Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/XXXX.2022.1234567

Shadow-Aware Inverse Rendering in Gaussian Splatting

Kotoka Matsunaga[1], Xuening Tian[2], Hideo Saito[1], Dieter Schmalstieg[2,3], and Shohei Mori[2,1]

[1]Keio University
[2]VISUS, University of Stuttgart
[3]Graz University of Technology

Corresponding author: Kotoka Matsunaga (email: kotoka@keio.jp).

This work was supported by the Alexander von Humboldt Foundation funded by the German Federal Ministry of Research, Technology and Space, and the German Research Foundation *DFG* under Germany's Excellence Strategy EXC 2120/2 (390831618).

ABSTRACT Inverse rendering decomposes a scene into geometry, materials, and lighting to enable physically consistent view synthesis and relighting. Inverse rendering approaches built upon Gaussian Splatting benefit from its efficient reconstruction pipeline, yet they struggle in the presence of strong shadows. Shadows are often absorbed into albedo, leading to view synthesis in which the original shadows remain, while newly computed shadows are simply overlaid during relighting. To address this issue, we present a shadow-aware inverse rendering method for 2D Gaussian Splatting that explicitly disentangles illumination effects from surface reflectance. Our approach estimates pixelwise directional visibility in image space from observed geometry, enabling accurate modeling of cast shadows. We further introduce material segmentation, which groups shadowed and non-shadowed pixels of the same surface into a single material, enforcing consistent reflectance and preventing shadows from being baked into the albedo. Our method produces shadow-free albedo and models shadows as directional attenuation, enabling stable and high-fidelity relighting under novel illumination.

INDEX TERMS Gaussian Splatting, Inverse rendering, Relighting

I. INTRODUCTION

Inverse rendering aims to decompose a scene into its physical components, such as geometry, materials, and lighting, from a captured image [1] to enable downstream tasks such as relighting [2], [3]. The observed colors result from the interaction of these components, described by the rendering equation.

3D Gaussian Splatting (3DGS) [4] represents scenes with a set of Gaussians that provide an explicit and effective basis for this task [5], [6]. These approaches incorporate physically based rendering (PBR) to model observed colors as interactions between lighting and surface materials. However, accurately disentangling lighting and material effects remains challenging [7]. Typically, shadows are absorbed into the albedo or indirect illumination, leading to physically inconsistent relighting results. This limitation has two likely causes—limited pixel-precise visibility formulations and material identification. In this paper, we propose using two cues directly observable from the input images: visibility (i.e., view-dependent occlusion) and material consistency. Previous work has used diffusion models [8] or multi-illumination data [9], [10] for these cues. We explicitly model these two cues from multi-view observations under one single illumination condition, because such images can be obtained rather easily.

To this end, we first introduce material segmentation to regularize regions with the same surface material. This discourages the optimizer from explaining shadowed pixels using albedo colors and encourages the attribution of the color variations to lighting. We then calculate pixelwise visibility from the recovered geometry in hemispherical sampling directions, allowing shadows to be modeled explicitly as directional light attenuation. Based on these cues, we propose an inverse rendering framework that separates geometry, material, light sources, and visibility according to their physical roles. This decomposition produces reusable albedo maps without baked shadows and enables stable relighting with coherent shadows.

Fig. 1. We present a shadow-aware inverse rendering method for Gaussian Splatting that separates illumination from material appearance and enables consistent relighting. Given an input image (left), our method removes the original shadows and generates new shadows consistent with the target lighting (Ours). In contrast, existing methods often fail to remove the original shadows, resulting in baked-in shadows and unnatural relighting results.

We demonstrate that the proposed method consistently outperforms prior methods in albedo quality, light estimation, shadow-region accuracy, and relighting performance (Fig. 1).

Our technical contributions can be summarized as follows:

- We propose an inverse rendering method for Gaussian Splatting that leverages material segmentation and pixelwise visibility to attribute shadowed colors to lighting rather than materials.
- We demonstrate physically consistent relighting that prevents shadow artifacts from being reintroduced into the scene.
- Our formulation yields high-quality albedo, lighting, and shadow separation, together with accurate and stable relighting results.

II. RELATED WORK

We review prior work on neural rendering and inverse rendering, and highlight that existing approaches have primarily focused on improving rendering quality.

A. NEURAL RENDERING AND NOVEL VIEW SYNTHESIS

Recently, neural rendering has emerged as a revolutionary paradigm in computer graphics. By steering away from explicit 3D representations and traditional rendering pipelines—such as rasterization and ray tracing—neural rendering utilizes continuous and differentiable neural networks to synthesize photorealistic images from learned implicit scene representations. The Neural Radiance Field (NeRF) [11] models a volumetric scene by querying a radiance field via a multi-layer perceptron (MLP). While groundbreaking, the original NeRF suffers from slow rendering speeds and limited generalization capabilities, prompting follow-up research [12]–[16] aimed at addressing these issues.

Recent advances have demonstrated that storing scene properties in explicit structures, such as voxels or point clouds [17], can significantly accelerate the volumetric rendering process. Building on this work, 3D Gaussian Splatting [4] introduced differentiable rasterization of Gaussians functions initialized from Multi-View Stereo (MVS) [18]. However, since 3DGS was designed primarily for novel-view synthesis, it inherently lacks explicit geometric constraints, often leading to degenerate surface geometry in unconstrained settings.

To address these limitations, recent work has proposed various geometric regularization strategies [19], [20]. A notable approach is 2D Gaussian Splatting [21], which replaces 3D ellipsoids with oriented elliptical disks, thereby enforcing consistent depth estimation and enabling the integration of surface normal constraints.

B. INVERSE RENDERING

Inverse rendering seeks to decompose captured images into intrinsic scene properties such as unknown geometry, materials, and light conditions to enable physically-based scene relighting and manipulation. Early approaches addressed this by optimizing over rigid geometric proxies or simplified light models [22]–[24]. Standard 3DGS implementations represent view-dependent color using Spherical Harmonics (SH). This dependence on SH inherently entangles lighting, material, and geometric properties within fixed coefficients, complicating the decomposition of the scene into its intrinsic components.

