Inverse Rendering - A Comparative Survey

Ong Lee Boon
Department of Computer Science
National Yang Ming Chiao Tung University
shawnong93.cs13@nycu.edu.tw

Lam Kin Ho
Department of Computer Science
National Yang Ming Chiao Tung University
isaaclam.cs13@nycu.edu.tw

Truong Thanh Nguyen
Department of Computer Science
National Yang Ming Chiao Tung University
thanhnguyenzh.ee13@nycu.edu.tw

Abstract

Inverse rendering aims to recover the physical scene properties from images - including geometry, materials, and lighting. This problem has been a longstanding challenge in computer vision and graphics. While traditional optimization-based methods have been extensively studied, the rapid advancement of deep learning techniques has opened new possibilities in this domain. Despite the growing interest and significant progress in neural inverse rendering, there lacks a comprehensive survey of these modern approaches. In this paper, we explore the related literature and unique challenges in inverse rendering and examine how current deep learning approaches address them through various techniques, including Neural Radiance Field, 3D Gaussian Splatting and Diffusion Models. Through our analysis, we identify emerging trends and promising research directions in this evolving field.

1 Introduction

The field of computer vision and graphics has long sought to bridge the gap between 2D image observations and their underlying 3D physical properties. Recent advances in deep learning have sparked a revolution in inverse rendering, introducing multiple competing paradigms that approach this challenge from distinct theoretical foundations.

1.1 Problem definition

The challenge of inverse rendering - reconstructing physical scene properties from images - stands as a fundamental problem in computer vision and graphics. When light interacts with a scene, it undergoes complex interactions with surfaces before reaching the camera sensor. Inverse rendering aims to decompose this process, inferring geometry, materials, and lighting conditions from images. While traditional physics-based approaches laid important groundwork, deep learning has revolutionized how we approach this inverse problem.

1.2 Motivation

With the last comprehensive survey of inverse rendering dating back to 2003 [27], there is a clear need for a modern review of this field. Furthermore, our study of computer vision fundamentals has revealed numerous recent advances - from differentiable rendering to neural radiance fields - that warrant systematic analysis. This motivates us to synthesize these developments and their applications in solving the inverse rendering problem.

Preprint. Under review.

1.3 Recent Advances in Deep Learning Approaches

Neural Radiance Field (NeRF)-based NeRF [26] have emerged as a powerful paradigm for scene representation and rendering. By encoding scene properties in continuous neural fields, NeRF-based methods can capture complex geometries and appearance variations. Recent extensions to inverse rendering have demonstrated promising results in decomposing scene properties. These methods have branched into several key directions: view synthesis and relighting [30], reflectance decomposition from image collections [3, 22], joint estimation of shape and material properties [36, 33], and full inverse rendering from arbitrary real-world images [5]. While each direction has made significant progress in their respective focus areas, common challenges persist across these approaches, particularly in handling indirect illumination and achieving reliable decoupling of material properties from lighting effects.

3D Gaussian Splatting (3DGS)-based 3DGS [21] represents a significant advancement in efficient scene representation, spawning various approaches to inverse rendering despite inherent geometric limitations. These methods have developed along several directions: direct BRDF decomposition and ray tracing for relighting [12], modeling of reflective surfaces through specialized shading functions [16], full inverse rendering with material and lighting estimation [23, 29], and advanced light transport simulation including subsurface scattering [10]. Some recent works like GeoSplatting [34] attempt to address the fundamental geometric limitations through additional geometric guidance. While these methods leverage the efficiency and real-time capabilities of 3DGS, they face common challenges in achieving robust geometry estimation, which is crucial for accurate shadow-material decoupling and high-quality inverse rendering.

Probabilistic Prior-based Diffusion Models (DM) [15, 28] have recently revolutionized various aspects of computer vision, including inverse rendering. These approaches branch into several directions: joint recovery of illumination and reflectance from single images [11], handling illumination uncertainty through posterior sampling [25], inverse rendering for arbitrary objects under unknown lighting ([7], and specialized material decomposition [24]. By leveraging learned priors from large-scale data, these methods excel at resolving inherent scene ambiguities, though at the cost of increased computational complexity.

To better organize the landscape of modern inverse rendering approaches, Figure 1 presents a taxonomy of recent methods categorized by their focus areas and underlying technical approaches. The methods are color-coded to indicate their foundational techniques: NeRF-based methods (red), 3D Gaussian Splatting-based methods (green), and diffusion-prior-based methods (blue).

1.4 Contributions

This study presents a focused comparative analysis of three influential deep learning paradigms in inverse rendering: Neural Radiance Fields (NeRF)-based methods, 3D Gaussian Splatting (3DGS)-based approaches, and Diffusion Model (DM)-based techniques. Given the rapid development in each direction, we carefully select seminal works that best represent the key innovations and approaches within each paradigm. Through this curated analysis, we examine how these methods have evolved to address inverse rendering challenges, how they are compared and how they should be further studied.

1.5 Paper Organization

The remainder of this paper is organized as follows: Section 2 provides essential background on inverse rendering fundamentals and theoretical foundations of the three deep learning paradigms. Sections 3, 4, and 5 present detailed analyses of neural field representations, point-based decomposition methods, and probabilistic diffusion approaches, respectively. Section 6 offers a comprehensive discussion of their comparative advantages and practical applications. We conclude with our findings and nudge promising future research directions in Section 7.

