ShadowSG: Spherical Gaussian Illumination from Shadows

Hanwei Zhang[1*] Xu Cao[2] Hiroshi Kawasaki[1] Takafumi Taketomi[2]
[1]Kyushu University, ISEE, Japan [2]CyberAgent, Japan
zhang.hanwei.706@s.kyushu-u.ac.jp kawasaki@ait.kyushu-u.ac.jp
{xu_cao, taketomi_takafumi}@cyberagent.co.jp

Abstract

This work leverages shadow cues in a scene to infer the surrounding illumination of the shadow-casting object. Unlike prior works that optimize a discrete environment map, we model scene illumination using a mixture of spherical Gaussians (SGs). SG illumination provides more intuitive relations to shadow appearance and offers a more compact parameterization compared to discrete environment maps. To estimate SG parameters, we employ an SG-based, differentiable, closed-form rendering equation to explain the shading of the shadow plane and minimize a photometric loss between the rendered and observed shadow plane shading. Experiments on synthetic and real-world images under various surrounding illumination demonstrate that our method estimates illumination more accurately than approaches based on discrete environment maps. With our estimated lighting, consistent shadow effects are realized when blending virtual objects into real-world images[1].

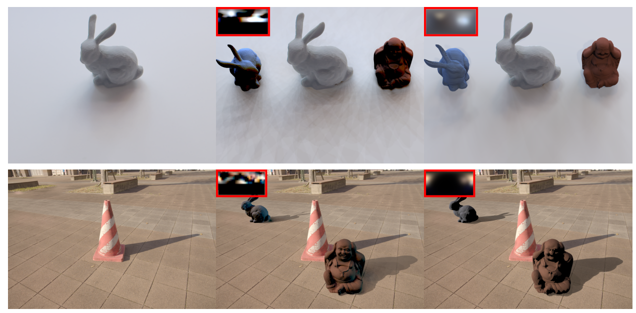

Figure 1. We estimate the surrounding illumination (highlighted in red) of a scene object from the appearance of the shadow plane. Unlike SH21 [28], which optimizes a discrete environment map, our method parameterizes illumination as a mixture of SGs. This results in more photorealistic and coherent shadow effects when blending virtual objects into real scenes.

1. Introduction

Inferring scene illumination from the appearance of scene components is a fundamental yet challenging task in computer vision and graphics, with significant applications in augmented reality. Accurate illumination estimation is essential for seamlessly blending virtual objects into real environments. Perhaps the most straightforward way is to use an omnidirectional camera to capture the observer's surrounding lighting as a high dynamic range (HDR) light probe image. However, this method is less effective when estimating incident illumination at a scene point distant from the observer. To address this, chrome mirror balls [9] or other reflective objects [24, 35] have been used, but such objects are not always present in a scene, limiting the applicability of these methods. Alternatively, learning-based approaches [7, 13, 31] have been explored and rely on data-driven priors to estimate scene lighting.

In this work, we specifically focus on shadow cues and investigate how accurately illumination can be recovered from them using an analysis-by-synthesis approach, as illustrated in Fig. 1. Shadows are a ubiquitous light transport phenomenon that occurs whenever a scene object obstructs incident light (*a.k.a.*, a pinspeck camera [6, 29]). As shown in Fig. 2, shadows provide essential lighting cues surrounding the shadow-casting object. For example, one can infer the light source's direction by examining the position of a shadow relative to its caster. Higher contrast in the shadow plane suggests that the light from a particular direction is brighter, while the shadow boundary's sharpness can indicate the light source's extent. A blurred boundary suggests that the light source is spread over a wide area.

Despite the intuitive observations, inferring the illumination surrounding a scene object from its cast shadows is challenging. Prior illumination-from-shadow approaches model the surrounding illumination as intensities from discretized incident directions [27, 28]. Since the incident directions are predefined, higher precision in lighting es-

[*]Work conducted during the first author's internship at CyberAgent.
[1]Code is available at https://github.com/CyberAgentAILab/ShadowSG.

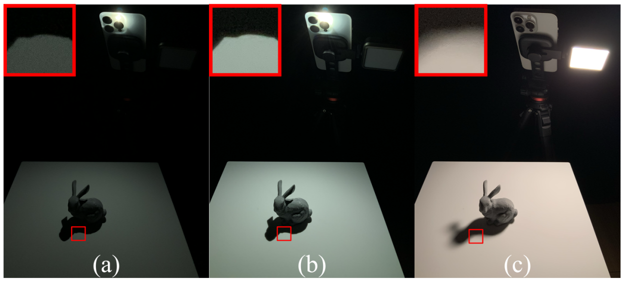

Figure 2. The appearance of shadows reveals the lighting conditions surrounding the shadow-casting object. **(a, b)** A brighter light source produces a higher contrast on the shadow plane. **(b, c)** Light sources spreading across a wider area create a more blurred shadow boundary.

timation requires increasing the number of sampled directions. This leads to a quadratic increase in the number of unknowns [28] or requires a more complex coarse-to-fine optimization process [27].

This work introduces a spherical Gaussian (SG) representation for inferring illumination from shadows. We model the illumination surrounding a scene object as a mixture of SGs and recover the SG parameters from the shadow plane's shading. The SG representation offers two main advantages: 1) it provides a more intuitive relationship to the appearance of shadows, and 2) it offers a more compact parameterization than a discrete environment map. Each SG consists of three parameters influencing the shadow's location, boundary sharpness, and contrast.

To model the shadow plane's appearance, we use an SG-based, differentiable, closed-form rendering equation [11, 32] which accounts for light visibility and efficiently computes shading. The SG parameters are recovered by minimizing the photometric loss between the rendered and observed shading of the shadow plane. Originally developed for forward rendering, our results suggest that this rendering equation is also effective for the inverse problem of recovering SG illumination from shadows.

