per-frame bpp. Figure 4-27 shows a comparison of sequences of a rendered bunny scene and a real-world fossil scene evaluated by a model trained to yield results at a ~ 0.15 bpp per hologram. In the selected frames of the bunny scene, the bunny takes a fast motion with non-rigid deformation, whereas the background stays relatively stationary. HiFiHC reduces 27%/39% of bpp for P/B-frames

with motion compensation, and the eyes of the bunny remain sharply focused across the GOP regardless of the frame type. In contrast, the eye sharpness noticeably degrades in the P/B-frames of the lower CRF video. In the selected frames of the fossil scene, the camera undergoes a revolving motion, and all pixels translate with a scaled inverse proportional to the distance to the camera. For this challenging case, HiFiHC still gains a 6%/14% reduction of bpp through motion compensation. Although HiFiHC is trained solely on the residual of a synthetic dataset, it well handles real-world scenes with sharper DoF images across all frame types, relatively. Evaluation of additional video scenes, offset distances, and rate-distortion curves are visualized in Fig. 4-29/4-30/4-31/4-32.

When constructing practical holographic head-mounted displays (HMDs), the 3D image formed by the hologram locates approximately one focal length behind the eyepiece, whereas the SLM is typically placed closer to reduce the display form factor. This induces a propagation distance from the hologram on the SLM to the volume of the 3D image, and this distance can vary based on different design choices and targeted applications of the HMDs. Here, We evaluate HiFiHC performance at various propagation distances. Specifically, we train HiFiHC models at 5mm, 10mm, 15mm, and 20mm propagation distances from the hologram to the 3D image. Figure 4-31 illustrates the compression performance of HiFiHC versus the conventional codecs. For both image and video compression, HiFiHC maintains the performance gain across all the tested propagation distances. In image compression, HiFiHC shows a slower performance drop when the propagation distance is prolonged. This is because conventional codecs are only optimized for images with natural statistics, and the hologram statistics differ more from it as the propagation distance prolongs. In contrast, the independently-trained HiFiHC model is less sensitive to the hologram statistics changes. For video compression, our hybrid approach of using HiFiHC to compute residual for a high CRF video has to suffer the performance drop of conventional codec. However, it still consistently outperforms a lower CRF video by 1dB or more. Figure 4-32 visualizes the rate-distortion(RD) curve for both image and video compression evaluated at the hologram plane and a stack of object planes, for later of which five layers are evenly sampled through the object space, and the mean PSNR is reported. The image RD curves exhibit very different statistics from the video RD curves. For the image RD

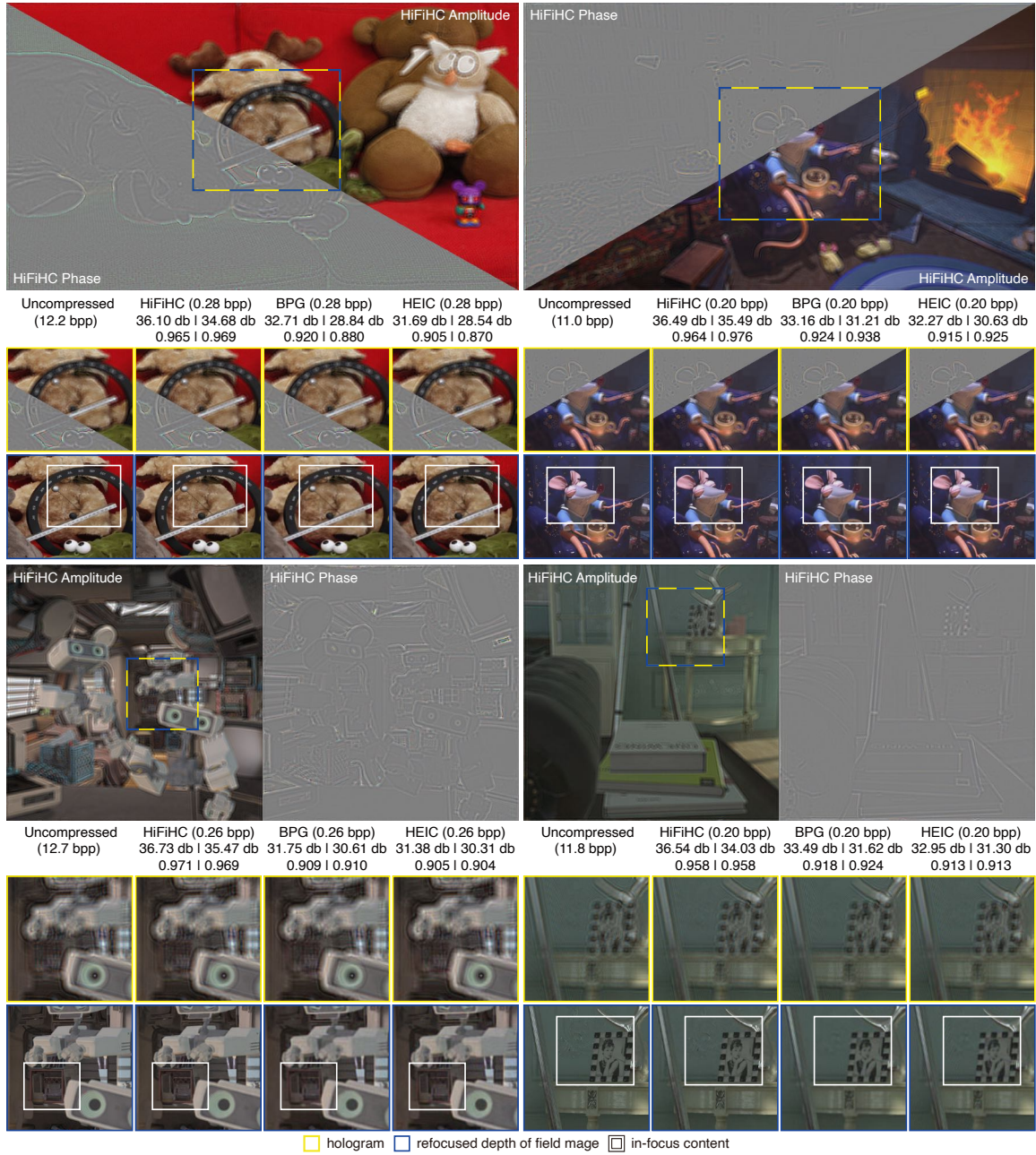


Figure 4-28: Additional comparison of HiFiHC, HEIC, and BPG performance on hologram images. Readers are encouraged to zoom in and examine details. The second and the third row in each label mark the peak signal to noise ratio (PSNR) and structure similarity index (SSIM) for the hologram amplitude (first number) and the refocused DoF image (second number). Source images: Couch (top left) from Kim et al. [139], Mr. Elephant by Glenn Melenhorst, Robot (bottom left), and Home (bottom right) scene from Xiao et al. [257]

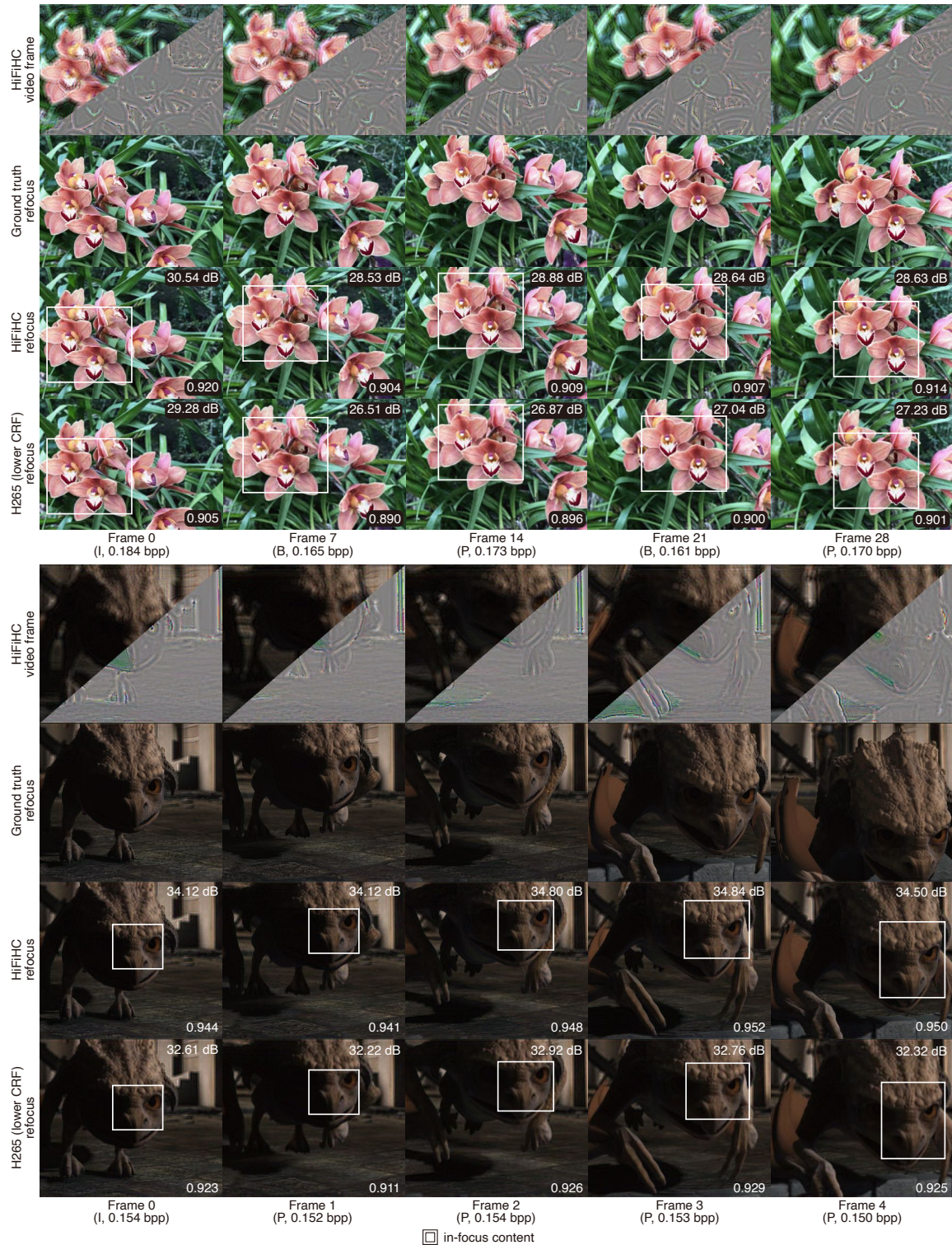


Figure 4-29: Additional comparison of HiFiHC and H.265 (at lower CRF) performance on hologram videos. Readers are encouraged to zoom in and examine details. The top and bottom numbers in each inset mark the PSNR and SSIM for the refocused DoF image. The second row in the frame label marks the frame type and the bpp of the HiFiHC latent code. Source images: Orchids (top) from [190], and Market_5 (bottom) from MPI Sintel dataset [34]. The H265 (lower CRF) results in the use CRF of 18 and 19 for Orchids and Market_5, respectively, both of which yield a similar amount of additional bpp compared to the HiFiHC.

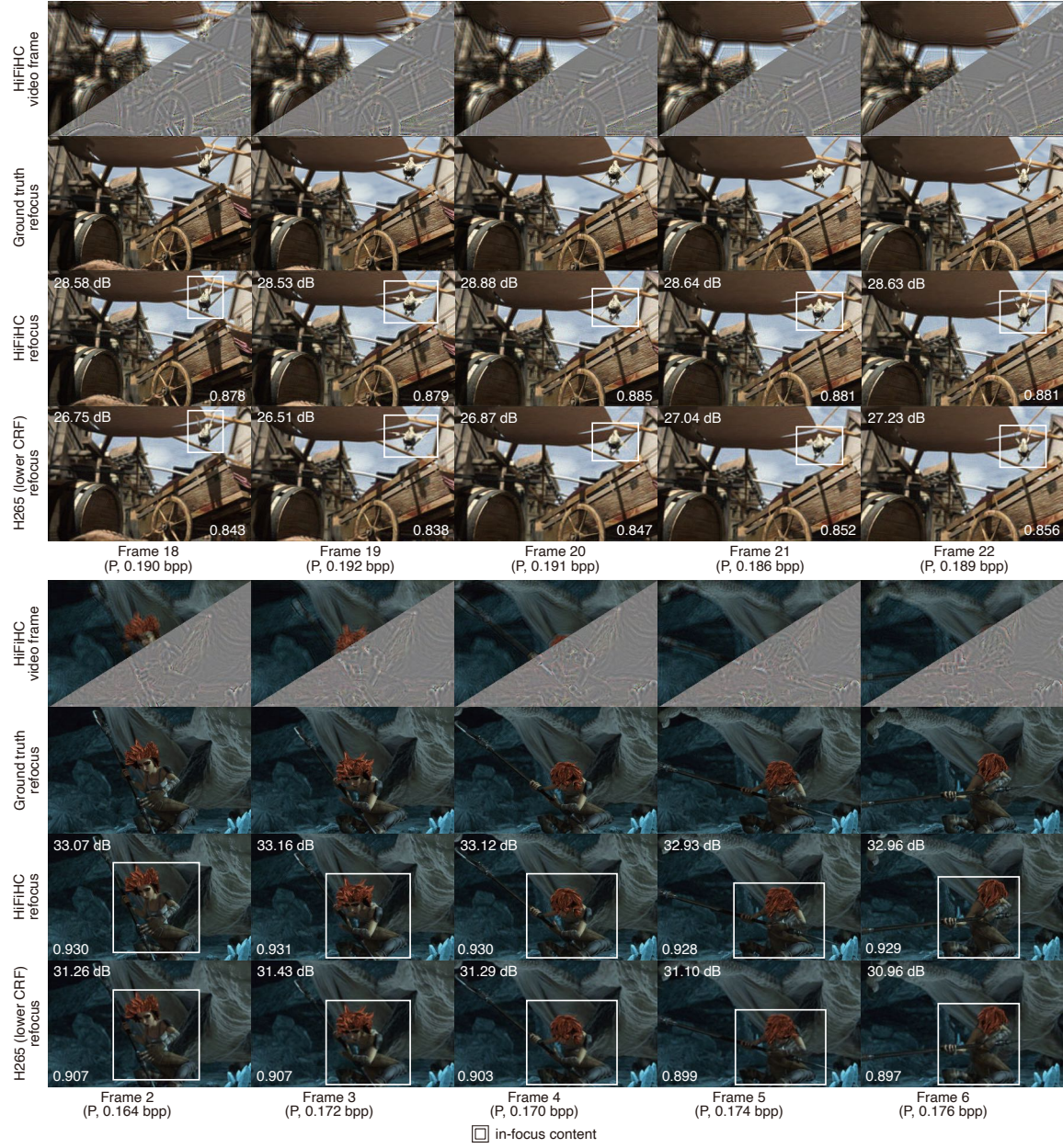


Figure 4-30: Additional comparison of HiFiHC and H.265 (at lower CRF) performance on hologram videos. Readers are encouraged to zoom in and examine details. Each inset's top and bottom numbers mark the PSNR and SSIM for the refocused DoF image. The second row in the frame label marks the frame type and the bpp of the HiFiHC latent code. Source images: Market_6 (top) and Cave_4 (bottom) from MPI Sintel dataset [34]. The H265 (lower CRF) results use CRF of 20 and 20 for Market_6 and Cave_4, respectively, both of which yield a similar amount of additional bpp compared to the HiFiHC.

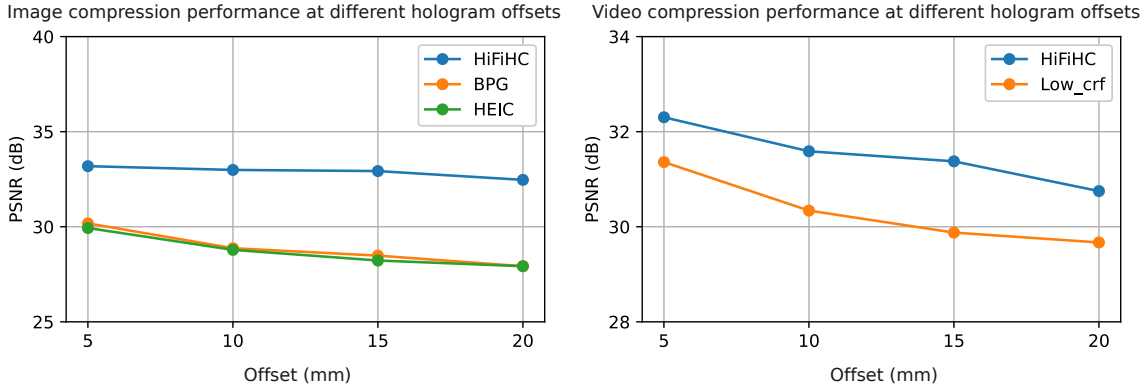


Figure 4-31: Performance comparison of hologram image compression (left) and video compression (right) under different offsets between the 3D volume and the hologram. The PSNRs are calculated for the refocused insets in Fig. 2, Fig. 4-28 (for image compression); Fig. 3, Fig. 4-29, Fig. 4-30 (for video compression). For both hologram image and video, the PSNR decreases as the offset increases.

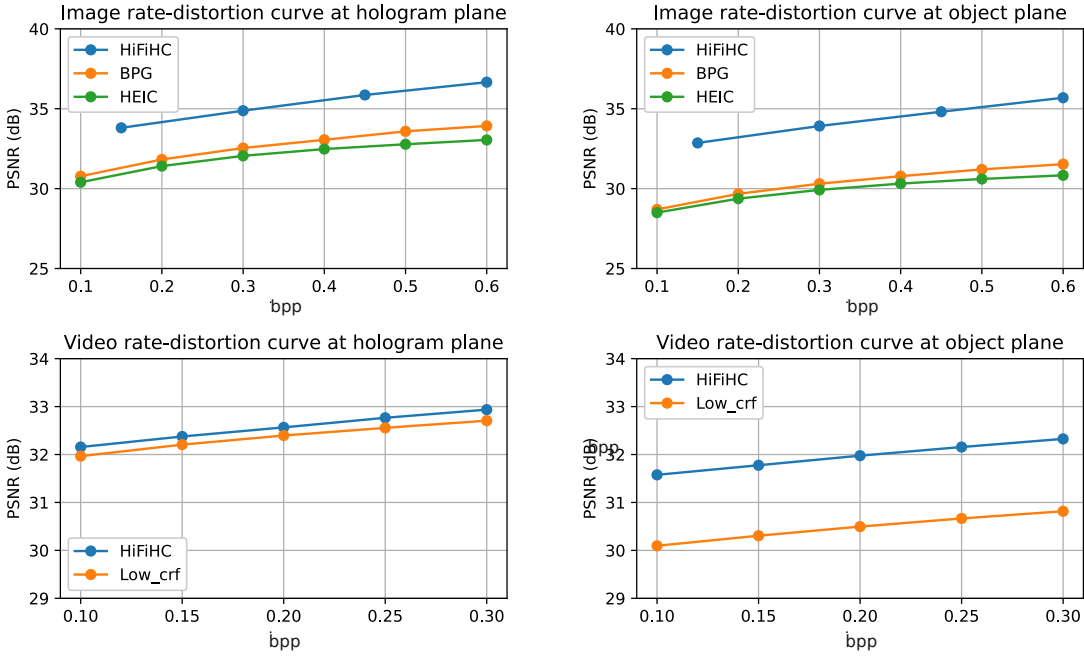


Figure 4-32: Rate-distortion curves for image and video compression evaluated at the hologram plane and a stack of object planes. For object planes, five layers are evenly sampled through the object space and the mean PSNR is reported. The PSNRs are calculated for the scenes in Fig. 2, Fig. 4-28 (for image compression); Fig. 3, Fig. 4-29, Fig. 4-30, and additional test scenes from Xiao et al. [257] (for video compression).

curves, HiFiHC achieves a $\sim 3\text{dB}$ improvement over BPG and HEIC at the hologram plane and a $+4\text{dB}$ improvement at the object plane. The relative improvement difference at the hologram plane and the object planes is minor. In contrast, video-version HiFiHC only achieves a $\sim 0.2\text{dB}$ improvement at the hologram plane but a $+1.4\text{dB}$ improvement at the object plane, resulting in a big relative improvement difference. This is caused by the fact that, in video compression, the CNN is only responsible for reconstructing the residual, which contributes little to the overall intensity of the image. However, the residual often presents the high-frequency information critical to the refocusing power of the hologram. The focal stack loss encourages the reconstructed residual to preserve the most essential details for refocusing, leading to a significant performance gap at the object planes that requires more than a 0.3bpp increase to match. We note that the PSNRs reported for images and videos are not directly comparable as the test scenes have no overlap, and readers should solely focus on the performance difference within each task category. The image compression method may be adopted in extreme cases where directly encoding each frame as an image outperforms video-based HiFiHC.

4.3.4 Discussion

In summary, these results demonstrate that HiFiHC can achieve superior image quality over the conventional image and video codecs on compressing complex holograms for diverse scenes of arbitrary resolution. Several challenges still need to be overcome for the practical deployment of such a learning-based system. Currently, HiFiHC has a considerable model size (2.2G) and runs at an interactive decoding speed (5 FPS for 1080P on an NVIDIA Tesla V100 GPU, without TensorRT optimization). Nevertheless, further performance engineering such as CNN model compression and TensorRT optimization can reduce the model size and improve runtime. With advances in low-power ASICS for accelerated CNN inference and the rapid development of memory-efficient CNN architectures, we envision proposed method to be available for the consumer holographic displays.

Chapter 5

Multi Color Holography

In recent years, the quest for high dynamic range (HDR) has revolutionized the way we view content on both TVs and monitors, offering a more lifelike and immersive experience. HDR TVs and monitors have the ability to display a wider range of colors, contrast, and brightness levels than traditional displays, resulting in more vivid and visually appealing images. Holographic displays, like other types of field sequential color displays, create full-color images by sequentially displaying single-color holograms, with each one dedicated to a specific wavelength of illumination (color channel). To display HDR content with this color driving scheme, one must increase the peak brightness of the illumination laser diode. However, doing so can give rise to various problems, such as the requirement for higher-grade (e.g., Class 3R: 1-5mw, Class 3B: 5-500mw) lasers that could pose a risk to eye safety, increased component cost (e.g., Thorlabs 635nm 5mw laser diode: \$27.13, Thorlabs 635nm 15mw laser diode: \$77.45), decreased power efficiency with undesired heat dissipation, and a greater likelihood of perceiving color break (also known as the "rainbow effect"). Without using more powerful laser diodes, this chapter explores an alternative approach for achieving HDR holography by optimizing **multi-color holograms** that can be illuminated simultaneously by multiple color primaries with co-optimized intensities. This method, termed HoloMultiColor, allows for higher utilization of each color primary for brighter projection by dedicating additional "on" time in every subframe to enhance the perceived brightness.

5.1 Motivation

Color perception is an essential aspect of human experience. It enables us to distinguish objects, interpret our surroundings, and influences our mood and emotions. Colors also convey information in many contexts, such as traffic lights, advertising, and branding, and can affect our behavior, including appetite, attention span, and purchasing decisions. For these reasons, accurate color reproduction is critical for display and projection systems since it directly impacts the quality of the delivered visual experience.

To display conventional 2D color images, three physical pixel pitches are required to represent three scalar values in a trichromatic color, as the images are produced in a "what you see is what you get" manner. In the case of LCD/OLED TVs and monitors, color array filters (CFAs) are used to create sub-pixel structures for color reproduction (e.g., often in the fashion of BGR, RGB, and RGBW). Projectors, on the other hand, use either 3 LCDs with R,G,B color filters or high-refresh DMD with spinning color wheels to produce color. In all cases, color reproduction necessitates 3x more pixels than grayscale reproduction, and is explicitly realized in either the spatial or the temporal domain.

In contrast, holograms differ significantly in how they create images. They function as a diffraction grating that directs light of different wavelengths in varying directions, but the grating's orientation and frequency vary across the hologram's plane to create the intended pattern in either the near or far field. Consequently, the appearance of each image pixel is a collaborative outcome of light deflected by numerous hologram pixels. This opens up the opportunity to engineer these locally-varying diffraction patterns to operate directly on polychromatic illumination, creating desired color responses at the image plane without dedicating a hologram pixel to a specific color primary. With improved pixel efficiency on color reproduction, we may co-optimize color driving schemes and hardware to achieve higher brightness or faster refresh rates.

Indeed, multicolor reflection and transmission holograms have been optically recorded for decades. In reflection, a thick hologram is highly selective in wavelength but insensitive to incident angles. Thus, we can jointly record three interference fringes inside the same recording material using aligned red, green, and blue lasers, and reconstruct the color

hologram with a single polychromatic light source as the hologram only reflects the wavelength it has been recorded with. In transmission, a thick hologram is angularly selective but wavelength tolerant. Thus, different angles shall be used for the three reference beams during recording, and reading beams shall match the reference beams in both angle and wavelength to yield position-aligned color images.

When a CGH is displayed on a spatial light modulator, it acts like a thin hologram with lower diffraction efficiency (i.e., multiple diffraction orders) and fewer modulation capacities compared to thick holograms, yet their operating principles are fundamentally identical. To date, however, no multicolor CGH algorithms have been developed nor accompanied hardware designs. This prevents utilizing the full modulation capacity of CGHs and hinders the potential improvement in image brightness and frame rate. In this work, we unlock this potential with HoloMultiColor, a CGH rendering framework that co-optimizes multicolor holograms with corresponding RGB laser intensities via a gradient descent-based optimizer with a combination of application-specific losses. By allowing polychromatic illumination of CGH, we explore two use cases of HoloMultiColor for projection systems: 1) enhancing image brightness through temporal multiplexing of three subframes as conventional field sequential color projection; 2) improving framerate using fewer than three subframes. We experimentally verify our findings using a holographic display prototype and show color-accurate results. The project codebase can be found at: <https://github.com/complight/holohdr>.

5.2 Method

To obtain multi-color holograms that simultaneously operate on three wavelengths of illumination, we depart from direct simulation methods (e.g., point-source or layered-based methods used in prior sections), and formulate the task as an iterative phase retrieval problem. Without loss of generality, we consider a normal incident plane wave as the illumination beam. To begin, optimizing conventional monochromatic holograms with field-

sequential color reproduction subjects to solving the following objective:

$$\hat{u}_p, \hat{l}_p \leftarrow \underset{u_p, l_p}{\operatorname{argmin}} \sum_{p=1}^3 \mathcal{L}(|l_p e^{iu_p} * h_p|^2, sI_p), \quad (5.1)$$

where p denotes the index of a color primary, u_p is the SLM phase (for the active primary, abbreviated thereafter), $\hat{u}_p$ is the optimized SLM phase, h_p is the wavelength-dependent light transport kernel for a chosen propagation distance of interest (i.e., the Fourier-space kernel can be found in Eq. 4.15 [184], $l_p \in [0.0, 1.0]$ is the laser amplitude, $\hat{l}_p$ is the optimized laser amplitude, I_p is the target image intensity, s is a global (across-primary) intensity scaling factor for creating a brighter or dimmer target (set by default to 1.0), $*$ denotes the convolution operation, and $\mathcal{L}$ denotes any valid loss function that measures the difference between the reconstruction and target.

To support multicolor holograms, we extend the modeling of subframes' responses to polychromatic illumination. Let T be the total subframes for reproducing one color image (i.e., 3 in the case of conventional holograms). Note that this is not to be confused with the repetition of subframes in time-multiplexing holography, which aims to reduce speckle noise [153, 48]. HoloMultiColor formulates the optimization problem as

$$\hat{u}_t, \hat{l}_{(p,t)} \leftarrow \underset{u_t, l_{(p,t)}}{\operatorname{argmin}} \sum_{p=1}^3 \underbrace{\left\| \left(\sum_{t=1}^T |l_{(p,t)} e^{i \frac{\lambda_p}{\lambda_{p_{\text{anchor}}}} u_t} * h_p|^2 \right) - sI_p \right\|_2}_{L_{\text{image}}}, \quad (5.2)$$

where p denotes the index of a color primary, $l_{(p,t)} \in [0, 1]$ represents the laser amplitude for the p -th primary at the t -th subframe, λ_p denotes the wavelength of the active primary, $\lambda_{p_{\text{anchor}}}$ represents the wavelength of the anchor primary, which is used to calibrate the nominal value of the SLM phase ($\lambda_{p_{\text{anchor}}}$ is set to 515 nm in our hardware prototype), u_t is the SLM phase for the t -th subframe, $\hat{l}_{(p,t)}$ is the optimized laser amplitude, $\hat{u}_t$ is the optimized SLM phase. In this formula, we use a ℓ_2 loss to measure the difference between the reconstruction and target. With $T = 3$, HoloMultiColor operates at the same refresh rate as conventional full-color holographic displays. Depending on the scene, a theoretical maximum intensity throughput of $s = 3$ is possible for every color primary since the human

visual system integrates subframes over time. When $T = 2$ or $T = 1$, HoloMultiColor can operate at a high refresh rate at the expense of color fidelity. To accelerate convergence and improve experimental results, we employ two additional losses in practice to ensure a robust HoloMultiColor hologram generation,

$$L_{\text{total}} = w_1 L_{\text{image}} + w_2 L_{\text{laser}} + w_3 L_{\text{variation}}. \quad (5.3)$$

where L_{image} equals the one defined in Eq. 5.2, w_1, w_2, w_3 are weights of each loss ($w_1 = 3.0, w_2 = 0.05, w_3 = 0.1$ in our implementation). The laser loss L_{laser} is given by

$$L_{\text{laser}} = \sum_{p=1}^3 \left(\left(\sum_{t=1}^T I_{(p,t)}^2 \right) - \max(I_p)s \right)^2. \quad (5.4)$$

For every color primary, L_{laser} encourages the sum of laser intensities across the subframes to match the scaled maximum intensity of the target image. It accelerates the convergence of L_{image} and consistently produces more accurate color in complex scenes (see Sec. 5.4 for an ablation study). The variation loss $L_{\text{variation}}$ is given by

$$L_{\text{variation}} = \sum_{t=1}^T \left(\left\| \nabla \left(u_t^{\text{mean}} + u_t^{\text{offset}} \right) \right\|_2^2 + \left\| \nabla \left(u_t^{\text{mean}} - u_t^{\text{offset}} \right) \right\|_2^2 + \sigma \left(u_t^{\text{mean}} + u_t^{\text{offset}} \right) + \sigma \left(u_t^{\text{mean}} - u_t^{\text{offset}} \right) \right), \quad (5.5)$$

where ∇ denotes the total variation operator, $\sigma(\cdot)$ denotes the standard deviation operator,

$$u_t(x, y) = \begin{cases} u_t^{\text{mean}}(x, y) + u_t^{\text{offset}}(x, y), & x + y \text{ is odd} \\ u_t^{\text{mean}}(x, y) - u_t^{\text{offset}}(x, y), & x + y \text{ is even} \end{cases}. \quad (5.6)$$

In our implementations, we use the total variation loss over an image pyramid of the reconstructed images. Here, we use a variant of the traditional double-phase formula to obtain the solution. Specifically, we add or subtract an offset phase u_t^{offset} from a mean phase u_t^{mean} to obtain a low phase and a high phase for double phase interlacing (Eq. 5.6). The variation loss discourages rapid change and large standard deviation for the low and high

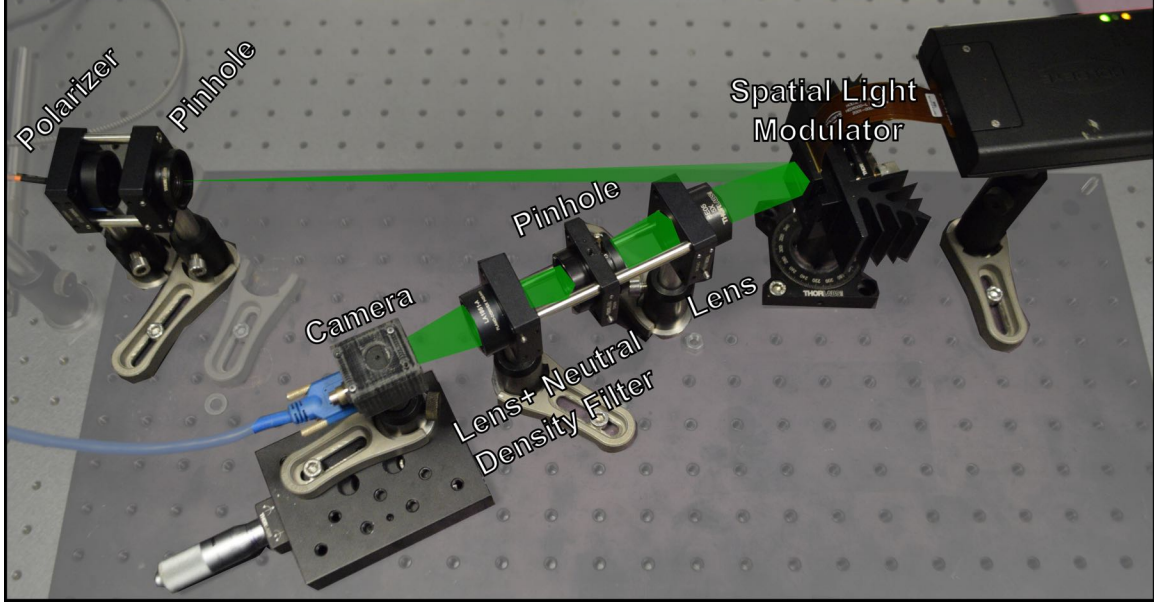


Figure 5-1: Holographic display prototype used in assessing HoloMultiColor holograms.

phase maps. It reduces the speckle artifacts commonly appearing in the experiments and accelerates the convergence of L_{image} . When manually setting s close to its theoretical limit (3 in case of $T=3$), a high-quality reproduction is not always guaranteed. Instead of finding the highest s through trials or always using a low s attainable for almost all scenes, we can jointly optimize s to be as high as possible under a user-specified image loss threshold ϵ_{image} ,

$$\hat{u}_t, \hat{l}_{(p,t)}, \hat{s} \leftarrow \underset{u_t, l_{(p,t)}, s}{\operatorname{argmin}} L_{\text{total}} - w_4 s, \text{ if } L_{\text{image}} < \epsilon_{\text{image}} \quad (5.7)$$

$$\hat{u}_t, \hat{l}_{(p,t)} \leftarrow \underset{u_t, l_{(p,t)}}{\operatorname{argmin}} L_{\text{total}}, \text{ if } L_{\text{image}} \geq \epsilon_{\text{image}}, \quad (5.8)$$

where w_4 is the weight hyperparameter for the scale. In Sec. 5.4, we show this conditional update strategy helps discover a content-dependent maximum scale.

5.3 Experimental Setup

We construct a benchtop holographic display prototype for experimental evaluation (see Fig. 5-1). Our prototype employs a fiber-coupled multi-wavelength laser light source, LA-

SOS MCS4, combining three laser light sources with wavelengths of 473nm, 515nm, and 639nm. Two ESP32 boards are used to control our polychromatic laser light source, with Digital-to-Analog Converters (DAC) available on the boards for precise power control. To limit the diverging beams from our fiber, we included a pinhole aperture (Thorlabs SM1D12) some distance from the fiber tip. After this pinhole aperture, we added a linear polarizer (Thorlabs LPVISE100-A) to align the polarization state with the fast axis of the phase-only Spatial Light Modulator (Holoeye Pluto-VIS) for the light beams. The phase-modulated beam is directed at a 4f imaging system consisting of two 50 mm focal length achromatic doublet lenses (Thorlabs AC254-050-A) and a pinhole aperture (Thorlabs SM1D12) to remove unmodulated and undiffracted light. In our experiments, we captured image reconstructions using a Ximea MC245CG-SY camera. We placed the camera on an X-stage (Thorlabs PT1/M travel range: 0-25 mm, precision: 0.01 mm) and moved it back and forth to take photographs from various depths.

5.4 Simulation & Experimental Assessment



Figure 5-2: $\times 1.63$ brightness optimized with dynamic scaling (100 ms exposure, source image: Midjourney).

We assess HoloMultiColor in terms of brightness, frame rate, image quality, and color production through simulation and experimentation.

Brightness Figure 5-3 shows photographs from our holographic display for conventional and HoloMultiColor schemes (more sample results can be found later). For such a scene in Fig. 5-3, HoloMultiColor can safely support up to $\times 1.8$ peak brightness without causing significant image distortions or artifacts. On the other hand, the conventional hologram fails to support peak brightness higher than

$\times 1.0$ as in Fig. 5-3 and leads to distortions. Beyond $\times 1.8$ peak brightness levels, images are typically heavily dominated by noise in the conventional case. In contrast, HoloMul-



Figure 5-3: Increasing peak brightness levels with HoloMultiColor. Photographs show that HoloMultiColor can enhance the peak brightness levels of the captures up to $\times 1.8$ without noticeable artifacts or distortions. In contrast, the conventional hologram fails to support beyond $\times 1.0$ (Source: MidJourney, Inc., 140 ms exposure).



Figure 5-4: Photographs showing HoloMultiColor generating higher brightness beyond $\times 2.0$ (source image: MidJourney, 50 ms exposure).

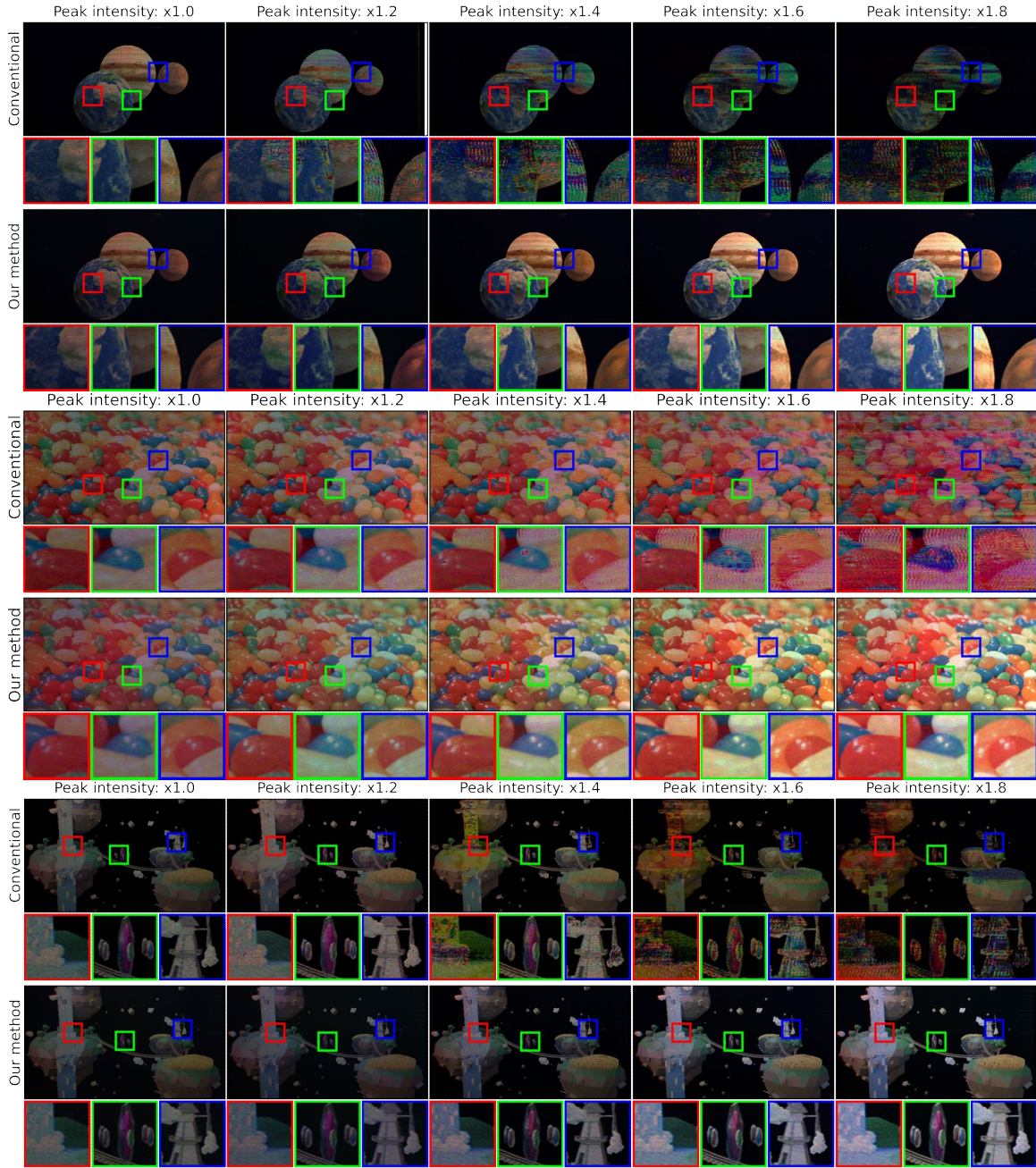


Figure 5-5: Increasing peak brightness levels with our multi-color holograms. All photographs are captured at a 140 ms exposure. Our multi-color holograms can enhance the peak brightness levels of the captures up to $\times 1.8$ without artifacts or distortions, whereas conventional holograms fail to support beyond $\times 1.0$ (140 ms exposure, source image: Mid-journey).

tiColor loses color integrity slightly or generates noises similar to the conventional case's $\times 1.2$ peak brightness case (see Fig. 5-4).