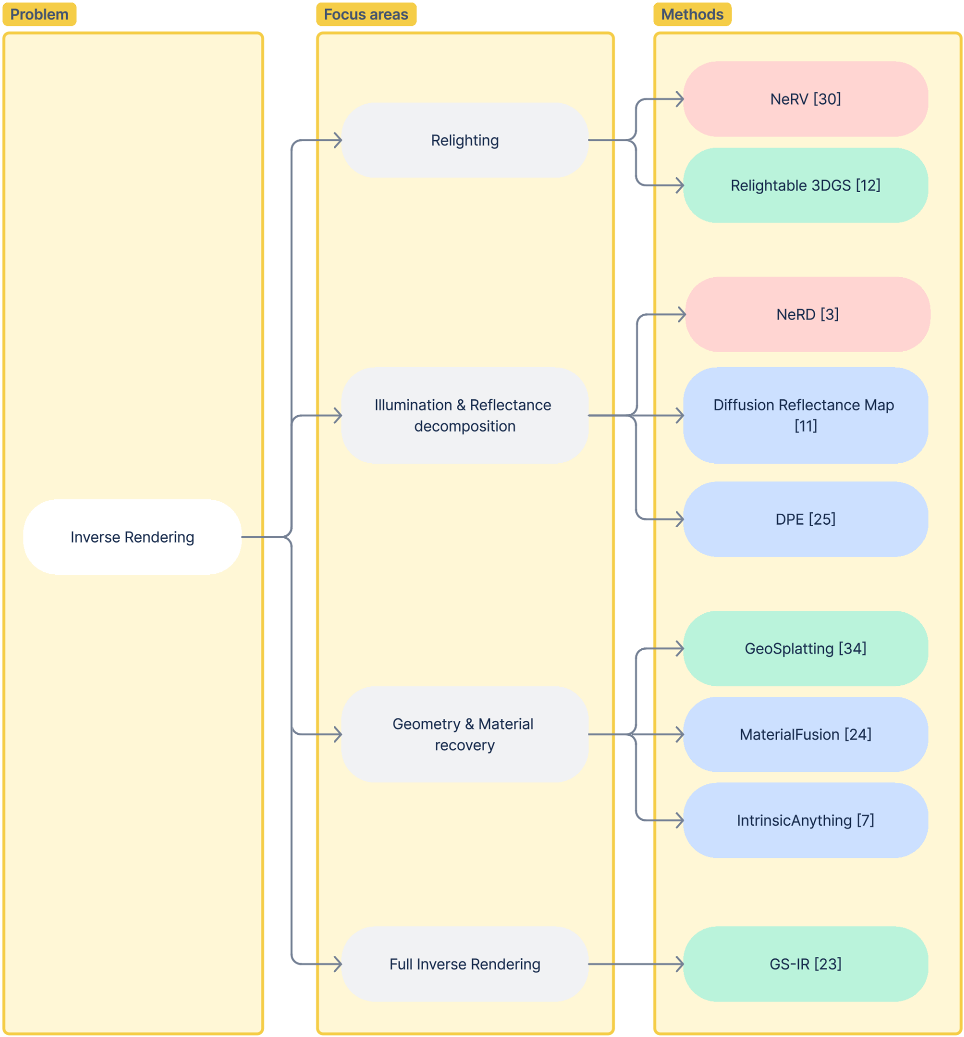

Figure 1: Overview of our selected modern inverse rendering approaches. Methods are organized by their primary focus areas (middle column) and colored by their underlying technical foundations: NeRF-based methods (red), 3D Gaussian Splatting-based methods (green), and diffusion-prior-based methods (blue). The taxonomy shows how different approaches have been applied to *increasingly complex* aspects of the inverse rendering problem, from relighting to full inverse rendering pipelines.

2 Background

2.1 Inverse Rendering

Inverse rendering aims to recover scene properties (geometry, materials, and illumination) from images by inverting the physics-based image formation process. This section covers the key concepts and models underlying modern inverse rendering approaches.

2.1.1 Image Formation Model

The rendering equation [19] describes how outgoing radiance at a surface point results from incoming light interacting with material properties:

$$L_o(x, \omega_o) = \int_\Omega L_i(x, \omega_i) f_r(x, \omega_i, \omega_o)(\omega_i \cdot n)\, d\omega_i \tag{1}$$

where:

- $L_o(x, \omega_o)$ is the outgoing radiance at point x in direction ω_o
- $L_i(x, \omega_i)$ is the incident radiance from direction ω_i
- $f_r(x, \omega_i, \omega_o)$ is the Bidirectional Reflectance Distribution Function (BRDF)
- n is the surface normal
- Ω is the hemisphere around the surface normal

2.1.2 Bidirectional Reflectance Distribution Function

The Bidirectional Reflectance Distribution Function (BRDF) is fundamental to physically based rendering, describing how light reflects off surfaces. Formally, it defines the ratio of reflected radiance to incident irradiance:

$$f_r(\omega_i, \omega_o) = \frac{dL_o(\omega_o)}{dE_i(\omega_i)} = \frac{dL_o(\omega_o)}{L_i(\omega_i)\cos\theta_i\, d\omega_i} \tag{2}$$

where ω_i and ω_o are incoming and outgoing light directions respectively, and θ_i is the angle between ω_i and the surface normal. The foundational physically-based BRDF model was introduced by Cook and Torrance [9], which established the microfacet theory widely used today. Their model represents specular reflection using three key terms: normal distribution (D), geometric attenuation (G), and Fresnel reflection (F).
A physically plausible BRDF must satisfy three key properties:

- Non-negativity: $f_r(\omega_i, \omega_o) \geq 0$
- Reciprocity: $f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i)$
- Energy conservation: $\int_\Omega f_r(\omega_i, \omega_o)\cos\theta_i, d\omega_i \leq 1$

Modern inverse rendering commonly uses the Disney Principled BRDF model [6], which builds upon these foundations while providing artist-friendly parameterization:

$$f_r = (1-\mu)\frac{\rho_d}{\pi}(f_{\text{diff}} + f_{\text{retro}}) + f_{\text{spec}} \tag{3}$$

where:

- μ is metalness (0 for dielectrics, 1 for metals)
- ρ_d is diffuse albedo
- f_{diff} is diffuse reflection
- f_{retro} is retro-reflection
- f_{spec} is specular reflection based on the GGX microfacet distribution

This model provides an intuitive parameterization while maintaining physical plausibility, making it particularly suitable for inverse rendering tasks. The majority of methods covered in subsequent sections adopt variants of this model for their material representations.

2.1.3 Material Models

Most methods adopt the Cook-Torrance microfacet BRDF model [9], which separates reflection into diffuse and specular components:

$$f_r = \frac{\text{diffuse}}{\pi} + \frac{DFG}{4(n \cdot \omega_i)(n \cdot \omega_o)} \tag{4}$$

where D (normal distribution), F (Fresnel term), and G (geometric shadowing) model different aspects of specular reflection. Modern approaches like the Disney Principled BRDF parameterize this using intuitive parameters:

- Basecolor: Overall surface color
- Metallic: Interpolation between dielectric and metallic behavior
- Roughness: Surface microscale roughness affecting specular reflection
- Normal: Local surface orientation

2.1.4 Illumination Representation

Environmental illumination is typically represented in one of two ways:

- Environment maps: HDR spherical images capturing incident light from all directions
- Spherical Gaussians (SG): Mixture of Gaussian lobes on the sphere, enabling efficient rendering:

$$L_i(\omega) = \sum_{k=1}^{K} \lambda_k G(\omega; \xi_k, \eta_k) \tag{5}$$

where λ_k, ξ_k, and η_k are the amplitude, axis, and bandwidth of each lobe.