We conduct extensive experiments using both synthetic and real-world data to verify the effectiveness of our method. Compared to prior illumination-from-shadow approaches, our method more accurately recovers surrounding illumination and achieves superior shadow plane rendering quality. Using our estimated illumination, we demonstrate that virtual objects can be inserted into real scenes with more photorealistic and coherent shadow effects.

In summary, our contributions are:

- We introduce spherical Gaussian lighting representations for inferring illumination from shadows;
- We show that the SG-based rendering equation, originally developed for forward rendering, is effective for the inverse problem of estimating SG light parameters from the shading of the shadow plane;
- We validate our method for illumination estimation and virtual object insertion using synthetic and real-world images under various environmental lighting conditions.

2. Related Work

2.1. Illumination from Shadows

Sato *et al*. [27] were among the first to study and formulate the problem of estimating illumination from shadows. They parameterized the surrounding illumination of the shadow-casting object as intensities from pre-defined incident directions. These intensities were estimated by solving a linear system using the known geometry of the shadow caster.

To enhance practicality, Ikeda *et al*. [14] employed an RGB-D camera to capture both the scene geometry and the shadow image. In addition to shadows, Li *et al*. [21] integrated multiple clues—including shadows, textures, and reflectance—to estimate illumination. Sato *et al*. [23] extended their technique by utilizing spherical harmonics and Haar wavelets to model cast shadows. Jiddi *et al*. [16] concentrated on estimating the positions and intensities of point light sources from cast shadows.

Recently, Swedish *et al*. [28] proposed a method that jointly solves for the diffuse albedo of the shadow plane and a discretized environment map. They introduced two regularization terms into the linear system to enhance stability. However, similar to the problems faced by [23, 27], the number of unknowns increases quadratically with higher-resolution environment map estimation, leading to instability. Consequently, their estimated environment map is limited to low resolutions (*e.g*., 16×64).

In contrast, we parameterize illumination using a mixture of SGs, which is more compact than a discrete environment map and avoids the issue of parameter proliferation. Our method is particularly effective in scenes with one or more dominant directional or area light sources while also performing well under general illumination conditions.

2.2. Spherical Gaussian Representation

Spherical Gaussian (SG) representations have been employed in both forward and inverse rendering tasks. In forward rendering, SG illumination is useful for fast rendering [11, 15, 30]. SGs have also been utilized to approximate surface reflection [12, 15, 32] and refraction [8]. In inverse rendering, SGs have been applied to approximate the specular component of surface reflectance [36] and illumination [4, 17, 33, 36, 38]. These methods typically rely on the shading of scene objects for illumination estimation, and cast shadows degrade the quality of inverse rendering [33].

Our method offers a new perspective by demonstrating that SG illumination can be recovered from cast shadows. We show that the SG-based forward rendering formula [32] is also effective for the inverse problem.

3. Approach

We aim to estimate the surrounding illumination of a scene object from the appearance of its shadows cast on the plane where the object is placed. Our method takes a single image as input and outputs the illumination parameterized as a mixture of spherical Gaussians (SGs).

Overview Figure 3 summarizes our method. Section 3.1 describes the physics-based rendering equation approximated by the SGs. Section 3.2 then derives a closed-form rendering equation differentiable with respect to the unknown SG parameters. Finally, Sec. 3.3 details our strategy for optimizing SG parameters based on the closed-form rendering equations.

Assumptions Following prior works [27, 28], we assume that the scene geometry is known, the shadow plane is Lambertian, and the surrounding illumination is sufficiently distant to be considered spatially invariant. As shown in Sec. 4.2, the scene geometry can be obtained via photogrammetry or monocular estimators when the shadow caster has strong prior information, such as human bodies.

3.1. Image Formation Model

We model the appearance of a diffuse, opaque shadow plane under natural lighting using Kajiya's rendering equation [18]. For a shadow plane point $\mathbf{x}$ with normal $\mathbf{n} \in \mathbb{S}^2$, the observed radiance L_o is given by:

$$L_o(\mathbf{x}) = k_d(\mathbf{x}) \int_{\mathbb{S}^2} L(\boldsymbol{\omega}) V(\mathbf{x}, \boldsymbol{\omega}) \max(\mathbf{n}^\top \boldsymbol{\omega}, 0)\, d\boldsymbol{\omega}. \quad (1)$$

In Eq. (1), $k_d(\mathbf{x})$ represents the direction-independent diffuse albedo. The light distribution $L(\boldsymbol{\omega})$ describes the light intensity from each incident direction $\boldsymbol{\omega}$. The light visibility $V(\mathbf{x}, \boldsymbol{\omega})$ is a binary function indicating whether the shadow plane point $\mathbf{x}$ is occluded by the shadow caster in the direction $\boldsymbol{\omega}$. Under the distant lighting assumption, the light distribution is spatially invariant, while light visibility depends on the scene geometry and varies across the shadow plane.

The key challenge in applying Eq. (1) for inverse rendering is evaluating the integral. Prior works discretize the hemisphere into solid angles. However, this can lead to the parameter proliferation problem for fine-grained illumination estimation. Instead, we approximate the light distribution and the cosine term using spherical Gaussians (SGs) to derive a closed-form rendering equation in terms of SG parameters. An SG is a Gaussian function defined on the surface of a sphere $\mathbb{S}^2$, given by:

$$G(\boldsymbol{\omega}; \boldsymbol{\mu}, \lambda, \mathbf{a}) = \mathbf{a} e^{\lambda(\boldsymbol{\omega}^\top \boldsymbol{\mu} - 1)}. \quad (2)$$

An SG is defined by three parameters: the unit lobe axis $\boldsymbol{\mu} \in \mathbb{S}^2$, the lobe sharpness $\lambda \in \mathbb{R}_+$, and the non-negative lobe amplitude $\mathbf{a} \in \mathbb{R}_+^m$, where m is the dimension of the amplitude (*e.g.*, $m = 3$ for RGB channels).

