Discovering, Learning, and Exploiting Visual Cues

by

Kushagra Tiwary

B.S., University of Illinois- Urbana Champaign (2019)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2023

AUTHOR

Kushagra Tiwary
Program in Media Arts and Sciences
May 19, 2023

CERTIFIED BY

Ramesh Raskar
Associate Professor of Media Arts and Sciences
Thesis Supervisor

ACCEPTED BY

Tod Machover
Academic Head
Program in Media Arts and Sciences

Discovering, Learning, and Exploiting Visual Cues

by

Kushagra Tiwary

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on May 18, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Animals have evolved over millions of years to exploit the faintest visual cues for perception, navigation, and survival. Complex and intricate vision systems found in animals, such as bee eyes, exploit cues like polarization of light relative to the Sun's position to navigate and process motion at $\frac{1}{300}^{th}$ of a second. In humans, the evolution of the eyes and the processing of visual cues are also tightly intertwined. Babies develop depth-of-field at 6 months, are often scared of their own shadows, and confuse their reflections with the real world. As the infant matures into an adult, they intuitively learn from their experiences how these cues instead provide valuable hidden information about their environments and can be exploited for depth perception and driving.

Inspired by our usage of visual cues, this thesis explores visual cues in the modern context of data-driven imaging techniques. We first explore how visual cues can be learned from and exploited by combining physics-based forward models with data-driven AI systems. We first map the space of physics-based and data-driven systems and show the future of vision lies in the intersection of both regimes. Next, we show how shadows can be exploited to image and 3D reconstruct the hidden parts of the scene. We then exploit multi-view reflections to convert household objects into radiance-field cameras that can image the world from the object's perspective in 5D. This enables applications of occlusion imaging, beyond field-of-view novel-view synthesis, and depth estimation from objects to their environments.

Finally, we discuss how current approaches rely on humans to design imaging systems that can learn and exploit visual cues. However, as sensing in space, time, and different modalities become ubiquitous, relying on human-designed systems is not sufficient to build complex vision systems. We then propose

a technique that combines reinforcement learning with computer vision to automatically learn which cues to exploit to accomplish the task without human intervention. We show how in one such scenario agents can start to automatically learn to use multiple cameras and the triangulation cue to estimate the depth of an unknown object in the scene without access to prior information about the camera, the algorithm, or the object.

Thesis Supervisor: Ramesh Raskar
Title: Professor of Media Arts and Sciences

Discovering, Learning, and Exploiting Visual Cues

by

Kushagra Tiwary

This thesis has been reviewed and approved by the following committee members:

THESIS SUPERVISOR

Ramesh Raskar
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

THESIS READER

Fadel Adib
Associate Professor of Media Arts and Sciences and of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

THESIS READER

Pulkit Agarwal
Steven and Renee Chair Professor
Massachusetts Institute of Technology

ACKNOWLEDGMENTS

This thesis would not be possible without the unconditional support and love from my parents, Devendra Nath Tiwary and Jaya Tiwary. They have always nurtured and given me the courage to explore my curiosity by creating an atmosphere that emphasized learning, perseverance, and spirituality which I will forever hold on to and be grateful for. Long walks and discussions with my father have been a source of inspiration and remain the reason for my love of science, engineering, and philosophy. My mother has a way of making us feel at home throughout our adventures across different countries in the world.

I am also thankful to Sudeep Bhardwaj, Vineeta Bhardwaj, and Ayush Bhardwaj for their unequivocal support, love, and affection. I cherish my trips to Chicago and our long drives filled with stories. My grandfather, Ashutosh Dubey, has been a constant source of learning since I was a little kid through chess, existential questions about time and physics, or Indian classical music. I am also thankful for other members of my family as well.

I would like to acknowledge Prof. Ramesh Raskar who has had an incredible impact on me over the past two years. Through him, I have learned much about research and life. In research, he has taught me how to pick impactful problems while at the same time being intellectually fearless at the same time- *"good ideas and great ideas often take the same amount of time"*. This has led to holistic growth these past two years. I am also thankful to Professor Pulkit Agarwal and Professor Fadel Adib for their insightful conversations and their time as thesis readers.

I am grateful for the ethos of Camera Culture and the students and staff in the lab for creating a welcoming and stimulating environment. During my time here, I have worked the most with Tzofi Klinghoffer and Sid Somasundaram. I have enjoyed our discussions, paper-deadline dinners, and conference trips, and hope to continue our close collaborations. I have also deeply enjoyed my philosophical discussions with Abhishek Singh. I am also thankful to Ayush Chopra for our long walks, impromptu dinners, squash games, and midnight discussions. In addition, I want to thank my collaborators: Nikhil Behari, Akshat Dave, and Bhavya Aggarwal. I am grateful for all the advice from Connor Henley, Praneeth Vepakomma, Subhash Chandra Sadhu, and Tristan Swedish.

Finally, I would also like to acknowledge the MIT Media Lab for its unique culture, and people. I am grateful and cherish Muddy with Wazeer Zulfikar, Hou Jason, Shen Jocelyn, and Aastha Shah, and hope to continue our traditions as we venture deeper into our PhDs. I am also thankful for the pseudo-randomness with Tobin South, Naana Obeng-Marnu, and Robert Mahari. I would also like to thank Ila Krishna Kumar, Kimaya Lecamwasam, Shayne Longpre, Ingrid Mantilla, Suyash Fulay, Kevin Dunnell, Cassandra Lee, and the rest of the MAS for all the laughs and learnings. Lastly, I am also thankful for the love and support of people outside the Media Lab: *613*, *rocket*, Anisha Karim, Ana Trapero, Charles Boury, and Arjun Balasingham. Finally, it'd be remiss if I don't mention a quote passed down by my father that I often look back to, *"utho parth, gandiv sambhalo"*.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

A | INTRODUCTION

Animal eyes have evolved for millions of years to exploit the faintest of visual cues in their surrounding in unique ways in order to overcome challenges posed by their environment. These biological vision systems have evolved from single-cell photoreceptors to complex compound eyes that can capture the environment beyond the limited visible spectrum available to humans, including infrared, ultraviolet, and even polarization. Moreover, animal eyes have also evolved to perform complex actions such as navigation, and detection of food and prey. For example, bee vision has evolved into a complex set of 5 eyes that use polarized light relative to the Sun's position to navigate and process motion at $\frac{1}{300}^{th}$ of a second. Moreover, their vision has further evolved to exploit the ultraviolet spectrum which gives them an advantage when seeking nectar.

On the contrary, human-designed perceptual systems are limited in sensing and analysis of visual cues for applications: i.e. human-designed vision systems capture limited information from the electromagnetic spectrum, analyze from a limited set of visual cues that are present in the scene, and, lastly, process at slower speeds compared to their biological counterparts. Moreover, in the modern context of data-driven vision techniques, visual cues are often second-class citizens- often ignored even if they provide valuable and hidden information about the scene.

We first explore how *known* visual cues can be parameterized in a modern machine-learning framework by using physical equations that govern the effects and interactions between light and matter. Next, the thesis shows applications of this framework. In particular, we show how the physics of two visual cues: shadows and reflections, and a property of light: polarization, can be integrated with modern data-driven neural rendering techniques to image the invisible parts of the scene, recover the 3D shape of objects from limited views, and create objects into radiance-field cameras. We then discuss how other cues can also be integrated with data-driven techniques. Finally, we show that this framework is limited by pre-discovered and known visual cues as it relies on explicit definitions of forward models of those cues by humans. The "designer-in-the-loop" must discover a cue in the imaging modality that can be exploited

through trial-and-error, and propose a forward model that can be integrated with machine learning. We discuss how such a system limits the evolution of perception systems that can see deeper inside our bodies, brains, and beyond.

In summary, the thesis addresses the following questions:

- How can the physics of light-matter interactions be merged with modern data-driven vision techniques?
- How can visual cues such as shadows, reflections, and polarization be exploited to image the invisible parts of the scene?
- How can we build imaging systems that don't rely on well-known and prediscovered visual cues for perception?

While the individual areas of machine learning, and exploiting visual cues through physics-based forward models are heavily studied in isolation, the thesis explores how those two fields can be combined which has recently emerged as an open research topic. The thesis consists of the culmination of the following works:

- Towards Neural Representations Through Shadows, *ECCV 2022* [107]
- Physics Vs. Learned Priors: Rethinking Camera and Algorithm Design for Task-Specific Imaging, *ICCP 2022* [48]
- ORCa: Glossy Objects as Radiance Field Cameras, *CVPR 2023* [106]
- DiSR: Discovery Imaging Systems Through Reinforcement Learning, *Under Review, ICCV 2023*

A.1 ANIMAL EYES

The evolution of our visual system is remarkable, and fascinating, and is deeply linked with the evolution of lifeforms that reside on earth today. We define the visual system with an eye as the sensor that focuses incoming light and converts it into electrical impulses, and the brain which processes the electrical impulses and makes "sense" of the electrical stimuli. The human eye today uses 100 million light-sensitive cells to "see", but the first eye traces its origins back to the emergence of single-celled organisms with rudimentary photoreceptor proteins- a single light-sensitive cell to "see". Early life forms such as the

cyanobacteria used light sensitivity to optimize their energy production using oxygenic photosynthesis- where the water is oxidized with the energy of the absorbed sunlight and Co2 is reduced to the level of energy-rich carbohydrates. This conversion from sunlight to chemical energy is one of the most important energy conversions on earth and is central to life formation.

Over time, single-cell photoreceptors evolved and became more complex, giving rise to the eyespots, pit eyes, and pinhole eyes found in more complex organisms. These eyespots and pit eyes gave life directional sensitivity to light in addition to sharper vision- mimicking a “pinhole" camera. The next leap in visual sensing abilities comes from the evolution of eyespots into compound eyes that are typically found in arthropods and crustaceans. The compound eye consists of thousands of individual photoreceptor units called ommatidia. These tiny individual photoreceptor units are spread along the surface, typically convex, of the compound eyes where each individual unit accepts light from a slightly different direction. This allows for a larger field of view in addition to serving as an extremely fast motion detector. The visual perception is then a mosaic- some combination of light coming in from each of the individual units.

A.2 VISUAL CUES

The evolution of the eyes and the processing of visual cues are tightly intertwined. As eyes became more sophisticated, the brain co-evolved to utilize a wide range of visual cues for navigation, communication, and survival. For example, forward-facing eyes allow for stereopsis or binocular vision to see and judge depth, vs. side-facing eyes allow for a larger peripheral vision (useful for detecting predators when grazing).

Humans rely heavily on visual cues such as binocular cues, or monocular cues such as shape, size, color, or shading. Our visual system has evolved to detect small inconsistencies in lighting and shadows in the scene and extract the most meaning from the visual scene. However, what is remarkable is that humans are not born with this capability. Even with the “hardware" in place, our brain learns how to perceive the world.

Toddlers begin to learn and recognize their own shadows from a young age often being scared or fascinated by light and shadows. Fig. 1.a shows how the toddler is scared of her own shadows as it is following her around. We also notice a similar behavior with reflections as babies and toddlers have yet to figure out how to process reflections of themselves onto mirrors. For

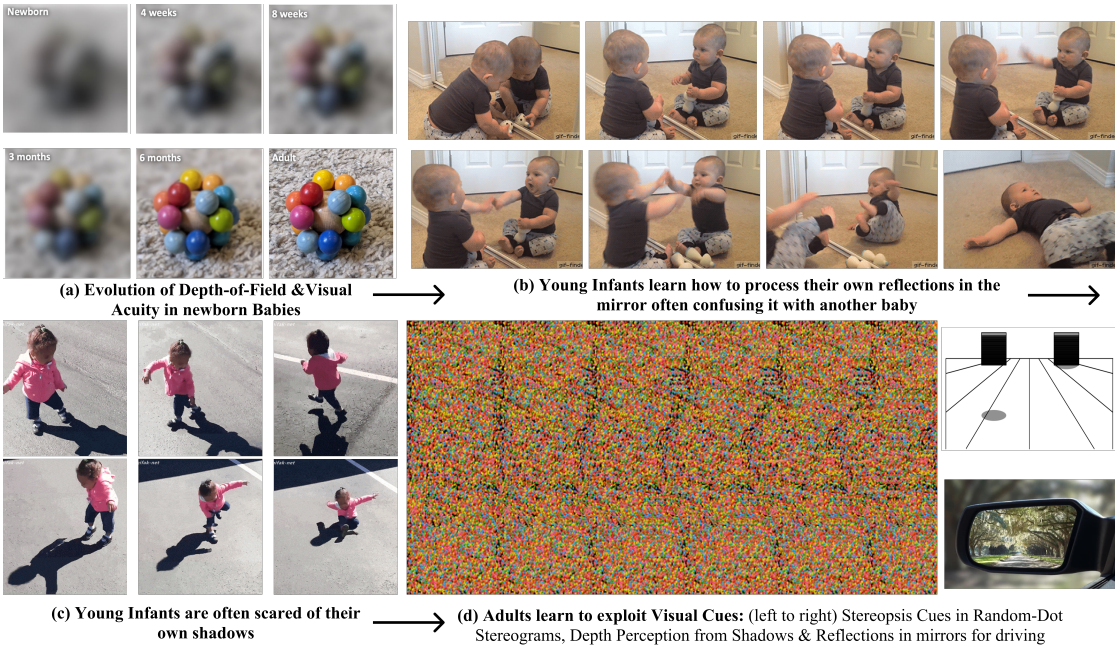

Figure 1: Exploiting Visual Cues at Different Stages in Humans. Toddlers and young infants are still learning how to exploit visual cues around them such as shadows and reflections. (a) shows how visual acuity evolves in newborn babies. This is often related to depth-of-field- the near image plane, roughly the distance from their mother's lap to the baby, is usually the first to come in focus. In (b) the infant has yet to pass the "Mirror Test" where young infants and toddlers are placed in front of mirrors to see how they respond- they often confuse it with another baby and have to learn to identify that it's their own reflections. In (c) the toddler thinks the shadow is chasing her, tries to run away, and eventually falls down. Babies are often very curious about light and visual stimuli and have yet to learn how shadows are and a consequence of lack of light. In (d) we show how adults have learned to exploit visual cues that they were once afraid or curious about. We show stereopsis cues using Random-Dot Stereograms, depth perception through shadows, and using reflections of mirrors for complex tasks of driving a car. *As a fun experiment, try to find the shark in the random-dot stereogram image (d). Hint: try to focus your eyes behind the image instead of on the image itself.*

example, psychologists often use the mirror test to recognize if the infants have understood that it is their reflection in the mirror. Young infants often think there's another baby in the mirror, while older babies are more hesitant. Toddlers typically begin to understand this with clear signals, such as they touch their own nose instead of the one in the mirror. Fig. 1.b the baby thinks

that there is another baby behind the mirror and tries to grab them. Fig. 1.a shows how depth of field and visual acuity evolve in newborn babies. in Fig. 1.d we finally show how adults have learned how to exploit these visual cues for depth perception (shadows and binocular cues) and for complex tasks such as driving (reflections on side mirrors to perform complex maneuvers). As a fun experiment, try to find the shark in the random-dot stereogram image (Fig. 1.d). *Hint: try to focus your eyes behind the image instead of on the image itself.*

Within a few years, the babies would have grown to learn to use the shadows to estimate object size and by 16 learn how to use reflections on side-car mirrors to drive and judge distances of other cars around them.

A.3 PARAMETRIZING VISUAL CUES USING LIGHT TRANSPORT

The importance of these cues for our visual system cannot be understated. Engineers, neuroscientists, and artists have spent many years understanding visual cues for applications in graphics, understanding the human visual system, and for effects in art and paintings. Here we take a computational perspective on visual cues and discuss how light transport can simulate and therefore perform computation on visual cues such as shadows, reflections, triangulation, and polarization.

There are many ways to model the behavior of light: ray optics, wave optics, electromagnetic optics, and quantum optics [43]. Reflection and refraction cues can be characterized by modeling light as rays, other cues like interference or polarization need to be modeled light as a wave. In this thesis, we deal with the behavior of light as rays. The rendering equation [45] models this behavior. It expresses the outgoing radiance, $L_\mathrm{o}\left(\mathbf{x}, \omega_\mathrm{o}\right)$, in the direction w at a point $\mathbf{x}$ as a sum of the emitted radiance: $\mathbf{L}_e(\mathbf{x}, \omega_\mathrm{o})$, and the total reflected radiance in the direction w at a point $\mathbf{x}$: $\mathbf{L}_r(\mathbf{x}, \omega_\mathrm{o})$. Here we have expanded $\mathbf{L}_r$ to be the sum of all the incoming radiance over a hemisphere Ω around the point $\mathbf{x}$.

$$L_\mathrm{o}\left(\mathbf{x}, \omega_\mathrm{o}\right) = L_\mathrm{e}\left(\mathbf{x}, \omega_\mathrm{o}\right) + \int_\Omega f_\mathrm{r}\left(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o}\right) L_\mathrm{i}\left(\mathbf{x}, \omega_\mathrm{i}\right)\left(\omega_\mathrm{i} \cdot \mathbf{n}\right) \mathrm{d}\omega_\mathrm{i} \tag{1}$$

Shadows. Equation 1 enables physically accurate renderings of scenes typically found in everyday life. Each ray that is traced must originate from a light source. A point is in hard shadow, for example, when there is no direct path from that point to a light source in the scene. Therefore by placing an

object in the scene, we restrict certain areas to have no path to the light source, but those same points have a ray towards the camera (if the cameras can see the shadows). Softer shadows form when those points have rays from some light sources and don't from others which cause those areas to be less well-lit when compared to other regions. These consistencies between light and camera sources in the scene have been exploited in graphics and inverse graphics for years to enable faster renderings [20]. The thesis will show that these physically accurate consistencies can also be exploited with machine learning methods to enable learning from shadows.

Reflections. Due to the recursive nature of the rendering equation, we can interchange and consider the point as a light source and a camera. Consider light originating from the sun and reflecting off a mirror onto a desk. We can render the same scene using the mirror as a light source instead of the sun. Moreover, we could also map the whole environment around the desk onto a hemisphere and render the scene using that hemisphere. Such approximation and techniques enable faster renderings as we can now start to approximate the scene and limit our recursion to a few bounces. The dual applies to the camera as well: consider the same scene but instead of rendering the scene from the camera looking at the desk, we use the reflection of the desk on the mirror and render it from the mirror's perspective. The thesis will also show that mapping between camera pixels and the object surface enables the conversion of the object into a virtual sensor. This enables the virtual sensor to estimate the depth of the scene from that object- effectively turning the objects in the scene into cameras.