2.1.5 Inverse Problem Challenges

The inverse rendering problem is inherently ill-posed due to several factors:

- Color ambiguity: Same appearance can arise from different combinations of material and lighting
- Frequency loss: Material roughness acts as a low-pass filter on illumination
- Missing information: Self-occlusions and limited viewpoints create unobserved regions
- Global illumination: Indirect light paths create complex interdependencies

Modern deep learning approaches address these challenges through:

- Neural scene representations that implicitly handle geometry and visibility
- Learning-based priors on natural materials and illumination
- End-to-end optimization frameworks incorporating physics-based rendering
- Explicit modeling of uncertainty in the reconstruction process

This provides foundational context for understanding the different technical approaches we analyze in subsequent sections.

2.2 Neural Radiance Field

NeRF, was proposed in [26] for Novel View Synthesis (NVS), by optimizing a radiance field from multi-view images of a scene. This can be achieved by using MLPs to learn the geometry and lighting of a 3D scene, and then perform volume rendering to render images for supervision. In NeRF, a scene is represented as 3D points, which are implicitly saved inside MLPs, Specifically, the color $\boldsymbol{c} = (r, g, b)$ and density σ can be queried using 3D position $\boldsymbol{x} = (x, y, z)$ and viewing directions θ, ϕ

$$F(\boldsymbol{x}, \theta, \phi) \rightarrow (\boldsymbol{c}, \sigma), \tag{6}$$

where F is implemented as a 8-layers MLP. Afterwards, volume rendering is performed to get the rendered image by shooting rays from every pixel to the 3D spaces to query the color $\boldsymbol{c}$ and density σ of each point $\boldsymbol{x}$. To reduce the computation costs, instead of taking account of all points on the ray, we perform sampling to consider only a number of discrete points. Formally, the color $\hat{C}$ for every pixels can be obtained by

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right), \tag{7}$$

where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. To further improve the sampling efficiency, they employ a coarse to fine sampling strategy using 2 different MLPs. Finally, to memorize the scene, NeRF model is supervised using the MSE loss between the rendered images and the corresponding ground-truth images

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\|\hat{C}_c(\mathbf{r}) - C(\mathbf{r})\|_2^2 + \|\hat{C}_f(\mathbf{r}) - C(\mathbf{r})\|_2^2\right], \tag{8}$$

where $\mathcal{R}$ is the set of rays in each batch, and $C(\mathbf{r})$, $\hat{C}_c(\mathbf{r})$, and $\hat{C}_f(\mathbf{r})$ are the ground truth, coarse volume predicted, and fine volume predicted RGB colors for ray r respectively.

NeRF has been shown to achieve high fidelity NVS, however their demands to query the MLPs with every pixels is extremely computationally expensive, which makes them less appealing to real-time systems. Ever since its proposal, many works have been done to improve the quality and inference speed of NeRF, opening valuable directions for NeRF-based applications.

2.3 3D Gaussian Splatting

Distinct from NeRF which describes the scene implicitly by querying MLPs, 3D Gaussian Splatting (3DGS) [21] employs explicit 3D Gaussian points as its rendering primitives. A 3D Gaussian point is mathematically defined as:

$$G(\boldsymbol{x}) = exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})), \tag{9}$$

where $\boldsymbol{\mu}$ and Σ denote the 3D spatial mean and covariance matrix, respectively. Each Gaussian is also equipped with an opacity o and a view-dependent color $\boldsymbol{c}$. Instead of using expensive ray marching in NeRF, volume rendering in 3D-GS are performed using splatting techniques to render images. Specifically, the Gaussians are first depth-sorted using a tile-based fast GPU sort, and then the color is derived by alpha blending the sorted 2D Gaussians from front to back to form the final color.

$$\mathcal{C} = \sum_{i\in N} T_i\alpha_i\boldsymbol{c}_i \tag{10}$$

with

$$T_i = \prod_{j=1}^{i-1}(1 - \alpha_j), \tag{11}$$

$$\alpha_i = (1 - exp(-o_i\delta_i)) \tag{12}$$

where T is the light transmittance, o is the density of the Gaussian and δ is the sampling interval over the ray.

While the covariance matrix Σ of 3D Gaussians can be directly optimized, this approach may lead to cases when Σ is not positive semi-definite, which is physically meaningless. Thus in implementation, Σ is decomposed into rotation matrix R and scaling matrix S as follows:

$$\Sigma = RSS^TR^T, \tag{13}$$

where R is saved as a quaternion q and S is saved as a 3D vector s. Therefore a Gaussian can be parameterized as $\{\boldsymbol{\mu}, \boldsymbol{q}, \boldsymbol{s}, o, \boldsymbol{c}\}$.

To optimize the number of 3D points, the author introduced an adaptive density control mechanism for 3D Gaussians. During the optimization process, the opacity and size of the initialized 3D points are dynamically adjusted to better represent the scene. Gaussians that are too large are subdivided into smaller ones, while those with low opacity (deemed insignificant) are pruned.

2.4 Diffusion Prior

Compared to the above, diffusion priors are a key concept in the field of generative modeling, particularly in diffusion models, which are a class of generative models used to solve nonlinear inverse problems. These models have gained significant attention for their ability to capture data distributions and to generate high-quality realistic images and other data types, conditioned on various inputs, such as text, images, or bounding boxes, balancing the intricate triangle between accuracy, naturalness, and ambiguity.

It typically involves a forward process (adding noise) and a reverse process (removing noise). For most models such as generative adversarial networks (GANs), the optimization process tends to get stuck in local optima, leading to the artifacts on the recovered material. But for diffusion prior, by adding noise to the data in the forward process, the model learns to escape narrow, local minima solutions.

Denoising diffusion probabilistic models (DDPMs) are good demonstration on generation process as a Markov chain of denoising steps. Formally, given the clean data $x_0 \sim q(x_0)$, addition of noise to the data can be described well as a Gaussian transition as below.

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{14}$$

where $0 < \beta_1 < \beta_2 < \cdots < \beta_T = 1$ are the fixed variance schedule. At the largest time step T, the noised data is transformed to a standard Gaussian distribution.

In the reverse process, the model does not rely solely on a deterministic optimization path. Instead, it learns how to gradually denoise the corrupted data. The denoising process inherently involves some stochasticity because the model needs to 'guess' how to reverse the noise at each step. Each step of the denoising process from x_t to x_{t-1} is defined as below

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\hat{\alpha}_t}}\varepsilon_\theta(x_t, t)) + \sigma_t z \tag{15}$$

where ε_θ is the learnable noise predictor, α_t and σ_t are predefined constant values, and z is a random value sampled from a standard Gaussian distribution.

Below is a general rendering equation on object material and lighting from images. For a 3D point x, its outgoing light $\mathcal{L}_0$ at direction ω_0 is defined as:

$$\mathcal{L}_0(\omega_0; x) = \int_\Omega \mathcal{L}_i(\omega_i) f_r(x; \omega_0, \omega_i)(\omega_i \cdot n) d\omega_i \tag{16}$$

where Ω is the hemisphere centered on the surface normal n, $f_r(x; \omega_0, \omega_i) = f_r(\omega_0, \omega_i)$ if a surface with homogeneous material properties is assumed. f_r is the bidirectional reflectance distribution function (BRDF) [6].