We represent the light distribution as a mixture of SGs:

$$L(\boldsymbol{\omega}) \approx \sum_{i=1}^{N} G(\boldsymbol{\omega}; \boldsymbol{\mu}_i, \lambda_i, \mathbf{a}_i) := \sum_{i=1}^{N} G_i, \quad (3)$$

where N is the total number of light SGs. The scene illumination is then parameterized by the set of unknown SG parameters $\{\boldsymbol{\mu}_i, \lambda_i, \mathbf{a}_i\}_{i=1}^N$.

The cosine term $\max(\mathbf{n}^\top \boldsymbol{\omega}, 0)$ is approximated by a single SG, where $\boldsymbol{\omega}$ is the variable and $\mathbf{n}$ is the lobe axis:

$$\max(\mathbf{n}^\top \boldsymbol{\omega}, 0) \approx G(\boldsymbol{\omega}; \mathbf{n}, \lambda_c, a_c) := G_c. \quad (4)$$

Following [32], we set $\lambda_c = 2.133$ and $a_c = 1.170$. There are no unknown parameters in G_c, as the normal of the shadow plane is assumed to be known.

With the SG approximations Eqs. (3) and (4), we rewrite the rendering equation Eq. (1) as:

$$\begin{aligned} L_o(\mathbf{x}) &\approx k_d(\mathbf{x}) \int_{\mathbb{S}^2} \left(\sum_{i=1}^{N} G_i(\boldsymbol{\omega}) \right) G_c(\boldsymbol{\omega}) V(\mathbf{x}, \boldsymbol{\omega})\, d\boldsymbol{\omega} \\ &= k_d(\mathbf{x}) \sum_{i=1}^{N} \int_{\mathbb{S}^2} G_i(\boldsymbol{\omega}) G_c(\boldsymbol{\omega}) V(\mathbf{x}, \boldsymbol{\omega})\, d\boldsymbol{\omega}. \end{aligned} \quad (5)$$

We omit the SG parameters for brevity. Equation (5) states that the appearance of the shadow plane is composed of N images, each generated by individual SGs $G_i(\boldsymbol{\omega})$.

Compared to discrete representations, each parameter of an SG contributes more intuitively to the shadow plane's appearance, as illustrated in Fig. 4. This compactness allows our method to achieve better results with fewer parameters, as discussed in more detail in Sec. 4.

3.2. Closed-form rendering equation

This section derives a closed-form equation for Eq. (5) regarding illumination SG parameters. When two SGs are multiplied, the result is a new SG with updated parameters:

$$\begin{aligned} & G_i G_c = G_i'(\boldsymbol{\omega}; \boldsymbol{\mu}_i', \lambda_i', \mathbf{a}_i') \\ \text{with} \quad & \boldsymbol{\mu}_i' = \frac{\lambda_i \boldsymbol{\mu}_i + \lambda_c \mathbf{n}}{\|\lambda_i \boldsymbol{\mu}_i + \lambda_c \mathbf{n}\|}, \quad \lambda_i' = \lambda_i + \lambda_c, \\ \text{and} \quad & \mathbf{a}_i' = a_c \mathbf{a}_i e^{\lambda_i'(\|\boldsymbol{\mu}_i'\| - 1)}. \end{aligned} \quad (6)$$

The key challenge then is to calculate the integral of the product between two spherical functions, the SG G_i' and the light visibility V. To approximate this integral, we first calculate the analytically derived integral of the SG G_i' over

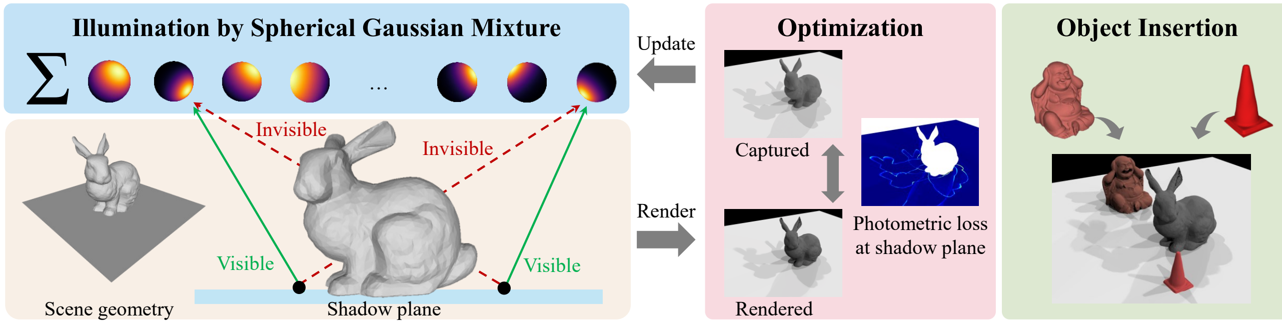

Figure 3. **Overview.** We represent scene illumination as a mixture of spherical Gaussians (SGs) (Sec. 3.1) and use a closed-form, differentiable rendering equation to model the shadow plane shading (Sec. 3.2). The SG parameters are estimated by minimizing the photometric loss on the shadow plane (Sec. 3.3). After optimization, the estimated SGs allow for seamlessly blending virtual objects into the scene.