A.4 OVERVIEW OF CONTRIBUTIONS

Finally, the thesis is organized in the following way:

- Chapter 3 proposes a framework to design task-specific cameras based on a combination of physics and data-driven methods. We show how the landscape of camera and algorithm design has evolved over time, and outline trends in the area. In addition, we also plot computational imaging, and computer vision among other fields based on how much "physics" and "data" they use.

- Chapter 4 shows how classical graphics techniques such as shadow mapping can be exploited with modern data-driven techniques such as Neural Radiance Fields (NeRFs) to image the hidden parts of the scene.
- Chapter 5 shows how physics-based methods in Computational Imaging such as Catadioptric Imaging Systems (CIS) can be exploited with NeRFs to convert objects in the scene into virtual cameras that can image the scene itself. We also show how polarization as a cue can also be exploited in this framework.
- Chapter 6 shows proposes a Computational Imaging Grammar and shows how the space of imaging and perception algorithms can be searched using reinforcement learning. This enables automatic learning of these cues without human supervision and thus a lesser dependency on humans to design imaging and perception systems.

In summary, the thesis explores the intersection of graphics and vision, and computational imaging and vision, and proposes tools. The number of visual cues and modalities is far too many to be addressed within the work, but my hope is that some of the tools and frameworks discussed in this thesis can lead to other cues being used to solve problems in different fields.

B PRELIMINARIES

The background is divided within the fields of Imaging and Graphics, and Computer Vision. Computer Vision is really an inverse graphics problem and serves to invert the forward models that govern light-matter interactions. We discuss first how visual cues are defined in graphics and computational imaging, followed by related works in modern data-driven computer vision. This thesis aims to bridge these worlds for visual cues such as shadows, reflections, and polarization.

B.1 VISUAL CUES IN IMAGING & GRAPHICS

Shape from Shadows. Shadowgram imaging deals with estimating the shape of an object through a sequence of shadow masks captured with light sources at various locations. These methods typically assume a controlled and fixed object scanning setup [89] [121]. Martin & Aggarwal [65] introduced a volumetric space carving approach to SfS which outputs a visual hull around the object by carving out voxels lying outside the visual cone. Other work takes a more probabilistic approach to the shape-from-silhouettes problem to make the algorithm more robust to errors [53]. However, interpreting shadows as silhouettes means that self-shadows are not handled, thus motivating Savarese et al. [89] to propose a method to “carve" out objects based on self-shadows to create more complete reconstructions.

B.1.1 Shadows in Graphics

Graphics deals with the forward model and shadow mapping [119] is one of the most efficient techniques to render shadows in a scene given the scene’s geometry, camera viewpoint and light position. While differentiability is not important for graphics, we make the shadow mapping framework differentiable to work with modern 3D reconstruction algorithms.

B.1.2 Catadioptric imaging systems (CIS)

Catadioptric imaging systems incorporate reflections on curved reflective mirrors to expand the field of view of conventional cameras [4, 72], to increase the baseline of light field cameras [23, 102] and to perform novel view synthesis from a single capture [117]. These works assume the geometry of the reflecting surface is known or calibrated while we create a catadioptric imaging system from everyday glossy objects of unknown geometry. Grossberg et al. [33] propose a generalized model for light transport through imaging systems, including catadioptric systems. Our work focuses on light transport reflecting off a general object in the scene.

B.1.3 Light field imaging

Light field imaging is shown to be effective for reflection removal [73, 55], reconstructing specular surfaces [42], intrinsic decomposition [1], and neural rendering [93]. These works typically consider planar reflections or require training on synthetic datasets, while our approach models reflections on complex geometry and is unsupervised. Prior works have also utilized additional light properties such as polarization [54, 21, 22] and time-of-flight [85, 41] for the separation of the reflected component and specular surface reconstruction. While the input of our approach is RGB images, there is scope for improving reconstruction quality by supplementing the algorithm with these additional cues.

B.2 COMPUTER VISION

Differentiable Rendering for 3D Computer Vision. Broadly speaking, a neural rendering framework is composed of a differentiable renderer, which can render the scene based on input parameters and is able to differentiate the scene w.r.t. those input parameters. While there are many formulations of differentiable renderers [77] [61] [56] [46] that can synthesize scenes, state-of-art approaches have shown tremendous success by relying on differentiable volumetric rendering [76]. Volumetric rendering approaches can realistically render complex scenes and are gradient-friendly. Thus, typical approaches train a neural network to encode the scene and optimize it for photometric consistency between input 2D images from different viewpoints [68] [94] [74] [75]. Recent

methods such as [35] [112] [96] [12] explicitly account for specularity, reflections, and other such phenomena, however, the goal of these works are to improve novel view synthesis. Thus, these methods still rely on learning the scene using photometric information.

B.2.1 Neural Radiance Fields (NeRFs)

Recent progress in neural radiance fields has enabled impressive novel view rendering and geometry reconstruction from multi-view images [69]. MipNeRF[7] demonstrates better novel view synthesis by modeling outgoing rays as cones to enable anti-aliasing. RefNeRF [113] proposes Integrated Directional Embeddings for improved novel view synthesis of reflections. NeRFReN [34] separates diffuse and specular radiance by using separate neural networks. Neural Catacaustics [50] propose a neural warping method to model reflections by learning the caustics of the surface. While these works focus on novel-view synthesis of the scene from the primary camera, we perform view synthesis that is beyond the line-of-sight of the primary camera, i.e., rendering views only visible to the objects present in the scene, while jointly estimating object geometry and separating diffuse and specular radiance.

B.2.2 Environment Estimation

Recovering underlying scene properties from multiple images is inherently ill-posed [84], but can be regularized using the natural statistics of scene properties as a prior [87, 6]. Recent works exploit this prior through deep neural networks and demonstrate inverse rendering of indoor scenes from a single image [29, 58, 116, 130]. However, these techniques typically recover only coarse representations of lighting and cannot reconstruct fine details of the environment. Lombardi *et al.* [59] recover environment and reflectance, assuming the scene is composed of known geometry and uniform material. Georgolis *et al.* [31] recover the environment map behind the camera from a single image of a glossy object, assuming the object is composed of textureless materials and using ground truth segmentation masks. Song *et al.*[95] estimate plausible environment maps by mapping reflections in the image and inpainting unmapped regions. Srinivasan *et al.* [97] capture stereo image pairs and estimate plausible spatially-coherent environment maps. NeRD [12], NeRFactor [128] and NeuralPIL [13] employ data-driven priors for lighting and BRDF in a NeRF-based approach for radiance decomposition. Park *et al.* [82] use RGB-D videos to

estimate environment map. Swedish *et al.* [101] recover high-frequency illumination map from the shadows of an object with known geometry. PhySG [125] and Munkberg *et al.* [71] perform inverse rendering from multi-view images by modeling the surface as signed distance functions. PANDORA [22] performs radiance decomposition from polarized RGB images.

B.3 CO-DESIGN OF IMAGING AND PERCEPTION

B.3.1 Joint Optimization of Optics & Algorithms

Our work is most closely related to the end-to-end optimization of cameras, which is an area of research focused on jointly optimizing components of cameras together with an algorithm, typically a neural network. Instead of relying on heuristics to generate visually pleasing images, the goal of end-to-end optimization is to produce images that optimize the pertinent information required for the task. Existing work primarily focuses on optimizing the parameters of the optical element, sensor, and image signal processor of a single camera. Applications of end-to-end optimization include extended depth of field and superresolution imaging [92], high dynamic range (HDR) imaging [67, 99], demosaicking [16], depth estimation [18, 37, 38], classification [17] and object detection [86, 80, 24]. For a more comprehensive review of end-to-end optimization, we refer readers to [48]. In contrast to end-to-end optimization methods, we focus on optimizing over the much larger space of possible imaging system designs, rather than the parameters of an individual camera. Our search space contains varying illumination sources, optics, sensors, and algorithms, each with many parameters. Rather than using stochastic gradient descent for optimization, we use reinforcement learning, allowing our approach to be used with non-differentiable simulators.

B.3.2 Reinforcement Learning

Deep reinforcement learning (RL) has become widely used in recent years as a way to do sequential decision-making for a wide array of problems, such as protein folding [44], learning faster matrix multiplication [28], and automated machine learning [3]. Many RL techniques focus on the *exploration-exploitation* trade-off, where an agent must learn to balance exploring new states with exploiting previously visited states that lead to high reward. RL is also used

for many combinatorial optimization problems [66]. In our work, we take inspiration from automated chip placement [70], which, like our approach, is formulated to allow an RL agent to place a new component at every step and select the placement of that component. Like many other problems RL has been applied to, imaging contains a high dimensional search space. In our work, we use proximal policy optimization (PPO) [91], which has been used for combinatorial search in past work [127].

Context-free grammars (CFGs) have been shown to be useful for designing machine learning (ML) pipelines, which are combinations of data-flows, ML operators, and optimizers [26][64][47]. Typically, ML pipeline design is done via a search over strings in the CFG using tree search algorithms, such as Monte Carlo tree search or upper confidence trees [49] [110]. We use CFG to functionally represent imaging systems as combinations of illumination, sensors, optics and algorithms such that the output string describes a camera configuration and perception model that can be used to solve a desired task.

C | PHYSICS AND LEARNED PRIORS

C.1 INTRODUCTION

What is a camera? Is it the hardware- the lens, optics, the sensor, or is it the software: image processing, low-light enhancement? What about a “camera" on a self-driving vehicle- is it the mounted cameras, the lidar, or the perception algorithm that detects pedestrians? Computation has enabled blurrier lines between the hardware and software, allowing for more seamless integration and interaction between the two. This has led to an incredible advancement in camera technology that can take incredible portrait and low-light photos from the phone, drive autonomous vehicles, or image the black hole.

Computational Imaging typically deals with solving tasks such as 3D shape reconstruction, phase estimation, and material estimation. These tasks rely on information beyond what the human eye can directly measure, so it no longer makes sense to design imaging systems based on the eye. Much like the evolution of animal vision, camera design has evolved to adapt to the needs of the task and environment [19]. By using known physics of light-matter interactions, physical cues such as polarization, interference, and spectrum are exploited to encode task-relevant information. Measurements of these cues can then be decoded into the scene parameter of interest by solving a model inversion problem. This idea of jointly exploiting physical cues and computation is the premise of the field of *computational imaging*.

Whereas imaging deals with capturing image representations of the world, *computer vision* extracts meaningful information from these images for high-level tasks, such as classification, detection, and segmentation. The modern era of computer vision was ushered in by advances in sensors, computing, and algorithms. High-resolution sensors paved the way to megapixel resolution, computing systems provided the bandwidth needed to process high-dimensional data, and deep learning provided a framework to learn from large amounts of data.

In Fig. 2, we plot inverse problems on the axis of how much physics they use vs. learned priors. Typical Deep Learning techniques are data-driven, relying on the model to implicitly learn the physics of the underlying scene.

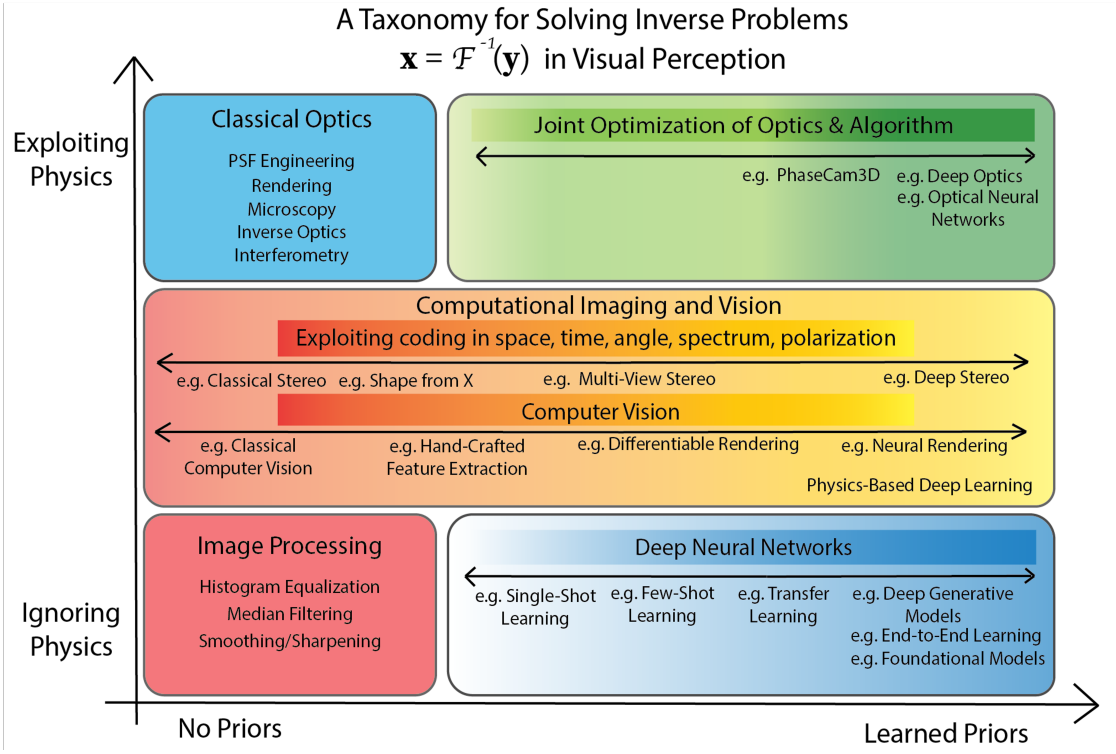

Figure 2: Our framework maps inverse problems in visual perception by how they parametrize $\mathcal{F}$. Deep learning focuses on learning priors through data-driven methods, whereas classical computer vision, optics, and computational imaging rely on physics. Each section of our paper [48] corresponds to a field/method shown in this chart. We anticipate future imaging systems will use physics and data for *joint optimization* (green box).

While these methods have had much success, recent progress has incorporated more physics through differentiable rendering [61] [56], or volumetric rendering [118] [68] [105, 104] based methods. This physics and machine learning framework has been highly effective and subsequent works have added additional physics-based priors such as reflectance models [96], [12], normal estimation [51], and shadow models [108] to enable better novel-view synthesis and 3D reconstruction. Moreover, these physics-based priors are now also used to train on classical computer vision tasks, such as object classification and segmentation, and show improved performance over purely data-driven techniques [120] [98]. These techniques are in the middle-right row moving upward (yellow to green) to exploit more physics, while many computational imaging techniques are in the middle-left row moving rightward (red-orange to yellow).

In the subsequent chapters we use this framework to solve inverse problems of 1) learning from shadows, and 2) converting objects into cameras using reflections and polarization. For 1) we convert the shape-from-shadows problems into a data-driven problem by learning a neural radiance field through a graphics-based shadow rendering model. In 2) we use concepts from Catadioptric Imaging Systems (CIS) that convert objects of known geometry and texture into wide field-of-view cameras. We show that by incorporating data-driven methods into CIS, we can learn a radiance field camera from an object with unknown geometry and texture. We also show how a polarization rendering loss can be added in this process for a better estimate of object geometry.

D | OBJECTS AS RADIANCE FIELD CAMERAS

D.1 INTRODUCTION

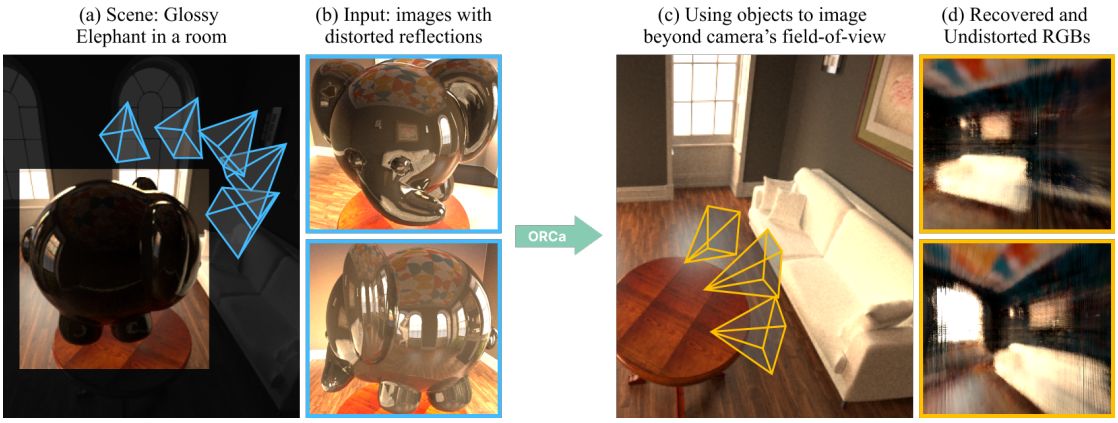

Figure 3: Objects as radiance-field cameras. We convert everyday objects with unknown geometry (a) into radiance-field cameras by modeling multi-view reflections (b) as projections of the 5D radiance field of the environment. We convert the object surface into a virtual sensor to capture this radiance field (c), which enables depth and radiance estimation of the surrounding environment. We can then query this radiance field to perform beyond field-of-view novel view synthesis of the environment (d).

Imagine that you're driving down a city street that is packed with lines of parked cars on both sides. Inspection of the cars' glass windshields, glossy paint, and plastic reveal sharp, but faint and distorted views of the surroundings that might be otherwise hidden from you. Humans can infer depth and semantic cues about the occluded areas in the environment by processing reflections visible on reflective objects, internally decomposing the object geometry and radiance from the specular radiance being reflected onto it- we use reflections on side-mirrors to drive, judge distances, and perform complex overtaking maneuvers. Our aim is to decompose the object from its reflections to "see" the world from the object's perspective, effectively turning the object into a camera that images its environment. However, reflections pose a long-standing challenge in computer

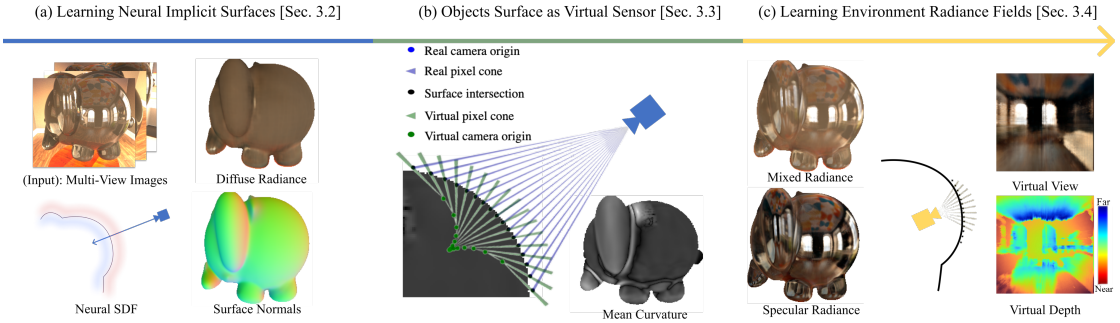

Figure 4: ORCa Overview. We jointly estimate the object's geometry and diffuse along with the environment radiance field estimation through a three-step approach. First, we model the object as a neural implicit surface (a). We model the reflections as probing the environment on virtual viewpoints (b) estimated analytically from surface properties. We model the environment as a radiance field queried on these viewpoints (c). Both neural implicit surface and environment radiance fields are trained jointly on multi-view images of the object using a photometric loss.

vision as the reflections are a 2D projection of an unknown 3D environment that is distorted based on the shape of the reflector.