2.5 Datasets

Inverse rendering methods have been evaluated on a diverse range of datasets, with many papers introducing their own custom datasets. Here we highlight some widely used benchmark datasets:

Synthetic datasets

- NeRF Synthetic dataset [26] for fixed illumination evaluation
- TensoIR Synthetic dataset [17] with ground truth BRDF parameters and environment maps
- Synthetic4Relight dataset [37] featuring 4 challenging scenes with complex materials and indirect lighting effects
- Common test scenes (Globe, Car Wreck, Chair) rendered under varied illumination

Real-world Datasets

- MipNeRF-360 dataset [2] for real uncontrolled scene captures

- nLMVS-Real dataset [32] with objects under 6 different illumination environments
- Real captures under varying natural illumination (e.g., Gnome, MotherChild scenes)

Note that this list represents commonly used benchmarks in the field, but many methods also employ custom datasets tailored to their specific objectives or technical requirements. The diversity of datasets reflects both the complexity of inverse rendering and the different focus areas of various approaches.

2.6 Quality Assessment Methods

The evaluation of inverse rendering methods employs various metrics depending on the specific aspects being assessed. Below are commonly used metrics, though methods often introduce additional task-specific evaluation criteria:

Image Quality

- PSNR and SSIM [31] for overall reconstruction accuracy
- LPIPS [35] for perceptual similarity to ground truth

Besides, some methods also reports inverse-rendering-oriented metrics:

- FID scores [14] for environment map naturalness
- PU21-NIQE [13] for HDR image quality assessment

Material Recovery

- Log-scale RMSE for BRDF parameter estimation
- Material-specific metrics for basecolor, metallic, and roughness
- Relighting error under novel illumination
- Cross-view material consistency

Illumination Estimation

- Direct comparisons with ground truth environment maps
- Relighting accuracy on test objects
- Evaluation of lighting detail preservation
- Shadow and indirect illumination accuracy

Additionally, many methods [12, 23, 16, 8, 34, 11] report computational metrics such as training time and inference speed. The choice of evaluation metrics often depends on the specific objectives of each method and the availability of ground truth data.

3 Neural Radiance Fields for Inverse Rendering

3.1 NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis

The first major advancement NeRV is developed by many of the same researchers behind the NeRF papers. The key improvement of NeRV over NeRF is its ability to handle relighting, whereas NeRF is focused solely on view synthesis.

NeRF generates results based on the outgoing light from a given location, which is a combination of multiple incident lights interacting with the object's material properties. In contrast, NeRV takes a different approach. It assumes that environmental light sources are known and, instead of outputting density fields and view colors, it directly predicts surface normals and material properties. These material properties can then be used as parameters in a Bi-Directional Reflectance Distribution Function (BRDF), which models how light interacts with surfaces.

However, directly applying this method would lead to a significant increase in computational complexity—beyond cubic time—which would make it impractical for real-time applications. To address this, the authors propose a visible field cloud technique, which reduces the complexity to linear time, making the approach much more efficient and feasible for practical use.

In NeRF, the observed radiance $L(\mathbf{c}, \boldsymbol{\omega}_o)$ at camera location $\mathbf{c}$ along direction $\boldsymbol{\omega_o}$ is computed by standard emission-absorption volume rendering [20] as the integral of the product of three quantities at any point $\mathbf{x}(t) = \mathbf{c} - t\boldsymbol{\omega_o}$ along the ray: the visibility $V(\mathbf{x}(t), \mathbf{c})$, which is the proportion of incident light towards camera at$\mathbf{c}$ from the emitted light at $\mathbf{x}(t)$., the density $\sigma(\mathbf{x}(t))$, and the emitted radiance $L_e(\mathbf{x}(t), \boldsymbol{\omega_o}$ along the viewing direction $\boldsymbol{\omega_o}$:

$$L(\mathbf{c}, \boldsymbol{\omega_o}) = \int_0^\infty V(\mathbf{x}(t), \mathbf{c})\sigma(\mathbf{x}(t))L_e(\mathbf{x}(t), \boldsymbol{\omega_o})dt \tag{17}$$

$$V(\mathbf{x}(t), \mathbf{c}) = \exp\left(-\int_0^t \sigma(\mathbf{x}(s))ds\right) \tag{18}$$

In NeRV, reflection light is considered over emission light, therefore the formula becomes:

$$L(\mathbf{c}, \boldsymbol{\omega_o}) = \int_0^\infty V(\mathbf{x}(t), \mathbf{c})\sigma(\mathbf{x}(t))L_r(\mathbf{x}(t), \boldsymbol{\omega_o})dt \tag{19}$$

$$L_r(\mathbf{x}(t), \boldsymbol{\omega_o}) = \int_{\mathcal{S}} L_i(\mathbf{x}, \boldsymbol{\omega_i})R(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)d\boldsymbol{\omega}_i \tag{20}$$

Where it is now an integral over the sphere $\mathcal{S}$ of the product of any incoming radiance L_i and the reflectance function function R. The reflectance function is also known as phase function in volume rendering that represent the quantities of light coming from direction $\boldsymbol{\omega}_i$ is reflected towards direction $\boldsymbol{\omega_o}$. In NeRV, the model is now output 3D diffuse albedo $\mathbf{a}$, and 1D roughnessγ for BRDF on top of the only density σ in NeRF.

By considering the reflected light and the physical properties of the surface material, NeRV enables the rendering of images under varying lighting conditions, producing a wide range of possible results. In contrast to NeRF, which requires the lighting conditions to remain consistent across all views, NeRV allows for synthesized images that can vary according to different lighting scenarios. However, it still assumes that the lighting conditions are known in advance.

3.2 NeRD: Neural Reflectance Decomposition from Image Collections

Another notable approach, NeRD [3] was introduced in the same year. NeRD builds upon NeRF [26] by allowing the model to take multiple images with different lighting conditions as input. Like NeRV [30], it learns independent reflectance parameters, enabling more flexibility in the rendering process. The improvement of NeRD come from the modifications of network architecture, sampling network and decomposition network.

The Sampling Network in NeRD introduces a crucial enhancement over previous inverse rendering methods like NeRF by employing an adaptive sampling strategy. Instead of uniformly sampling 3D points along rays as in NeRF, where the number of samples is fixed and evenly distributed, NeRD adapts its sampling density based on the complexity and information content of the scene. This adaptive approach allows the network to allocate more computational resources to areas of the scene that exhibit high detail, such as surfaces, boundaries, or regions with complex lighting or material properties, while reducing the number of samples in areas with little variation, like empty space or homogeneous regions.