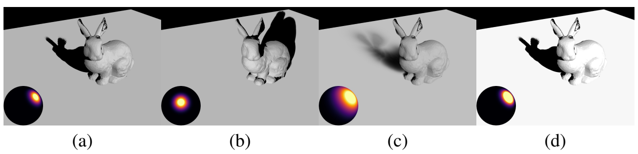

Figure 4. Each parameter of an SG intuitively influences the appearance of the cast shadow. Each image is rendered by varying one parameter from the SG in **(a)**. The bottom left of each image shows the corresponding SG visualized on a sphere. **(a, b)** The lobe axis adjusts the shadow's position. **(a, c)** Lobe sharpness controls the sharpness of the shadow's boundary. **(a, d)** Lobe amplitude affects the contrast of the shading on the shadow plane.

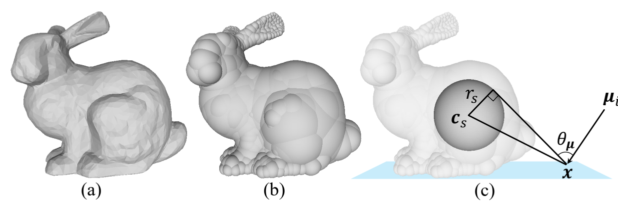

Figure 5. **(a, b)** The shape of the shadow-casting object is approximated by a mixture of spheres. **(c)** The angular distance $\theta_{\boldsymbol{\mu}}$ from an SG lobe axis $\boldsymbol{\mu}_i$ to a sphere s is analytically calculated using Eq. (9). The smallest angular distance is the minimum of the angles to all spheres (Eq. (10)).

a hemisphere centered at the lobe axis and then scale the result using a sigmoid function [32]:

$$\int_{\mathbb{S}^2} G_i' V \, d\boldsymbol{\omega} \approx \sigma \int_{\mathbb{S}^2} G_i' \, d\boldsymbol{\omega} = \sigma \frac{2\pi \mathbf{a}_i'}{\lambda_i'} \left(1 - e^{-\lambda_i'}\right). \quad (7)$$

This sigmoid function σ depends on two factors: 1) the SG sharpness λ_i', and 2) the signed angular distance $\theta_{\boldsymbol{\mu}}$ from the SG lobe axis $\boldsymbol{\mu}_i$ to the closest visible boundary:

$$\sigma(\lambda_i', \theta_{\boldsymbol{\mu}}) = \frac{1}{1 + \exp(-k(\lambda_i')\theta_{\boldsymbol{\mu}})}, \quad (8)$$

where $k(\lambda_i') = 0.204\lambda_i'^3 - 0.892\lambda_i'^2 + 2.995\lambda_i' + 0.067$ designed by [32]. Equations (7) and (8) suggest that: 1) the integral of a sharper SG is less affected by light visibility, as the SG values are more concentrated around the lobe axis, and 2) an SG with a lobe axis far from the light visibility boundary is also less affected since the SG values diminish rapidly away from the lobe axis.

The remaining challenge is efficiently finding the minimal angular distance $\theta_{\boldsymbol{\mu}}$. A naïve approach would sample incoming directions, calculate their angular distances to the SG lobe axis, and select the direction on the visibility boundary with the smallest angular distance [32]. However, this approach reintroduces significant computational complexity, negating the compactness of SG parameterization.

Inspired by [11], we approximate the shadow caster's geometry as a mixture of spheres using the adaptive media-axis approximation [5]. This reduces the evaluation from the number of incident directions to the number of spheres used to approximate the shadow caster's geometry. Specifically, for a shadow plane point $\mathbf{x}$, the minimal angular distance from the SG lobe axis $\boldsymbol{\mu}_i$ to the visibility boundary introduced by a sphere s can be calculated as

$$\theta_{\boldsymbol{\mu}}(\mathbf{x}, s) = \arccos\left(\frac{\mathbf{s}^\top \boldsymbol{\mu}_i}{||\mathbf{s}||}\right) - \arcsin\left(\frac{r_s}{||\mathbf{s}||}\right), \quad (9)$$

where $\mathbf{s} = \mathbf{c}_s - \mathbf{x}$ is the vector connecting the sphere center and the shadow plane point, and r_s is the sphere's radius. We calculate $\theta_{\boldsymbol{\mu}}(\mathbf{x}, s)$ for each sphere s, and $\theta_{\boldsymbol{\mu}}$ is the smallest one among them:

$$\theta_{\boldsymbol{\mu}}(\mathbf{x}) = \min_s \theta_{\boldsymbol{\mu}}(\mathbf{x}, s). \quad (10)$$

We use a tree structure to store the mixture of spheres, organized from coarse to fine. The smallest $\theta_{\boldsymbol{\mu}}$ can be found in a top-down manner, further reducing evaluation number [11].

3.3. Illumination optimization

With the closed-form rendering equation described in Sec. 3.2, we optimize the light SG parameters

$\{\boldsymbol{\mu}_i, \lambda_i, \mathbf{a}_i\}_{i=1}^N$ such that the rendered radiances of the shadow plane are close to the observed radiance. We define the objective function as

$$\mathcal{L} = \frac{1}{N_b} \sum_{j=1}^{N_b} (L_{o,j} - I_j)^2, \tag{11}$$

where $L_{o,j}$ and I_j are the rendered and observed radiance at a pixel j sampled from the shadow plane, and N_b is the total number of pixels sampled in each iteration.

Reparameterization To enforce nonnegative SG lights (*i.e.*, $\mathbf{a}_i \geq 0 \quad \forall i$), we optimize the absolute value $|\mathbf{a}_i|$ instead of $\mathbf{a}_i$. This reparameterization ensures non-negativity during rendering, stabilizes the optimization process, and eliminates the necessity to pre-determine the exact number of light SGs. Once a subset of SGs accurately reproduces the observed radiance, the remaining SGs can be optimized to have tiny amplitudes, thereby minimizing their influence on the rendered image.