Supporting an artifact and distortion-free solution strictly at $\times 1.8$ peak brightness levels is not always guaranteed with HoloMultiColor, as each target scene's content heavily influences the results. Therefore, we also offer a dynamic scale option for HoloMultiColor as introduced in Sec. 5.2. Figure 5-2 shows a sample result from this dynamic intensity scale approach when enforcing the image loss to be below 0.01. In this sample result, the dynamic intensity scale for HoloMultiColor automatically chooses the intensity level of a targeted scene as $\times 1.63$ rather than hardcoding as any other value (e.g. $\times 1.8$). Thus, the dynamic intensity scale for HoloMultiColor offers a content-adaptive solution for choosing peak brightness levels.

Color production. Accurately reproducing colors for a scene can be complex since it also involves identifying the relationship between laser control and image intensity. As illustrated in previous figures, there is already a visible increase in intensity in multi-color holograms. We must, however, assess whether these results are faithful reproductions of the target scene's color. To improve color reproduction, we built a multi-layer perceptron (MLP) model to control the colors generated by our method. Specifically, this MLP with four hidden layers identifies the relationship between the laser powers suggested by the optimization, $\hat{l}_{(p,t)}$, and values provided to the laser driver. We evaluate the color reproduction of our results in Fig. 5-6 by comparing the color histogram of a target scene, the conventional hologram reconstruction, and the multi-color hologram reconstruction for each color primary. Our method's histogram approximates the target, whereas the conventional hologram fails to follow the trend beyond $\times 1.0$ peak intensity. We underline that our method does not aim for color gamut enhancement. In addition, for curious readers, we provide a theoretical analysis in the supplementary on addressable color gamut generated by conventional and multi-color holograms.

Number of Subframes and color primaries In our evaluations, we use three subframes, $T = 3$. However, as discussed in Sec. 5.2, our method could also use a lower number of

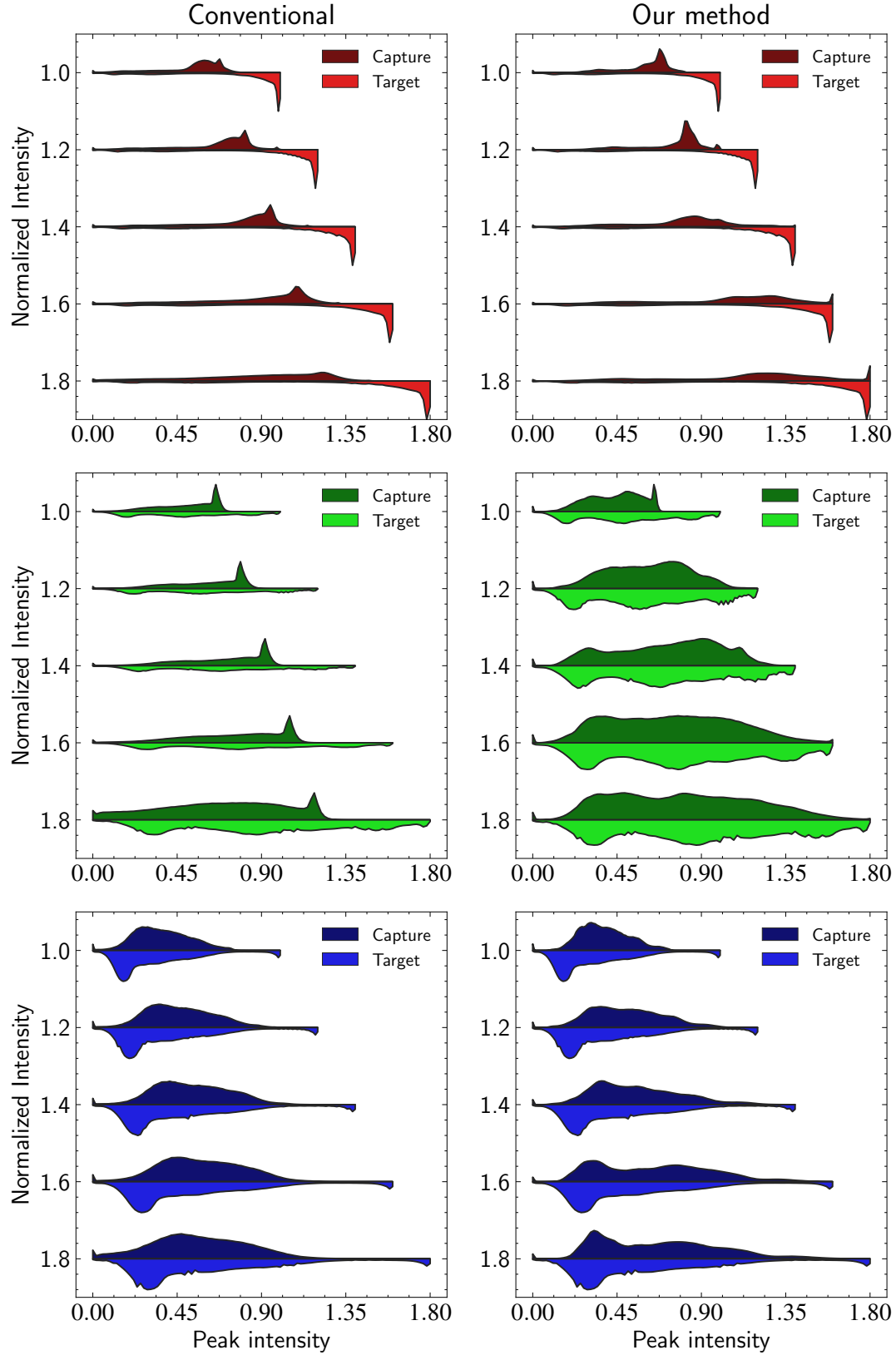


Figure 5-6: Comparing red, green, and blue histograms of a target image with conventional and multi-color schemes for varying intensity levels (Same target as Fig. 5-5, $\times 1 - \times 1.8$ intensity, 140 ms - 240 ms exposure).

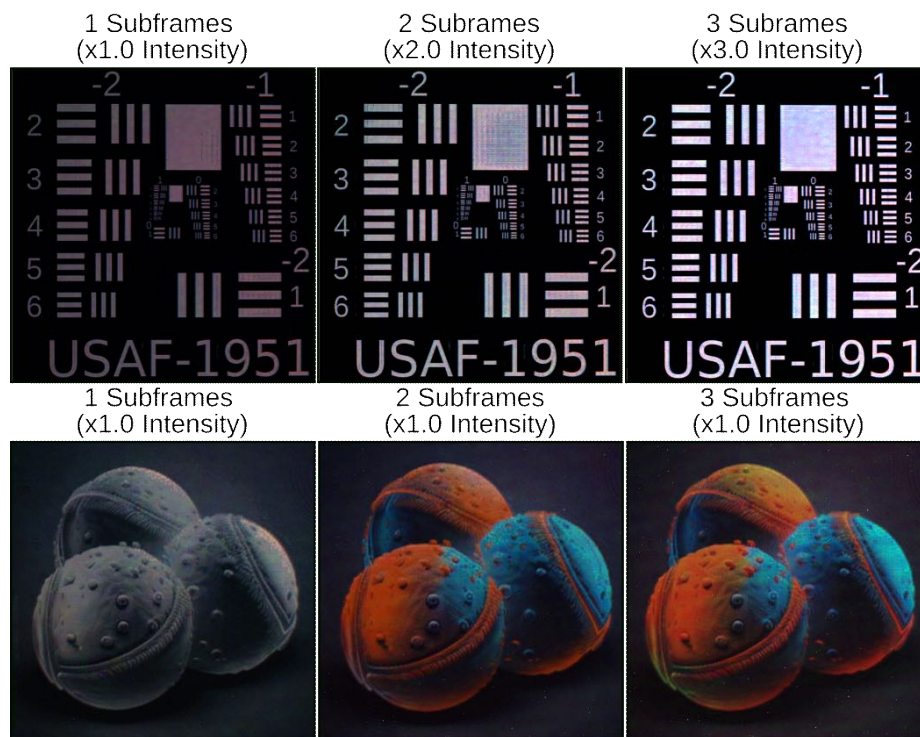


Figure 5-7: Using fewer subframes with HoloMultiColor. The first row shows photographs of HoloMultiColor results with a peak brightness increase for a grayscale content (50 ms exposure). The second row demonstrates the color reproduction quality increase for a full-color scene with the increasing number of subframes (200 ms exposure, source image: Midjourney.)

frames, $T \in \{1, 2\}$ (see Fig. 5-7). Fewer subframes can increase the refresh rate when monochrome and lower intensity target images are used (see Fig. 5-7 top row). Similar to the work by Huang et al. [122], using two subframes can also help display less colorful target images. In addition, inspired by conventional displays with multiple color primaries [135], there could be a variant of our method with more color primaries or spatially structured illumination [122], but holographic. In this way, our multi-color hologram optimization could benefit from identifying the right set of color primaries or spatial distribution of the illumination source [131].

Long Propagation Distances We report our results with images generated at the plane of SLM for conventional and multi-color schemes. When generating images away from an SLM, the behavior of color reproduction can change noticeably due to the complex point spread functions induced at various propagation distances and wavelengths. Figure 5-8 reveals such a case with simulated results generated at various distances from 0 cm to 15 cm for our hardware’s color primaries. An important observation from Fig. 5-8 is longer propagation distances may help with accurate color reproduction using fewer subframes, as each pixel’s color is now controlled by a larger subhologram, which endows more degree of design freedom. This freedom stems from the varying size of diffracted light spread with the changing wavelength and distances. At the extreme, $T = 1$, a long propagation distance of 15 cm could roughly match the color, promising the possibility of using our multi-color holograms to improve the frame rate. However, the frame rate reduction process could also largely depend on targeted color content.

Image Quality We compile Table 5.1 to provide an image quality comparison of Holo-MultiColor against the conventional scheme for increasing brightness levels. In our assessments, we use commonly accepted image quality metrics of PSNR, SSIM, and LPIPS [275]. Note that PSNR, SSIM, and LPIPS traditionally do not necessarily account for noise or distortions in the images. Thus, they are not an indicator of true image quality. We invite readers to observe the raw captures in result figures. The reported numbers suggest that HoloMultiColor consistently yields a higher quality reconstruction with an increasing

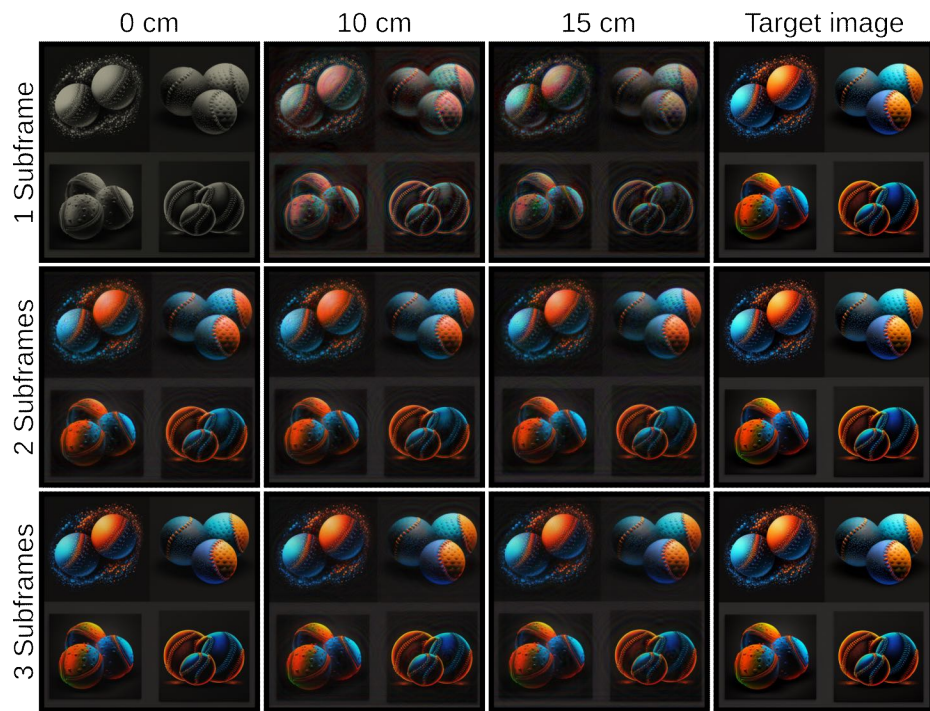


Figure 5-8: Given a target image (right) with $\times 1.0$ brightness and varying projection distances (from left to right), simulations of our multi-color holograms suggest an improvement in color reproduction capabilities when the projection distance prolongs, and may mean using fewer subframes to achieve the same image quality (source image: Midjourney).

Table 5.1: Image quality evaluation of conventional and HoloMultiColor schemes for various levels of peak brightness. Blue color indicates values for HoloMultiColor.

Scene	Metrics	Peak Brightness				
		$\times 1.0$	$\times 1.5$	$\times 2.0$	$\times 2.5$	$\times 3.0$
AR Glasses (Fig. 5-3)	PSNR (db)	30.3/29.9	23.8/24.8	16.2/22.4	12.2/18.0	9.7/15.1
	SSIM	0.9/0.9	0.9/0.9	0.6/0.8	0.4/0.3	0.2/0.7
	LPIPS	0.3/0.3	0.4/0.3	0.5/0.4	0.6/0.4	0.7/0.5
Fruit lady (Fig. 5-4)	PSNR (dB)	30.1/29.7	22.2/25.4	13.8/22.3	9.4/19.5	6.8/15.6
	SSIM	0.9/0.9	0.8/0.9	0.5/0.8	0.3/0.8	0.2/0.7
	LPIPS	0.4/0.4	0.5/0.4	0.6/0.4	0.7/0.5	0.7/0.5
Dog (Fig. 5-2)	PSNR (dB)	33.2/31.0	23.9/29.3	18.2/26.7	15.2/24.4	13.2/21.2
	SSIM	0.9/0.8	0.8/0.8	0.7/0.8	0.5/0.8	0.4/0.8
	LPIPS	0.3/0.3	0.4/0.3	0.5/0.4	0.5/0.4	0.6/0.4

brightness level, but a slightly inferior reconstruction at $1.0\times$ due to the contrast loss when optimizing holograms for multiple wavelengths (see Sec. 5.5 for detailed discussion).

Ablation Study We conduct an ablation study on our optimization model to identify the contribution of several components in our loss function and problem formulation. Note that we conduct our study using actual results from our display hardware, but not simulations, as simulation models do not account for hardware imperfections, leading to perfect results in simulation but not in actual display hardware. We provide the results from this study in Table 5.2, where we use the PSNR, SSIM, and LPIPS image quality metrics. In our ablation study, we remove one and only one component at each time. There are four studies, and these studies involve removing double phase constrain (Eq. 5.6), total variation loss (Eq. 5.5), laser loss (Eq. 5.4), and running the complete optimization pipeline without removing any components. We conduct this study by targeting $\times 1.8$ brightness values, using 1000 iteration steps and a 0.015 learning rate (Adam Optimizer [141]). Our study suggests that each component plays a crucial role in the success of HoloMultiColor optimization routine.

Three-dimensional Images To enable three-dimensional support, L_{image} has to be replaced with a loss term supporting multiplanes (we use work the loss from work by Kavakli et al. [136]). In addition, the optimization formulated in Eq. 5.2 shall be applied to each

Table 5.2: Ablation Study for HoloMultiColor. We remove only one component (not multiple) from the HoloMultiColor pipeline at each study and report image quality metrics. Without the “-” component refers to the complete model.

Scene	Without	PSNR (dB)	SSIM	LPIPS
AR	Phase Constrain	11.5	0.3	0.7
Glasses	TV Loss	13.7	0.6	0.6
(Fig. 5-3)	Laser Loss	19.0	0.8	0.4
	-	19.2	0.8	0.4
AR	Phase Constrain	12.3	0.4	0.6
Glasses	TV Loss	18.2	0.8	0.4
(Fig. 5-3)	Laser Loss	23.8	0.8	0.4
	-	26.3	0.6	0.4
AR	Phase Constrain	8.4	0.1	1.0
Glasses	TV Loss	12.4	0.4	0.7
(Fig. 5-3)	Laser Loss	18.9	0.8	0.5
	-	18.8	0.8	0.5

plane, and the losses must be accumulated. The results in Fig. 5-9 show that high-quality three-dimensional images are possible with our multi-color holograms.

5.5 Discussion

The proposed multi-color holograms holds the potential to improve realism in the next-generation holographic displays. However, it still suffers from several limitations, which we will discuss in the following as well as potential means to improve.

Contrast and Dynamic Range Despite that our multi-color holograms achieve a peak brightness increase, it does not increase and could slightly decrease the contrast (i.e. dynamic range). The reason is mainly two-fold. Firstly, a phase modulation is typically tuned to a specific wavelength. Thus, when operating with three colors, the SLM performs with a full phase modulation range for one color while having limited phase modulations for the other two. This loss of modulation accuracy leads to reduced diffraction efficiency and, consequently, lower contrast. Secondly, unlike conventional models focusing on achieving optimal response for a single color, each sub-frame in multi-color holograms needs to

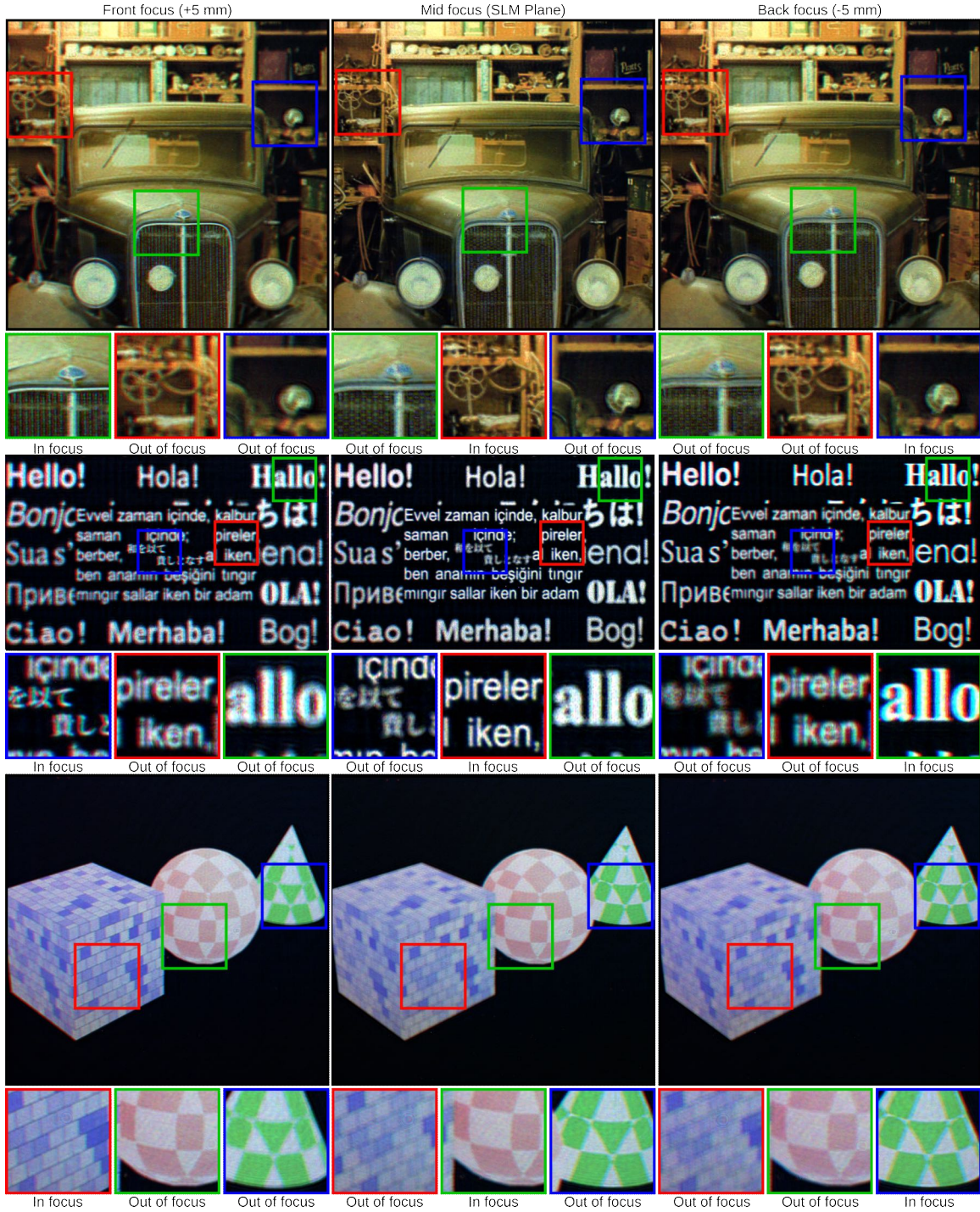


Figure 5-9: 3D scenes using our multi-color holograms. Each row shows a multiplane scene generated by our multi-color scheme with three focus planes. The targeted brightness level is $\times 1.8$ (140 ms exposure, source image: DIV2K [4], Midjourney)

balance and ensure that the intensities for all three color channels approximate the desired scaled target image. Thus, our method may choose intensities that could lead to slight deviations in color production. We measure the Michelson contrast, $\frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}$ to have a preliminary assessment of the situation. We measure for the highest and lowest brightness regions achieved in the top row example of Fig. 5-5. We report the Michelson contrast as 0.94 for our method versus 0.99 for conventional in $\times 1.0$ brightness. But we also observe a trend with the increasing brightness scale. For example, in $\times 1.8$ brightness case, the Michelson contrast is measured as 0.99 for multi-color and conventional cases. We speculate the loss of contrast issue may be mitigated by using loss functions dedicated to preserving contrast in the future.

Calculation Speed Convergence in our multi-color optimizations typically requires many steps (e.g. 1000) and a small learning rate (e.g. 0.015), leading to slow calculations (not interactive rate). Specifically, a three plane multi-color hologram takes about ten minutes of optimization time with thousands steps on a NVIDIA RTX 3090. However, a conventional hologram could calculate each subframe independently and concurrently with fewer steps (e.g. 60) and memory demand. Our multi-color optimizations could be formulated like a conventional hologram if required laser powers and targets for each subframe are known for a given content at the start of an optimization. Our current multi-color optimizations could help generate a dataset where holograms with their corresponding laser powers and targets are provided. Training a model with this dataset helps estimate the required laser powers at each subframe for a given target image before the optimizations.

Chapter 6

Ergonomic-Centric Holography

The preceding chapters primarily focused on enhancing the practicality of CGH by improving image quality, reducing computational cost, improving brightness, energy efficiency, and optimizing data transmission. However, an equally crucial design factor, ergonomics, has been largely ignored. Ergonomics involves achieving realism, comfort, and immersion, where enhancements in each of these areas shall be seamlessly integrated to create a genuine metaverse experience that bridges the gap between the virtual and real worlds, as well as ensuring comfort during an extended period of use. This chapter presents computational solutions for each of the aforementioned axes and carefully interweaves them to overcome otherwise difficult obstacles. In particular, we examine the achievement of speckle suppression and realistic defocus blur via temporal time-multiplexing and incoherent wave propagation simulation, the enhancement of comfort by modeling higher-order diffractions to minimize the display form factor, and the improvement of immersion through pupil optimization to expand the effective eyebox.

6.1 Motivation

This chapter is motivated by some of the key challenges unique to holographic displays in the context of building ergonomic-centric near-eye displays.

High-order Diffractions Grating structures cause monochromatic light to diffract and constructively interfere at locations where the optical path length difference is a multiple of the wavelength. This multiple is referred to as the order of diffraction. An SLM-generated CGH can be considered a general grating, with orders of diffractions manifesting as displaced replicas of the reconstructed 3D images. To only observe the principal replica (center copy), optical filtering has to be carried out in the Fourier space to remove the higher-order diffractions. This is typically accomplished using a 4-f system and an iris in the Fourier plane, however, the extra physical space required to accommodate the additional optics, optomechanics, and distance of propagation makes it difficult to design small form factor headsets. Gopakumar et al. [94] recently introduced high-order gradient descent (HOGD), an optimization algorithm for CGH that takes into account the effects of high-order diffractions. However, the successful demonstration of HOGD has only been presented for 2D content.

Realistic Incoherent Defocus Blur & Speckle Noise Tradeoff Our daily environment is illuminated by incoherent light sources, which emit waves that are not in phase with each other and that do not oscillate at the same frequency. When imaging a natural 3D scene with a lens (e.g., the cornea of an eye), the pixel readout of a sensor is the *accumulated intensity* (i.e., *linear in intensity*) of overlapping rays incidents from different angles. Without the wave nature of light manifesting itself, out-of-focus regions recorded in such images appear near uniformly blurred within the circle of confusion and free from ringings and wiggles. In contrast, holographic displays necessitate coherent light to create 3D images, and when imaged, the pixel readout is the *intensity of the accumulated complex amplitude* (i.e., *linear in complex amplitude*). Consequently, if CGH is calculated without associating random phases to target objects or optimized without regularizing defocus responses, the out-of-focus regions will present fringes and wiggles due to structured interference arising from spatial coherence, and the extent of the blur will be comparably smaller as interference distributes energy non-uniformly in the space. In such cases, the discrepancy in defocus patterns between the hologram and the real world destroys the sense of the relationship between depth and blur, which is crucial in the context of depth perception and perceived

realism.

This mismatch can be mitigated by associating random phases to target objects in diffraction simulation or forcing the defocus response to match the blur under incoherent illumination. The CGH calculated in this manner breaks the spatial coherence of the incident illumination, causing random constructive and destructive interference between adjacent hologram pixels at the image plane. The random interference produces a uniform-like defocus response, yet the persistence of temporal coherence introduces speckle noise, a granular noise texture superimposed on the produced 3D image. As a result, there is a direct trade-off between the realism of the defocus response and the level of speckle noise. We note that speckle noise can be reduced by using light sources with low temporal coherence, such as superluminescent light-emitting diodes (sLEDs), but this comes with an alternative cost of reduced image sharpness and a shortened depth range [203]. To reduce speckle noise without compromising image quality, temporal multiplexing has been shown to be highly effective by averaging out the speckle patterns over the persistence of the eye perception [48]. However, little research has been conducted in conjunction with the modeling of higher-order diffractions and the explicit supervision of incoherent defocus response.

Field of View & Eyebbox Tradeoff For all displays, including holographic displays, the product of the field of view and the eyebbox is known as the space-bandwidth product or etendue. This value is a constant for a given display system and defines the maximum amount of light or information it can potentially carry with. Therefore, display designers must balance the FOV and eyebbox to optimize the user experience for a given hardware.

For holographic displays, this task can be more complex since the effective etendue depends not only on the SLM but also on the characteristics of the CGH. For example, initializing the object point phase with the position-dependent formula introduced in Chapter 4 preserves spatial coherence and results in smooth phase holograms. Such holograms are in favor of image quality by computationally limiting the frequency of the phase image to reduce random speckles. However, this also leads to the underutilization of etendue since only the low-frequency components of the spectrum are used to carry the information, lead-

6.2 Method

6.2.1 Simulation of Incoherent Depth of Field Targets for 3D CGH

EC-H begins by rendering a focal stack that matches real-world defocus response using incoherent wave propagation. Consider a 3D scene with a thickness of V and N evenly spaced (for convenience) recording planes (32 in our case), the space-domain incoherent wave propagation kernel h_i for propagating a scene point at depth d to the n -th recording plane is given by:

$$h_i(x, y) = F\{\mathcal{H}_c\}F^*\{\mathcal{H}_c\}M_{h_i} \quad (6.1)$$

$$\mathcal{H}_c(f_x, f_y) = \begin{cases} e^{i\frac{2\pi}{\lambda}\sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}(d-\frac{nV}{N})}, & \text{if } \sqrt{f_x^2 + f_y^2} < \frac{1}{\lambda}, \\ 0 & \text{otherwise} \end{cases}, \quad (6.2)$$

where λ is the wavelength, $\mathcal{H}_c$ is the frequency-domain band-limited coherent propagation kernel, and M_{h_i} is an optional binary mask for space-domain filtering (e.g., conforming the kernel to produce a circular-shaped out-of-focus response, or forcing a deep depth of field). To efficiently and completely model a 3D scene, we use LDI, an advanced multi-layer RGB-depth image representation, to record both foreground and background points. For each point, we perform ray tracing with occlusion processing to integrate its incoherent sub-hologram kernel (Eq. 6.1) at each recording plane and render the target focal stack with unquantized per-pixel depth defocus. We show superior image quality over the simple blending and masking method proposed by Lee et al. [153] in Fig. 6-1. In practice, we set M_{h_i} as a circular binary mask to induce a more common circular blur spot.

6.2.2 Optimization of Time-multiplexed 3D CGH with High-order Diffractions

EC-H enhances the HOGD algorithm with temporal multiplexing and dynamic pupil modeling. Denote the pixel pitch as Δp , the number of orders to model as α , the total number of frames to time multiplex as T , the total number of circularly-shaped pupils to si-

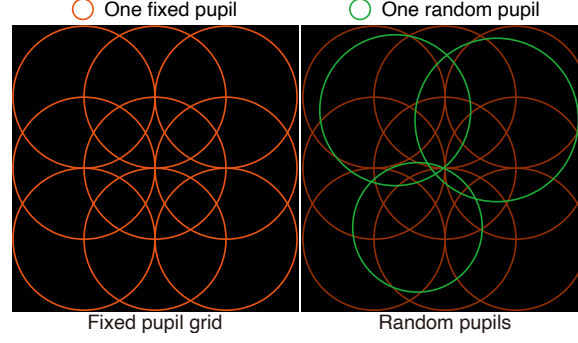


Figure 6-2: Visualization of fixed pupils and random pupils used in pupil optimization. The fixed pupils are underlaid in faint orange in the visualization of random pupils.

multaneously optimize as P , the radius of a pupil as r , the support of the eye box as $\{x \in \mathbb{R} : x_{\min} + r \geq x \leq x_{\max} - r; y \in \mathbb{R} : y_{\min} + r \geq y \leq y_{\max} - r\}$, where (x, y) is the center of the pupil, and $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ are the min and max limit along x and y-axis that defines the boundary of the eye box. Throughout optimization, we maintain a set of P_F fixed pupils that form a uniform pupil sampling grid over the eye box, forcing the energy frequency to be structurally distributed across the whole eye box. At each iteration, we also generate $P_R = P - P_F$ random pupils to account for pupil variations within the lattice of the fixed pupils (see Fig. 6-2 for a visualization). For the t -th SLM pattern ϕ_t and p -th pupil mask M_p , the camera sensor captured field $\hat{u}_p(\phi; z)$ (at the same resolution of the SLM) at a distance of z is given by:

$$\begin{aligned}
 \hat{u}_p(\phi; z)[x, y] &= \sqrt{\sum_{i=1}^{\alpha} \sum_{j=1}^{\alpha} \frac{1}{\alpha^2} |u_p(\phi; z)[\alpha x + i, \alpha y + j]|^2} \quad (\text{downsampling to SLM resolution}) \\
 u_p(\phi_t; z) &= \iint U(f_x, f_y; \phi_t) A_p(f_x, f_y; z) e^{i2\pi(f_x x + f_y y)} df_x df_y, \quad (\text{propagated hologram with high orders}) \\
 U(f_x, f_y; \phi_t) &= \sum_{j, k \in \alpha} F \left\{ e^{i\phi_t} \right\} \left(f_x + \frac{j}{\Delta p}, f_y + \frac{k}{\Delta p} \right) \quad (\text{hologram spectrum with high orders}), \\
 A_p(f_x, f_y; z) &= \mathcal{H}_c(f_x, f_y; z) \text{Sinc}(\pi f_x \Delta p) \text{Sinc}(\pi f_y \Delta p) M_p, \quad (\text{high-order propagation kernel}) \\
 M_p(f_x, f_y; c_{p_x}, c_{p_y}, r_p) &= \begin{cases} 1, & \text{if } (f_x - c_{p_x})^2 + (f_y - c_{p_y})^2 < r_p^2 \\ 0 & \text{otherwise} \end{cases}, \quad (\text{pupil mask})
 \end{aligned} \tag{6.3}$$

Given a target incoherent focal stack $\{a_n | n = 1, \dots, N\}$, we use gradient descent to optimize the batch of time-multiplexed holograms with objective

$$\underset{\{\phi_t | t=1, \dots, T\}}{\operatorname{argmin}} \sum_{n=1}^N \sum_{p=1}^P \left\| \frac{\sqrt{\frac{1}{T} \sum_{t=1}^T |\hat{u}_p(s_g s_l \phi_t; z_n)|^2}}{s_p} - a_n \right\|, \quad (6.4)$$

where s_p is an optional non-optimizable normalization scale that accounts for the pupil size variation, s_l is a non-optimizable per-pixel scale that compensates the non-uniformity of the incident illumination [224, 204], and s_g is an optimizable global scale to match the total field intensity with the targets. We initialize it as

$$s_g^{i=0} = \sqrt{\frac{\frac{1}{N} \sum_{n=1}^N \overline{a_n^2}}{\frac{1}{N} \sum_{n=1}^N |\overline{u(\phi; z_n)}|^2}}, \quad (6.5)$$

where $\bar{\cdot}$ is the mean operator.

During optimization, we consider the central 3×3 orders ($\alpha = 3$). Orders higher than 3 are omitted as they contribute negligibly. For each scene, we optimize for 6 focal planes, typically chosen to have objects of interest in focus. For unfiltered results and pupil-mimicking results, we use 5 and 3 sub-frames for time-multiplexing, respectively. Each phase sub-frame is initialized randomly over 2π . The pupil-mimicking results use a fewer number of sub-frames to balance the increased cost in computation and time for modeling pupils. For all experiments, the propagation distance from the SLM to the center of the 3D volume is set to 60 mm and the thickness of the 3D volume is 6 mm. In Sec. 6.4, we evaluate how propagation distance influences the performance of the proposed algorithm.

6.2.3 Software Implementation

We implement ray-tracing for incoherent focal stack rendering in the Taichi programming language (version 1.4), which offers Python-style syntax for parallel computing. For a full HD resolution LDI with 5 layers, it takes 2.2 seconds to render one depth of field image on an NVIDIA RTX A6000 GPU. EC-H optimization is implemented in PyTorch 1.13. For CGH optimization, we use a propagation distance of 60mm for both unfiltered and

pupil-mimicking results with a 6mm wide 3D volume. To handle the large memory cost of optimizing at $3\times$ spatial resolution (5760×3240 px) for multiple holograms at multiple focal planes and pupils (e.g., naive implementation takes 100+GB memory for 3 holograms at 6 depths planes and 9 pupils), we implement gradient accumulation [1] to split optimization into mini-batches such that the forward/backward evaluation of each batch fits in the GPU memory. For optimizing RGB hologram with 6 depth planes, one iteration takes ~ 1.5 seconds on an NVIDIA A6000 GPU. For all presented results, we run 8000 iterations for unfiltered results and 2000 iterations for pupil-optimization results, after which the losses typically plateau.

For experimental results, we captured R, G, and B sub-frame holograms separately. The intensity scaling is achieved optically by adjusting laser power according to the optimized global scale. We use an aperture size of f2.8 and a shutter speed of 1/100 of a second for all captures. For all optimizations, we use sRGB images of the focal stack as the target amplitudes.

Because pupil size variation causes a difference in the amount of energy inlet in the Fourier domain, we add an optional amplitude normalization factor to account for the potential large variations in the resulting image. Specifically, we let the base pupil radius as the reference pupil radius r_{ref} , for which we assume the reconstruction amplitude should absolutely match the target amplitude. We then normalize the amplitude obtained from other pupil sizes by $s_p = r_p/r_{\text{ref}}$, assuming the image intensity is ideally proportional to the area of the pupil (amplitude proportional to the radius). In practice, this ideal scale is impossible to achieve since it's impossible to distribute the energy evenly in the Fourier domain; therefore, its usage is rather optional, but we find it helps stabilize the convergence.

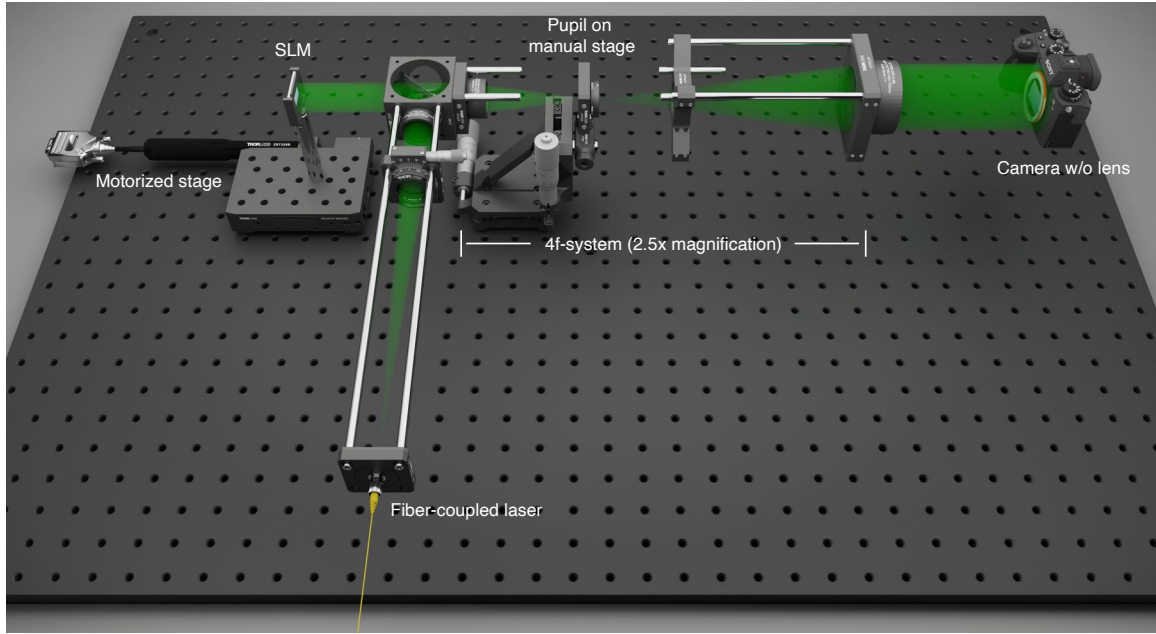


Figure 6-3: Schematic rendering of the experimental setup. Only green laser is visualized. The control boxes of the SLM and laser are omitted.

