Reconstructing and Rendering Shiny Surface with Hybrid Neural Fields

Ruofan Liang
Department of Computer Science
University of Toronto
ruofan@cs.toronto.edu

Neural Radiance Fields (NeRF) is a popular method for modeling and rendering 3D scenes with photorealistic quality. However, NeRF models often struggle to accurately represent shiny surfaces with high specular reflectance. In this paper, we propose a new method for improving the representation of specular reflectance in NeRF models. Our method uses a hybrid approach that combines traditional volume rendering methods with deep learning to accurately capture the complex reflectance of shiny surfaces. We also introduce a special parameterization of the environment illumination that allows us to decompose the environment representation from the NeRF model. We evaluated our method on challenging shiny 3D scenes and compared it to several baselines. The results showed that our method performed on par with the SOTA baselines but with much faster training and rendering speed.

1 Introduction

Neural Radiance Fields (NeRF) [1] nowadays become a hot topic in the area of computer vision and graphics. By combining deep learning and traditional volume rendering methods, NeRF achieves great improvements in modeling and reconstructing 3D scenes with photo-realistic rendering quality. NeRF's strong ability to represent 3D objects is favored by a wide range of related 3D vision or graphics tasks, including novel view synthesis [2], [3], 3D content creation [4], [5], visual SLAM [6], [7], relighting [8]–[10], etc.

Although NeRF method is able to synthesize novel view images with promising visual qualities, NeRF models often fail to accurately represent shiny surfaces with high specular reflectance. Instead of learning a solid, smooth surface for those shiny regions, NeRF tends to misinterpret the view-dependent specular reflectance as emitting light sources under the real surfaces (e.g., Figure **??**). This erroneous behavior of NeRF also results in the poor quality of the extracted surface on the shiny regions, because the fake internal light sources have to be able to transmit through the internal volume for synthesizing view-dependent reflectance effects. Verbin et al. also discussed this issue in their work [11], please refer to their paper for a more detailed analysis.

There are two lines of work that attempt to address the problem of learning specularity in NeRF. The first direction is to make NeRF's rendering more like a surface rendering [12]–[14]. Instead of using volume density to represent 3D geometry, the methods in this line use surface-based implicit geometry representation to ensure higher surface reconstruction quality for their surface-based rendering. The work in the second direction sticks with NeRF's volume representation and puts more effort into the modeling of view-dependent radiance prediction MLP (e.g., Ref-NeRF [11]). They thus improve the model's capability for representing complex reflectance on the pure object surface.

These abovementioned works are able to improve the quality of representation for the challenging specular reflectance, but they do have limitations. First, these methods are costly to train. Because their optimization process involves the normal computation (the gradient from the estimated density/SDF), the actual training time for these methods could be even longer than vanilla NeRF [1]. Second, the current top-performing NeRF models still encode the illumination from the global environment in the object volume. Although their color MLPs do learn the environment light information, the learned knowledge of the environment is entangled with the base color of the object and is hard to extract or swap. It should note that there are works that explicitly learn the environment during the training [10], [15], but these methods have lower rendering quality compared to the state-of-the-art NeRF models [2], [11]. To this end, our goal is to propose a more efficient NeRF model that is able to capture accurate specular reflectance with special parameterization of the environment illumination in the NeRF model.

To achieve the aforementioned goals, we make efforts in the following two aspects. **First**, to accelerate the training and rendering, we replace NeRF's fully implicit MLP representation with an implicit-explicit hybrid neural representation (e.g., instant-NGP [16]). However, directly applying hybrid neural representations such as instant-NGP will hurt the continuity of represented surfaces due to their discrete

This is an independent project for course CSC2529.

