Inverse Rendering of Glossy Objects via the Neural Plenoptic Function and Radiance Fields: Supplementary Materials

This supplementary material provides additional information and experiment results pertaining to the main paper, "Inverse Rendering of Glossy Objects via the Neural Plenoptic Function and Radiance Fields", including detailed descriptions of the training process, a mathematical derivation of a θ_c, and more visual results to complement the experiments reported in the main manuscript.

1. Implementation Details

In this section, we elaborate on the training methodology and loss functions utilized across the two stages of our method, and outline the specific strategies to stabilize and optimize the learning process for both geometry and material properties.

1.1. Fields Learning Stage

In the fields learning stage, our loss function comprises four components:

$$L_1 = \lambda_1 L_c + \lambda_2 L_{\text{eik}} + \lambda_3 L_{\text{stable}} + \lambda_4 L_{\text{curv}}, \quad (1)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are loss weights. The color loss L_c aligns the rendered images with the input data, as elaborated in the main paper. Accompanying it is the eikonal loss, L_{eik}, which regulates the SDF value to keep its gradient norm equal 1. This loss is critical for enforcing the SDF constraint and is mathematically represented as:

$$L_{\text{eik}} = \sum_{\mathbf{x}} \left(\|\nabla \text{SDF}_{\mathbf{x}}\| - 1\right)^2. \quad (2)$$

To improve the stability of the training, especially in the initial phase, we incorporate L_{stable}, which pushes the SDF values of sampled points near the unit sphere bounds toward zero following NeRO [4].

Surface smoothness [6] is accounted for by L_{curv}, which constrains the curvature of the normals. By evaluating points slightly perturbed along the tangent plane, this loss ensures that the reconstructed geometry does not exhibit unwarranted roughness, thereby promoting a smoother surface representation:

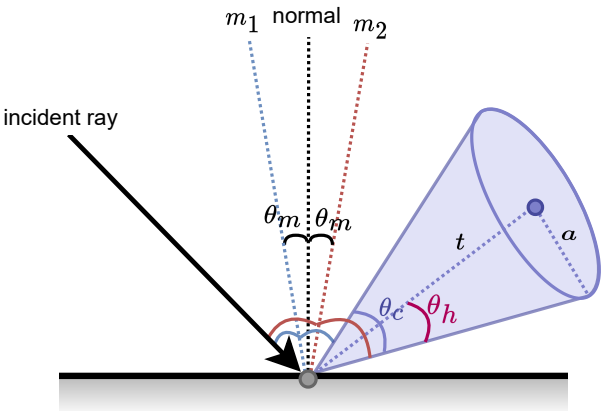

Figure 1. The geometry relationship of angles, where the angles notated in the same color equal.

$$L_{\text{curv}} = \sum_{\mathbf{x}} (\mathbf{n} \cdot \mathbf{n}_\epsilon - 1)^2, \quad (3)$$

where $\mathbf{n}_\epsilon$ is the sampled point on the tangent plane and $\mathbf{n}$ is the normal vector at position $\mathbf{x}$.

1.2. Material Learning Stage

In the subsequent material learning stage, we adopt a straightforward plain L1 loss as RGB loss L_{rgb}, which is supervised by the GT pixel colors of the target object. As a complement, we use a regularization loss L_{reg}, which ensures the uniformity of the predicted material attributes across nearby points for position $\mathbf{x}$, preserving the smoothness items for roughness, metallic, albedo, and diffuse light properties. For example, the roughness smoothness item is:

$$L_{\text{reg-roughness}} = \sum_{\mathbf{x}} ||r - r_\epsilon||, \quad (4)$$

where r and r_ϵ are the predicted roughness at position $\mathbf{x}$ and slightly shifted position $\mathbf{x} + \epsilon$.