This selective sampling not only improves the efficiency of the rendering process by lowering computational overhead, but it also enhances the precision of scene reconstruction, particularly in regions with occlusions, fine geometry, or material transitions. By focusing on the most informative regions, NeRD refines ray integration, ensuring that critical features are captured with higher accuracy while reducing unnecessary computation in less complex areas. Ultimately, this adaptive sampling mechanism allows for faster, more efficient rendering without compromising on the quality of the output, making it a significant improvement over earlier methods that used uniform sampling strategies.

The Decomposition Network in NeRD improves scene representation by breaking down the scene into distinct components, each handled by a specialized sub-network. This decoupling leads to more efficient and accurate scene modeling compared to NeRF's monolithic approach. The key sub-networks in NeRD are:

1. Geometry Sub-network: Using voxel grids or point clouds, to model the shape and structure of the scene.
2. Material Sub-network: The material properties such as reflectance and absorption are learned to capture how materials interact with light.
3. Lighting Sub-network: Achieving photorealistic rendering by modeling view-dependent illumination, reflections, and shadows.

This decomposition allows NeRD to independently refine each component (geometry, material, and lighting), resulting in a more detailed and flexible scene representation. By separately optimizing these factors, NeRD improves its ability to handle complex lighting conditions, diverse material types, and intricate geometries, leading to higher-quality and more realistic renderings.

During training, the scene is decomposed into multiple feature maps, each corresponding to one of these components. Each sub-network learns its respective feature map, and these are later combined during rendering using techniques like volume rendering. By treating each factors separately, NeRD can better capture the intricacies of real-world environments and achieve a more detailed, flexible, and accurate representation of complex scenes. The result is a more realistic and robust, especially for scenes with complex interactions between light, materials, and geometry.

3.3 Summary

While NeRV focuses on improving visibility and light transport, NeRD decomposes the scenes into multiple properties and learns the feature separately. However, both approaches share the ability to reconstruct 3D models under different lighting conditions using only 2D images. While the NeRF itselfs can only render the model for same lighting and not adjustable. There is still a limitation of these improved methods that assume that the geometry and lighting condition are given. Moreover, multiple images of the same scene are required for the model to learn the pathway of the light ray.

4 Point-Based Methods for Inverse Rendering

While Neural Radiance Fields have demonstrated impressive results in inverse rendering, their computational complexity and real-time performance limitations have motivated the exploration of point-based approaches, particularly those built upon 3D Gaussian Splatting (3DGS). This section examines three key directions in 3DGS-based inverse rendering methods.

4.1 BRDF Decomposition with Ray Tracing

The first significant advancement in adapting 3DGS [21] for inverse rendering came with Relightable 3D Gaussians (R3DG) [12]. This seminal work extends the basic 3DGS framework by incorporating physically-based rendering (PBR) properties into each Gaussian primitive. Following the rendering equation, for a 3D point x and viewing direction v, its outgoing radiance is defined as:

$$L_o(x, v) = \int_{\Omega} L_i(x, l) f_r(l, v)(l \cdot n) dl \tag{21}$$

where L_i is the incident light from direction l, n is the surface normal, and f_r is the BRDF modeled using the Cook-Torrance microfacet model:

$$f_r(l, v) = (1 - m)\frac{a}{\pi} + \frac{DFG}{4(n \cdot l)(n \cdot v)} \tag{22}$$

Here m is the metallic parameter, a is the albedo, and D, F, G represent the distribution, Fresnel, and geometry terms respectively.

R3DG introduces efficient point-based ray tracing using Bounding Volume Hierarchy (BVH) for visibility computation. The incident light is split into direct illumination via an environment map and indirect illumination modeled through spherical harmonics. This combination enables real-time relighting while maintaining the rendering efficiency of 3DGS.

4.2 Full Material and Lighting Decomposition

GS-IR [23] advances the capabilities of 3DGS-based inverse rendering by introducing a comprehensive framework for material and lighting estimation. The method consists of three well-designed stages:

1. Depth-guided normal estimation that leverages rendered depth maps to regularize normal prediction
2. A baking-based occlusion method for efficient indirect lighting computation
3. Split-sum approximation for physically-based rendering, enabling real-time material editing

A key innovation in GS-IR is the efficient occlusion and indirect illumination handling. Instead of expensive ray tracing, the method introduces a baking-based spherical harmonics representation. For each volume grid point i, the occlusion is expressed as:

$$O(\theta, \phi) = \sum_{l=0}^{deg} \sum_{m=-l}^{l} f^{o}_{i(lm)} Y_{lm}(\theta, \phi) \tag{23}$$

where Y_{lm} are spherical harmonics bases and $f^{o}_{i(lm)}$ are the cached coefficients. This enables efficient modeling of illumination:

$$I_d(x) = (1 - O(x))I_d^{dir}(x) + O(x)I_d^{indir}(x) \tag{24}$$

where I_d^{dir} and I_d^{indir} represent direct and indirect illumination components. Combined with split-sum approximation for PBR, this enables real-time material-lighting decomposition.

4.3 Geometry-Guided Optimization

A fundamental limitation of 3DGS-based inverse rendering has been the accuracy of geometric reconstruction, which directly impacts the quality of material-lighting decomposition. GeoSplatting [34] addresses this challenge by introducing explicit geometric guidance through:

1. A hybrid representation bridging isosurface and 3D Gaussians
2. Differentiable mesh-to-Gaussian conversion via their proposed MGAdapter
3. End-to-end optimization that maintains both geometric accuracy and rendering efficiency

The key component is the MGAdapter T that maps mesh geometry M to Gaussian parameters:

$$(\mu, S, R, n) = T(M) \tag{25}$$

For each Gaussian, its position μ is computed through barycentric interpolation of mesh vertices:

$$\mu = \sum_{i=1}^{3} w_i p_i \tag{26}$$

where w_i are barycentric weights and p_i are vertex positions. The scale S and rotation R are derived to align with surface geometry, while normal n inherits from the mesh. This explicit geometric guidance leads to more accurate normal estimation and improved material-lighting decomposition.

The method demonstrates that improved geometry estimation leads to better inverse rendering results, particularly in cases involving complex surface interactions and detailed normal maps.

4.4 Summary

These three directions represent key milestones in 3DGS-based inverse rendering, each addressing different aspects of the challenge. R3DG pioneered the integration of physically-based rendering with 3D Gaussians, while GS-IR demonstrated efficient material-lighting decomposition without ray tracing. GeoSplatting tackled the fundamental geometry limitation through explicit surface guidance. Together, they showcase how point-based methods can achieve high-quality inverse rendering while maintaining the real-time performance characteristic of 3DGS. However, challenges remain in balancing geometric accuracy, rendering efficiency, and decomposition quality.