To mitigate this, recent literature has explored incorporating physically-based rendering (PBR) models directly into the splatting pipeline [5], [6]. These methods fall into two groups with respect to shadow modeling. The first group, represented by GS-IR [6], [25], [26], uses approximate occlusion models. These methods divide 3D space with a voxel grid and estimate the occlusion state of each Gaussian. During rendering, they mainly use the occlusion value along the surface normal direction to reproduce local shading effects similar to ambient occlusion. They are effective for contact shadows and local occlusion, but they do not model direction-dependent cast shadows between objects well. GI-GS improves this framework by introducing pixelwise visibility within a single view [27]. GIR further extends occlusion estimation with voxel grids to multiple directions [28]. However, these methods do not sufficiently remove existing shadows from albedo, which limits natural relighting after scene editing.

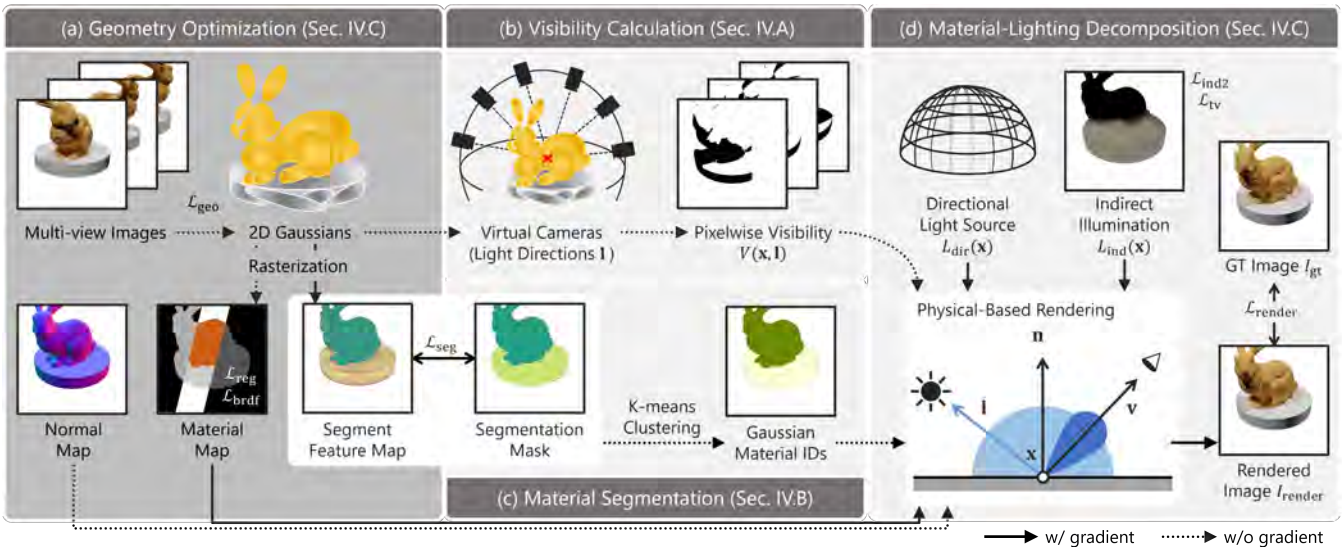

Fig. 2. Overview of the proposed method. (a) Our method first optimizes scene geometry using 2D Gaussian Splatting from multi-view images and rasterizes normal and material maps. (b) Based on the recovered geometry, it computes pixelwise directional visibility to model shadows as light attenuation. (c) It segments the scene into material-consistent regions, which serve as constraints in the subsequent decomposition. (d) Finally, it performs material and lighting decomposition using a physically-based model supervised by the ground-truth image.

The second group, represented by Relightable3DGaussian [5], [29], [30], uses explicit visibility models. These methods assign direction-dependent visibility to each Gaussian and evaluate incident light from each direction with ray tracing. This enables physically based relighting with cast shadows under known lighting conditions. However, during inverse rendering, shadows in the input images remain baked into the estimated albedo. When relit under new illumination, the original shadows may remain, while new shadows are added.

As a result, existing methods do not achieve both accurate shadow separation from albedo and physically consistent shadow recomputation after scene editing. Our method addresses both issues. We estimate visibility along light directions for observation-based shadow modeling. We introduce semantic regularization with material segmentation masks to preserve material consistency across Gaussians, leading to more robust and physically plausible decomposition.

III. PRELIMINARIES

A. 3D GAUSSIAN SPLATTING (3DGS)

3DGS [4] represents a scene as a set of 3D Gaussian primitives. Each Gaussian consists of its center $\mathbf{p}_i$ and covariance matrix $\mathbf{\Sigma}_i$: $G(\mathbf{p}) = \exp\left(-\frac{1}{2}(\mathbf{p}-\mathbf{p}_i)^\top \mathbf{\Sigma}^{-1}(\mathbf{p}-\mathbf{p}_i)\right)$. To render them, each Gaussian is transformed to camera coordinates using the world-to-camera matrix $\mathbf{W}$ and projected onto the image plane with a local affine approximation $\mathbf{J}$, yielding $\Sigma' = \mathbf{JW\Sigma W}^\top\mathbf{J}^\top$. For each pixel $\mathbf{x}$, the color $C(\mathbf{x})$ is computed by α-blending Gaussians sorted by depth: $C(\mathbf{x}) = \sum_{i\in N} T_i\,\alpha_i\,c_i$ where $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$. Here, N is the set of Gaussians, c_i denotes the color of the i-th Gaussian; α_i is its opacity, and T_i represents the accumulated transmittance along the ray.

B. 2D GAUSSIAN SPLATTING (2DGS)

Compared to 3DGS, 2DGS [21] uses planar Gaussian primitives to approximate surface geometry. Each primitive lies on a local tangent plane where the Gaussian is evaluated in local 2D coordinates (u, v) as $G(u) = \exp\left(-(u^2+v^2)/2\right)$. To render 2D Gaussians, each of them is projected onto the image plane by a homogeneous transformation, $\mathbf{x}' = \mathbf{WH}(u, v, 1, 1)^\top$, where $\mathbf{H} \in \mathbb{R}^{4\times4}$ defines the geometry of the 2D Gaussian. Here, $\mathbf{x}' = (xz, yz, z, z)^\top$ denotes a homogeneous ray that passes through pixel (x, y) and intersects the splat at depth z. For each pixel $\mathbf{x}$, the color is computed by α-blending Gaussians sorted by depth, as in 3DGS. We use 2DGS for our scene representation basis due to its accurate geometry.

