Generative Multiview Relighting for 3D Reconstruction under Extreme Illumination Variation

Supplementary Material

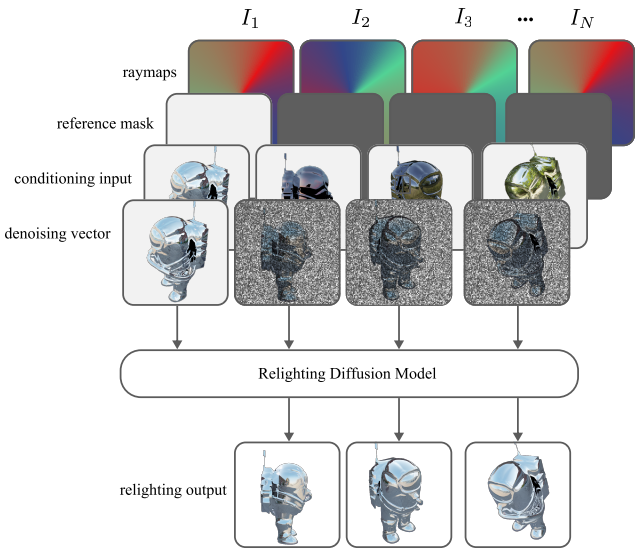

Figure 1. **Relighting model overview.** For each input image, we pass to the model raymaps containing the pose information, a binary mask to highlight which frame is being treated as the reference, and the conditioning images of the object under varying illumination. We concatenate all the inputs to the image being currently denoised. *Note that both the conditioning inputs and target images are the latent encoding of the images, but we show them as images here for simplicity.*

Figure 2. **Extracted illumination from relit images.** We use our relighting model to relight a spherical light probe with varying illumination, and recover the environment maps from the relit images. Notice that while the content of the environemnt map is preserved, we can observe some warping inconsistencies between samples. This motivates our solution of using per-image shading embeddings to accommodate for this type of inconsistencies.

A. Supplemental Webpage and Videos

Please refer to our webpage https://relight-to-reconstruct.github.io to view our video reconstruction results and baseline comparisons on our full synthetic dataset, as well as examples from NAVI. We also show output examples from our relighting model and compare the consistency of our relightings and recent state-of-the-art generative relighting from IllumiNeRF [4].

B. Diffusion Training and Sampling

In Figure 1 we show the detailed inputs of our diffusion model. Although our diffusion model operates in the latent space (so the conditioning and targets are both the latents of the encoded images), we refer to them as images for clarity. The raymaps consist of the ray origins, ray directions, and focal length associated with the image for each pixel. We downscale them from 512×512 to 64×64 to match the dimensions of our latents. The reference mask is a 64×64 binary mask that is 1 for the reference image, and 0 for the images we are relighting. For each denoising step during training and inference, we also pass the latents original images that we want to relight. Note that we do not denoise the reference image, and pass a clean copy of it to the model. While this design redundantly passes the reference image twice, this is needed since the model parameters are shared for all the images.

We train the relighting diffusion model using a DDPM schedule, with beta values that start at $8.5 \cdot 10^{-4}$ and end at $1.2 \cdot 10^{-2}$ increasing linearly over 1024 steps. For objective, we use velocity prediction. During inference, we use DDIM sampling with 50 inference steps. We use a learning rate of 10^{-4}, with 10K warm-up steps. Note that we reset the learning rate schedule each time we fine-tune the model to relight larger number of frames.

C. Shading Embeddings Visualization

To reconstruct a highly reflective object perfectly, we would need the reflections to be exactly consistent across all the input views. Otherwise, any inconsistencies would appear as flickering in the 3D reconstruction. However, we observe that while our relighting model preserves the reflection content, some of the inconsistencies can appear as warping in the reflected environment. In Figure 2, we relight a spherical light probe that we rendered using random poses with

different environment maps. Each of the relit images can be treated as a separate light probe from which we can extract the illumination, and compare the different extracted environment maps in order to visualize the inconsistencies. Here we find that while the content is largely preserved, some objects like the house and the trees suffer from some distortion across the different relit views. This motivates our novel shading embeddings: it allows the model to optimize the radiance field under a single constant illumination condition, while optimizing for per-image surface normals used for rendering reflections.