5 Probabilistic Inverse Rendering Approaches via Diffusion Models

With increasing popularity on generative models and significant improvement on generative AI, diffusion prior, as part of diffusion model, has opened up tons of possibilities on recovery of illumination, materials, lighting, albedo and specular shading etc. Most discussions and researches are focused on decomposition of general rendering equation and also guided prior in the denoising process to have a more desirable result in term of visual appearance and evaluation metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

5.1 Decomposition of BRDF and 2-Stage Training

IntrinsicAnything [7] splits the BRDF into diffuse and specular shading terms, and thus formulates the material prior as diffusion models of albedo and specular. The key idea here is to learn the distribution of albedo and specular shading with generative models, which are then used to regularize the optimization process. By Disney BRDF model, f_r is defined as below, where D, G, F mean the normal distribution, geometry attenuation and Fresnel effect. k_d is object material albedo while ρ is roughness.

$$f_r(x; \omega_0, \omega_i) = \frac{k_d}{\pi} + \frac{D \cdot G \cdot F}{4(n \cdot \omega_0)(n \cdot \omega_i)} \tag{27}$$

$$L_0(\omega_0; x) = S_{diff}(x, L_i, k_d) + S_{spec}(\omega_0; x, \rho, L_i) \tag{28}$$

In addition, a coarse-to-fine 2-stage training strategy is developed to leverage estimated materials to guide diffusion models to satisfy multi-view consistent constraints, leading to more stable and accurate results.

1. The model is supervised by images and diffusion priors to output coarse albedo and roughness. The coarse materials are then used to guide diffusion models to provide more multi-view consistent constraints.
2. The guided sampling first calculates the L2 loss between the guidance and one-step denoised signals, and then adds the gradient of the L1 loss to the output of the noise predictor.

5.2 Illumination Uncertainty Handling Through Posterior Sampling

DPE [25] investigates the use of diffusion posterior sampling (DPS) as a prior for inverse rendering on indoor and outdoor dataset. To this end, DDPM is trained on natural environment maps and then integrated into an optimization framework involving a differentiable path tracer. Crucially, DPS is extended to incorporate a measurement function with trainable parameters. The novel optimization scheme allows sampling from combinations of illumination and spatially-varying materials that are natural, diverse, and explain the image observations. Rendering function is inferred from global light transport [18] under distant illumination formulated as below. x_0 is a natural environment map and k_{arm} means spatially-varying surface materials

$$\mathcal{L}(p, \omega_0) = \int_{\Omega_+} x_0(\omega_i) T(p, \omega_0, \omega_i, k_{arm})(\omega_i \cdot n) d\omega_i \tag{29}$$

$T(p, \omega_0, \omega_i, k_{arm})$ is the radiance transfer function, describing how distant illumination from direction ω_i is scattered through the scene (via potentially many bounces) to finally leave at p in direction ω_0. The paper first pre-trains a DDPM that generates realistic environment maps unconditionally. Then, given input images and geometry, a series of denoising processes is set up. In every time step t , the differentiable path tracer takes materials and the posterior estimation of the clean environment map $\hat{x_t}$ as input and produces a rendered image. The gradient from the rendering loss is used to optimize materials and gets incorporated into a posterior score function that enforces the DDPM to generate a natural environment map that faithfully explains the input images. Hence, reasonable results are expected only if the distribution of environment maps captured by generative models matches the task.

5.3 Stochastic Recovery of Illumination and Reflectance Through Single Image

Diffusion Reflectance Map (DRM) [11] introduces the first single-image stochastic inverse rendering method, a principled approach for recovering the attenuated frequency spectrum of the illumination and reflectance of an object of known geometry from a single image by seamlessly integrating a neural generative process in inverse rendering. The key idea is to learn to generate the illumination from the object appearance with a diffusion model on the reflectance map, an appearance representation invariant to the underlying geometry, by learning to reverse the image formation with a novel diffusion model which is referred to as the Diffusion Reflectance Map Network (DRMNet), which consists of two subnetworks.

The first is a diffusion model, IllNet, a U-Net that stochastically converts the current reflectance map into that of a sharper reflectance until it becomes perfectly mirror.. The second is a reflectance estimator, RefNet, a simple network that takes in the observed and current reflectance maps and the current time-step and estimates the object reflectance as a parametric model.

The inverse generative process can be considered as steps taken along the reflectance space towards a perfect mirror.

$$f_r(\omega_i', \omega_0'; \Psi := \{\rho_d, \rho_s, \alpha, \gamma\}) = (1-\gamma)\frac{\rho_d}{\pi}(f_{diff}(\omega_i', \omega_0') + f_{retro}(\omega_i', \omega_0'; \alpha)) + f_{spec}(\omega_i', \omega_0'; \rho_d, \rho_s, \alpha, \gamma) \tag{30}$$

$\rho_d, \rho_s, \alpha, \gamma$ mean the diffuse color, specular strength, roughness and metallicness respectively. All of the parameters are normalized to $[0, 1]$.

Given the single image of an object whose geometry as surface normals are known, we can compute its reflectance map by mapping each pixel onto a Gaussian hemisphere, i.e. a hemisphere parameterized by the surface normals with its north pole pointing towards the viewpoint. Each pixel p gives a pair of surface radiance L_p and surface normal n_p related.

$$L_p = L_r(x_p, R_{n_p}, w_{o,p}) \tag{31}$$

where x_p is the surface point of pixel p. We assume orthographic projection and use the camera coordinate frame so that $w_{o,p} = [0, 0, 1]$ and a homogeneous surface $L_p = L_r(n_p)$. The reflectance map $L_r(n_p)$ can thus be computed by extracting the surface normal of each image pixel.

5.4 3D Inverse Rendering Approach with 2D Material Diffusion Prior

MaterialFusion [24]seeks to address the intrinsic challenge of disentangling albedo and material properties from input images because the recovered components often fail to render accurately under new lighting conditions. The paper introduces MaterialFusion, an enhanced conventional 3D inverse rendering pipeline that incorporates StableMaterial, a 2D diffusion prior on texture and material properties, that refines multilit data to estimate the most likely albedo and material from given input appearances. MaterialFusion reconstructs an object's geometry, Physically Based Rendering (PBR) materials, and environmental illumination from a set of multi-view images under a fixed lighting condition. In addition to the reconstruction and regularization losses computed between our rendered images $\hat{x}$ and reference RGB images x, MaterialFusion employs priors from our pre-trained StableMaterial to enhance PBR material reconstruction. Specifically, it calculates an score distillation sampling (SDS) loss for the rendered albedo and ORM components, $\hat{I_d}$ and $I_{\hat{orm}}$ conditioned on x. Finally, the total loss consists of the reconstruction loss, regularization loss, and SDS loss, where γ_i is an iteration dependent hyperparameter which decays as training progresses.

$$\mathcal{L}_{MaterialFusion} = \mathcal{L}_{recon} + \mathcal{L}_{reg} + \gamma_i \mathcal{L}_{SDS+} \tag{32}$$

This paper introduces another approach for 3D reconstruction by taking a 2D diffusion prior as guided prior on texture and material properties in the generative tasks.