C. PHYSICALLY-BASED RENDERING (PBR)

PBR models the outgoing radiance as the interaction between light and surface material. The outgoing radiance at a surface point $\mathbf{x}$ in viewing direction $\mathbf{v}$ is given by the rendering equation [31]:

$$L_o(\mathbf{x}, \mathbf{v}) = \int_\Omega L_{\text{in}}(\mathbf{x}, \mathbf{l}) f_r(\mathbf{l}, \mathbf{v}, \mathbf{x})(\mathbf{l}\cdot\mathbf{n})\,\mathrm{d}\mathbf{l}, \tag{1}$$

where $\mathbf{l}$, $\mathbf{v}$, and $\mathbf{n}$ denote the incident light direction, the viewing direction, and the surface normal, respectively. $L_{\text{in}}(\mathbf{x}, \mathbf{l})$ represents the incident radiance arriving at $\mathbf{x}$ from direction $\mathbf{l}$, and $L_o(\mathbf{x}, \mathbf{v})$ denotes the outgoing radiance in direction $\mathbf{v}$. Integration is taken over the upper hemisphere Ω centered on $\mathbf{x}$. The term $f_r(\mathbf{l}, \mathbf{v}, \mathbf{x})$ is the bidirectional reflectance distribution function (BRDF), which describes

how incoming light is reflected toward the viewing direction depending on the surface material.

We model surface reflectance using the Cook-Torrance BRDF model, which decomposes the BRDF into diffuse and specular components:

$$f_r(\mathbf{l},\mathbf{v},\mathbf{x}) = \underbrace{\frac{(1-m(\mathbf{x}))\,a(\mathbf{x})}{\pi}}_{\text{diffuse}} + \underbrace{\frac{D(\mathbf{l},\mathbf{v},\mathbf{x})\,F(\mathbf{l},\mathbf{v},\mathbf{x})\,G(\mathbf{l},\mathbf{v},\mathbf{x})}{4(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})}}_{\text{specular}}. \quad (2)$$

where D, F, and G denote the normal distribution function, Fresnel term, and geometry term, respectively, whose formulations depend on the roughness $\rho(\mathbf{x})$ and metallic $m(\mathbf{x})$ [32], [33]. Substituting this BRDF into the rendering equation yields:

$$L_o(\mathbf{x},\mathbf{v}) = L_d(\mathbf{x}) + L_s(\mathbf{x},\mathbf{v}), \quad (3)$$

where

$$L_d(\mathbf{x}) = \int_\Omega L_{\text{in}}(\mathbf{x},\mathbf{l})\,\frac{(1-m(\mathbf{x}))\,a(\mathbf{x})}{\pi}(\mathbf{l}\cdot\mathbf{n})\,\mathrm{d}\mathbf{l}, \quad (4)$$

$$L_s(\mathbf{x},\mathbf{v}) = \int_\Omega L_{\text{in}}(\mathbf{x},\mathbf{l})\,\frac{D(\mathbf{l},\mathbf{v},\mathbf{x})\,F(\mathbf{l},\mathbf{v},\mathbf{x})\,G(\mathbf{l},\mathbf{v},\mathbf{x})}{4(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})}(\mathbf{l}\cdot\mathbf{n})\,\mathrm{d}\mathbf{l}. \quad (5)$$

IV. METHOD

We propose a shadow-aware inverse rendering model for 2DGS that provides explicit reasoning for shadowed pixels, accounts for light sources and visibility, and avoids baked-in shadows (Fig. 2). To model the shadow caused by geometric occlusions in the lighting, we compute pixelwise visibility. Existing visibility-aware methods estimate directional visibility or a single value (i.e., occlusion) per voxel [6], [25]–[28], Gaussian [5], [30], or vertices [34], which potentially exploit limited spatial resolution and fail to capture local structures that produce clear shadows. Given 2D Gaussians of a scene (Fig. 2a), we instead compute pixelwise directional visibility (Fig. 2b).

Regardless of the granularity of the visibility models, existing methods tend to encode clear shadows into the albedo map. We speculate that encoding shadows into albedo can preserve the original appearance rather than attributing it to the geometry-lighting relationship. To keep the optimizer from taking the easy route, we enforce the same properties to be assigned to points having the same material (Fig. 2c).

Based on these observations, our method consists of two components. The first component estimates pixelwise directional visibility from the recovered 2D Gaussians to model shadows as light attenuation. The second component enforces material consistency through segmentation to prevent shadows from being absorbed into albedo. Together, these components form a visibility-aware inverse rendering approach (Fig. 2d).

A. PIXELWISE VISIBILITY CALCULATION

Existing visibility estimation approaches use voxel grids, Gaussians, or vertices as units to define visibility in their underlying representations. This spatial mismatch hinders accurate visibility in input views (Fig. 4). To close the gap, we implement pixelwise visibility inspired by conventional shadow mapping [35], and leverage this computation within our segment-wise material-consistency constraints to achieve shadow-aware inverse rendering in 2DGS. We thus formulate a pixelwise visibility term,

$$V(\mathbf{x},\mathbf{l}) = \mathbb{1}\left[\Delta(\mathbf{x},\mathbf{l}) \le \tau\right], \quad (6)$$

for each pixel $\mathbf{x}$ and light direction $\mathbf{l}$, where τ is a threshold that determines whether a point is considered visible.

We unproject a pixel $\mathbf{x}$ into 3D space and transform it into other views to determine whether it is visible. We calculate the centroid of all Gaussians and surround it with N_l virtual viewpoints in each direction $\mathbf{l}$ at the average distance to all training views. $\mathbf{x}_l$ denotes the pixel location in view $\mathbf{l}$ obtained by projecting the unprojected 3D point corresponding to $\mathbf{x}$. Given d_l of a depth value and d_g of the rendered depth of 2DGS in the virtual view, we compute

$$\Delta(\mathbf{x},\mathbf{l}) = \frac{d_l(\mathbf{x}_l) - d_g(\mathbf{x}_l)}{d_l(\mathbf{x}_l) + \epsilon} \quad (7)$$

and validate the visibilities at all N directions for Eq. 6.

B. MATERIAL SEGMENTATION

Shadows that persist across a scene are hard to distinguish from albedo. Existing methods do not explicitly separate albedo from lighting, and shadows and other illumination effects appear baked into the albedo in inverse rendering. We address this by assuming piecewise-constant materials guided by given segments, which encourages the separation of albedo and illumination.