Fig. 1. NeRF's fake emitters under the object surface.

neural features. To address this problem, we leverage the recent success of SDF-based volume rendering [13], [14] to predict the SDF with NeRF's geometry MLP and then convert the SDF to the volume density. To ensure the continuity of the predicted SDF and smoothness of the represented surface in a discrete hybrid NeRF model, we propose 2 new SDF regularizations based on SDF's geometry property. **Second**, we propose to add extra neural representation to encode the environment for learning specular surface color. Since the environment is unknown during the training and we can only indirectly observe the environment through the object reflectance. Thus, we use the surface incoming light direction (the reflected direction of outgoing radiance w.r.t. surface normal) to condition our neural environment encoding. Note that Ref-NeRF [11] also uses reflected direction with special encodings for learning more accurate specular reflectance, but Ref-NeRF does not have separate parameterization of the environment, which limits its capability of inverse rendering. Our proposed encoding has the potential to extract the environment for more flexible editability[1].

To summarize, our contributions are:

- We propose new regularization terms for hybrid NeRF models with discrete neural features to learn accurate and smooth object surfaces.
- We add an additional environment encoding MLP to better learn the global illumination for rendering shiny surfaces.
- Our model achieves better rendering quality compared to prior SDF-based volume rendering methods and with much short training and rendering time.

2 Related Work

Neural rendering and NeRF. Neural rendering is a class of reconstruction and rendering approaches that use deep networks to learn complex mappings from captured images to novel images [17]. The recent progress in neural rendering has shown promising results in various tasks such as texture mapping [18], surface reconstruction [12], image generation [19], etc. Neural radiance field (NeRF) [20] is one representative work that shows how the current state-of-the-art methods [2], [3], [11] combine volume rendering and neural network based implicit representation for photo-realistic novel view synthesis.

NeRF with surface reconstruction. Follow-up methods that combine surface SDF estimation and neural volume rendering [13], [14] further improve the quality of surface reconstruction that the original NeRF performs not well.Both paper uses Eikonal loss to regularize the learned SDF. MonoSDF [21] uses monocular depth estimated by pretrained model to regularize the learned geometry, which improves reconstructed surface.

NeRF acceleration with hybrid models. In order to make NeRF's training and rendering more efficient, hybrid neural representations that combine implicit MLP and other discrete spatial representations such as voxel grids [22]–[24], point cloud [25], [26] and hash table [27] have been proposed and achieved significant speed boost. By leveraging the explicit spatial data structure provided by these hybrid approaches, the ray-marching process can efficiently skip empty regions and only sample point around the real surface. Besides, since the neural features are discretized over the spatial data structure, we can only use some shallow or tiny MLP to achieve fast training speed without much loss of rendering quality.

3 Preliminaries

3.1 Neural Radiance Field.

NeRF [1] is a new representation method for 3D scenes. The scene is represented as an implicit function $F_\Theta : (\mathbf{x}, \hat{\mathbf{v}}) \rightarrow (\mathbf{c}, \sigma)$, whose input is a position $\mathbf{x} = (x, y, z)$ and a direction[2] $\hat{\mathbf{v}} = (\theta, \phi)$ and the output is the emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. In practice, the direction is represented as a 3D unit vector in Cartesian coordinates, and the function F_Θ is approximated with an MLP with parameters Θ. For rendering each pixel in the image, NeRF adapts the classical volume ray casting [28]. The radiance of one pixel is computed by the ray marching through the pixel with a ray $\mathbf{r}$, sampling N point positions $\{\mathbf{x}_t \mid t = 1, 2, \cdots, N\}$, and accumulating the radiance $(\mathbf{c}_t, \sigma_t)$ at each position as

$$\mathbf{C}(\mathbf{r}) = \sum_t w_t \mathbf{c}(\mathbf{x}_t)$$
$$\text{with } w_t = \exp(-\sum_{j=1}^{t-1} \sigma(\mathbf{x}_t)\delta_t)(1 - \exp(-\sigma(\mathbf{x}_t)\delta_t)) \quad (1)$$

where δ_t denotes the distance between adjacent sampled positions along the ray.