Adaptive pruning To further improve rendering quality, we regularly prune SGs that contribute only marginally to the rendered image during optimization. We observe that SGs with tiny amplitudes and large sharpness can still produce almost unnoticeable hard shadows, thereby degrading the overall rendering quality. However, optimizing these SGs is difficult because their gradients nearly vanish. To reduce the influence of such SGs, we employ an adaptive pruning strategy [19]. Specifically, we use two thresholds: one for $\mathbf{a}_i$ and another for its gradient $\nabla \mathbf{a}_i$. An SG is pruned when both the infinity norm $\|\mathbf{a}_i\|_\infty$ and $\|\nabla \mathbf{a}_i\|_\infty$ fall below their respective thresholds. This strategy further makes our method less sensitive to the initial number of SGs since redundant ones are eventually pruned.

4. Experiments

We evaluate our method against baselines using synthetic images in Sec. 4.1 and real-world images in Sec. 4.2.

Baselines We compare our method with two illumination-from-shadow methods SS03 [27] and SH21 [28]. We re-implemented both methods due to the lack of publicly available implementations. Both methods represent surrounding illumination as light intensities from predefined incident directions. We define the incident directions using a latitude-longitude format with a resolution of 16×64 (*a.k.a.*, environment map). Both methods formulate the problem as a linear system, and we use PyTorch's *lstsq* as the solver. For SS03 [27], we additionally apply Scipy's non-negative least-squares solver *nnls*, which enforces the non-negative values in the estimated environment map, and denote this variant as SS03$_+$ [27].

Evaluation metrics. We use PSNR, SSIM, and LPIPS [37] to evaluate the re-rendered image quality after optimization. The metrics are calculated for pixels on the shadow plane. To evaluate the estimated environment maps, we use angular error following [10, 35]. The angular error is calculated as the average angular distance between the RGB values of the estimated and ground truth environment maps.

Implementation details We implement our method using the PyTorch framework [25]. The surrounding illumination is initialized in all experiments as $N = 256$ SGs. The initial SG lobe axes are uniformly sampled from the hemisphere, while the lobe amplitudes and sharpness are randomly initialized as positive values. We approximate the object mesh using a sphere tree with 3 levels and 8 branches, yielding a total of 512 spheres, following the method in [5].

For optimization, we use the Adam optimizer [20] with an initial learning rate of 0.001, which is annealed by 0.9 every 5000 step. At each step, we sample $N_b = 1024$ pixels and optimize for a total of 20000 steps. Adaptive pruning begins at step 500 and terminates at step 15000. The thresholds for SG lobe amplitude and its gradient during the adaptive pruning are set to 0.005 and 0.001, respectively. The experiments are conducted on an NVIDIA RTX A6000 GPU, with a runtime of approximately 5 minutes per scene.

4.1. Evaluations on synthetic images

Rendering setup We use Blender Cycles to render synthetic HDR images. The scene consists of a Stanford Bunny placed on a plane with diffuse reflectance. Three types of surrounding lighting are used: **(a)** a single or a mixture of directional lights, implemented using Blender's SUN light type and denoted as DIRECTION1 and DIRECTIONS; **(b)** a single or a mixture of area lights, implemented using Blender's AREA light type and denoted as AREA1 and AREAS; and **(c)** real-world indoor or outdoor lighting using publicly available HDR light probe images. This way, we ensure that the lighting conditions are varied and realistic.

Results and discussions Table 1 reports the quantitative evaluations, and Fig. 6 displays the qualitative results of the re-rendered image and estimated illumination. Our method consistently outperforms the baselines across most illumination types, particularly excelling in complex lighting conditions such as DIRECTIONS and real-world HDR maps. These results demonstrate the robustness and effectiveness of our approach in accurately estimating lighting and maintaining high-quality rendered images.

Figure 7 and Table 2 quantitatively evaluate the rendered image quality when additional objects are inserted. For this evaluation, we render a pair of images using Blender:

Figure 6. **Qualitative results on synthetic images.** In each image block, the left side shows the re-rendered scene after illumination optimization, the upper right displays the estimated lighting in latitude-longitude format, and the bottom right presents the error map between the input and re-rendered shadow planes.

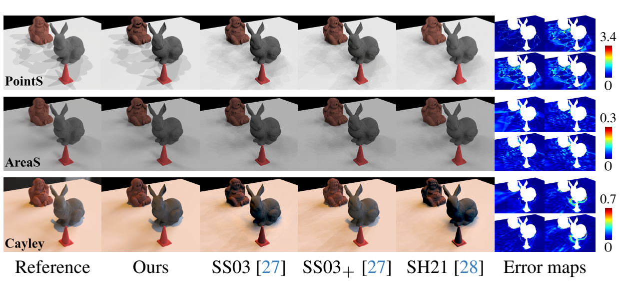

Figure 7. **Qualitative results of virtual object insertion on synthetic images**. Our approach produces more coherent shadow effects when blending additional objects with the original scenes.

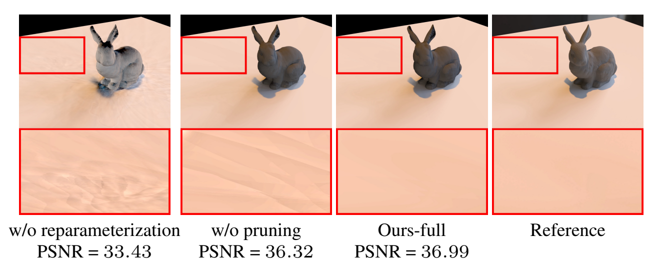

Figure 8. **Ablation Study.** Our reparameterization and pruning strategies enhance SG optimization and improve rendering quality. Close-up views are sharpened for better visualization.

one with only the shadow-casting object used for illumination estimation and another with additional objects for re-rendering evaluation. After blending the virtual objects with the original image, we assess the rendering quality on the shadow plane. As shown in Table 2, our method achieves more accurate shadow plane rendering, indicating the coherence of rendered shadows. Figure 7 further confirms that our method renders more photorealistic shadow effects.