6.3 Experiment & Assessment

6.3.1 Experimental Setup

Our experimental setup uses a Holoeye Pluto SLM with a resolution of $1,080 \times 1,920$ and 8-bit phase control across visible wavelengths (see Fig. 6-3 for a schematic rendering). The SLM is mounted on a motorized translation stage (Thorlabs Z825B) to programmably shift position for focus control. Coherent illumination is provided by a FISBA RGBeam fiber-coupled laser with central wavelengths at 632 nm (red), 520 nm (green), and 450 nm (blue). The laser light is collimated by a lens, linearly polarized, and directed to the SLM by a beamsplitter. A 4f system with lenses of 80 mm (first) and 200 mm (second) is used to relay and magnify the image to fulfill a full-frame camera sensor (Sony A7III). An optional iris (Thorlabs ID12), mounted on a manual xz-stage (Thorlabs XRN25P/M, XRN-XZ/M), is placed at the Fourier plane to mimic an eye pupil. When the first lens of the 4f system acts as an eyepiece, the central order of the red light diffraction creates an eye box of approximately $6.4\text{mm} \times 6.4\text{mm}$ at the Fourier plane. In unfiltered mode, the iris is

¹https://pytorch-lightning.readthedocs.io/en/stable/advanced/training_tricks.html

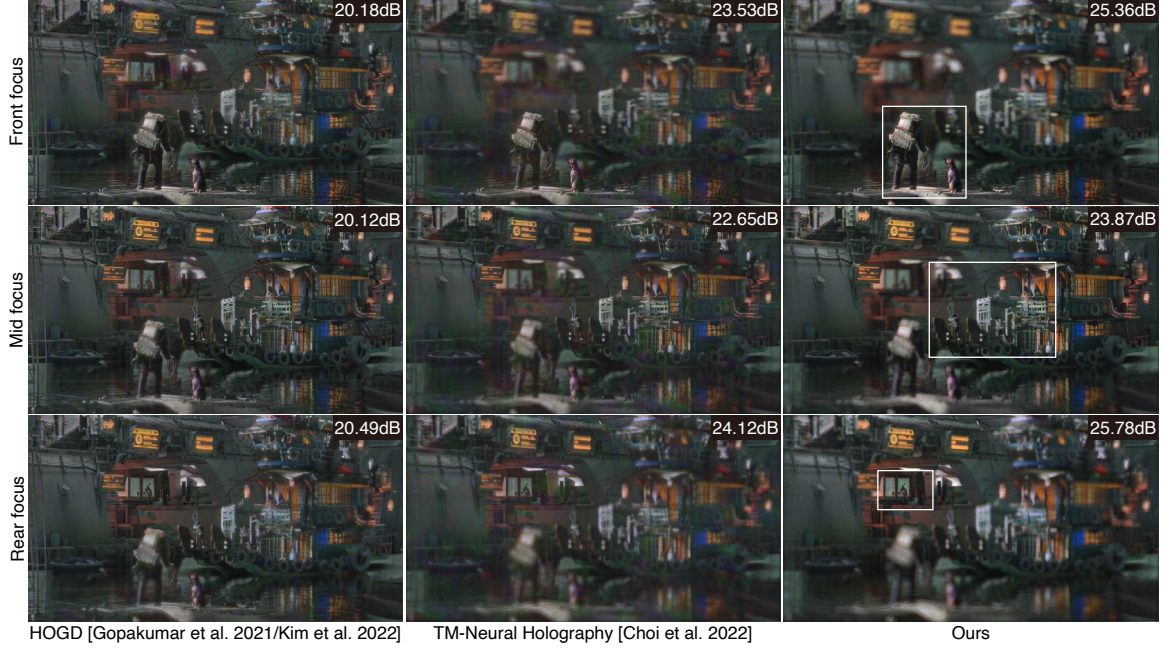


Figure 6-4: Comparison of 3D CGH algorithms on experimental results captured under unfiltered mode. The camera focuses are marked by white rectangles (plane 4, 16, 28), and the numbers indicate the PSNR with respect to the target image. Source image: “PartyTug 6:00AM”, reproduced under terms of the CC-BY-SA license, Copyright 2020, by Ian Hubert, published with the author’s permission.

absent. In pupil-mimicking mode, the iris is inserted and positioned at different locations. Miniaturization of such a setup (the parts before the pupil on the manual stage) has been demonstrated by Kim et al. [140] in a compact eyeglass form factor.

6.3.2 Unfiltered Mode Evaluation

Figure 6-4 compares experimentally captured holograms using EC-H, time-multiplexed neural holography (TM-NH) [48], and HOGD [94, 140] in the unfiltered mode. We use our LDI-computed focal stacks to supervise the optimization of TM-NH, as the code to generate their focal stack is not yet publicly available. EC-H outperforms TM-NH by effectively reducing replicas and rainbow-like artifacts caused by wavelength-dispersed high-order diffractions. This leads to tangibly improved image contrast while preserving the depth-dependent incoherent defocus throughout the 3D volume. Unlike TM-NH, HOGD does not suffer from high-order diffractions. However, it produces coherent defocus responses due to a lack of supervision for out-of-focus regions. For all methods, time multiplexing

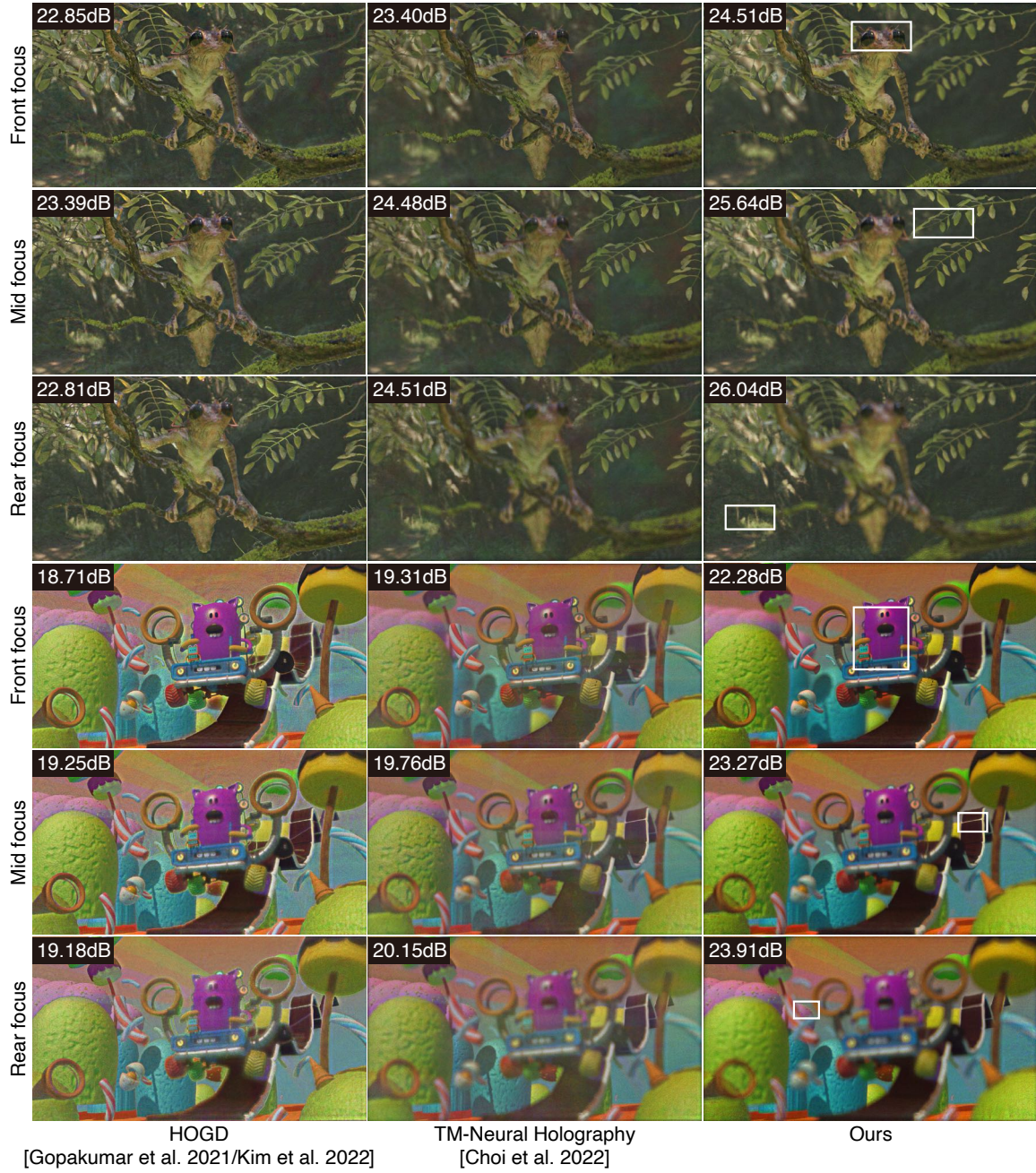


Figure 6-5: Additional experimental results captured in the unfiltered mode. The captured results are presented in 900 DPI resolution for close-up examination. The “Tree Creature” shows recording planes 9, 17, 28, the “Racing Car” shows recording planes 5, 15, 27.

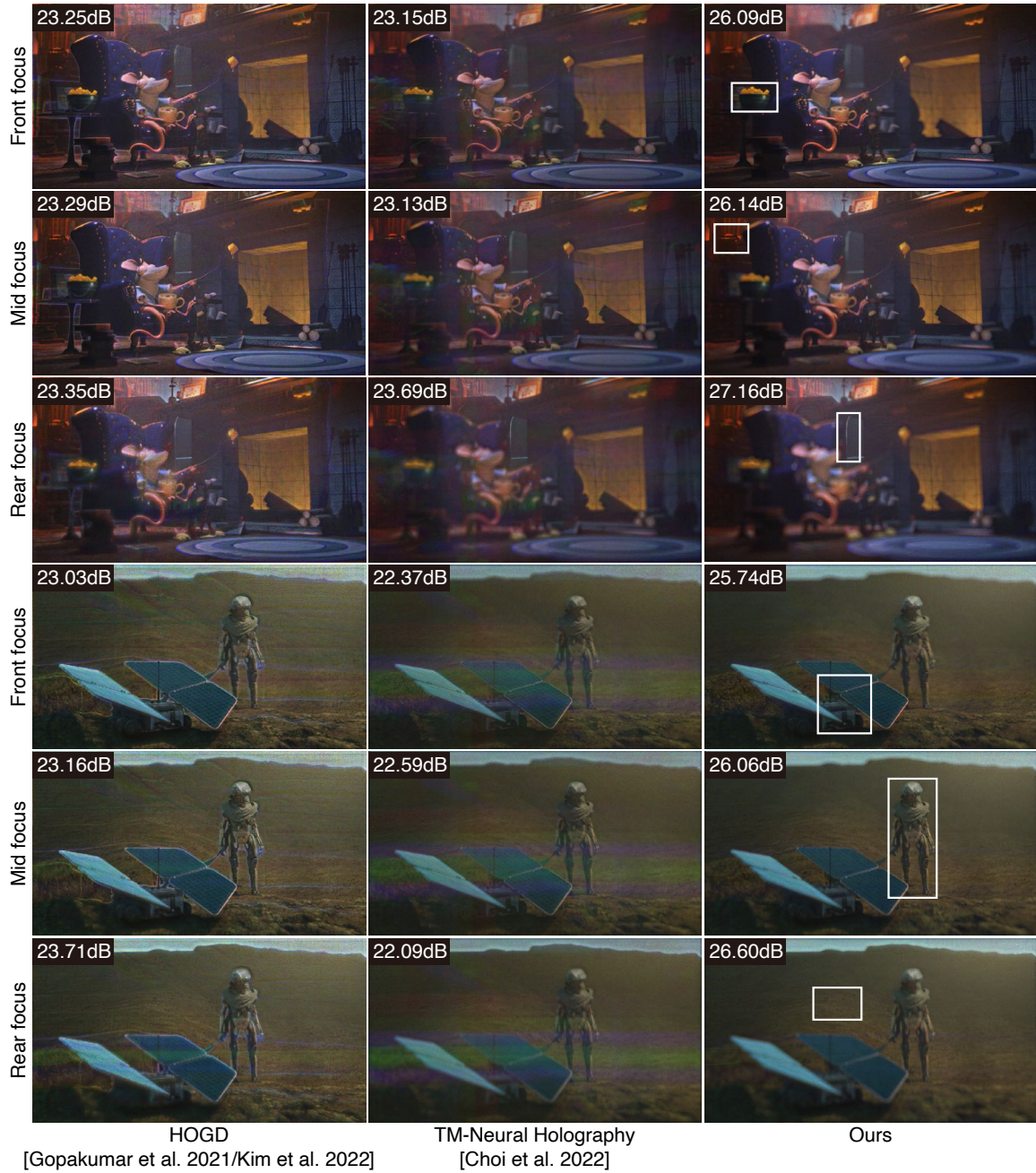


Figure 6-6: Additional experimental results captured in the unfiltered mode. The captured results are presented in 900 DPI resolution for close-up examination. Source image (bottom): “Wanderer”, reproduced under terms of the CC-BY-SA license, Copyright 2018, by Daniel Bystedt, published with the author’s permission. The “Mr. Elephant” shows recording planes 4, 15, 29, the “Wanderer” shows recording planes 4, 10, 15.



Figure 6-7: Comparison of 3D CGH algorithms on experimental results captured at various pupil locations and sizes. The bottom right corners in column 1 images mark the pupil position and size used to capture the results in the respective row. The numbers indicate the PSNR with respect to the target image. Source image: “Mr. Elephant”, reproduced under terms of the CC-BY-ND license, Copyright 2017, by Glenn Melenhorst, published with the author’s permission.

effectively reduces the speckle noise. Results of four additional examples can be found in Figs. 6-5 and 6-6.

6.3.3 Pupil-Mimicking Mode Evaluation

To optimize for image quality across the eye box, EC-H consider an 8×8 mm eye box defined by $(x_{\min}, y_{\min}, x_{\max}, y_{\max}) = (-4, -4, 4, 4)$ mm, a size bigger than the theoretical maximum as we show modeling high-order diffractions effectively extends the eye box formed by the central diffraction order. We use $P = 25, P_F = 9$ (a 3×3 grid), $P_R = 16$. We set $r = 2$ mm as the base pupil size to form the uniform sampling grid for the fixed



Figure 6-8: Additional experimental results captured in the pupil-mimicking mode. The captured results are presented in 900 DPI resolution for close-up examination. Source image (top): “The Junk Shop”, reproduced under terms of the CC-BY license, Copyright 2019, by Alex Treviño/ original concept by Anaïs Maamar.



Figure 6-9: Additional experimental results captured in the pupil-mimicking mode. The captured results are presented in 900 DPI resolution for close-up examination.

pupils. For the random pupils, their locations are randomly selected within the eye box, and their sizes are scaled between half and twice the base size to account for pupil variations within the lattice of the fixed pupil grid. We note that because different wavelengths of light create different eye box sizes, a fixed physical pupil size induces different pixelated aperture openings at the Fourier plane. Figure 6-7 compares experimentally captured holograms obtained from EC-H, pupil-aware holography (PW-H) [37], and Pupil-HOGD [140]. Note that the original paper of PW-H optimizes coherent defocus for two-plane results. As Pupil-HOGD covers reproducing coherent defocus, we augment PW-H to reproduce incoherent defocus to emphasize other improvements made by EC-H.

At eccentric pupils, the Pupil-HOGD method suffers from a significant loss of intensity when the pupil is shifted and reduced to the extent when it fails to fully encompass the DC term. This is evident in the transition from a 6mm pupil in row 2 to a 4mm pupil in row 3, both shifted to the center top of the eye box. Alternatively, PA-H exhibits pronounced rainbow-like artifacts with reduced contrast, a faster decay in image brightness (see row 3), and a stronger reduction in the extent of reproduced defocus blur compared to EC-H (see the orange box in column 1, row 3 versus the green box above, with the same regions in column 3). They are caused by PA-H’s vulnerability to high-order diffractions and the absence of fixed pupils during optimization, which further push the energy spectrum structurally to the mid/high frequencies (see row 1 bottom right for phase-only holograms). EC-H better maintains the image intensity and quality as the pupil moves away and reduces. It also produces artifact-free images outside the central-order-created eye box. The extended eye box allows the perception of noticeable motion parallax which is critical for depth perception without significant artifacts or intensity loss (see Sec. 6.3.7 for detailed discussion). Results of four additional examples can be found in Figs. 6-8 and 6-9.

6.3.4 Ablation Study of Pupil Optimization Scheme

We conduct an ablation study by comparing 25 random pupils and 25 fixed pupils in Fig. 6-10 to more systematically analyze the benefit of using mixed pupils. It is evident that the utilization of fully randomized pupils, coupled with high-diffraction modeling, sig-

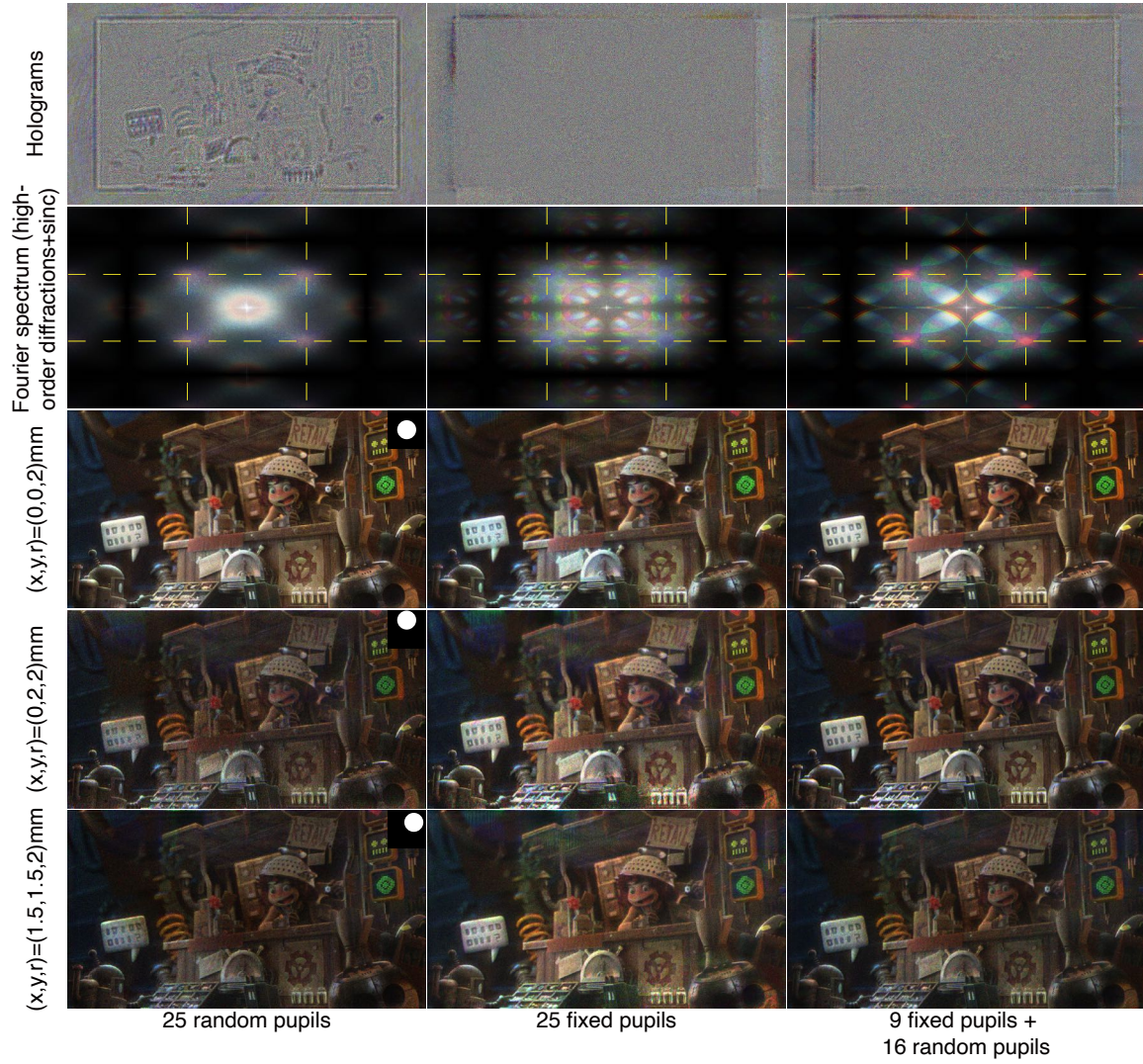


Figure 6-10: Ablation study of pupil-mimicking optimization. Here, we compare mixed pupil optimization with using only random pupils or fixed pupils at the same number of total pupils.

nificantly mitigates the rainbow-like artifacts commonly seen in pupil-aware holography. Nonetheless, image intensity at peripheral pupils still suffers from a faster decay of intensity as well as a non-uniform decay, particularly in this example. On the other hand, when 25 fixed pupils are employed, the image intensity is more consistently maintained across the pupils, as more energies are pushed to the mid and high frequencies (i.e., center low-frequency “+” reduces from fully random to mixed to fully fixed.). However, this approach generates more noticeable rainbow-like artifacts, serving as a trade-off. Ultimately, determining the optimal configuration for pupils remains an open question and is user-dependent based on their eye sensitivity to different artifacts. Thus, we advocate for more extensive and rigorous exploration in constructing optimal pupil batches for future research.

As shown in Fig. 6-15 EC-H holograms optimized for pupil-mimicking mode possess similar statistics across different examples and differ visually from ones optimized by prior state-of-the-art methods. In Fig. 6-10, we compare the Fourier spectrum of holograms optimized for Fig. 6-8 top example. For pupil-HOGD (column 1), a significantly greater portion of the energy is concentrated in the low-frequency band near the DC, forming an extended bright “+” shape. By modeling a dynamic pupil (column 2), the concentrated energy is noticeably disseminated toward the mid-frequency, forming a bright oval around the center. Pupil-shaped patterns can also be vaguely seen, which match the spectrum visualization in the [37]. However, because of the absence of high-order diffraction modeling, bright spots and contour lines are formed at the intersection and borders between different orders. They contribute to the rainbow-like artifacts observed in the reconstructed images and prevent the extended use of the eye box. When changing to the proposed mixed pupil optimization with high-order diffraction modeling, the bright spots and contour lines disappear, and the energy in the low-frequency “+” further reduces. The extra energy, instead of uniformly disseminated to the mid-frequency, is more structurally distributed to both the mid and high-frequency regions, preventing significant intensity drop at the periphery of the eye box.



Figure 6-11: Comparison of simulation and experimental results obtained using coherent integration and incoherent integration.

6.3.5 Evaluation of Downsampling Operation in High-Order Diffraction Propagation

When numerically implementing Eq.(3), the Fourier domain replication $U(f_x, f_y; \phi_t)$ changes the output field resolution from $[H, W]$ to $[\alpha H, \alpha W]$, where H and W are the width and height (spatial resolution) of the input field. However, the spatial resolution of amplitude images in the target focal stack a_n in Eq.(4) remains at $[H, W]$, demanding a dimension reduction in Eq.(3) (i.e., the equation in its first line) to $[H, W]$ to evaluate the objective function (loss). According to Fourier optics, increasing the Fourier domain resolution doesn't change the support of the field in the space domain, but increases the spatial resolution within the support; thus, the dimension reduction is implemented as downsampling instead of cropping.

In practice, the downsampling is implemented as a strided convolution operation with a stride α and a uniform kernel of size $[\alpha, \alpha]$ and value $\frac{1}{\alpha^2}$ (a normalization factor that eases s_g initialization). Physically, this mimics the process of virtually placing an imaging sensor with resolution $[H, W]$ at the output plane and integrating the electric field stimulated by

the photons arriving at each sensor pixel. This integration shall be implemented over the intensity (incoherently) instead of the complex amplitude (coherently) of the output field since every group of $[\alpha, \alpha]$ pixels arrive at different areas of their corresponding virtual sensor pixel and trigger photoelectric effect independently. Mathematically, denote the complex amplitude of the digitized output field $\mathbf{u}(\phi; z) \in \mathbb{C}^{\alpha H \times \alpha W}$, the amplitude of the virtual sensor captured field $\hat{\mathbf{u}}(\phi; z) \in \mathbb{R}^{H \times W}$, their relation is given by

$$\hat{\mathbf{u}}(\phi; z)[x, y] = \sqrt{\sum_{i=1}^{\alpha} \sum_{j=1}^{\alpha} \frac{1}{\alpha^2} |\mathbf{u}(\phi; z)[\alpha x + i, \alpha y + j]|^2} \quad (\text{incoherently, correct}) \quad (6.6)$$

instead of

$$\hat{\mathbf{u}}(\phi; z)[x, y] = \sum_{i=1}^{\alpha} \sum_{j=1}^{\alpha} \frac{1}{\alpha^2} \mathbf{u}(\phi; z)[\alpha x + i, \alpha y + j] \quad (\text{coherently, wrong}) \quad (6.7)$$

Figure 6-11 displays a significant difference in results derived from Eq.(6.6) and Eq.(6.7) for the "PartyTug 6:00AM" scene. The term "coherent integration" here refers to scenarios where the hologram is optimized using coherent integration, while the simulated reconstruction employs incoherent integration to accurately and physically mirror experimental behavior. Due to the mismatch between the simulation model and the physical process, the results of coherent integration exhibit pronounced high-order artifacts and diminished contrast. For both results, when the simulation utilizes incoherent integration, a precise correspondence between the simulation and experiment is observed, which validates the accuracy of Eq. 6.6.

6.3.6 Evaluation of Number of Time-multiplexed Frames for Speckle Reduction and Incoherent Defocus Reproduction

Figure 6-12 demonstrates the influence of the number of time-multiplexed frames on speckle reduction and incoherent defocus reproduction. We refer readers to Lee et al. [153] for a detailed speckle analysis. In brief, the speckle contrast is inversely proportional to the square root of the number of time-multiplexed frames. In this example, the gain visually

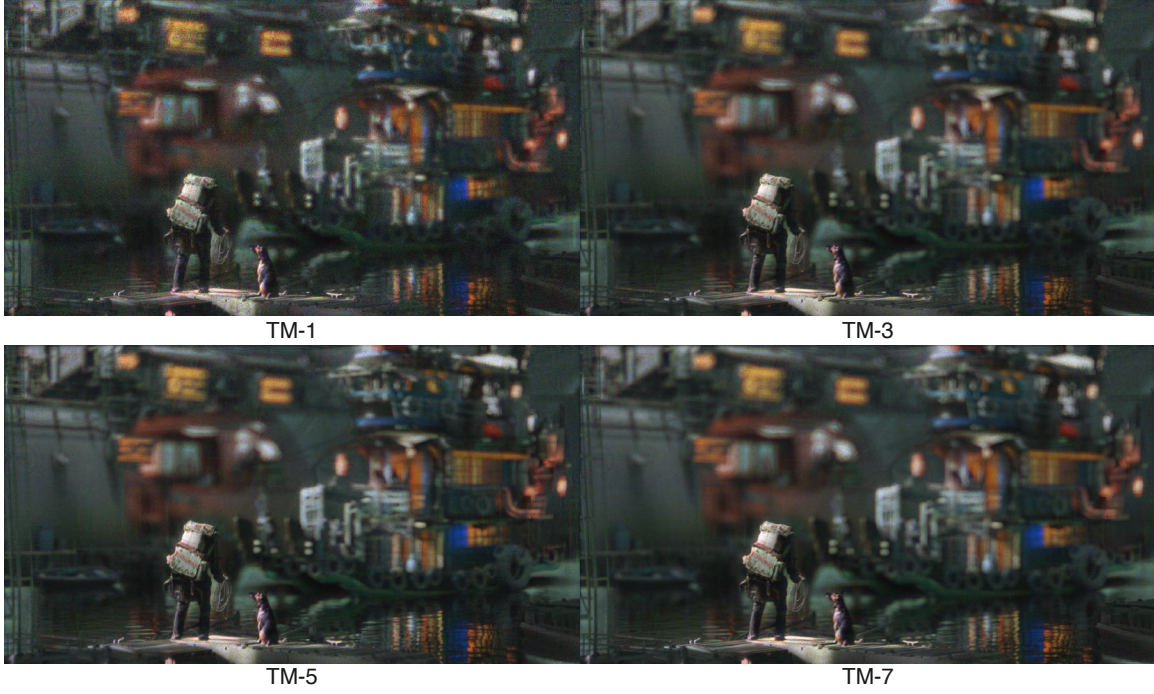


Figure 6-12: Comparison of experimentally captured EC-H holograms using different numbers of time-multiplexed frames.

diminishes after adding more than 5 subframes. The number also dictates the quality of incoherent defocus reproduction since the image formation for every subframe is yet governed by the coherent imaging model, and the model mismatch has to be bridged by the incoherent accumulation across subframes. Note that we use 8-bit phase SLM compared to the 4-bit phase SLM used in Choi et al. [48], potentially reducing the number of subframes needed to achieve the same noise level and increasing the contrast of the experimental results due to better reproduction of dark pixels.

6.3.7 Reproduction of Intra-pupil Motion Parallax

Fig. 6-14, we horizontally shift the pupil center from -2mm to 2mm and capture the 3D holograms. The right bounding boxes show how the rear racing track is progressively occluded by the foreground tree. A unique advantage of the extended eye box enabled by the EC-H is the ability to achieve noticeable intra-pupil depth-dependent motion parallax. In Figure 4 and Figure 6-13, we present horizontal and vertical motion parallax visualizations based on the experimentally captured results.

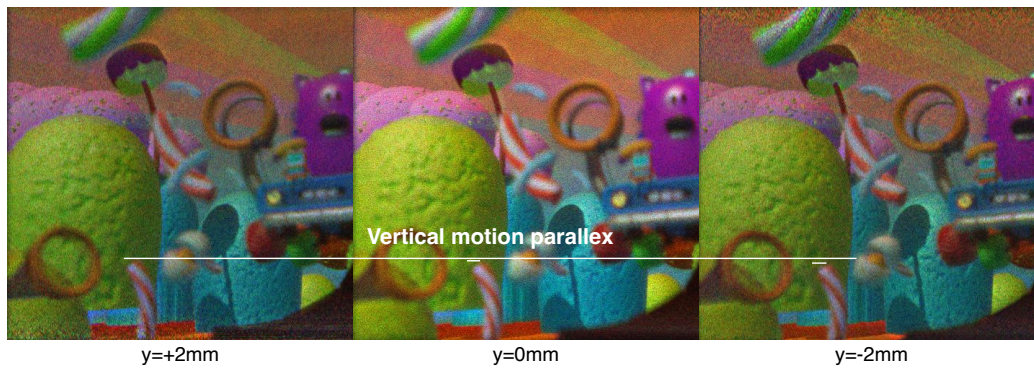


Figure 6-13: Experimentally captured vertical motion parallax for the “Racing Car”. The horizontal white line and small white bars help perceive the motion parallax.

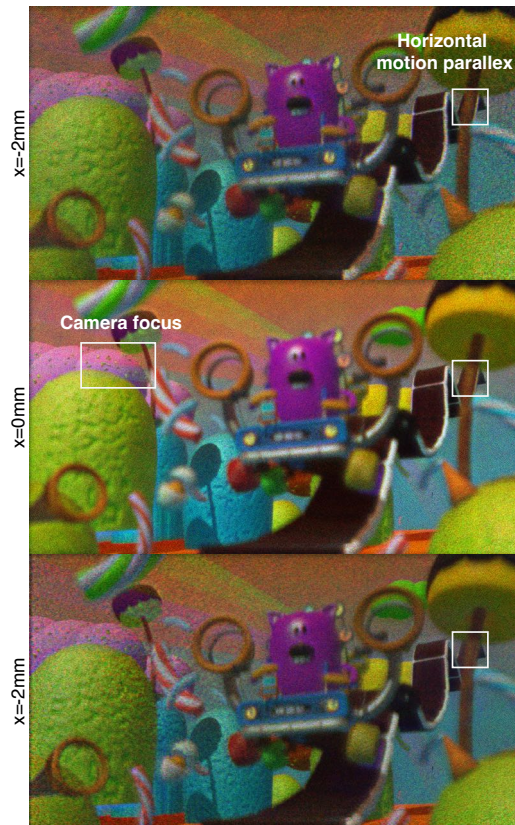


Figure 6-14: Experimentally captured horizontal motion parallax. The bounding box on the right side of the image shows the occlusion of the racing track by the foreground free as the viewpoint moves. The left label marks the pupil offset along the x-axis. Source image: “Racing Car”, reproduced under terms of the CC-BY-SA license, Copyright 2016, by Pokedstudio, published with the author’s permission.

We would like to emphasize that for the most accurate representation of motion parallax, the depth of field images used to supervise the optimization process should ideally undergo a depth-dependent shift. This shift is necessary to reflect the observed motion parallax at a non-center pupil location, which occurs due to changes in the observation position. To achieve this, one can shift the camera position and perspective during the LDI rendering process, generating a grid of LDIs across the full eye box to create position-dependent focal stacks.

However, due to the current SLM specification, the motion parallax induced by the extended eye box is still tiny. Therefore, we decided to omit this additional step, while acknowledging that it deviates from the most physically accurate implementation. Nonetheless, we discovered that even when using the same focal stack to supervise the optimization at different pupil positions, the motion parallax effect can still be induced. This effect arises from the fundamental modeling of holographic image formation with pupil masking, which has been the standard technique to simulate the large motion parallax effect induced by fabricated ultra-high resolution hologram (see Zhang et al. [270] Fig.7/8). In our case, the optimizer attempts to extrapolate the content accordingly. As a result, artifacts in the upward, downward, leftward, and rightward edges can be observed due to the extrapolation process. It will be really interesting to see if image quality and realism can be further improved after closing this gap. We defer this fine-grained optimization to future work.

6.3.8 Additional Phase Patterns & Focal Stacks

Figure 6-15 shows additional phase-only holograms optimized under unfiltered and pupil-mimicking modes. Due to statistical similarity, only one subframe from all time-multiplexed frames is visualized for each scene. The visual appearance of the produced holograms undergoes significant changes when incorporating pupil optimization, as analyzed in the previous section. When optimized under the unfiltered mode, the holograms present low-frequency patterns that vaguely outline the shape of a dilated target image. This becomes invisible in holograms optimized under the pupil-mimicking mode as the design of the pupil batch structurally distributes the energy distribution towards the mid and high-frequency



Figure 6-15: Visualization of exemplary focal stack images and holograms optimized for unfiltered and pupil-mimicking modes for scenes used in this paper.

regions.

6.4 Limitations & Discussions

With the proposed improvements, the EC-H still shares two limitations with HOGD [94]. First, we find that time multiplexing and better global scale initialization cannot effectively address the challenge of reproducing sparse targets. Figure 6-16 illustrates an example of the USAF resolution target with three depth planes. In such cases, concealing the high-order diffractions within the rest of the scene is difficult, as a significant portion of the scene remains black. Time multiplexing, which integrates sub-frames incoherently rather than coherently, does not provide additional freedom to mitigate high-order diffractions as the frame increases (see row 2). One potential solution is to incorporate a second spatial light modulator, enabling the coherent addition of the complex amplitudes of the two SLMs to cancel out the remaining high-order diffractions [50]. We also note that when optimizing under pupil-mimicking mode (see row 3), the artifacts can be largely hidden at the center pupil location since it effectively performs a low-pass filter; however, when shifted toward the periphery of the eye box, localized high-order diffractions can still be observed.

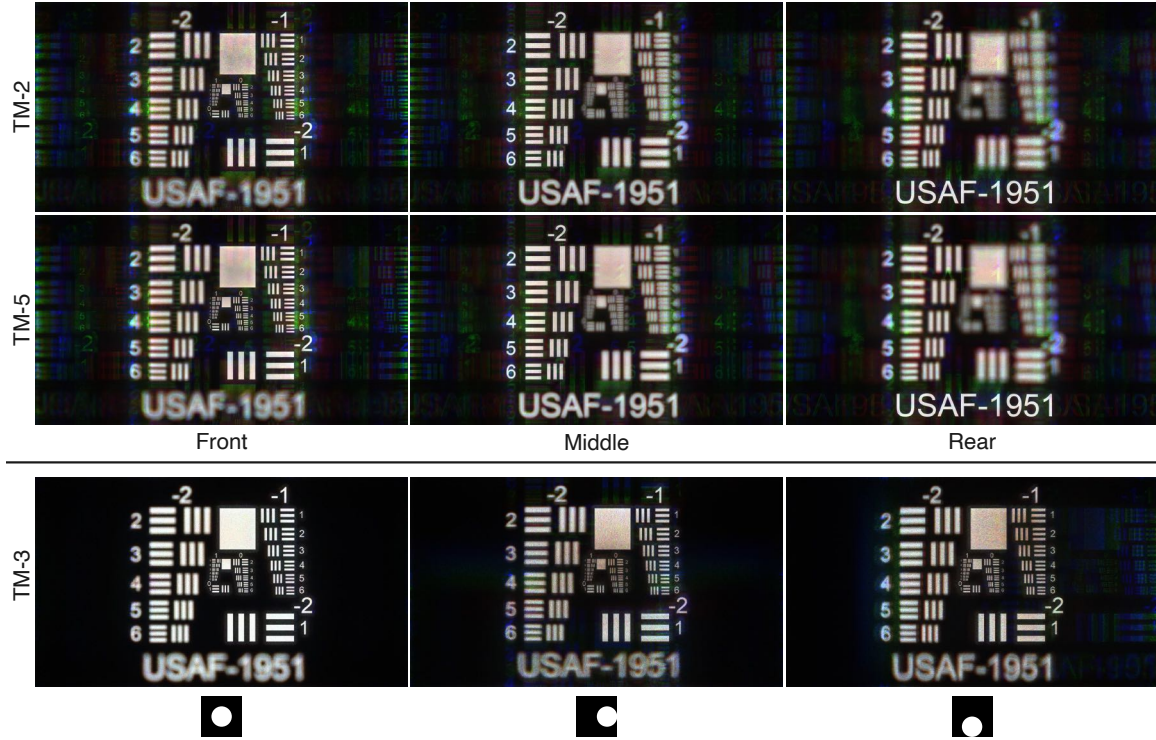


Figure 6-16: Simulated reconstruction of the USAF-1951 target under unfiltered and pupil-mimicking modes.

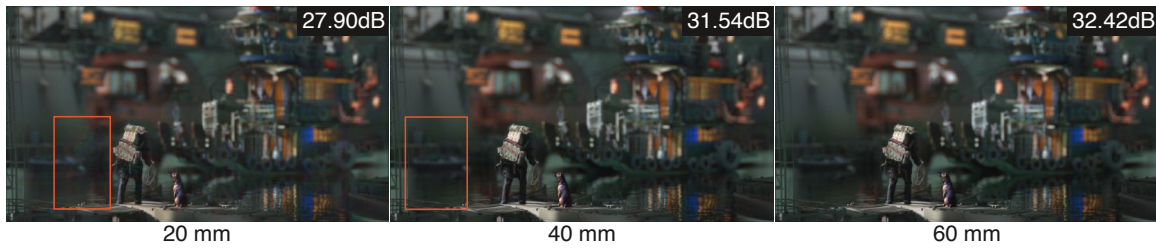


Figure 6-17: Simulated reconstruction at different propagation distances with numbers indicating PSNR with respect to the target image.

Another limitation is the sensitivity of image quality to the propagation distance. In the absence of filtering and with the SLM plane having a unit amplitude, a certain propagation distance is necessary to diffract the field into the desired target amplitude distribution (e.g., 60mm for the results in this paper). As the propagation distance increases, high-order copies separate further away from the central order, reducing the overhead and difficulty of concealing them within the central diffraction. Figure [6-17](#) compares the “PartyTug 6:00AM” image (focused on the man) reconstructed with propagation distances of 20mm,

40mm, and 60mm. The high-order copies of the man are distinctly visible with a 20mm propagation distance, gradually diminishing as the distance extends. This limitation will naturally be resolved as the pixel pitch of the SLM becomes smaller, since a wider diffraction angle will lead to a greater separation of the high-order diffractions from the central order under the same propagation distance.

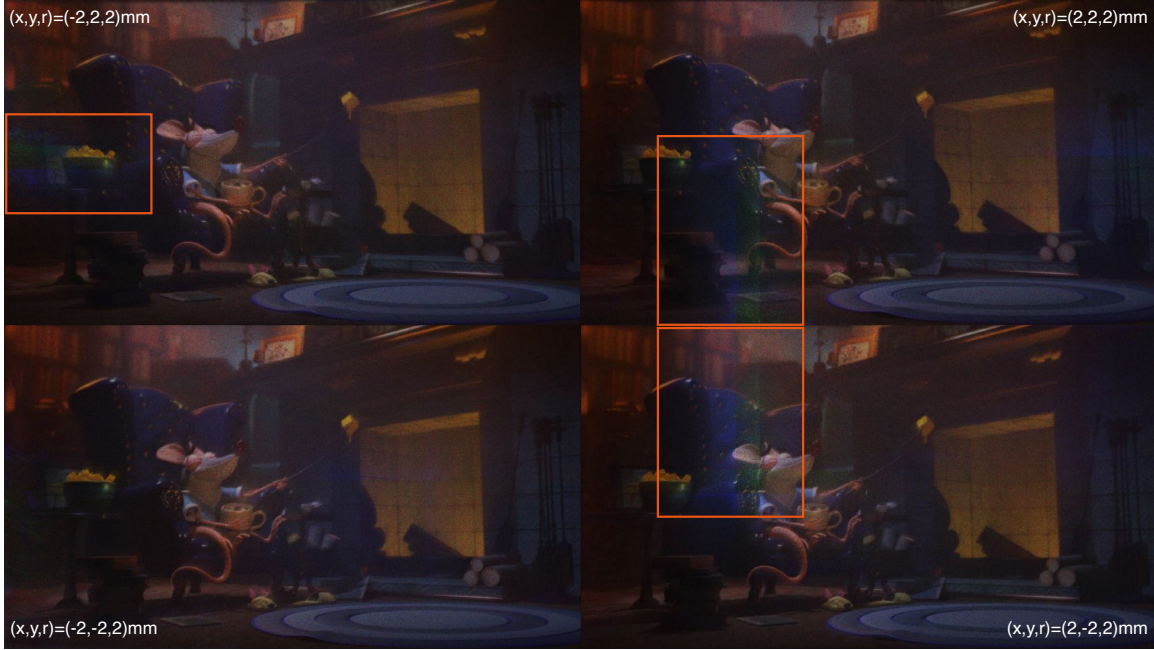


Figure 6-18: Simulated reconstructions at pupils shifted far away from the eye box center.