Inspired by the color-based segmentation approach [36], we segment Gaussians by the given material segmentation masks. Similarly to Click-Gaussian [37] (Fig. 2c), each 2D Gaussian is associated with a D-dimensional feature vector $f_i \in \mathbb{R}^D$ for segmentation. Following the rasterization process of 2DGS, the feature at each pixel is computed as $F(\mathbf{x}) = \sum_{i\in N} T_i\,\alpha_i\,f_i$. We train these features using cosine similarity-based contrastive learning. Given a mask M, we sample pixel sets X_1 and X_2 and define two losses. For pixels $\mathbf{x}_1$ and $\mathbf{x}_2$ with $M(\mathbf{x}_1) = M(\mathbf{x}_2)$, we maximize their cosine similarity $S(\mathbf{x}_1,\mathbf{x}_2)$:

$$\mathcal{L}_{\text{cont}}^{\text{pos}} = -\frac{1}{|X_1||X_2|}\sum_{\mathbf{x}_1\in X_1}\sum_{\mathbf{x}_2\in X_2}\mathbb{1}_{M(\mathbf{x}_1)=M(\mathbf{x}_2)}\,S(\mathbf{x}_1,\mathbf{x}_2). \quad (8)$$

For pixels with $M(\mathbf{x}_1) \neq M(\mathbf{x}_2)$, we constrain their similarity with a margin δ:

$$\mathcal{L}_{\text{cont}}^{\text{neg}} = \frac{1}{|X_1||X_2|}\sum_{\mathbf{x}_1\in X_1}\sum_{\mathbf{x}_2\in X_2}\mathbb{1}_{M(\mathbf{x}_1)\neq M(\mathbf{x}_2)}\mathbb{1}_{S(\mathbf{x}_1,\mathbf{x}_2)>\delta}\,S(\mathbf{x}_1,\mathbf{x}_2). \quad (9)$$

We also introduce a mask loss. For each mask k, we compute a prototype μ_k by averaging features within the mask and define

$$\mathcal{L}_{\text{mask}} = \mathcal{L}_{\text{intra}} + \mathcal{L}_{\text{inter}}, \quad (10)$$

where $\mathcal{L}_{\text{intra}}$ pulls features toward their prototype and $\mathcal{L}_{\text{inter}}$ separates different prototypes. We further introduce a norm regularization term $\mathcal{L}_{\text{norm}}$ to stabilize training. Since cosine similarity depends only on the direction of feature vectors, this term constrains their magnitude by encouraging unit-length features, preventing scale instability and facilitating

effective contrastive learning. The overall loss for segmentation is given as

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{cont}}^{\text{pos}} + \lambda_{\text{neg}}\mathcal{L}_{\text{cont}}^{\text{neg}} + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}. \quad (11)$$

After training, we apply K-means clustering to the learned features to assign a segmentation ID to each Gaussian. This enables consistent parameter estimation within each material.

C. INVERSE RENDERING

1) GEOMETRY OPTIMIZATION

We first obtain a standard 2DGS model [21] from multi-view images as our prior to subsequent processes (Fig. 2a). We prefer 2DGS over 3DGS, because the latter can appear deformed around shadow areas in geometry (cf. GS-IR and GI-GS relighting results in Fig. 3). We render normalized depth and normal maps in addition to color maps:

$$\{D, N\} = \sum_{i=1}^{N} w_i \{d_i, \mathbf{n}_i\}, \quad w_i = \frac{T_i \alpha_i}{\sum_{j=1}^{N} T_j \alpha_j}. \quad (12)$$

Inspired by IRGS [30], we use a binary cross-entropy loss [38] to suppress floaters using the binary object mask M_b:

$$\mathcal{L}_{\text{o}} = -M_b \log O - (1 - M_b) \log(1 - O), \quad (13)$$

where $O = \sum_{i=1}^{N} T_i \alpha_i$ is the accumulated opacity map. The overall loss for geometry optimization

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{rend}} + \lambda_{\text{n}}\mathcal{L}_{\text{n}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}} + \lambda_{\text{o}}\mathcal{L}_{\text{o}}. \quad (14)$$

includes the RGB reconstruction loss

$$\mathcal{L}_{\text{rend}} = \|I_{\text{rend}} - I_{\text{gt}}\|_1, \quad (15)$$

where I_{rend} denotes the rendered images, I_{gt} denotes the ground-truth image, a normal consistency loss $\mathcal{L}_n$, and a depth distortion loss $\mathcal{L}_d$ [21].

2) MATERIAL AND LIGHTING DECOMPOSITION

In PBR, we relate the illumination to the pixelwise visibility (Fig. 2d). Existing methods often average visibility and evaluate it only in the normal direction [6], [25]–[27]. In contrast, we represent lighting as the product of directional visibility and a directional light source.

We decompose the incident radiance in (1) as

$$L_{\text{in}}(\mathbf{x}, \mathbf{l}) = V(\mathbf{x}, \mathbf{l})\, L_{\text{dir}}(\mathbf{l}) + L_{\text{ind}}(\mathbf{x}), \quad (16)$$

where $L_{\text{dir}}(\mathbf{l})$ denotes direct illumination from direction $\mathbf{l}$, $V(\mathbf{x}, \mathbf{l}) \in [0, 1]$ is the proposed directional visibility, and $L_{\text{ind}}(\mathbf{x})$ represents indirect illumination modeled with SH. Here, we assume white illumination similar to that of sunlight.

Substituting (16) into (1) yields

$$L_o(\mathbf{x}, \mathbf{v}) = \textstyle\sum_{\mathbf{l}\in\Omega} \left(V(\mathbf{x}, \mathbf{l})\, L_{\text{dir}}(\mathbf{l}) + L_{\text{ind}}(\mathbf{x})\right) f_r(\mathbf{l}, \mathbf{v}, \mathbf{x})(\mathbf{l} \cdot \mathbf{n})\, \Delta\omega, \quad (17)$$

where $\Delta\omega$ denotes the solid angle of each sampled direction.