To capture the 3D world from the object's perspective, we model the object's surface as a virtual sensor that captures the 2D projection of a 5D environment radiance field surrounding the object. This environment radiance field consists largely of areas only visible to the observer through the object's reflections. Our use of environment radiance fields not only enables depth and radiance estimation from the object to its surroundings but also enables beyond *field-of-view* novel-view synthesis, i.e. rendering of novel views that is only directly visible to the glossy object present in the scene but not the observer. Unlike conventional approaches that model the environment as a 2D map, our approach models it as a 5D field without assuming the scene is infinitely far away. Moreover, by sampling the 5D radiance field, instead of a 2D map, we can capture depth and images around occluders, such as close-by objects in the scene, as shown in Fig. 5. These applications cannot be done from a 2D environment map.

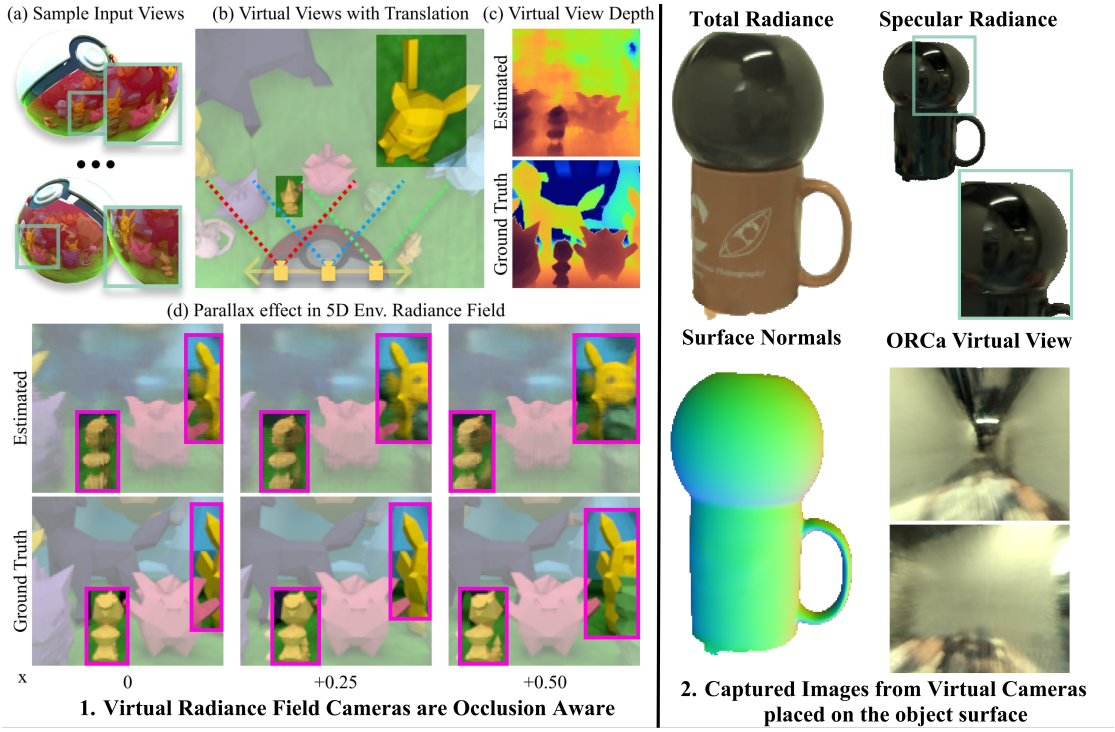

Figure 5: Advantages of Virtual Radiance Field Cameras. In SubFigure 1 we show how Modeling reflections on object surfaces (a) as a 5D env. radiance field enables beyond *field-of-view* novel-view synthesis, including the rendering of the environment from translated virtual camera views (b). Depth (c) and environment radiance of translated and parallax views can further enable imaging behind occluders, for example revealing the tails behind the primary Pokemon occluders (d). In SubFigure 2, we show how the Virtual Radiance Field camera can be queried at novel positions to render novel views of the hallway. The resolution of the rendering image is related to the relationship between the size of the object and the baseline between the real-world camera positions. We refer our readers to [106] for more information.

We aim to decompose reflections on the object's surface, from its surface and exploit those reflections to construct a radiance field surrounding the object, therefore capturing the 3D world in the process. This is a challenging task because the reflections are extremely sensitive to local object geometry, viewing direction and inter-reflections due to the object's surface. To capture this radiance field, we convert glossy objects with unknown geometry and texture into radiance-field cameras. Specifically, we exploit neural rendering to estimate the local surface of the object viewed from each pixel of the real camera. We then convert this local surface into a virtual pixel that captures

radiance from the environment. This virtual pixel captures the environment radiance as shown in Fig 7. We estimate the outgoing frustum from the virtual pixel as a cone that samples the scene. By sampling the scene from many virtual pixels on the object surface, we construct an environment radiance field that can be queried independently of the object surface, enabling beyond *field-of-view* novel-view synthesis from previously unsampled viewpoints.

Our approach jointly estimates object geometry, diffuse radiance, and the environment radiance field from multi-view images of glossy objects with unknown geometry and diffuse texture in three steps. First, we use neural signed distance functions (SDF) and an MLP to model the glossy object's geometry as a neural implicit surface and diffuse radiance, respectively, similar to PANDORA [22]. Then, for every pixel on the observer's camera, we estimate the virtual pixels on the object's surface based on the estimated local geometry from the neural SDF. We analytically compute the parameters of the virtual cone through the virtual pixel. Lastly, we use the cone formulation in MipNeRF [7] to cast virtual cones from the virtual camera to recover the environment radiance.

D.1.1 Contributions

To summarize, we make the following contributions:

- We present a method to convert implicit surfaces into virtual sensors that can image their surroundings using virtual cones. (Sec. D.2.3)
- We jointly estimate object geometry, diffuse radiance, and estimate the 5D environment radiance field surrounding the object. (Fig. 7)
- We show that the environment radiance field can be queried to perform *beyond-field-of-view* novel viewpoint synthesis, i.e render views only visible to the object in the scene (Section D.2.4)

Scope. We only model glossy objects with low roughness as such specular reflections tend to have a high signal-to-noise ratio, therefore, are a sharper estimate of the environment radiance field. However, we note that the virtual cone computation can be extended to model the cone radius as a function of surface roughness. Deblurring approaches can further improve the resolution of estimated environment. In addition, we approximate the local curvature using mean curvature, which fails for objects with a varying radius of curvature along the tangent space. We explain how our virtual cone curvature estimation can

be extended to handle general shape operators in the supplementary material. Lastly, similar to other multi-view approaches, our approach relies on a sufficient virtual baseline between virtual viewpoints to recover the environment's radiance field.

D.2 METHOD

D.2.1 Overview

Reflections on glossy objects offer a glimpse into the surrounding environment beyond the camera's field of view. From multi-view images of a glossy object with unknown geometry and albedo, we aim to recover the 5D radiance field of the surrounding environment. The mapping from images captured by the observer to the surrounding environment depends on the glossy object's surface properties, in particular, the surface normals and curvature. We first cast a cone from the observer camera's center-of-projection through each pixel viewing the scene. When the cone intersects the object's surface, it reflects, causing the cone to be transformed (Fig. 7.a). The transformed cone, referred to as a virtual cone, samples the environment and is primarily responsible for the specular radiance observed on the glossy object. Our key insight is that the reflections captured by the observer's camera can be modeled as a projection of the environment radiance field onto the object's surface. By modeling the reflected rays as a cone and computing the parameters of the cone, we can more accurately estimate the projected environment radiance field onto the object surface, as shown in Fig. 7.b.

ORCa is composed of three steps: modeling the object's geometry as a neural implicit surface (Sec. D.2.2), converting the object's surface into a virtual sensor (Sec. D.2.3), and modeling the environment radiance field as a projection along these virtual cones (Sec. D.2.4). The learned environment radiance field can then be queried on novel viewpoints to show occluded areas in the scene. Fig. 4 depicts our output for each component on a scene rendered with a complex glossy object and 3D environment. We now describe each step in detail.

D.2.2 Learning Neural implicit Surfaces

Neural Signed Distance Function We model the object geometry as a neural signed distance function (SDF). $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. SDFs provide a helpful inductive

bias for learning smooth surface geometry [122, 115, 79] that assists downstream tasks in our pipeline. Moreover, the surface properties crucial for our framework, surface normals and curvature, can be conveniently computed from SDFs in a differentiable manner. Consider the 3D spatial coordinates, $\mathbf{x}$, in the scene. The glossy object surface, $\mathcal{S}$ is then represented by the zero-level set of the SDF

$$\mathcal{S} = \{f_{\mathcal{S}}(\mathbf{x}) = 0 | \mathbf{x} \in \mathbb{R}^3\} \tag{2}$$

Similar to Yariv *et al.* [122], we model the SDF $f_{\mathcal{S}}$ as a coordinate-based MLP.
Surface Normals Gradients of the SDF at the zero level set point $\mathcal{S}$ towards the surface normals $\mathcal{S}$,

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} f_{\mathcal{S}}(\mathbf{x})}{\|\nabla_{\mathbf{x}} f_{\mathcal{S}}(\mathbf{x})\|} \quad \mathbf{x} \in \mathcal{S} \tag{3}$$

Surface Curvature We employ differential geometry techniques developed by Novello *et al.* [78] to estimate curvature for neural implicit surfaces. In particular, we estimate the mean curvature $K(\mathbf{x})$ for the implicit surface from the divergence, $\boldsymbol{\nabla}$ of the surface normals

$$K(\mathbf{x}) = \frac{\boldsymbol{\nabla} \cdot \mathbf{n}(\mathbf{x})}{2} \tag{4}$$

Mean curvature approximates the surface with an osculating sphere. Our approach also works for more generalized notions of curvature through the shape operator, at the cost of higher computational complexity. We refer our readers to the supplement for the general case.
Diffuse Radiance We separate the captured radiance at the observer camera with diffuse radiance, which depends on the glossy object's albedo, and specular radiance which depends on the environment radiance. The diffuse radiance does not have any view dependence and only depends on surface point $\mathbf{x}$. We denote the diffuse radiance as f_d and model it using a coordinate-based MLP.
Volume Rendering As proposed in [122], we perform volumetric rendering on the SDF. We define the volume density $\sigma(\mathbf{x})$ as the cumulative distribution function (CDF), denoted as $\Psi(s)$, applied to $f_{\mathcal{S}}$:

$$\sigma(\mathbf{x}) = \alpha \Psi_{\beta}(f_{\mathcal{S}}) \tag{5}$$

In contrast to [122], however, we only aim to recover the diffuse radiance of the object along a particular ray. We define a function f_d that estimates the

diffuse radiance at each point, $\mathbf{x}$, along the ray. To get the final diffuse radiance along a given primary ray, $\mathbf{r}_p(t)$, we perform volumetric rendering:

$$\hat{\mathbf{c}}_d(\mathbf{r}) = \int_0^\infty f_d(\mathbf{r}(t), f_S^k(\mathbf{r}(t))\tau(t)dt \tag{6}$$

Note that there is no view dependence in Eq. 6 and intermediate features, f_S^k, are used as input. $\tau(t)$ is the accumulated transmittance along the ray.

D.2.3 Objects Surface as Virtual Sensor

Each pixel, $\mathbf{p}$, with a finite surface area, $\mathbf{dA}_p$, on the real-camera sensor views the surface of the object through a frustum originating at that pixel. The object then samples the environment radiance field through this finite surface converting the finite surface into a virtual pixel with surface area, $\mathbf{dS}$. Through this model, we can interpret the object surface as a virtual sensor consisting of many virtual pixels that sample radiance from the environment field based on the geometry of the object and observer viewing direction. We now formulate a virtual pixel based on real camera pose and implicit surface geometry. Please refer to Fig. 7 for a visualization of the virtual sensor.

Consider a real camera origin as $\mathbf{o}$ and a pixel on the real sensor $\mathbf{p}_{i,j}$ that corresponds to ray direction $\mathbf{d}$. The primary ray for pixel $\mathbf{p}_{i,j}$ is parameterized with ray length t as $\mathbf{r_p}(t) = \mathbf{o} + t\mathbf{d}$

Casting Real Cones We can approximate the outgoing conical frustum from pixel $\mathbf{p}_{i,j}$ as a cone originating at $\mathbf{o}$ with axis-of-direction $\mathbf{d}$ and radius $\dot{r}$, equivalent to half the distance of the pixel in the x and y directions. We represent the real cone as a parametric volume,

$$\mathbf{r}_{cone}(\dot{r}, s, \theta) = \dot{r}s\cos(\theta)\hat{\mathbf{e}}_u + \dot{r}s\sin(\theta)\hat{\mathbf{e}}_v + \dot{r}s\mathbf{d}, \tag{7}$$

where $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{e}}_v$ are basis vectors in the plane perpendicular to $\mathbf{d}$, $\theta \in [0, \pi]$ and $s \in [0, t_{max}]$

Virtual Pixel. Virtual pixels are characterized by the intersection of the real cone with the object surface. In Sec. D.2.2, we model local surface properties using mean curvature which enables efficient analytical computations for the virtual pixel parameters even though our approach works with general shape operators. For a sampled point t_i along the ray, we have the surface normals $\mathbf{n}(t_i)$ from Eq. 3 and estimated mean curvature $K(t_i)$ from Eq. 4. The local object surface at t_i,

can be approximated with an osculating sphere, $\mathcal{O}(t_i)$, centered at $\hat{\mathbf{o}}_\mathbf{S}(t_i)$ with radius, $R(t_i)$ as

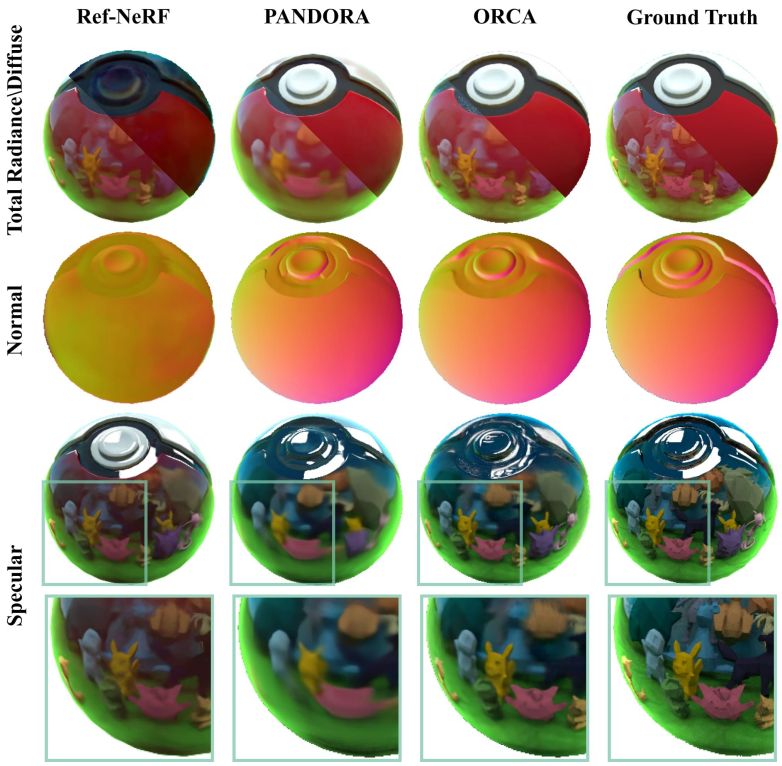

Figure 6: Qualitative comparisons of diffuse-specular separation and geometry estimation on rendered dataset. The environment contains nearby objects with complex occlusions when seen through reflections on the glossy object. RefNeRF fails to perform accurate diffuse-specular separation and PANDORA blurs the nearby objects in the specular map. ORCA can model the complex specular reflections through environment radiance field.

$$R(t_i) = \frac{2}{K_{t_i}} \hat{\mathbf{o}}_\mathbf{S}(t_i) = \mathbf{r}_p(t_i) + R(t_i) \cdot \hat{\mathbf{n}}(t_i)$$

For concave surfaces $K_{t_i} < 0$. So, $\hat{\mathbf{o}}_\mathbf{S}$ will lie outside the object. For $K_{t_i} > 0$, $\hat{\mathbf{o}}_\mathbf{S}$ will lie inside the object.

The edges of the virtual pixel for $\mathbf{r_p}(t_i)$ would lie at the intersection of the osculating sphere $\mathcal{O}(t_i)$ and the primary cone given by $\mathbf{r}_{cone}$. Computing exact cone-sphere intersections are computationally expensive so we approximate the cone-sphere intersection using rays bound cone-sphere intersectional surface

dS. We consider four rays that bound the cone and sample them at $\theta_j \in \{0, \pi/2, \pi, 3\pi/2\}$ with Eq. 7. We perform intersections of the corresponding bounding rays with the osculating sphere $\mathcal{O}(t_i)$ to get corners of the virtual pixel $\mathbf{ds}_j$. These ray sphere intersections can be computed analytically in an efficient manner.

