TexHOI: Reconstructing Textures of 3D Unknown Objects in Monocular Hand-Object Interaction Scenes

Alakh Aggarwal
The University of Texas at Dallas
alakh.aggarwal@utdallas.edu

Ningna Wang
The University of Texas at Dallas
ningna.wang@utdallas.edu

Xiaohu Guo
The University of Texas at Dallas
xguo@utdallas.edu

Abstract

Reconstructing 3D models of dynamic, real-world objects with high-fidelity textures from monocular frame sequences has been a challenging problem in recent years. This difficulty stems from factors such as shadows, indirect illumination, and inaccurate object-pose estimations due to occluding hand-object interactions. To address these challenges, we propose a novel approach that predicts the hand's impact on environmental visibility and indirect illumination on the object's surface albedo. Our method first learns the geometry and low-fidelity texture of the object, hand, and background through composite rendering of radiance fields. Simultaneously, we optimize the hand and object poses to achieve accurate object-pose estimations. We then refine physics-based rendering parameters—including roughness, specularity, albedo, hand visibility, skin color reflections, and environmental illumination—to produce precise albedo, and accurate hand illumination and shadow regions. Our approach surpasses state-of-the-art methods in texture reconstruction and, to the best of our knowledge, is the first to account for hand-object interactions in object texture reconstruction. Please check our work at: https://alakhag.github.io/TexHOI-website/

Keywords: Radiance Field, Differentiable Renderer, Spherical Gaussian, Albedo Prediction

1. Introduction

Every day, we interact with common objects like sugar boxes, snack containers, and cleaning supplies. Our hands naturally pick them up, move them, and manipulate them to complete routine tasks. These seemingly simple interactions, however, involve complex hand-object dynamics that challenge even the most advanced computer vision systems in accurately modeling and replicating their texture in virtual environments.

In recent years, there has been an increase in the availability of standardized datasets designed to capture hand-object interactions, such as HO3D [21] and DexYCB [7], which are built on YCB-models [56], or Contact-Pose [5] dataset. These datasets provide crucial insights into how hands manipulate everyday objects, like sugar boxes and snack containers, in dynamic scenarios. However, a major challenge remains in effectively utilizing these datasets due to the complex shadowing and reflection effects that hands cast on object surfaces. Accurately capturing and rendering these effects is critical for a wide range of applications, from enhancing realism in virtual and augmented reality environments to improving robotic manipulation and object recognition systems. Several recent works have focused on reconstructing hand-object interactions. Methods like BundleSDF [52], DiffHOI [64], and HOLD [17] primarily address geometry reconstruction. However, these methods struggle with generating accurate textures under varying environmental illumination conditions. Moreover, shadows and indirect illumination caused by hand movements introduce significant errors in texture prediction. To the best of our knowledge, no recent work explicitly addresses the influence of shadows cast by dynamic elements like hands on 3D object reconstruction, particularly texture reconstruction. Other works such as PhySG [66], InvRender [68], and RefNeRF [47] focus on modeling the influence of environmental illumination on object surfaces based on material properties, typically represented as Spherical Harmonics or Spherical Gaussian. However, these methods do not account for interacting non-rigid parts like hands, making them unable to handle hand-induced shadows and reflections. Additionally, they rely on precise camera pose estimations based on object-coordinate systems, which can introduce errors akin to motion blur when these assumptions are violated. Given the above challenges, our primary

research question is: How can we achieve accurate texture and geometry predictions in dynamic hand-object interaction scenarios using monocular camera data?

In this paper, we present a novel framework to accurately predict the object's true albedo, specular properties, and surface geometry, given a sequence of frames captured by a single camera showing a hand interacting with a rigid object. Our focus is on accounting for the indirect effects of hand movements, such as shadows, reflections, and visibility changes, across different frames. Ultimately, we aim to enhance the realism and accuracy of virtual object rendering in complex hand-object interaction scenarios.

Our approach begins by optimizing both hand and object poses using a composite rendering technique that separately models the hand, object, and background. In this process, we simultaneously refine hand and object poses while learning neural radiance fields for each entity, ensuring accurate geometric and pose representations. Using these fine-tuned poses, we represent the environmental illumination with Spherical Gaussian (SG) representations. This allows us to model the interaction between the object's material properties — such as specularity and roughness — and the environment. To efficiently compute the shadows and reflections caused by interacting hands, we introduce a novel representation of the hand using 108 parameterizable spheres. For each surface point on the object, we calculate the visibility and indirect illumination caused by each sphere by measuring the fraction of its projection that occludes the SG on the object's surface. This method avoids the need for time-consuming ray-tracing [67] or learning pseudo-ground truth visibility using MLPs [68]. We then combine all the elements in the physics-based rendering (PBR) framework to obtain the final object color, which is compared with the ground truth camera-captured images to optimize all those physics-based rendering parameters, i.e., roughness, specularity, albedo, hand visibility, skin color reflections, and environmental illumination.

Our approach surpasses state-of-the-art methods in texture reconstruction quality, for dynamic hand-object interaction scenarios. We summarize our major contributions as follows:

- To the best of our knowledge, this is the first method to account for dynamic hand-object interaction in the context of texture prediction, for 3D object reconstruction from monocular videos.

- We introduce a novel hand representation where the canonical hand is simplified as 108 parameterizable spheres. Skinning weights are allocated to each sphere based on the distance to surface points, enabling accurate pose transformations. The radius of each sphere is recalculated dynamically according to the transformed surface points. These spheres enable efficient calcu-

lation of Spherical Gaussian occlusion by the hand, eliminating the need for time-consuming ray-tracing computations for occlusion computation.

- We develop a two-stage framework for 3D object reconstruction in hand-object interactions. The first stage uses compositional volumetric rendering to optimize object and hand poses, while the second stage uses Spherical Gaussian-based surface rendering to refine object texture by removing hand occlusions and lighting effects, ensuring high-fidelity reconstruction.

2. Related Works

Creating realistic rendered images of a target object, for its re-usability in varying virtual environment has been a major focus in the study of computer graphics for much of its history. Inverse rendering of an object from their visual data refers to predicting camera poses and scene descriptions, including the geometry of the object, the reflectance of the object surface, which consists of its albedo texture and other microfacet surface properties like roughness, specularity, etc., and environment lighting used to illuminate the scene [33]. Many research works have built the basic framework for upcoming advancements in this area of research.

2.1. Traditional Inverse Rendering for 3D Object Reconstruction

Traditional inverse rendering approaches broadly divided the research field into three categories - **inverse lighting** to predict the environment light that illuminates the scene in visual input such as images, **inverse reflectometry** to predict object texture or object BRDF from visual input, or **shape reconstruction** to obtain the geometry of the object. Traditional approaches only focus on solving one of the three mentioned problems. They may solve inverse lighting problem [26] which allows re-lighting of rendered scenes under different environment illumination [36, 45]. Some other works solve inverse reflectometry problems to obtain realistic texture [46, 54] or accurate reflective model [13, 16] on the object surface. Meanwhile, other approaches optimize object geometry [37, 42, 24, 53]. These approaches assume other aspects of the inverse rendering, except their focus problem, to be known. To our knowledge, no traditional approach comprehensively addresses all aspects of the inverse rendering problem in a single study. Recent advancements in neural representations address the comprehensive computation of these different aspects.

2.2. Neural Representation in Inverse Rendering

In recent years, several recent neural network-based research works have sparked a wave of innovative method-

ologies to solve inverse rendering problems. These include volumetric rendering based approaches like NeRF [34], NeuS [50], DVGO [44], TensoRF [8], Instant-NGP [35], etc., surface rendering based approaches like IDR [63] or point-based inverse rendering approaches like 3D-Gaussian-Splatting [27]. Moreover, these approaches give a general framework solution for inverse rendering problems but do not solve complex problems that arise from varying material properties of objects, environment illumination in the scene, or occlusion and visibility handling in multi-object interactions in the scene. Applying the physics-based knowledge, and addressing the composition of multiple-object interactions in the scene further improves this research area, as will be seen in the next two subsections.