For the pupil-mimicking mode, we find that achieving results with consistent minor artifacts becomes extremely challenging when further pushing the pupil away from the eye box center (e.g., $(x, y, r) = \pm 2, \pm 2, 2$)mm, even when including them in the fixed pupil grid as visualized in Fig. 6-18, where we show the reconstructions at extreme corners of the four quadrants. Note that while one of the results only has minor artifacts, artifacts in the other three results have become prominently noticeable. We note that the maximum expansion of the eye box is roughly 2x of the original eye box created by the shortest wavelength. This is caused by the fact that regardless of the optimization strategy (see Fig. 6-10), the energy distribution has a zero crossing at 1/6 and 5/6 of the expanded spectrum due to the Sinc modulation, making it impossible to maintain the level of brightness at these locations. This presents a limit on how much we can extend the eye box formed by the central order

diffraction using high-order diffractions. As discussed in the main paper, including higher-resolution phase masks during joint optimization may ideally improve the display etendue.

Finally, we want to highlight that the utilization of the camera (hardware)-in-the-loop (CILT) technique has gained popularity in enhancing the quality of experimental results [204]. The proposed mixed-batch pupil optimization method remains compatible with such optimization approaches, as users can leverage a motorized translation stage and a motorized iris to simultaneously control the position and size of the pupil. By accumulating and averaging the gradients computed for each pupil setting, CILT can encompass the contribution of image discrepancy across all pupil positions and sizes.

Chapter 7

Differentiable Procedural Material Modeling and Capture

In virtual and augmented reality, advancing display technologies to visually render content in 3D is just one aspect; equally crucial is the quality of appearance modeling for digital content, which profoundly impacts perceived realism. A pivotal aspect of this modeling is estimating material appearance, which involves capturing the spatially-varying bidirectional reflectance functions (SVBRDF), which dictates how light rays interact with a material, considering both incoming and outgoing directions at a specific position.

In the realm of computer graphics, SVBRDF is typically represented via a collection of material maps (such as albedo, normal, roughness, and metallic properties in Disney’s principled BRDF model [33]), conventionally stored as image-based textures. These textures are often derived from scanned real-world objects or directly crafted using image processing tools like Adobe Photoshop. However, this authoring workflow presents challenges. It can be difficult to create meaningful appearance variations, manage changing resolutions, ensure seamless tiling, and maintain low memory usage.

More recently, modern material authoring tools like substance 3D designer and Blender address these challenges by offering procedural material modeling capabilities. Procedural material represents materials constructively using a program, which overcomes many limitations associated with image-based materials. Typically modeled as a directed acyclic node graph, a procedural material uses a sequence of filters with user-tunable parameters

to progressively modify input patterns/noises to collectively control the output material appearance. By crafting diverse graph structures and optimizing associated parameters, procedural material graphs serve as powerful mathematical abstractions, capable of creating high-quality materials that are resolution-independent, tileable, and easily editable. However, procedural material authoring requires substantial domain expertise and time effort. These limitations not only impede experienced artists from efficiently producing high-quality materials for professional applications with escalating demands for realism but also hinder ordinary users with limited expertise from crafting realistic virtual environments and digital twins for emerging AR/VR applications.

This chapter copes with these limitations with a machine-learning-assisted procedural material capture system to recover SVBRDF of a near-planar surface from a single flash image without leveraging user expertise. This involves the development of a domain-specific differentiable procedural material modeling library, coupled with a pipeline to identify an appropriate procedural model from a model library, and estimating the node parameters of that model to best match the target appearance using gradient-based iterative optimization. We demonstrate that our approach can recover high-quality procedural materials from real-world photographs, outperforming prior procedural material capture method that don't have access to the differentiability of the graph.

7.1 Preliminaries

7.1.1 Rendering Equation & Material Modeling

The rendering equation describes the process of generating realistic images by simulating the behavior of light. It expresses the total amount of light leaving a particular point in a scene, known as the outgoing radiance, as the sum of emitted light and the reflected light from all other points in the scene [132]. Mathematically, it can be expressed as follows:

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = L_e(\mathbf{p}, \boldsymbol{\omega}_o) + \int_{\Omega} f_r(\mathbf{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{p}, \boldsymbol{\omega}_i) (\boldsymbol{\omega}_i \cdot \mathbf{n}) d\boldsymbol{\omega}_i \quad (7.1)$$

Where:

1. $L_o(\mathbf{p}, \omega_o)$ is the outgoing radiance from point $\mathbf{p}$ in the direction ω_o .
2. $L_e(\mathbf{p}, \omega_o)$ is the emitted radiance at point $\mathbf{p}$ in the direction ω_o .
3. $f_r(\mathbf{p}, \omega_i, \omega_o)$ is the bidirectional reflectance distribution function (BRDF), which describes how light is reflected from a surface at point $\mathbf{p}$ with incident direction ω_i and outgoing direction ω_o .
4. $L_i(\mathbf{p}, \omega_i)$ is the incoming radiance from the direction ω_i .
5. $(\omega_i \cdot \mathbf{n})$ represents the cosine factor that accounts for the angle of incidence with the surface normal $\mathbf{n}$.
6. Ω represents the hemisphere of possible incoming light directions.

This equation forms the basis for many rendering algorithms, including ray tracing and path tracing, which simulate the propagation of light through a scene to generate realistic images.

Material modeling concerns about modeling $f_r(\mathbf{p}, \omega_i, \omega_o)$, a 6-dimensional SVBRDF function where $\mathbf{p}$ describes a 2D location over an object's surface, $\omega_i = (\alpha_i, \phi_i)$, and $\omega_o = (\alpha_o, \phi_o)$ are the incident and outgoing directions parameterized in the spherical coordinate. One can create a tabulated SVBRDF by densely sampling every lighting and view direction over the hemisphere [186], however, the effort of measurement and the amount of data easily becomes prohibitive. In practice, the angular distribution of real-world BRDFs typically lies on a low-dimensional manifold that can be well-approximated by parameterized functions with significantly fewer parameters.

A common-used physically-based BRDF model is microfacet model [205]. It assumes that a surface consists of countless small, flat microfacets that are oriented randomly. Each microfacet behaves like a perfect mirror, reflecting light according to the laws of geometric optics, such as Snell's law and the law of reflection. However, due to surface roughness, the microfacets are not perfectly aligned, resulting in a spread of reflected light directions.

The microfacet SVBRDF is typically defined using the following equation:

$$f_r(\omega_i, \omega_o, \mathbf{p}) = \frac{\sigma_{\mathbf{p}}}{\pi} + \frac{F(\omega_i, \mathbf{h}, F_0)D(\mathbf{h}, \alpha_{\mathbf{p}})G(\omega_i, \omega_o, \alpha_{\mathbf{p}})}{4(\mathbf{n} \cdot \omega_i)(\mathbf{n} \cdot \omega_o)} \quad (7.2)$$

Here, $\sigma_{\mathbf{p}}$ is the material's diffuse color at $\mathbf{p}$, $\alpha_{\mathbf{p}}$ is the material's roughness at $\mathbf{p}$, $\mathbf{n}$ is the surface normal, F_0 is the reflectance of a surface when light is perpendicular (normal) to the surface, $\mathbf{h}$ is the halfway vector between ω_i and ω_o , defined as $\mathbf{h} = (\omega_i + \omega_o) / \|\omega_i + \omega_o\|$, $F(\omega_i, \mathbf{h})$ is the Fresnel reflection term, which describes the amount of light reflected at the interface between the surface and the surrounding medium, $D(\mathbf{h}, \alpha_{\mathbf{p}})$ is the microfacet distribution function, which characterizes the distribution of microfacet normals around the halfway vector $\mathbf{h}$, $G(\omega_i, \omega_o, \alpha_{\mathbf{p}})$ is the geometric attenuation factor, which accounts for shadowing and masking effects due to the surface microstructure. We refer readers to [205] for the details of the Fresnel reflection term, microfacet distribution function, and geometric attenuation factor.

The microfacet SVBRDF model allows for realistic rendering of a wide range of surfaces, from glossy to rough, and from conductors to dielectric. The thesis adopts the Disney principled BRDF that encodes microfacet SVBRDF parameters as four maps: albedo, normal, roughness, and metallic. The normal and roughness maps are directly mapped to $\mathbf{n}$ and $\alpha_{\mathbf{p}}$ in Eq. 7.2, while $\sigma_{\mathbf{p}} = (1.0 - \text{metallic}) * \text{albedo}$ and $F_0 = 0.04 * (1.0 - \text{metallic}) + \text{albedo} * \text{metallic}$.

7.1.2 Procedural Materials

Procedural material is a set of mathematical functions, often represented as nodes in a graph-based system, that describe the visual properties of the material. The node graph largely consists of two types of nodes: *generators* and *filters*. Generator nodes create spatial textures from scratch based on user-specific parameters, and include both noise generators (like Perlin noise) and structured pattern generators. Filter nodes manipulate input textures using operations ranging from pixel value manipulations (like color or contrast edits) to image processing (like filtering, warping, blending, etc.); these nodes are parameterized by the control parameters of the functions they implement (for example, kernel size for a box filter or opacity for a blending node). The connectivity of the graph defines a sequence of operations that start with 2D scalar or vector maps (usually created by generator or data store nodes) and manipulate them (usually operations defined by the filter nodes) to finally

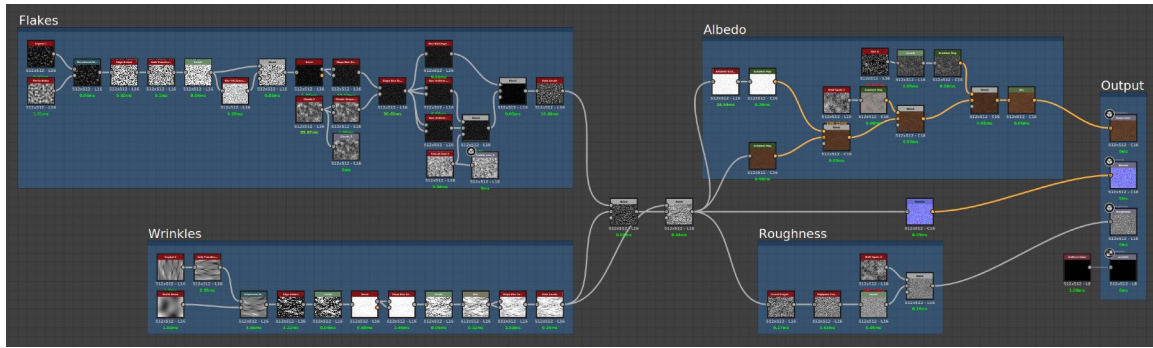


Figure 7-1: An exemplar leather material graph created using substance 3D designer. The graph starts with several generator nodes that provide initial patterns for subsequent filter nodes to manipulate. The graph is internally partitioned into subgraphs that create different visual attributes of the leather pattern. It outputs four material maps that define the parameters of an SVBRDF.

output a set of materials maps. Figure [7-1](#) shows an example leather procedural material graph. This graph has 43 filter nodes and 127 node parameters with complex connectivity that allows it to produce a wide range of photorealistic leather material appearances. Typical production procedural materials are more complex than this example. Compared to image-based materials, procedural materials possess several advantageous characteristics. One of the key advantages is their scalability. Because they are generated algorithmically, procedural textures can be rendered at any resolution without loss of detail, making them ideal for high-resolution rendering and real-time applications such as video games and virtual reality. Additionally, procedural materials are often tileable, meaning they can seamlessly repeat across surfaces without visible seams or artifacts.

Another benefit is their flexibility and efficiency. Since they are generated programmatically, procedural textures can be easily modified and adjusted in real time by changing node parameters, allowing for rapid iteration and experimentation during the creative process. Artists can also design high-level semantic parameters, often referred to as exposed parameters, by grouping several node parameters within a functional graph. This facilitates more advanced appearance controls, greatly simplifying the reuse of materials by others. Furthermore, procedural materials can be stored and manipulated using relatively small amounts of data (i.e., only programs and parameters but not the final material maps), making them well-suited for applications with limited storage or bandwidth constraints.

7.2 Motivation

Capturing the parameters of SVBRDF is a classic problem in computer graphics; please see Guarnera et al. [95] for a recent survey of this body of work. While one could scan the entire 6-dimensional SVBRDF using a spherical gantry, this is expensive and rarely necessary. The goal of many previous methods is to reduce the measurement effort and output parameter maps for diffuse color, normal, specular roughness/glossiness, and specular albedo. Among the first solutions that scanned such maps using a linear light was Gardner et al. [81]. Recently, Aittala et al. [6] used Fourier patterns projected onto an LCD screen in combination with a general optimization algorithm to recover image-based SVBRDF, but their number of measurements was still quite high (hundreds).

The work of Li et al. [163] and Deschaintre et al. [60] introduced a direct end-to-end prediction approach using convolutional neural networks, mapping single-image measurements to material parameter maps. These approaches have also been extended to handle multiple input images, either via pooling of features [61] or inverse rendering-based optimization in an auto-encoder latent space [80]. For large-scale materials, Deschaintre et al. [62] fine-tuned the network on close-up flash photographs of the test sample to achieve better generalization for the entire material. While these results can be impressive and obtained with significantly reduced time and domain knowledge, the reconstruction quality of obtained image-based SVBRDF varies with the input, often showing blurring or loss of quality for novel views or light positions. Moreover, the resulting maps have a low resolution (typically 256×256), are not tile-able nor easily editable afterwards. This is not only because single-image SVBRDF estimation is an ill-posed (under-constrained) problem, but also because image-based method lacks prior knowledge of the underlying materials.

Hu et al. [113] recently introduced a single-image procedural material capture method to overcome disadvantages inherited in image-based methods, where the procedural graph can serve as a material prior to regularize the output. Given an input image, their method selects a procedural material graph from a library, and uses a neural network to directly estimate graph parameters to best match the given image. However, they constrain themselves to a Lambertian BRDF model (predicting only diffuse albedo and surface normals)

and predict only a subset of the graph parameters (the most high-level semantic parameters). Moreover, these network predictions can be coarse, leading to apparent differences between the input image and the predicted graph result. Finally, their work requires a separate deep parameter prediction network for every single graph in their library, leading to a substantial memory footprint.

In contrast to such a feed-forward prediction approach, we propose an optimization-based method, where we treat the procedural material graph as analogous to a neural network whose parameters can be estimated; in our case, the parameters are not network weights, but the various node parameters in the graph. This approach requires a key element: a differentiable version of a given material graph that can be evaluated forward and backward (for gradient computation). We accomplish this by introducing a procedural material modeling library DiffMat to translate procedural material node graphs into differentiable programs. We demonstrate that DiffMat can handle production-grade procedural material graphs (with hundreds of nodes and thousands of parameters) representing complex SVBRDFs. We combine the translated differentiable material graphs with a differentiable rendering layer to create an end-to-end differentiable graph parameter-to-image pipeline that enables parameter estimation through gradient descent. It is worth note Guo et al. [100] also address parameter estimation for procedural material models. Their focus is more on a Hamiltonian Monte Carlo sampling approach in a Bayesian framework. However, their procedural materials are small hand-written pieces of code and don't scale to production-grade materials or materials created by industry-standard software.

We evaluate the proposed framework using both synthetic and real-world material photos. Our optimization-based approach achieves a closer match to the input photo compared to other procedural-material-based methods. Additionally, it also avoid blurriness, loss of detail, and low-resolution problems associated with image-based material capture methods from a single image.

7.3 DiffMat: A Differentiable Procedural Material Modeling Library

Recent work on learning-based inverse graphics has found that combining neural networks with graphics models can lead to more interpretable, and in many cases, more accurate results [118, 163, 60, 243]. However, this requires that the graphics model itself be differentiable to allow for training via back-propagation. This has prompted the development of differentiable libraries for computer vision [210], rendering [105, 161, 197, 168], geometry processing [74] and spatially sparse computing [117]. Similarly, we wish to explicitly use procedural node graphs as the output of material capture as they provide a compact and interpretable material representation. This is currently not possible because there exist no tools to efficiently evaluate procedural materials (forward and especially backward) in an optimization framework.

To this end, we develop DiffMat, a PyTorch-based differentiable library that supports the functionality and expressiveness of production-grade procedural modeling tools while allowing for easy integration with deep learning tools. We use substance 3D designer² as an industry-standard reference software and build DiffMat to provide differentiable node routines that can match it with per-pixel accuracy. We do this for two reasons: first, by matching its functionality, our framework can be integrated into real-world procedural material design workflows. Second, there exist large public libraries of high-quality artist-designed substance 3D designer material graphs, that we leverage in our material capture framework. That said, our implementation is independent of substance 3D designer, and can be adapted to match the capabilities of other node graph authoring applications. DiffMat is publicly released at: <https://github.com/mit-gfx/diffmat>.

7.3.1 Design Overview

Given the similarity between a material node graph—that sequentially processes inputs using filter nodes—and a neural network, we follow the general design of PyTorch in Diff-

²<https://www.substance3d.com/products/substance-designer/>



Figure 7-2: The code structure of DiffMat

Mat’s API design. Specifically, DiffMat defines function routines (similar to *torch.nn.functional*) for stateless evaluation of node operations (defined in `core/material/functional.py`), and wrapper classes as their optimizable equivalent (similar to *torch.nn*), whose internal attributes represent node parameters that can be optimized (defined in `core/material/node.py`). The input noise and pattern created by the generators are largely based on a differentiable implementation of the FX-Map (defined in `core/fxmap`), a generalized noise generation procedure that recursively subdivides the texture canvas into smaller regions and renders local texture patterns in these regions. A translator module (defined in `translator/`) decodes material graph structures defined in XML format, sourced from substance 3D designer. It automatically translates these structures into a material graph class within our library and facilitates exportation once material optimization is complete.

7.3.2 Generator Nodes

Generator nodes generate two types of inputs for the procedural graph: noise and patterns. These nodes often come with parameters that can be adjusted to control properties like scale, intensity, and randomness, allowing for extensive customization of the generated textures.

The core functionality to create many generator nodes is the FX-Map node, a generalized noise generation procedure that recursively subdivides the texture canvas into smaller regions and renders atomic texture patterns in these regions. Functionally, FX-Map represents a generalized noise generation procedure that recursively subdivides the texture canvas into smaller regions and renders atomic texture patterns in these regions. The subdivision follows a quadtree structure with a canvas area evenly split into four squares in the next quadtree layer. The 2^{2d} squares at the d -th quadtree layer constitute an octave of depth d . The rendering step draws one or multiple atomic patterns at the center of each square (d, i, j) ($i, j \in \{0, 1, \dots, 2^d - 1\}$). Each atomic pattern $F_{d,i,j,k}(x, y)$ (k is the pattern index; x, y are pixel coordinates) undergoes a user-defined spatial transformation $\xi_{d,i,j,k}[\cdot]$ that can be either deterministic or random. Deguy and Benassi [58] showed that the FX-Map procedure achieves a close approximation of intermittent locally self-similar Gaussian processes

(ILSSGP), of which fractional Brownian motion (fBM) is a special case.

7.3.3 Filter Nodes

Filter nodes consist of image processing operations that are closely analogous to operations common in convolutional neural networks (CNNs). With very few exceptions, filter nodes either act as convolutions (blurs, edge detectors, etc.) or as per-pixel operations (curves, blends, thresholds) to each pixel of the input map(s) separately. This is closely related to the typical components of CNNs: convolutions and point-wise operations (activation functions such as ReLU, sigmoids, etc.).

In practice, filter nodes are divided into atomic filters and non-atomic filters. Atomic filter nodes are the basic building blocks, namely any graph when decomposed into its core operations, will consist of only these nodes. These include basic operators such as blur, warp, blend, distance, gradient map, and color adjustments, but also procedural modeling-specific operations like FX-Map and Pixel/Value-Processor. DiffMat provides implementations of all 21 atomic filter nodes in substance 3D designer and accurately reproduces their behavior.

Non-atomic filter nodes are pre-made graph instances (compound nodes) that are constructed using atomic filter nodes and are designed to reproduce specific visual effects (e.g., edge detector) and enable high-level artistic control over material appearance. substance 3D designer has more than 150 non-atomic filter nodes. DiffMat implements the 110 most frequently used non-atomic filter nodes in the Substance Source dataset and is thus able to represent a significant portion of these graphs. We fuse inefficient calls of atomic filter nodes in these pre-made graphs to improve their computational efficiency in DiffMat's implementation.

We consider continuous parameters (e.g., opacity in the blend node) as optimizable parameters and discrete parameters (e.g., number of tiles along the row in the tile generator node) as non-optimizable parameters. The wrapper class holds a trainable parameter list and initial values for continuous parameters. Calling the wrapper class with required input images and discrete parameters will evaluate its functional counterpart. Because different

continuous parameters have different ranges, we store them in the wrapper class as trainable parameters with a range of $[0, 1]$ along with non-trainable, node-specific minimum/maximum parameters that are used to map the parameters to their final values. To prevent the optimization from driving a trainable parameter out of its domain, the wrapper class applies clamping internally before calling the forward function. Implementing the nodes in PyTorch gives us the gradient computation for free by using PyTorch’s auto-differentiation.

7.3.4 Automatic Graph Translator

A substance 3D designer document encodes the procedural material graph and its associated exposed parameters in XML. The DiffMat translator parses and analyzes this XML document and generates Python programs that replicate the functionality of the graph. This translation is performed as follows:

1. **Graph replication.** The translator first detects and stores the default values of all exposed parameters of the input graph. It then replicates the whole graph in DiffMat by using the corresponding PyTorch operations with the same connectivity. Graph nodes without input connections are regarded as generators and those without DiffMat implementation are converted to precomputed input bitmaps using the Substance Automation Toolkit and serve as non-optimizable patterns.
2. **Node parameter conversion.** For every filter node, the translator interprets and converts its parameters according to a series of node type-specific rules. If a node parameter is defined as a dynamic function of one or more exposed parameters, its initial value is set by constructing and evaluating its *value-processor* using the pre-stored exposed parameters. For node graphs that use unimplemented filter nodes, the translator automatically replaces them with *passthrough* nodes.
3. **Data flow analysis.** After constructing the graph structure, the translator examines node contributions to the graph outputs to eliminate redundancy. While a graph may produce many material maps, we only consider four (albedo, normals, roughness, and metallicity). Therefore, the translator runs a backward breadth-first search (BFS)

and a forward BFS to identify the largest subgraph that is accessible from these four output nodes and trims all unused nodes.

4. **Program generation.** Finally, the translator performs topological sorting on the final graph and generates a node sequence consistent with data dependency. Based on this sequence, the translator organizes the results from previous steps and outputs a complete Python program.

To enable effortless high-level post-editing, we offer the option to retain the exposed parameters during graph translation. In the subsequent optimization, users can choose to optimize either exposed parameters alone or exposed parameters plus the rest of node parameters not directly tied to the exposed parameters. If users wish to explore a larger design space, they can further choose to remove the *value-processors* prescribed by the exposed parameters and optimize all node parameters freely. This is often useful if users wish to create their custom exposed parameters after the optimization.

7.4 End-to-end Procedural Material Capture

Given a library of differentiable graphs, we follow a two-step process to recover a procedural material from a target photograph. First, we identify the most relevant graphs for the input photograph. We train networks to, given an image, directly predict the node parameters of each graph. We process the input photograph with these parameter prediction networks, evaluate the error between images rendered with these predictions and the input image, and pick the top-3 graphs with the lowest errors. Second, we use an optimization-based method to fit the parameters of each of these graphs to the target image. Our entire pipeline is illustrated in Figure [7-3](#).

7.4.1 Image-based Parameter Optimization

Given a specific translated graph G , our goal is to estimate node parameters that will produce a SVBRDF whose rendered appearance will reproduce a target image, I^* . In particu-

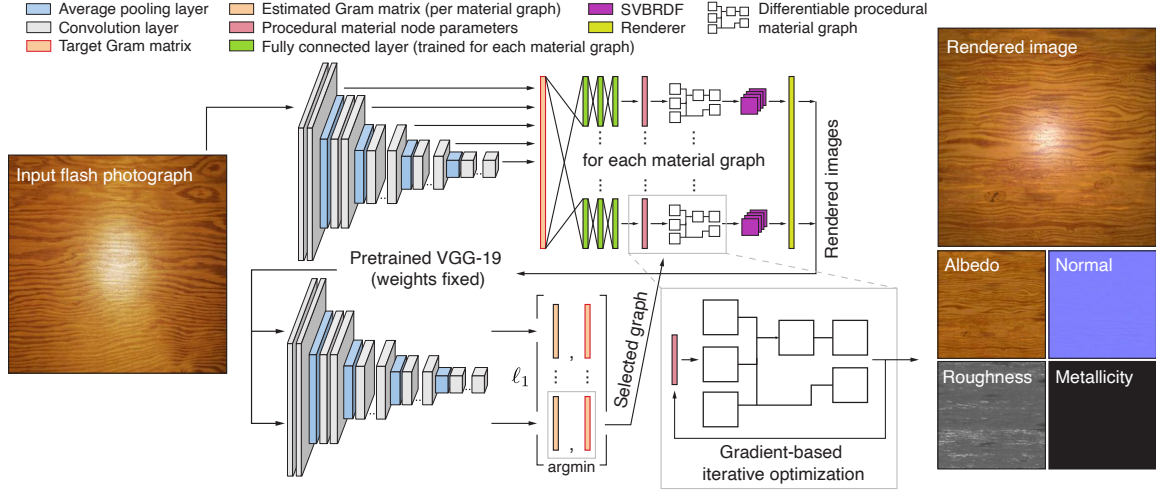


Figure 7-3: Overview of MATch’s fully differentiable procedural material design pipeline. Given an input target image (potentially a flash photograph captured using a cellphone), we compute an input texture descriptor (flattened Gram matrix) using a pre-trained VGG network. For each procedural material in our library, we train a parameter prediction network by appending fully connected layers to this texture descriptor. We use these models to predict node parameters of the target image for every material graph, and generate the corresponding SVBRDFs using the differentiable material graphs translated by DiffMat. We render images from these SVBRDFs, compute their corresponding texture descriptors and select the top-3 material graphs with the lowest ℓ_1 texture descriptor difference with respect to the input. Finally, we refine the previously predicted node parameters using a gradient-based optimization with our differential node graphs to improve the match with the target image. The material graph (along with the refined node parameters) that produces the closest matched result is output as the final result. Here, we visualize this pipeline with a real, captured wood material sample.

lar, we are interested in the material parameter vector θ of length k , a concatenation of all k continuous parameters of the node graph G , which can be optimized via backpropagation.

We define the parameter map evaluation operator M that encompasses the evaluation of G . Given θ as input, it produces parameter maps of a microfacet BRDF (see [7.1.1](#)): albedo $\mathbf{a}$, normal vector $\mathbf{n}$, roughness $\mathbf{r}$ and metallicity $\mathbf{m}$. Therefore,

$$(\mathbf{a}, \mathbf{n}, \mathbf{r}, \mathbf{m}) = M(\theta). \quad (7.3)$$

The rendering operator R takes the generated maps and computes a rendered image under known illumination using the rendering equation. We assume a single target image captured by centered co-located point light and camera (simulating a cellphone camera

with flash), though extensions to other configurations, including multiple lights or views, is straightforward. The predicted synthetic image can be written as:

$$\mathbf{I} = R(M(\boldsymbol{\theta})) = R(\mathbf{a}, \mathbf{n}, \mathbf{r}, \mathbf{m}). \quad (7.4)$$

Note that both operators M and R are differentiable; this allows for gradient computation by backpropagating through the entire expression $R(M(\boldsymbol{\theta}))$. Differentiable rendering operators have also been previously used for material capture, albeit with image-based SVBRDF representations [163, 60].

Optimizing for $\boldsymbol{\theta}$ requires us to define a loss function between the rendered image, $\mathbf{I}$ and target image, $\mathbf{I}^*$. Such a loss cannot rely on pixel-perfect alignment of texture features because the spatial patterns between the two images are unlikely to match exactly. We use the popular style loss function that was proposed by Gatys et al. [83, 82] for image style transfer, and has been used by many previous methods, including material capture [5, 100]. This loss function can be written as:

$$L_G = \|\mathbf{T}_G(\mathbf{I}) - \mathbf{T}_G(\mathbf{I}^*)\|_1, \quad (7.5)$$

where $\mathbf{T}_G$ is a Gatys texture descriptor defined by the concatenation of the Gram matrices of the five feature maps before each pooling layer of the VGG [233] network.

We compute the Gram matrix using the VGG-19 network without batch normalization, provided by the *torchvision* library. We also replace the max-pooling by the average pooling, as suggested by Gatys et al. We normalize images by the mean/variance for ImageNet data before feeding them to the VGG network. We compute the Gram matrix descriptors for images at multiple resolutions (specifically resolutions of 512, 256, and 128); this improves the perceptual match over multiple scales.

We use Adam optimizer [141] with a learning rate of 5e-4 to optimize the Gram matrix loss. Because Eq. 7.4 is fully differentiable, we can backpropagate through the rendering and the whole material graph to update the values of $\boldsymbol{\theta}$ directly. This is in contrast to previous work that used the style transfer loss to optimize for material maps/images [5].

7.4.2 Network-based Graph Selection and Initialization

The choice of the graph used to match a target image affects the quality of reproduction. Yet, when a large set of material graphs is given, manually choosing the appropriate graph is non-trivial; this is especially so, because as shown in Fig. 7-5, production-grade procedural graphs are often “over-parameterized” and can represent a wide range of material appearance beyond their semantic class. Therefore, we propose an automatic graph selection method. This method first directly predicts the parameters of each graph for the target image and uses this prediction to select candidate graphs. These predictions can also be used as initializations to the optimization, accelerating convergence, and leading to a better solution for difficult input targets.

Similar to Hu et al. [113], we predict the material parameter vector using a neural network that takes the target image I^* and returns an estimate of θ . Hu et al. train a parameter prediction network, based on the AlexNet [147] backbone, for *every* procedural graph in their dataset. This incurs a significant effort to train models for each material completely from scratch and has a large storage footprint.

Instead of training individual deep networks end-to-end, we extract image features using a pretrained VGG network and only train multi-layer perceptrons with three fully connected layers for each procedural material. As shown in Figure 7-3, we construct the same Gram matrix descriptor T_G as above, which (when flattened) produces a vector of length l . We pass this feature through three fully connected layers of size $l \times 3k$, $3k \times 3k$ and $3k \times k$, with a ReLU unit in between the layers, where k is the number of graph parameters.

A final sigmoid produces estimates in the range $[0, 1]$, which are then remapped to each parameter’s actual range. Note that because the length of T_G is invariant to the input image resolution, the same architecture can be applied to varying input resolutions. This approach combines the advantages of a powerful feature extractor with the small compute and memory requirements of training small per-graph networks.

We train the fully connected layer weights with synthetic data generated at training time; thus no dataset storage nor I/O cost is incurred. For each iteration, we randomly generate a ground truth material vector θ^* and evaluate the corresponding maps a^*, n^*, r^*, m^*

(Eq. 7.3) and the ground truth synthetic image I^* (Eq. 7.4). We then pass I^* through the (fixed) Gram matrix descriptor T_G and the (trainable) fully connected layers, resulting in an estimated parameter vector θ . Because of our fully differentiable pipeline, we can generate the material maps and rendered images for this parameter vector, resulting in predicted maps a, n, r, m and image I .

We train the weights of the fully connected layers with a weighted combination of a parameter loss and style transfer loss (Eq. 7.5):

$$L_{\text{net}} = \lambda_{\theta} L_{\theta} + \lambda_g L_G. \quad (7.6)$$

The parameter loss L_{θ} is defined as the ℓ_1 difference between material parameter vectors:

$$L_{\theta} = \|\theta - \theta^*\|_1. \quad (7.7)$$

The style transfer loss serves as an image-based regularizer and teaches the network how the predicted parameters influence the visual appearance of the output material. We note that this term is not available to Hu et al. [113] due to their non-differentiable material graph. We have also experimented with a per-pixel ℓ_2 loss on the images and maps, but found the predicted parameters tend to produce blurrier maps.

We use Adam optimizer with a learning rate of 1e-4 and a time-based learning decay rate of 0.97 to train the network. We set $\lambda_{\theta} = 1.0$ and $\lambda_g = 0.0$ for the first 1000 iterations, and then update $\lambda_g = 0.05$ for the rest of training. We adopt this approach as we find the regularization of the style transfer loss is more effective when the estimated material appearance becomes reasonably close to the target. The number of required iterations varies depending on the complexity of the material graph, but we found 10000 iterations with a batch size of 5 is generally sufficient for most of the material graphs we tried. As we show in Figure 7-4, combining a shared high-quality VGG feature extractor with a small number of per-material layers yields high accuracy prediction at lower training footprints.

Given a user-input image, we extract its texture descriptor feature and feed it to the parameter prediction networks for every graph. The predicted parameters are then fed to their material graphs and rendered as images. We select the top-3 graphs whose renderings

have the lowest error in the Gatys texture descriptors with respect to the input image. Finally, we optimize all these graphs (Sec. 7.4.1) to match the target image—starting from the predicted parameter as initialization—and keep the result with the lowest error.

While our graph selection and initialization approach has similarities to Hu et al. [113], it also has several advantages. First, our graph selection and parameter prediction reuse the same network, whereas Hu et al. uses pre-trained VGG-19 and clustering for graph selection, and individual deep networks for parameter prediction. Second, when a new material graph is added to the material graph set, Hu et al. would require re-running the clustering with *all* existing material graphs; our approach only operates on the *new* material and does not require re-computing any existing networks. Finally, we select the material graph from the entire pool of materials, while Hu et al. only select from the graphs made for the matched materials. As we show later, selecting from all materials is beneficial, allowing us to use graphs intended for very different materials to successfully reproduce the input image.

7.5 Results

We evaluate the performance of MATch over both synthetic images and captured real-world photographs. All results are computed from a single 512×512 input photograph captured (or rendered) under flash illumination. We use “Default” to denote rendering of materials that created by the out-of-the-box, originally-designed node parameters of a graph, and “Target” to denote the target image whose appearance we would like to reproduce. Unless otherwise specified, all results are predicted/optimized against *all* the filter node parameters of the graph.

7.5.1 Material Capture from Synthetic Images

We begin by testing MATch using synthetic photographs. In our experiment, we manually set the parameters of a graph to generate new materials. We then render these materials and use the resulting images to evaluate both MATch’s direct prediction networks and the optimization. Figure 7-4 displays the results for 5 different material graphs, comparing

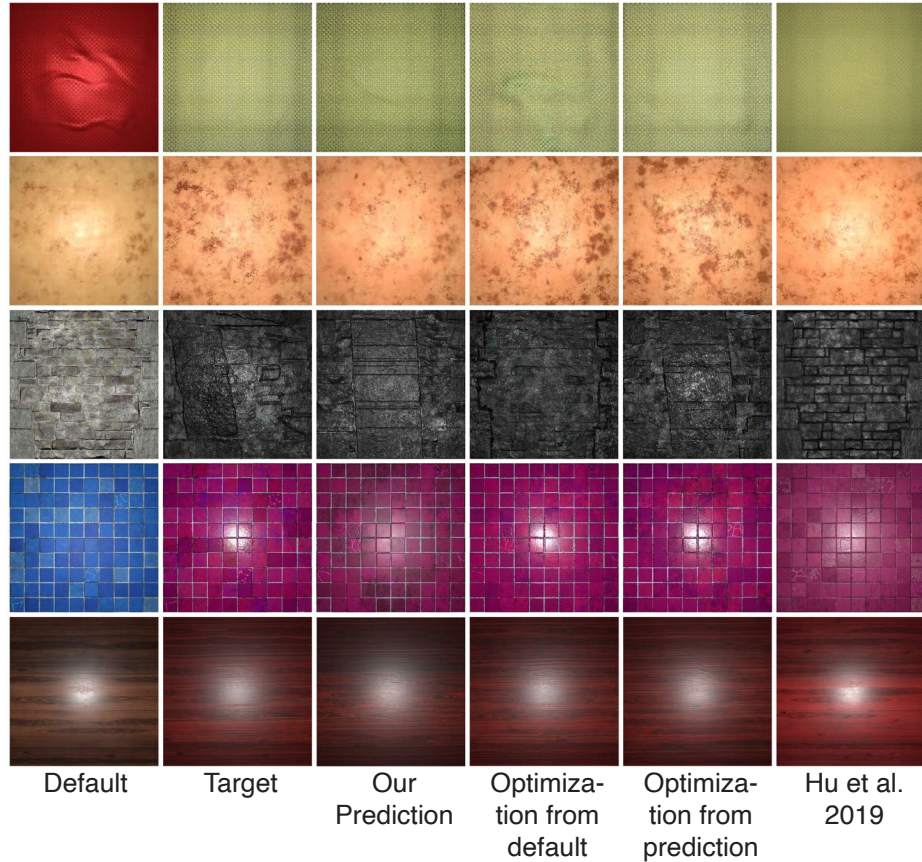


Figure 7-4: Prediction and optimization results for synthetic materials. For a majority of synthetic examples, optimization alone is sufficient to reproduce the target materials. Compared to Hu et al. [113] (our reimplementation on our data and SVBRDF model), our network prediction produces more accurate color, fine features, and glossiness with the help of rendering loss. The prediction result is further improved by the optimization step to precisely match the target material.

optimization only, prediction only, and prediction plus optimization. For most cases, optimization alone (starting from default parameters) closely matches the target appearance. Direct prediction generally provides a rough match but often misses details that optimization can recover. However, for challenging examples like the black bricks, optimization from predicted parameters improves reproduction quality compared to starting from default parameters. It’s worth noting that during both optimization and prediction, we intentionally use *different* noise patterns in the rendered images, which can be seen in the details. Nevertheless, our style-based loss effectively handles this discrepancy in basic structures to accurately reproduce the overall material appearance.

7.5.2 Material Capture from Real Data

Optimization for Real-world Materials In Figure 7-5, we evaluate MATch on 30 real photographs with 27 images captured by us and 3 images from the dataset of Aittala et al. [5]. In the supplementary material, we visualize additional real-world examples (44 in total) and the optimized SVBRDF maps.

We use the prediction network to pick the material graphs, but use the default parameters as initialization to demonstrate the capability of optimization. We note that being able to reproduce high-quality materials using just optimization allows users to quickly try out manually picked material graphs and skip the time-consuming network training step. Figure 7-5 (top) shows that we are able to reproduce a wide variety of materials starting from the same graph and initialization. This indicates the diversity of materials that can be generated by manipulating all the parameters in large-scale graphs, as well as the ability of our optimization-based approach to converge to a good solution.

In Figure 7-5 (middle), we show that when the chosen graphs are designed for a material type quite different from the targets, they are often expressive enough to represent the target materials (and the ability of the optimization to robustly discover these matches).

Prediction plus Optimization for Real-world Materials While we find that optimization alone is sufficient for many examples, the prediction network generally provides a better initial guess of the node parameters, which accelerates the convergence of optimization and improves the final results. We demonstrate both cases in Figure 7-6 on real-world materials. On average, the predicted parameters reduce the required iterations for convergence by more than half.