Virtual Cone Origin. From the virtual pixel surface area, we can now compute the virtual cone that samples the environment. We first compute normal vectors at virtual pixel corners $\mathbf{ds_j}$ from the center of osculating sphere $\hat{\mathbf{o_S}}$

$$\hat{\mathbf{n_j}} = \frac{\mathbf{ds_j} - \hat{\mathbf{o_S}}}{||\mathbf{ds_j} - \hat{\mathbf{o_S}}||} \tag{8}$$

At each virtual pixel corner, we compute the reflected ray directions, ω_j^r, by computing the dot product between the incoming ray directions, ω_k^i, and the normals, $\hat{\mathbf{n}}_k$, where ω_0^r is the primary ray's reflected vector.

$$\omega_0^r = \mathbf{d} - \big(\mathbf{d} \cdot \hat{\mathbf{n}}(t_i)\big)\hat{\mathbf{n}}(t_i) \tag{9}$$

$$\omega_j^r = \mathbf{d}_j - \big(\mathbf{d}_j \cdot \hat{\mathbf{n}}_j(k)\big)\hat{\mathbf{n}}_j(k) \tag{10}$$

$\mathbf{d}_j$ are the incident directions to the virtual pixel corners $\mathbf{ds}_j$. The virtual cone origin is the intersection of these reflected rays at the pixel corners and pixel center. However, these rays might not intersect at a single point so we approximate a virtual origin to be the point that minimizes the sum of distances to the reflected rays ω_j.

$$\mathbf{v_o} = \text{argmin}_{\mathbf{v}} \sum_j |(\mathbf{v} - \mathbf{ds_j}) \times \omega_j^r| \tag{11}$$

We pose this as a linear least squares problem and estimate the virtual cone origin efficiently through pseudo-inverse.

Virtual Cones Direction. The reflected ray at the center of the virtual pixel reflects the object surface along the direction ω_0^r from Eq. 9. We consider this as the direction-of-axis of the virtual cone.

$$\hat{\mathbf{v}}_d = \omega_0^r \tag{12}$$

Virtual Cone Radius. We compute the radius of the cone by treating the reflection vectors of the bounding rays as the neighboring "pixel" directions.

Similar to [7], we can compute the distance between $\{\omega^r_{k_\theta}\}^{2\pi}_{\theta=0}$ and the primary reflected ray ω^r_0 in the (x, y) components (omitted below for clarity).

$$\hat{\mathbf{v}}_{\dot{r}} = \|\{\omega^r_{k_\theta}\}^{2\pi}_{\theta=0} - \omega^r_0\| \tag{13}$$

Finally, for each sampled point t_i, we can characterize our single-pixel virtual sensor located at the object surface $\mathbf{dS}$ as a virtual cone with $\hat{\mathbf{v}}_o$ as its apex, $\hat{\mathbf{v}}_d$ as axis-direction, $\hat{\mathbf{v}}_{\dot{r}}$ as the radius.

Connections to caustics. Our work takes inspiration from catadioptric imaging systems. To convert objects into cameras, we compute the surface and find a corresponding center-of-projection for this surface-as-sensor. However, unlike conventional perspective cameras, objects don't have a fixed center-of-projection, other than in a few special configurations [5], but a locus of viewpoints that vary with object geometry and viewing direction. These viewpoints lie on the "caustic surface" of the object. While typical works in catadioptric imaging analytically compute the caustic surface by assuming known geometry [32, 100], or making assumptions about placement of the observer [102], our formulation approximates the caustic surface of unknown geometry through the intersection of reflected rays on virtual pixels. We empirically show in the supplement that as the surface area of the virtual pixel goes to 0, $d\mathbf{S} \to 0$, our method estimates the true caustic of the object without assuming geometry. Our method also has applications in estimating the caustic surface of the unknown geometry.

D.2.4 Environment Radiance Fields

Our goal is to capture a 5D environment radiance field of the scene by imaging the world through these single-pixel virtual sensors located at the object's surface. We use our formulation of virtual cones to recover 5D environment radiance fields. We define an environment radiance field as $f_{\mathcal{E}} : (\hat{\mathbf{v}}_o, \hat{\mathbf{v}}_d) \to (\sigma^{Env}, c_s)$,

where $f_{\mathcal{E}}$ outputs opacity and radiance along sampled virtual cones. We note that this view-dependent radiance is equivalent to the specular radiance at

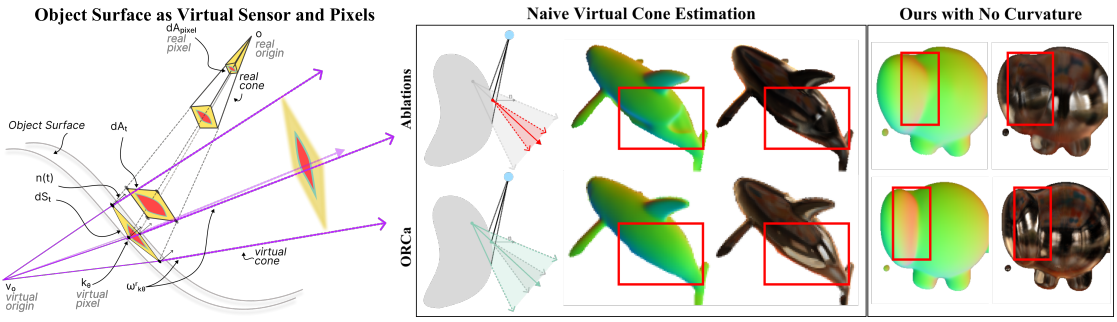

Figure 7: Virtual Sensors and Ablation on Virtual Cone Comparison. *Column 1:* We image the world through the object by modeling each pixel's specular radiance as a projection of the 5D radiance field of the environment onto the object's surface. We capture the radiance field by treating the surface area on the object that the pixel views, $\mathbf{dS}_t$, as a single-pixel virtual camera with its center-of-projection at $\mathbf{v}_o$. We cast virtual cones through the virtual sensor to capture the 5D radiance field of the environment. Columns 2 and 3 show that accurate estimation of the virtual cones (Sec. D.2.3) is crucial to model environment radiance fields. If the virtual cone origin is assumed to be at the object surface (left column) or if the surface is assumed to locally have no curvature (right column), the surface normal and specular radiance outputs suffer from artifacts (red boxes).

point t_i sampled along the primary-camera ray $\mathbf{r_p}(t)$. We can render the final specular radiance at pixel $\mathbf{p}_{i,j}$ as follows:

$$\hat{\mathbf{c}}_s(\mathbf{r}) = \int_0^\infty f_\mathcal{E}(\hat{\mathbf{v}}_o, \hat{\mathbf{v}}_d)\tau(t)dt$$
$$\hat{\mathbf{c}} = \hat{\mathbf{c_d}} + \hat{\mathbf{c_s}}$$

Intuitively, $f_\mathcal{E}$ learns the 5D radiance field by sampling single-pixel virtual sensors from the object surface and must learn geometry and environment radiance that is consistent with the multi-view reflections. Moreover, we can query $f_\mathcal{E}$ to render novel viewpoints and associated depths that are beyond the field-of-view of the real camera. We volume render each virtual cone by dividing them into conical frustums using Integrated-Positional Encoding as proposed in MipNeRF [7]. Our formulation of virtual cones works well with Mip-Nerf's rays-as-cones method.

Approach	Diffuse Radiance PSNR ↑ (dB)	Diffuse Radiance SSIM ↑	Specular Radiance PSNR ↑ (dB)	Specular Radiance SSIM ↑	Mixed Radiance PSNR ↑ (dB)	Mixed Radiance SSIM ↑	Normals MAE ↓ (
Ref-NeRF	18.80	0.7304	16.99	0.6633	20.19	0.7890	43.690
PANDORA	18.25	0.7260	16.25	0.6483	18.90	0.7284	7.606
ORCA	**19.84**	**0.7893**	**20.74**	**0.7535**	**22.00**	**0.7947**	**2.339**

Table 1: Average evaluation metrics on rendered scenes. We compare ORCa to other neural rendering techniques that model reflections, including Ref-NeRF and PANDORA, on six simulated scenes. ORCa provides consistent improvements in geometry estimation, diffuse-specular separation and novel view synthesis. Please refer to the supplement for additional metrics.

Scene: Ball-cup in hallway		
Approach	Mixed Radiance PSNR ↑ (dB)	Mixed Radiance SSIM ↑
Ref-NeRF	32.75	0.9617
PANDORA	28.83	0.9758
ORCA	30.86	0.9799

Scene: Owl in hallway		
Approach	Mixed Radiance PSNR ↑ (dB)	Mixed Radiance SSIM ↑
Ref-NeRF	26.65	0.8890
PANDORA	27.24	0.9343
ORCA	26.84	0.9299

Table 2: Quantitative metrics for real-world, captured scenes. ORCa demonstrates comparable performance in novel view synthesis on scenes from the PANDORA real dataset.

D.3 EXPERIMENT AND RESULTS

D.3.1 Implementation Details

As in PANDORA, we parameterize f_S with an 8-layer MLP to estimate the surface, and, as in MipNeRF, f_d with 4-layer MLP with input geometric features of size 512 from f_S. We follow the SDF-to-opacity conversion and the iterative sampling of the ray proposed in [122]. To aid the network to learn geometry quickly, we also train f_S with a mask-net as proposed in [22]. We use five losses in our architecture: photometric loss, mask loss [22], normal loss [113], eikonal loss [122], and distortion loss [8]. Additional training details are discussed in the supplement.

D.3.2 Datasets

We conduct experiments on both simulated and real-world datasets. Simulated datasets are rendered in Mitsuba2 [77]. Simulated datasets contain a range of increasingly complex object geometries (elephant, Pokeball, and orca) and scenes (living room and Pokemon). We train with 200 views for simulated datasets. We also show results for a real-world dataset [22] capturing a glossy cup with a black vase sitting atop it using 35 views.

D.3.3 Advantages of Environment Radiance Fields

Other neural rendering techniques that handle reflections, Ref-NeRF and PANDORA, estimate the environment as a 2D map, while ORCa recovers a 5D environment radiance field. Fig. 5 shows the advantages of recovering a 5D environment radiance field. Close-by surrounding objects often cause occlusions that cannot be modeled by 2D environment maps. By estimating the radiance field, we can image behind occluders through sampling novel viewpoints such as the translated viewpoints shown in Fig. 5. Moreover, we can also show the depth of the surroundings from these virtual viewpoints. We also quantitatively evaluate the estimated depth of the surroundings for the synthesized virtual views in Fig. 5 and show that ORCa provides reliable depth estimates, especially for nearby objects. We provide additional examples of depth estimation and beyond *field-of-view* novel-view synthesis in the supplement.

D.3.4 Impact of Virtual Cone Computation

We base our method on a physically accurate formulation by modeling ray-cone intersections and using the surface as a virtual sensor, as described in Sec. D.2.3. We demonstrate the importance of this step by setting up two ablation experiments. In the first experiment, which we term Naive Virtual Cone, we place the origin of the cone at the object surface instead of computing the virtual cone origin based on the curvature. In the second experiment, which we term No Curvature, we assume that locally the surface is like a flat mirror and has no curvature. In Fig. 7, we show that for both of these ablation experiments, we see worse performance as demonstrated by the artifacts in estimated surface normals and specular components (shown as red boxes).

D.4 CONCLUSION

We present a method to convert glossy objects with unknown geometry and texture into radiance-field cameras that capture the environment's radiance field around them. Our method recovers object geometry and diffuse radiance, in addition to capturing the depth and radiance of the object's surroundings from its perspective. Our modeling of the environment as a radiance field is effective in recovering close-by objects (Fig. 7) and is occlusion aware (Fig. 5). From the recovered environment radiance field, we can perform beyond *field-of-view* novel-view synthesis. Our work can unleash applications in virtual object insertion and 3D perception, e.g. inferring information beyond the line-of-sight of the camera using predicted virtual views and depth. Our formulation goes beyond the conventional direct-line-of-sight radiance fields and can enable further areas of research to extract more information from multi-view images directly from the environment and the objects present in it.

D.5 ADDITIONAL DETAILS

D.5.1 General Shape Operator

General Implicit Curvature Estimation. To approximate the virtual pixel lying on the object-cone intersection surface, we find intersection points along rays that bound the cone and approximate the surface by finding intersection points with the surface and the rays. Ideally, we would query the sdf MLP for points along the bounding rays to get the intersection points with the surface, however, due to computing requirements we approximate the surface using the second-order derivative of the local geometry. We approximate this surface in Sec. 3.2 using mean curvature sampled around a point, t_i, on ray $r_p(t)$. However, this choice was solely based on compute and efficiency constraints, and other approximations such as gaussian or principal curvature can also be used. Since our surfaces are neural implicit surfaces, we use techniques in differential geometry for neural implicit functions as proposed in [78] to estimate curvature. For a general case, we can define a shape operator, $\mathbf{d}N$, on the tangent plane at

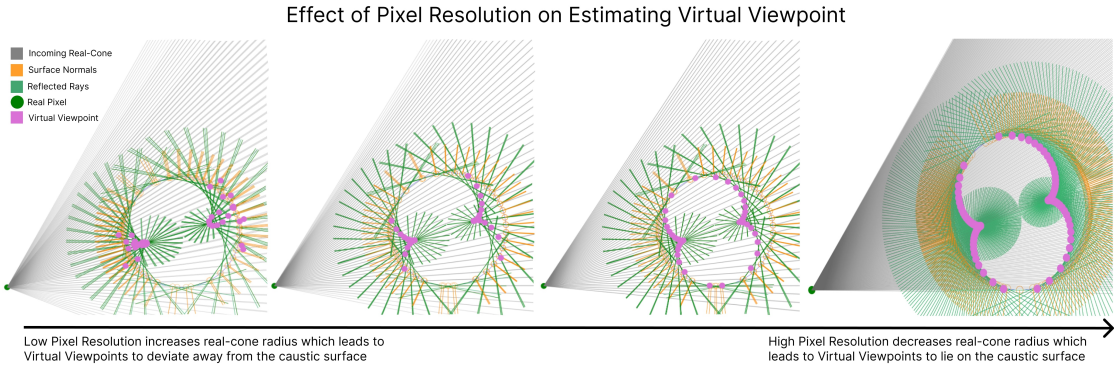

Figure 8: Effect of Pixel Resolution On Virtual Viewpoints. We cast a real cone (grey) from each pixel (dark green) with decreasing radii (indicating a higher resolution) in different directions. The cone, parametrized by 3 rays intersects the circle and we compute the surface normals (yellow) and reflected rays (light green). We find the closest intersection point between the reflected rays by solving least-squares and denote that as the virtual viewpoint (magenta). As we decrease the real cone radii, the virtual pixel surface area, $\mathbf{ds}_j$ also decreases and the reflected rays are closer together pointing in similar directions. As $\mathbf{ds}_j \to 0$ the virtual viewpoint starts to form a catacaustic of a circle-which denotes the true loci of virtual viewpoints of the object-as-camera.

point t_i. The shape operator, $\mathbf{d}N$, can be expressed as follows, where $\mathbf{H}$ is the Hessian operator:

$$\mathbf{d}N = \left(I - \hat{\mathbf{n}}(t) \cdot \hat{\mathbf{n}}(t)^T \right) \frac{\mathbf{H} f_{\mathcal{S}}}{||\nabla f_{\mathcal{S}}||} \tag{14}$$

From the shape operator, we can find the curvature along any vector $\mathbf{v}$:

$$\kappa_{\mathbf{v}} = \left\langle -\mathbf{d}N \cdot \mathbf{v}, \mathbf{v} \right\rangle \tag{15}$$

, where $\langle ., . \rangle$ is the inner product. Using Eq. 15 we can compute principal, mean or gaussian curvatures to estimate the differential surface at t_i. By using gaussian curvature, for instance, we can approximate our surface to be locally quadric such as handling surfaces that are hyperbolic. Our ray-sphere intersection will now be able to change to ray-ellipse, ray-hyperbolic, ray-parabolic, or ray-planar intersection depending on the sign of the curvature.

Note that for concave surfaces $K_{t_i} < 0$, so $\hat{\mathbf{o}_{\mathbf{S}}}$ will lie outside the object and, for convex, $K_{t_i} > 0$, $\hat{\mathbf{o}_{\mathbf{S}}}$ will lie inside the object. This is a useful property as

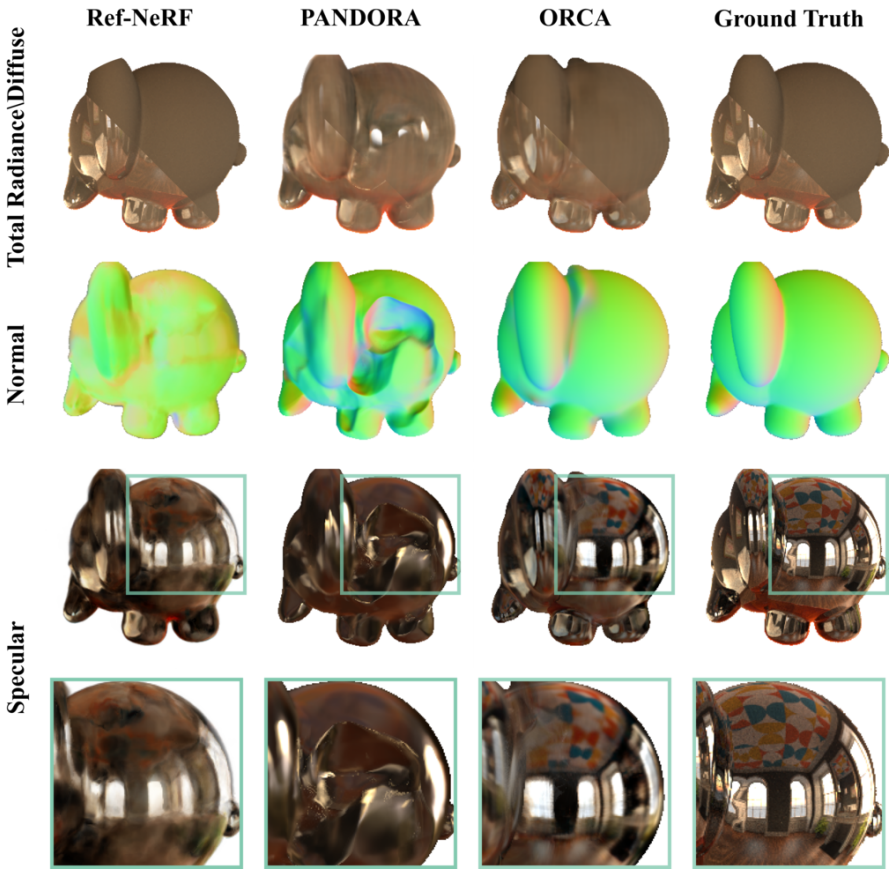

Figure 9: **Comparisons on Elephant-in-the-Room dataset**. We compare a sample test viewpoint against existing techniques that only capture an environment map. We show that our method outputs smoother surface normals, and diffuse and specular separation, in addition to the recovery of finer details such as the textured ceiling and the high-frequency illumination on the elephant through the windows.

the virtual-cone apex changes based on the curvature and our formulation is generalizable to locally concave and convex surfaces.