3.2 SDF-based neural volume rendering.

Although NeRF is able to synthesize promising novel view images, the underlying 3D geometry of the reconstructed scene is not as good as the synthesized images. Recent works show that implicit surface representation can improve the quality of NeRF's reconstructed geometry [13], [14], [29]. Yariv et al. [13] illustrate a concrete mathematical conversion of volume density $\sigma(\mathbf{x})$ from predicted SDF $d_\Omega(\mathbf{x})$ of the space $\Omega \subset \mathbb{R}^3$ where the target object locates:

$$\sigma(\mathbf{x}) = \frac{1}{\beta}\Psi_\beta(-d_\Omega(\mathbf{x})) \quad (2)$$

β is a learnable parameter, and Ψ_β is the cumulative distribution function (CDF) of the Laplace distribution:

$$\Psi_\beta(s) = \begin{cases} \frac{1}{2}\exp(\frac{s}{\beta}) & \text{if } s \leq 0, \\ 1 - \frac{1}{2}\exp(-\frac{s}{\beta}) & \text{if } s > 0. \end{cases} \quad (3)$$

1. At the point of writing this report, this goal has not been realized, this is one future work.

2. The hat symbol $\hat{}$ in this paper only denotes unit directional vector.

Following this parameterization, we get a high-quality smooth surface from the zero level-set of predicted SDF values, and surface normal can be computed as the gradient of the predicted SDF w.r.t. surface point $\mathbf{x}$, i.e., $\hat{\mathbf{n}}_{\mathbf{x}} = \frac{\nabla d_\Omega(\mathbf{x})}{\|\nabla d_\Omega(\mathbf{x})\|}$.

3.3 Multi-resolution Hash encoding.

Recently, Instant-NGP [16] proposes to represent the whole 3D space with multi-resolution grids stored in a hash table. Instead of using a fully implicit MLP, Instant-NGP uses a combination of implicit and explicit neural networks to represent the 3D scene. For each query point position $\mathbf{x}$, instant-NGP's hash encoding outputs its neural feature $\mathbf{h}_{\mathbf{x}}^{(l)}$ by interpolating the feature grids at each level l. Then the features from all resolution levels are concatenated together as the point's encoded neural features $\mathbf{h}_{\mathbf{x}}$. Then these features are fed into a shallow MLP for predicting density and radiance. Thanks to this efficient structure, Instant-NGP accelerates the training and rendering stage of NeRF by a large margin without obvious performance degradation.

4 Study on the Combination of Hybrid NeRF and SDF-based Volume Rendering

We start with a naive solution to show why the naive combination of hybrid NeRF and SDF-based volume rendering is not feasible. In this experiment, we use VolSDF as the baseline SDF-based volume rendering method. To combine the hybrid NeRF representation with VolSDF, we simply replace VolSDF's fully implicit MLP with Instant-NGP (VolSDF+NGP) and keep the rest parts unchanged (including Eikonal loss [30]). We show the comparison of the surface normal in Figure 2(a). We can see that the surface normal of the VolSDF+NGP is much worse than the baseline VolSDF. To understand the reason, we show the SDF and compositing weight curves of sampled points along the camera ray in Figure 2(b). We can see that the SDF curve of our naive combination has a lot of fluctuations near the surface. This leads to an unconcentrated compositing weight curve and thus a bumpy surface.

One possible reason for the poor performance of the naive combination of hybrid NeRF and SDF-based volume rendering is that the Eikonal loss used in VolSDF can not guarantee the smoothness of the hybrid NeRF that has discrete local neural parameters. The Eikonal loss [30] is designed to minimize the error between the continuous SDF value and the distance to the surface, while the hybrid NeRF representation uses discrete neural features with local interpolation to approximate the SDF function. Therefore, directly replacing the fully implicit MLP with Instant-NGP may not produce good results.

A better solution would be to design new regularization terms that can better handle the combination of hybrid NeRF and SDF-based volume rendering. These new regularizations should take into account the discrete nature of the hybrid NeRF representation and the continuous nature of the SDF-based volume rendering. By carefully designing such a loss function, it may be possible to achieve good results with the combination of hybrid NeRF and SDF-based volume rendering. We will show our solution in the later section.