7.5.3 Comparisons with Previous Work

Comparison to Hu et al. In Figure 7-4 and Figure 7-7, we compare our results to the state-of-the-art procedural material capture method of Hu et al. [113]. Their original work uses a different set of material graphs, assumes Lambertian BRDFs captured under natural illumination, and only predicts the exposed parameters of the graph. For fair comparison,

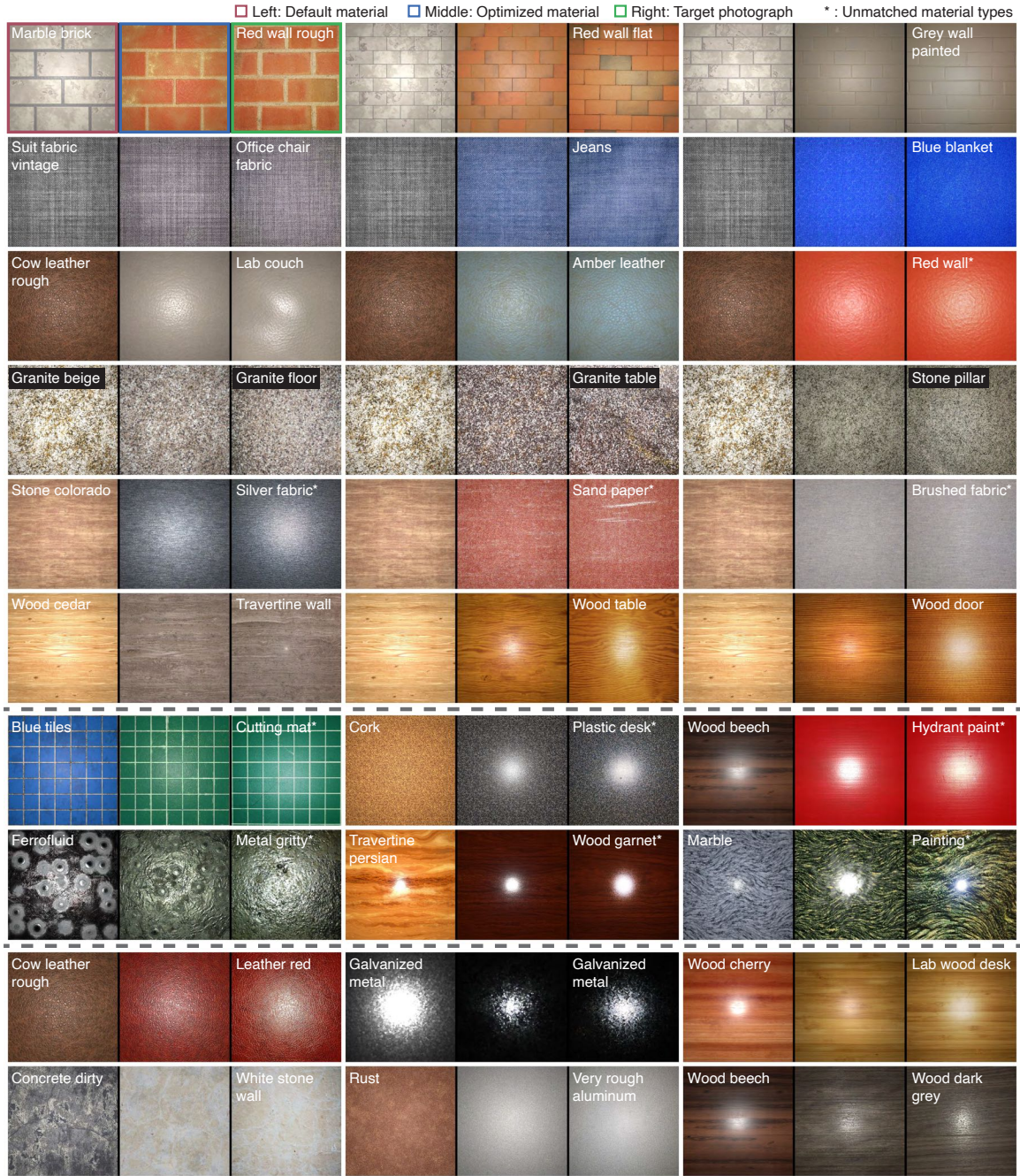


Figure 7-5: Optimization results for real-world materials. **Top:** At each row, we show multiple real-world materials matched by one single procedural material graph using the same default initial parameters. The * symbol marks the examples that are optimized against a target material of a different material type. **Middle:** Additional examples on unmatched material and graphs. **Bottom:** Additional examples on matched material and graphs.

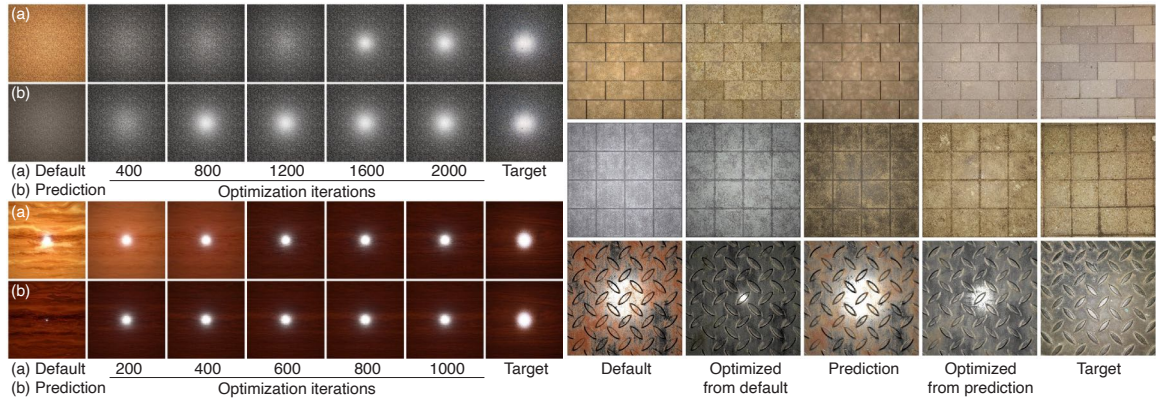
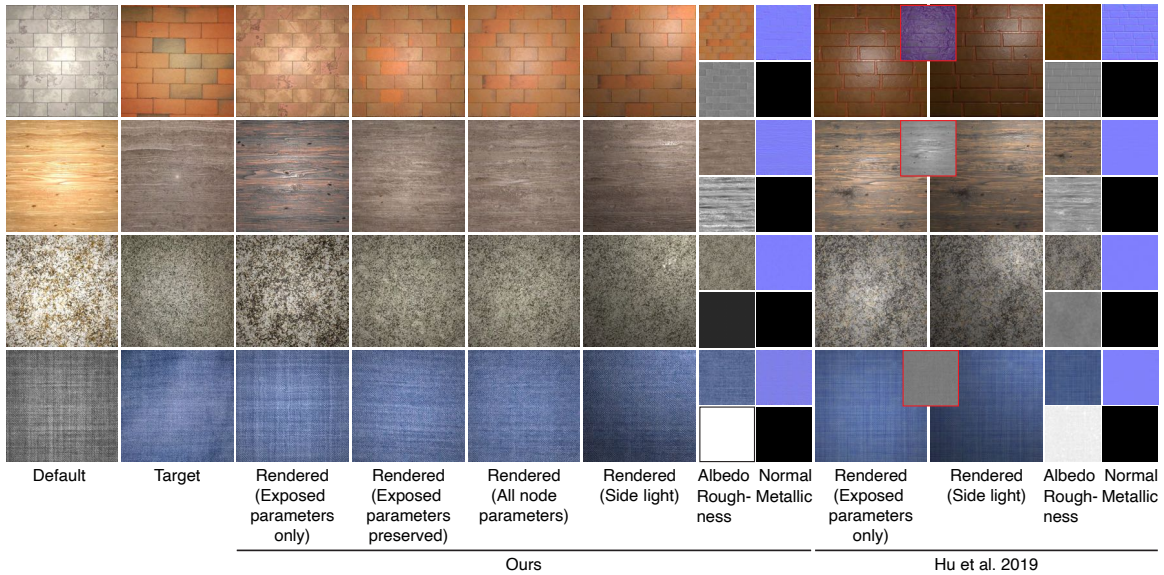


Figure 7-6: Prediction plus optimization results for real-world materials. **Left:** Speed of convergence comparison between optimizing default and predicted node parameters. **Right:** Comparison of optimization results on challenging target materials. Optimizing from the default parameters stuck in the local minimum, while optimizing from the predicted parameters produces a significantly closer reproduction.



□ Prediction by Hu et al.'s network when the graph consists of highly-correlated exposed parameters (color here).

Figure 7-7: Comparison of our method to Hu et al. [113] network prediction (without non-differentiable style transfer step) for real-world captured materials. For our results, we show optimization over exposed parameter only, exposed parameters with the rest of independent node parameters, and all node parameters. Our method overall produces a better match through optimizing a significantly larger design space and iteratively update the appearance to match the target. In contrast, the prediction by Hu et al. network is one-shot and can't not be iterative refined through back-propagation.

we re-trained their AlexNet-based models on our data (graphs, material model, and illumination). Following their paper, we restrict their prediction to the exposed graph parameters (and all parameters if no exposed parameters are defined). As can be seen on the synthetic tests in Figure 7-4, our direct prediction is more accurate than the result from Hu et al., despite that our parameter prediction networks are more compact than theirs. We attribute these improvements to the use of the image-space perceptual loss in training the prediction networks, which is possible only because of our differentiable material graphs. Moreover, our optimization-based approach further improves over our direct prediction results.

These results are also consistent with the real-world captures shown in Figure 7-7. Here, we demonstrate that different variants of our optimization scheme, where we optimize different combinations of full or exposed parameters all outperform Hu et al. During experimentation, we also find their network has a hard time disambiguating correlations between parameters due to the lack of knowledge of the material graph. For example, a graph consists of two exposed parameters both control the color of the diffuse albedo (one exposed parameter for RGB manipulation, the other for HSV manipulation), their network can get conflated, resulting in poor color reproduction (see Figure 7-7 red insets). This forces users to manually clean up the exposed parameters and prevents expanding their network to full parameter prediction for a majority of the procedural graphs.

Comparison with Single-image Per-pixel Capture In Figure 7-8 compares MATch to the state-of-the-art deep learning-based single-image SVBRDF capture methods of Deschaintre et al. [60] (based on direct SVBRDF prediction) and Gao et al. [80] (based on optimization in an autoencoder latent space). These methods predict the image-based material maps that are well aligned to the target photo. However, because of the unconstrained nature of the problem, they suffer from noticeable artifacts such as blurring, color fringing, and degraded fidelity for novel light positions. These methods are also limited to the resolution and appearance of these results. By capturing into a procedural model, our method produces higher-quality materials that can be edited, tiled, and synthesized at arbitrary resolutions. This does come at the cost of not matching the target texture patterns exactly, but this is often not important in many material capture scenarios where users would want

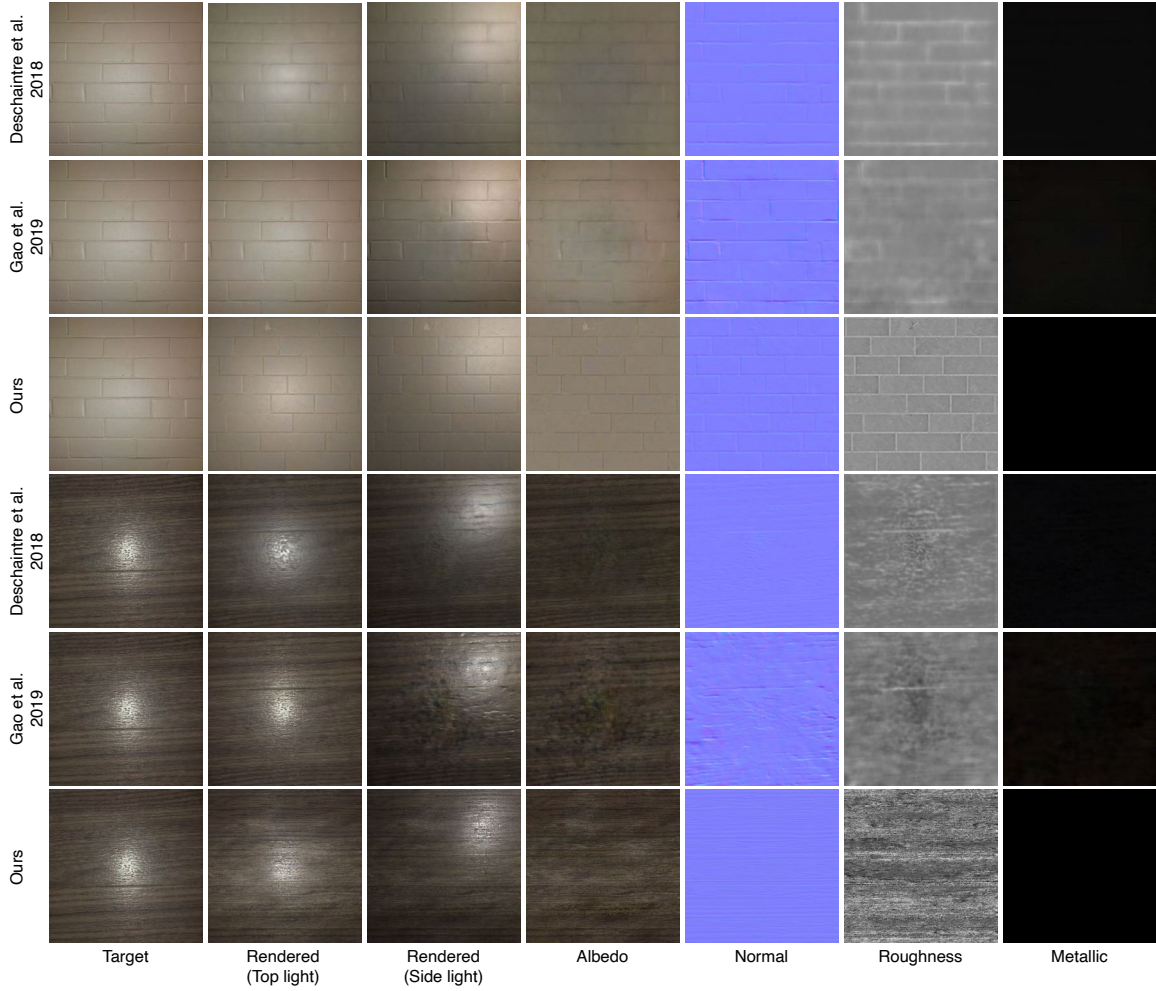


Figure 7-8: Comparison of our method to single-image per-pixel SVBRDF capture methods of Deschaintre et al. [60] and Gao et al. [80] illustrate the key difference to our method. While the per-pixel methods are able to produce texture patterns more closely aligned to the exact target photo, they suffer from various artifacts such as blurring and color fringing. Our method, predicting node graph parameters instead of pixels directly, is much more robust to these problems, though of course at the cost of not matching the target texture patterns exactly.

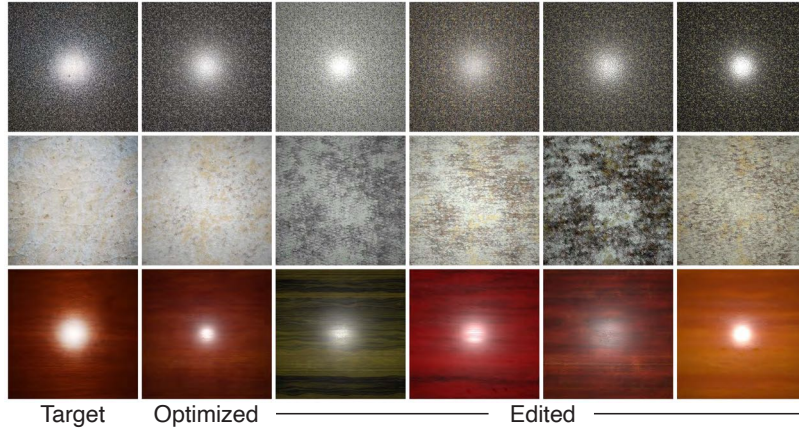


Figure 7-9: Material editing. In each row, we show an optimized real-world material and variants generated by manually adjusting the optimized node parameters. Note that all the variants are generated from the same group of noise patterns, yet display a wide range of appearances.

to capture the general appearance of the material, but not the exact structure of the input image.

7.5.4 Applications

Material Editing A key advantage of procedural materials is their editability. Starting with the optimized results, artists can manually fine-tune the parameters to reproduce the exact material appearance they would like or sample around the optimized parameters to generate a family of similar textures. We demonstrate this application in Figure 7-9, where we adjusted the node parameters of the optimized materials to create four material variants for each example.

High-resolution Material Synthesis Procedural materials are independent of resolution: once the parameters are optimized at a base resolution (512×512 in our results), we can synthesize high-resolution material maps using the same set of node parameters. In Figure 7-10, we picked 4 optimized, detail-rich materials and rendered their appearance at a resolution of 2048×2048 pixel (figures are cropped to fit the space). At higher resolutions, fine-grain details are revealed without loss of image sharpness. Such super-sampling would be very difficult to achieve with per-pixel material capture methods.

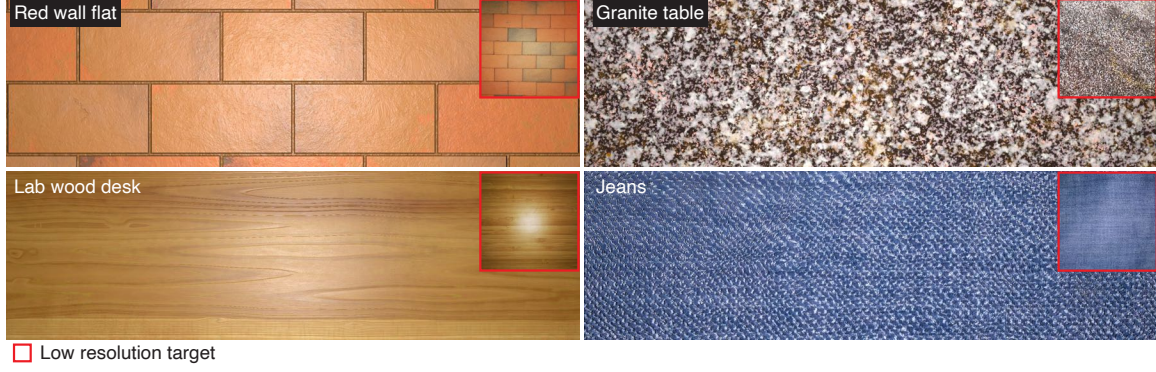


Figure 7-10: High resolution procedural materials optimized for real-world photograph inputs. Readers are encouraged to zoom in and examine details.

Exposed Parameter Translation Procedural materials are not always created with labeled subgraphs and exposed high-level parameters. These graphs are challenging to reuse because artists have to deduce the purpose of low-level nodes and manually modify their parameters. However, many other graphs within the same material category may have well-defined exposed parameters. We can use a straightforward yet effective method for distilling and translating continuous exposed parameters from a reference graph to a target graph. Our method stems from a fundamental insight—the semantics of an exposed parameter are characterized by its impact on material appearance. Therefore, an exposed parameter e_R in a reference graph G_R should induce the same perceptual change in the output texture after being migrated to a target graph $G_T \neq G_R$. This premise enables us to pinpoint the node parameters in G_T that correlate the most with e_R by matching texture descriptor changes.

Let $\theta_R(e_R)$ denote the vector of continuous node parameters in the reference graph controlled by e_R . We first grid-sample e_R in G_R to compute a set of Gatys texture descriptors (the texture rendering function is denoted by $R_{G_R}(\cdot)$). Assuming that all continuous parameters have a normalized domain $[0, 1]$, we use a step size of $\Delta s = 1/N$ and compute the i -th texture descriptor $T_{G_R}^{(i)}$ using

$$T_{G_R}^{(i)} = T_G(R_{G_R}(\theta_R(e_R = i\Delta s))) \quad (i = 0, 1, \dots, N) \quad (7.8)$$

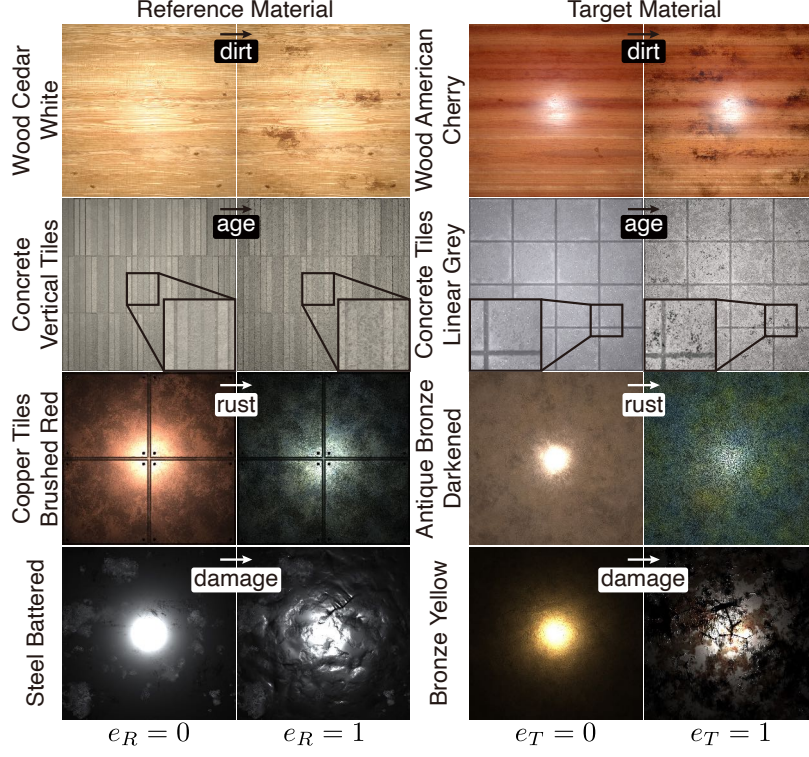


Figure 7-11: Results of exposed parameter transfer between material graphs. The labeled arrow between each image pair indicates the name of the exposed parameter. We direct readers to the Supplementary Videos for transition animations when varying exposed parameters from 0 to 1.

The appearance change in the reference graph from $e_R = 0$ to $e_R = i\Delta s$ is then derived by

$$\Delta T_{G_R}^{(i)} = T_{G_R}^{(i)} - T_{G_R}^{(0)} \quad (7.9)$$

Our method transfers these texture descriptor changes to G_T through node parameter optimization. Here, we reuse the notations above to define θ_G and $R_{G_T}(\cdot)$ for G_T . Additionally, θ_G^* denotes G_T 's default parameter values. The following problems are solved to match the impact of e_R in G_R at each sampled value ($i = 0, 1, \dots, N$):

$$\hat{\theta}_G^{(i)} = \arg \min_{\theta_G} \left(\left\| T_G(R_{G_T}(\theta_G)) - T_G(R_{G_T}(\theta_G^*)) - s\Delta T_{G_R}^{(i)} \right\| + \mathcal{L}_{\text{reg}}^{(i)} \right) \quad (7.10)$$

where s is a hyperparameter amplifying (or attenuating) the transferred texture descriptor difference. $\mathcal{L}_{\text{reg}}^{(i)}$ is a regularization term that compares θ_G against the previous best solution

$\hat{\theta}_G^{(i-1)}$ to encourage a smooth transition in material appearance

$$\begin{aligned}\mathcal{L}_{\text{reg}}^{(i)}(\theta_G) &= 0 & (i = 0) \\ \mathcal{L}_{\text{reg}}^{(i)}(\theta_G) &= \beta \left\| R_{G_T}(\theta_G) - R_{G_T}(\hat{\theta}_G^{(i-1)}) \right\| \\ &\quad + \gamma \left\| T_G(R_{G_T}(\theta_G)) - T_G(R_{G_T}(\hat{\theta}_G^{(i-1)})) \right\| & (i > 0)\end{aligned}\tag{7.11}$$

where β and γ are the weights for texture image and texture descriptor losses, respectively.

Finally, we select the top K node parameters in G_T based on the magnitude of texture descriptor difference (between $\hat{\theta}_G^{(N)}$ and θ_G^*) contributed by each node parameter. This step discerns the essential node parameters underlying the transferred exposed parameter e_T . For each selected parameter k , we extract its optimized value sequence from $\hat{\theta}_G^{(i)}$ ($i = 0, 1, \dots, N$) and fit an M -th degree polynomial function $\mathcal{P}_k(e_T) = \sum_{m=0}^M \beta_{k,m} e_T^m$ from e_T using least squares, where $\beta_k = \{\beta_{k,0}, \dots, \beta_{k,M}\}$ is the set of polynomial coefficients for the k -th selected parameter. These polynomial functions can be easily constructed using S3D's function graphs.

Fig. 7-11 illustrates the translation of various exposed parameters from a reference material graph to a target material graph. These exposed parameters alter details in albedo (R1), normal (R2), color tone (R3), and material structure (R4), showing case the robustness of the proposed scheme in transferring different types of exposed parameters.

7.5.5 Limitations & Discussion

MATch does not optimize the graph structure, thus its success ultimately relies on the expressiveness of the chosen graph. Consequently, it performs poorly or fails when unmodelled patterns are present in the input or the complexity of the input pattern exceeds the granularity of the deployed filters. For example, the optimized wood example in Figure 7-5 does not precisely reproduce the dot-like grooves in the normal map because the normal sub-graph cannot achieve this granularity, instead a similar visual effect is approximated through a noisy roughness map. A very similar example can also be found in Figure 7-12 second row. Figure 7-12 first row visualizes a more exaggerated example, where the substrate behind the input metal grate is completely unmodelled, and the optimization fails.

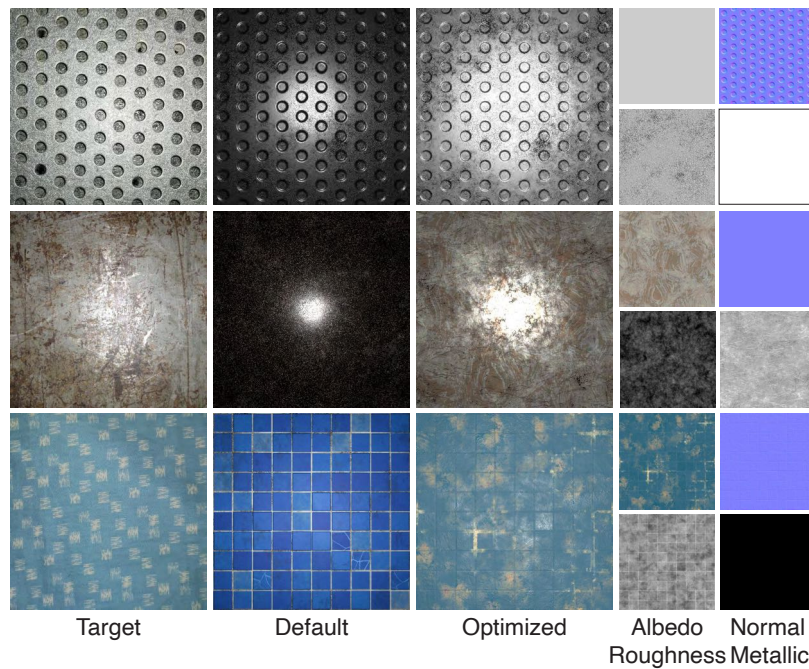


Figure 7-12: Failure cases due to unmodeled objects (substrate behind the metal grate), limited expressiveness of overly simple graph structures (subgraphs that produce metallic of the metal grate, normal of the scratched metal, and the bed sheet example), and inability to optimize the generator nodes to match the input pattern (albedo of the scratched metal, the bed sheet example).

MATch currently does not optimize the generator nodes, thus unmatched input noise patterns result in poor reproduction. The optimized “Metal gritty” in Figure 7-5 is an example of this, where the ferrofluid pattern persists after the optimization despite the doughnut-like color gradient gets removed. More subtle examples can be found in “Amber leather” and “Red wall” of Figure 7-5. Figure 7-12 (third row) demonstrates a more exaggerated example, where a tile graph is used to reproduce a bed sheet with scattered square patterns. Because the tile pattern is an input of the generator node, MATch can’t relocate individual tiles, which remain visible in the final result.

Finally, MATch currently does not support full-fledged Fx-map Substance node and derived nodes such as tile generator and splatter. Thus, it is not able to convert graphs that use these nodes as intermediate nodes (only as generators). This makes conversion of certain types of materials such as grass difficult. This is a limitation of our current implementation, which does not occur in black-box approaches (e.g. Hu et al. [113]) since differentiability is not a requirement.

In a work led by my co-author, we have developed an improved version of MATch and Diffmat v2 that addresses many of discussed limitations [159]. In particular, this new pipeline is able to translate generator nodes, including full-fledged Fx-map Substance node, and co-optimize discrete parameters to capture a considerably broader array of materials, encompassing those with highly regular or stochastic geometries. We refer readers to the paper for more details.

Chapter 8

3D Printing for Multispectral Fine Art Reproduction

In preceding chapters, our main emphasis has been on addressing the visual Turing test within the domain of virtual and augmented reality. Nonetheless, the scope of the visual Turing test can expand to encompass the physical realm. This entails questioning whether we can replicate a physical object in a manner that is indistinguishable from the original. This chapter delves into this inquiry with a particular focus on a captivating subject: fine arts. Fine art is a crucial element of human culture, but our reliance on museums to exhibit original paintings and sculpture inherently limits access and leaves those precious originals vulnerable to deterioration and damage. The availability of high-quality facsimiles has the potential to not only broaden exposure but also leave the originals unharmed, in such scenarios as restoration practice, conservatory studies and education in museums [71]. Paintings are brilliant examples of fine art artifacts and natural first candidates for an appearance reproduction effort.

In this chapter, we present a novel workflow for reproducing paintings with accurate spectral reflectance utilizing a multi-material 3D printer. Leveraging the unique capabilities of 3D printers to vertically combine highly concentrated inks across numerous layers, we show the spectral gamut of a designated set of inks can be greatly expanded. Our approach employs a data-driven methodology to predict the spectrum of a printed ink stack and optimize the layout of these stacks to best match a target spectrum. Additionally, we

introduce a spectral vector error diffusion algorithm, which combines color contoning [14] and halftoning techniques to address layout discretization and color quantization challenges simultaneously, with precision and efficiency. The proposed method surpasses existing models for spectral prediction and layout optimization. We showcase the successful reproduction of several real paintings and historically significant pigments using our prototype implementation, which employs 10 custom inks with varying spectra and a resin-based 3D printer.

8.1 Motivation

Traditional color reproduction of paintings is carried out using 2D printers. However, they have severe shortcomings for high-fidelity color reproduction as a consequence of two interrelated problems. First, their color gamut is limited because of their fixed set of inks, which usually includes only cyan, magenta, yellow, and black (CMYK). Although the limited gamut can be overcome with a larger number and a wider variety of inks, only a limited amount of ink can be deposited within a given region. Exceeding the so-called total ink limit results in deteriorated image quality, ink blotting, or mechanical malfunction of the printer [13]. Second, 2D printers predominantly use *colorimetric color reproduction*, in which the color matching is carried out under a reference illuminant and assumes a set of standard observers. This can lead to *metamerism*, a well-known problem in color reproduction wherein a good reproduction is obtained under one light source, but not under another [256].

The recent breakthrough in high-resolution, multi-material 3D printing carries the potential for groundbreaking advances in fine art reproduction. While 3D printers are powerful devices for fabricating objects with custom and complex geometry, they can also create multi-material composites that result in new appearance properties. Among them, the spectral color is often considered as the most important appearance attribute for most forms of paintings. The spectral reflectance of a surface encodes the complete information about its color. Therefore, the *spectral color reproduction* is invariant to the color of the light source under which the reproduction is observed.

Good spectral reproduction requires a large spectral gamut. We, therefore, equip our 3D printer with 10 inks and take advantage of its ability to combine highly concentrated inks with a large number of thin layers, inspired by the recently proposed color contoning technique [14]. In doing so, we expand the gamut realized by our set of inks significantly. The use of high-concentration inks in contoning, however, comes at the cost of color quantization artifacts. Therefore, we introduce a novel spectral vector error diffusion halftoning [138] that uses the ink stacks of all possible layer combinations as halftone primaries with an unprecedented efficiency. This turns the spectral vector error diffusion of potentially billions of primaries into a viable approach, with significantly alleviated artifacts (due to an insufficient number of primaries). Furthermore, the halftoning resolves the problems associated with the discrete nature of ink layers, making even a simple layer rounding reliable in practice.

Accurate modeling of the complex light transport across ink layers, in the presence of printing inaccuracies, is a challenging task if pursued using physical modeling methods. We instead take a data-driven approach, modeling the bidirectional mapping between the spectral reflectance and the ink stack layout using artificial neural networks, which are learned from the actual printed ink stacks. We produce superior performance in both reproduction quality and run-time, compared to previous physically-based or data-driven approaches. We validate and evaluate our approach using historically important pigments in painting and a number of real paintings.

Our main contributions are:

- A complete physical reproduction framework, comprising both acquisition and fabrication, tailored for the re-creation of paintings;
- A data-driven spectral color prediction model based on neural networks that outperforms existing methods for both predicting the spectrum of a stack of ink layers and finding an optimal ink stack for a target spectrum;
- A highly efficient spectral vector error diffusion method that combines ideas from contoning (combining thin layers of inks) and halftoning (employing spatial modulation) to achieve smooth yet accurate color reproduction;

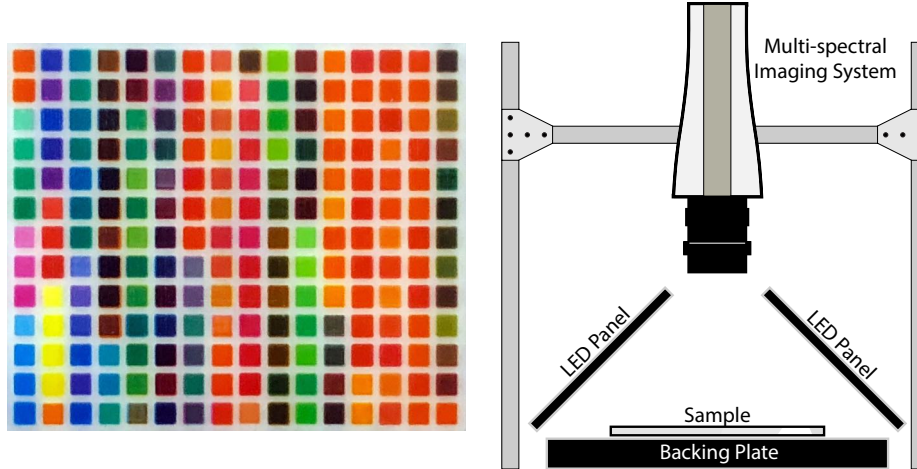


Figure 8-1: *Left:* A photograph of one of our printed sample patches. Each color square is $1\text{ mm} \times 1\text{ mm}$. *Right:* Our spectral acquisition setup.

- A dataset, which we will make it publicly available, of 20,878 contone ink stack spectra and layouts, spectrally captured oil paintings, together with their optimized layouts using our ink library, and photographs of our printed reproductions under multiple illuminations.

8.2 System Overview

Our workflow for multi-spectral color reproduction of painting consists of two major components. First, a multi-spectral acquisition setup is constructed to accurately capture high-dynamic-range scanning of fine art paintings. We use MultiFab [234], a laboratory-scale, multi-material inkjet 3D printer with a photopolymer printing process similar to commercial printers developed by Stratasys [238] and 3D Systems [242]. Our printer provides greater flexibility on the selection of printing materials.

8.3 Hardware & Fabrication & Ink Set

We use a spectral imaging setup as shown in Fig 8-1 to capture both calibration samples and paintings to reproduce. At top, the Nuance FX multispectral imaging system is coupled with a Coastal Optical 60 mm 1:4 UV-VIS-IR APO macro lens for multispectral image

capture. Within the camera, a tunable liquid crystal filter can be dynamically adjusted to transmit one narrow range of wavelengths at a time, while the sensor captures a 1392×1040 -pixel monochromatic image of that spectral band. We perform high-dynamic-range (HDR) multi-spectral capture [89] within the spectral range of 420–720 nm, at 10 nm steps, yielding a 31-dimensional spectrum per pixel. Note that an alternative to the tuned LCD filter would be to use advanced hyperspectral recovery algorithms that can reconstruct the spectra using wide-band RGB channels [15, 47].

At the bottom of the setup, we illuminate the sample with two 12-watt ROSCO Daylight (5800 K) 12×12 inch (30×30 cm) LED LitePads at 45 degrees to horizontal, centered symmetrically around the multi-spectral camera. The light emitted by each panel is diffused by a ROSCO #3029 diffusing sheet to maximize the lighting uniformity over the camera’s field of view. A black acrylic slab under the camera serves as a backing board to minimize reflection and indirect illumination. The entire setup is fully sealed in an enclosure to eliminate environmental illumination.

We correct radial distortion by photographing checkerboards and solving for radial distortion parameters using the MATLAB Camera Calibration Toolbox [29]. We image X-Rite white balance reference and color-checkers to calibrate and compensate the light fall-off (vignetting), lighting non-uniformity, and spatial non-uniformity of pixel gain [24]. We normalize the spectrum of the illumination by dividing every multi-spectral measurement by the multi-spectral measurement of a calibrated X-Rite white reference target and multiplying by the reference’s spectrum.

We use MultiFab [234], a laboratory-scale, multi-material inkjet 3D printer with a photopolymer printing process similar to commercial printers developed by Stratasys [238] and 3D Systems [242]. Our printer provides greater flexibility on the selection of printing materials. Its spatial resolution (xy -resolution) and vertical resolution (z -resolution) are $35 \mu\text{m}$ and $11.25 \mu\text{m}$ respectively. Our printer has five channels, enabling us to print with up to 15 inks using three passes of the printer. Our inks comprise a UV curable clear photopolymer carrier mixed with commercially available color pigments. We prepare the inks by first mixing the carrier and pigment and then milling the inks using a bead mill.

We have developed a set of 11 inks, including 9 color inks, a low-concentration white

ink, and a high-concentration white ink (Figure 8-2). The color inks include the standard cyan, magenta, yellow, and black (CMYK) inks. Following the n -ink 2D printers, we also add red, green, blue, orange, and violet inks to our printer’s palette. We emphasize that, though the orange at our disposal shares a similar response with the red, it is more reflective in long wavelengths and has a narrower ($\sim 30\text{--}40\text{ nm}$) absorption band at short wavelengths. Given the importance and dominant use of orange pigment in art history and its reputation as difficult to be reproduced [198], we decided to include it. Inspired by the *veiling* technique used by painters, we add a low-concentration white ink, which we call transparent white, on top of the ink stack. The transparent white “veils” the highly saturated color inks beneath it and enables low-saturation colors. The high-concentration white ink, which we call opaque white, is the background of our prints and plays the same role as the paper in 2D printing. The transparent and opaque white use different concentrations of the same pigment.

The color ink concentrations are optimized for saturation, and are bounded by the hardware constraints of the printing process. In principle, we desire highly saturated colors with minimal ink-stack thickness. We start by finding, for each individual ink, the maximum concentration that the printer can properly print. Exceeding these concentrations, causes unstable jetting, printhead clogging and incomplete UV curing. Then, we experimentally determine the maximum number of layers required to obtain saturated colors at their highest concentration. This number depends on the inks and for a majority of the more transparent inks is around 30 layers. We adjust the concentration of the remaining inks such that they all saturate at 30 layers. For example, for some inks, such as violet and black, the color of the resulting stack saturates more quickly. Therefore, we decrease the concentration such that they saturate at 30 layers.

We opt for the highest concentration because when ink concentration increases, the number of layers required to obtain saturated colors decreases. A lower number of layers is preferred because the mechanical and optical dot gain, and therefore blurring, increase with the number of layers [14]. On the other hand, highly concentrated inks cause pronounced quantization artifacts. We partially address this challenge by incorporating veiling top layers using the transparent white, and therefore the concentration of that ink is tuned to allow

Color	Ink product	%
Transparent White	Penncolor	1
Cyan	RJA	1
Magenta	RJA	1
Green	Lansco 3136	1
Blue	Keystone 3R	0.5
Orange	BASF D2905	1
Yellow	BASF D1155	1
Red	Lansco 1722	1
Violet	Lansco 1233	0.15
Black	RJA	0.4
Opaque White	Penncolor	5

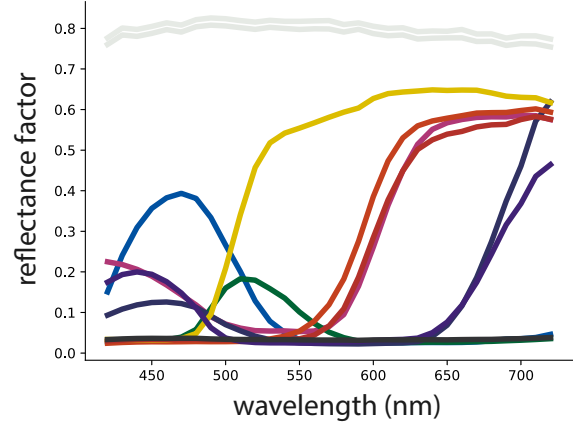


Figure 8-2: Our ink library. *Left*: The pigments and concentrations used in our inks, arranged in the order from top layer to bottom. *Right*: The spectra of saturated color of our inks printed with 30 layers. The color of each line corresponds to the color of an ink. The higher white line represents the Transparent White and the lower represents the Opaque White.

the colors beneath to remain visible, although it is handled in a more principled way by spectral vector error diffusion (§8.4.3). Finally, we set the concentration of the opaque white to be as high as possible. This prevents subsurface scattering and results in a brighter white substrate.

We prepare a dataset of contone ink-stacks to train our model. Throughout this work, the order of inks in each stack is fixed: a section of 30 layers consisting of any inks, arranged from the most transparent to the least transparent starting from the top (in the order shown in the table of Figure 8-2), followed by additional 20 opaque white layers at the bottom. Although we fix the order of inks in the ink stacks, the number of layers and that of inks lead to about 0.8 billion possible layer layouts. To efficiently sample from all layer-layouts to build our training dataset, we introduce the following heuristic sampling rules to maximize the coverage of the spectral gamut while keeping the number of samples manageable. In the following, the “color” inks include the transparent white as well, i.e., denote the first ten inks in the table of Figure 8-2, while still excluding the opaque white ink.

- Due to the smooth change of spectral properties of traditional inks, combinations of a large number of different inks does not result in large spectral variety [214]. We

thus limit the maximum number of different color inks in a stack to 5.

- We limit the number of color ink layers depending on the number of different inks used in the stack. Specifically, when using 3, 4, or 5 color inks, we set the maximum number of color layers to 10, 15, or 20, respectively. The remaining layers are complemented with additional opaque white layers at the bottom.
- If an ink stack contains 2 or 3 “dark” colors, we further limit the number of color layers to 8 or 4, respectively. Dark colors are the colors that have high absorption and/or scattering, which in our case are blue, violet, and black inks.
- For any layer-layout consisting of more than 2 color inks, we enforce the layer increment step of an ink to be 2.