To separate albedo and lighting, we optimize parameters stored in each 2D Gaussian, including albedo $a(\mathbf{x})$, metallic $m(\mathbf{x})$, and roughness $\rho(\mathbf{x})$ as well as the lighting parameters $L_{\text{dir}}(\mathbf{l})$ and $L_{\text{ind}}(\mathbf{x})$ by minimizing the overall loss for PBR,

$$\mathcal{L}_{\text{pbr}} = \mathcal{L}_{\text{rend}} + \lambda_{\text{ind2}}\mathcal{L}_{\text{ind2}} + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}} + \lambda_{\text{brdf}}\mathcal{L}_{\text{brdf}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \quad (18)$$

Here, $\mathcal{L}_{\text{ind2}} = \|L_{\text{ind}}\|^2$ and $\mathcal{L}_{\text{tv}} = TV(L_{\text{ind}})$ regularize indirect illumination by limiting magnitude and enforcing smoothness. $\mathcal{L}_{\text{brdf}}$ is implemented as a total variation loss on material parameters, encouraging spatial smoothness. $\mathcal{L}_{\text{reg}} = \mathbb{E}[1 - \rho] + \mathbb{E}[m]$ biases roughness towards diffuse values and metallic toward zero. In addition to the above losses, we enforce segment-wise material consistency as a hard constraint. After each optimization step, material parameters within the same segment are averaged, ensuring that each segment shares a consistent appearance. In our implementation, this constraint is applied primarily to albedo. This loss design separates material and lighting and assigns shadows to visibility as attenuation of direct illumination.

V. EXPERIMENTS

We validate the effectiveness of the proposed method for separating lighting and material, particularly for disentangling shadows from albedo and enabling physically consistent relighting.

A. EXPERIMENTAL SETUP

1) IMPLEMENTATION DETAILS

As described in Section IV, our method consists of two stages. In the first stage, we optimize the geometry over 40,000 iterations with hyperparameters λ_n, λ_d, and λ_o set to 0.05, 1000, and 0.01, respectively. In the second stage, we perform material and lighting optimization for 5,000 iterations. Material segmentation is first performed for 1,000 iterations. To obtain material-segmentation masks from input images, we use the Segment Anything Model (SAM3) [39]. We use its Automatic Mask Generation module. When applying K-means clustering, we empirically set the number of clusters to 2 for Bunny, 5 for Armadillo, 40 for Hotdog, and 10 for Ficus. During this stage, λ_{neg}, λ_{norm}, and λ_{mask} are set to 0.1, 0.0001, and 0.5, respectively; the learning rate is 0.005, the temperature τ is 0.3, and 10,000 samples are used. The optimization is performed using Adam [40]. For visibility estimation, the virtual camera is placed at a distance 30 times the average distance between the scene centroid and the training cameras, with a depth threshold of 0.0008. We set $\lambda_{\text{ind2}} = 0.01$, $\lambda_{\text{tv}} = 0.01$, $\lambda_{\text{brdf}} = 1$, and $\lambda_{\text{reg}} = 0.001$. We use 128 directions for environment lighting, and indirect illumination is represented using SH of degree 2. The solid angle $\Delta\omega$ is set uniformly as $\frac{2\pi}{N_l}$, where N_l is the number of sampled directions. The entire training process takes approximately 50 minutes, including 20 minutes for the first stage and 30 minutes for the second stage. Rendering the 200 test views takes approximately 100 seconds. All experiments are conducted on a single NVIDIA RTX A6000 GPU.

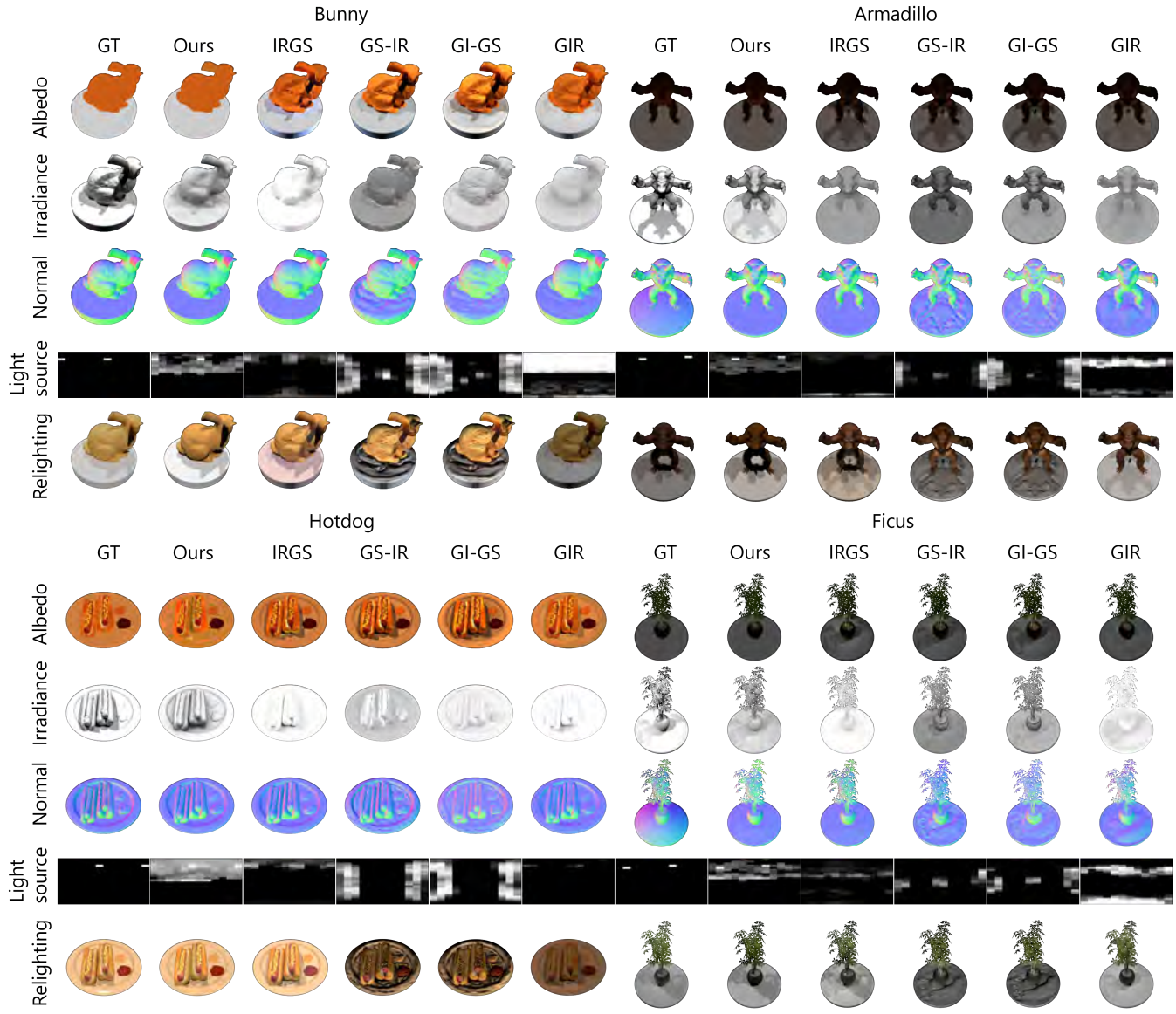

Fig. 3. Visualization of intrinsic decomposition and relighting results on the synthetic dataset. Our method recovers shadow-free albedo and models shadows as illumination effects, producing relighting results with shadows consistent with the target illumination.