5 Methodology

This section describes our proposed method in detail. We first introduce our new SDF regularization terms, then our design of the hybrid NeRF representation for learning challenging shiny surfaces.

5.1 SDF Constraints

As analyzed in Section 4, the discretization of implicit neural representation can cause the loss of continuity of the represented signals [24], [31], because the predicted signals now are mainly controlled by the local neural features instead of a global MLP. To address this issue, we add two regularization terms to the predicted SDF values based on SDF's geometric properties.

The first regularization is to improve the continuity and consistency of predicted SDF values. Since the signed distance field defines the local spatial distance information of 3D positions, we can utilize this property to regularize the predicted SDF values along the sampled rays. Given two adjacent sampled points $\mathbf{x}_t, \mathbf{x}_{t+1}$ with sufficiently small distance along the same ray $\mathbf{r}_i$, the difference of predicted SDF values $\Delta d_t = d_{\mathbf{x}_{t+1}} - d_{\mathbf{x}_t}$ can be approximated by the projection of the relative distance $\delta_t = \|\mathbf{x}_{t+1} - \mathbf{x}_t\|$ in the normal direction $\hat{\mathbf{n}}_{\mathbf{x}_t}$ (Fig. 3), which is $\Delta d'_t \approx (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}_{\mathbf{x}_t})\delta_t$. We can therefore use the real distance value δ_t to constrain the changes of predicted SDF $d_{\mathbf{x}}$ with another L2 loss term

$$\mathcal{L}_d = \sum_t \|\Delta d'_t - \Delta d_t\|^2 \tag{4}$$

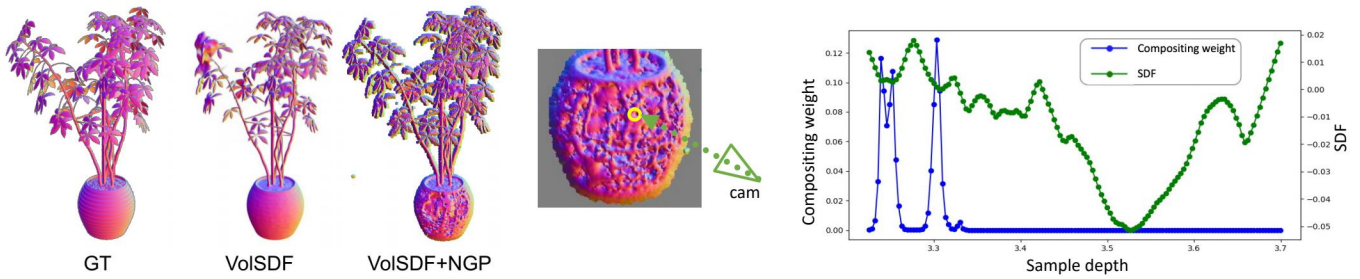
This regularization term can help our hybrid NeRF model better learn the geometry of the target object without being affected by the discrete representation.

The second regularization is to stabilize the SDF prediction for points inside the surface. We can see from Fig. 3 and Fig. 4 that simply adding SDF continuity regularization is not enough for accurate SDF prediction. The SDF curve shown in Fig. 3 still has a lot of fluctuations. We want our estimated SDF to have a stable decreasing curve along the ray sampled points when crossing a solid surface. To achieve this goal, we propose a back-face suppression regularization that tries to penalize the segments of ray-sampled SDF curves that have positive slopes and also have large corresponding compositing weights. We formulate this regularization as a loss term:

$$\mathcal{L}_b = \sum_t w_t \max(\Delta d_t, 0) \frac{\Delta d_t}{\delta_t^2 + \Delta d_t^2} \tag{5}$$

where w_t is the compositing weight for radiance accumulation. The term $\frac{\Delta d_t^2}{\delta_t^2 + \Delta d_t^2}$ is the squared sine value of the angle of slope on the SDF curve. By constraining the SDF value of the internal points, this regularization term can also prevent the model from incorrectly learning internal emitters [11] instead of a solid surface when there are specular highlights on the surface.