Table 1. **Quantitative evaluation on re-rendered image quality and estimated lighting accuracy using synthetic images.** Our method outperforms the baselines, particularly in scenarios with dominant light sources such as DIRECTIONS, PINE, and RURAL.

Illumination	Method	PSNR↑	Image SSIM↑	LPIPS↓	Env. map Ang err.↓
DIRECTION1	SS03 [27]	19.64	0.924	0.22	-
	$SS03_+$ [27]	19.41	0.931	0.12	-
	SH21 [28]	19.35	0.927	0.20	-
	Ours	**27.79**	**0.962**	**0.04**	-
DIRECTIONS	SS03 [27]	27.67	0.961	0.21	-
	$SS03_+$ [27]	27.65	0.965	0.17	-
	SH21 [28]	27.31	0.967	0.20	-
	Ours	**34.49**	**0.982**	**0.06**	-
AREA1	SS03 [27]	34.80	0.986	0.11	71.46
	$SS03_+$ [27]	34.40	0.984	0.12	73.96
	SH21 [28]	34.80	0.986	0.11	**71.21**
	Ours	**36.30**	**0.996**	**0.03**	80.55
AREAS	SS03 [27]	35.41	0.986	0.14	80.31
	$SS03_+$ [27]	35.42	0.985	0.14	80.44
	SH21 [28]	34.93	0.987	0.14	**80.25**
	Ours	**38.00**	**0.997**	**0.04**	85.07
CAYLEY	SS03 [27]	30.44	0.983	0.19	41.36
	$SS03_+$ [27]	33.70	0.987	0.15	68.19
	SH21 [28]	31.34	0.986	0.15	43.33
	Ours	**36.99**	**0.996**	**0.05**	**9.61**
PINE	SS03 [27]	32.18	0.981	0.21	40.04
	$SS03_+$ [27]	33.45	0.985	0.16	56.90
	SH21 [28]	31.68	0.986	0.20	47.83
	Ours	**40.46**	**0.998**	**0.06**	**16.95**
RURAL	SS03 [27]	21.71	0.953	0.06	51.38
	$SS03_+$ [27]	21.58	0.961	0.06	66.86
	SH21 [28]	21.70	0.954	0.06	44.54
	Ours	**30.98**	**0.991**	**0.02**	**16.53**

Table 2. **Quantitative evaluation on virtual object insertion.** Our method delivers more consistent shadow effects when blending virtual objects with real scenes.

	DIRECTIONS			AREAS			CAYLEY		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SS03 [27]	26.07	0.940	0.22	**32.47**	0.978	0.17	30.66	0.977	0.21
$SS03_+$ [27]	25.91	0.944	0.20	32.42	0.976	0.17	**33.09**	0.983	0.17
SH21 [28]	25.67	0.947	0.22	32.23	0.980	0.17	31.21	0.982	0.19
Ours	**31.44**	**0.967**	**0.09**	31.44	**0.993**	**0.06**	32.80	**0.993**	**0.06**

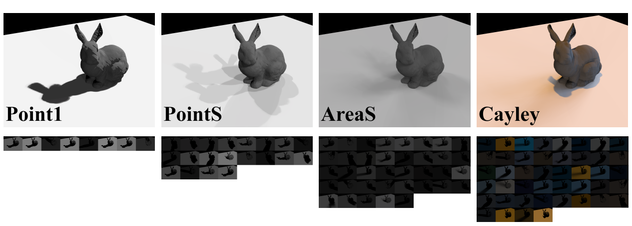

Figure 9. Our pruning strategy results in a varying number of SGs after optimization. More SGs are retained under more complex lighting conditions. **(Top)** Image rendered by compositing all SGs. **(Bottom)** Image arrays rendered by individual SGs.

Figure 8 studies the effectiveness of our reparameterization and adaptive pruning optimization strategies. Without reparameterization, the SG amplitude can become negative, leading to unstable optimization. Without pruning, SGs with low amplitude and high sharpness may persist, negatively affecting the quality of the re-rendered shadow plane. By combining both strategies, our method more accurately reproduces the shading of the shadow plane.

Table 3. **Quantitative evaluation on re-rendered image quality using real-world images.** Our method outperforms baselines in most real-world scenarios, even when the assumption of a shadow plane with uniform albedo is unmet.

	3D-PRINT1		3D-PRINT2		PHOTOGRAMETRY1		PHOTOGRAMETRY2		MONO	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
SS03 [27]	27.08	0.983	28.41	0.986	22.31	0.588	25.68	0.696	11.11	**0.271**
$SS03_+$ [27]	29.90	0.984	29.74	0.986	22.42	0.590	25.82	0.703	10.89	0.266
SH21 [28]	30.06	0.985	30.78	0.985	**22.50**	0.589	25.80	0.703	**11.69**	0.270
Ours	**33.82**	**0.995**	**32.52**	**0.995**	22.46	**0.591**	**25.94**	**0.713**	10.59	0.268

Figure 9 shows that our method is insensitive to the initial number of SGs, as the optimization ultimately retains only the necessary SGs. More complex lighting conditions require a greater number of SGs. For example, 8 SGs remain when there is a single directional light source, while 44 SGs are retained when there is a combination of directional and area light sources in the case of CAYLEY.

The discrete environment map-based method optimizes 3072 parameters, even for a low-resolution 16×64 map. In contrast, our optimization starts with 1536 parameters (256 SGs with 6 parameters each) and eventually reduces to fewer parameters (*e.g.*, 48 parameters for 8 SGs). Despite using fewer parameters, our method achieves better rendering quality, thanks to the compact SG representation.