We obtain 20,878 valid layer layouts following these rules. Each ink stack occupies a $1\text{ mm} \times 1\text{ mm}$ square in the calibration print. We aggregate 13×16 ink stacks into one calibration patch, resulting in the total of 101 patches (Figure 8-1, left). Within each patch, the neighboring ink stacks are separated by a 0.3 mm wall made of transparent material to prevent pixel cross-talk. We favor transparent wall over black pigment wall for both better curability and free of color contamination, with the black backing further minimizes reflection, resulting negligible sub-surface scattering. To ensure measurement accuracy, the spectral reflectance of each ink stack is calculated by averaging the measurement over the $0.5\text{ mm} \times 0.5\text{ mm}$ central region of each square.

Based on the maximum operation area of our printer, we print 4×4 patches at each printing iteration. Printing proceeds in 3 passes, and is done upside down—i.e., the topmost layers are printed first, in order to achieve a diffuse surface finish. In the first pass, the top white layers and the transparent wall are printed and cured. In the next two passes, the first 5 color layers and the next 5 color layers are sequentially printed. Completion of all three passes takes approximately 3.5 hours. The HDR multi-spectral acquisition takes 5 minutes to complete measurement for one calibration patch.

8.4 Data-driven Spectral Prediction & Layout Planning

8.4.1 Spectral Prediction Model

Spectral reflectance is represented as a 31-dimensional real vector, each element denoting the reflectance at a discretized wavelength. A layout is represented as a 11-dimensional non-negative integer vector, encoding the numbers of layers used for each ink. We learn the spectral prediction function $\mathbf{F} : \mathbb{Z}_+^{11} \rightarrow \mathbb{R}_+^{31}$, which maps a layout $\boldsymbol{\xi}$ to its spectral reflectance $\boldsymbol{\psi}$, from our ink-stack dataset. Given the low dimensionality of our problem, we model the function using a straightforward fully-connected feed-forward neural network (also known as a multi-layer perceptron) [54]. Our model includes 4 hidden layers, each with 300 neurons (hidden units), forming a network that is wider than it is deep. Both the hidden layers and the final output layer use the rectified linear unit (ReLU) as a nonlinear activation function.

Spectral Loss

We optimize the network for the minimal spectral error, which is defined as a Euclidean distance between the prediction and the measurement (scaled by the square root of the number of wavelength bands, in accordance with the practice in colorimetry literature [126]):

$$E_{\text{spec}}(\boldsymbol{\xi}, \boldsymbol{\psi}^*) = \frac{1}{\sqrt{31}} \|\mathbf{F}(\boldsymbol{\xi}) - \boldsymbol{\psi}^*\|_2. \quad (8.1)$$

In our implementation, we normalized the input ink layout vector by the total number of layers (i.e., 30) such that each layout sums to one.

8.4.2 Layout Prediction Model

Ultimately, we are interested in the inverse function of $\mathbf{F}$, through which we retrieve the optimal layout for a given spectral reflectance. Given that $\mathbf{F}$ is a differentiable function that maps layouts to spectra, a straightforward approach for the inverse problem would be fixing $\mathbf{F}$'s parameters and running gradient descent on layouts by minimizing the spectral error. However, the distribution of spectra over layouts will likely be highly non-uniform

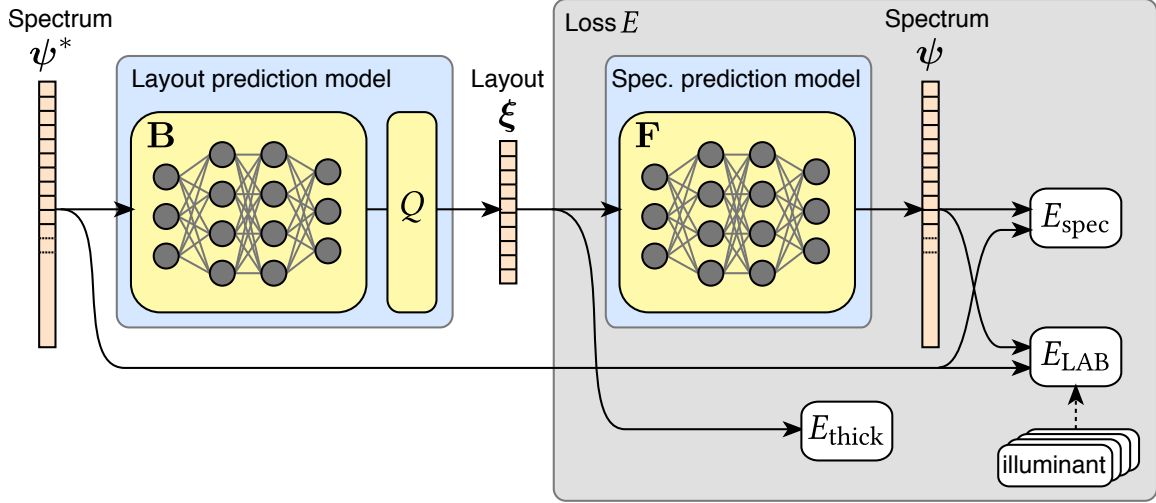


Figure 8-3: Illustration of our network structure. $\mathbf{F}$ denotes our spectral prediction model (§8.4.1), which is used to estimate the spectrum of the predicted layout by our layout prediction model $\mathbf{B}$ (§8.4.2) and provides the means to measure the quality of it. Q represents our soft quantization layer (§8.4.3). ψ denotes the spectrum and ξ the layer-layout vector. Both $\mathbf{F}$ and $\mathbf{B}$ are implemented as fully-connected feed-forward neural networks.

and multi-modal. Further, as the layout is inherently discrete, its relationship with the spectrum forms a non-convex space. In practice, we find that running gradient descent directly on a randomly initialized layout usually leads to a local minimum and results in a poor prediction.

To pick a good initial guess, reducing the risk of being trapped in a local minimum, we could train a separate network that would learn the inverse mapping of $\mathbf{F}$ and provide a reasonable initial layout estimate from a target spectrum, from which we could initiate the gradient descent through $\mathbf{F}$. However, it is difficult to measure how good one predicted layout is compared to another, since the L^2 difference between layouts does not translate directly to the spectral difference, in particular when different layouts could lead to an identical spectrum. A better alternative is to construct an encoder-decoder model that uses $\mathbf{F}$ to “decode” the predicted layout to the spectrum and evaluate the error with respect to the target input spectrum; i.e., to use the forward network $\mathbf{F}$ (decoder) to set up the loss and train the reverse mapping (encoder) through back-propagation directly [244] (Figure 8-3).

We define a layout prediction function $\mathbf{B} : \mathbb{R}_+^{31} \rightarrow \mathbb{Z}_+^{11}$ that predicts an optimal layout given a spectrum and is implemented by another fully-connected neural network. $\mathbf{B}$ consists

of 8 hidden layers, deeper than $\mathbf{F}$, as it attempts to learn a more complicated distribution. Each hidden layer consists of 160 neurons with the ReLU activation function. As for $\mathbf{F}$, the layout vector is normalized. A softmax layer is applied to the output layer to guarantee the validity of the layout estimate—nonnegative and summing to one.

Perceptual Color Loss

We first experimented with training $\mathbf{B}$ with the spectral loss (8.1) only. However, we observed that the reproduced spectral reflectance (from the predicted layout) of an out-of-gamut target spectrum varied noticeably across trained models, although all shared a similar L^2 error. Since we do not explicitly perform spectral gamut mapping nor apply any gamut constraints, the trained model was allowed to freely select any in-gamut spectrum minimizing the L^2 error. We, however, wish the network to predict the one among all solutions that also minimizes the perceptual color difference under a variety of standard illuminants and common light sources [193]. To implement this, we define a perceptual color loss over chosen illuminants:

$$E_{\text{LAB}}(\xi, \psi^*) = \sum_i \Delta E_{ab}^* (\text{LAB}_{I_i}(\mathbf{F}(\xi)), \text{LAB}_{I_i}(\psi^*)) , \quad (8.2)$$

where $\text{LAB}_{I_i}(\cdot)$ represents a series of transformations from a spectrum to CIEXYZ coordinates under the illuminant I_i with the CIE 1931 2° standard observer, and further to CIELAB color. Each I_i denotes the spectrum of a particular illuminant, and for ΔE_{ab}^* we use the CIE 1976 color difference (ΔE_{76}) in CIELAB space [256]. For this perceptual loss, we used 8 different light sources: halogen, incandescent, fluorescent (2700 K), Philips Hue LED (5000 K), ROSCO Daylight LED (5800 K) as well as the CIE standard illuminants D65, D50 and A. The ΔE_{76} is chosen for simplicity; ΔE_{94} , also differentiable, could as well be an accurate perceptual loss metric [135]. Although more accurate, ΔE_{00} is not differentiable.

We observe more stable and consistent training and preferred results with the addition of the perceptual loss (8.2). We emphasize that, for in-gamut input spectra, adding the perceptual loss will *not* misguide the network to learn metameric pairs since only the desired

spectra drive both the spectral loss and perceptual loss to zero. However, training with only the perceptual loss also yields suboptimal results, since the mapping from a low dimensional CIELAB space to a high dimensional spectral space is multi-modal. Without the measurement of spectral fidelity and under a set of limited illuminants only, the network can easily converge to a metameric spectrum that gives an equivalently low perceptual difference.

Layer Thickness Loss

In addition to minimizing the perceptual difference, we also minimize the number of color layers (color inks plus transparent white) in a layout estimate (excluding the twenty or more *opaque* bottom white layers). Babaei et al. [14] show that fewer color layers result in smaller minimum feature size and reduced optical and physical dot gain in color contoning. In both 2D- and contoning-based colorimetric reproduction, the mapping from a CIELAB color to CMYK ink intensity is multi-modal, which is a result of the trade-off between the black ink (K) and three chromatic inks (CMY). Our ink set also shares this problem, and the layout with the thickest black layer often results in the minimal total layer thickness (i.e., the number of color layers). In addition, given the significantly increased number of primaries, a target spectrum may also be reproduced by combination of different inks. Therefore, to encourage the use of a minimal number of color inks, we define the layer thickness loss:

$$E_{\text{thick}}(\xi) = \sum_{i=1}^{10} |\xi_i| = \|\xi\|_1 - |\xi_{11}|, \quad (8.3)$$

where ξ_{11} is the number of opaque white ink layers in the predicted layout.

The backward function **B** is trained to minimize the following loss over our training dataset:

$$E = E_{\text{spec}} + \alpha E_{\text{LAB}} + \gamma E_{\text{thick}}, \quad (8.4)$$

with $\alpha = 10^{-3}$ and $\gamma = 10^{-3}$.

8.4.3 Spectral Vector Error Diffusion

When using the backward model in order to print an ink layout best reproducing a given spectrum, we witness two problems which, at first glance, seem independent. First, a denormalized layout prediction of $\mathbf{B}$ has continuous values and thus has to be discretized to integer layers for actual printing. Second, because of the high concentration of our inks, a mere contone-based model would suffer from color quantization [14]. Although the use of transparent white mitigates this problem, quantization artifacts are still visible at regions with smooth color gradients. This is in essence a halftoning problem: we have a spatially-varying continuous input (from $\mathbf{B}$) that needs to be represented with a discrete ink layout at each location. Halftoning techniques simulate continuous tone images through spatial modulation of a limited number of primaries [17].

Layer Layout Discretization

A straightforward approach would be to round the continuous output to the nearest integer. Alternatively, we can enumerate all possible combinations of rounding up and down for the prediction on the thickness of each ink, which would amount to 2^{11} possibilities, and search for the best combination, in terms of spectral accuracy, in a brute-force manner.

Since the complexity of enumeration grows exponentially, it would be preferable if the network could directly generate integer-valued layouts. However, rounding is not differentiable, and thus cannot be directly used during training. latercheck the equation :

To address that, we apply a soft quantization layer to encourage the network to predict close-to-integer layouts, mitigating the influence of a posterior rounding operation on the prediction. Our soft quantization layer is placed between the spectrum prediction network $\mathbf{F}$ and the layout prediction network $\mathbf{B}$, such that the loss is calculated on (soft) rounded layout predictions. The soft quantization layer is constructed as a sum of cascaded soft unit-step functions:

$$Q(\xi_i) = \frac{1}{N_s} \sum_{k=1}^{N_s} s(N_s \cdot \xi_i - (k-1)), \quad (8.5)$$

where the number of steps equals to the number layers $N_s = 30$, and the unit-step function $s(\cdot)$ could be any sigmoid function that has a steep transition around a half and has

vanishing gradients outside the unit interval. We use a shifted logistic function:

$$s(\xi) = \sigma(\kappa \cdot (\xi - 0.5)), \quad (8.6)$$

where $\sigma(x) = (1 + \exp^{-x})^{-1}$ and κ controls the steepness of the transition, which we set $\kappa = 15$ for training our network.

Error Diffusion

Since in contoning, the only way of modulating color is to add or remove a layer, when highly-concentrated inks are used, the color space is not covered as densely as required, especially at low-saturation areas of the gamut. Our solution to the color quantization problem is to perform an error diffusion with contone stacks as primaries. This bears resemblance to spectral vector error diffusion (sVED) algorithms. However, our proposed method is significantly more efficient. Unlike sVED algorithms that for each pixel look for the nearest neighbor among all primaries, which in our case spans 0.8 billion 31-dimensional points, we simply round the continuous layout predicted by $\mathbf{B}$ and feed the rounded layout to $\mathbf{F}$. The resulting spectrum is subtracted from the target spectrum to obtain the spectral error vector. The spectral error is then carried over to the neighborhood pixels, wavelength by wavelength, according to the weights specified by the Floyd-Steinberg [77] error diffusion kernel. The subsequent prediction for the neighborhood pixels then takes into account the carried-over spectral error. To further improve the computational efficiency, we parallelized the error diffusion following the optimal scheduling proposed by Metaxas [189].

The advantage of this approach is twofold. First, the discretization of the continuous layout from the backward model can be a simple rounding, which is orders of magnitude faster than a search among the huge space of possible ink stacks. More importantly, the color quantization artifacts are removed through spatial mixing of a large number of contone-stack primaries. Unlike with RGB or CMYK dithering, the extremely large pool of primaries makes halftoning artifacts virtually invisible, since for any input spectrum there exists a very similar “primary.”

8.4.4 Training Protocol

We randomly split (and fixed) our spectral ink-stack dataset into a training set of 18,878 samples and a test set of 2,000 samples. We verified that the number of layers for each color ink in the test set follows the distribution in the entire dataset.

We train our spectral prediction network $\mathbf{F}$ using the spectral loss (8.1) only. After the training of $\mathbf{F}$ is finished, we train the layout prediction network $\mathbf{B}$ using the combined loss (8.4), with the weights of the trained $\mathbf{F}$ fixed. During the reproduction, a target multi-spectral image is fed to $\mathbf{B}$ to yield an optimal layout prediction, which subsequently is quantized by the spectral error diffusion. The resulting (denormalized) layout map consists of 11-dimensional vectors, each of which dictates how many layers of each ink have to be deposited in order.

Both networks are trained using Adam [141], a stochastic optimization algorithm. We use an initial learning rate of 10^{-3} for $\mathbf{F}$ and 10^{-4} for $\mathbf{B}$, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for both models. Training samples are drawn randomly from the training dataset. Both models are trained for 500,000 iterations with batch size of 64. A learning rate decay of 0.1 is employed every 50,000 iterations. Weights are regularized by penalizing their L^2 norm weighted by 10^{-5} .

8.5 Experiments & Results

We evaluate the performance of our method both quantitatively and qualitatively. In addition to individual spectra that have been historically considered important to reproduce, we validate our full reproduction pipeline with a number of real paintings, which are intentionally painted to maximize the diversity in color, and include challenging textures and smooth gradients.

8.5.1 Model Evaluation

In this section, we evaluate our spectral and layout prediction models, and spectral vector error diffusion. We also provide the reproduction quality of our model tested on historically

Models	E_{spec}		ΔE_{00} (D65)		ΔE_{00} (TL84)		ΔE_{00} (A)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
	<i>Median</i>	<i>Max</i>	<i>Median</i>	<i>Max</i>	<i>Median</i>	<i>Max</i>	<i>Median</i>	<i>Max</i>
Contoning	6.91	2.82	9.08	5.67	9.13	5.53	9.11	5.20
	<i>5.48</i>	<i>61.73</i>	<i>8.56</i>	<i>41.49</i>	<i>8.56</i>	<i>41.73</i>	<i>8.65</i>	<i>40.77</i>
1C-KM	4.90	2.82	9.89	5.67	9.02	5.53	8.26	5.20
	<i>4.22</i>	<i>25.73</i>	<i>9.12</i>	<i>43.37</i>	<i>8.14</i>	<i>34.37</i>	<i>7.31</i>	<i>35.33</i>
2C-KM	3.83	3.09	6.64	4.03	6.13	3.96	5.72	3.93
	<i>2.99</i>	<i>34.29</i>	<i>5.85</i>	<i>39.79</i>	<i>5.24</i>	<i>40.10</i>	<i>4.80</i>	<i>40.80</i>
4-Flux	1.99	1.60	3.57	2.84	3.70	2.85	3.54	2.70
	<i>1.60</i>	<i>27.40</i>	<i>2.92</i>	<i>30.45</i>	<i>3.03</i>	<i>30.87</i>	<i>2.93</i>	<i>27.46</i>
Ours	1.44	1.06	2.50	1.58	2.38	1.58	2.20	1.69
	<i>1.16</i>	<i>10.95</i>	<i>2.19</i>	<i>11.79</i>	<i>1.97</i>	<i>11.36</i>	<i>1.72</i>	<i>12.24</i>

Table 8.1: The accuracy of different spectral reflectance prediction models, in terms of the spectral error E_{spec} (Eq. 8.1) and the CIEDE2000 (ΔE_{00}) [174] under three illuminants. We report the mean, standard deviation, median, and maximum. The lowest errors are denoted in boldface.

important colors.

Spectral Reflectance Prediction Model

We compare our spectral reflectance prediction model with several physically-based and data-driven models: the one-constant Kubelka–Munk [148] and two-constant Kubelka–Munk models [57] (1C-KM and 2C-KM, respectively); the four-flux model [217]; and the color contoning model [14]. We follow different strategies for building these models. For 1C-KM model, where the model parameters (absorption to scattering ratio at each wavelength) are straightforward to compute, we use measurements. For more advanced models, i.e., 2C-KM and four-flux, since measuring their parameters is prone to error, we fit their parameters using our training set and evaluate the model accuracy on the test set. The fitting procedure ensures the maximum capacity of these models as it does not rely on physical measurements. For training data-driven models (all except 1C-KM), including ours, we use the training samples of our dataset.

In Table 8.1, we show the spectral and colorimetric accuracies of all models. We report the statistics of the spectral error and ΔE_{00} under three illuminants, D65, TL84, and A. Under all metrics, our model achieves a significantly lower prediction error, and is on average close to the measurement noise (1%), which was measured through repeated scanning. In Figure 8-4(a), we show the quality of spectral prediction with a number of randomly-drawn

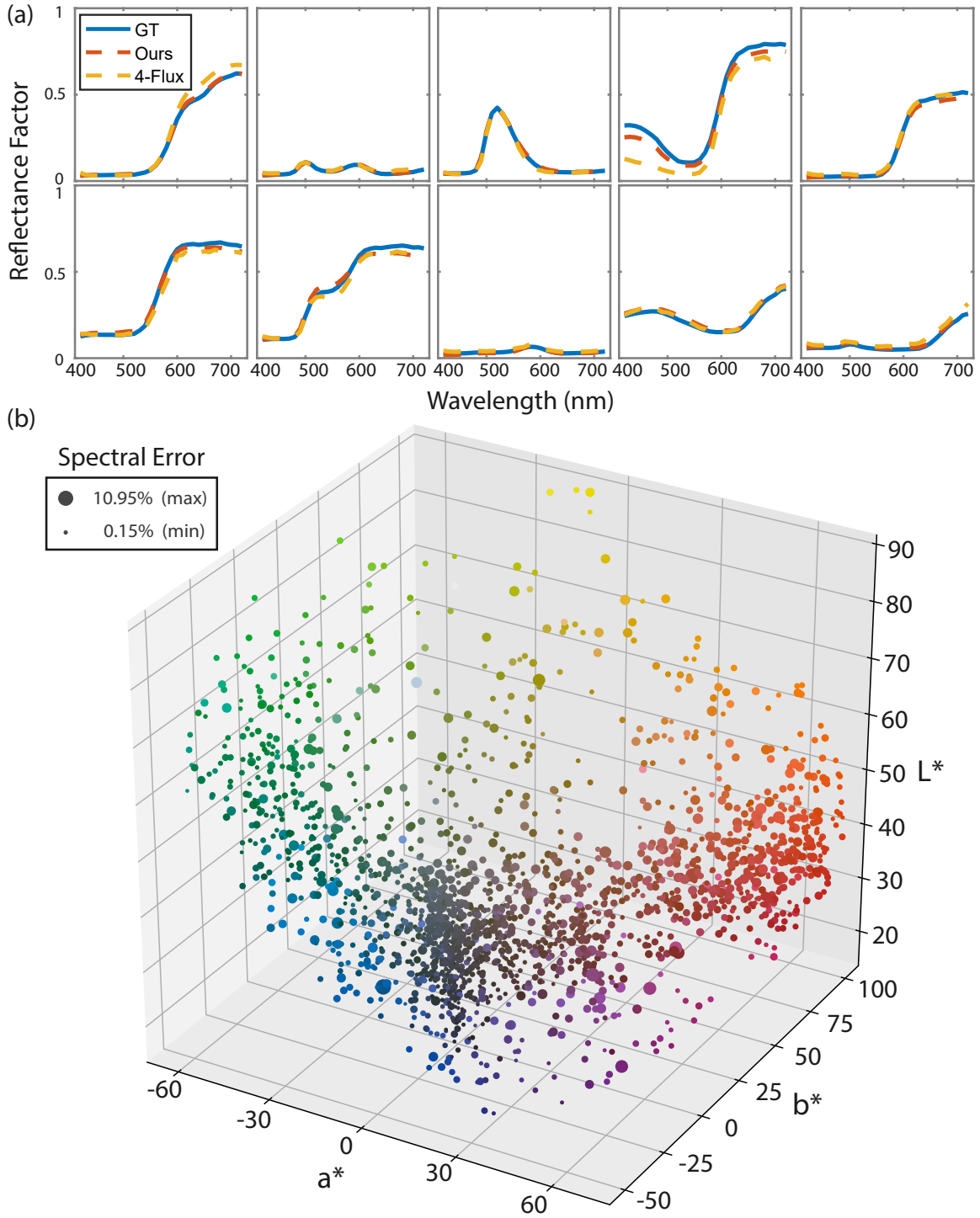


Figure 8-4: (a) Comparison of test spectra from our contouring ink stack dataset predicted by our spectral prediction model $\mathbf{F}$ and by the 4-flux method. (b) Projection of predicted spectra onto CIELAB space (under D65) with the dot volume proportional to the spectral error E_{spec} .

test samples of varying spectra, where our prediction closely follows the measured spectra. In Figure 8-4(b), we plot the predicted test set spectra (2000 samples) in CIELAB space with dot volume proportional to the spectral error. In the supplementary material, we provide additional visualization of equivalent 2D plots with 9 different luminance bins. The standard deviation of mean spectral error at each quadrants is 0.10%, showing no obvious bias towards towards a particular tone. The samples at high luminance region are more sparsely populated, likely due to the use of high concentration inks. Using more layers with lower concentration shall provide denser coverage at the cost of more severe blurriness.

The merit of physically-based models is that they rely on only a limited physical measurements. Therefore, they are scalable to a large number of inks and do not require a large dataset and lengthy training process required by data-driven models. While they usually work well within a limited operational range, these models have difficulties to incorporate the added complexity of the printing process. The contoning model, which is also a data-driven method, assumes only absorbing materials and is unable to predict the spectral properties of our complex stack that includes different inks with a wide range of absorption and scattering.

Layer Prediction Model

In order to evaluate our layout prediction model, we compare the performance of the model trained with varying losses (Table 8.2). We report the errors using the same metrics as in the spectral prediction model, measured from the undiscretized network output and the discretized output, which has been rounded to the nearest integer.

With a negligible difference on the achieved spectral error, adding perceptual loss and layer thickness loss results in roughly 30% reduction on the perceptual difference, and one layer reduction on the total color layers. However, we choose to apply small weights to the perceptual loss and layer thickness loss, since a high perceptual loss weight results in a metameric reproduction. A high weight for layer-thickness loss induces high quantization errors. Depending on the application and number of inks, weights for perceptual loss and ink thickness loss may be further tuned.

The layer prediction model could be directly trained without the pre-trained spectral

Losses	E_{spec}		ΔE_{00} (D65)		ΔE_{00} (TL84)		ΔE_{00} (A)		thickness	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
(a) E_{spec}										
–	0.81	0.49	1.71	1.28	1.84	1.54	1.52	1.53	16.20	3.32
rounded	<u>2.28</u>	2.39	3.75	2.41	3.64	2.59	3.17	2.39	15.73	3.36
(b) $E_{\text{spec}} + E_{\text{LAB}}$										
–	0.81	0.50	1.20	<u>0.93</u>	1.37	<u>1.17</u>	1.17	1.18	16.57	3.16
rounded	2.32	<u>2.31</u>	<u>3.44</u>	<u>2.32</u>	<u>3.25</u>	<u>2.24</u>	<u>2.94</u>	2.10	16.13	3.28
(c) $E_{\text{spec}} + E_{\text{LAB}} + E_{\text{thick}}$										
–	0.82	0.49	<u>1.34</u>	0.91	<u>1.50</u>	1.15	<u>1.31</u>	<u>1.20</u>	14.43	3.17
rounded	2.52	2.38	3.71	2.38	3.57	2.34	3.13	<u>2.07</u>	14.03	3.26
(d) $E_{\text{spec}} + E_{\text{LAB}} + E_{\text{thick}}$ w/ soft quantization layer										
–	1.20	0.71	2.02	1.27	2.16	1.44	1.82	1.36	<u>14.72</u>	3.20
rounded	1.60	1.09	2.70	1.62	2.65	1.61	2.31	1.53	<u>14.80</u>	3.23

Table 8.2: The accuracy of our layer prediction model trained with different objectives, measured in the spectrum error E_{spec} (Eq. 8.1) and the CIEDE2000 ΔE_{00} under three illuminants. We report the mean and standard deviation. The lowest errors are shown boldface and the second lowest underlined.

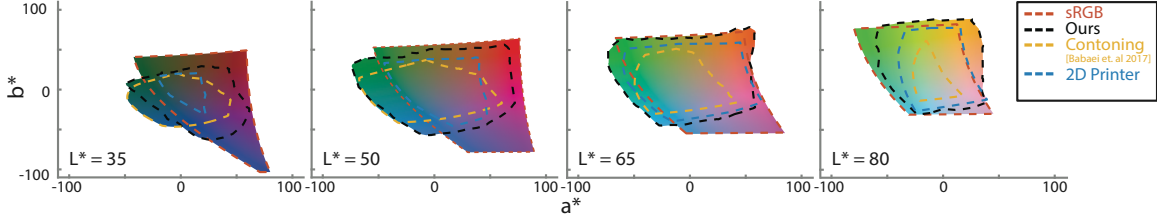


Figure 8-5: Comparison of our 10-ink gamut with two other printing methods and sRGB (under D65). The figure shows a^*b^* slices at different L^* values.

prediction model as a loss, where the loss itself would also be learned through, e.g., a conditional generative adversarial network [127]. We experimented with a model consisting of our **B** as a generator and a fully-connected feedforward network with a 2-way softmax at the end, resembling our **F** but performing binary classification, as a discriminator, trained with our loss terms. We found, however, that a such-trained model was prone to overfitting with poor generalization.

Spectral Vector Error Diffusion

We evaluate the performance of our proposed soft quantization layer and sVED. Figure 8-6 presents the simulated reproduction of a painting under D65 illuminant using varying configurations. Our soft quantization layer helps reduce the errors introduced by rounding (e

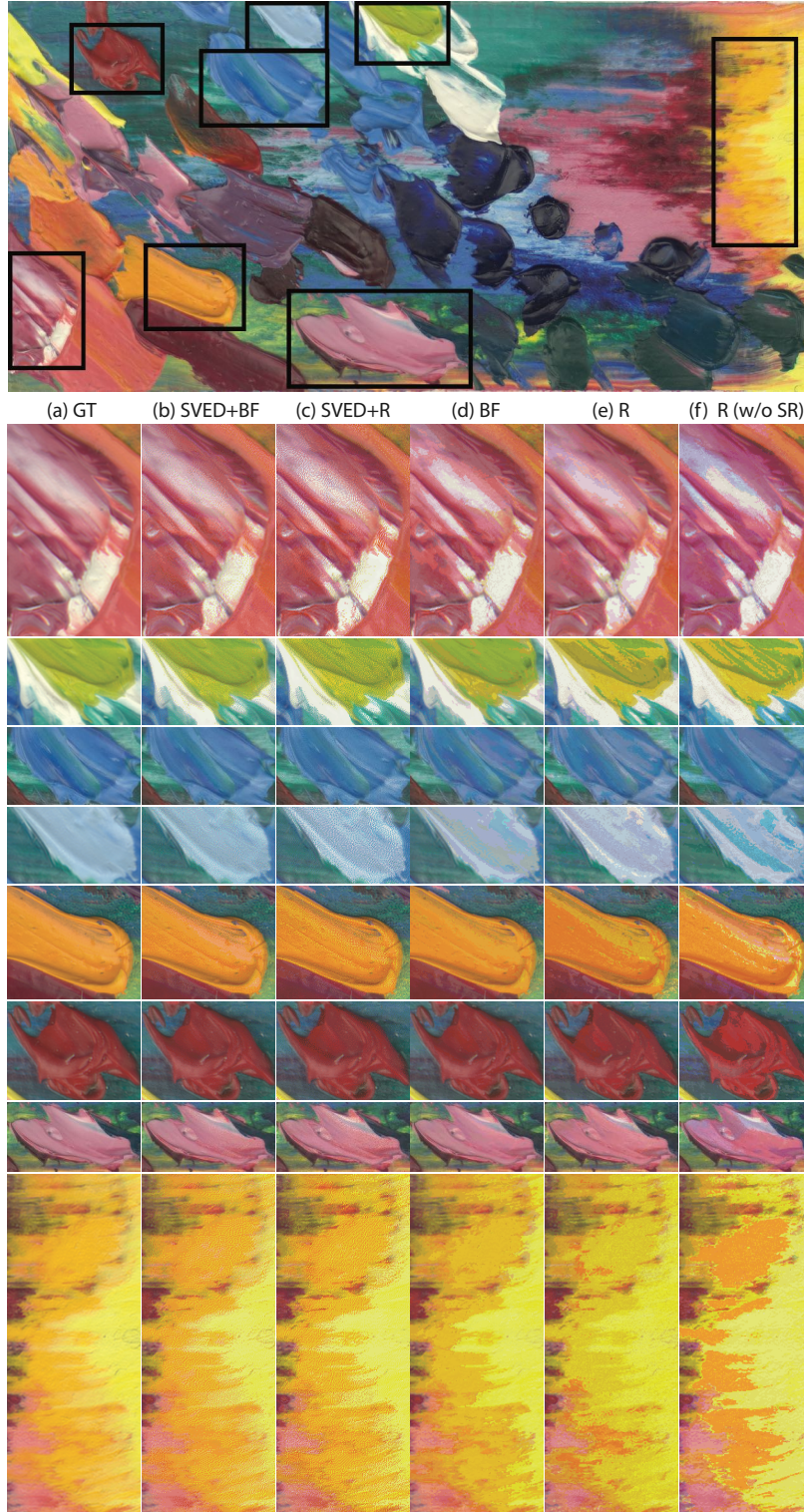


Figure 8-6: Comparisons of different quantization methods. (a) ground truth; (b) spectral vector error diffusion with brute-force enumeration; (c) spectral vector error diffusion with nearest-integer rounding; (d) per-pixel brute-force enumeration; (e) per-pixel nearest-integer rounding; (f) per-pixel nearest-integer rounding *without* using our soft quantization layer. The results in (b–e) were created using the network *with* the soft quantization layer. Each row at bottom corresponds to a rectangular region in the painting at top. Paintings ©Azadeh Asadi.

vs. f). While the local (per-pixel) brute-force enumeration (d) helps alleviate artifacts, the use of our spectral error diffusion greatly improves overall reproduction quality (c). Combined with the brute-force search (b), the sVED shows the best results, with less noticeable graininess, which can be best seen in the electronic version of the paper when zoomed in. Yet the sVED on both the brute-force enumeration (b) and nearest-integer rounding (c) exhibit comparable quality, but with rounding exhibiting better run-time performance (see §8.5.3). We note that, at high luminance and region with smooth gradient (last row), the halftone pattern becomes more visible, likely due to the relatively sparse coverage over the particular gamut area and the high color contrast.

The effectiveness of the soft quantization layer is validated quantitatively in Table 8.2. The reproduction error of soft-quantized results before rounding (d; first row) is slightly higher than that of the results without the soft quantization layer (a–c; first rows each). However, while the error increases sharply after rounding without the soft quantization layer (a–c; second rows), there is a much smaller increase for the soft-quantized results (d; second row).

Color Gamut Evaluation

We compute the color gamut volume of our proposed 10-ink setup in CIELAB color space under D65 illumination. We densely sample 16.8 million layouts, including all possible layouts for stacks with up to 5 inks as well as layouts for stacks with more than 5 inks at a minimal increment step of 2. The layouts are fed to the our spectral prediction model $\mathbf{F}$ to obtain the resulting spectra. We reconstruct the non-convex gamut surface of our CIELAB point cloud using the ball-pivoting algorithm [23]. We then compute the volume enclosed by this surface, which for our 10-ink setup is 79% of the sRGB gamut volume. This may be compared to 42% for a standard CMYK 2D printer based on halftoning, 45% for a typical CMYK contoning system, and 65% for a fluorescent-ink printer (CMYK plus two fluorescent inks). We observe that our gamut is significantly larger than the fluorescent-ink gamut, the state-of-the-art method for wide-gamut printing [215]. In Figure 8-5, we visualize the color gamut comparisons over several iso- L^* slices in CIELAB. As shown in the figure, our 10-ink setup provides additional gamut coverage over color contoning most

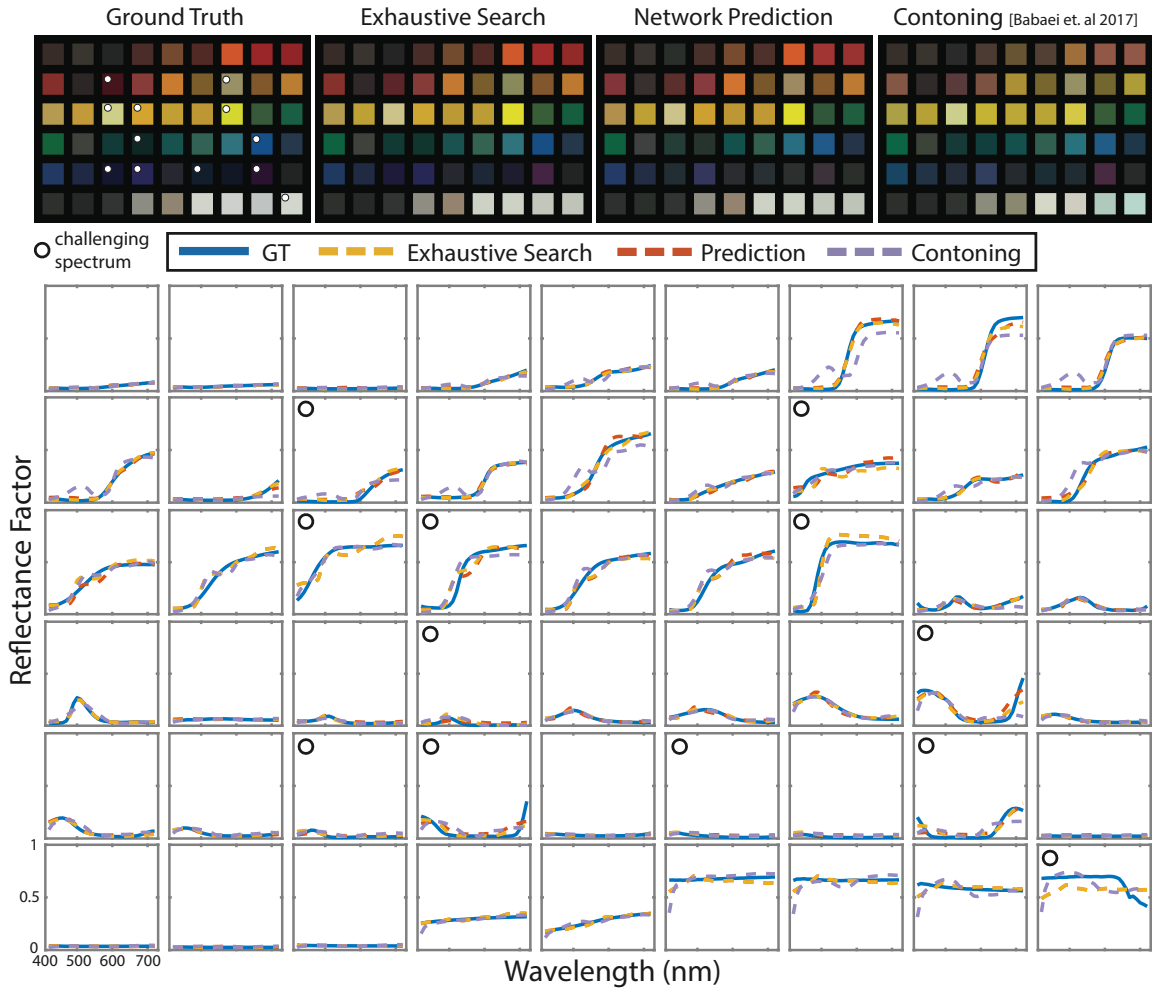


Figure 8-7: Comparison of historical pigment spectra reproduced by the proposed neural network and the nearest neighbor search. The white dots mark the challenging spectra that are difficult for our ink set.

prominently at green, red, orange and pink tone regions, particularly when the luminance is high. To account for the sRGB gamut, further incorporation of blue and purple tone primaries would be necessary.

Reproducing Historical Pigments

We evaluate the ability of our proposed model to reproduce real pigments that have been widely used for artwork. We use the FORS spectral database, which contains 54 pigments of historical interest [53]. This dataset contains the spectra of the pigments in pure powder form, as well as mixed with different binders (6 sets in total). Among these, we reproduce

the measured spectra of pigments mixed with gum arabic, which gives the most saturated colors.

In Figure 8-7, we compare the spectra and the resulting color under the D65 illuminant predicted by color contoning [14] and our model (with rounding enumeration), to the exhaustive search over 16.8 millions of densely sampled spectra which we used to compute the contone color gamut (§8.5.1). The color contoning achieves $E_{\text{spec}} = 4.02\%$ and $\Delta E_{00}(D65) = 5.64$ on average, while our network achieves $E_{\text{spec}} = 2.60\%$ and $\Delta E_{00}(D65) = 4.77$, compared to $E_{\text{spec}} = 2.31\%$ and $\Delta E_{00}(D65) = 2.39$ of the exhaustive search. The spectral error and LAB error of individual pigments are reported in the supplementary material Table 1. For most in-gamut spectra, our network prediction yields results very close to (or indeed identical to) exhaustive search over 16.8 million samples.

We identify 12 challenging (out-of-gamut) spectra that are not well reproduced by our ink set ($E_{\text{spec}} > 5\%$ or $\Delta E_{00}(D65) > 7$ when predicted by our model). For example, the cobalt blue (row 4 at column 8 in Figure 8-7; R4C8 hereafter) and smalt (R5C4) share a steep tail after 700 nm, and similarly, the cobalt violet (R5C8) possesses a steep head before 450 nm, all of which cannot be reproduced by our violet without causing a fat tail from 630-700 nm or a fat head from 450-500 nm. The fat head also appears in the carmine lake (R2C3), whose ramp starts from 600 nm and is 50 nm behind the spectra of our red, orange and magenta. The cadmium yellow (R3C7) starts the ramp at 450 nm, which is 30 nm earlier than our saturated yellow in and causing a lifted yellow spectra (with white layers) to match the curve. The downhill of lithopone (R6C9), which starts at 630 nm, also cannot be reproduced, given that none of our inks' spectra has a descending trend after 600 nm. Overall, our current ink set has difficulties in reproducing yellow- and gray-tone pigments. Ultimately, we believe a further expanded and more carefully selected ink set is necessary to reproduce all pigments' spectra well.