After applying both constraints, the hybrid NeRF model learns significantly better surface normal compared to the one without any constraints. The smooth and accurate surface and surface normal are critical for encoding environment illumination in our next step.



(a) The comparison of the surface normal. (b) The SDF and compositing weight curves of sampled points along the camera ray.

Fig. 2. **The study of the naive combination of hybrid NeRF and SDF-based volume rendering.** (b) shows the curves of sampled points along the camera ray that corresponds to the circled pixel shown on the image patch.

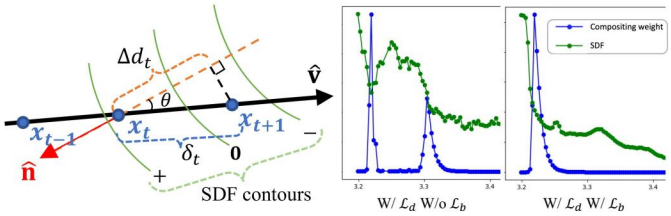


Fig. 3. Visual illustration of our SDF constraints. Left: illustration of our SDF approximation with relative point distance. Right: the effect of our back face suppression.

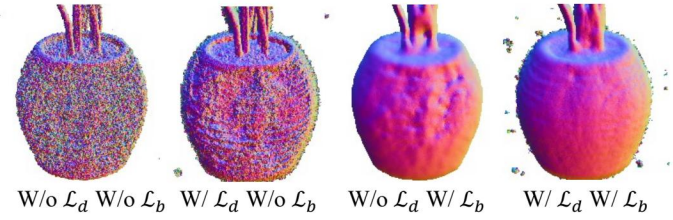


Fig. 4. The effect of our proposed SDF constraints. Note that the evaluated models are only after 100 epochs of training. No Eikonal loss is used.

5.2 Model Design

This subsection describes the proposed model for the combination of hybrid NeRF and SDF-based volume rendering. The overall architecture of the model is shown in Figure 5(a).

The model consists of three main components: a hybrid NeRF module, a color decomposition module, and an environment lighting module. The hybrid NeRF module uses Instant-NGP to encode spatial coordinate input. The color decomposition module decomposes the color of each point on the surface into diffuse and specular components. This allows the model to capture the different lighting effects of different materials in the scene. The environment lighting module uses a relatively large multi-layer perceptron (MLP) to implicitly learn global illumination from the environment. This allows the model to produce more realistic lighting effects in the final rendering.

Since our objective is to render shiny surfaces. The decomposition of color can help the model better learn the specular reflectance, and such decomposition is very common in the modern rendering pipeline. We learn that specular reflectance is the result of interaction between the surrounding environment and surface. Instead of letting the spatial volume feature encode all the reflectance information, our additionally added environment MLP can better learn the globally consistent environment illumination. These learn environment encodings are then concatenated with other volume features and properties to finally predict the specular color. Given a mirror-like surface (i.e., directional BRDF), the outgoing radiance from the view direction is mainly determined by its reflected direction w.r.t. the surface normal. Therefore, we assume using the reflected view direction with positional encoding [1] can well condition our environment encoding MLP. Figure 5(b) shows an example of decomposed rendering components. Our decomposed module accurately learns different types of color and jointly enable the high-quality shiny object surface.

5.3 Training

The training process is essentially the same as other NeRF methods, but with the above regularization terms. The full loss function is defined as

$$\mathcal{L} = \mathcal{L}_{color} + \lambda_d \mathcal{L}_d + \lambda_b \mathcal{L}_b \tag{6}$$

where $\mathcal{L}_{color}$ is an L1 loss between the ground truth and the rendered images, λ_d, λ_b are loss hyperparameters. In order to feed stable normal vectors to the specular MLP branch and to keep the specular MLP from being impacted by diffuse color, we exclusively train the diffuse MLP branch for a few training steps in the beginning, then jointly train both diffuse and specular branches.