2.3. Physics-based Inverse Rendering

The Physics-based Rendering (PBR) equation gives the rendered output color c at a surface point x of an object along the view direction ω_o, given the illumination on the surface $L(\omega_i, x)$ from incoming direction ω_i, BRDF property of the object $\phi(x, \omega_o, \omega_i)$ and object normal n at x. The equation is defined as follows:

$$c(\omega_o; x) = \int_{\omega_i \in \Omega^+(n)} L(\omega_i, x)\phi(x; \omega_o, \omega_i)(\omega_i \cdot n), \quad (1)$$

where ω_i is a direction in hemi-sphere $\Omega^+(n)$, defined with the up direction as surface normal n. $L(\omega_i, x)$ is obtained by inverse lighting, $\phi(x; \omega_o, \omega_i)$ is obtained by inverse reflectometry and normal n is obtained by shape reconstruction. Occlusions between the illumination source and surface point x may break down $L(x)$ term, requiring visibility and indirect illumination calculation. Many recent approaches have done extensive research on improving the inverse rendering output by predicting the above terms of the PBR equation. NeRV [41] assumes known illumination. Other methods like NeRD [2], Neural Ray-Tracing [29], NeRFactor [67], Neural-PIL [4], PS-NeRF [59], Neilf [62], SAMURAI [3], NVDiffRec [22], Neural-PBIR [43], TensoIR [25], TensoSDF [30], MIRReS [14], IRON [65] etc. use volumetric-rendering based methods to optimize neural radiance fields, and extract geometry, illumination, texture and reflectance model. NeRO [32], NeP [48], PBIR-NIE [6] extend these methods to glossy or reflective surfaces. Relightable-3D-Gaussian [18] uses 3D Gaussian splats, and methods like IntrinsicAnything [9] learn diffusion priors to obtain accurate albedo and material properties. Methods like PhySG [66] and InvRender [68] use surface-based rendering approaches and approximate illumination and BRDF as Spherical Gaussians (SGs). An SG comprises of a lobe with its direction ξ, intensity μ and width η, and is defined on a spherical surface as follows:

$$G(v; \xi, \mu, \eta) = \mu e^{\eta(v \cdot \xi - 1)}, \quad (2)$$

where, v is any direction on the sphere and $G(v)$ is the value on the sphere along direction v. The above methods either ignore visibility computation [66], perform computationally-extensive Monte-Carlo ray tracing [67], or estimate pseudo-ground truth visibility using a Multi-layer Perceptron (MLP) [68]. Monte-Carlo ray-tracing can take up to 30 minutes per frame on a single GPU to train. L-Tracing [10] utilizes sphere-tracing, and the concavity/convexity of geometry to avoid learning visibility. However, both InvRender [68] and L-Tracing [10] work for a single static rigid object, and fail to handle multiple dynamic object visibility estimation. NeRF-DS [57] handles dynamic objects but does not utilize the PBR equation to obtain the reflectometry information in the scene. Instead, the method is only restricted to glossy objects. Moreover, all the above methods assume a single rigid object with accurate pose estimations in the scene. These methods do not handle multiple dynamic object interactions in the scene.

2.4. Handling Dynamic Elements in Multi-Object Interaction

To handle multiple object interactions, along with background, many recent researches have come up, with compositional-NeRF methods [51, 15, 55, 58]. Hand-NeRF [20] extends this compositional-NeRF methodology for multiple interacting hands. Other methods named Hand-NeRF [12], BundleSDF [52], HOLD [17], etc. focus on dynamic hand-object reconstruction. However, their primary focus is geometry reconstruction, and do not focus on high-fidelity texture reconstruction. Hence, they do not focus on physics-based accuracy in results. Our research primarily focuses on 3D geometry and high-fidelity texture reconstruction of hand-object interactions from single-camera data capture sequences. We aim to improve the output by resolving the baking of hand effects into the rendered object texture, and obtain a high-fidelity object albedo prediction following a two-stage training process. In the first stage, we fine-tune hand and object poses using a compositional volumetric rendering approach [17]. In the second stage, we use a surface rendering approach with Spherical Gaussians to obtain different elements in the PBR equation, including hand effects on the object surface. In Sec. 4, we will describe the above two stages of training in-depth.

3. Preliminaries

3.1. Volumetric Rendering

For each pixel in the image space, the volumetric rendering samples the points uniformly along a ray originating from the camera pointing in the direction of the pixel. These points are distributed between the near and far bounds of the camera's viewing range. Based on the predicted density values at these uniformly sampled points, an importance

sampling technique refines the sampling process, drawing points closer to the estimated implicit surface. For each sampled point, the color is predicted using the view direction and the point's properties. The predicted colors are then aggregated along the ray through an integral, resulting in the final color for that pixel. Formally, along a camera ray $r(t) = o + td$ with camera location o and view direction d, near and far bounds t_n and t_f, density at a point is estimated as $\sigma(r(t))$ and color at a point is estimated as $c(r(t), d)$. The expected final color on the pixel $C(r(t))$ can thus be expressed as the following integral:

$$C(r(t)) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt, \qquad (3)$$

where $T(t) = exp(-\int_{t_n}^{t} \sigma(r(s))ds)$ represents the accumulated transmittance, which accounts for the occlusion of light as it passes through the scene. This approach captures both the density and color contributions along the entire ray, resulting in realistic volumetric effects. Methods like NeRF [34] use volumetric rendering.

3.2. Surface Rendering

The surface rendering focuses on identifying the exact intersection point $\hat{t}$ of the ray $r(\hat{t}) = o + \hat{t}d$ with the implicit surface geometry using the sphere tracing approach. Once the intersection point is determined, the color at that point is predicted as $c(r(\hat{t}), d)$. After calculating the surface color, it is assigned to the corresponding pixel in the image, forming the final rendered image. Methods like IDR [63], PhySG [66], etc. use surface rendering.

4. Methodology

This section outlines the methodological approach adopted to achieve high-fidelity geometry and texture reconstruction of hand-object interaction from monocular camera data sequence. We discuss in-depth both the stages of the two-stage process, called TexHOI, namely volumetric rendering based compositional NeRF as **Stage 1** (Sec. 4.1) to fine-tune object and hand pose and obtain a low-level of geometry and texture information, and surface rendering based Physics-Based Rendering (PBR) equation, approximated with Spherical Gaussians as **Stage 2** (Sec. 4.2) to generate hand shadows, reflections and environmental illumination, along with studying the material properties. We also discuss the data acquisition/processing steps in both stages, including hand and object pose estimation in stage 1, and visibility and reflectance property computation in stage 2. This two-stage process of computation of different elements leads to a high-fidelity albedo texture prediction. Fig. 1 shows the full pipeline.

4.1. Stage 1: Compositional Neural Radiance Field for Pose Refinement

The input of stage 1 of our system is the frame sequences from a monocular camera, along with an estimated hand and object poses and generates low-level geometry and texture predictions of hand and object.

This stage utilizes three Neural Radiance Fields (NeRF). NeRF for hand optimizes the hand pose transformation and generates hand from monocular camera frames in MANO [38] canonical pose. Similarly, a NeRF for the object optimizes the object pose transformation and generates the rigid object in its canonical pose, or world origin. Meanwhile, another NeRF learns background image representation. Finally, the outputs of the three NeRFs are composed to generate a composite rendered image similar to the ground-truth image frame. In this section, we will discuss in detail the above-mentioned NeRFs and their composition.

The geometry prediction implicit network, along with the fine-tuned hand and object poses prepare the input of the second stage of our pipeline.

4.1.1 Object NeRF

Object NeRF uses object pose estimation to calculate neural network-based radiance field. The points sampled along the camera ray are inverse-transformed to the object coordinate system using object pose. The density and color are thus predicted in the object coordinate system, or object canonical space. Given object pose with rotation R and translation t, we get the canonical space point x_{obj} from observation space point x' as follows:

$$x_{obj} = R^{-1}(x' - t). \qquad (4)$$

Object NeRF, then, trains on the transformed points in object canonical space.