D.5.2 Relation to Caustics

To convert the object into a camera, we model the object's surface as a sensor. As discussed, the center-of-projection of the object-as-camera changes wrt geometry and viewing direction, however, as shown by [32] [100], it must lie on the caustic surface of the object. One way to estimate the virtual viewpoints or the apex

of the virtual cone is using the known caustic surface of the object, however, our formulation assumes unknown geometry therefore the surface is unknown. To account for this approximate the virtual viewpoint with the closest point to the reflected rays. We visualize this method (Sec. 3.3) in flatland using ray-circle intersection in Figure 8. We shoot real cones from a single pixel at different angles, approximated by 2 bounding rays and 1 primary ray, and intersect the real-cone with the object. We compute surface normals (yellow) and compute the associated reflected rays (green) and the virtual viewpoint (magenta) using our closest-point to reflected-rays method in Sec. 3.3. We show that by increasing the pixel resolution, the real cone radius decreases projecting a smaller virtual-pixel surface area on the object, $\mathbf{ds}_j$. We can calculate the virtual viewpoint for this pixel and empirically show that as $\mathbf{ds}_j \to 0$, the virtual viewpoints along the surface tend to form the catacaustic of the object. We can also use our method to approximate the caustic of unknown geometry and has applications in Catadioptric Imaging Systems (CIS). Moreover, we also note that our method is limited by the resolution of the camera viewing the object- for lower resolution or objects further away, the virtual viewpoint will not be accurate.

D.5.3 Roughness

We also capture the environment radiance field on a globe with high roughness and show the specular and diffuse radiance, depth from the object's surface to its surroundings in addition to a virtual view. We note that even for rougher objects our framework can recover an environment radiance field. However, the recovered radiance field is blurry due to the roughness acting like a low pass filter that removes the high-frequency components such as the cushion on the sofa or blurs the textures on the ceiling. The recovered radiance field, associated virtual views, and the depth from the object surface are therefore blurrier and coarse respectively. For example with high roughness, while we are able to recover coarse depth, the depth-from-object-surface is smoother at the ceiling with the globe with low roughness. In future work, we can also expand our cone formulation to include a roughness parameter, similar to RefNeRF [113], that can change the radius and apex of the virtual cone to account for rougher objects.

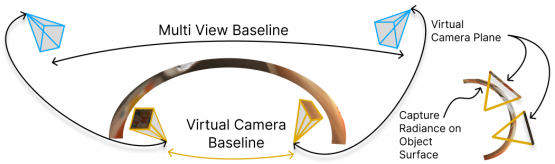

Figure 10: **Glossy object's size acts as virtual baseline** On the left, we show that the baseline for the virtual views is fundamentally limited by the object size. On the right, we show that our environment radiance field must learn to map radiance accumulated on the object-surface-as-sensor to the new virtual camera image plane with a new virtual center-of-projection to perform novel view synthesis. The distortion is high for objects with varying geometry or a low radius of curvature, but we show in our paper that our formulation of virtual cones can handle this undistortion well even for complex geometries.

D.5.4 Object size as virtual baseline

As Figure 10 shows, the virtual baseline for convex objects will lie inside the object's surface or near the object's surface for concave objects. This means that the virtual baselines are much smaller and limited by object geometry- as the object size decreases, the virtual cones' apex will be close to each other, and the multi-view virtual baseline will tend to 0, effectively acting as a monocular setup. This also means that associated radiance field and depth maps will also be more coarse for small objects.

E LEARNING FROM SHADOWS

E.1 INTRODUCTION

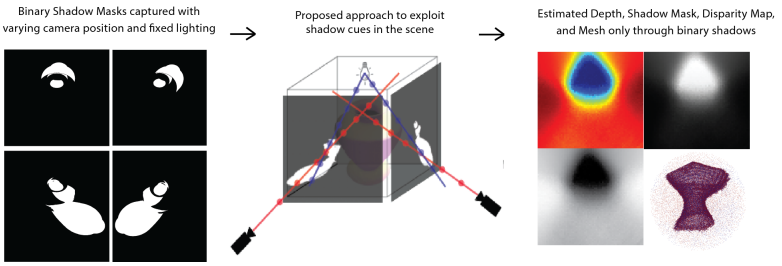

Figure 11: Exploiting physical cues in neural rendering. Our approach takes sparse binary shadow masks captured with varying camera positions under fixed lighting and uses our proposed differentiable shadow rendering model to estimate shadow maps, thereby learning neural scene representations. We can visualize the learned implicit representations by rendering estimated depth maps and estimated shadow maps from novel views. We also run marching cubes [62] on our learned representations to get explicit meshes for a quantitative analysis.

Recovering 3D geometry from 2D images remains an extremely important, yet unsolved problem in computer vision and inverse graphics. Considerable progress has been made in the field when assumptions are made, such as bounded scenes, diffuse surfaces, and specific materials. However, reconstruction algorithms still remain largely susceptible to real world effects, such as specularity, shadows, and occlusions [114]. This susceptibility is largely due the variation in different materials and textures, and a non-unique mapping from 3D geometries to 2D images. Even though these effects cause issues for many methods, they also provide valuable information about the scene and geometry of the object. For example, cues like self-shadows provide vital information about an object's concavities, while shadows cast on the ground plane provide information about its geometry. Moreover, shadows are independent of textures and surface reflectance models and are a strong cue in overhead imagery where vertical surfaces, like facades, are sampled poorly, whereas oblique

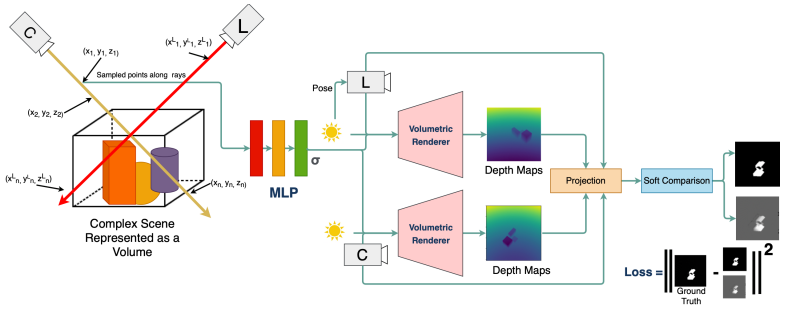

Figure 12: Overview of the proposed pipeline We train a neural network to predict opacity at points along the camera and light rays. The opacities are used by the volumetric renderer to output the ray-termination distance which we use to estimate the *z-buffer* from the camera and the light perspective, the latter also known as the shadow map. The estimated *z-buffer* is fed into a **Projection** step that projects the camera pixels and their associated depths into the light's reference frame. The shadow map is indexed to obtain the corresponding depth values at these new points. The projected depths and indexed depths go through a **Soft Comparison** step which outputs predicted cast shadows in the scene from the camera's perspective. A loss is computed on the *predicted* and the *ground-truth* shadow mask.

lighting can expose this geometry. Exploiting, instead of ignoring these cues, can make algorithms robust and the fundamental problem of 3D reconstruction less ill-posed.

Previous works in recovering 3D shape of objects by exploiting physical cues has relied on constructing inverse models to explicitly handle and exploit cues such as shadows, shading, motion, or polarization [11] [129] [126]. These approaches are physically anchored as they use properties of light or surface reflectance models to exploit cues and only need up to a single image to reconstruct simple objects. Albeit successful under strict assumptions about lighting, camera, and the object, these models typically cannot handle complex scenes and do not translate well into real-world scenarios as creating inverse models to capture complex physical phenomena soon becomes intractable and hard to optimize.

To combat the problem of real world variability, modern methods such as [109] [68] [81] [94] [123] [60] have largely been data-driven by directly learning 3D representations on real-world scenes based on photometric consistency. Such methods employ an *analysis-by-synthesis* approach to solve the problem by using machine learning to search the space of possible 3D geometries and an inverse model to synthesize the scene based on the predicted geometries.

These approaches typically only optimize the photometric loss between different camera viewpoints and show success in learning implicit representation by rendering novel views. However, because they do not explicitly handle these physical cues in their forward model, they fail in scenarios with complex lighting [96], specularity [124], or reflections [35].

Motivated by the above observations, we explore what can be learned by exploiting physical cues in a data-driven neural rendering framework. In this paper, we investigate whether the neural rendering framework can learn geometry from physical cues without the assumptions made by the aforementioned methods. We study the use of shadows cast by objects onto themselves and nearby surfaces as the only source of information for 3D reconstruction. While modern approaches for 3D reconstruction ignore such cues, we aim to exploit them. Our unsupervised approach uses *only* shadows to reconstruct the scene by leveraging recent advances in volumetric rendering and machine learning, and therefore proposes a physically anchored data-driven framework to the problem of shape from shadows. Moreover, unlike previous work in shape from shadows, we present a novel method that uses differentiable rendering in the loop to iteratively reconstruct the object based on a loss function instead of iteratively refining the object through explicit carving. Specifically, we use an efficient shadow rendering technique called shadow mapping as the forward model and make it differentiable so that it can be used as an inverse model to iteratively reconstruct the object. Our work also reveals that from limited cues the differentiable volumetric rendering component can *quickly converge to localize and reconstruct a coarse estimate of the object when such cues are explicitly modeled by a forward model*. Our work also suggests that neural rendering can exploit shadows to recover hidden geometry, which otherwise may not be discovered by photometric cues.

E.1.1 Contributions

The paper makes the following contributions:

- A framework that directly exploits physical cues like shadows in neural renderers to recover scene geometry.
- A novel technique that integrates volumetric rendering with a graphics-inspired forward model to render shadows in an end-to-end differentiable manner.

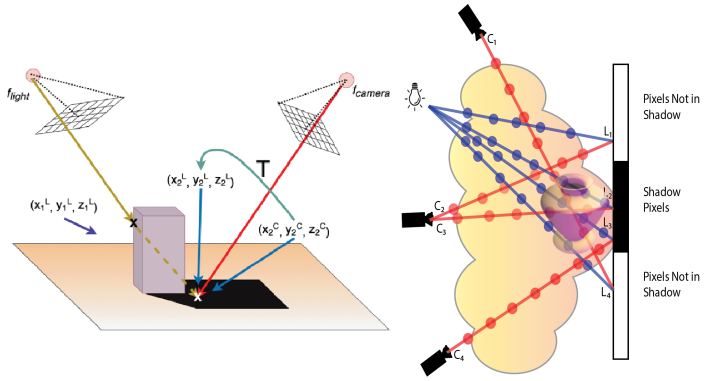

Figure 13: Figure **(a)**: A point $\mathbf{x} \in \mathbb{R}^3$ in the scene is defined to be in shadow if no direct path exists from the point $\mathbf{x}$ to the light source, implying that there **must** be an occluding surface between $\mathbf{x}$ and the light source. We differentiably render the scene's depth from the camera and the light's perspective at each pixel and then project the camera pixel and its depth into the light's frame of reference. We then index the light's depth map, or z-buffer, to get z_1^L. We note that z_1^L is less than z_2^L, i.e. there must be an occluding surface as a ray projected from the light's perspective terminates early. This implies that this point is in shadow. Figure **(b)** shows a 2D slice of our approach and represents a volume (cloud) with the shadow mask unraveled. The network learns an opacity per point (dots) via the shadow mapping objective which penalizes predicted geometries that don't cast perfect ground truth shadows. Through this, the network learns 3D geometry that is consistent across all shadows maps for all cameras given a particular light source.

- Results showing that our framework can learn coarse scene representations from just shadows masks. To the best of our knowledge, we are the first to show that it is possible to learn neural scene representations directly from binary shadow masks.

E.2 NEURAL REPRESENTATIONS FROM SHADOWS

Our goal is to recover the scene through shadows cast on the other objects or onto itself. Our method recovers shadows in an image by applying a threshold on that image thereby making no distinction between types of shadows. We show how we model the shape-from-shadows problem using differentiable

rendering and implicit representations in Section E.2.1 and our graphics-inspired differentiable forward model in Section E.2.2. In Section E.2.3, we discuss our additional techniques that we use to enable optimization on binary shadow masks.

E.2.1 Scenes as Neural Shadow Fields

Implicit Scene Representations. Similar to Mildenhall *et al.* [68], we represent a continuous scene by parametrizing it using a learnable function f_θ. However, our approach does not include any photometric component, therefore we represent the scene as a 3D function with input $\mathbf{x} = (x, y, z)$ and a volumetric density σ as output.

$$\gamma(\mathbf{x}) = \Big(\sin(2^0\pi\mathbf{x}), \cos(2^1\pi\mathbf{x}), ..., \sin(2^L - 1\pi\mathbf{x}), \cos(2^L - 1\pi\mathbf{x}) \Big)$$
$$f_\theta : \mathbb{R}^L \rightarrow \mathbb{R}^+; (\gamma(\mathbf{x})) \mapsto (\sigma) \tag{16}$$

We use a positional-encoded 3D point $\gamma(\mathbf{x})$, $\{\gamma(\mathbf{x}) \in \mathbb{R}^L, \mathbf{x} \in \mathbb{R}^3\}$ as input, which maps to an associated volumetric density $\sigma \in \mathbb{R}^+$ [68] [103]. In contrast, f does not encode view dependant color and is independent to viewing direction.

Volumetric Renderer. We define a volumetric renderer $\mathbf{R}_{\text{vol}}$ that takes N opacities $\{\sigma\}_{i=1}^N$ at N discretely sampled points $\{\mathbf{x}\}_{i=1}^N$ along a ray $\mathbf{r}$.

$$\mathbf{R}_{\text{vol}} : \left[\mathbb{R}^+\right]_{i=1}^N \rightarrow \left[\mathbb{R}^+\right]_{i=1}^N; (\{\sigma\}_{i=1}^N) \mapsto (\mathbf{d}) \tag{17}$$

Since we only have binary shadows as input, we modify the renderer to output the ray termination distance, $\mathbf{d}$, instead of the radiance at that ray. $\mathbf{R}_{\text{vol}}$ is not a trainable component, but the ray termination distance, $\mathbf{d}$, is differentiable w.r.t. the input opacities. The estimated ray-termination distance, range, is computed as follows:

$$\hat{\mathbf{D}}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i t_i; T_i = \prod_{j=1}^{i-1} \left(1 - \alpha_j\right); \alpha_i = \left(1 - e^{-\sigma_i \delta_i}\right) \tag{18}$$

We sample $\mathbf{r}(t)$ at points $\{t_0, ..., t_N\}$ and evaluate the function $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ to get sampled points $\{x_0, ..., x_N\}$ in the scene. T_i is defined as the cumulative transmittance from t_0 until t_i and $\delta_i = t_{i+1} - t_i$ which is the distance between two samples. σ_i is the estimated opacity at point i by a learned function f_θ.

Intuitively, the renderer gives us the ray termination distance for each ray shooting through a pixel.

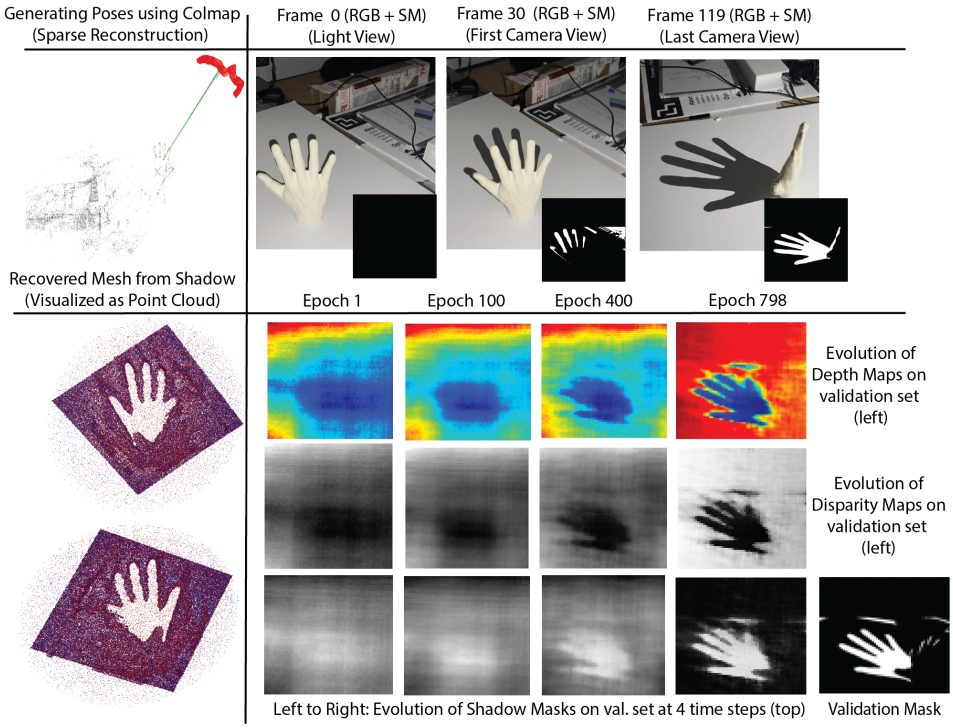

Figure 14: Real-World Experimentation: We use the *exact same pipeline and training scheme to reconstruct a 3D mesh from real-world data*. We take a video on the iPhone to generate poses for light and camera using COLMAP [43](video) and extract shadows using an intensity threshold. We show that our method can reconstruct a finer mesh of the hand from the real-world images. We highlight that our method can more easily generalize from sim2real in comparison to photometric approaches since we learn from only shadow masks, which are invariant to many real-world effects, such as texture.

E.2.2 Differentiable Shadow Mapping

We define any point $\mathbf{x} \in \mathbb{R}^3$ in the scene to be in shadow if no direct path exists from point $\mathbf{x}$ to the light source $\mathbf{L}$. This logic implies that there **must** be some object or an occluding surface between the point $\mathbf{x}$ and $\mathbf{L}$ that occludes the light ray from reaching point $\mathbf{x}$. In graphics, shadow mapping [119] uses this observation to construct a forward model to render efficient and accurate shadows in the scene based on known light and camera sources. Our approach

makes this efficient shadow rendering forward model differentiable so that it can be used as an inverse model. We then pose the problem of shape from shadows and use our proposed inverse model to estimate the 3D geometry of the scene.