2) DATASET

Existing datasets contain no images with hard shadows and are therefore insufficient to evaluate shadow-aware decomposition. We created a synthetic dataset that included shadows. We used the Lego, Hotdog, Ficus, and Armadillo objects from the TensoIR dataset [41], along with the Stanford Bunny [42]. To enable shadow casting, we added a base to each object (except Hotdog, which has a plate) and illuminated the scene using two directional light sources placed in opposite directions. We created object masks to segment out the object areas. All other settings, including camera placement and capture conditions, follow TensoIR dataset.

3) METRICS

Our evaluation focused on whether the method properly separates lighting and material, especially in disentangling shadows from albedo, which is critical for accurate relighting and practical scene editing. Following prior inverse rendering work [6], we evaluate albedo and relighting using PSNR, SSIM [43], and LPIPS [44]. To assess shadow-specific performance in decomposition and relighting, we further compute the cosine similarity of light-source maps and the intersection-over-union (IoU) of shadow regions. The shadow masks are derived from diffuse illumination by thresholding the irradiance. To assess practical usability for interactive scene editing, we also report rendering speed for 200 test views. Although previous work [6], [27], [30]

Table 1. Quantitative comparison on the synthetic dataset. A higher intensity of the red color signifies a better result.

	Albedo			Light	Shadow	Relighting			Render Time
	PSNR↑	SSIM↑	LPIPS↓	Cosine↑	IoU↑	PSNR↑	SSIM↑	LPIPS↓	minutes↓
IRGS	19.98	0.939	0.111	0.005	0.408	18.10	0.872	0.144	11.1
GI-GS	17.68	0.895	0.150	< 0.001	0.659	10.63	0.755	0.216	2.1
GS-IR	18.00	0.902	0.143	0.001	0.589	10.92	0.758	0.212	1.5
GIR	21.76	0.939	0.105	< 0.001	0.421	15.15	0.871	0.137	0.51
Ours	34.02	0.979	0.054	0.324	0.811	17.20	0.873	0.136	1.7

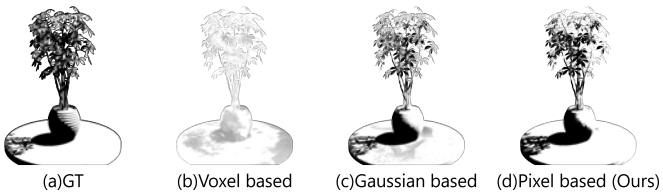

Fig. 4. Comparison of visibility estimation methods. Using image space visibility produces results that better match the rendered appearance and contain fewer artifacts than existing approaches.

Table 2. Visibility computation comparison in average IoU.

Method	Mean IoU
Voxel-based	0.297
Gaussian-based	0.698
Pixel-based (Ours)	**0.750**

generally applies per-channel scaling to a single object for ground-truth alignment, we extend this to all objects, as our scenes include multiple objects with varied colors.

B. RESULTS AND DISCUSSION

We compare the proposed method with state-of-the-art inverse rendering approaches based on 3DGS and 2DGS that include visibility mechanisms for shadow modeling and publicly release their code, namely IRGS [30], GI-GS [27], GS-IR [6], and GIR [28]. All baselines employ physically inspired formulations and operate on multi-view images captured under an unknown lighting condition. Fig. 3 summarizes results on the synthetic dataset. We visualize the intensity of the estimated light sources on a 22×12 map, with individual blocks showing their average intensities to facilitate visual comparisons with spots of light sources (i.e., directional light). The irradiance results show the joint contribution of direct diffuse and indirect diffuse illumination, supporting the visual comparison of the shadowed regions in terms of intensity.

Table 1 shows the superiority of our method across albedo estimation, light-source estimation, shadow-region IoU, and relighting metrics, and demonstrates our main goal of clear separation of materials and lighting. Compared to IRGS [30], which uses ray tracing for light transport, our method shows slightly lower relighting performance in PSNR, but better results both in SSIM and LPIPS. Qualitative results in Fig. 3 demonstrate that the evaluated methods mix the original shadows with newly added shadows in the relighting, whereas ours diminishes the original shadows fairly. Ours renders the relighting scenes in 1.7 minutes on average, which is 6.5 times faster than IRGS. In ours, precomputing visibility with SH further reduces the time to 0.22 minutes, the fastest among the compared methods.

Our method removes shadow components from the estimated albedo and preserves the original surface colors, whereas all compared methods retain lighting and shadow effects in their albedo estimates. The segments produced by SAM3 [39] are object-level rather than material-level, and therefore, a single segment may contain multiple materials. In the Hotdog scene, for example, this requires creating additional clusters to capture finer material distinctions. However, increasing the number of clusters also forces unnecessary splits in single-material regions, such as the plate, resulting in over-segmentation and patchy albedo colors. Our method also recovers light sources at the correct locations as sharp intensity peaks. As a result, shadows rendered from the estimated lighting match the ground truth more closely in position, shape, and strength. In contrast, the compared methods often produce blurred peaks or less accurate light positions. Our method and IRGS [30], both based on 2DGS, recover smooth surface normals even in shadowed regions. Methods with inaccurate normals often misinterpret shadows as geometry-induced shading rather than illumination effects. In GS-IR [6] and GI-GS [27], these normal errors appear directly as shading artifacts.

Relighting results show that our method produces consistent results due to stable geometry and shadow-free albedo. Methods with unstable geometry reveal surface errors after the light source changes. Strong illumination exposes artifacts that are less visible in the original view. In GIR [28], shadows remain unchanged in all results, while the specular component in the Hotdog scene responds to lighting changes. This inconsistency between diffuse shading and specular highlights leads to unrealistic relighting. IRGS [30] models light transport with ray tracing and visibility. However, shadow components remain in the estimated albedo. After relighting, the original shadows remain, while new shadows

Table 3. Ablation study of visibility and segmentation.