4.1.2 Hand NeRF

Calculating the radiance field for hand NeRF follows a similar setup as object NeRF. The points sampled along the camera ray are inverse-transformed to the hand's canonical pose, i.e., stretched hand, using MANO [38] pose parameters. Hence, we again need to perform the inverse transformation for hand. Given transformation matrices T_i for i^{th} of n_{jnt} joints, with interpolated skinning weight $w_i(x')$ on the observation space point x', its corresponding canonical space point x_{hand} is calculated as follows:

$$x_{hand} = (\Sigma_{i=1}^{n_{jnt}}(w_i(x')T_i)^{-1}x'. \qquad (5)$$

4.1.3 Compositional NeRF

To generate a composite rendered image similar to a ground-truth frame, we also learn Background NeRF. Af-
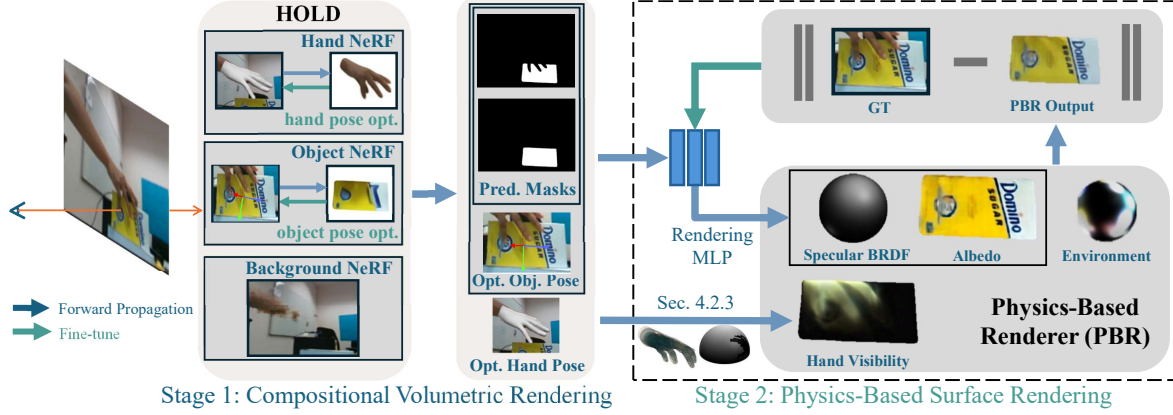
Figure 1. The overall pipeline of our proposed TexHOI method. In the first stage (Sec. 4.1), the hand and object poses are fine-tuned along with composite radiance fields of hand, object, and background. Using the predicted object segmentation and object geometry mask, in the second stage (Sec. 4.2), the optimized hand and object poses are used to accurately learn material properties on the object surface, *i.e.* albedo, BRDF, and hand occlusions, using physics-based rendering with Spherical Gaussian approximations.

ter obtaining color and density predictions from each NeRF, we arrange all the sample points in sorted order from their distance. The other information at the sample points, i.e., density and color are also sorted similarly. We then calculate the aggregate integral and get the expected color at the pixel corresponding to the ray direction.

Mathematically, sampling n points for object and inverse transforming the points into object-coordinate system 4.1.1, and n points for hand and performing linear inverse blend-skinning into hand's canonical coordinate system 4.1.2, we obtain the foreground (hand and object) color C_{fg} and foreground mask M_{fg}. The values are calculated by sorting the $2n$ points, and the predicted density σ_i and radiance c_i values of each point. Similarly, using the calculated foreground mask M_{fg} and the background color C_{bg}, we obtain final predicted color along a ray r as follows:

$$C(r) = C_{fg}(r) + (1 - M_{fg}(r)) \cdot C_{bg}(r), \quad (6)$$

where the foreground color is the volumetric rendering output of:

$$C_{fg}(r) = \Sigma_{i=1}^{2n} \tau_i c_i. \quad (7)$$

Here $\tau_i = exp(-\Sigma_{j<i}\sigma_j\delta_j)(1 - exp(-\sigma_i\delta_i))$ and δ_i is the distance between i^{th} and $(i+1)^{th}$ sample points. Similarly, the foreground mask, is given as:

$$M_{fg}(r) = \Sigma_{i=1}^{2n} \tau_i. \quad (8)$$

Individually, mask for hand along a ray r can be obtained using Hand NeRF in Sec. 4.1.2 as $M_{hand}(r)$. Similarly, masks for object (as shown in Fig. 2(b)) along a ray r can be obtained using Object NeRF in Sec. 4.1.1 as $M_{obj}(r)$. Whereas, the background mask can be obtained using the foreground mask as $M_{bg}(r) = 1 - M_{fg}(r)$. Finally, along a ray r, the one-hot encoding of hand mask $M_{hand}(r)$, object

mask $M_{obj}(r)$ and background mask $M_{bg}(r)$ gives us the predicted segmentation $M(r)$:

$$M(r) = \arg max\{M_{hand}(r), M_{obj}(r), M_{bg}(r)\}. \quad (9)$$

For stage 2, we also need visible object mask by following similar methodology as Eqs. 7, 8. We can set the color prediction of the sample points for object to 1, and color predictions for hand as 0. Along a ray r, this allows only the rays from the visible object regions to transmit, and stops the transmittance from the object surface points occluded by hands, giving us visible object mask M_{ho}, as shown in Fig. 2(c).

4.1.4 Pose Refinement

In this stage, we optimize both the hand and object pose in the camera-coordinate system, leveraging the composite rendering methodology. The initial hand and object poses are estimated. However, to achieve accurate alignment between the rendered and observed images, further optimization is performed to refine these pose estimates. The optimization process minimizes the discrepancy between the rendered and observed images using a differentiable renderer. Specifically, we iteratively adjust the hand and object poses, along with training compositional NeRF, by minimizing a loss function. RGB values of the predicted image are consistent with the ground truth image through the following loss function:

$$\mathcal{L}_{rgb1} = \Sigma_r ||C(r) - \hat{C}(r)||, \quad (10)$$

where $C(r)$ and $\hat{C}(r)$ are the predicted and groundtruth RGB colors along the ray r. Hand, object, and background masks are ensured consistent with ground truth by calculating the loss between predicted segmentation one-hot label

$M(r)$, calculated using Eq. 9, and estimated ground truth segmentation obtained using SAM-Track [11] $\hat{M}(r)$, that takes image frames as input, and user input for target object in the image for the first frame, and outputs segmentation masks tracking the segmented object, following the same methodology as HOLD [17]:

$$\mathcal{L}_{seg1} = \Sigma_r ||M(r) - \hat{M}(r)||. \tag{11}$$

Eikonal loss $\mathcal{L}_{eikonal1}$ regularizes the canonical geometry, and $\mathcal{L}_{hand-sdf1}$ calculates the loss between predicted density and ground truth MANO model. Finally, a contact loss $\mathcal{L}_{contact1}$ encourages hand's fingertip vertices V_{tip} to be close to object's vertices V_o, obtained via implicit networks:

$$\mathcal{L}_{contact1} = \Sigma_i \min_j ||V_{tip}^i - V_o^j||. \tag{12}$$

The final loss that optimizes all the NeRF parameters, along with object and hand poses is defined as:

$$\mathcal{L}_{s1} = \mathcal{L}_{rgb1} + \lambda_{seg1}\mathcal{L}_{seg1} + \lambda_{eikonal1}\mathcal{L}_{eikonal1} + \\ \lambda_{hand-sdf1}\mathcal{L}_{hand-sdf1} + \lambda_{contact1}\mathcal{L}_{contact1}, \tag{13}$$

where λ_{seg1} is set to 1.1 and progressively decreased to 0.1 until 30,000 iterations, $\lambda_{eikonal1}$ is 1, $\lambda_{hand-sdf1}$ is set to 5.0 and $\lambda_{contact1}$ is initially set to 0 and progressively increased to 1.0 until 30,000 iterations. For a given frame
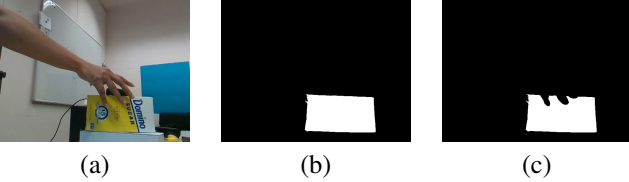

(a)　　　　　　(b)　　　　　　(c)

Figure 2. After Stage 1, for each (a) input image, predicted (b) object mask M_{obj} and (c) hand-object mask M_{ho} are calculated.

(Fig. 2 (a)), this stage prepares the object's implicit network, hand and object pose, object mask M_{obj} (Fig. 2 (b)) and hand-object mask M_{ho} (Fig. 2 (c)) for the second stage of our method.