Estimated z-buffer. We first evaluate the renderer from the camera's perspective to get the estimated ray termination distance, or range map, $\hat{\mathbf{D}}_{cam}$ for all rays coming out of the binary shadow map. However, shadow mapping requires the depth perpendicular to the image plane, i.e along the z axis of the camera's local coordinate system. This depth is equivalent to a *z-buffer* in graphics and we refer to this value as the *depth* at that pixel. We define a function g to estimate the *z-buffer* $\hat{\mathbf{Z}}$ from the range map $\hat{\mathbf{D}}$.

$$\hat{\mathbf{z}}_{u,v} = g(\mathbf{d}_{u,v}) = \frac{\mathbf{d}_{u,v}}{||(u, v, 1) \cdot \mathcal{E}||_2} \tag{19}$$

The function takes a ray shooting from a pixel (u, v) and a predicted range, $\hat{\mathbf{D}}_{cam}^{u,v}$ as input. $\mathcal{E}$ is the rotational component of the camera's extrinsic matrix, $\mathbf{d}_{u,v}$ is the ray termination distance from camera's focal point, and $\hat{\mathbf{z}}_{u,v}$ is the depth along the z-axis from the pixel (u, v). We also compute the estimated z-buffer from the light's perspective, which we refer to as the estimated *shadow map*.

Projection. With the estimated depths at each pixel from the camera and the light source, we now need to estimate which camera pixels are in shadow given the particular light source. As illustrated in Figure 13, we do this by projecting all pixels and their associated depths visible by the camera into the light's frame of reference. We then use this projected coordinate to index the shadow map to get the depth to that point from the light's perspective. We formally write this as follows:

$$\begin{aligned} (U^l_{cam}, V^l_{cam}, \hat{\mathbf{Z}}^l{}_{cam}) &= (U_{cam}, V_{cam}, \hat{\mathbf{Z}}_{cam}) \cdot P_{light\_from\_cam} \\ \hat{\mathbf{Z}}^{U^l_c, V^l_c}_{light} &= \hat{\mathbf{Z}}_{light}\left[U^l_{cam}, V^l_{cam}\right] \end{aligned} \tag{20}$$

Here, $\hat{\mathbf{Z}}_{cam} \in \mathbb{R}^{H \times W}$ is the estimated z-buffer from the camera's perspective at pixels $\{U_{cam}, V_{cam}\} \in \mathbb{R}^{H \times W}$. $P_{light\_from\_cam}$ is the projection matrix to the light's reference frame from the camera's. We denote $(U^l_{cam}, V^l_{cam}, \hat{\mathbf{Z}}^l{}_{cam})$ as the pixels and depth in camera's frame (subscript) projected into the light's frame, denoted by the superscript l. We index the shadow map, $\hat{\mathbf{Z}}_{light} \in \mathbb{R}^{H \times W}$, at the projected camera pixels to retrieve the depth of the projected camera pixels from

the light source. This is denoted as $\hat{\mathbf{Z}}_{light}^{U_c^l, V_c^l}$ which is the shadow map indexed at pixel locations U_c^l, V_c^l. In practice, not all pixels will project within the shadow map's height and width constraints specified at the start of training. In graphics, these pixels are usually ignored, however, we clamp all our projections to lie within the height and width bounds to maintain differentiability.

Soft Comparison. Once we have the depths to the projected camera pixels and the depths from the light source to those pixels in the same reference frame, we can then compare them to discover if the camera pixel is in shadow. As illustrated by Figure 13, if the depth from the light source to a point is less than the depth from the camera projected into the light's frame, it means that the light ray must have intersected an object before reaching that point. Thus, that point must be in shadow. Based on this logic, we formulate a soft comparison, which compares different depths to output the predicted binary shadow mask as follows:

$$\begin{aligned} \Delta \hat{Z}_{light} &= \left(\hat{\mathbf{Z}}^l{}_{cam} - \hat{\mathbf{Z}}_{light}^{U_c^l, V_c^l} \right) \\ \hat{\mathbf{M}}_{binary} &= \mathbf{max}\left(\frac{\Delta \hat{Z}_{light}}{\beta}, \epsilon \right) \end{aligned} \tag{21}$$

We denote $\hat{\mathbf{M}} \in \mathbb{R}^{H \times W}$ as the output of the entire pipeline: predicted shadow masks. The input to our soft comparison is the projected camera z-buffer into the light's frame, $\hat{\mathbf{Z}}^l{}_{cam}$, and the shadow map indexed at the projected points $\hat{\mathbf{Z}}_{light}^{U_c^l, V_c^l}$ from the **Projection** step. β is a scaling hyper-parameter used to enlarge or decrease the difference, and ϵ is a threshold. We also formulate a "smoother" version of the predicted shadows:

$$\hat{\mathbf{M}}_{smooth} = \mathbf{S}\Big(\mathbf{normalize}\big(\Delta \hat{Z}_{light}, \mu_{min}, \mu_{max} \big) \Big) \tag{22}$$

Here, μ_{min}, μ_{max} are used to control the normalization function and $\mathbf{S}$ is the sigmoid function.

E.2.3 Optimization

To enable convergence, we smooth the binary ground truth shadow masks $\mathbf{M}$ to better guide the framework in predicting accurate shadow masks.

Distance Transform. Binary images contain limited information for differentiation as the gradient is zero everywhere except for the edges where it is one. To

Scene	RMSE Shadow Mesh	RMSE Vanilla NeRF
Cuboid	**0.0078**	0.097
Vase	**0.010**	0.0.011
Bunny	0.0109	**0.0106**
Chair	**0.0092**	0.0096

Table 3: We quantitatively analyze the quality of the reconstructed meshes by running ICP [9] on meshes generated by our proposed method, which only uses binary shadows masks, and meshes generated by a vanilla NeRF trained on full RGB images. We show RGB images from Vanilla NeRF in the supplementary along with training details.

encourage our model to estimate better shadow masks, thereby learning a better 3D model, we use a distance transform on the ground truth shadow masks. Specifically, we scale pixel intensities of a binary shadow mask by their distance to the nearest shadow edge. We modify the weighted distance transform in [88] for our approach. The transformed binary shadow mask, $w(\mathbf{M}, \sigma) = \mathbf{M}_w$ is computed as follows:

$$w(\mathbf{M}, \sigma) = \mathbf{M} + \left(w_c(\mathbf{M}) + w_0 \cdot \exp\left(- \frac{(d_1(\mathbf{M}) + d_2(\mathbf{M}))^2}{2\sigma^2} \right) \right) \tag{23}$$

Here, $\mathbf{M}$ is the ground truth binary shadow mask computed after applying a fixed threshold on binary images. w_c is weight map to balance class frequencies, w_0 and σ are hyper parameters. d_1 and d_2 are distances to the nearest and second nearest cell, respectively. We note from our experiments that this particular distance transform yields the most consistent convergence compared to other distance transforms, such as blurring.

Shadow Mapping Loss. We optimize our entire framework on binary shadow masks and train the MLP on the following loss:

$$\mathcal{L}_{sm} = ||w(\mathbf{M}, \sigma) - \hat{\mathbf{M}}||^2 \tag{24}$$

Here, $w(\mathbf{M}, \sigma)$ is the σ weighted ground truth shadow mask, and $\hat{\mathbf{M}}$ is the predicted shadow mask from Equation (22).

E.3 RESULTS

E.3.1 Simulated 3D Reconstruction Results.

We show the learned scene representations qualitatively by converting them to explicit meshes and rendering them using a signed distance function (SDF). Figure 15 shows the estimated meshes from our method on four object types. We compare our meshes to meshes generated by running vanilla NeRF on RGB images, and the ground truth by running marching cubes on the volume. Our datasets are rendered with overhead camera viewpoints, which enables shadows to be exploited. Given the binary and sparse nature of shadow masks in terms of their information content, we observe that our forward model coupled with the differentiable rendering framework converges to good coarse estimates of object geometry. Moreover, in the case of vases, the mesh reconstruction benefits from exploiting shadows as the algorithm can use *hidden* cues present in the scene, such as the curvature of the vase, which are only partially visible when relying on photometric cues.

E.3.2 Real-World Reconstruction Results.

We show our method's ability to converge to a fine mesh on real-world data of a hand in Fig. 14. Information on data acquisition is provided in the supplementary materials. We first note that our method is robust to coarse light poses as there are visible shadows from the estimated light's pose in Fig. 14 (please refer to the main paper [107] for details). Our method is able to converge to a fine mesh of the hand, including the fingers and the space between them. We use only 74 shadow masks which makes our method versatile to environments with limited camera views and rarer objects, and no object priors. Moreover, we also show the convergence of the estimated shadow masks, disparity and depth maps from a novel viewpoint. The final estimated shadow mask shown in Fig. 14 is similar to the validation shadow mask and also contains some shadow artifacts due to the threshold segmentation. Additionally, the sim2real gap is not present for our pipeline as it only uses object shadows.
Lastly, we briefly discuss how the data-driven components find the easiest solution that is consistent with our physics of the defined forward model, not the actual world. We note that our training data only has views of cast shadows and does not contain any self-shadows which are present on the back of the hand. This causes the algorithm to instead estimate the mesh of the table and

create a hollow imprint of the hand such that the specified shadow constraint is met. A stand-alone mesh of the hand can be recovered from this imprint and the recovered mesh is a possible solution given the shadow masks and proposed model. Imposing object priors or adding an extra view of self-shadow to the training set could result in a stand-alone mesh.

E.3.3 Quantitative Analysis

We also run our datasets on a vanilla NeRF [68] implementation [83]. At lower resolutions and overhead viewpoints, we see that the NeRF approach fails to provide a reasonable fine mesh. We believe this failure is due to the down-sampling of images to 64×64, which may also be a reason why our meshes fail to capture fine details. We run ICP [9] on the generated points cloud and show on-par results to the NeRF approach. Our goal, however, is not to outperform NeRF but to show the effectiveness of a differentiable rendering framework in exploiting physical cues instead of ignoring them. The main takeaway from Table 3 is that differentiable volumetric renderers do not need to rely on 8-bit RGB information to reconstruct accurate meshes, but can also leverage other sources of information in the image in addition to relying on photometric cues.

E.4 DISCUSSION

One of the major goals of our work is to propose a framework within neural rendering that can readily exploit and learn from, instead of ignore, sparse physical cues, such as shadows. We believe that Fig. 14 shows that sparse physical cues like shadows, actually encode a lot of *hidden* information about the scene and can indeed be exploited. By constructing explicit differentiable forward models and leveraging gradient-friendly volumetric rendering, we can exploit these cues in conjunction with relying on photometric consistency between images.

E.4.1 Limitations

In cases such as the cuboid and the vase, we observe that the renderer converges to a predicted mesh that minimizes the shadow masks and the predicted shape even though it is typically a coarse estimate that envelopes the entirety of the object. This means that we see artifacts such as the pointed curve in the vase

mesh, or the curvature of the bunny. Since our algorithm only has geometry information where the binary shadow mask is true, areas that are never in shadow have no surface, which leads to incomplete meshes. Imposing a prior can be a solution to this problem. Moreover, our method also assumes a known lighting position, which may not always be available.

E.4.2 Future Work

We observe that volumetric rendering can converge onto coarse estimates of the object geometry by only relying on shadows, and can be extended to problems such as non-line-of-sight imaging (NLOS) [111] and imaging behind occluders [39]. As shadows themselves are never the only cue present to reconstruct the scene, our work can also be easily integrated with existing NeRF approaches that rely only on photometric cues as our shadow loss 22 can be used as a regularizer or an auxiliary loss, especially as shadows are invariant to viewpoint changes, surface reflectance properties, or texture changes.

E.4.3 Conclusions

We show that modern neural rendering techniques can learn neural scene representations (neural shadow fields) and encode 3D geometry just from binary shadow masks. We are motivated by traditional shape-from-X algorithms that typically construct physics-driven inverse models that can exploit cues for 3D reconstruction. We observe that data-driven neural rendering frameworks ignore cues such as shadows, relying on photometric cues instead. We thus propose a graphics-inspired differentiable shadow rendering component that leverages a volumetric renderer to encode a scene solely from its shadows.

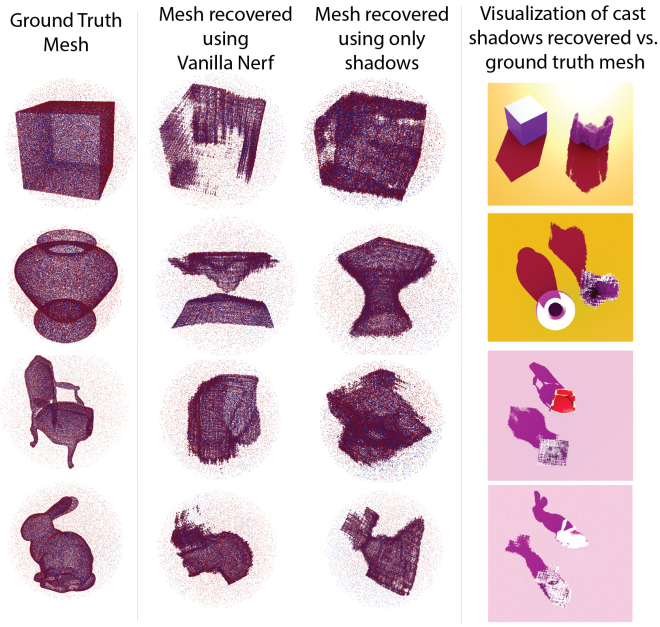

Figure 15: Qualitative Results. We observe that for overhead views of the scene where the vertical surface of the vase is sampled poorly in the RGB space, vanilla NeRF fails to exploit geometry cues hidden in cast shadows compared to our approach. Our method doesn't impose any object priors therefore it infers a geometry that will minimize the difference between the predicted and true shadow. Column 4 illustrates that rendered shadows are very similar, indicating that the differentiable rendering framework can indeed learn geometry from sparse shadow cues. Some parts of the objects such as the upper face of cuboid are never in shadow, therefore our approach yields no reconstruction for those surfaces, further showing that the geometry is indeed *only* learnt from cast shadows. We extract the mesh from the volume using marching cubes and visualize it here using a point-cloud SDF representation.

F | TOWARDS DISCOVERY OF VISUAL CUES

F.1 INTRODUCTION

Suppose you're an Imaging designer. Consider the task of 3D reconstruction of a vase in a room that is hidden behind an occluder with the shadows of the vase visible in the camera. When designing a camera, one must discover that among all the RGB information present, shadows provide the most valuable information about the hidden object. Now let's consider a situation where the shadows aren't visible but the designer is now able to illuminate the scene. Then one could potentially learn the relationship between shining light onto nearby walls and using cast shadows as a result- which serves as a more robust method to perform 3D reconstruction of the vase. This technique was proposed by [40].

Such non-trivial tasks are found throughout various applications in biology, autonomous driving, and remote sensing. Current methods in Imaging require a human designer and expertise, as the designer must know 1) which cues to consider and exploit in the scene, 2) which hardware configuration to use: optics, illumination, and sensing, and 3) which algorithm to use to decode the scene. However, as imaging becomes ubiquitous and sensing becomes possible in other modalities in time, and other wavelengths of the electromagnetic spectrum, relying on a human designer to discover which cues and imaging system to design is not sufficient.

In this chapter, we ask the question: how do we build systems that can automatically discover which camera configuration and cues to use to solve the task? In short: given a task and some constraints, how do we automatically discover what camera configurations, cues, and algorithms can solve the task? For example, in the example above, the system must realize 1) the shadow pixels provide the most information about the hidden vase, 2) a relationship between pixels and illumination, and 3) mapping from shadow pixels to 3D reconstruction of the vase. In essence, such a system must figure out which cues to exploit to solve the task and build a corresponding hardware and software setup that can complete the task.

In this section, we present a method that automatically searches over the space of camera designs and perception models. More specifically, we first introduce a context-free grammar, called Computational Imaging Grammar, with which we plan to co-design imaging and perception models. Second, we show that this space can be searched over using reinforcement learning using an agent, Camera Designer (CD), that outputs hardware camera configurations for the desired task, and a Perception Model (PM), that outputs the task objective. We are inspired by how animal eyes and brains are tightly integrated [52], our approach jointly trains the CD and PM, using the performance of the PM to inform how the CD is updated during training. We apply our method to depth estimation using stereo cues, demonstrating the viability of jointly learning imaging and perception. We construct an environment that is devoid of monocular cues to force (1) the CD to learn to obtain multi-view cues and (2) the PM to learn to exploit these cues. Our results show that the agent and perception model co-learn to exploit non-monocular cues to estimate accurate depth. Finally, we make the following contributions:

- **Imaging CFG**: We introduce a context-free grammar (CFG) for imaging system design, which enumerates possible combinations of illumination, optics, sensors, and algorithms. The CFG can be used as a search space and theoretical framework for imaging system design.
- **Co-Design**: We demonstrate how task-specific camera configurations can be co-designed with the perception model by transforming the CFG into a state-action space and using reinforcement learning. Our approach can converge despite the reward function being jointly trained with the policy and value functions.
- **Experimental Validation**: We demonstrate our method for co-design by applying it to the task of depth estimation using stereo cues.

While our approach is validated on depth estimation using known visual cues, it can also be applied to other tasks within visual computing in the future. Discovery is enabled by our use of RL to search over the large space of imaging components, allowing new combinations of hardware and software created by the learned RL policy. For more information, we urge the reader to our paper: DiSER: Designing Imaging Systems with Reinforcement Learning.

$$\mathrm{R} \rightarrow \mathrm{X}\mathcal{S}\mathrm{X}\mathrm{A} \tag{25}$$
$$\mathrm{X} \rightarrow \mathcal{I}\mathrm{X}|\mathrm{O}\mathcal{S}\mathrm{X}|\mathrm{A}_2\mathrm{X}|\epsilon \tag{26}$$
$$\mathrm{O} \rightarrow \mathcal{O}\mathrm{O}|\epsilon \tag{27}$$
$$\mathrm{A}_1 \rightarrow \mathcal{A}_1\mathrm{A}_1|\mathcal{A}_1 \tag{28}$$
$$\mathrm{A}_2 \rightarrow \mathcal{A}_2\mathrm{O}\mathcal{S}|\mathcal{A}_2\mathcal{I}|\mathcal{A}_2\mathcal{S}|\epsilon \tag{29}$$
$$\mathcal{S} := \left\{s_p s_{hw} s_t s_\lambda s_q\right\}_{p\in\mathbb{R}^6, h,w,t,q\in\mathbb{Z}} \tag{30}$$
$$\mathcal{O} := \left\{o_f o_d\right\}_{f\in\mathbb{R}, D\in\mathbb{Z}} \tag{31}$$
$$\mathcal{I} := \left\{i_p i_i\right\}_{p\in\mathbb{R}^6, i\in\mathbb{Z}} \tag{32}$$
$$\mathcal{A}_1 := \{a_{nn}, a_{fourier}, ...\} \tag{33}$$
$$\mathcal{A}_2 := \{\text{autofocus}, ...\} \tag{34}$$

Figure 16: Context-free grammar (CFG) for imaging: Production rules (1-5) and alphabets (6-10) for our proposed CFG for designing imaging systems. R is the starting symbol from which a design starts. All imaging systems must have at least one sensor, $\mathcal{S}$, and one algorithm, $\mathcal{A}$. The grammar allows arbitrary physically plausible combinations of illumination ($\mathcal{I}$) optics ($\mathcal{O}$), sensors ($\mathcal{S}$), and algorithms ($\mathcal{A}$), each defined in their respective alphabet above. A_1 refers to algorithms that process the output of hardware, while A_2 refers to algorithms that control hardware.