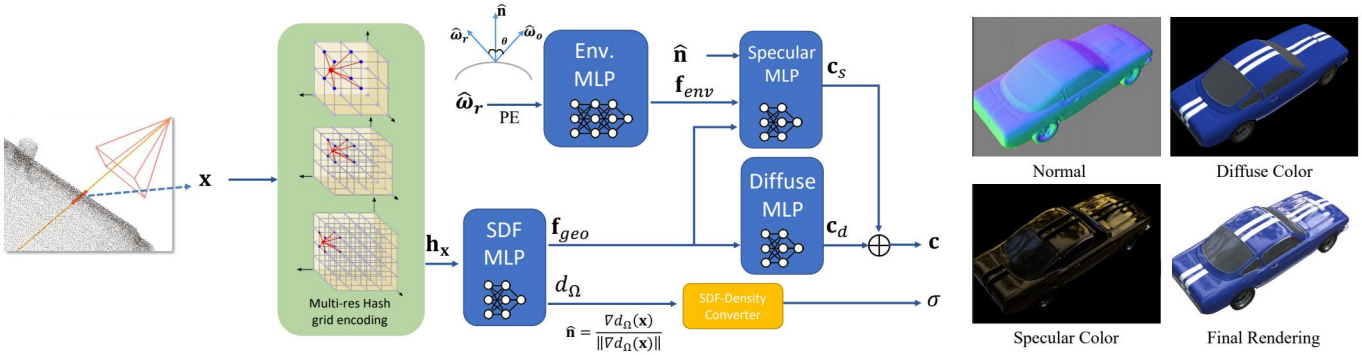
6 Evaluation

We evaluate our method on challenging shiny 3D objects. We make comparisons with prior works based on the quality of view synthesis and surface normal.

Datasets. We use 3 shiny scenes from two datasets for evaluation: 2 NeRF's Blender synthetic scenes [20] (Ficus and Materials) and 1 shiny blender synthetic scene from Ref-NeRF [11].

Baselines. We choose vanilla Instant-NGP[3] [16], VolSDF[4] [13], and Ref-NeRF[5] [26] as baselines. We train or use pre-trained checkpoints to get the results of the baseline.

3. https://github.com/ashawkey/torch-ngp
4. https://github.com/nexuslrf/Accel-RF
5. https://github.com/kakaobrain/NeRF-Factory/



(a) Model overview. (b) Our decomposed components.

Fig. 5. **Model overview.** We use an Instant-NGP like hybrid model with SDF-based volume density (e.g., VolSDF). Our model decomposes color with diffuse and specular color with two shallow MLP branches. Using a relatively large env. MLP with reflected direction as the only input to implicitly learn global illumination from the environment.

Implementation. Our implementation is based on pytorch-ngp[6]. For a fair comparison, we do add a very MLP model in our model shown in Fig. 5(a). The size of diffuse and specular MLP is smaller than Instant-NGP's color MLP. Our env. MLP is only a 4-layer MLP with 128 hidden units per layer. Same as the vanilla Instant-NGP, we train our model for 2000 epochs to report the results.

Evaluated Metrics. We use PSNR, SSIM, and LPIPS to measure the quality of the novel view synthesis. We use MAE (mean angular error) to measure the quality of the estimated surface normal.

6.1 Results Comparison

The results of the comparison are shown in Table 1 and Figure 6. Our method has performance on par with the baselines in terms of PSNR, SSIM, and LPIPS, indicating that our method can produce high-quality synthesized views of shiny objects. Our method also performs well in terms of MAE, achieving the lowest error on the Ficus and Materials scenes.

By jointly comparing the estimated normal and final rendered image. We can see our method does learn the smooth surface and accurate illumination from the environment. Compared to state-of-the-art method Ref-NeRF, our still have relatively lower rendering quality. For example, our result still lacks the high-frequency detail on the metallic ball. Since our environment MLP cannot model the indirect illumination caused by the reflectance from the neighboring objects. Our rendered results do not well preserve the reflectance from the neighbor balls in the materials example. However, this results also indicate that our environment does learn the accurate global illumination from the environment. Another problem of our method is that our estimated surfaces are over-smoothed, this might be related to our relatively large loss weights on the two SDF regularizations terms. We can later improve it by gradually decreasing the loss weights during the training to learn more accurate surface detail.