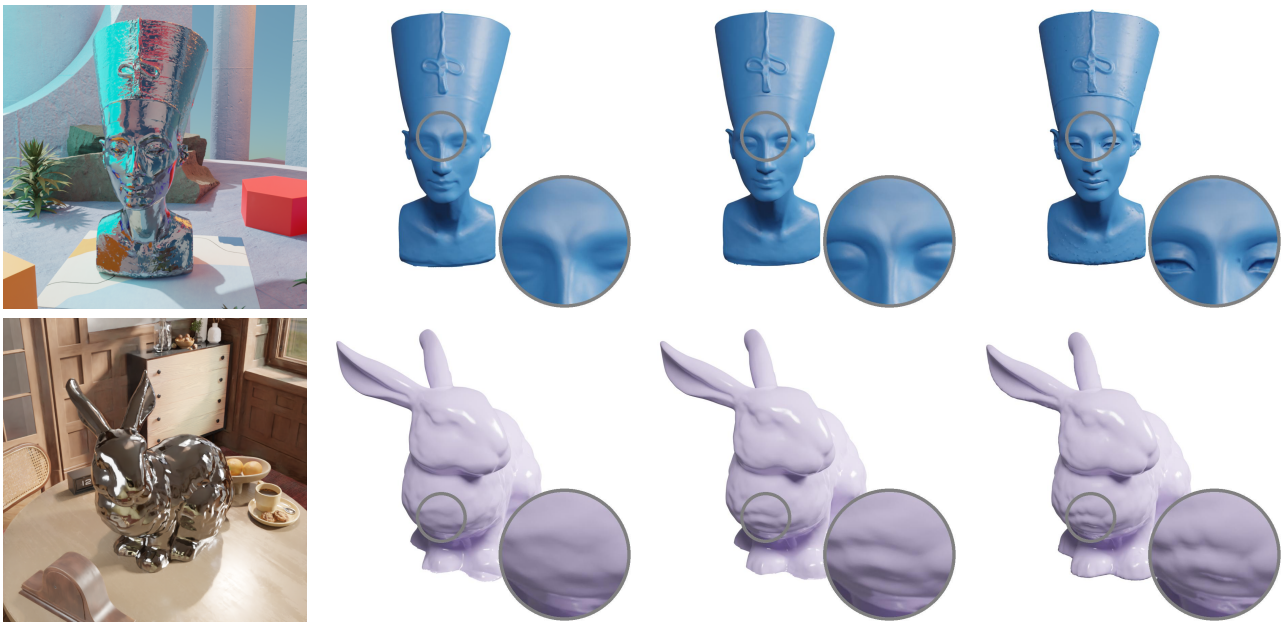

Figure 2. Additional geometry reconstruction cases compared with NeRO. This comparison further validates the robustness and fidelity of the generated object geometries.

1.3. Implementation and Training Time

Our implementation is based on the Nerfstudio framework [7] and PyTorch. We utilize the PyTorch adaptation of NeRF for the environmental field and object field. Despite the faster training potential of Instant-NGP [5] version, our practice shows instability when modifications are introduced to the default instant-NGP model. The training time for our fields learning stage is approximately 4–5 hours, and the material learning stage takes about 1 hour, both using a single NVIDIA RTX 4090 GPU. For reference, training NeRO with identical hyperparameters and training steps also requires 4-5 hours for stage 1 and about 1 hour for stage 2.

2. Derivation of θ_c

The variable θ_c plays a pivotal role in our material-aware cone sampling strategy. Here we present a step-by-step mathematical derivation of θ_c, elucidating its theoretical underpinnings and practical applications within our framework.

2.1. Mathematical Formulation

Given the probability density function (PDF) for the azimuth and elevation angles of GGX distribution:

$$p_m(\theta,\phi) = \frac{r^4 \cos\theta \sin\theta}{\pi((r^4-1)\cos^2\theta+1)^2}, \tag{5}$$

we integrate it over the azimuth angle to yield a function of the elevation angle only, as the GGX distribution is isotropic:

$$p_m(\theta) = \int_0^{2\pi} p_m(\theta,\phi)d\phi = \frac{2r^4 \cos\theta \sin\theta}{((r^4-1)\cos^2\theta+1)^2}. \tag{6}$$

The cumulative distribution function (CDF), which represents the probability of the microfacet normal being within a cone defined by angle θ, is then:

$$\begin{aligned} P_m(\theta) &= \int_0^{\theta_m} p_m(\theta)d\theta \\ &= \frac{r^4}{(r^4-1)\left((r^4-1)\cos^2(\theta)+1\right)}\Bigg|_0^{\theta_m} \\ &= \frac{\sin^2(\theta_m)}{(r^4-1)\cos^2(\theta_m)+1}. \end{aligned} \tag{7}$$

To find the value of θ that captures a predefined portion β of the light energy, we solve the equation $P_m(\theta) = \beta$:

$$\begin{aligned} \beta &= \frac{\sin^2(\theta_m)}{(r^4-1)\cos^2(\theta_m)+1} \\ \Rightarrow\ \beta &= \frac{\tan^2(\theta_m)}{(r^4-1)+(1+\tan^2(\theta_m))} \\ \Rightarrow\ \theta_m &= \arctan\left(r^2\sqrt{\frac{\beta}{1-\beta}}\right) \end{aligned} \tag{8}$$

This θ_m defines the range for sampling the orientation of microfacet normals and correlates with the spatial extent of the BRDF lobe. Based on θ_m, it is simple to get the representation of the cone vertex angle $\theta_c = 4\theta_m$ via geometry relationship illustrated in Fig. 1