4.2. Stage 2: Physics-based Inverse Rendering using Spherical Gaussians

In the second stage, we adopt a surface-based rendering approach to fine-tune the object's texture. The surface rendering-based approach allows for optimizing surface-level material properties like roughness, reflectance, Fresnel, shadows, etc., as will be discussed in this section.

This stage utilizes the refined masks and poses of the hand and object from Stage 1, as well as the original image sequences captured from a monocular camera. The output includes high-fidelity geometry, albedo, specular maps, and hand visibility, in conjunction with environment illumination based on the object's learned material properties.

By incorporating specularity and hand visibility, we prevent environmental factors such as shadows and reflections from being baked into the object's texture, ensuring accurate albedo predictions. We will discuss the above-mentioned different elements of the Physics-Based Rendering (PBR) equation in this section.

4.2.1 Environment Illumination

We implement environmental illumination using a sum of 128 Spherical Gaussians (SGs), which approximates the environmental light interacting with the object at any given point. These SGs represent light coming from various directions in the scene. This approximation allows us to efficiently simulate how the dynamic object is lit in an environment, ensuring that the lighting conditions are captured accurately in the final render. Each SG models the incident light distribution, and we apply these SGs to estimate the environmental contribution to the light reflected from the object's surface. This step enhances the realism by simulating how light behaves in real-world settings. Mathematically, from Eqs. (1) and (2), the direct illumination can be represented as:

$$L_d(\omega_i) = \Sigma_{j=1}^{128} G(\omega_i; \psi_j, \mu_j, \eta_j). \tag{14}$$

Since the captured object is dynamic, in its object coordinate system the environment rotates in the counter direction of the object's transformation. Therefore, according to the input object pose, the lobe directions ψ_j are rotated in the counter direction of the object's pose transform.

4.2.2 Bidirectional Radiance Distribution Field

Bidirectional Radiance Distribution Field (BRDF) on the surface point depends on diffuse albedo and specular effect. Diffuse BRDF is calculated using a Multi-layer Perceptron (MLP) predicting albedo color at the surface point $a(x)$, which defines the base color of the surface point x. Specular BRDF depends on material surface properties, like MLP-predicted roughness $r(x)$ and specular reflectance s. The roughness parameter controls how sharp or spread out the reflections are, and the specular reflectance parameter defines the metallic property of the object. Hence, the BRDF is the sum of diffuse BRDF ϕ_d and specular BRDF ϕ_s:

$$\phi(x; \omega_o, \omega_i) = \phi_d(x) + \phi_s(x; \omega_o, \omega_i). \tag{15}$$

The diffuse BRDF depends on predicted albedo color $a(x)$. The specular BRDF depends on Fresnel F and self-shadowing g terms, which are dependent on predicted specular reflectance s and roughness $r(x)$.

$$\phi_d(x) = \frac{a(x)}{\pi}, \phi_s(x; \omega_o, \omega_i) = \frac{F(s, r(x))g(r(x))}{4(\omega_o \cdot n)(\omega_i \cdot n)}, \tag{16}$$

where n is the surface normal on the point x.

4.2.3 Hand Occlusion

One of the key challenges in rendering hand-object interactions is handling occlusion caused by the hand. To model this, we represent the hand as a set of 108 parameterized spheres similar to power cells [49]. We first place all spheres $\{p_i, r_i\}_{i=1}^{108}$ manually in the canonical MANO hand volume Ψ, with p_i and r_i being their centers and radii.
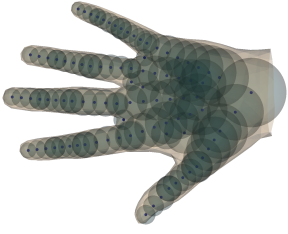

Figure 3. Canonical MANO hand is packed with 108 parameterizable spheres for hand-occlusion computation.

Initially, the sphere centers $\{p_i\}_{i=1}^{108}$ are determined on the canonical hand skeleton, manually selected between the MANO joints. The sphere radii $\{r_i\}_{i=1}^{108}$ requires partitioning the surface of the hand model into individual spheres. These spheres define a power diagram that partitions the MANO hand volume Ψ into a set of power cells. Each power cell Ψ_i^{pow} consists of the points x that are closest to a particular sphere (c_i, r_i):

$$\Psi_i^{pow} : \{x \in \Psi | d_{pow}(x, c_i, r_i) \leq d_{pow}(x, c_j, r_j), j \neq i\}, \tag{17}$$

where $d_{pow}(x, p_i, r_i) = ||x - p_i||^2 - r_i^2$ is the power distance between the point x and the sphere (p_i, r_i). The power diagram partitions all vertices $\{v_j\}$ of canonical MANO hand model into each individual set $\mathcal{V}_i = \{v_j | v_j \in \Psi_i^{pow}\}$. Whenever the MANO hand model is deformed by its pose and shape parameters, the center $\hat{p}_i$ and radius $\hat{r}_i$ of each sphere can be updated correspondingly by the following rule:

$$
\begin{aligned}
\hat{p}_i &= \frac{1}{|\mathcal{V}_i|} \sum_{v_j \in \mathcal{V}_i} \hat{v}_j, \\
\hat{r}_i &= \frac{1}{|\mathcal{V}_i|} \sum_{v_j \in \mathcal{V}_i} ||\hat{n}_j \cdot (\hat{v}_j - \hat{p}_i)||,
\end{aligned} \tag{18}
$$

where $\hat{v}_j$ and $\hat{n}_j$ are the vertex position and normal associated with vertex v_j in its deformed pose and shape. In this way, the centers and radii of these 108 spheres are parameterized by the MANO hand shape and pose.

This granular representation allows us to model how different parts of the hand obstruct the light and cast shadows on the object's surface. For each surface point on the object, we divide its lobe of Spherical Gaussian (SG) into a grid of 64x64 patches, each representing a small region of the environment illumination and BRDF. Each one of the 108 spheres representing MANO hand is projected onto the SG lobe, and the patches belonging to the occluded regions

are collected. Finally, the fractional value of the integral of the occluded patches gives a measure of the extent to which each part of the hand blocks light from reaching the object. For the following integral of a generic Spherical Gaussian (ISG) in polar format [66] with lobe sharpness η,

$$S(\theta, \phi, \eta) = \int_0^\theta \int_0^\phi e^{\eta(\cos \phi' - 1)} d\phi d\theta, \tag{19}$$

where ϕ' is angle between the lobe direction of the Spherical Gaussian, and some direction ω_i along the hemisphere. $S(\theta, \phi, \eta)$ represents the accumulated Spherical Gaussian over a patch of the hemisphere, where θ and ϕ are angular coordinates. This means when $\theta = \pi$ and $\phi = \pi$, it covers the whole hemisphere. To compute the fractional value for a given angular region, we normalize by dividing the integral result by $S(\pi, \pi, \eta)$ [23], as shown below:

$$\hat{S}(\theta, \phi, \eta) = \frac{S(\theta, \phi, \eta)}{S(\pi, \pi, \eta)}. \tag{20}$$

Iwasaki et. al. [23] call this normalized term as normalized ISG, and approximate it as follows:

$$\hat{S}(\theta, \phi, \eta) \approx \frac{1}{1 + e^{-g(\eta)(\theta - \pi/2)}} \frac{1}{1 + e^{-h(\eta)(\phi - \pi/2)}}, \tag{21}$$

where they define g and h as fourth-degree polynomial function approximations over η as follows:

$$
\begin{aligned}
g(\eta) &= g_4(\eta/100)^4 + g_3(\eta/100)^3 + g_2(\eta/100)^2 + g_1(\eta/100) \\
h(\eta) &= h_4(\eta/100)^4 + h_3(\eta/100)^3 + h_2(\eta/100)^2 + h_1(\eta/100),
\end{aligned} \tag{22}
$$

where g_4, g_3, g_2, g_1, h_4, h_3, h_2, h_1 are constant values $-2.6856e^{-6}$, $7e^{-4}$, -0.0571, 3.9529, 17.6028, $-2.6875e^{-6}$, $7e^{-4}$, -0.0592, 3.9900, 17.5003 respectively. The patches $\tilde{\Omega}$ occluded by hand spheres can be seen in Fig. 6. The normalized value helps us calculate the proportion of the spherical Gaussian that falls within a specific patch of the sphere. From Eq. 21 and Eq. 22, for a patch defined along some direction ω_i by the angular coordinates (θ_0, ϕ_0), (θ_1, ϕ_0), (θ_0, ϕ_1), and (θ_1, ϕ_1) (See Fig. 5), the final output is determined by computing the contributions from the patch corners:

$$
\begin{aligned}
\mathcal{F}(x, \theta_0, \theta_1, \phi_0, \phi_1) = \hat{S}(\theta_1, \phi_1, \eta) - \hat{S}(\theta_1, \phi_0, \eta) \\
- \hat{S}(\theta_0, \phi_1, \eta) + \hat{S}(\theta_0, \phi_0, \eta).
\end{aligned} \tag{23}
$$