8.5.2 Painting Reproduction Evaluation

We scanned small oil paintings with varying scales, color and texture characteristics, using the same spectral acquisition setup we used for measuring our dataset. During capture,

the distance between the camera and painting is adjusted to match the measurement resolution to the printer’s spatial resolution (35 μm). For paintings larger than the camera’s field of view, we scan them part-by-part and stitch the measurements into a single spectral image [30].

We captured the photographs of all results reported in the paper using a Canon 5D Mark-III DSLR camera with a Canon 100 mm f/2.8 macro lens. The photos of the paintings and their reproductions were taken simultaneously in a single shot under the same light source, then cropped appropriately. We used a fluorescent light and a halogen light to represent cold and warm color temperatures. Additionally, we used a Philips Hue light, which includes different colors of LEDs inside to simulate different lighting.

We include the high-resolution images of both simulations and physical printings in our paper, and refer the readers to the electronic version of it, where more details of our reproduction can be seen and examined when zoomed in.

In Figure 8-8, we show the side-by-side comparisons of the selected paintings patches and our printed reproductions under different physical light sources. The physical dimension of each painting patch is 3.2cm $\times$ 2.4cm. Our results faithfully reproduce a variety of color used in real paintings under a wide range of color. In particular, the paintings exhibits different characteristics: the “water lily” is characterized by its desaturated colors and smooth gradients, while the others use more saturated colors.

We compare our reproduction with the color contoning method [14] in Figure 8-9, using simulation. Our sVED based on 10-ink stacks outperforms the CMYK contoning method with more faithful reproduction of various colors. This is expected as our gamut volume is significantly larger than the color contoning gamut. In contoning, the ink concentrations are kept low to prevent color quantization. Our use of more saturated inks help achieve a larger gamut at the cost of more prominent color quantization, which is handled by our spectral vector error diffusion method (the halftoning pattern can be examined in the electronic version when zoomed in).

In Figure 8-10 we show the printed reproductions of two paintings generated using our 10-ink pipeline, and the same pipeline but with *our* CMYK inks only. Thanks to our sVED method, which allows for high-concentration inks, we can achieve a significantly large

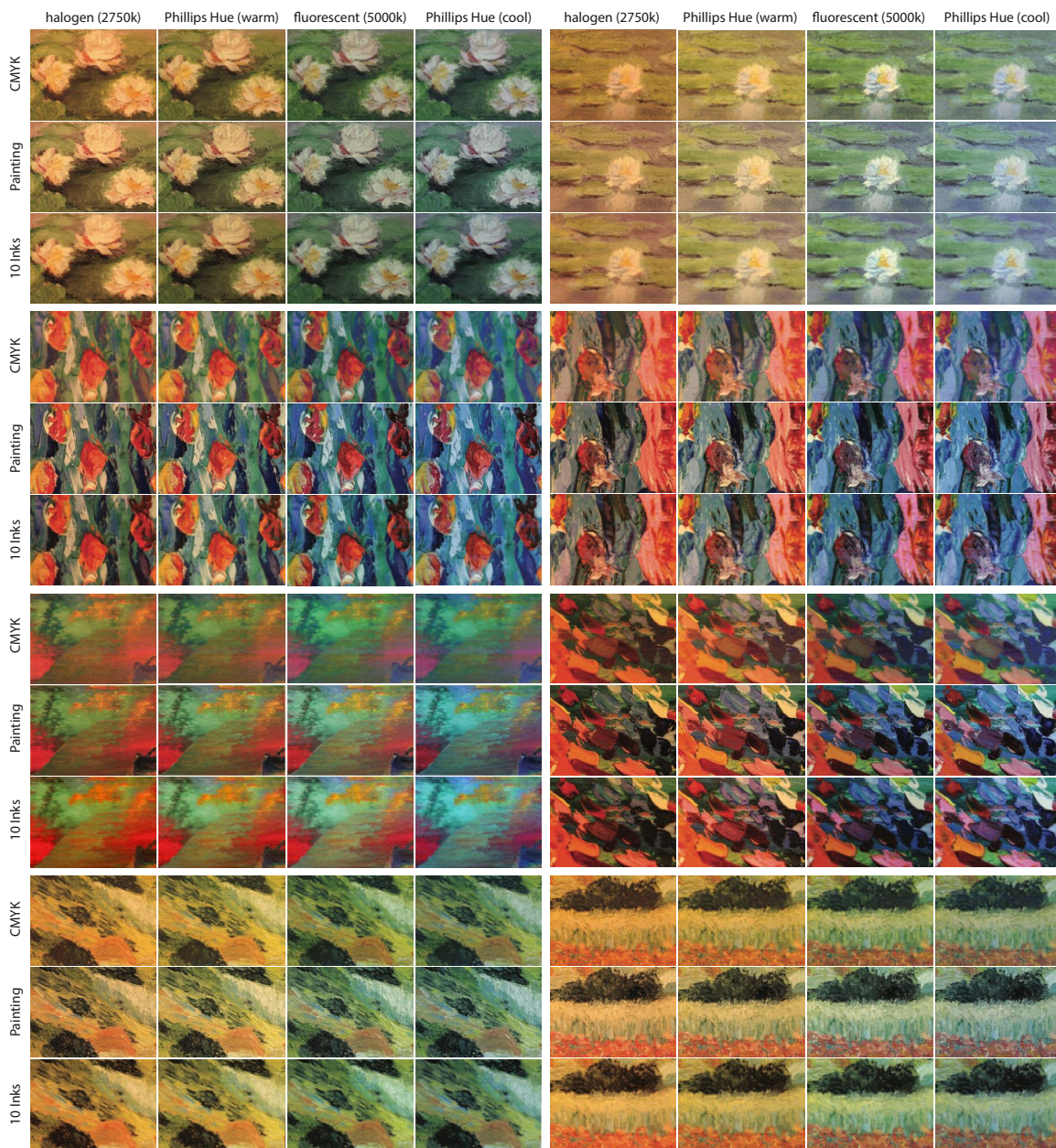


Figure 8-8: Reproduced paintings using our method. We show 4x2 sets of our printed reproductions. For each set, we show our printed reproduction with CMYK inks (top row), the original painting (middle row), and our printed reproductions with 10 inks (bottom row). Each painting is lit by four different light sources (halogen (2750K), Phillips Hue LED (Warm), fluorescent (5000K), and Phillips Hue LED (Cool)), which are shown so ordered. The spectrum of each light is illustrated in the supplementary material Figure A.1. Paintings ©Azadeh Asadi.

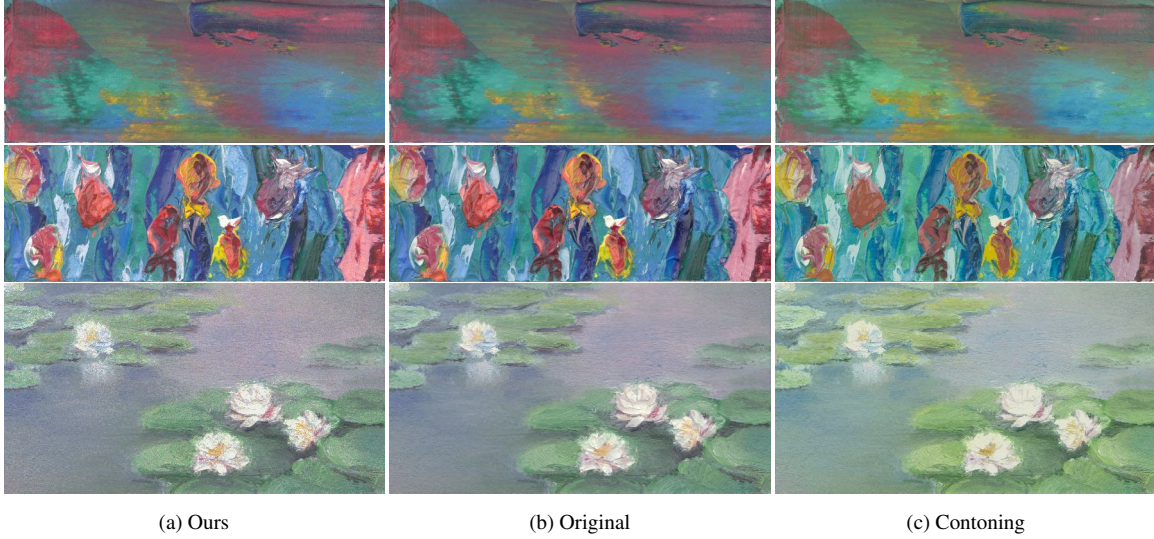


Figure 8-9: Simulated comparisons under D65 illumination of our method to the CMYK contoning. Our reproduction (a) of three paintings are compared against the original painting (b) and the CMYK contining method (c) of [14]. Our method provides consistently more faithful reproductions to the original. Paintings ©Azadeh Asadi.

color gamut even with CMYK. However, although our highly-concentrated CMYK inks yield a large gamut close to a fluorescent-ink printer (65% of sRGB), the 10-ink method shows consistently better reproduction quality under different lights. This justifies the use of a larger number of inks for high-fidelity spectral reproduction. For a quantitative evaluation, we also show the spectra of some sampled points from the original painting, the reproduction by our 10-ink method, and the reproduction by our CMYK inks, along with spectral and colorimetric errors. In general, the CMYK set, even though at high concentration, is not able to reproduce some of spectra faithfully (on average $E_{\text{spec}} = 6.22\%$ and $\Delta_{00} = 7.97$). Our 10-ink printer, on the other hand, gives excellent spectral match between the paintings and prints (on average $E_{\text{spec}} = 2.78\%$ and $\Delta_{00} = 3.80$). Note the spectral difference in the inset of Figure 8-10, where the 10-ink set consistently gives less spectral error. Concerning colorimetric reproduction, the 10-ink set outperforms or does equally well compared to the CMYK. There are a few exceptions, especially out-of-gamut spectra, such as spectrum number 3 in the second painting, a brilliant red color. Although the color reproduction accuracy of the CMYK set is relatively good, it has unacceptable spectral accuracy, indicating the risk of a metameric reproduction. On the other hand, the 10-ink reproduction has much better spectral reproduction accuracy, while the colorimetric error

is high, typical for out-of-gamut colors.

The reproduced paintings have to undergo a series of color transformations (e.g., by the camera, publishing software, and the printer or display) to be presented in this paper, whether it is printed or viewed electronically. Thus, the difference between our reproductions and those by other methods may have been washed out or become subtle. To further provide evidence of the significant differences between them when seen in person, we carried out a small perceptual study, in which we compared our 10-ink with our CMYK prints. We asked participants to evaluate the color fidelity of the two candidate methods as compared to the original, under three different lighting conditions (fluorescent at 3500 K, LED at 5800 K, and cloudy daylight).

We asked participants to evaluate printed patches on a scale between -3 , indicating a strong preference *against* ours, and $+3$, *for* ours, with 0 being indifferent. Based on a study involving 8 participants, the mean preference score was $+1.69$ with standard deviation of 1.14, which shows a consistent and statistically significant ($p < 0.001$) preference for our reproduction.

8.5.3 Run-time Performance

Our model is implemented using TensorFlow [3], and trained and tested on an NVIDIA Titan X (Pascal) GPU. It takes about 15 minutes to train the spectral prediction network **F** and about 45 minutes for the layout generation network **B**. Note that these are one-time computations and once the model is built, it can be used for different input paintings. To predict the layout of a painting consisting of one million spectral pixels, it takes on average 0.3 seconds.

Our spectral vector error diffusion is implemented in Python (with NumPy) and TensorFlow. We follow the optimal scheduling pattern proposed by Metaxas [189] to batch the evaluations of uncorrelated pixels at each time step, which significantly reduces the number of network inference calls. During error propagation, a damping factor between 1 and 0.6 (for highly saturated paintings) is applied to the error diffusion kernel to prevent the algorithm from diverging. Error diffusion on a painting of one million spectral pixels takes

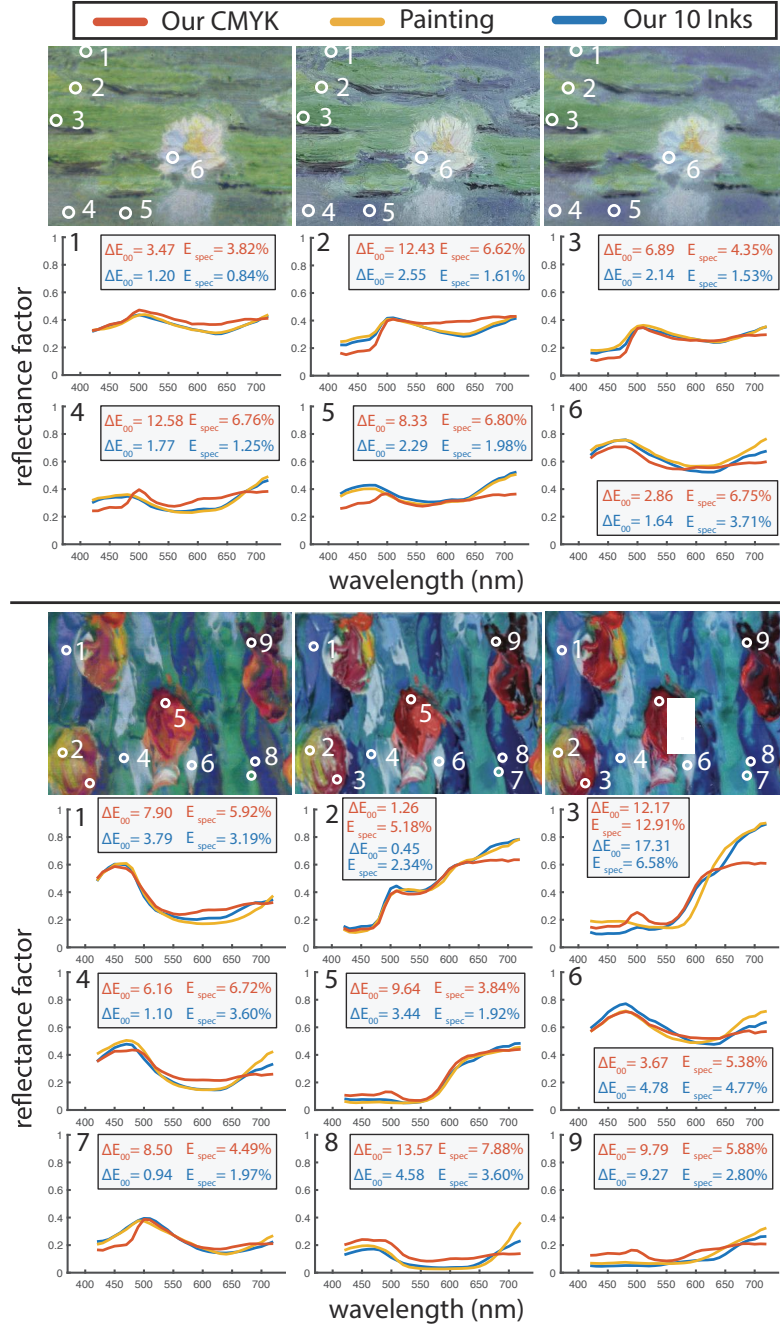


Figure 8-10: Comparisons of the printed results and measured spectral reflectance between *our* CMYK reproduction, 10-ink reproduction and the original painting. In the lily example (top), the reconstruction of green and purple hues are extensively examined, which are known to be difficult for CMYK. In the bottom example, we examine a wide variety of both saturated and desaturated colors. In both results, 10 inks result consistently outperforms the CMYK both perceptually and in terms of spectral RMSE. Paintings ©Azadeh Asadi.

on average 23 seconds using simple rounding and 590 seconds using brute-force rounding enumeration.

For physical spectral prediction models, although the forward model can be very efficient, the backward model needs a model inversion with iterative optimization algorithms, which renders the print-data generation into a slow process. We tested the color contoning run-time performance using our training dataset. The forward model takes more than 40 hours in MATLAB on a MacBook Pro with a 2.8 GHz i7 processor and 16 GB of RAM. This is because the model inverts a large weight matrix ($n \times n$, n being the number of training data samples) for predicting every layout. We did not test the contoning backward model, as a nearest neighbor search among 0.8 billion points in a 31-dimensional spectral space is intractable.

8.6 Discussion & Limitation

In this chapter, we propose a complete pipeline capable of reproducing spectral reflectance using a 3D printer. To this end, we present a data-driven approach for predicting the optimal stack of different inks. Our framework provides accurate and efficient forward and backward predictions. Our proposed spectral model outperforms state-of-the-art physical and data-driven prediction models, mostly by large margins. We demonstrate the effectiveness of our workflow by reproducing a number of challenging oil paintings, painted by our artist collaborator [11]. We further propose a novel spectral vector error diffusion that combines both halftoning and contoning techniques in a complementary manner to leverage the potential of both methods. Our sVED algorithm uses error diffusion on a very large number of potential contone stacks, very efficiently, thereby resolving both layout discretization and color quantization problems. Our framework takes the first step in the exciting direction of fine art reproduction, and we hope it triggers more works, leading to the exploitation of rapid advances in fabrication technologies and computational techniques to protect our cultural heritage. Below, we discuss the limitation of our current work and propose future research directions.

Our experiment with the historical pigments revealed that our ink library is suboptimal,

unable to faithfully reproduce certain spectral curve shapes, such as cobalt blue. An exciting extension of our system is to further expand the ink library. With an extended ink library, an efficient painting-specific ink selection algorithm would be required to allocate inks from the library to the limited number of printer channels.

A data-driven model fits well with our current problem size. However, for maximum scalability, striving for more advanced physical prediction models is an important research direction for the future. For this, one can take inspiration from physically-based models for rendering of layered materials [129]. Another appealing direction is to construct a physically-aware neural network that, instead of working with ink labels, is built on proper spectral measurements. By learning from the physics of the problem, we expect the resulting network to be generalizable to arbitrary inks not present in the training set. However, we expect a significant larger training set to be necessary given the increase in the number of degrees of freedom.

Compared to the previous contoning method [14], the dot gain in our approach is significantly lower due to the thin (high-concentration) ink stacks. Consequently, the blur is much less significant without any preprocessing [14] or optimizing the material arrangement for the least crosstalk [69]. Having said that, we believe that improving the blur problem through careful tuning of material thickness and concentration, optimizing halftoning and contoning parameters, and finding optimal material arrangements [69] is necessary for high-quality reproduction.

Finally, a painting reproduction without considering the rich spatially-varying gloss and translucency found in many paintings as well as the 3D trail of the brush-stroke is far from complete. Incorporating gloss and microgeometry will also improve the spectral reproduction by recognizing the unavoidable highlights measured during the capture, and compensating for them by surface reflection (and not by diffuse color). Although there have been recent efforts in combined fabrication of these appearance attributes [70], there is still a long way to go for archival-quality fabricated fine art.

Chapter 9

Conclusion & Future Work

In order for virtual and augmented reality to become widely embraced as a personal and social computing platform, the displays of tomorrow need to deliver 3D imagery with wide field of view, high resolution, high brightness, and comfort all packaged into a compact form resembling everyday eyeglasses for extended period of wear. Previous techniques like parallax barrier, integral imaging, or multi-layer light field displays have intrinsic limitations in realizing all these aspects. On the other hand, there's a pressing need to develop lightweight appearance modeling and authoring methods for both expert and non-expert users to efficiently create realistic appearance in a virtual environment. Classical measurement-based appearance acquisition methods require expansive and carefully-calibrated setups, as well as exhaustive measurement. Forward material design using authoring tools like Blender and Substance 3D Designer demands specialized expertise plus a laborious and time-consuming design process. In a separate thread, we investigate the extension of the visual Turing test into the physical domain, with a specific emphasis on replicating fine arts, notably paintings. This endeavor presents significant challenges, particularly in accurately reproducing the spectral reflectance of the original artwork.

This dissertation has presented a series of computational methods towards tackling above challenges, in particular: 1) algorithms to efficiently generate high-quality 3D holograms for near-eye displays; 2) inverse material modeling algorithms that procedurally capture the appearance from a user-input example (i.e., a single flash image of the material) and enable efficiently material authoring. 3) a novel workflow for reproducing physical appear-

ance of paintings with accurate spectral reflectance utilizing a multi-material 3D printer. In the following, we briefly summarize the key contributions of this dissertation and discuss potential future research directions.

9.1 Discussion

Computer Generated Holography Numerous algorithms have been devised to address persistent and critical challenges in computer-generated holography. In Chapter 3, we introduced a novel light field display system based on CGH. Our approach includes a specialized light field rendering geometry tailored for hologram calculation, meticulously designed to eliminate aliasing in both oversampling, undersampling, and avoid geometry clipping. Through rigorous theoretical analysis, we established the limitations regarding aliasing, sampling, spatial resolution, and design parameters critical for utilizing holograms to faithfully reproduce light fields. Additionally, we showcased a GPU implementation based on the Fresnel diffraction integral, which achieved a remarkable speed enhancement of five orders of magnitude compared to prior methodologies, enabling interactive performance at 1080P resolution with per-pixel depth calculation for the first time. To validate our rendering and calculation techniques, we constructed a see-through color display prototype with complex modulation capabilities, supporting continuous focus cues and intra-pupil occlusion. Fully color 3D scenes captured at different focal distances with properly reconstructed defocus blur were verified on the prototype.

In Chapter 4, we presented pioneering deep learning-based techniques for 3D hologram generation, achieving real-time computational speeds (i.e., 60Hz for 1080p content) crucial for future AR/VR applications while preserving image quality without perceptual loss. This advancement stems from several pivotal findings and contributions. We first observe that for AR/VR application where the viewer’s head is tracked, the hologram can be computed on-demand based on the current viewpoint, allowing us to simplify the input to an RGB-D. Meanwhile, we recognize that hologram computation is fundamentally a spatially-varying large kernel convolution, and we show it can be efficiently approximated by a convolutional neural network. To successfully train the CNN, a carefully designed large-scale 3D

hologram dataset is introduced with proper scene complexity and distribution, as well as, a novel physics-informed loss function that explicitly infuses the knowledge of optical propagation. Finally, a fast variant of double-phase method allows for reduction of speckle noises commonly presented during complex to phase-only conversion, allowing physical instantiation of the hologram with high-quality and low noise. The proposed system is validated on a full-color phase-only holographic display prototype and full-color 3D hologram video playback system at 1080p resolution was first demonstrated. In a second iteration of the system, we further integrated phase-only encoding into an end-to-end training pipeline, eliminating the need for manual tuning as previously required by the double-phase method. Layered depth images replace RGB-D inputs to accurately capture wavefronts from background objects, thereby enhancing the quality of generated holograms. Additionally, we introduce a Zernike-polynomial-based computational vision aberration correction technique to customize holograms for individual users, resulting in improved visual comfort and reduced system complexity for holographic near-eye displays.

Furthermore, we introduced a neural compression system tailored for hologram images and videos, recognizing the limitations of conventional compression algorithms for CGHs due to their distinct characteristics. For hologram images, an end-to-end trained encoder network compresses the hologram into a low-dimensional latent code, outperforming state-of-the-art formats like HEIC or BPG in compression rate. For hologram videos, we achieve superior compression rates by leveraging motion vectors in the H.265 video codec to assist the encoder-decoder network in compressing the high-frequency residual essential for 3D image formation, while using H.265 with a high constant rate factor compresses the low and mid-frequency content. This comprehensive approach ensures efficient compression while preserving the fidelity necessary for immersive holographic experiences.

In Chapter 5, we introduced the first optimization-based approach for generating multi-color holograms capable of simultaneous illumination by multiple color primaries with co-optimized intensities. This method, without requiring alterations to display hardware, enables a more efficient utilization of each color primary, enhancing brightness by dedicating additional "on" time in every subframe. To ensure high-quality multi-color holograms, we proposed a suite of application-specific losses aimed at regularizing laser intensity, phase

smoothness, and color uniformity during gradient-based optimization. Our experimental results demonstrate a consistent enhancement in perceived brightness by 1.8x for natural image targets, without inducing significant distortion or color artifacts. Additionally, we observed an average quantitative improvement of PSNR by 8dB. In simulation, we also illustrated the method’s potential in increasing display framerate by reducing the number of subframes required to achieve the same level of color reproduction.

In Chapter 6, we introduced the first ergonomic-centric optimization-based method aimed at systematically enhancing the realism, comfort, and immersion of holographic displays—areas that were only addressed individually in previous works. To achieve greater realism, we demonstrated the efficacy of temporal time-multiplexing and incoherent wave propagation simulation in reducing speckle noise and generating realistic defocus blur. In the realm of comfort, we incorporated higher-order diffractions into the image formation process, eliminating the necessity for optical filtering and enabling the miniaturization of display form factors. Lastly, to augment immersion, we conducted explicit optimization of perceived 3D images across various pupil sizes and positions, thereby expanding the effective eyebox. Our experimental results validate the effectiveness of the proposed method, showcasing a simultaneous improvement in perceived experimental image quality by a factor of 3dB in PSNR, alongside a 2x expansion in eyebox size without necessitating a Fourier filter.

Appearance Modeling In Chapter 7, a by-example inverse procedural material modeling system is introduced to recover SVBRDF of a near-planar surface from a single flash image without leveraging user expertise. The system comprises the first differentiable procedural material modeling library, designed to translate Adobe Substance 3D materials into differentiable computational graphs; a hybrid workflow that trains a CNN model to predict graph parameters that initially match a target image, followed by a refinement step based on gradient-based optimization to further close the perceptual appearance gap. The system is validated on a diverse set of synthetic and real-world captured material photos, demonstrating successful recovery of SVBRDFs with high visual similarity and computational efficiency (e.g., the total inference and optimization time reduces manual forward design

from hours or days to a few minutes). Subsequent enhancements to the system include the integration of discrete and continuous optimization techniques, broadening its capacity to capture a wider range of materials, match highly regular and stochastic patterns, and decrease the need for post-manual fine-tuning. Additionally, it introduces an efficient method for translating high-level semantic appearance control from one exemplar graph to a target graph, empowering users to parametrically edit the output material appearance in a controlled manner, thereby enhancing usability and flexibility. The proposed system is commercially licensed to Adobe and under integration into Substance 3D Designer.

In Chapter 8, we introduced an innovative data-driven workflow for accurate physical reproduction of painting appearances, capturing their spectral reflectance with precision through the utilization of a multi-material 3D printer. Traditional 2D printers, limited by a fixed set of four inks and reliant on half-toning techniques, inherently suffer from a restricted color gamut and loss of details/sharpness, impeding faithful reproduction of references and appearance matching under varying lighting conditions. By harnessing the capabilities of a 3D printer equipped with ten distinct transparent inks, our proposed system combines halftoning with contoning—a method involving the vertical stacking of thin ink layers to achieve continuous tones. We employ a pair of multi-layer perceptrons trained to predict both the spectral reflectance of a given ink stack and the optimal ink stack required to reproduce a given spectral reflectance. Leveraging insights from these models, we develop a spectral vector error diffusion algorithm that seamlessly integrates halftoning with contoning, effectively addressing challenges related to layout discretization, color quantization, and screen door effect associated with half-toning alone. We showcase the successful reproduction of several real paintings and historically significant pigments using our prototype implementation. The availability of high-quality painting facsimiles holds significant potential, not only for expanding exposure but also for preserving the integrity of the originals. This is particularly valuable in contexts such as restoration practices, conservatory studies, and educational initiatives within museums.

9.2 Future Work

One of the central themes in recent CGH research revolves around the co-design of software and hardware, offering promising avenues to unlock various captivating aspects of the visual experience in the future. This synergy is especially potent when delving into the spatio-temporal intricacies of human perception. An example of potential future work aligning with this theme is computational pixel super-resolution, a technique aimed at augmenting the perceived resolution of holographic displays. This concept has been implemented by hardware in existing consumer projector systems, where an actuator shifts the digital micro-mirror device (DMD) by sub-pixel distances to create multiple sub-frames that are optically time-multiplexed to form a higher-resolution image. While this approach is effective in the context of projectors, hardware-based implementation for wearable holographic display is challenging due to the inherent complexity of the optical system. By computationally modeling sub-pixel shift in optical propagation to create multiple perceptually overlapped shifted-view holograms and time-multiplexing them using high refresh rate piston-based phase-only spatial light modulators (e.g., TI DLP6750Q1EVM), we could potentially achieve software-based pixel super-resolution in a compact holographic displays. Similar approach could be applied to improve other critical design parameters of holographic displays, such as field of view through time-multiplexing different off-axis illuminations, and compactness through co-design generation algorithms with waveguide. It is also exciting to explore the latest advancement in machine learning such as diffusion models and large language models to further enhance the quality of conditional hologram image/video generation or form direct bridge from other modalities to holograms, such as text. These applications have been extensively explored in the context of 2D image generation, but their potential in holographic displays remains largely untapped.

Meanwhile, existing spatial light modulators still lack the ideal property for holographic displays, such as ultra high resolution, complex modulation, and high refresh rate coupled with high bit depth and dynamic range of modulation. For example, when complex modulation is available on a single SLM, the time-consuming phase retrieval optimization may be omitted and complex hologram obtained from single-step forward propagation or

few-step optimization could be displayed directly, significantly reducing the computational overhead, while not compromising the system complexity and compactness. Thus, further development of SLM technology is another key area for future research.

Besides algorithmic and hardware advances, perceptual quality of projected 3D imagery ultimately matters in a near-eye displays. Exploring the perceptual implications of various CGH algorithms and target content types represents a promising avenue for future research. In recent years, a diverse array of target content, including 2.5D RGB-depth maps, 3D focal stacks, and 4D light fields, has emerged to guide CGH generation. However, the perceptual consequences stemming from the selection of algorithm and target content type remain unclear. Establishing a perceptual testbed to assess the realism of holographic displays across different CGH supervision formats is a crucial step toward unraveling the perceptual implications inherent in diverse CGH algorithms and target content types.

For inverse appearance modeling, there are many exciting directions to further explore. First, leveraging recent advancements in large language models and language-vision models opens up the possibility of integrating multi-modal inputs into the material authoring pipeline. While Chapter 7 primarily utilizes images as inputs, recent work by Hu et al. [115] has begun exploring the integration of text prompts. However, the text prompts utilized in their study tend to focus predominantly on macro-scale descriptions, overlooking the nuanced aspects of micro-scale material appearance. Conversely, examples or targets for inverse modeling could encompass a wide array of forms, including audio, sketches, and drawings, each catering to different user preferences. Moreover, inputs may stem from non-traditional environments such as augmented reality/virtual reality (AR/VR). Offering solutions that accommodate diverse inputs across varying environments can significantly enhance the appeal and versatility of material authoring tools.

Development of public large scale material dataset is contingent with the rise of large language models. The success of large language models in natural language processing can be attributed to the availability of vast amounts of text data. Similar success is observed for 3D generation with the rise of Objaverse and Objaverse-XL [59] with 10M+ 3D objects. In contrast, the scarcity of large-scale material datasets poses a significant challenge to the

training of large models for material authoring. The success of Hu et al. [115] for generating procedural graphs from text prompt and Vecchione et al. [249] for training diffusion models to map flash photo to SVBRDFs is largely attributed to the company-owned large material libraries, which are not publicly available. Having a public large-scale material dataset would greatly facilitate material authoring and understanding.

Scaling the capabilities of inverse material modeling systems to handle large-scale scenes featuring multiple objects and complex lighting conditions is still in its nascent stages. While recent advancements in neural rendering have propelled the development of neural scene representations capable of recovering the shape and spatially-varying reflectance of single or few objects under unknown lighting conditions [276, 28, 167], extending these methods to encompass room-scale or outdoor scenes with intricate environmental lighting remains computationally intensive and challenging. Additionally, efforts have been made to employ explicit optimization-based methods, leveraging differentiable rendering for shape and material reconstruction. Notably, inverse procedural material optimization, utilizing the pipeline and library proposed in Chapter 7, has been combined with differentiable rendering to extend material modeling to room-scale scenes by Yeh et al. [265] and Yan et al. [262].

Finally, for physical appearance reproduction of paintings using 3D printing, our work only tapped upon more accurate color reproduction, yet there exist numerous other facets of appearance that warrant further exploration. These include considerations such as surface roughness, brush stroke and reliefs, aging effects, cracks, glossiness, translucency, and subsurface scattering. Reproducing these nuanced aspects necessitates more sophisticated printing and modeling systems. For instance, achieving glossiness could involve printing with varnish layers and employing different curing processes, while capturing brush strokes and reliefs might require printing with varying layer heights. Additionally, data-driven approaches could enhance Monte Carlo light transport simulations for more efficient modeling of translucency and subsurface scattering given a print material distribution. We anticipate that future research will delve into these areas to achieve more precise and faithful reproduction of paintings.

Bibliography

- [1] <https://bellard.org/bpg>.
- [2] <https://nokiatech.github.io/heif/technical.html>.
- [3] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16*, pages 265–283, Berkeley, CA, USA, 2016. USENIX Association.
- [4] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [5] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Trans. Graph.*, 35(4):1–13, July 2016.
- [6] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Practical SVBRDF capture in the frequency domain. *ACM Trans. Graph.*, 32(4):1–12, July 2013.
- [7] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot SVBRDF capture for stationary materials. *ACM Trans. Graph.*, 34(4):1–13, July 2015.
- [8] Kurt Akeley, Simon J Watt, Ahna Reza Girshick, and Martin S Banks. A stereo display prototype with multiple focal distances. *ACM Trans. Graph.*, 23(3):804–813, August 2004.
- [9] Miguel A Alonso. Wigner functions in optics: describing beams as ray bundles and pulses as particle ensembles. *Adv. Opt. Photon., AOP*, 3(4):272–365, December 2011.
- [10] Victor Arrizón and David Sánchez-De-la Llave. Double-phase holograms implemented with phase-only spatial light modulators: performance evaluation and improvement. *Appl. Opt.*, 41(17):3436–3447, June 2002.

- [11] Azadeh Asadi. Freelance artist. <http://www.azadehasadi.com/index.html>, 2017. Accessed: 2018-08-15.
- [12] Louis-Philippe Asselin, Denis Laurendeau, and Jean-Francois Lalonde. Deep SVBRDF estimation on real materials, 2020.
- [13] Vahid Babaei and Roger D Hersch. N-ink printer characterization with barycentric subdivision. *IEEE Transactions on Image Processing*, 25(7):3023–3031, 2016.
- [14] Vahid Babaei, Kiril Vidimče, Michael Foshey, Alexandre Kaspar, Piotr Didyk, and Wojciech Matusik. Color contoning for 3D printing. *ACM Trans. Graph.*, 36(4):1–15, July 2017.
- [15] Seung-Hwan Baek, Incheol Kim, Diego Gutierrez, and Min H Kim. Compact single-shot hyperspectral imaging using a prism. *ACM Transactions on Graphics (TOG)*, 36(6):217, 2017.
- [16] J Ballé, D Minnen, S Singh, S J Hwang, and others. Variational image compression with a scale hyperprior. *arXiv preprint arXiv*, 2018.
- [17] Farhan Baqai, Je-Ho Lee, Jan P Allebach, et al. Digital color halftoning. *Signal Processing Magazine, IEEE*, 22(1):87–96, 2005.
- [18] James Barabas, Sundeep Jolly, Daniel E Smalley, and V Michael Bove, Jr. Diffraction specific coherent panoramagrams of real scenes. In Hans I Bjelkhagen, editor, *Practical Holography XXV: Materials and Applications*. SPIE, February 2011.
- [19] H O Bartelt, K-H Brenner, and A W Lohmann. The wigner distribution function and its optical production. *Opt. Commun.*, 32(1):32–38, January 1980.
- [20] M J Bastiaans. Wigner distribution function and its application to first-order optics. *J. Opt. Soc. Am.*, 69(12):1710, December 1979.
- [21] Heinz H Bauschke, Patrick L Combettes, and D Russell Luke. Hybrid projection-reflection method for phase retrieval. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.*, 20(6):1025–1034, June 2003.
- [22] Stephen A Benton and V Michael Bove, Jr. *Holographic Imaging*. Wiley-Interscience, USA, 2008.
- [23] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359, 1999.
- [24] Roy S. Berns, Brittany D. Cox, and Farhad Moghareh Abed. Wavelength-dependent spatial correction and spectral calibration of a liquid crystal tunable filter imaging system. *Appl. Opt.*, 54(12):3687–3693, Apr 2015.

- [25] Tobias Birnbaum, Ayyoub Ahar, David Blinder, Colas Schretter, Tomasz Kozacki, and Peter Schelkens. Wave atoms for digital hologram compression. *Appl. Opt.*, 58(22):6193–6203, August 2019.
- [26] William P Bleha and LiJuan Alice Lei. Advances in liquid crystal on silicon (lcos) spatial light modulator technology. pages 87360A–87360A, 2013.
- [27] L. Bommers, X. Lin, and J. Zhou. Mvmed: Fast multi-object tracking in the compressed domain. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1419–1424, 2020.
- [28] Mark Boss, Raphael Braun, V Jampani, J Barron, Ce Liu, and H Lensch. NeRD: Neural reflectance decomposition from image collections. *ICCV*, pages 12664–12674, December 2020.
- [29] Jean-Yves Bouguet. A release of a camera calibration toolbox for matlab. 2008.
- [30] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.
- [31] Alan Brunton, Can Ates Arikan, Tejas Madan Tanksale, and Philipp Urban. 3D printing spatially varying color and translucency. *ACM Trans. Graph.*, 37(4):1–13, July 2018.
- [32] Alan Brunton, Can Ates Arikan, and Philipp Urban. Pushing the limits of 3D color printing: Error diffusion with translucent materials. *ACM Trans. Graph.*, 35(1):1–13, December 2016.
- [33] Brett Burley. Physically-based shading at disney. In *ACM SIGGRAPH 2012 Courses*, 2012.
- [34] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision – ECCV 2012*, pages 611–625. Springer Berlin Heidelberg, 2012.
- [35] O Cakmakci and J Rolland. Head-worn displays: a review. *J. Display Technol.*, 2(3):199–216, September 2006.
- [36] Ali Cem, M Kivanc Hedili, Erdem Ulusoy, and Hakan Urey. Foveated near-eye display using computational holography. *Sci. Rep.*, 10(1):14905, September 2020.
- [37] Praneeth Chakravarthula, Seung-Hwan Baek, Florian Schiffers, Ethan Tseng, Grace Kuo, Andrew Maimone, Nathan Matsuda, Oliver Cossairt, Douglas Lanman, and Felix Heide. Pupil-Aware holography. *ACM Trans. Graph.*, 41(6):1–15, November 2022.
- [38] Praneeth Chakravarthula, Yifan Peng, Joel Kollin, Henry Fuchs, and Felix Heide. Wirtinger holography for near-eye displays. *ACM Trans. Graph.*, 38(6):1–13, November 2019.

- [39] Praneeth Chakravarthula, Ethan Tseng, Tarun Srivastava, Henry Fuchs, and Felix Heide. Learned hardware-in-the-loop phase retrieval for holographic near-eye displays. *ACM Trans. Graph.*, 39(6):1–18, November 2020.
- [40] Praneeth Chakravarthula, Zhan Zhang, Okan Tursun, Piotr Didyk, Qi Sun, and Henry Fuchs. Gaze-Contingent retinal speckle suppression for Perceptually-Matched foveated holographic displays. *IEEE Trans. Vis. Comput. Graph.*, 27(11):4194–4203, November 2021.
- [41] C Chang, K Bang, G Wetzstein, B Lee, and L Gao. Toward the next-generation VR/AR optics: a review of holographic near-eye displays from a human-centric perspective. *Optica*, 2020.
- [42] Chenliang Chang, Wei Cui, and Liang Gao. Foveated holographic near-eye 3D display. *Opt. Express*, 28(2):1345–1356, January 2020.
- [43] J-S Chen and D P Chu. Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications. *Opt. Express*, 23(14):18143–18155, July 2015.
- [44] J.-S. Chen and D. P. Chu. Improved layer-based method for rapid hologram generation and real-time interactive holographic display applications. *Opt. Express*, 23(14):18143–18155, Jul 2015.
- [45] Rick H-Y Chen and Timothy D Wilkinson. Computer generated hologram from point cloud using graphics processor. *Appl. Opt.*, AO, 48(36):6841–6850, December 2009.
- [46] Pavel A Cheremkhin and Ekaterina A Kurbatova. Quality of reconstruction of compressed off-axis digital holograms by frequency filtering and wavelets. *Appl. Opt.*, AO, 57(1):A55–A64, January 2018.
- [47] Inchang Choi, Daniel S Jeon, Giljoo Nam, Diego Gutierrez, and Min H Kim. High-quality hyperspectral reconstruction using a spectral prior. *ACM Transactions on Graphics (TOG)*, 36(6):218, 2017.
- [48] Suyeon Choi, Manu Gopakumar, Yifan Peng, Jonghyun Kim, Matthew O’Toole, and Gordon Wetzstein. 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*, number Article 32 in SIGGRAPH ’22, pages 1–9, New York, NY, USA, July 2022. Association for Computing Machinery.
- [49] Suyeon Choi, Manu Gopakumar, Yifan Peng, Jonghyun Kim, and Gordon Wetzstein. Neural 3D holography: learning accurate wave propagation models for 3D holographic virtual and augmented reality displays. *ACM Trans. Graph.*, 40(6):1–12, December 2021.