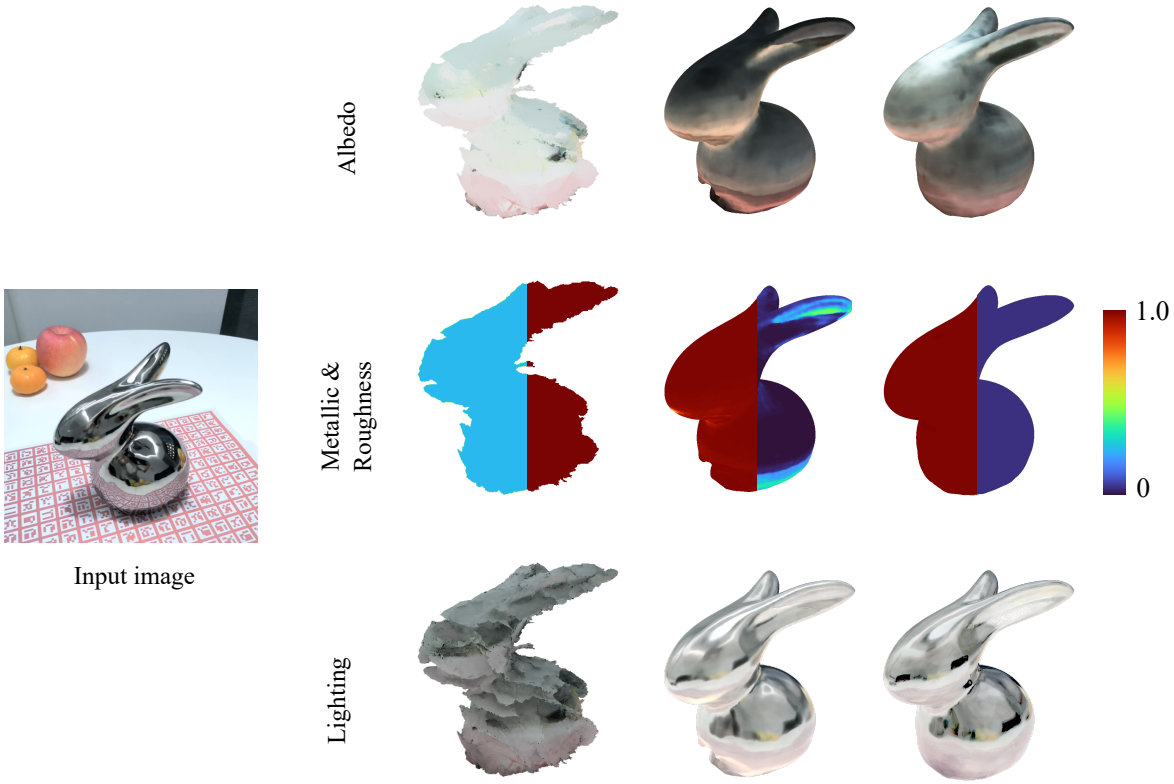

Figure 3. Comparative material reconstruction results with the additional method NvdiffrecMC, which highlights the capability of high-quality material recovery of our approach

2.2. Practical Implications

In mathematical, we have $\theta_c = 4\theta_m$, with θ_m being expressible in terms of the roughness parameter r. However, obtaining the actual value of θ_m necessitates computing the arctangent function, which incurs additional computational overhead.

In practice, to avoid this inefficiency, we do not employ θ_c directly. Instead, we utilize its half angle $\theta_h = 2\theta_m$. Given that Integral Positional Encoding (IPE) is defined in terms of the radius a of the circle at position $\mathbf{x}$, we can calculate a directly using the formula $a = \tan(\theta_h)t$, where t is the distance from the ray origin to $\mathbf{x}$. This computation is further simplified by leveraging the trigonometric identity:

$$\tan(\theta_h) = \frac{2\tan(\theta_m)}{1 - \tan^2(\theta_m)}, \tag{9}$$

thereby streamlining the calculation and reducing the time complexity.

3. More Visual Results

To further illustrate the effectiveness of our proposed method, we include more visual results that showcase the model's performance across various scenes of the proposed dataset.

3.1. Extended Geometry Experiments

We show additional cases in our geometry reconstruction experiments to further validate the robustness and accuracy of our approach. These extended cases provide more scenarios, ensuring a comprehensive evaluation. The results of these additional experiments are illustrated in Fig. 2.