The sum over all the occluded patches gives us the fractional value $\mathcal{F}(x)$, and is the fraction of ISG over occluded patches $\tilde{\Omega}$. Therefore, we can approximate the integration of a partially-occluded Spherical Gaussian $G(\omega_i)O(x, \omega_i)$, with respect to the direction ω_i over the hemisphere Ω^+ centered around the surface point x, using the above integral

Figure 4. Physics-based rendering calculates an integral over a hemispherical region, centered around the surface normal of a surface point, to calculate the final color of the surface point. Based on SG approximation, direct illumination is defined as a sum of 128 SGs, hand occlusion is calculated based on the parameterizable spherical representation of MANO hand, indirect illumination from occluding hand is a learned parameter, albedo represents the base color of the object without hand shadows or environment reflections, and specularity of the object is calculated based on the material properties - roughness, specular reflectance, etc.

over the occluded patches $\tilde{\Omega}$ and its approximation $\mathcal{F}(x)$ as follows:

$$\int_{\Omega^+} G(\omega_i) O(x, \omega_i) d\omega_i = \int_{\tilde{\Omega}} G(\omega_i) d\omega_i \qquad (24)$$
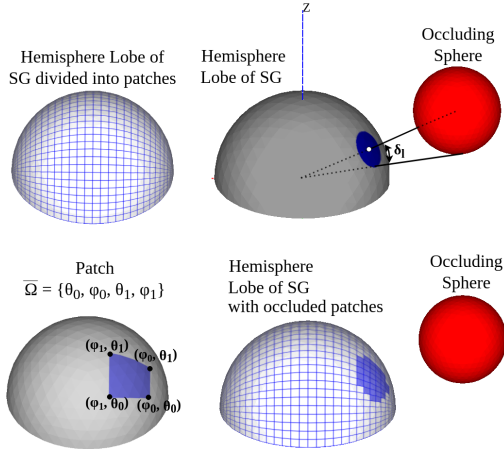$$\approx \mathcal{F}(x) * \int_{\Omega^+} G(\omega_i) d\omega_i.$$

Figure 5. An SG hemispherical lobe, centered around the Z-axis, is divided into 64x64 patches. An occluding sphere projected onto the SG lobe covers some patches. The occluded patches are calculated, and their fractional value is calculated to get hand occlusion.

4.2.4 Final Illumination

We discussed direct illumination $L_d(\omega_i)$ from environment, as a sum of Spherical Gaussian along direction w_i of hemisphere in Eq. 14, and integral of SG with hand occlusion in Eq. 24. In the regions of hand occlusion, the hand casts its own indirect illumination, which can be perceived as the hand's reflection on the object's surface. Hence, we learn
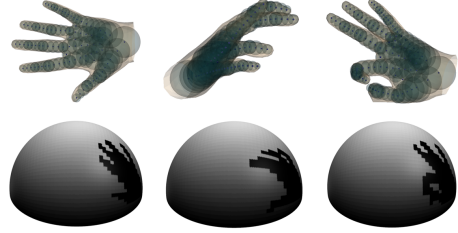

Figure 6. The parameterizable spheres are updated with hand pose transformation, and occlude the patches of Spherical Gaussian.

a constant indirect illumination L_i, starting from a generic skin color ($\#E0AC69$) from an online search. The final illumination $L(\omega_i, x)$ from Eq. 1 can thus be substituted as follows:

$$L(\omega_i, x) = L_d(\omega_i) * (1 - O(x, \omega_i)) + L_i * O(x, \omega_i). \quad (25)$$

Note that this equation also denotes a sum of 128 SGs. To use the cosine term with other SG terms in PBR Eq. (1), we approximate the cosine term $\omega_i \cdot n$, for incident direction ω_i at a surface point x with surface normal n:

$$\omega_i \cdot n \approx G(\omega_i; 0.0315, 32.7080, n) - 31.7003. \quad (26)$$

Combining illumination $L(\omega_i, x)$ from Eq. (25), BRDF $\phi(x; \omega_o, \omega_i)$ from Eq. (15), and cosine term $\omega_i \cdot n$ from Eq. (26), we get the final color $c(\omega_o, x)$ at the surface point x with view direction ω_o as the integral equation given in PBR Eq. (1). To maintain the consistency of object shape, and avoid the hand's involvement in object geometry and texture reconstruction, we propose the following loss functions. An RGB loss is calculated on the color c^{ho} predicted on N_{ho} sampled pixels in the intersection of the predicted object mask and groundtruth hand-object mask M_{ho}

(Fig. 2(c)):

$$\mathcal{L}_{rgb2} = \frac{1}{N_{ho}}\Sigma_{i=1}^{N_{ho}}||c_i^{ho} - \hat{c}_i||, \qquad (27)$$

where $\hat{c}_i$ is groundtruth color of the N_{ho} sampled pixels. The below mask loss calculates minimum SDF S^{no} on the N_{no} rays corresponding to the sampled pixels outside the intersection of the predicted object mask and ground truth object mask M_{obj} (Fig. 2(b)):

$$\mathcal{L}_{mask2} = \frac{1}{N_{no}}\Sigma_{j=1}^{N_{no}}\frac{ln(1 + e^{-50*S_j^{no}})}{50}. \qquad (28)$$

Just like in stage 1, $\mathcal{L}_{eikonal2}$ regularizes object's geometry. The final loss optimized is as follows:

$$\mathcal{L}_{s2} = \mathcal{L}_{rgb2} + \lambda_{mask2}\mathcal{L}_{mask2} + \lambda_{eikonal2}\mathcal{L}_{eikonal2}, \quad (29)$$

where λ_{mask2} is set to 100.0 and $\lambda_{eikonal2}$ is 0.1.

4.3. Implementation Details

We utilize monocular camera capture datasets that encompass all viewpoints around the object, ensuring comprehensive coverage for high-fidelity texture and geometry learning. For the first stage, the 3D annotations estimated include hand-object pixel-wise segmentation [11, 28, 61, 60, ?, 19], hand MANO pose estimation [31], and object pose estimation [39, 40] in the camera-coordinate system.

The first stage of our pipeline involves the composite volumetric rendering of the hand, object, and background, following a pipeline similar to HOLD [17], using the initial estimates of the hand and object poses, along with segmentation masks. HOLD optimizes hand and object poses, geometry, and an initial estimate of texture.

Using the optimized poses and object geometry from Stage 1, we derive the object mask and hand-object segmentation mask. These are then used in the second stage, which focuses on physics-based surface rendering of the object, following a pipeline inspired by PhySG [66]. A key modification in our method is the computation of occlusion to account for hand-object interaction, based on the approach in Iwasaki et al. [23]. Prior works, such as PhySG, do not compute occlusion in this way, setting our method apart. In Stage 2, we optimize the following key parameters at a surface point x:

- Roughness $r(x)$ and albedo $a(x)$, predicted using an MLP layer, which contribute to the specular BRDF $\phi_s(x)$ and diffuse BRDF $\phi_d(x)$ calculations, respectively.

- Spherical Gaussian (SG) parameters related to direct illumination $L_d(\omega_i)$ are optimized to accurately model light interactions with the object's surface.

- Additionally, an indirect illumination term L_i, learned as an RGB value, accounts for the effects of hand shadows and reflections on the object.