F.2 METHOD

F.2.1 Computational Imaging Grammar

We define the configuration space of imaging systems using context-free grammar (CFG) as it allows for a flexible configuration space that can be searched. A typical context-free grammar, G, is represented as a tuple, $G = (V, \Sigma, P, R)$, where V corresponds to non-terminal symbols in the grammar, Σ corresponds to terminal symbols, P corresponds to the production rules, and R is the start symbol. The goal of our proposed CFG is to allow the construction of strings to represent arbitrarily complex imaging systems, which usually consist of illumination sources, optical elements, sensors to convert light into digital signals, and algorithms that decode the scene. For example, consider the task of depth estimation that can be done in numerous ways. One solution is depth from stereo, which involves placing two cameras, c_1, c_2, in the scene at points, p_1, p_2, with some baseline. Each camera has an optical element, $o_1 = (f, d)$, with a focal length, f, and aperture, d, and a sensor, $s_1 = ((h, w), t)$, with spatial and temporal resolutions, (h, w) and t, respectively. Thus the cameras can

be expressed as $c_1 = (o_1, s_1)$ and $c_2 = (o_2, s_2)$. An algorithm can decode the outputs of the two cameras to produce depth, and can be implemented with correspondence-matching [14], (a_{st}), or deep stereo [63], (a_{ds}). The full system can be described as a string, $s_1 =$ "$c_1 c_2 a_{st}$" or $s_2 =$ "$c_1 c_2 a_{ds}$". Another way to estimate depth is with active illumination or time-of-flight (ToF) imaging. We can represent lidar as an algorithm, a_{control}, that illuminates the scene at the same point with a laser, l_1, and ToF sensor, s_{ToF}. We can describe this system as $s_{\text{lidar}} = a_{\text{control}} l_1 s_{ToF} a_{ToF}$. These examples illustrate how CFG can represent imaging systems with different illumination, optics, sensors, and algorithms as strings. The goal of the proposed CFG is not to describe how the individual components of an imaging system are made, e.g. their electronics, but rather to describe the function of each component. Next, we define the grammar's alphabet and production rules.

Grammar. Our proposed CFG can be stated as $G = (V, \Sigma, P, R)$. We define the variables as $V = \{\text{X}, \text{O}, \text{A}_1, \text{A}_2\}$, each defined in the following sections, and the terminals, Σ, which we refer to as alphabets, as $\Sigma = \{\mathcal{I}, \mathcal{O}, \mathcal{S}, \mathcal{A}_1, \mathcal{A}_2\}$, where $\{\mathcal{I}\}$ is illumination, $\{\mathcal{O}\}$ is optics, $\{\mathcal{S}\}$ is sensors, and $\{\mathcal{A}_1\}$ and $\{\mathcal{A}_2\}$ are algorithms. Each alphabet contains possible components and parameters, defined in lower case, e.g. a_{nn}. Each component within an alphabet is parameterized by its functionality, e.g. focal length, rather than an off-the-shelf component. We describe each alphabet below and in Fig. 16.

Illumination. The illumination alphabet, $\mathcal{I}$, functionally represents different types of possible illuminations. In imaging, illumination can be represented with many parameters, such as duration (d), intensity (i), color, wavelength (λ), polarization η, pose (position & orientation), (p) and modulation in space and time [10]. In the scope of this work, we consider pose and intensity. These can later be extended to other forms of illumination.

Optics. We define the optics alphabet, $\mathcal{O}$, to capture the most important (but not exhaustive) optical properties in an imaging system: focal length (f) and aperture (D). The optics alphabet can be extended to include more complex techniques such as phase masks or diffractive optical elements (DOE). The non-terminal O indicates that optical elements can be stacked to create a multi-lens system.

Sensors. The sensor alphabet, $\{\mathcal{S}\}$, functionally describes different types of sensors, such as RGB and SPAD. We parameterize a sensor by its pose s_p, spatial (or angular) resolution s_{hw}, temporal resolution s_t, bit quantization s_q and wavelength s_λ. For example, a SPAD sensor has higher temporal resolution (picosecond scale) and generally lower spatial resolution (on the order of 1,000 to 100,000 pixels), while a typical RGB sensor (CMOS) has a higher spatial

resolution (hundreds of megapixels), but a lower temporal resolution (30 fps). Similarly, quantization (for example) can be varied between 1, 8 or 12 bits. The pose is the position (x, y, z) and the orientation (pitch, yaw, and roll) of the sensor in 3D space, $s_p \in \mathbb{R}^6$.

Algorithms. Algorithms are needed to decode raw images and control other alphabets. We denote the alphabet for algorithms with two sets: $\{\mathcal{A}_1, \mathcal{A}_2\}$. $\mathcal{A}_2$ is the set of algorithms that affect subsequent illumination, optics, and sensors (e.g. autofocus, controlling where to shine illumination), whereas $\mathcal{A}_1$ are algorithms that decode the incoming data from the sensors for a given task. These algorithms include standard imaging operators, such as the Fourier transform, back-projection, Radon transform, Gerchberg-Saxton algorithm, photometric stereo, and more. Additionally, $\mathcal{A}_1$ includes neural networks, which can perform detection, classification, etc. Due to the production rule, $\mathrm{A} \rightarrow \mathcal{A}_1\mathrm{A}|\mathcal{A}_1$, $\mathcal{A}_1$ can be repeated and stacked together. For example, an algorithm can be designed that takes the Fourier transform of the input data and feeds it through a multilayer perceptron (MLP).

Production Rules. We define a set of production rules, shown in Fig. 16, that can produce strings representing possible imaging system configurations. In our formulation, every imaging system includes at least one sensor and algorithm. The X accounts for imaging systems with different illumination, optics and sensors. In all cases, the string must end with at least one algorithm that outputs the desired task. Additionally, each $\mathcal{A}_2$ also requires an illumination, optics component, or sensor that it controls. The production rules account for multiple sensors and illuminations that illuminate and sense different parts of the scene.

F.2.2 Imaging Design with Reinforcement Learning

The proposed context-free grammar (CFG) defines ways of combining illumination, optics, sensors, and algorithms to form an imaging system. The goal of our work is to automate imaging system design by searching over the CFG. Because the output of the cameras in the imaging system must be well suited for a specific, downstream task, we co-design them with the task-specific perception model (PM). We next propose using a learned camera designer (CD) to automatically search over the CFG. We implement the CD with reinforcement learning (RL) because (1) the combination of continuous variables in our CFG causes an explosion in the search space, which, as a result, makes search with methods such as Monte Carlo tree search (MCTS) [15] or alpha-beta search [90] intractable, and (2) many advanced imaging simulators are not differentiable

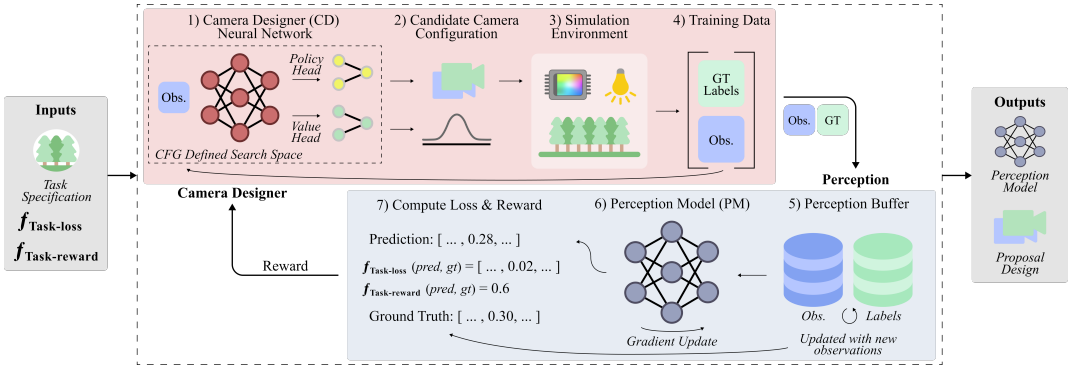

Figure 17: Approach: Our approach allows camera configuration and a perception model (PM) to be co-designed for task-specific imaging applications. At every step of the optimization, the camera designer (CD), implemented with reinforcement learning, proposes candidate camera configurations (1-2), which are used to capture observations and labels in a simulated environment (3-4). The observations and labels are added to the perception buffer (5) and used to compute the loss and reward, while the N most recent observations in the perception buffer are used to train the PM. The reward is propagated to the CD agent which proposes additional changes to the candidate camera configuration. After the episode terminates, the CD agent is trained using proximal policy optimization (PPO) [91] until convergence.

[30, 27, 36], and thus gradient descent cannot be directly applied. Our problem is well suited for sequential decision making because the task performance achieved with each choice of camera configurations directly affects subsequent design choices.

Overview: Our approach is illustrated in Fig. 17. The input is a task-specific loss and reward function. When optimization starts, the imaging system contains no hardware. At each step, the CD selects whether to add a component into the system and the component's parameters (Fig. 17a-b). A simulator can then be used to collect observations from the candidate camera configuration (Fig. 17c). These observations are used by the perception model to compute the reward and loss (Fig. 174-7). The reward is used to train the CD and the loss is used to train the perception model. This loop repeats until a camera configuration and perception model have been created that maximize task accuracy.

RL Formulation: We transform the CFG into a state-action space which the RL agent, henceforth referred to as the CD, can search over. We use proximal policy optimization (PPO) to train the CD and model the RL problem with the following states, actions, and rewards:

- states, S: the possible states of the world, which, in our case, are the possible enumerations of illumination, optics, and sensors, and possible observations that can be captured from each enumeration.
- actions, A: the actions an agent can take at any step, which, our case, consist of choosing illumination, optics, sensors, algorithms, and all parameters.
- reward, R: the reward for taking an action in a state, which, in our case, is computed by passing observations from the candidate camera configuration into the PM to compute accuracy for a target task.

F.3 STEREO DEPTH ESTIMATION

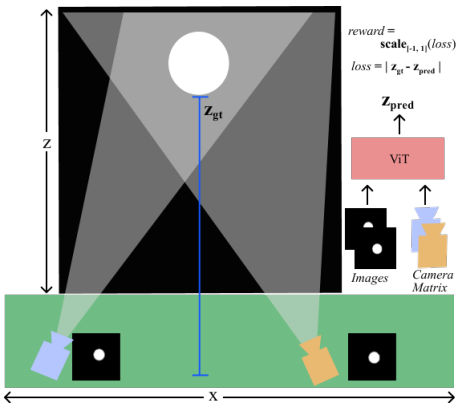

Figure 18: Depth from Stereo Setup: The goal of this experiment is to estimate the depth of a sphere using stereo cues. The camera designer (CD) places up to C cameras within the green box. Camera poses and images are input to the perception model (PM) which outputs a predicted depth. We render environments that are devoid of monocular cues to force (1) the CD to learn to obtain multi-view cues and (2) the PM to learn to exploit these cues.

F.3.1 Experimental Setup

Environment: The goal of the first experiment is to estimate the depth of a sphere using stereo cues. The CD is allowed to place a maximum of C cameras in the scene (though it can also place fewer cameras). In theory, the CD could place a single camera and learn monocular cues (e.g. shading/lighting, texture,

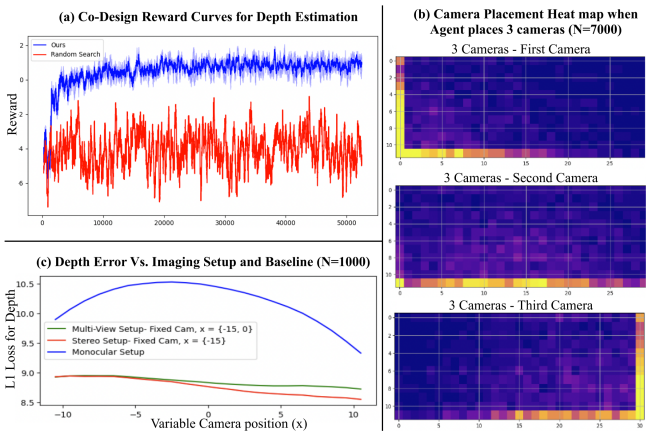

Figure 19: Joint Camera and Perception Design for Stereo Depth. We train the CD and PM from scratch to estimate depth of a sphere. (a) Our reward function consistently improves, even though it constantly changes due to the PM concurrently training with the CD. (b) The CD learns to maximize the baseline between different cameras over the course of 1000 experiments when placing 3 cameras. (c) The loss decreases with more placed cameras and larger distances between the cameras, which shows that the PM learns to exploit multi-view cues.

linear perspective). However, we simulate an environment where monocular cues are unavailable, making monocular depth estimation ill-posed.

Our environment consists of a randomly placed white sphere with a random radius, as shown in Fig. 18. We use PyRedner [57] to render images. The sphere position and radius are randomly sampled per episode from $(r, x, z) = \{r \in [3, 9], x \in [-10, 10], z \in [1, 60]\}$. The depth is the z distance from the sphere to the average position of the placed cameras. The scene is illuminated such that shading cues and the position of the light source are absent as cues. The only feedback that the PM and CD receive is a loss between the predicted and ground truth depth. The goal of rendering such an environment is to determine whether the CD can adapt to the context and realize that only a multi-view system can estimate depth. In parallel, the PM learns to exploit multi-view stereo cues. We show the supervised results of this experiment for validation in the supplement.

Action Space: The action space for depth estimation is $(p, x, z, \theta) = \{p \in [0, 1], x \in [-15, 15], z \in [69, 80], \theta \in [-60°, 60°]\}$, where p is camera placement probability, (x, z) is location (see Fig. 18) and θ is yaw. FoV is 45°.

Experiment Details: We use a modified version of the vision transformer (ViT) architecture [25] [2] that accepts an arbitrary number of images of fixed

resolution and their corresponding camera parameters as input, and outputs a scalar depth. The spatial resolution is fixed to $(128, 128)$. The maximum number of cameras the CD can place is set to $C = 5$. The CD's PPO backbone and the perception model share the same network architecture and are initialized randomly. The reward is computed before updating the perception model and is re-scaled to $[-1, 1]$. Additional information about the training is provided in the supplement.

F.3.2 Learned Agent & Model Analysis

We evaluate the joint training (Fig. 19a), the learned policy (Fig. 19b), and the perception model (Fig. 19c) in isolation. Fig. 19a illustrates how our system maximizes reward when co-designing the PM with the camera design. The reward function is dictated by the output of the PM, but the PM is concurrently training with the camera design, which results in inconsistent rewards during training for the same states. In spite of this fact, our model is able to consistently increase the reward, even at the beginning of training when the PM is untrained and with random initialization. Our results show that the CD and PM are able to learn intuitions that hold true in conventional multi-view stereo.

STRATEGY #1 – MAXIMIZE COVERAGE: When given the option to place up to 5 cameras, the CD places 1 camera 7.6% of the time and 2, 3, 4, and 5 cameras 27.7%, 36.6%, 22.7%, 5.4% times, respectively. Fig. 19b shows the heatmaps of where the CD decides to place each camera, specifically when the CD chose to place exactly three cameras. The heatmaps denote the number of times the CD placed the camera at a particular location over the course of 7000 experiments, where each experiment denotes the placement of a new random size sphere at a random location. From the heatmaps, we see that the CD strategically placed the cameras at locations that maximize the baselines between different cameras. Camera 1 was predominantly placed on the left side of the allowed region, camera 2 at the center bottom, and camera 3 at the right. From these results, we see that the CD optimizes to place more cameras spaced far apart. However, placing more cameras doesn't necessarily mean that the CD is obtaining multiple views of the object (e.g. some cameras may be pointed in the opposite direction of the object). Therefore, we account for this case by defining the metric of *coverage*, which defines the number of cameras that have at least one pixel viewing the object. The CD policy learns a configuration that maximizes coverage of the allowed region. We find that performance improves

as coverage increases from 0 to 3, with the L1 loss being 14.0, 9.2, 7.2, and 5.7 as the coverage increases. Coverage is discussed in detail in the supplementary.

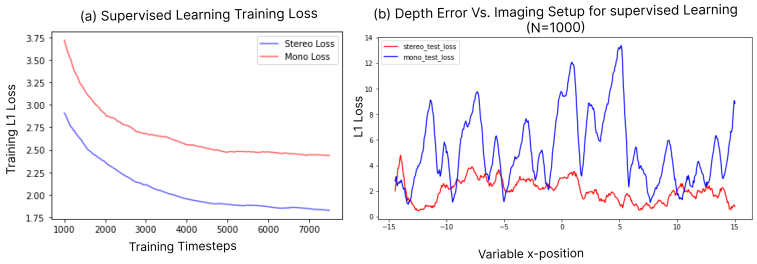

Figure 20: Learning Stereo Cues with Supervised Learning: We train two PMs – one on a one-camera configuration and one on a two-camera configuration. We show that PM trained with a two-camera configuration outperforms the one trained with one camera both during training and when evaluated on the same test set (5.40 vs. 3.78). This result verifies that the lack of monocular cues in our environment enables stereo setup to better estimate depth. In (b) we perform the baseline experiment (described in the main text) on the supervised models and show that the PM model trained in conjunction with the CD shows similar behavior of lower overall depth error and variance with the two-camera setup.

Strategy #2 – Multi-View Cues and Maximal Baseline: Fig. 19c shows that the PM learns to exploit stereo cues when presented with multiple images. The experiment shown here compares the PM performance on a one-camera, two-camera, and three-camera system when estimating the depth of a sphere (averaged over 1000 different spheres of varying size and depth). All three systems have a camera that can be moved along the x axis, the two- and three-camera system have a fixed camera at $x = -15$, and the three-camera system has an additional fixed camera at $x = 0$. The blue curve illustrates the L1 loss between the ground truth and one-camera system predictions. The red and green curves illustrate the performance of the two-camera and three-camera system respectively. The three-camera system performs slightly better than the two-camera system, and both perform significantly better than the one-camera system. The multi-view systems also see a decrease in loss (and variance) as the baseline between the cameras increases (i.e. as the movable camera moves along the $+x$ axis). These curves indicate that the PM has learned similar wisdom to that of conventional stereo, which states that multiple views with a large baseline enable better depth estimation [10].