4.2. Evaluations on real-world images

Capture setups We evaluate our lighting estimation on real-world images across three scenarios, each differing in how the scene geometry is obtained. **(3D Print)** First, we place a 3D-printed object on a matte whiteboard. In this setup, the geometry of the scene object is known, and the shadow plane's albedo is uniform. The object and shadow plane are positioned in front of an OLED display that illuminates the scene, and the raw image is captured with an iPhone 13 Pro. Intrinsic camera parameters are calibrated using a checkerboard in Metashape [2], and the perspective-n-points algorithm [22] is used to estimate the object pose. Correspondence points are manually selected from the mesh and the captured image using Meshlab [1]. **(Photogrammetry)** Second, we capture outdoor scenes from multiple viewpoints. The camera parameters, shadow-casting object geometry, and shadow plane normal are estimated using Metashape [2]. One of the multi-view images is then used for illumination estimation. **(Monocular Estimation)** Third, we use a portrait image captured in natural, uncontrolled conditions sourced from the Internet. In this scene, the human body casts a shadow on the ground. We recover the human body geometry as a SMPL-X mesh [26] using a monocular human body shape estimation method [34], estimate the ground surface normal using a monocular normal estimation method [3], and treat the camera as orthographic.

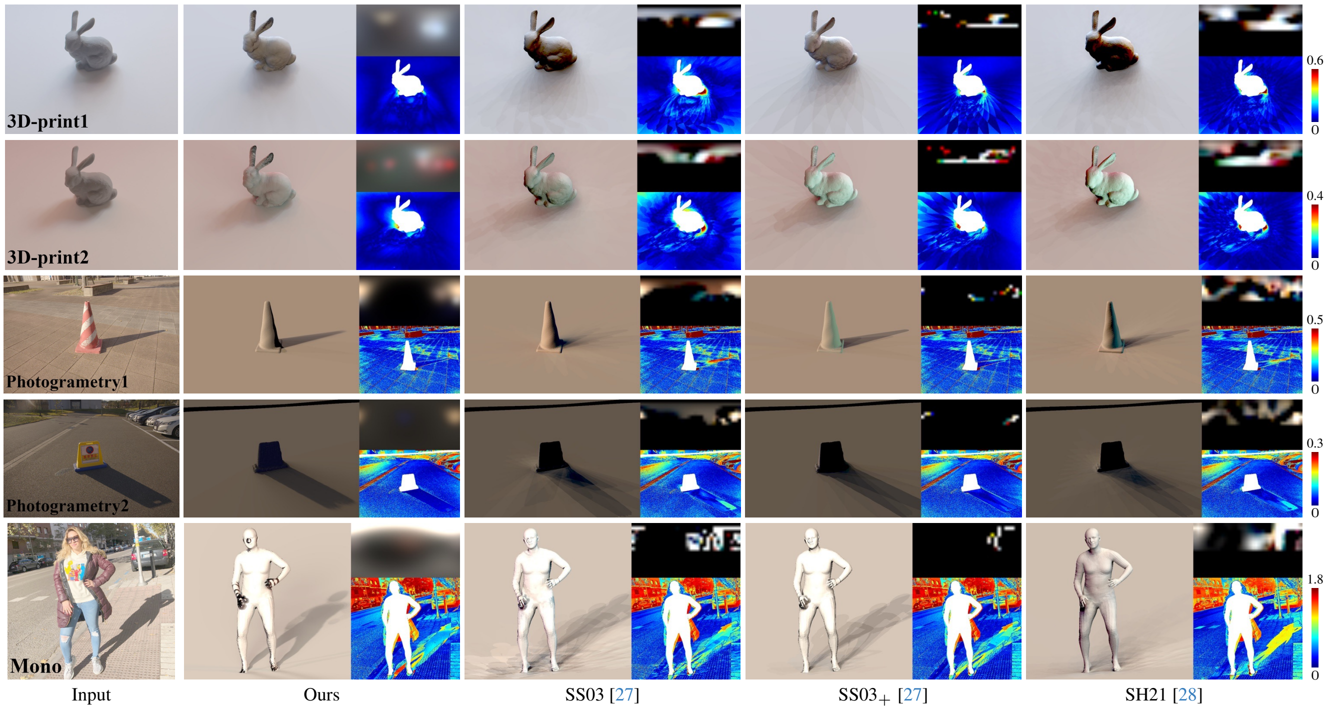

Figure 10. **Qualitative results on re-rendered image and estimated lighting using real-world images.** The layout of each image block follows the format shown in Fig. 6.

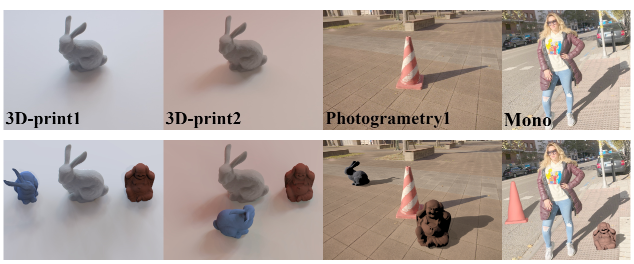

Figure 11. **Virtual object insertion on real-world images. (Top row)** Captured real-world images. **(Bottom row)** Images with blended virtual objects. The estimated surrounding lighting enables rendering photorealistic and coherent shadow effects for inserted virtual objects.

Results and discussions Figure 10 and Table 3 present the qualitative and quantitative results of re-rendered image quality using real-world images. Consistent with the results on synthetic images, our method achieves superior re-rendering quality of the shadow plane. Despite violating the uniform albedo assumption, our method remains effective even when the shadow plane is textured. Further, although our method requires known scene geometry, it can work with a rough shape estimated from a single image, especially when the shadow caster has strong prior information, such as human bodies.