It is important to note that Stage 2 is dependent on Stage 1, as it requires the hand-object pose and segmentation data obtained in the first stage. Therefore, Stage 2 is not object agnostic. However, Stage 1 is object agnostic, enabling the reconstruction of 3D geometry and texture for objects with unknown shapes and textures. Each scenario is trained on a 12 GB Nvidia GeForce RTX 2080 Ti GPU, and training takes approximately 14 hours for each stage.

5. Experiments

5.1. Dataset

Our data consists of a sequence of the right hand of a subject interacting with a dynamic rigid object. To evaluate our approach, we use two types of datasets:

- **HO3D**: The training data of HO3D [21] dataset contains 9 objects from YCB [56] models. We use monocular frames of six different objects from the dataset. These six objects were specifically selected based on the presence of sufficient texture details, which is essential for evaluating the effectiveness of our texture reconstruction. The six objects (and their subjects) are - sugar box ("ShSu"), cracker box ("MC"), bleach cleanser ("ABF"), meat can ("GPMF"), mustard bottle ("SM"), and drilling machine ("MDF"). The 3 left objects in training data - scissors, banana, and mug contain no detailed-texture information for comparison.

- **In-the-wild**: We also use monocular frame sequences of three objects captured in real-world settings ("in-the-wild") [17]. These objects provide further diversity in testing by presenting challenging hand-object interactions and varied environmental conditions. The objects are - bottle, kettle, and rubrics cube.

5.2. Baselines

We use HOLD [17] as our primary baseline for performance comparison due to its proven effectiveness in dynamic hand-object interaction tasks, where it outperforms methods such as DiffHOI [64] and BundleSDF [52]. This makes HOLD an appropriate benchmark for assessing the improvements introduced by our method. Our approach, however, goes beyond what HOLD achieves by focusing on texture reconstruction and illumination effects, areas that HOLD does not fully address. To highlight the improvements, we compare our full pipeline results with HOLD alone. This comparison demonstrates how our method enhances texture accuracy, particularly in capturing the complex interactions between hands and objects, such as shadows, reflections, and visibility changes, which are often
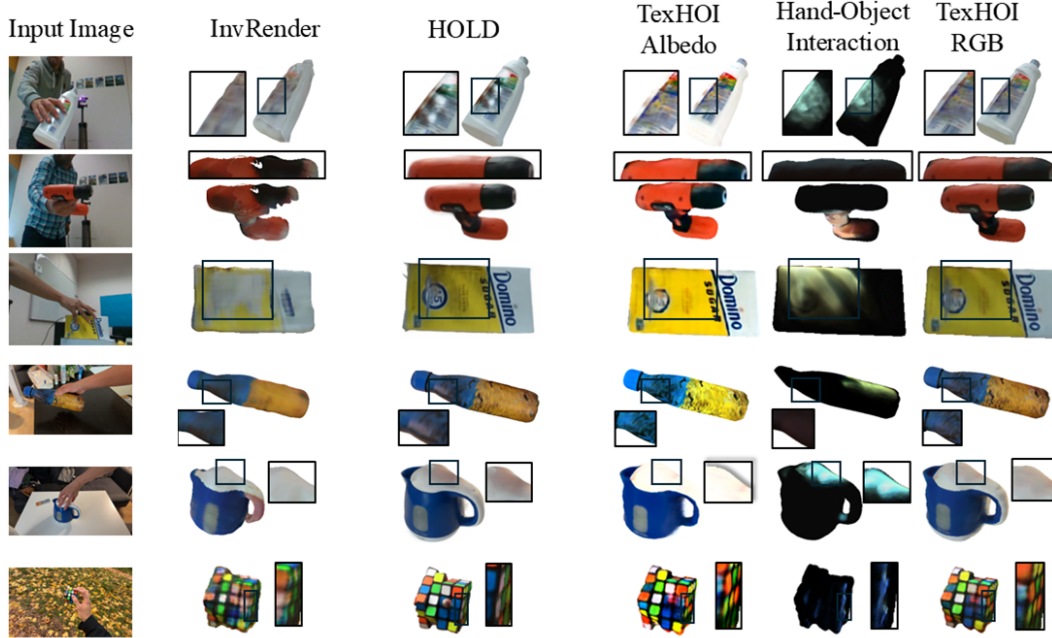
Figure 7. Qualitative Comparison: TexHOI albedo $a(x)$ predictions avoid baked hand-occlusions and environment-illumination. Hand occlusions can be observed in "Hand-Object Interaction" $\mathcal{F}(x)$ (0-1 normalized). The final TexHOI RGB $c(\omega_o, x)$ is calculated using Physics-Based Rendering, and is compared with other state-of-the-art approaches, HOLD [17] and InvRender [68].

	InvRender			HOLD			TexHOI		
HO3D Object	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Sugar Box	22.36	0.9501	0.110	22.20	0.9508	0.091	**23.55**	**0.9543**	**0.067**
Bleach Cleanser	18.76	0.9344	0.173	19.62	0.9383	0.109	**20.78**	**0.9454**	**0.105**
Drilling Machine	14.51	0.9097	0.222	**18.12**	0.9258	0.152	17.64	**0.9259**	**0.139**
Cracker Box	17.64	0.8910	0.197	19.52	0.8919	0.095	**19.88**	**0.8972**	**0.090**
Mustard Bottle	21.86	0.9476	0.138	**22.08**	0.9516	**0.103**	21.93	**0.9550**	0.110
Meat Can	20.26	0.9491	0.085	**22.16**	0.9543	**0.057**	22.02	**0.9547**	0.060
Average	19.23	0.9301	0.154	20.61	0.9354	0.101	**20.96**	**0.9387**	**0.095**

Table 1. Quantitative comparison is performed between our method, TexHOI, and other state-of-the-art approaches that reconstruct geometry and texture. HOLD bakes hand-occlusions and illumination on its texture prediction, InvRender does not handle dynamic occlusions.

overlooked in prior works. Additionally, for comparison with InvRender [68], we employ the 3D reconstruction from our Stage 1 and modify InvRender to incorporate object segmentation masks for computing pixel-wise losses in hand-object interaction images. We also adapt the approach by introducing Spherical Gaussian (SG) lobe rotations to effectively handle dynamic rigid object rotations, which are essential for handling both static and dynamic elements in the scene.

5.3. Evaluation Metrics

Since the objects manipulated in HO3D [21] dataset have their ground truth 3D models and textures in YCB [56], we use them for our quantitative evaluation. For each video sequence in HO3D, we subsample one frame out of every 10 frames, resulting in around 200-400 frames per sequence as our experimental data. We use Mitsuba [1] renderer and render the ground truth 3D models and textures to generate ground truth images corresponding to the poses of those sampled image frames, to compare our TexHOI's predicted albedo with other baseline methods. The quantitative evaluation metrics used are as follows:

- **Peak Signal-to-Noise Ratio (PSNR)**: This metric is used to evaluate the quality of the reconstructed texture by measuring how much signal (texture detail) is present compared to noise (artifacts or blurring).

- **Structural Similarity Index Measure (SSIM)**: We use SSIM to assess the visual similarity between our reconstructed textures and the ground truth. This metric helps to measure the perceptual quality of textures.