6. https://github.com/ashawkey/torch-ngp

7 Conclusion

In this paper, we proposed a method that combines hybrid NeRF and SDF-based volume rendering for rendering accurate shiny surfaces. Our method uses Instant-NGP to encode spatial features, and decomposes the color into diffuse and specular components. This allows the model to capture the different lighting effects of different materials in the scene. We also use a MLP model to implicitly learn global illumination from the environment, allowing the model to produce realistic lighting effects in the final rendering.

We evaluated our method on challenging shiny 3D objects and compared it to several baselines. The results showed that our method performed on par with the baselines in terms of PSNR, SSIM, and LPIPS, indicating that our method can produce high-quality synthesized views of shiny objects. Our method also performed well in terms of MAE, achieving the lowest error on the Ficus and Materials datasets. Overall, the results demonstrate the effectiveness of our method for the combination of hybrid NeRF and SDF-based volume rendering. However, our method still has some limitations, such as over-smoothing of the estimated surfaces and lack of high-frequency detail in the final rendering. Future work can focus on improving these limitations and further exploring the combination of hybrid NeRF and SDF-based volume rendering, such as extracting the environment image from our environment encoding MLP, modeling indirect illumination, and image-based relighting.

References

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[4] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3835–3844.

TABLE 1
Quantitative Result Comparison. Note that RefNeRF* are the results reported in the original paper.

	PSNR↑			SSIM↑			LPIPS↓			MAE°↓		
	ficus	materials	car	ficus	materials	car	ficus	materials	car	ficus	materials	car
iNGP	**31.68**	28.07	26.14	**0.988**	0.962	0.939	**0.015**	0.034	0.061	61.35	70.81	73.25
VolSDF	24.26	28.56	27.02	0.957	0.966	0.951	0.059	0.041	0.055	46.58	32.49	**46.92**
RefNeRF	29.88	**33.06**	**30.01**	0.982	**0.987**	0.967	0.021	**0.013**	**0.026**	68.00	42.48	57.19
Ours	29.23	29.44	29.50	0.982	0.971	**0.969**	0.020	0.026	0.036	**41.46**	**21.09**	50.64
RefNeRF*	33.91	35.41	30.82	0.983	0.983	0.955	0.019	0.022	0.041	41.05	9.53	14.93