Cam Config	Mean (x,z)	Std (x,z)	Mean Yaw	Std Yaw
1	$(-4.6, 79.2)$	$(10.0, 1.9)$	-15.7	39.8
2	$(-8.3, 78.3)$	$(7.8, 2.7)$	-3.6 8.8	43.3
	$(4.6, 77.7)$	$(9.1, 3.2)$		43.7
3	$(-10.4, 77.8)$	$(6.4, 2.9)$	-0.6 9.3	43.7
	$(-1.1, 77.6)$	$(8.6, 3.1)$	15.4	43.1
	$(8.5, 77.3)$	$(7.2, 3.3)$		41.2
4	$(-11.4, 77.7)$	$(5.4, 3.0)$	3.2 11.4	45.1
	$(-4.3, 77.6)$	$(7.5, 3.2)$	15.1	43.5
	$(3.5, 77.2)$	$(7.5, 3.2)$	17.0	41.7
	$(10.9, 77.4)$	$(5.5, 3.2)$		40.7
5	$(-12.1, 77.7)$	$(4.6, 3.1)$	5.4 8.0	43.4
	$(-6.5, 77.9)$	$(6.7, 3.0)$	14.0	44.2
	$(-0.17, 77.4)$	$(7.3, 3.3)$	17.9	41.5
	$(6.6, 77.1)$	$(6.8, 3.4)$	18.7	41.7
	$(12.2, 77.2)$	$(4.5, 3.3)$		40.5

Table 4: **Distribution CD Actions**: We show the mean and standard deviation of the actions taken by the CD after training. The CD always chooses to place a camera in the back of the allowed region (green box in Fig. 4) while spreading the rest of the cameras across the x-axis (mean x-position cover the entire box). For instance, the largest baseline between 3,4 and 5 cameras are roughly the same as the CD maximizes the spread of cameras along the x-axis while minimizing the z-axis variation. Additionally, the yaw has the largest variance of the parameters, which suggests that the CD has learned a strategy that exploits the yaw to find the object instead of the position.

F.4 ADDITIONAL RESULTS

We provide additional experimental results and details below. We refer to the camera designer as CD and the perception model as PM.

F.4.1 Depth Estimation

We show that the CD and PM are able to learn intuitions that hold true in conventional multi-view stereo. We evaluate the individual components, the CD and PM, in isolation in Fig. 5 of the main paper. We now discuss additional results by analyzing the distribution of actions taken by CD, and the results from supervised training of the PM.

Coverage	L1 Loss
0	14.0
1	9.2
2	7.2
3	5.7

Table 5: We show that the L1 loss consistently decreases as more cameras see the sphere.

Distribution of Actions: Table 4 shows the mean and standard deviations (std) over 7,000 trials for actions taken by the CD based on the final camera configuration (number of cameras) at the end of the episode. We notice that regardless of the camera configuration, the CD almost always chooses to place the camera at the back of the allowed region, maximizing distance to the scene, and thus allowing more of it in its field of view. It maximizes z-position (max: 80) and has a very small std. The mean x-position of the camera is always maximized. For example, the largest baseline between the furthest cameras for camera configurations with 3, 4, and 5 cameras is roughly the same. For the 2 camera-configuration, the left camera is placed at -8.3, and the right camera is placed at 4.6 which is not as wide as it could be. The mean yaw of this configuration shows that on average the cameras face opposing directions: -3.6° for the left camera, and $+8.8^\circ$ for the right. Moreover, the x-position std is high for the right camera so the limited baseline could be due to the narrow FoV (45°).

We also note the distribution of yaw angles in the camera configurations. In the 2-camera configuration, the yaw angles oppose each other and, as the CD adds more cameras, the yaw of the left-most camera reduces to 0° while the right ones have a yaw 15° to the right. Lastly, we note that the yaw angles have the highest std when compared to the x and z positions' std, which suggests that the CD might have learned to fix the cameras around a certain area and rather exploit the yaw, range of $[-60, 60]$, to find the object.

Supervised Learning: To verify if the PM can indeed learn to estimate accurate depth with stereo, rather than monocular, cues in our environment, we conduct a supervised experiment. We first create two datasets, monocular and stereo, by randomly sampling the sphere from the same region described in Section 4.1.1 of the main text. For each sampled sphere, we sample a random position directly in front of the sphere and place a monocular camera looking at the object. We also randomly sample two cameras a random distance apart (such that the sphere is visible to both cameras). The two images from two cameras form the stereo pair for the two camera-configuration setup. We sample

7,500 training samples and then train two PMs in a supervised fashion (same architecture): one network on the one camera-configuration dataset and another on the two camera-configuration dataset.

We show plots in Figure 20 of the training loss and baseline experiments. We show that the PM model jointly trained with the CD, shown in the main text, achieves similar results as the supervised model. Specifically, Fig. 20.a shows that the training loss for the two camera-configuration PM is substantially lower, verifying that the lack of monocular cues in our environment enables the stereo setup to achieve more accurate depth estimation. Moreover, on the test sets, the stereo setup outperformed the monocular with a test L1 loss of 3.78 vs. 5.40 respectively. In Fig. 20.b, we perform the baseline experiment, described in Sec.4.1.2 of the main text. We show that the behavior of the PM is similar to the joint training setup from the main text i.e. the PM model trained with stereo setup estimates more accurate depth and has lower variance than the monocular setup- which is subject to the size of the sphere and position of the camera w.r.t sphere. However, the primary difference between the experiments is that only the number of cameras, not the baseline between the cameras, has an effect on the L1 error in the supervised settings.

BIBLIOGRAPHY

[1] Anna Alperovich, Ole Johannsen, and Bastian Goldluecke. Intrinsic light field decomposition and disparity estimation with deep encoder-decoder network. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2165–2169. IEEE, 2018.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826, 2021.

[3] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. In *International Conference on Learning Representations*.

[4] S. Baker and S.K. Nayar. A theory of catadioptric image formation. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 35–42, 1998.

[5] Simon Baker and Shree K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35:175–196, 2004.

[6] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2014.

[7] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.

[8] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[9] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[10] Ayush Bhandari, Achuta Kadambi, and Ramesh Raskar. *Computational Imaging*. 2022.

[11] Daniel G. Bobrow. *Comment on "Numerical shape from shading and occluding boundaries"*, pages 89–94. The MIT Press, 1994.

[12] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021.

[13] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34:10691–10704, 2021.

[14] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[15] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[16] Ayan Chakrabarti. Learning sensor multiplexing design through back-propagation. *Advances in Neural Information Processing Systems*, 29, 2016.

[17] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific reports*, 8(1):1–10, 2018.

[18] Julie Chang and Gordon Wetzstein. Deep optics for monocular depth estimation and 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10193–10202, 2019.

[19] Thomas W Cronin, Sönke Johnsen, N Justin Marshall, and Eric J Warrant. Visual ecology. In *Visual Ecology*. Princeton University Press, 2014.

[20] Franklin C Crow. Shadow algorithms for computer graphics. *Acm siggraph computer graphics*, 11(2):242–248, 1977.

[21] Akshat Dave, Yannick Hold-Geoffroy, Miloš Hašan, Kalyan Sunkavalli, and Ashok Veeraraghavan. Snapshot polarimetric diffuse-specular separation. *Optics Express*, 30(19):34239–34255, 2022.

[22] Akshat Dave, Yongyi Zhao, and Ashok Veeraraghavan. Pandora: Polarization-aided neural decomposition of radiance. *arXiv preprint arXiv:2203.13458*, 2022.

[23] Michael De Zeeuw and Aswin C Sankaranarayanan. Wide-baseline light fields using ellipsoidal mirrors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[24] Philipp Del Hougne, Mohammadreza F Imani, Aaron V Diebold, Roarke Horstmeyer, and David R Smith. Learned integrated sensing pipeline: Reconfigurable metasurface transceivers as trainable physical layer in an artificial neural network. *Advanced Science*, 7(3):1901913, 2020.

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[26] Iddo Drori, Yamuna Krishnamurthy, Raoni Lourenço, Rémi Rampin, Kyunghyun Cho, Cláudio T. Silva, and Juliana Freire. Automatic machine learning by pipeline synthesis using model-based reinforcement learning and a grammar. *CoRR*, abs/1905.10345, 2019.

[27] Epic Games. Unreal engine.

[28] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

[29] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2019.

[30] Joseph M Geary. *Introduction to lens design: with practical ZEMAX examples*. Willmann-Bell Richmond, VA, USA:, 2002.

[31] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5170–5178, 2017.

[32] J. Gluckman and S.K. Nayar. Planar catadioptric stereo: geometry and calibration. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 22–28 Vol. 1, 1999.

[33] Michael D Grossberg and Shree K Nayar. The raxel imaging model and ray-based calibration. *International Journal of Computer Vision*, 61(2):119, 2005.

[34] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18409–18418, June 2022.

[35] Yuanchen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. *CoRR*, abs/2111.15234, 2021.

[36] John K Haas. A history of the unity game engine. 2014.

[37] Harel Haim, Shay Elmalem, Raja Giryes, Alex M Bronstein, and Emanuel Marom. Depth estimation from a single image using deep learned phase coded mask. *IEEE Transactions on Computational Imaging*, 4(3):298–310, 2018.

[38] Lei He, Guanghui Wang, and Zhanyi Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, 27(9):4676–4689, 2018.

[39] Connor Henley, Tomohiro Maeda, Tristan Swedish, and Ramesh Raskar. Imaging behind occluders using two-bounce light. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 573–588, Cham, 2020. Springer International Publishing.

[40] Connor Henley, Tomohiro Maeda, Tristan Swedish, and Ramesh Raskar. Imaging behind occluders using two-bounce light. In *European Conference on Computer Vision*, pages 573–588. Springer, 2020.

[41] Connor Henley, Siddharth Somasundaram, Joseph Hollmann, and Ramesh Raskar. Detection and mapping of specular surfaces using multi-bounce lidar returns. *Optics Express*, 31(4):6370–6388, 2023.

[42] Jan Jachnik, Richard A Newcombe, and Andrew J Davison. Real-time surface light-field capture for augmentation of planar specular surfaces. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 91–97. IEEE, 2012.

[43] Wojciech Jarosz. *Efficient Monte Carlo Methods for Light Transport in Scattering Media*. PhD thesis, UC San Diego, September 2008.

[44] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[45] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, page 143–150, New York, NY, USA, 1986. Association for Computing Machinery.

[46] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] Michael Katz, Parikshit Ram, Shirin Sohrabi, and Octavian Udrea. Exploring context-free languages via planning: The case for automating machine learning. *Proceedings of the International Conference on Automated Planning and Scheduling*, 30(1):403–411, Jun. 2020.

[48] Tzofi Klinghoffer, Siddharth Somasundaram, Kushagra Tiwary, and Ramesh Raskar. Physics vs. learned priors: Rethinking camera and algorithm design for task-specific imaging. *arXiv preprint arXiv:2204.09871*, 2022.

[49] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[50] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics*, 41(6):Article–201, 2022.

[51] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. NeROIC: Neural object capture and rendering from online image collections. *Computing Research Repository (CoRR)*, abs/2201.02533, 2022.

[52] Michael F Land and Dan-Eric Nilsson. *Animal eyes*. OUP Oxford, 2012.

[53] José Luis Landabaso, Montse Pardàs, and Josep Ramon Casas. Shape from inconsistent silhouette. *Comput. Vis. Image Underst.*, 112:210–224, 2008.

[54] Rui Li, Simeng Qiu, Guangming Zang, and Wolfgang Heidrich. Reflection separation via multi-bounce polarization state tracing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 781–796. Springer, 2020.

[55] Tingtian Li, Daniel PK Lun, Yuk-Hee Chan, et al. Robust reflection removal based on light field imaging. *IEEE Transactions on Image Processing*, 28(4):1798–1812, 2018.

[56] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.

[57] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.

[58] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020.

[59] Stephen Lombardi and Ko Nishino. Reflectance and natural illumination from a single image. In *European Conference on Computer Vision*, pages 582–595. Springer, 2012.

[60] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019.

[61] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[62] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[63] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5695–5703, 2016.

[64] Radu Marinescu, Akihiro Kishimoto, Parikshit Ram, Ambrish Rawat, Martin Wistuba, Paulito P. Palmes, and Adi Botea. Searching for machine learning pipelines using a context-free grammar. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):8902–8911, May 2021.

[65] Worthy N. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):150–158, 1983.

[66] Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021.

[67] Christopher A Metzler, Hayato Ikoma, Yifan Peng, and Gordon Wetzstein. Deep optics for single-shot high-dynamic-range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1375–1385, 2020.

[68] 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 *The European Conference on Computer Vision (ECCV)*, 2020.

[69] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[70] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

[71] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022.

[72] Shree K Nayar and Simon Baker. Catadioptric image formation. In *Proceedings of the 1997 DARPA Image Understanding Workshop*, pages 1431–1437, 1997.

[73] Yun Ni, Jie Chen, and Lap-Pui Chau. Reflection removal on single light field capture using focus manipulation. *IEEE Transactions on Computational Imaging*, 4(4):562–572, 2018.

[74] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. *https://arxiv.org/abs/2011.12100*, 2020.

[75] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[76] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[77] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)*, 38(6):1–17, 2019.

[78] Tiago Novello, Guilherme Schardong, Luiz Schirmer, Vinicius da Silva, Helio Lopes, and Luiz Velho. Exploring differential geometry in neural implicits, 2022.

[79] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021.

[80] Emmanuel Onzon, Fahim Mannan, and Felix Heide. Neural auto-exposure for high-dynamic range object detection. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7710–7720, 2021.

[81] Jeong Joon Park, Pete Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.

[82] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1417–1427, 2020.

[83] Chen Quei-An. Nerf_pl: a pytorch-lightning implementation of nerf, 2020.

[84] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128, 2001.

[85] Raskar Ramesh and James Davis. 5d time-light transport matrix: What can we reason about scene properties? Technical report, 2008.

[86] Nicolas Robidoux, Luis E Garcia Capel, Dong-eun Seo, Avinash Sharma, Federico Ariza, and Felix Heide. End-to-end high dynamic range camera pipeline optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6297–6307, 2021.

[87] Fabiano Romeiro and Todd Zickler. Blind reflectometry. In *European conference on computer vision*, pages 45–58. Springer, 2010.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[89] Silvio Savarese, Holly Rushmeier, Fausto Bernardini, and Pietro Perona. Shadow carving. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 190–197. IEEE, 2001.

[90] Jonathan Schaeffer and Aske Plaat. New advances in alpha-beta searching. In *Proceedings of the 1996 ACM 24th annual conference on Computer science*, pages 124–130, 1996.

[91] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[92] 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 Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[93] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34:19313–19325, 2021.

[94] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019.

[95] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6918–6926, 2019.

[96] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis, 2020.

[97] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2020.

[98] Karl Stelzner, Kristian Kersting, and Adam R. Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation, 2021.

[99] Qilin Sun, Ethan Tseng, Qiang Fu, Wolfgang Heidrich, and Felix Heide. Learning rank-1 diffractive optics for single-shot high dynamic range imaging. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1386–1396, 2020.

[100] R. Swaminathan, M.D. Grossberg, and S.K. Nayar. Caustics of catadioptric cameras. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 2–9 vol.2, 2001.

[101] Tristan Swedish, Connor Henley, and Ramesh Raskar. Objects as cameras: Estimating high-frequency illumination from shadows. In *Proceedings of*

the IEEE/CVF International Conference on Computer Vision, pages 2593–2602, 2021.

[102] Yuichi Taguchi, Amit Agrawal, Ashok Veeraraghavan, Srikumar Ramalingam, and Ramesh Raskar. Axial-cones: Modeling spherical catadioptric cameras for wide-angle light field rendering. *ACM Trans. Graph.*, 29(6), dec 2010.

[103] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020.

[104] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhöfer. State of the art on neural rendering, 2020.

[105] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhoefer, and Vladislav Golyanik. Advances in neural rendering, 2021.

[106] Kushagra Tiwary, Akshat Dave, Nikhil Behari, Tzofi Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. Orca: Glossy objects as radiance field cameras, 2022.

[107] Kushagra Tiwary, Tzofi Klinghoffer, and Ramesh Raskar. Towards learning neural representations from shadows. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 300–316. Springer, 2022.

[108] Kushagra Tiwary, Tzofi Klinghoffer, and Ramesh Raskar. Towards learning neural representations from shadows, 2022.

[109] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Regognition (CVPR)*, 2018.

[110] Hernan Ceferino Vazquez, Jorge Sánchez, and Rafael Carrascosa. GramML: Exploring context-free grammars with model-free reinforcement learning. In *Sixth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2022.

[111] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G. Bawendi, and Ramesh Raskar. Recovering threedimensional shape around a corner using ultrafast time-of-flight imaging. *Nature*, page 745, 2012.

[112] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *arXiv*, 2021.

[113] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022.

[114] Oliver Vogel, Levi Valgaerts, Michael Breuß, and Joachim Weickert. Making shape from shading work for real-world images. In Joachim Denzler, Gunther Notni, and Herbert Süße, editors, *Pattern Recognition*, pages 191–200, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[115] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[116] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. Learning indoor inverse rendering with 3d spatially-varying lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12538–12547, 2021.

[117] Ziyu Wang, Liao Wang, Fuqiang Zhao, Minye Wu, Lan Xu, and Jingyi Yu. Mirrornerf: One-shot neural portrait radiance field from multi-mirror catadioptric imaging. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2021.

[118] Jakob Weiss and Nassir Navab. Deep direct volume rendering: Learning visual feature mappings from exemplary images. *arXiv preprint arXiv:2106.05429*, 2021.

[119] Lance Williams. Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, 1978.

[120] Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown. Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling, 2021.

[121] Shuntaro Yamazaki, G Srinivasa Narasimhan, Simon Baker, and Takeo Kanade. The theory and practice of coplanar shadowgram imaging for acquiring visual hulls of intricate objects. *International Journal of Computer Vision*, 81, 03 2009.

[122] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[123] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[124] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In *Conference on Neural Information Processing Systems*, 2021.

[125] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021.

[126] Ruo Zhang, Ping-Sing Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[127] Tianyu Zhang, Amin Banitalebi-Dehkordi, and Yong Zhang. Deep reinforcement learning for exact combinatorial optimization: Learning to branch. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3105–3111. IEEE, 2022.

[128] 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 Transactions on Graphics (TOG)*, 40(6):1–18, 2021.

[129] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):680–702, 1991.

[130] Rui Zhu, Zhengqin Li, Janarbek Matai, Fatih Porikli, and Manmohan Chandraker. Irisformer: Dense vision transformers for single-image inverse rendering in indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2822–2831, 2022.