5.5 Summary

All mentioned papers use diffusion prior approach to tackle the inverse rendering problem. Most works are performed in decomposition of BRDF rendering problem into respective concern, such as material, texture, lighting and illumination etc. Each achieves superior result compared to traditional inverse rendering solutions, but geometry must be given in advance for each experiment. Data distribution of environment maps captured by generative models should also matched distribution of input to produce desirable output. The training process is also computationally expensive and lack interpretability.

6 Discussion

This section analyzes the current state of the field, compares different methodological approaches along with their strengths and weaknesses. We begin by examining the key technical contributions and limitations of recent methods, followed by a detailed comparison of their capabilities and trade-offs.

6.1 Method Comparison and Analysis

Recent years have seen numerous approaches to inverse rendering, each with different technical choices in terms of scene representations, material models, and lighting handling. Table 1 provides a comprehensive comparison of representative state-of-the-art methods across key dimensions. We compare these methods in terms of their input requirements, geometry and material representations, lighting models, global illumination capabilities, and real-time performance.

6.2 Evolution of Inverse Rendering Approaches

The field of inverse rendering has seen significant evolution through three major approaches:

NeRF-based Methods Early methods like NeRV [30] and NeRD [3] used neural radiance fields to represent scene geometry and appearance. While these methods could achieve high-quality results, they suffered from slow rendering speeds and difficulty in handling global illumination. NeRV introduced visibility fields to approximate global effects, while NeRD focused on material decomposition using spherical Gaussians for lighting.

3DGS-based Methods The introduction of 3D Gaussian Splatting led to a new generation of methods. Relightable 3DGS [12], GS-IR [23], and GeoSplatting [34] leverage the efficiency and quality of 3D Gaussian representations while incorporating physically-based rendering. These methods achieve real-time performance while supporting complex lighting effects. GeoSplatting further improves the accuracy by introducing explicit geometric guidance.

Diffusion-prior Methods Most recently, DPE [25], DRM [11], IntrinsicAnything [7] and MaterialFusion [24] have leveraged powerful diffusion priors while using explicit mesh representations. These methods excel at handling ambiguities in the inverse rendering process by learning strong priors from large-scale datasets. This approach shows promising results in disentangling materials and lighting.

6.3 Technical Trade-offs

Each approach makes different trade-offs in several key aspects as highlighted in Table 2:

Each category of methods has its unique strengths and limitations that make it more suitable for certain applications. NeRF-based methods [30, 3, 22, 4, 36, 5, 1] excel in representing complex geometry and view synthesis but struggle with rendering speed and explicit property editing. 3DGS-based methods [12, 16, 23, 29, 8, 10, 34] achieve a good balance between quality and efficiency, making them practical for real-time applications, though they face challenges with certain geometric details. Methods that use diffusion prior [25, 11, 7, 24] leverage powerful learned priors to handle ambiguities effectively but are heavily dependent on training data and may struggle with unusual scenarios.
These trade-offs inform method selection based on specific application requirements:

- For applications requiring real-time performance and easy editing, 3DGS-based methods are often the best choice.
- When dealing with complex materials and lighting conditions, diffusion-prior methods typically perform better.

Table 1: Comparison of Representative Inverse Rendering Methods. We analyze recent methods across key aspects: input requirements, geometry representation, material model, lighting representation, global illumination support, and real-time rendering capability. Methods are categorized into NeRF-based, 3DGS-based, and Diffusion-prior-based approaches.

Method	Year	Category	Input	Geometry Rep.	Material Rep.	Lighting Rep.	Global Illum.	Real-time
NeRV [30]	2021	NeRF-based	Multi-view	Neural Field	BRDF	HDR Map	✓	×
NeRD [3]	2021	NeRF-based	Multi-view	Neural Field	SVBRDF	SG	×	✓
Relightable 3DGS [12]	2023	3DGS-based	Multi-view	3D Gaussian	BRDF	HDR Map	✓	✓
GS-IR [23]	2023	3DGS-based	Multi-view	3D Gaussian	SVBRDF	HDR Map	✓	✓
GeoSplatting [34]	2024	3DGS-based	Multi-view	3D Gaussian	SVBRDF	HDR Map	×	✓
DPE [25]	2023	Diffusion-prior	Multi-view	Mesh	SVBRDF	HDR Map	×	✓
IntrinsicAnything [7]	2024	Diffusion-prior	Multi-view	Mesh	SVBRDF	HDR Map	✓	✓
DRM [11]	2024	Diffusion-prior	Single-view	Known Mesh	SVBRDF	HDR Map	✓	✓
MaterialFusion [24]	2024	Diffusion-prior	Multi-view	Mesh	SVBRDF	HDR Map	×	✓

Table 2: Detailed comparison of advantages and disadvantages of different inverse rendering approaches. We analyze each category of methods across multiple technical aspects including representation capability, computational efficiency, rendering quality, and practical considerations.

Approach	Advantages	Disadvantages
NeRF-based	• Flexible implicit representation capable of modeling complex geometry • Continuous and differentiable scene representation • Good view synthesis quality for novel viewpoints • No explicit mesh extraction required • Natural handling of semi-transparent materials • Automatic handling of topology	• Slow rendering speed due to multiple network queries • High memory consumption • Difficulty in handling high-frequency lighting details • Limited global illumination support • Hard to modify/edit scene properties after training • Lacks explicit geometry for downstream tasks
3DGS-based	• Real-time rendering capability • Efficient point-based representation • Good balance between quality and speed • Supports physically-based rendering • Explicit surface geometry • Memory efficient compared to NeRF • Better material-light disentanglement	• Can produce floating artifacts • Limited accuracy in normal estimation • Challenges in handling view-dependent effects • May struggle with thin structures • Requires careful point density control • Less flexible than implicit representations
Diffusion-prior	• Strong learned priors for materials and lighting • Better handling of ambiguities • High-quality material decomposition • Robust to complex lighting conditions • Good generalization to unseen scenarios • Explicit mesh representation for easy editing	• Dependent on training data quality • Can be computationally intensive during training • May produce over-smoothed results • Limited by the quality of geometry reconstruction • Requires large-scale training data • Can struggle with unusual materials/lighting

- For scenarios requiring high geometric accuracy without real-time constraints, NeRF-based methods might be preferable.