3.2. Material Comparison Experiments

Notwithstanding the space constraints of the main paper, we compare NeRO only in our main paper. Thus, here we include an additional method NvdiffrecMC [1], in our material comparison experiments within this supplementary document. The results are depicted in Fig. 3. Although several noteworthy methods have been proposed recently, such as NeILF++ [9], ENVIDR [3], and TensoIR [2], we found

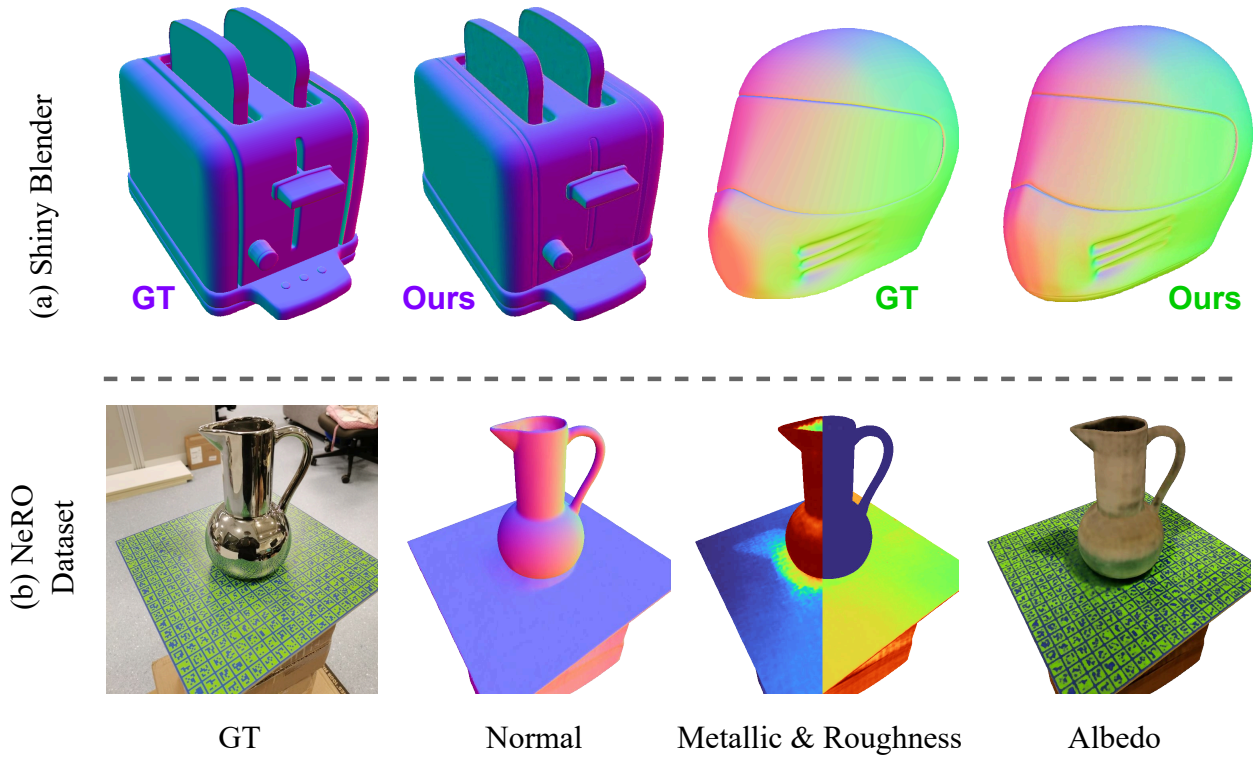

Figure 4. Evaluation results on the Shiny Blender and NeRO datasets.

ENVIDR and TensoIR have different decomposition formulations as ours. *e.g.*, they did not explicitly model certain properties such as metallic. Additionally, the implementation of NeILF++ supports the DTU dataset only, limiting its comparability with our approach. Therefore, these methods were not included in our comparative analysis.

3.3. Extended Evaluation on Additional Datasets

In this section, we extend the evaluation of our proposed method to include its performance on the Shiny Blender dataset [8] (Fig. 4.a) and the NeRO dataset [4] (Fig. 4b). Specifically, for the Shiny Blender dataset, we present only the geometric outputs, as our method is unable to optimize the environmental field and materials in the absence of background information.

3.4. Supplementary Video Results

For a more comprehensive view of our method's performance, we refer the reader to the supplementary videos attached. These videos provide dynamic visualizations of the reconstructed geometry and materials comparison in the main paper, offering a vivid demonstration of our method's capabilities.

References

[1] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. In *NeurIPS*, 2022. 3

[2] 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*, 2023. 3

[3] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. Envidr: Implicit differentiable renderer with neural environment lighting. In *ICCV*, 2023. 3

[4] Yuan Liu, Peng Wang, Cheng Lin, Xiaoxiao Long, Jiepeng Wang, Lingjie Liu, Taku Komura, and Wenping Wang. Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. In *SIGGRAPH*, 2023. 1, 2, 4

[5] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):1–15, 2022. 2

[6] Radu Alexandru Rosu and Sven Behnke. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *CVPR*, 2023. 1

[7] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, Justin Kerr,

and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, 2023. 2

[8] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 4

[9] Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf++: Inter-reflectable light fields for geometry and material estimation. In *ICCV*, 2023. 3