- [50] Suyeon Choi, Jonghyun Kim, Yifan Peng, and Gordon Wetzstein. Optimizing image quality for holographic near-eye displays with michelson holography. *Optica, OPTICA*, 8(2):143–146, February 2021.
- [51] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, S Mohamed, and A Vedaldi. Describing textures in the wild. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, November 2013.
- [52] T Claasen and W Mecklenbräuker. Time-frequency signal analysis. *Philips J. Res.*, 1980.
- [53] Antonino Cosentino. Fors spectral database of historical pigments in different binders. 2:57–68, 09 2014.
- [54] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.
- [55] Dengxin Dai, Hayko Riemenschneider, and Luc Van Gool. The synthesizability of texture examples. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3027–3034. IEEE, June 2014.
- [56] Kristin J Dana, Bram van Ginneken, Shree K Nayar, and Jan J Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, January 1999.
- [57] Hugh R. Davidson and Henry Hemmendinger. Color prediction using the two-constant turbid-media theory*. *J. Opt. Soc. Am.*, 56(8):1102–1109, Aug 1966.
- [58] Sebastien Deguy, Albert Benassi, and Université Blaise Pascal. A flexible noise model for designing maps. <http://citeseerx.ist.psu.edu> > *viewdoc* > *summary*-<http://citeseerx.ist.psu.edu> > *viewdoc* > *summary*, 2001.
- [59] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10m+ 3D objects. July 2023.
- [60] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. *ACM Trans. Graph.*, 37(4):1–15, August 2018.
- [61] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF capture with a multi-image deep network. *Comput. Graph. Forum*, 38(4):1–13, July 2019.
- [62] Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. *Comput. Graph. Forum*, 39(4):91–105, July 2020.

- [63] J-M Dischler, K Maritaud, B Lévy, and D Ghazanfarpour. Texture particles. *Comput. Graph. Forum*, 21(3):401–410, September 2002.
- [64] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: recovering spatially varying surface reflectance under unknown lighting. *ACM Trans. Graph.*, 33(6):1–12, November 2014.
- [65] Yue Dong, Jiaping Wang, Xin Tong, John Snyder, Yanxiang Lan, Moshe Ben-Ezra, and Baining Guo. Manifold bootstrapping for SVBRDF capture. In *ACM SIGGRAPH 2010 papers*, number Article 98 in SIGGRAPH ’10, pages 1–10, New York, NY, USA, July 2010. Association for Computing Machinery.
- [66] David Dunn, Cary Tippets, Kent Torell, Petr Kellnhofer, Kaan Akşit, Piotr Didyk, Karol Myszkowski, David Luebke, and Henry Fuchs. Wide field of view varifocal Near-Eye display using See-Through deformable membrane mirrors. *IEEE Trans. Vis. Comput. Graph.*, 23(4):1322–1331, April 2017.
- [67] Christian Eisenacher, Chuck Tappan, Brent Burley, Daniel Teece, and Arthur Shek. Example-based texture synthesis on disney’s tangled. In *ACM SIGGRAPH 2010 Talks*, number Article 32 in SIGGRAPH ’10, page 1, New York, NY, USA, July 2010. Association for Computing Machinery.
- [68] Anas El Rhammad, Patrick Gioia, Antonin Gilles, Marco Cagnazzo, and Béatrice Pesquet-Popescu. Color digital hologram compression based on matching pursuit. *Appl. Opt.*, 57(17):4930–4942, June 2018.
- [69] Oskar Elek, Denis Sumin, Ran Zhang, Tim Weyrich, Karol Myszkowski, Bernd Bickel, Alexander Wilkie, and Jaroslav Křivánek. Scattering-aware texture reproduction for 3D printing. *ACM Trans. Graph.*, 36(6):1–15, November 2017.
- [70] W S Elkhuisen, T T W Essers, B Lenseigne, C Weijkamp, Y Song, S C Pont, J M P Geraedts, and J Dik. Reproduction of gloss, color and relief of paintings using 3D scanning and 3D printing. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage*, GCH ’17, pages 183–187, Goslar, DEU, September 2017. Eurographics Association.
- [71] Willemijn S Elkhuisen, Tim Zaman, Wim Verhofstad, Pieter P Jonker, Joris Dik, and Jo MP Geraedts. Topographical scanning and reproduction of near-planar surfaces of paintings. In *Measuring, Modeling, and Reproducing Material Appearance*, volume 9018, page 901809. International Society for Optics and Photonics, 2014.
- [72] Cass Everitt. Interactive order-independent transparency. *gamedevs.org*, 2001.
- [73] M Hossein Eybposh, Nicholas W Caira, Mathew Atisa, Praneeth Chakravarthula, and Nicolas C Pégard. DeepCGH: 3D computer-generated holography using deep learning. *Opt. Express*, 28(18):26636–26650, 2020.

- [74] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [75] J R Fienup. Phase retrieval algorithms: a comparison. *Appl. Opt.*, 21(15):2758–2769, August 1982.
- [76] Scott S Fisher, John O Merritt, and Mark T Bolas, editors. *Spatial-light-modulator-based three-dimensional multiplanar display*, volume 3012 of *Stereoscopic Displays and Virtual Reality Systems IV*, February 1997.
- [77] Robert W. Floyd and Louis Steinberg. An adaptive algorithm for spatial greyscale. In *Proceedings of Society for Information Displays*, volume 17, pages 75–77, 1976.
- [78] B Galerne, A Leclaire, and L Moisan. Texton noise. *Comput. Graph. Forum*, 36(8):205–218, December 2017.
- [79] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. Gabor noise by example. *ACM Trans. Graph.*, 31(4):1–9, July 2012.
- [80] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Trans. Graph.*, 38(4):1–15, July 2019.
- [81] Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear light source reflectometry. *ACM Trans. Graph.*, 22(3):749–758, July 2003.
- [82] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, June 2016.
- [83] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.
- [84] GERCHBERG and R W. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.
- [85] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A Wilson, and Paul Debevec. Estimating specular roughness and anisotropy from second order spherical gradient illumination. *Comput. Graph. Forum*, 28(4):1161–1170, June 2009.
- [86] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A Wilson, and Paul Debevec. Circularly polarized spherical illumination reflectometry. In *ACM SIGGRAPH Asia 2010 papers*, number Article 162 in SIGGRAPH ASIA ’10, pages 1–12, New York, NY, USA, December 2010. Association for Computing Machinery.
- [87] Abhijeet Ghosh, Tim Hawkins, Pieter Peers, Sune Frederiksen, and Paul Debevec. Practical modeling and acquisition of layered facial reflectance, 2008.

- [88] Guillaume Gilet, Basile Sauvage, Kenneth Vanhoey, Jean-Michel Dischler, and Djamchid Ghazanfarpour. Local random-phase noise for procedural texturing. *ACM Trans. Graph.*, 33(6):1–11, November 2014.
- [89] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.*, 32(6):162:1–162:13, November 2013.
- [90] Daniel Glasner, Todd Zickler, and Anat Levin. A reflectance display. *ACM Trans. Graph.*, 33(4):61:1–61:12, July 2014.
- [91] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1060–1071, June 2010.
- [92] Joseph W Goodman. *Introduction to Fourier Optics*. Roberts and Company Publishers, 2005.
- [93] Joseph W Goodman. Holography viewed from the perspective of the light field camera. In *Fringe 2013*, pages 3–15. Springer Berlin Heidelberg, 2014.
- [94] Manu Gopakumar, Jonghyun Kim, Suyeon Choi, Yifan Peng, and Gordon Wetzstein. Unfiltered holography: optimizing high diffraction orders without optical filtering for compact holographic displays. *Opt. Lett.*, 46(23):5822–5825, December 2021.
- [95] Dar’ya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. BRDF Representation and Acquisition. *Computer Graphics Forum*, 2016.
- [96] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3D graphics. *ACM Trans. Graph.*, 31(6):1–10, November 2012.
- [97] Paul Guerrero, Miloš Hašan, Kalyan Sunkavalli, Radomír Měch, Tamy Boubekeur, and Niloy J Mitra. MatFormer: a generative model for procedural materials. *ACM Trans. Graph.*, 41(4):1–12, July 2022.
- [98] Jie Guo, Shuichang Lai, Chengzhi Tao, Yuelong Cai, Lei Wang, Yanwen Guo, and Ling-Qi Yan. Highlight-aware two-stream network for single-image SVBRDF acquisition. *ACM Trans. Graph.*, 40(4):1–14, July 2021.
- [99] Yu Guo, Milos Hasan, Lingqi Yan, and Shuang Zhao. A bayesian inference framework for procedural material parameter estimation. December 2019.
- [100] Yu Guo, Milos Hasan, Lingqi Yan, and Shuang Zhao. A bayesian inference framework for procedural material parameter estimation, 2019.
- [101] Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. MaterialGAN: reflectance capture using a generative SVBRDF model. *ACM Trans. Graph.*, 39(6):1–13, November 2020.

- [102] Eric Heitz and Fabrice Neyret. High-Performance By-Example noise using a Histogram-Preserving blending operator. *Proc. ACM Comput. Graph. Interact. Tech.*, 1(2):1–25, August 2018.
- [103] Philipp Henzler, Valentin Deschaintre, Niloy J Mitra, and Tobias Ritschel. Generative modelling of BRDF textures from flash images. *ACM Trans. Graph.*, 40(6):1–13, December 2021.
- [104] Jean Hergel and Sylvain Lefebvre. Clean color: Improving multi-filament 3D prints. *Comput. Graph. Forum*, 33(2):469–478, May 2014.
- [105] Kato Hiroharu, Ushiku Yoshitaka, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [106] Daisuke Hiyama, Tomoyoshi Shimobaba, Takashi Kakue, and Tomoyoshi Ito. Acceleration of color computer-generated hologram from RGB-D images using color space conversion. *Opt. Commun.*, 340:121–125, April 2015.
- [107] David M Hoffman, Ahna R Girshick, Kurt Akeley, and Martin S Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *J. Vis.*, 8(3):33–33, March 2008.
- [108] Michael Holroyd, Jason Lawrence, and Todd Zickler. A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. *ACM Trans. Graph.*, 29(4):1–12, July 2010.
- [109] Ryoichi Horisaki, Yohei Nishizaki, Katsuhisa Kitaguchi, Mamoru Saito, and Jun Tanida. Three-dimensional deeply generated holography [invited]. *Appl. Opt.*, 60(4):A323–A328, February 2021.
- [110] Ryoichi Horisaki, Ryosuke Takagi, and Jun Tanida. Deep-learning-generated holography. *Appl. Opt.*, 57(14):3859–3863, May 2018.
- [111] C K Hsueh and A A Sawchuk. Computer-generated double-phase holograms. *Appl. Opt.*, 17(24):3874–3883, December 1978.
- [112] Xinda Hu and Hong Hua. High-resolution optical see-through multi-focal-plane head-mounted display using freeform optics. *Opt. Express*, 22(11):13896–13903, June 2014.
- [113] Yiwei Hu, Julie Dorsey, and Holly Rushmeier. A novel framework for inverse procedural texture modeling. *ACM Trans. Graph.*, 38(6):1–14, November 2019.
- [114] Yiwei Hu, Julie Dorsey, and Holly Rushmeier. A novel framework for inverse procedural texture modeling. *ACM Trans. Graph.*, 38(6):1–14, November 2019.
- [115] Yiwei Hu, Paul Guerrero, Milos Hasan, Holly Rushmeier, and Valentin Deschaintre. Generating procedural materials from text or image prompts. In *ACM SIGGRAPH 2023 Conference Proceedings*, number Article 4 in SIGGRAPH ’23, pages 1–11, New York, NY, USA, July 2023. Association for Computing Machinery.

- [116] Yiwei Hu, Chengan He, Valentin Deschaintre, Julie Dorsey, and Holly Rushmeier. An inverse procedural modeling pipeline for SVBRDF maps. *ACM Trans. Graph.*, 41(2):1–17, January 2022.
- [117] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. DiffTaichi: Differentiable programming for physical simulation. *ICLR*, 2020.
- [118] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics*, 37(2):1–17, July 2018.
- [119] Hong Hua. Enabling focus cues in Head-Mounted displays. *Proc. IEEE*, 105(5):805–824, May 2017.
- [120] Hong Hua and Bahram Javidi. A 3D integral imaging optical see-through head-mounted display. *Opt. Express*, 22(11):13484–13491, June 2014.
- [121] Fu-Chung Huang, Kevin Chen, and Gordon Wetzstein. The light field stereoscope: immersive computer graphics via factored near-eye light field displays with focus cues. *ACM Trans. Graph.*, 34(4):1–12, July 2015.
- [122] Fu-Chung Huang, Dawid Pająk, Jonghyun Kim, Jan Kautz, and David Luebke. Mixed-primary factorization for dual-frame computational displays. *ACM Trans. Graph.*, 36(4):1–13, July 2017.
- [123] Fu-Chung Huang, Gordon Wetzstein, Brian A. Barsky, and Ramesh Raskar. Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays. *ACM Trans. Graph.*, 33(4):59:1–59:12, July 2014.
- [124] Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C Sankaranarayanan. Reflectance capture using univariate sampling of BRDFs, 2017.
- [125] Tsubasa Ichikawa, Kazuhiro Yamaguchi, and Yuji Sakamoto. Realistic expression for full-parallax computer-generated holograms with the ray-tracing method. *Appl. Opt.*, 52(1):A201–9, January 2013.
- [126] F Imai, M Rosen, and R Berns. Comparative study of metrics for spectral match quality. *Conf Colour Graph Imaging Vis*, pages 492–496, January 2002.
- [127] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5967–5976, 2017.
- [128] R Jagannathan, R Simon, E C G Sudarshan, and R Vasudevan. Dynamical maps and nonnegative phase-space distribution functions in quantum mechanics. *Phys. Lett. A*, 120(4):161–164, February 1987.

- [129] Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics (ToG)*, 33(4):118, 2014.
- [130] Changwon Jang, Kiseung Bang, Minseok Chae, Byoungcho Lee, and Douglas Lanman. Waveguide holography for 3D augmented reality glasses. *Nat. Commun.*, 15(1):66, January 2024.
- [131] Youngjin Jo, Dongheon Yoo, Dukho Lee, Minkwan Kim, and Byoungcho Lee. Multi-illumination 3d holographic display using a binary mask. *Optics Letters*, 47(10):2482–2485, 2022.
- [132] James T Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986.
- [133] Hoonjong Kang, Takeshi Yamaguchi, and Hiroshi Yoshikawa. Accurate phase-added stereogram to improve the coherent stereogram. *Appl. Opt.*, 47(19):D44–54, July 2008.
- [134] Kaizhang Kang, Cihui Xie, Chengan He, Mingqi Yi, Minyi Gu, Zimin Chen, Kun Zhou, and Hongzhi Wu. Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM Trans. Graph.*, 38(6):1–12, December 2019.
- [135] Isaac Kauvar, Samuel J Yang, Liang Shi, Ian McDowall, and Gordon Wetzstein. Adaptive color display via perceptually-driven factored spectral projection. *ACM Transactions on Graphics (TOG)*, 34(6):165, 2015.
- [136] K Kavaklı, Y Itoh, H Urey, and K Akşit. Realistic defocus blur for multiplane computer-generated holography. *2023 IEEE Conference*, 2023.
- [137] Koray Kavaklı, Liang Shi, Hakan Ürey, Wojciech Matusik, and Kaan Akşit. Holo-hdr: Multi-color holograms improve dynamic range, 2023.
- [138] Takayuki Kawaguchi, Norimichi Tsumura, Hideaki Haneishi, Yoichi Miyake, and M Kouzaki. Vector error diffusion method for spectral color reproduction. In *PICS*, pages 394–397, 1999.
- [139] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):1–12, July 2013.
- [140] Jonghyun Kim, Manu Gopakumar, Suyeon Choi, Yifan Peng, Ward Lopes, and Gordon Wetzstein. Holographic glasses for virtual reality. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.
- [141] D P Kingma and J Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [142] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [143] Hyunsuk Ko and Hui Yong Kim. Deep Learning-Based compression for Phase-Only hologram. *IEEE Access*, 9:79735–79751, 2021.
- [144] Robert Konrad, Emily A Cooper, and Gordon Wetzstein. Novel optical configurations for virtual reality: Evaluating user preference and performance with focus-tunable and monovision near-eye displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI ’16, pages 1211–1220, New York, NY, USA, May 2016. Association for Computing Machinery.
- [145] Tomasz Kozacki. On resolution and viewing of holographic image generated by 3D holographic display. *Opt. Express*, 18(26):27118–27129, December 2010.
- [146] Gregory Kramida. Resolving the Vergence-Accommodation conflict in Head-Mounted displays. *IEEE Trans. Vis. Comput. Graph.*, 22(7):1912–1931, July 2016.
- [147] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [148] P. Kubelka and F. Munk. Ein beitrag zur optik der farbanstriche. *Zeitschrift für technische Physik*, 12:593–601, 1931.
- [149] Ares Lagae, Peter Vangorp, Toon Lenaerts, and Philip Dutré. Procedural isotropic stochastic textures by example. *Comput. Graph.*, 34(4):312–321, August 2010.
- [150] Marc Lambooi, Marten Fortuin, Ingrid Heynderickx, and Wijnand IJsselsteijn. Visual discomfort and visual fatigue of stereoscopic displays: A review. *J. Imaging Sci. Technol.*, 53(3):30201–1–30201–14, May 2009.
- [151] Douglas Lanman and David Luebke. Near-eye light field displays. *ACM Trans. Graph.*, 32(6):220:1–220:10, November 2013.
- [152] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Trans. Graph.*, 25(3):735–745, July 2006.
- [153] Byounghyo Lee, Dongyeon Kim, Seungjae Lee, Chun Chen, and Byoungho Lee. High-contrast, speckle-free, true 3D holography via binary CGH optimization. *Sci. Rep.*, 12(1):2811, February 2022.
- [154] L Lefebvre and Pierre Poulin. Analysis and synthesis of structural textures. *Graphics Interface*, 2000.

- [155] Sylvain Lefebvre, Samuel Hornus, and Anass Lasram. By-example synthesis of architectural textures. In *ACM SIGGRAPH 2010 papers*, number Article 84 in SIGGRAPH '10, pages 1–8, New York, NY, USA, July 2010. Association for Computing Machinery.
- [156] D Leseberg and C Frère. Computer-generated holograms of 3-D objects composed of tilted planar segments. *Appl. Opt.*, 27(14):3020–3024, July 1988.
- [157] L B Lesem, P M Hirsch, and J A Jordan. The kinoform: A new wavefront reconstruction device. *IBM J. Res. Dev.*, 13(2):150–155, March 1969.
- [158] U Levy, E Marom, and D Mendlovic. Simultaneous multicolor image formation with a single diffractive optical element. *Optics letters*, 26(15):1149–1151, 2001.
- [159] Beichen Li, Liang Shi, and Wojciech Matusik. End-to-end procedural material capture with Proxy-Free Mixed-Integer optimization. *ACM Trans. Graph.*, 42(4):1–15, July 2023.
- [160] Rujia Li and Liangcai Cao. Progress in phase calibration for liquid crystal spatial light modulators. *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, 9(10):2012, May 2019.
- [161] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics*, 37(6):222:1–222:11, July 2018.
- [162] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Trans. Graph.*, 36(4):1–11, July 2017.
- [163] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 72–87, 2018.
- [164] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.*, 37(6):1–11, December 2018.
- [165] G Lippmann. Reversible events photos integrals. *Academic Accounts-Academic*, 146, 1908.
- [166] Sheng Liu, Hong Hua, and Dewen Cheng. A novel prototype for an optical See-Through Head-Mounted display with addressable focus cues. *IEEE Trans. Vis. Comput. Graph.*, 16(3):381–393, 2010.
- [167] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. NeRO: Neural geometry and BRDF reconstruction of reflective objects from multiview images. *ACM Trans. Graph.*, 42(4):1–22, July 2023.

- [168] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–169, 2014.
- [169] Gordon D Love, David M Hoffman, Philip J W Hands, James Gao, Andrew K Kirby, and Martin S Banks. High-speed switchable lens enables the development of a volumetric stereoscopic display. *Opt. Express*, 17(18):15716–15725, August 2009.
- [170] Mark Lucente. *Diffraction-specific fringe computation for electro-holography*. PhD thesis, Massachusetts Institute of Technology, 1994.
- [171] Mark Lucente. Interactive three-dimensional holographic displays: seeing the future in depth. *SIGGRAPH Comput. Graph.*, 31(2):63–67, May 1997.
- [172] Mark Lucente and Tinsley A. Galyean. Rendering interactive holographic images. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95*, pages 387–394. ACM, 1995.
- [173] Mark E. Lucente. Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1):28–34–7, 1993.
- [174] M Ronnier Luo, Guihua Cui, and B Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001.
- [175] Andrew Maimone, Andreas Georgiou, and Joel S Kollin. Holographic near-eye displays for virtual and augmented reality. *ACM Trans. Graph.*, 36(4):1–16, July 2017.
- [176] Andrew Maimone and Junren Wang. Holographic optics for thin and lightweight virtual reality. *ACM Trans. Graph.*, 39(4):67:1–67:14, July 2020.
- [177] Andrew Maimone, Gordon Wetzstein, Matthew Hirsch, Douglas Lanman, Ramesh Raskar, and Henry Fuchs. Focus 3d: Compressive accommodation display. *ACM Trans. Graph.*, 32(5):1–13, October 2013.
- [178] Ghaith Makey, Özgün Yavuz, Denizhan K. Kesim, Ahmet Turnali, Parviz Elahi, Serim Ilday, Onur Tokel, and F. Ömer Ilday. Breaking crosstalk limits to dynamic holography using orthogonality of high-dimensional random vectors. *Nature Photonics*, 13(4):251–256, 2019.
- [179] D Marpe, H Schwarz, and T Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):620–636, July 2003.
- [180] Nathan Matsuda, Alexander Fix, and Douglas Lanman. Focal surface displays. *ACM Trans. Graph.*, 36(4):1–14, July 2017.

- [181] Kyoji Matsushima. Computer-generated holograms for three-dimensional surface objects with shade and texture. *Appl. Opt.*, 44(22):4607–4614, August 2005.
- [182] Kyoji Matsushima and Sumio Nakahara. Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method. *Appl. Opt.*, 48(34):H54–63, December 2009.
- [183] Kyoji Matsushima, Masaki Nakamura, and Sumio Nakahara. Silhouette method for hidden surface removal in computer holography and its acceleration using the switch-back technique. *Opt. Express*, 22(20):24450–24465, October 2014.
- [184] Kyoji Matsushima and Tomoyoshi Shimobaba. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Opt. Express*, 17(22):19662–19673, October 2009.
- [185] Kyoji Matsushima and Tomoyoshi Shimobaba. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Opt. Express*, 17(22):19662–19673, Oct 2009.
- [186] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Trans. Graph.*, 22(3):759–769, July 2003.
- [187] Omel Mendoza-Yero, Gladys Mínguez-Vega, and Jesús Lancis. Encoding complex fields by using a phase-only optical element. *Opt. Lett.*, 39(7):1740–1743, Apr 2014.
- [188] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33, 2020.
- [189] Panagiotis Takis Metaxas. Optimal parallel error diffusion dithering. In *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts IV*, volume 3648, pages 485–495. International Society for Optics and Photonics, 1998.
- [190] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):1–14, July 2019.
- [191] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020*, pages 405–421. Springer International Publishing, 2020.
- [192] D Minnen, J Ballé, and G Toderici. Joint autoregressive and hierarchical priors for learned image compression. *arXiv preprint arXiv:1809.02736*, 2018.
- [193] P Morovic, J Morovič, Jordi Arnat, and J M García-Reyero. Revisiting spectral printing: A data driven approach. *Color Imaging Conf*, pages 335–340, January 2012.

- [194] Raees Kizhakkumkara Muhamad, Tobias Birnbaum, Antonin Gilles, Saeed Mahmoudpour, Kwan-Jung Oh, Manuela Pereira, Cristian Perra, Antonio Pinheiro, and Peter Schelkens. Jpeg pleno holography: scope and technology validation procedures. *Appl. Opt.*, 60(3):641–651, Jan 2021.
- [195] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical SVBRDF acquisition of 3D objects with unstructured flash photography. *ACM Trans. Graph.*, 37(6):1–12, December 2018.
- [196] Giljoo Nam, Joo Ho Lee, Hongzhi Wu, Diego Gutierrez, and Min H Kim. Simultaneous acquisition of microscale reflectance and normals. *ACM Trans. Graph.*, 35(6):1–11, November 2016.
- [197] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics*, 38(6):203:2–203:17, November 2019.
- [198] Victor Ostromoukhov. Chromaticity gamut enhancement by heptatone multicolor printing. In *Device-Independent Color Imaging and Imaging Systems Integration*, volume 1909, pages 139–151. SPIE, August 1993.
- [199] Nitish Padmanaban, Robert Konrad, Tal Stramer, Emily A Cooper, and Gordon Wetzstein. Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays. *Proceedings of the National Academy of Sciences*, 114(9):2183–2188, 2017.
- [200] Nitish Padmanaban, Yifan Peng, and Gordon Wetzstein. Holographic near-eye displays based on overlap-add stereograms. *ACM Trans. Graph.*, 38(6):1–13, November 2019.
- [201] Daniele Panozzo, Olga Diamanti, Sylvain Paris, Marco Tarini, Evgeni Sorkine, and Olga Sorkine-Hornung. Texture mapping real-world objects with hydrographics. *Comput. Graph. Forum*, 34(5):65–75, August 2015.
- [202] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans. Graph.*, 35(6):179, 2016.
- [203] Yifan Peng, Suyeon Choi, Jonghyun Kim, and Gordon Wetzstein. Speckle-free holography with partially coherent light sources and camera-in-the-loop calibration. *Science Advances*, 7(46):eabg5040, 2021.
- [204] Yifan Peng, Suyeon Choi, Nitish Padmanaban, and Gordon Wetzstein. Neural holography with camera-in-the-loop training. *ACM Trans. Graph.*, 39(6):1–14, 2020.
- [205] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, October 2016.

- [206] Weidong Qu, Huarong Gu, Hao Zhang, and Qiaofeng Tan. Image magnification in lensless holographic projection using double-sampling fresnel diffraction. *Appl. Opt.*, 54(34):10018–10021, December 2015.
- [207] Tim Reiner, Nathan Carr, Radomír Měch, Ondřej Št’ava, Carsten Dachsbacher, and Gavin Miller. Dual-color mixing for fused deposition modeling printers. *Comput. Graph. Forum*, 33(2):479–486, May 2014.
- [208] Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. Pocket reflectometry. *ACM Trans. Graph.*, 30(4):1–10, July 2011.
- [209] Zhenbo Ren, Ping Su, and Jianshe Ma. Information content compression and zero-order elimination of computer-generated hologram based on discrete cosine transform. *Opt. Rev.*, 20:469, November 2013.
- [210] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch, 2020.
- [211] J Riviere, P Peers, and A Ghosh. Mobile surface reflectometry. *Comput. Graph. Forum*, 35(1):191–202, February 2016.
- [212] J P Rolland, M W Krueger, and A Goon. Multifocal planes head-mounted displays. *Appl. Opt.*, 39(19):3209–3215, July 2000.
- [213] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015.
- [214] Mitchell R Rosen, Edward F Hattenberger, and Noboru Ohta. Spectral redundancy in a six-ink ink jet printer. *Journal of Imaging Science and Technology*, 48(3):194–202, 2004.
- [215] Romain Rossier. *Framework for Printing with Daylight Fluorescent Inks*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, 2013.
- [216] Filippus S Roux. Geometric phase lens. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.*, 23(2):476–482, February 2006.
- [217] Claude Rozé, Thierry Girasole, and Anne-Gaelle Tafforin. Multilayer four-flux model of scattering, emitting and absorbing media. *Atmospheric environment*, 35(30):5125–5130, 2001.
- [218] Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung. Computational thermoforming. *ACM Trans. Graph.*, 35(4):1–9, July 2016.

- [219] Young-Ho Seo, Hyun-Jun Choi, Ji-Sang Yoo, Gang-Seong Lee, Chung-Hyeok Kim, Seung-Hyun Lee, Sang-Hun Lee, and Dong-Wook Kim. Digital hologram compression technique by eliminating spatial correlations based on MCTF. *Opt. Commun.*, 283:4261, November 2010.
- [220] Jonathan Shade, Steven Gortler, Li-Wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, July 1998. Association for Computing Machinery.
- [221] Liang Shi, Vahid Babaei, Changil Kim, Michael Foshey, Yuanming Hu, Pitchaya Sitthi-Amorn, Szymon Rusinkiewicz, and Wojciech Matusik. Deep multispectral painting reproduction via multi-layer, custom-ink printing. *ACM Trans. Graph.*, 37(6):1–15, December 2018.
- [222] Liang Shi, Fu-Chung Huang, Ward Lopes, Wojciech Matusik, and David Luebke. Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3D computer graphics. *ACM Trans. Graph.*, 36(6):1–17, November 2017.
- [223] Liang Shi, Beichen Li, Miloš Hašan, Kalyan Sunkavalli, Tamy Boubekeur, Radomir Mech, and Wojciech Matusik. Match: differentiable material graphs for procedural material capture. *ACM Trans. Graph.*, 39(6):1–15, November 2020.
- [224] Liang Shi, Beichen Li, Changil Kim, Petr Kellnhofer, and Wojciech Matusik. Towards real-time photorealistic 3D holography with deep neural networks. *Nature*, 591(7849):234–239, 2021.
- [225] Liang Shi, Beichen Li, and Wojciech Matusik. End-to-end learning of 3D phase-only holograms for holographic display. *Light Sci Appl*, 11(1):247, August 2022.
- [226] Liang Shi, Beichen Li, and Wojciech Matusik. End-to-end learning of 3D phase-only holograms for holographic display. *Light Sci Appl*, 11(1):247, August 2022.
- [227] Liang Shi, Donghun Ryu, and Wojciech Matusik. Ergonomic-centric holography: Optimizing realism, immersion, and comfort for holographic display. *Laser Photon. Rev.*, 18(4), April 2024.
- [228] Takashi Shibata, Joohwan Kim, David M Hoffman, and Martin S Banks. The zone of comfort: Predicting visual discomfort with stereo displays. *J. Vis.*, 11(8):11, July 2011.
- [229] Tomoyoshi Shimobaba, Nobuyuki Masuda, and Tomoyoshi Ito. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane. *Opt. Lett.*, 34(20):3133–3135, Oct 2009.
- [230] Shinichi Shiwa, Katsuyuki Omura, and Fumio Kishino. Proposal for a 3-d display with accommodative compensation: 3DDAC. *J. Soc. Inf. Disp.*, 4(4):255–261, December 1996.

- [231] Alison E Shortt, Thomas J Naughton, and Bahram Javidi. A companding approach for nonuniform quantization of digital holograms of three-dimensional objects. *Opt. Express, OE*, 14(12):5129–5134, June 2006.
- [232] Alison E Shortt, Thomas J Naughton, and Bahram Javidi. Compression of digital holograms of three-dimensional objects using wavelets. *Opt. Express, OE*, 14(7):2625–2630, April 2006.
- [233] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.
- [234] Pitchaya Sitthi-Amorn, Javier E. Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, and Wojciech Matusik. Multifab: A machine vision assisted platform for multi-material 3d printing. *ACM Trans. Graph.*, 34(4):129:1–129:11, July 2015.
- [235] Vincent Sitzmann, Steven Diamond, Yifan Peng, Xiong Dun, Stephen Boyd, Wolfgang Heidrich, Felix Heide, and Gordon Wetzstein. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Trans. Graph.*, 37(4):1–13, July 2018.
- [236] Quinn Y J Smithwick, James Barabas, Daniel E Smalley, and V Michael Bove, Jr. Real-time shader rendering of holographic stereograms. In *Practical Holography XXIII: Materials and Applications*, volume 7233, page 723302. International Society for Optics and Photonics, February 2009.
- [237] Adrian Stern and Bahram Javidi. Sampling in the light of wigner distribution. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.*, 21(3):360–366, March 2004.
- [238] Stratasys. Stratasys J750 the ultimate full-color multi-material 3D printer. <http://www.stratasys.com/3d-printers/production-series/stratasys-j750>, 2016. [Online; accessed 15-October-2016].
- [239] Toshiaki Sugihara and Tsutomu Miyasato. System development of fatigue-less hmd system 3ddac (3d display with accommodative compensation: system implementation of mk. 4 in light-weight hmd. In *ITE Technical Report 22.1*, pages 33–36. The Institute of Image Information and Television Engineers, 1998.
- [240] Xiaomeng Sui, Zehao He, Guofan Jin, Daping Chu, and Liangcai Cao. Band-limited double-phase method for enhancing image sharpness in complex modulated computer-generated holograms. *Opt. Express*, 29(2):2597–2612, January 2021.
- [241] Q Sun, F C Huang, J Kim, L Y Wei, D Luebke, and others. Perceptually-guided foveation for light field displays. *ACM Trans. Graph.*, 36(6):1–13, November 2017.
- [242] 3D Systems. Projet 5500x multi-material 3d printer, 2013.

- [243] Ayush Tewari, Michael Zollhöfer, Hyeonwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3735–3744, 2017.
- [244] Shoji Tominaga. Color control using neural networks and its application. In *Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, volume 2658, pages 253–261. International Society for Optics and Photonics, 1996.
- [245] Robert Toth, Jon Hasselgren, and Tomas Akenine-Möller. Perception of highlight disparity at a distance in consumer head-mounted displays. In *Proceedings of the 7th Conference on High-Performance Graphics*, HPG '15, pages 61–66, New York, NY, USA, 2015. ACM.
- [246] Peter Tsang, K W K Cheung, T-C Poon, and Changhe Zhou. Demonstration of compression ratio of over 4000 times for each digital hologram in a sequence of 25 frames in a holographic video. *J. Opt.*, 14:125403, December 2012.
- [247] Peter Tsang, Kayton W K Cheung, and Ting-Chung Poon. Low-bit-rate computer-generated color fresnel holography with compression ratio of over 1600 times using vector quantization [invited]. *Appl. Opt.*, 50(34):H42–H49, December 2011.
- [248] Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Trans. Graph.*, 32(4):1–12, July 2013.
- [249] Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. ControlMat: A controlled generative approach to material capture. September 2023.
- [250] Koki Wakunami and Masahiro Yamaguchi. Calculation for computer generated hologram using ray-sampling plane. *Opt. Express*, 19(10):9086–9101, May 2011.
- [251] David R Walton, Koray Kavaklı, Rafael Kuffner dos Anjos, David Swapp, Tim Weyrich, Hakan Urey, Anthony Steed, Tobias Ritschel, and Kaan Akşit. Metameric varifocal holography. *arXiv*, October 2021.
- [252] Jiaping Wang, Shuang Zhao, Xin Tong, John Snyder, and Baining Guo. Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Trans. Graph.*, 27(3):1–9, August 2008.
- [253] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, April 2004.
- [254] James P Waters. Holographic image synthesis utilizing theoretical methods. *Appl. Phys. Lett.*, 9(11):405–407, December 1966.

- [255] Hongzhi Wu, Zhaotian Wang, and Kun Zhou. Simultaneous localization and appearance estimation with a consumer RGB-D camera. *IEEE Trans. Vis. Comput. Graph.*, 22(8):2012–2023, August 2016.
- [256] Gunter Wyszecki and Walter Stanley Stiles. *Color Science*, volume 8. Wiley New York, 1982.
- [257] L Xiao, S Nouri, M Chapman, A Fix, D Lanman, and others. Neural supersampling for real-time rendering. *ACM Transactions on*, 2020.
- [258] Lei Xiao, Anton Kaplanyan, Alexander Fix, Matthew Chapman, and Douglas Lanman. DeepFocus: learned image synthesis for computational displays. *ACM Trans. Graph.*, 37(6):1–13, December 2018.
- [259] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal BRDF sampling for two-shot near-field reflectance acquisition. *ACM Trans. Graph.*, 35(6):1–12, November 2016.
- [260] Y K Lam Y. K. Lam, W C Situ W. C. Situ, and P W M Tsang P. W. M. Tsang. Fast compression of computer-generated holographic images based on a GPU-accelerated skip-dimension vector quantization method. *Chin. Opt. Lett.*, 11(5):050901–050905, 2013.
- [261] Masahiro Yamaguchi, Hideshi Hoshino, Toshio Honda, and Nagaaki Ohyama. Phase-added stereogram: calculation of hologram using computer graphics technique. In *Practical Holography VII: Imaging and Materials*, volume 1914, pages 25–31. International Society for Optics and Photonics, September 1993.
- [262] Kai Yan, Fujun Luan, Miloš Hašan, Thibault Groueix, Valentin Deschaintre, and Shuang Zhao. PSDR-Room: Single photo to scene using differentiable rendering. In *SIGGRAPH Asia 2023 Conference Papers*, number Article 28 in SA '23, pages 1–11, New York, NY, USA, December 2023. Association for Computing Machinery.
- [263] Genzhi Ye, Sundeep Jolly, V Michael Bove, Qionghai Dai, Ramesh Raskar, and Gordon Wetzstein. Toward BxDF display using multilayer diffraction. *ACM Trans. Graph.*, 33(6):1–14, November 2014.
- [264] Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Single image surface appearance modeling with self-augmented CNNs and inexact supervision. *Comput. Graph. Forum*, 37(7):201–211, October 2018.
- [265] Yu-Ying Yeh, Zhengqin Li, Yannick Hold-Geoffroy, Rui Zhu, Zexiang Xu, Milos Hasan, Kalyan Sunkavalli, and Manmohan Chandraker. PhotoScene: Photorealistic material and lighting transfer for indoor scenes. June 2022.
- [266] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–644. IEEE, July 2017.

- [267] Jiyang Yu, Zexiang Xu, Matteo Mannino, Henrik Wann Jensen, and Ravi Ramamoorthi. Sparse sampling for image-based SVBRDF acquisition, 2016.
- [268] Marina Zannoli, Gordon D. Love, Rahul Narain, and Martin S. Banks. Blur and the perception of depth at occlusions. *Journal of Vision*, 16(6):17, 2016.
- [269] Alejandro Velez Zea, Astrid Lorena Villamizar Amado, Myrian Tebaldi, and Roberto Torroba. Alternative representation for optimized phase compression in holographic data. *OSA Continuum, OSAC*, 2(3):572–581, March 2019.
- [270] Hao Zhang, Liangcai Cao, and Guofan Jin. Computer-generated hologram with occlusion effect using layer-based processing. *Appl. Opt.*, 56(13):F138–F143, May 2017.
- [271] Hao Zhang, Liangcai Cao, and Guofan Jin. Three-dimensional computer-generated hologram with fourier domain segmentation. *Opt. Express*, 27(8):11689–11697, April 2019.
- [272] Hao Zhang, Yan Zhao, Liangcai Cao, and Guofan Jin. Fully computed holographic stereogram based algorithm for computer-generated holograms with accurate depth cues. *Opt. Express*, 23(4):3901–3913, Feb 2015.
- [273] Hao Zhang, Yan Zhao, Liangcai Cao, and Guofan Jin. Fully computed holographic stereogram based algorithm for computer-generated holograms with accurate depth cues. *Opt. Express*, 23(4):3901–3913, February 2015.
- [274] Jingzhao Zhang, Nicolas Pégard, Jingshan Zhong, Hillel Adesnik, and Laura Waller. 3D computer-generated holography by non-convex optimization. *Optica, OPTICA*, 4(10):1306–1313, October 2017.
- [275] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, January 2018.
- [276] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. NeRFactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6):1–18, December 2021.
- [277] Yizhong Zhang, Yiyong Tong, and Kun Zhou. Coloring 3D printed surfaces by thermoforming. *IEEE Trans. Vis. Comput. Graph.*, 23(8):1924–1935, August 2017.
- [278] Yizhong Zhang, Chunji Yin, Changxi Zheng, and Kun Zhou. Computational hydrographic printing. *ACM Trans. Graph.*, 34(4):1–11, July 2015.
- [279] Zhengyun Zhang and Marc Levoy. Wigner distributions and how they relate to the light field. In *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, April 2009.

- [280] Yan Zhao, Liangcai Cao, Hao Zhang, Dezhao Kong, and Guofan Jin. Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method. *Optics express*, 23(20):25440–25449, 2015.
- [281] Yan Zhao, Liangcai Cao, Hao Zhang, Dezhao Kong, and Guofan Jin. Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method. *Opt. Express*, 23(20):25440–25449, Oct 2015.
- [282] Xilong Zhou and Nima Khademi Kalantari. Adversarial single-image SVBRDF estimation with hybrid training. *Comput. Graph. Forum*, 40(2):315–325, May 2021.
- [283] Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. Sparse-as-possible SVBRDF acquisition. *ACM Trans. Graph.*, 35(6):1–12, November 2016.