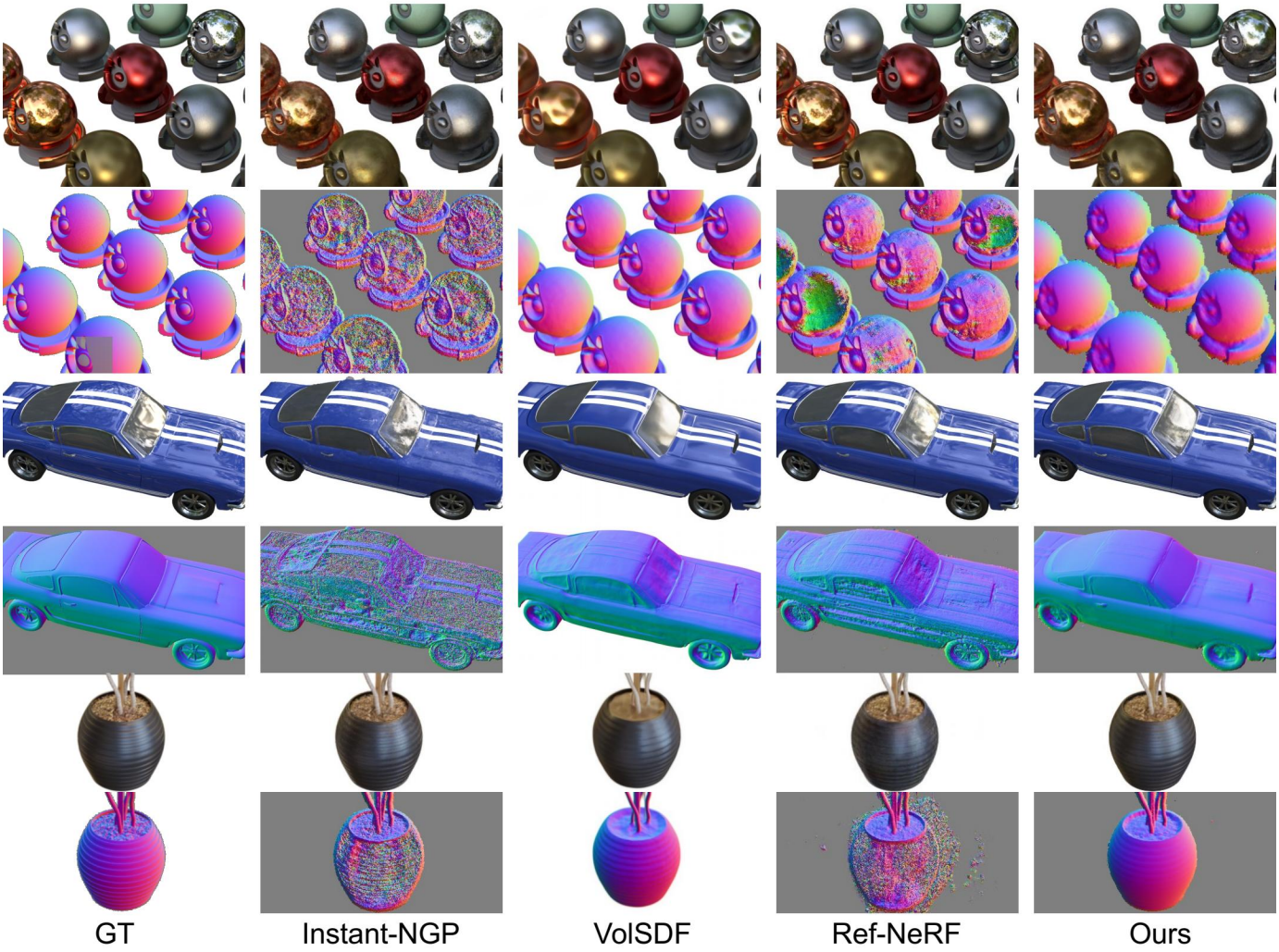


Fig. 6. **Qualitative Result Comparison**

[5] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[6] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[7] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[8] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 684–12 694.

[9] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480496

[10] M. Boss, V. Jampani, R. Braun, C. Liu, J. Barron, and H. Lensch, "Neural-pil: Neural pre-integrated lighting for reflectance decomposition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 691–10 704, 2021.

[11] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," *arXiv preprint arXiv:2112.03907*, 2021.

[12] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2492–2502, 2020.

[13] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Thirty-Fifth Conference on Neural*

Information Processing Systems, 2021.
[14] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *NeurIPS*, 2021.
[15] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. 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*, 2021, pp. 5453–5462.
[16] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *arXiv preprint arXiv:2201.05989*, 2022.
[17] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner *et al.*, "State of the art on neural rendering," in *Computer Graphics Forum*, vol. 39, no. 2. Wiley Online Library, 2020, pp. 701–727.
[18] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.
[19] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, "Efficient geometry-aware 3d generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 123–16 133.
[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.
[21] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
[22] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *NeurIPS*, 2020.
[23] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.
[24] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.
[25] J. Ost, I. Laradji, A. Newell, Y. Bahat, and F. Heide, "Neural point light fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 419–18 429.
[26] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," *arXiv preprint arXiv:2201.08845*, 2022.
[27] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127
[28] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.
[29] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5589–5599.
[30] M. Atzmon and Y. Lipman, "Sal: Sign agnostic learning of shapes from raw data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2565–2574.
[31] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 335–14 345.