	Albedo			Light	Shadow	Relighting		
	PSNR↑	SSIM↑	LPIPS↓	Cosine↑	IoU↑	PSNR↑	SSIM↑	LPIPS↓
w/o visibility	32.78	0.978	**0.047**	0.142	0.746	15.35	0.873	0.141
w/o segmentation	21.94	0.942	0.118	0.245	0.765	**17.37**	**0.875**	**0.136**
Ours	**34.02**	**0.979**	0.054	**0.324**	**0.811**	17.20	0.873	**0.136**

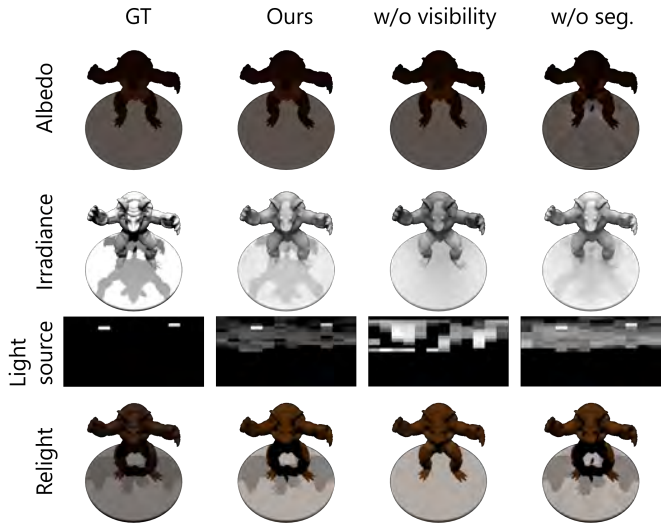

Fig. 5. Effect of ablating visibility and segmentation. Removing visibility suppresses shadow generation, while removing segmentation causes shadows to be baked into the albedo.

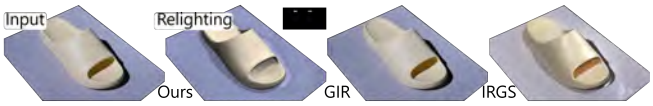

Fig. 6. Relighting results on real-world data. Our method produces shadows consistent with the lighting conditions. In contrast, existing methods retain baked-in shadows, leading to inconsistent relighting results.

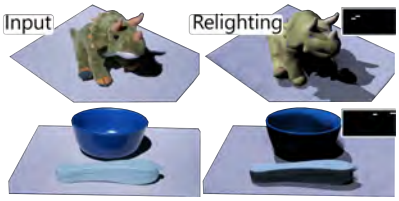

Fig. 7. Failure case in challenging real-world scenes. Segmentation-based smoothing may oversmooth fine details.

are added, which yields unnatural results. These results show that accurate relighting requires proper separation of geometry, material, and lighting.

1) ANALYSIS OF VISIBILITY COMPUTATION

We validate the effectiveness of pixelwise visibility computation. Accurate visibility computation is essential for properly modeling shadows. As described in Section IV A, we compute pixelwise visibility in image space. Table 2 shows IoU scores of visibility maps from different computation methods, evaluated against synthetic ground truth. The reported values are the average over eight light directions. Compared with voxel-based [6] and Gaussian-based approaches [30], pixelwise visibility is more accurate in representing visibility.

Fig. 4 illustrates that methods based on Gaussians produce smoother visibility than pixel-based computation but introduce artifacts due to their reliance on Gaussian distributions (Fig. 4b). Voxel-based methods capture only coarse angular coverage due to their limited voxel resolution (Fig. 4c). In contrast, pixelwise visibility uses observed image-space information, producing fewer artifacts and a more visually consistent representation (Fig. 4d).

C. ABLATION STUDY

Pixelwise visibility models shadows explicitly, while material segmentation separates lighting and material into intrinsic components. We conduct ablation studies on the TensoIR Synthetic dataset to evaluate the contributions of visibility and material segmentation.

In the "w/o visibility" setting in Table 3, we remove the visibility term during training, rendering, and relighting. As shown in Fig. 5, the model cannot generate shadows without visibility, demonstrating that visibility estimation is essential for modeling shadows as a lighting effect.

D. REAL-WORLD RESULT

In the "w/o segmentation" setting in Table 3, we remove the material segmentation mask and its regularization during training. As shown in Fig. 5, without the segmentation constraint, shadow components remain baked into the albedo and distort the underlying material colors. During relighting, both the original shadows and newly generated shadows appear, producing unnatural results. This demonstrates that region-wise material consistency is effective in separating lighting from material. Although the "w/o segmentation" setting achieves better relighting metrics, its albedo contains strong shadow components. This suggests that the segmentation constraint is effective in suppressing shadow baking, while the piecewise-constant albedo assumption may also suppress genuine spatial variations in reflectance within a material region.

We demonstrate the applicability of our method to real-world data. The real-world scene is captured in a laboratory environment under near-darkroom conditions using a single white-light source. For Gaussian segmentation, we empirically use 10 clusters for the Slipper scene, 30 for the Dinosaur scene, and 10 for the Tableware scene. As shown in Fig. 6, our method produces shadows that are consistent with the lighting conditions. This indicates that our approach effectively separates illumination from material, yielding shadow-free albedo while preserving the intrinsic surface appearance. Furthermore, our visibility term enables consistent shadows under novel lighting conditions. In contrast, the other methods retain baked shadows in the albedo, leading to inconsistent relighting. Shadows may remain unchanged, or new ones may be added on top of the original ones, resulting in visually implausible outcomes.

Our method removes baked-in shadows and produces consistent relighting. Fine appearance variations caused by strong specular reflections or complex textures are smoothed by the segmentation constraint (Fig. 7). In the Dinosaur scene, using object-based masks as a proxy for material segmentation groups different material regions into one segment, resulting in overly similar albedo colors. These results highlight a limitation of our segmentation-based material constraint in representing fine-grained material variations.

VI. CONCLUSION

We present a shadow-aware inverse rendering approach for Gaussian Splatting using pixelwise visibility and material segmentation. The proposed method separates material appearance from illumination by treating shadows as lighting effects, allowing physically consistent relighting in scenes with strong shadows. Experimental results show that handling shadows is a key challenge in inverse rendering based on Gaussian Splatting, and that the proposed method provides a practical approach to achieving physically consistent decomposition and relighting.