HO3D Object	w/o ind-illum			w/o occ			full		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Sugar Box	22.69	**0.9548**	0.113	23.54	0.9510	0.107	**23.55**	0.9543	**0.067**
Bleach Cleanser	20.45	0.9416	0.116	20.16	0.9382	0.118	**20.78**	**0.9454**	**0.105**
Drilling Machine	17.49	**0.9263**	0.146	17.58	0.9248	0.145	**17.64**	0.9259	**0.139**
Cracker Box	20.17	0.8971	0.124	**20.22**	0.8906	0.123	19.88	**0.8972**	**0.090**
Mustard Bottle	22.54	0.9544	**0.108**	21.77	0.9527	0.118	**21.93**	**0.9550**	0.110
Meat Can	21.91	0.9541	0.067	21.83	0.9538	0.068	**22.02**	**0.9547**	**0.060**
Average	20.87	0.9380	0.112	20.85	0.9351	0.113	**20.96**	**0.9387**	**0.095**

Table 2. Quantitative ablative comparison of our model with the versions without indirect illumination, and without hand occlusions. Without illumination, the interacting hand casts a darker texture. Without occlusion handling, artifacts occur in major hand grasp regions.
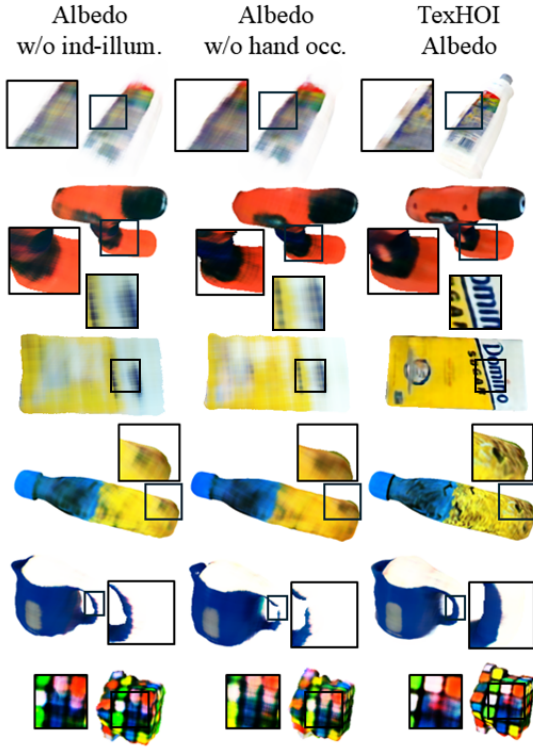

Figure 8. Ablation studies compare our model with the versions trained without indirect illumination, and without any occlusions. Illumination effects on surface albedo, and hand occlusions on surface can be observed. The scenes are in same order as Fig. 7

- **Learned Perceptual Image Patch Similarity (LPIPS)**: LPIPS is used to measure the perceptual distance between the predicted texture and the reference, focusing on perceptual similarity as seen by human vision. Lower LPIPS scores indicate higher perceptual quality.

5.4. Baseline Comparisons

From Fig. 7, it is evident that while HOLD produces final images that closely resemble the input image, it achieves this by baking in the hand shadows and environmental re-
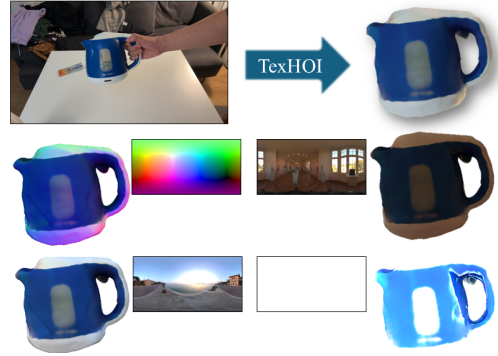

Figure 9. For an object reconstructed using our method, TexHOI, it can be relighted in different environments, based on different environment illumination.

flections directly into the object's texture. This, in turn, reduces the model's ability to generalize across different lighting environments and hand poses. In contrast, our method, TexHOI, can disentangle the object's intrinsic texture (albedo) from external environmental influences, resulting in a cleaner and more accurate reconstruction of the object's surface. This observation is further supported by the evaluation metrics in Tab. 1. HOLD achieves better performance on metrics such as RGB similarity because it embeds hand shadows and environmental reflections into the predicted texture, making them appear closer to the ground truth in terms of pixel-wise accuracy. However, our method excels in producing albedo textures that are more accurate representations of the object's inherent properties, without conflating them with transient environmental elements.

5.5. Ablation Studies

To evaluate the importance of different components in our approach, we conduct ablation studies by removing specific features, such as indirect illumination and hand occlusion handling. Specifically, we compare the results of our full methodology against versions without indirect illumination modeling and without explicit hand occlusion consideration. Our results, from Fig. 8 and Tab. 2 indicate that indirect illumination significantly contributes to the realism

of the reconstructed texture, particularly in complex scenes where light bounces off the occluding hand surface. Without this component, the reconstructed textures appear flatter and lack depth. Similarly, ignoring hand occlusions results in visual artifacts, as the shadowing and occlusion effects caused by interacting hands are not properly accounted for, leading to inconsistencies in the texture and geometry. These experiments highlight the necessity of modeling both indirect illumination and occlusions to achieve high-quality, realistic reconstructions. The few examples where the version without indirect illumination surpasses our full model in Tab. 2 are again because the final prediction bakes the hand-environment interactions in the final prediction.
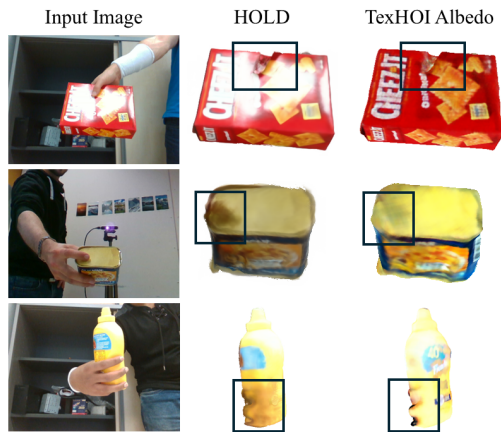

Figure 10. Our method is limited in geometry and texture reconstruction in cases when the hand is static/fixed w.r.t. object.

5.6. Limitations

From Fig. 10, we observe that while our method is effective in generating high-detail geometry and textures, and successfully reconstructs albedo without baked-in environmental influences, it struggles in scenarios where the hand remains static relative to the object. In such cases, if the hand is not dynamic but remains fixed throughout the dataset, the impression of the hand tends to become baked into the geometry and texture of the object. This occurs because the model relies on changes in perspective and movement to distinguish between the object and interacting elements, and a lack of variability leads to ambiguity in the learned representation.

6. Conclusion and Future Work

In this research, we demonstrate that TexHOI significantly outperforms existing state-of-the-art methods in reconstructing high-detail albedo textures free from environmental artifacts, particularly in challenging scenarios involving hand occlusions and environmental reflections. We achieve this by parameterizing occluding spheres, fitted into the MANO hand model, and calculating the fraction of occluded Spherical Gaussians to accurately represent hand occlusions. Furthermore, in many state-of-the-art methods, environmental illumination often reflects on the texture predictions, leading to inconsistencies when the object is viewed under different lighting conditions. Our approach tackles this problem by incorporating Physics-Based Rendering techniques, approximating illumination effects using Spherical Gaussians. This allows us to produce textures that are consistent across varying lighting environments, enhancing the robustness of the reconstructed model.

Despite these advancements, there are several areas for future improvement. One potential direction is incorporating a more sophisticated model for indirect illumination, particularly one that accounts for the specific properties of skin and other materials under different lighting conditions, could enhance the realism of the reconstructions. Although the centers of the hand spheres remain outside the object's surface, there can be inter-penetrations of the spheres with object near the fingertips. This causes negligible errors in the computation of hand occlusion, and can be an area of improvement in the future. Another promising direction is exploring temporal consistency across frames in dynamic sequences, avoiding minor temporal artifacts. By incorporating temporal coherence into the reconstruction process, we could achieve smoother transitions and more stable textures in video sequences. 3D Gaussian splat-based inverse rendering have not been explored in the area of hand-object interactions, according to our knowledge, and can be a promising direction of progress in the research. In the future, the work can also be optimized end-to-end using composite surface-based inverse rendering or a more efficient 3D Gaussian splat-based inverse rendering, eliminating the need for a 2-step process to correct poses and optimize texture, as all elements can be optimized simultaneously. Finally, extending our model to handle a broad range of interaction scenarios, such as multiple hands, or handling inter-penetrations of hand and object using collision detection or physical simulation modules, would make it more applicable to real-world use cases with complex interactions.