Figure 11 shows the qualitative results of inserting virtual objects into captured real-world images. Our method produces coherent shadow effects for virtually inserted objects in various real-world scenarios. The shadows rendered by our method can be either hard or soft, depending on the lighting conditions. This flexibility ensures they align well with the specific lighting of each scene, leading to a seamless integration of virtual and real objects.

5. Concluding remarks

We have proposed an effective method for estimating surrounding illumination based on the appearance of shadows. Our approach leverages the SG illumination representation and SG-based rendering equation. Despite using fewer parameters, the compactness of SG parameterization allows for more accurate illumination estimation and higher shadow plane re-rendering quality than methods based on discrete illumination representation. Through comprehensive experiments on both synthetic and real-world images across various lighting conditions and scenarios, we have demonstrated the robustness and superiority of our method.

Future work Our results highlight the potential of using cast shadows as a reliable cue for SG illumination estimation. However, it relies on the assumption that most scene components are known, which limits its applicability. In the future, we aim to integrate our method into an inverse rendering pipeline, where all scene components are jointly estimated from multi-view images.

References

[1] Meshlab. https://www.meshlab.net/#description. 7

[2] Agisoft metashape. https://www.agisoft.com. 7

[3] Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 7

[4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. NeRD: Neural reflectance decomposition from image collections. In *ICCV*, pages 12684–12694, 2021. 2

[5] Gareth Bradshaw and Carol O'Sullivan. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics (TOG)*, 23(1):1–26, 2004. 4, 5

[6] Adam Lloyd Cohen. Anti-pinhole imaging. *Optica Acta: International Journal of Optics*, 29(1):63–67, 1982. 1

[7] Mohammad Reza Karimi Dastjerdi, Jonathan Eisenmann, Yannick Hold-Geoffroy, and Jean-François Lalonde. EverLight: Indoor-outdoor editable HDR lighting estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7420–7429, 2023. 1

[8] Charles De Rousiers, Adrien Bousseau, Kartic Subr, Nicolas Holzschuch, and Ravi Ramamoorthi. Real-time rough refraction. In *Symposium on Interactive 3D Graphics and Games*, pages 111–118, 2011. 2

[9] Paul E. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998. 1

[10] Graham D Finlayson, Roshanak Zakizadeh, and Arjan Gijsenij. The reproduction angular error for evaluating the performance of illuminant estimation algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 39(7): 1482–1488, 2016. 5

[11] Wataru Furuya, Kei Iwasaki, Yoshinori Dobashi, and Tomoyuki Nishita. Efficient calculation method of spherical signed distance function for real-time rendering of dynamic scenes. In *SIGGRAPH Asia 2011 Sketches*, pages 1–2. 2011. 2, 4

[12] Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. Frequency domain normal map filtering. In *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*. 2007. 2

[13] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6927–6935, 2019. 1

[14] Takuya Ikeda, Yuji Oyamada, Maki Sugimoto, and Hideo Saito. Illumination estimation from shadow and incomplete object shape captured by an RGB-D camera. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 165–169. IEEE, 2012. 2

[15] Kei Iwasaki, Wataru Furuya, Yoshinori Dobashi, and Tomoyuki Nishita. Real-time rendering of dynamic scenes under all-frequency lighting using integral spherical Gaussian. In *Computer Graphics Forum*, pages 727–734. Wiley Online Library, 2012. 2

[16] Salma Jiddi, Philippe Robert, and Eric Marchand. Estimation of position and intensity of dynamic light sources using cast shadows on textured real surfaces. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1063–1067. IEEE, 2018. 2

[17] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. TensoIR: Tensorial inverse rendering. In *CVPR*, pages 165–174, 2023. 2

[18] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 3

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 5

[20] DP Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Yuanzhen Li, Hanqing Lu, and Heung-Yeung Shum. Multiple-cue illumination estimation in textured scenes. In *ICCV*, pages 1366–1373. IEEE, 2003. 2

[22] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2633–2651, 2016. 7

[23] Takahiro Okabe, Imari Sato, and Yoichi Sato. Spherical harmonics vs. haar wavelets: Basis for recovering illumination from cast shadows. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages I–I. IEEE, 2004. 2

[24] Jeong Joon Park, Aleksander Holynski, and Steven M. Seitz. Seeing the world in a bag of chips. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1414–1424, 2020. 1

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 5

[26] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 7

[27] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):290–300, 2003. 1, 2, 3, 5, 6, 7, 8

[28] Tristan Swedish, Connor Henley, and Ramesh Raskar. Objects as cameras: Estimating high-frequency illumination from shadows. In *ICCV*, pages 2593–2602, 2021. 1, 2, 3, 5, 6, 7, 8

[29] Antonio Torralba and William T Freeman. Accidental pinhole and pinspeck cameras: Revealing the scene outside the picture. *IJCV*, 110:92–112, 2014. 1

[30] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*, 2006. 2

[31] Guangcong Wang, Yinuo Yang, Chen Change Loy, and Ziwei Liu. StyleLight: HDR panorama generation for lighting estimation and editing. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[32] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH Asia)*, 2009. 2, 3, 4

[33] Haoqian Wu, Zhipeng Hu, Lincheng Li, Yongqiang Zhang, Changjie Fan, and Xin Yu. NeFII: Inverse rendering for reflectance decomposition with near-field indirect illumination. In *CVPR*, pages 4295–4304, 2023. 2

[34] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 7

[35] Hong-Xing Yu, Samir Agarwala, Charles Herrmann, Richard Szeliski, Noah Snavely, Jiajun Wu, and Deqing Sun. Accidental light probes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12521–12530, 2023. 1, 5

[36] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 2

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[38] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, pages 18643–18652, 2022. 2