Our method assumes an object-wise constant albedo, limiting its ability to represent fine-grained surface variations that prior methods often capture in albedo. As such, details should instead be modeled as geometry, improving geometric fidelity is an important direction for future work.

REFERENCES

[1] H. Barrow, J. Tenenbaum, A. Hanson, and E. Riseman, "Recovering intrinsic scene characteristics from images," *Comput. vis. syst*, vol. 2, no. 3-26, p. 2, 1978.

[2] Z. Kuang, Y. Yang, S. Dong, J. Ma, H. Fu, and Y. Zheng, "Olat gaussians for generic relightable appearance acquisition," in *ACM SIGGRAPH Asia Conference Papers*, 2024.

[3] Z. Bi, Y. Zeng, C. Zeng, F. Pei, X. Feng, K. Zhou, and H. Wu, "Gs3: Efficient relighting with triple gaussian splatting," in *ACM SIGGRAPH Asia Conference Papers*, 2024.

[4] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. on Graphics (ToG)*, vol. 42, no. 4, 2023.

[5] J. Gao, C. Gu, Y. Lin, Z. Li, H. Zhu, X. Cao, L. Zhang, and Y. Yao, "Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 73–89.

[6] Z. Liang, Q. Zhang, Y. Feng, Y. Shan, and K. Jia, "Gs-ir: 3d gaussian splatting for inverse rendering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 21 644–21 653.

[7] E. Garces, C. Rodriguez-Pardo, D. Casas, and J. Lopez-Moreno, "A survey on intrinsic images: Delving deep into lambert and beyond," *Int. J. Comput. Vision*, vol. 130, no. 3, pp. 836–868, 2022.

[8] K. Du, Z. Liang, Y. Shen, and Z. Wang, "Gs-id: Illumination decomposition on gaussian splatting via adaptive light aggregation and diffusion-guided material priors," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025, pp. 26 220–26 229.

[9] J. Kaleta, K. Kania, T. Trzciński, and M. Kowalski, "Lumigauss: Relightable gaussian splatting in the wild," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[10] K. Jiang, J.-M. Sun, Z. Li, D. Wang, T.-M. Li, and R. Ramamoorthi, "Differentiable light transport with gaussian surfels via adapted radiosity for efficient relighting and geometry reconstruction," *ACM Trans. on Graphics (ToG)*, vol. 44, no. 6, 2025.

[11] 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," in *European Conference on Computer Vision (ECCV)*, 2020.

[12] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4576–4585.

[13] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4190–4200.

[14] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 5752–5761.

[15] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. on Graphics (ToG)*, vol. 41, no. 4, 2022.

[16] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 5865–5874.

[17] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5438–5448.

[18] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky, "Neural point-based graphics," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 696–712.

[19] J. Chung, J. Oh, and K. M. Lee, "Depth-regularized optimization for 3d gaussian splatting in few-shot images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 811–820.

[20] Y. Li, C. Lyu, Y. Di, G. Zhai, G. H. Lee, and F. Tombari, "Geogaussian: Geometry-aware gaussian splatting for scene rendering," in *European Conference on Computer Vision (ECCV)*, 2024, pp. 441–457.

[21] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH Conference Papers*, 2024.

[22] G. Nam, J. H. Lee, D. Gutierrez, and M. H. Kim, "Practical svbrdf acquisition of 3d objects with unstructured flash photography," *ACM Trans. on Graphics (ToG)*, vol. 37, no. 6, 2018.

[23] Y. Yu and W. A. P. Smith, "Inverserendernet: Learning single image inverse rendering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3150–3159.

[24] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2472–2481.

[25] Y. Zhou, F. Zhang, Z. Wang, and L. Zhang, "Rtr-gs: 3d gaussian splatting for inverse rendering with radiance transfer and reflection," in *ACM International Conference on Multimedia (MM)*, 2025, p. 6888–6897.

[26] Z. Liang, H. Li, K. Jia, K. Guo, and Q. Zhang, "Gus-ir: Gaussian splatting with unified shading for inverse rendering," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 47, no. 10, pp. 8364–8378, 2025.
[27] H. Chen, Z. Lin, and J. Zhang, "Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering," in *International Conference on Learning Representations (ICLR)*, vol. 2025, 2025, pp. 75 713–75 742.
[28] Y. Shi, Y. Wu, C. Wu, X. Liu, C. Zhao, H. Feng, J. Zhang, B. Zhou, E. Ding, and J. Wang, "Gir: 3d gaussian inverse rendering for relightable scene factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025.
[29] Y. Guo, Y. Bai, L. Hu, Z. Guo, M. Liu, Y. Cai, T. Huang, and L. Ma, "Prtgs: Precomputed radiance transfer of gaussian splats for real-time high-quality relighting," in *ACM International Conference on Multimedia (MM)*, 2024, pp. 5112–5120.
[30] C. Gu, X. Wei, Z. Zeng, Y. Yao, and L. Zhang, "Irgs: Inter-reflective gaussian splatting with 2d gaussian ray tracing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 10 943–10 952.
[31] J. T. Kajiya, *The rendering equation*, 1998, pp. 157–164.
[32] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM Trans. on Graphics (ToG)*, vol. 1, no. 1, pp. 7–24, 1982.
[33] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces," in *Eurographics Symposium on Rendering (EGSR)*, 2007, p. 195–206.
[34] K. Ye, C. Gao, G. Li, W. Chen, and B. Chen, "Geosplatting: Towards geometry guided gaussian splatting for physically-based inverse rendering," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025, pp. 28 991–29 000.
[35] L. Williams, "Casting curved shadows on curved surfaces," in *ACM SIGGRAPH Conference Papers*, 1978, p. 270–274.
[36] X. Lv, L. Li, and Z. Chen, "Semantic-aware hierarchical clustering for inverse rendering in indoor scenes," *SIGGRAPH Comput. Graph.*, vol. 129, p. 104236, 2025.
[37] S. Choi, H. Song, J. Kim, T. Kim, and H. Do, "Click-gaussian: Interactive segmentation to any 3d gaussians," in *European Conference on Computer Vision (ECCV)*, 2025, pp. 289–305.
[38] 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," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
[39] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang *et al.*, "Sam 3: Segment anything with concepts," 2025.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.
[41] H. Jin, I. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su, "Tensoir: Tensorial inverse rendering," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 165–174.
[42] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *ACM SIGGRAPH Conference Papers*, 1994, pp. 311–318.
[43] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.
[44] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.