References

[1] Mitsuba3 Renderer. Available online: https://mitsuba-renderer.org/ (accessed on ???). 10

[2] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch. NeRD: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12684–12694, 2021. 3

[3] M. Boss, A. Engelhardt, A. Kar, Y. Li, D. Sun, J. Barron, H. Lensch, and V. Jampani. SAMURAI: Shape and material from unconstrained real-world arbitrary image collections. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 26389–26403, 2022. 3

[4] M. Boss, V. Jampani, R. Braun, C. Liu, J. Barron, and H. Lensch. Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 10691–10704, 2021. 3

[5] S. Brahmbhatt, C.-H. Tang, C. D. Twigg, C. C. Kemp, and J. Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 361–378. Springer, 2020. 1

[6] G. Cai, F. Luan, M. Hašan, K. Zhang, S. Bi, Z. Xu, I. Georgiev, and S. Zhao. PBIR-NIE: Glossy object capture under non-distant lighting. arXiv preprint arXiv:2408.06878, 2024. 3

[7] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, et al. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9044–9053, 2021. 1

[8] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. TensoRF: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 333–350. Springer, 2022. 3

[9] X. Chen, S. Peng, D. Yang, Y. Liu, B. Pan, C. Lv, and X. Zhou. IntrinsicAnything: Learning diffusion priors for inverse rendering under unknown illumination. arXiv preprint arXiv:2404.11593, 2024. 3

[10] Z. Chen, C. Ding, J. Guo, D. Wang, Y. Li, X. Xiao, W. Wu, and L. Song. L-tracing: Fast light visibility estimation on neural surfaces by sphere tracing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–233. Springer, 2022. 3

[11] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. Segment and track anything. arXiv preprint arXiv:2305.06558, 2023. 6, 9

[12] H. Choi, N. Chavan-Dafle, J. Yuan, V. Isler, and H. w. Park. HandNeRF: Learning to reconstruct hand-object interaction scene from a single rgb image. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 13940–13946. IEEE, 2024. 3

[13] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 303–312, 1996. 2

[14] Y. Dai, Q. Wang, J. Zhu, D. Xi, Y. Huo, C. Qian, and Y. He. MIRReS: Multi-bounce inverse rendering using reservoir sampling. arXiv preprint arXiv:2406.16360, 2024. 3

[15] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1755–1768. PMLR, 2023. 3

[16] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 173–182, 1995. 2

[17] Z. Fan, M. Parelli, M. E. Kadoglou, X. Chen, M. Kocabas, M. J. Black, and O. Hilliges. HOLD: Category-agnostic 3d reconstruction of interacting hands and objects from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 494–504, 2024. 1, 3, 6, 9, 10

[18] J. Gao, C. Gu, Y. Lin, H. Zhu, X. Cao, L. Zhang, and Y. Yao. Relightable 3D Gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. arXiv preprint arXiv:2311.16043, 2023. 3

[19] Y. Gong, Y.-A. Chung, and J. Glass. AST: Audio spectrogram transformer. In *Proceedings of Interspeech*, pages 571–575, 2021. 9

[20] Z. Guo, W. Zhou, M. Wang, L. Li, and H. Li. HandNeRF: Neural radiance fields for animatable interacting hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21078–21087, 2023. 3

[21] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. HOnnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3196–3206, 2020. 1, 9, 10

[22] J. Hasselgren, N. Hofmann, and J. Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:22856–22869, 2022. 3

[23] K. Iwasaki, W. Furuya, Y. Dobashi, and T. Nishita. Real-time rendering of dynamic scenes under all-frequency lighting using integral spherical gaussian. *Computer Graphics Forum*, 31:727–734, 2012. 7, 9

[24] M. R. James and S. Robson. Straightforward reconstruction of 3d surfaces and topography with a camera: Accuracy and geoscience application. *Journal of Geophysical Research: Earth Surface*, 117(F3), 2012. 2

[25] H. Jin, I.-C. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su. TensoIR: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2023. 3

[26] J. K. Kawai, J. S. Painter, and M. F. Cohen. Radioptimization: Goal based rendering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 147–154, 1993. 2

[27] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):139:1–139:19, 2023. 3

[28] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. arXiv preprint arXiv:2304.02643, 2023. 9

[29] J. Knodt, J. Bartusek, S.-H. Baek, and F. Heide. Neural Ray-Tracing: Learning surfaces and reflectance for relighting and view synthesis. arXiv preprint arXiv:2104.13562, 2021. 3

[30] J. Li, L. Wang, L. Zhang, and B. Wang. TensoSDF: Roughness-aware tensorial representation for ro-

bust geometry and material reconstruction. arXiv preprint arXiv:2402.02771, 2024. 3

[31] K. Lin, L. Wang, and Z. Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1954–1963, 2021. 9

[32] Y. Liu, P. Wang, C. Lin, X. Long, J. Wang, L. Liu, T. Komura, and W. Wang. NeRO: Neural geometry and brdf reconstruction of reflective objects from multiview images. *ACM Transactions on Graphics (ToG)*, 42(4):1–22, 2023. 3

[33] S. R. Marschner. *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell University, 1998. 2

[34] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3, 4

[35] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 3

[36] J. S. Nimeroff, E. Simoncelli, and J. Dorsey. Efficient re-rendering of naturally illuminated environments. In *Photorealistic Rendering Techniques*, pages 373–388. Springer, 1995. 2

[37] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017. 2

[38] J. Romero, D. Tzionas, and M. J. Black. Embodied Hands: Modeling and capturing hands and bodies together. arXiv preprint arXiv:2201.02610, 2022. 4

[39] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 9

[40] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 9

[41] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7495–7504, 2021. 3

[42] H. Su, Q. Huang, N. J. Mitra, Y. Li, and L. Guibas. Estimating image depth using shape collections. *ACM Transactions on Graphics (ToG)*, 33(4):1–11, 2014. 2

[43] C. Sun, G. Cai, Z. Li, K. Yan, C. Zhang, C. Marshall, J.-B. Huang, S. Zhao, and Z. Dong. Neural-pbir reconstruction of shape, material, and illumination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18046–18056, 2023. 3

[44] C. Sun, M. Sun, and H.-T. Chen. Improved direct voxel grid optimization for radiance fields reconstruction. arXiv preprint arXiv:2206.05085, 2022. 3

[45] P. C. Teo, E. P. Simoncelli, and D. J. Heeger. Efficient linear re-rendering for interactive lighting design. Technical report, CiteSeer, 1997. 2

[46] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 289–298, 1991. 2

[47] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[48] H. Wang, W. Hu, L. Zhu, and R. W. Lau. Inverse rendering of glossy objects via the neural plenoptic function and radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19999–20008, 2024. 3

[49] N. Wang, B. Wang, W. Wang, and X. Guo. Computing medial axis transform with feature preservation via restricted power diagram. *ACM Transactions on Graphics (TOG)*, 41(6):1–18, 2022. 7

[50] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689, 2021. 3

[51] Y. Wang, W. Wu, and D. Xu. Learning unified decompositional and compositional NeRF for editable novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18247–18256, 2023. 3

[52] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield. BundleSDF: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–617, 2023. 1, 3, 9

[53] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012. 2

[54] A. Witkin and M. Kass. Reaction-diffusion textures. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 299–308, 1991. 2

[55] Q. Wu, X. Liu, Y. Chen, K. Li, C. Zheng, J. Cai, and J. Zheng. Object-compositional neural implicit surfaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 197–213. Springer, 2022. 3

[56] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199, 2017. 1, 9, 10

[57] Z. Yan, C. Li, and G. H. Lee. NeRF-DS: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8285–8295, 2023. 3

[58] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13779–13788, 2021. 3

[59] W. Yang, G. Chen, C. Chen, Z. Chen, and K.-Y. K. Wong. PS-NeRF: Neural inverse rendering for multi-view photometric stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 266–284. Springer, 2022. 3

[60] Z. Yang, Y. Wei, and Y. Yang. Associating objects with transformers for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 9

[61] Z. Yang and Y. Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 9

[62] Y. Yao, J. Zhang, J. Liu, Y. Qu, T. Fang, D. McKinnon, Y. Tsin, and L. Quan. NeILF: Neural incident light field for physically-based material estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 700–716. Springer, 2022. 3

[63] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 2492–2502, 2020. 3, 4

[64] Y. Ye, P. Hebbar, A. Gupta, and S. Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19717–19728, 2023. 1, 9

[65] K. Zhang, F. Luan, Z. Li, and N. Snavely. IRON: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5565–5574, 2022. 3

[66] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. 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 (CVPR)*, pages 5453–5462, 2021. 1, 3, 4, 7, 9

[67] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron. NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2, 3

[68] Y. Zhang, J. Sun, X. He, H. Fu, R. Jia, and X. Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18643–18652, 2022. 1, 2, 3, 10