The ongoing challenge in the field is to develop methods that can combine the advantages of these different approaches while minimizing their respective limitations. Recent works show promising steps in this direction, with hybrid approaches beginning to emerge.

7 Conclusion

This paper has reviewed recent advances in inverse rendering, focusing on three main approaches: NeRF-based, 3DGS-based, and diffusion-prior methods. Starting with NeRF-based approaches, the field established the potential of neural scene representations for high-quality view synthesis. The emergence of 3D Gaussian Splatting then provided more efficient and explicit representations while enabling real-time rendering. Most recently, methods incorporating diffusion priors have offered a principled way to address the inherent ambiguities in decomposing appearance into materials and lighting. While each approach has advanced the field in significant ways, major challenges remain in handling complex real-world scenarios with varying materials, lighting conditions, and indirect illumination effects. Current trends suggest that combining the complementary strengths of these approaches - the efficiency of 3D Gaussians, the quality of physically-based rendering, and the decomposition power of learned priors - may provide promising directions for future research.

References

[1] Benjamin Attal et al. “Flash Cache: Reducing Bias in Radiance Cache Based Inverse Rendering”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 20–36.

[2] Jonathan T. Barron et al. “Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields”. In: *CVPR* (2022).

[3] Mark Boss et al. “Nerd: Neural reflectance decomposition from image collections”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12684–12694.

[4] Mark Boss et al. “Neural-pil: Neural pre-integrated lighting for reflectance decomposition”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 10691–10704.

[5] Mark Boss et al. “Samurai: Shape and material from unconstrained real-world arbitrary image collections”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 26389–26403.

[6] Brent Burley. *Physically-Based Shading at Disney*. Tech. rep. Walt Disney Animation Studios, 2012.

[7] Xi Chen et al. “Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination”. In: *European Conference on Computer Vision*. Springer. 2025, pp. 450–467.

[8] Hoon-Gyu Chung, Seokjun Choi, and Seung-Hwan Baek. “Differentiable Point-based Inverse Rendering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 4399–4409.

[9] R. L. Cook and K. E. Torrance. “A Reflectance Model for Computer Graphics”. In: *ACM Trans. Graph.* 1.1 (Jan. 1982), pp. 7–24. ISSN: 0730-0301. DOI: 10.1145/357290.357293. URL: https://doi.org/10.1145/357290.357293.

[10] Jan-Niklas Dihlmann et al. “Subsurface Scattering for 3D Gaussian Splatting”. In: *arXiv preprint arXiv:2408.12282* (2024).

[11] Yuto Enyo and Ko Nishino. “Diffusion Reflectance Map: Single-Image Stochastic Inverse Rendering of Illumination and Reflectance”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 11873–11883.

[12] Jian Gao et al. “Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing”. In: *arXiv preprint arXiv:2311.16043* (2023).

[13] Param Hanji et al. “Comparison of single image HDR reconstruction methods — the caveats of quality assessment”. In: *ACM SIGGRAPH 2022 Conference Proceedings*. SIGGRAPH ’22. Vancouver, BC, Canada: Association for Computing Machinery, 2022. ISBN: 9781450393379. DOI: 10.1145/3528233.3530729. URL: https://doi.org/10.1145/3528233.3530729.

[14] Martin Heusel et al. “GANs trained by a two time-scale update rule converge to a local nash equilibrium”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6629–6640. ISBN: 9781510860964.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS ’20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[16] Yingwenqi Jiang et al. “Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 5322–5332.

[17] Haian Jin et al. “TensoIR: Tensorial Inverse Rendering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[18] James T Kajiya. “The rendering equation”. In: *In Proceedings of the 13th annual conference on Computer graphics and interactive techniques, 143-150* (1986).

[19] James T. Kajiya and Brian P Von Herzen. “Ray tracing volume densities”. In: *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’84. New York, NY, USA: Association for Computing Machinery, 1984, pp. 165–174. ISBN: 0897911385. DOI: 10.1145/800031.808594. URL: https://doi.org/10.1145/800031.808594.

[20] James T. Kajiya and Brian P Von Herzen. “Ray tracing volume densities”. In: *SIGGRAPH Comput. Graph.* 18.3 (Jan. 1984), pp. 165–174. ISSN: 0097-8930. DOI: 10.1145/964965.808594. URL: https://doi.org/10.1145/964965.808594.

[21] Bernhard Kerbl et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering”. In: *ACM Transactions on Graphics* 42.4 (July 2023). URL: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

[22] Julian Knodt et al. “Neural ray-tracing: Learning surfaces and reflectance for relighting and view synthesis”. In: *arXiv preprint arXiv:2104.13562* (2021).

[23] Zhihao Liang et al. “Gs-ir: 3d gaussian splatting for inverse rendering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 21644–21653.

[24] Yehonathan Litman et al. “MaterialFusion: Enhancing Inverse Rendering with Material Diffusion Priors”. In: *arXiv preprint arXiv:2409.15273* (2024).

[25] Linjie Lyu et al. “Diffusion posterior illumination for ambiguity-aware inverse rendering”. In: *ACM Transactions on Graphics (TOG)* 42.6 (2023), pp. 1–14.

[26] Ben Mildenhall et al. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[27] Gustavo Patow and Xavier Pueyo. “A survey of inverse rendering problems”. In: *Computer graphics forum*. Vol. 22. 4. Wiley Online Library. 2003, pp. 663–687.

[28] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[29] Yahao Shi et al. “Gir: 3d gaussian inverse rendering for relightable scene factorization”. In: *arXiv preprint arXiv:2312.05133* (2023).

[30] Pratul P Srinivasan et al. “Nerv: Neural reflectance and visibility fields for relighting and view synthesis”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7495–7504.

[31] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[32] Kohei Yamashita et al. “nLMVS-Net: Deep Non-Lambertian Multi-View Stereo”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023.

[33] Yao Yao et al. “Neilf: Neural incident light field for physically-based material estimation”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 700–716.

[34] Kai Ye et al. “GeoSplatting: Towards Geometry Guided Gaussian Splatting for Physically-based Inverse Rendering”. In: *arXiv preprint arXiv:2410.24204* (2024).

[35] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 586–595. DOI: 10.1109/CVPR.2018.00068.

[36] Xiuming Zhang et al. “Nerfactor: Neural factorization of shape and reflectance under an unknown illumination”. In: *ACM Transactions on Graphics (ToG)* 40.6 (2021), pp. 1–18.

[37] M. Zwicker et al. “EWA splatting”. In: *IEEE Transactions on Visualization and Computer Graphics* 8.3 (2002), pp. 223–238. DOI: 10.1109/TVCG.2002.1021576.