Glossy Object Reconstruction with Cost-effective Polarized Acquisition

Supplementary Material

1. Neural Radiance Field

Neural implicit surface. We apply the neural volume rendering framework to represent implicit surfaces and follow VolSDF [6] to parameterize the density values with the transformation of an SDF. For each pixel, we sample N points along the camera ray and approximate the color $\hat{C}$ by:

$$\hat{C} = \sum_{i=1}^{N} w_i c_i, \tag{1}$$

$$\text{with } w_i = T_i \left(1 - \exp(-\sigma_i \delta_i)\right),\ T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{2}$$

where w_i is the weight of rendering, σ_i and c_i denote the density and color at each sampled point i on the ray, and δ_i is the distance between adjacent samples. The density is defined as Laplace's cumulative distribution function applied to a signed distance d, as follows:

$$\sigma(d) = \begin{cases} \frac{1}{2\beta} \exp(\frac{d}{\beta}) & if\ d \leq 0 \\ \frac{1}{\beta}\left(1 - \frac{1}{2}\exp(1 - \frac{d}{\beta})\right) & if\ d > 0 \end{cases}. \tag{3}$$

Herein, β is a learnable parameter during network training. In practice, we use MLPs to take 3D coordinates as input and output the corresponding signed distance as well as a global geometric feature vector. Referring to Eq. 3, the estimated SDFs are transformed to density values for volumetric integration of Eq. 2.

Decomposed radiance fields. The outgoing radiance c of a sampled point $\mathbf{x}$ on the camera ray can be decomposed into diffuse radiance c^d and specular radiance c^s, respectively, as follows:

$$c^d = f_\theta(\mathbf{b}, \mathbf{x}),\ c^s = g_\theta(\mathbf{b}, \mathbf{IDE}(\eta, \omega_r)),\ \text{and } c = \gamma(c^d + c^s), \tag{4}$$

where $f_\theta(\cdot)$ and $g_\theta(\cdot)$ denote MLPs with learnable parameters, and $\mathbf{b}$ is the geometric feature vector as mentioned above. Following the representations in Eq. 7, the diffuse surfaces should satisfy the property of Lambertian, thus c^d in fact is only a function of position. However, for spatially-varying specular effects, following Verbin *et al*. [5], the radiance has strong correlations with surface roughness η and the reflective direction of light ω_r. With integrated directional encoding (IDE), the directions are encoded with a set of spherical harmonics, which enables the network to better reason about the inherent properties of the material. Finally, the diffuse and specular components are combined together with a fixed tone mapping function γ.

2. Polarimetric BRDF Model

In this work, we only consider linear polarization and build a scalable setup for the polarization image acquisition. To provide a clearer understanding of how polarization information is utilized in our method, We begin by presenting in the following the fundamental concepts.

Stokes vector. The polarization state of light is often characterized by the Stokes vector $\mathbf{s}$, which is usually computed by taking a series of measurements with different rotation angles, for example, polarized images with four different polarizing angles $0°$, $45°$, $90°$ and $135°$, represented by $\mathbf{I}_0$, $\mathbf{I}_{45}$, $\mathbf{I}_{90}$ and $\mathbf{I}_{135}$:

$$\mathbf{s} = [s_0, s_1, s_2, s_3]^T = [\mathbf{I}_0 + \mathbf{I}_{90}, \mathbf{I}_0 - \mathbf{I}_{90}, \mathbf{I}_{45} - \mathbf{I}_{135}, 0]^T. \tag{5}$$

Mueller matrix. Any change of the polarization state due to the interaction with optical elements, such as linear polarizers or object surfaces, can be denoted as a multiplication of the corresponding Stokes vector with a Mueller matrix $\mathbf{M} \in \mathbb{R}^{4\times4}$. The incident and outgoing Stokes vector, represented by $\mathbf{s}^{\text{in}}$ and $\mathbf{s}^{\text{out}}$, respectively, are related by

$$\mathbf{s}^{\text{out}} = \mathbf{M}\mathbf{s}^{\text{in}}. \tag{6}$$

For surface reflection, considering the distant incident illumination L_i, which is commonly assumed to be unpolarized, its corresponding Stokes vector is denoted as $\mathbf{s}_i = L_i[1, 0, 0, 0]^T$. Based on the pBRDF model proposed by Baek *et al*. [1], the Mueller matrix can be decomposed as the sum of diffuse component $\mathbf{M}^d$ and specular component $\mathbf{M}^s$, i.e., $\mathbf{M} = \mathbf{M}^d + \mathbf{M}^s$. Therefore, the outgoing Stokes vector can be reformulated as follows:

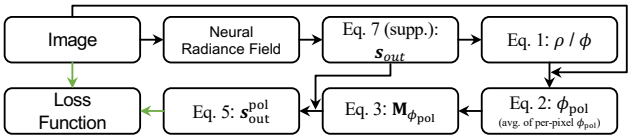
$$\mathbf{s}^{\text{out}} = (\mathbf{M}^d + \mathbf{M}^s)\mathbf{s}^{\text{in}} = \underbrace{L_i k_d (\mathbf{n}\cdot\mathbf{i})}_{c^d} \begin{bmatrix} T_o^+ T_i^+ \\ T_o^- T_i^+ \beta_o \\ -T_o^- T_i^+ \alpha_o \\ 0 \end{bmatrix} + \underbrace{L_i k_s \frac{DG}{4(\mathbf{n}\cdot\mathbf{o})}}_{c^s} \begin{bmatrix} R^+ \\ R^- \gamma_o \\ -R^- \chi_o \\ 0 \end{bmatrix}. \tag{7}$$

In essence, $\mathbf{M}^d$ and $\mathbf{M}^s$ depend on surface albedo, surface normals, refractive index, and lighting conditions. In short, $\mathbf{n}$, $\mathbf{i}$, and $\mathbf{o}$ represent surface normal, incident and outgoing light direction, respectively. k_d is the diffuse albedo, k_s is the specular albedo, T and R are the Fresnel transmission and reflection coefficients. Refer to Baek *et al*. [1] for

detailed explanations and computation of remaining parameters. Herein, we denote the coefficients of the two terms on the right side of Eq. 7 as diffuse radiance c^d and specular radiance c^s.

3. Polarization Rendering

As shown in Fig. 2 and the following rendering pipeline, we use a polarization image $\mathbf{I}_{\phi_{\text{pol}}}$ as the input and leverage polarimetric BRDF model, characterized by the neural radiance field, to estimate the outgoing Stokes vectors $\mathbf{s}^{\text{out}}$, which lay the foundation for polarization rendering. Refer to Eq. 7 in the supp. for how to render $\mathbf{s}^{\text{out}}$ using diffuse, specular, and roughness components. Subsequently, we present a differentiable processing pipeline to estimate the ϕ_{pol}, eliminating the need for precise polarization angle measurements and facilitating the implicit rendering of desired polarized images $\mathbf{I}^{\text{out}}_{\phi_{\text{pol}}}$ for loss calculation.



4. Implementation Details

The SDF network takes the 3D coordinate as input and applies the positional encoding (PE) to spatial locations using 6 frequencies. This encoded input is then processed through 8 fully connected layers with 256 channels each, utilizing ReLU activations. Additionally, the encoded input vector is connected to the output feature at the 4^{th} layer through a skip connection. The network outputs the signed distance value and an extra 256-dimensional geometric feature vector. Notably, surface normals can be obtained as the normalized gradient of the neural SDF. To initialize parameters of the SDF network, we utilize geometric initialization methods as described by Gropp *et al*. [2].

The diffuse radiance f_θ, roughness, and mask prediction functions share similar network architectures. They take the concatenation of the geometric feature vector and the encoded spatial locations with 10 frequencies as input. The network is composed of 4 MLP layers with a width of 512 channels. The output structures contain 3 channels with *sigmoid*, 1 channel with *softplus*, and 1 channel with *sigmoid*, respectively. For the estimation of specular components [5], we enable the network to reason about radiances with the integrated directional encoding of roughness and the encoded reflective directions with PE of 2 frequencies. g_θ also uses 4 fully connected MLP layers with 512 channels per layer and outputs 3 channels with the *softplus*.

Our algorithms are implemented in Pytorch [4]. In our experiments, we use a batch size of 512 rays, each sampled at 128 locations. We use the Adam optimizer [3] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning rate that begins at 5×10^{-4} and decays exponentially to 5×10^{-5} during training. To better warm up the training, in the early 10k iterations, we define $\mathcal{L}_{\text{rgb}}$ as the loss between the predicted radiance c in Eq. 4 and the ground truth. In the next 5k iterations, we replace c with the diffuse components of $\mathbf{s}^{\text{out}}_{\phi_{\text{pol}}}$, which are subsequently used for loss computation. In addition, The refractive index of the object is set to 1.5. The optimization for a single object typically takes around 200k iterations to converge on a single NVIDIA Titan X GPU ($\sim$ 2 days).

References

[1] Seung-Hwan Baek, Daniel S Jeon, Xin Tong, and Min H Kim. Simultaneous acquisition of polarimetric svbrdf and normals. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 37(6): 268–1, 2018. 1

[2] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *Proc. Int. Conf. on Machine Learning*, 2020. 2

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. Int. Conf. on Learning Representations*, 2014. 2

[4] 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. In *Proc. Neural Information Processing Systems*, 2019. 2

[5] 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 *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5481–5490. IEEE, 2022. 1, 2

[6] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Proc. Neural Information Processing Systems*, 34:4805–4815, 2021. 1