D. Additional 3D Reconstruction Details

This section provides additional details on our 3D reconstruction approach described in Section 3.2 of the main paper. Our method is based on NeRF-Casting [3] with a few modifications.

First, NeRF-Casting is designed for real scenes without masks. In order to apply it to our setting where background content is masked out with white pixels, we replace its "reflection features", which volume renders a field of features along the reflected ray $\mathbf{o}'+t\mathbf{d}'$, with a single feature queried infinitely far away, *i.e.*, at:

$$\lim_{t\to\infty} \mathcal{C}\left(\mathbf{o}'+t\mathbf{d}'\right) = 2\mathbf{d}', \tag{1}$$

where $\mathcal{C}$ is the contraction function from Zip-NeRF [1]:

$$\mathcal{C}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\| \le 1, \\ \left(2 - \frac{1}{\|\mathbf{x}\|}\right)\frac{\mathbf{x}}{\|\mathbf{x}\|} & \text{if } \|\mathbf{x}\| > 1. \end{cases} \tag{2}$$

This simplifies the derivation in Section 4.2 of NeRF-Casting, so that the feature corresponding to the reflected ray $\mathbf{p}'$ is:

$$\bar{\mathbf{f}} = \frac{1}{K}\sum_{j=1}^{K} \mathbf{f}(2\mathbf{d}'_j) \odot \operatorname{erf}\left((\sqrt{8}\boldsymbol{\nu}\sigma)^{-1}\right), \tag{3}$$

where $\{\mathbf{d}'_j\}_{j=1}^K$ are NeRF-Casting's K unscented reflection directions, $\sigma = 2\gamma(\dot{r}+\bar{\rho})$ is the scaling parameter defined in NeRF-Casting for infinitely distant content, $\boldsymbol{\nu}$ is a vector (with the same dimension as $\mathbf{f}$) containing the scale of NGP grid resolutions, $\odot$ denotes elementwise multiplication. See NeRF-Casting for additional information.

We also make a few additional small modifications to NeRF-Casting's optimization:

1. We optimize our NeRF for 25K iterations rather than 50K. We use the same learning rate schedule as in NeRF-Casting.
2. We initialize density around $\exp(-1)$ instead of $\exp(2)$.
3. We use a faster coarse-to-fine rate: using the notation from Appendix C.1. in [3], we set $m = 16$ and $s = 50$.
4. We remove the view direction as input into the color prediction network.

Finally, for scenes from NAVI [2], which have imprecise camera poses, we found that adding a simple mask loss improved our results. For a ray with rendering weights $\{w_i\}_{i=1}^N$ we use the following loss:

$$\mathcal{L}_{\text{mask}} = \lambda_{\text{mask}} \cdot \left(\sum_{i=1}^{N} w_i - \alpha\right), \tag{4}$$

where $\sum_{i=1}^N w_i$ is the opacity of the ray, and α is 1 for object pixels and 0 for background ones. Since the object masks provided with NAVI are also imprecise, we do not apply the mask loss in Equation 4 to pixels that are within 7 pixels from a boundary. In our NAVI experiments we set $\lambda_{\text{mask}} = 0.01$.

E. Ablation Study Details

Number of Sampled Frames. To compare the effects of the number of frames the model relights simultaneously, we fine-tune our final model to relight 1 frame, 8 frames, 16 frames, and 32 frames at once. Then, using each model, we sequentially relight the entire 64-frame input under the same reference image. Our hypothesis is that with fewer frames, the relit outputs would be less consistent, and exhibit a drop of performance in the reconstruction. This was indeed the case, as showed in Table 3 of the main paper. We find that we gain a significant boost in performance going from single-frame relighting (what prior work follows) to 8-frame relighting, and additional gain as we increase the number of relit frames to 64. However, as expected, we notice diminishing returns where the benefit become more subtle as we increase the number of frames we relight at once.

Dataset Ablation. To investigate the benefit of augmenting our training data with highly reflective materials, we trained two models: one model only on standard assets, and one model where we randomly sample from standard assets, and assets augmented with highly reflective materials. For the sake of efficiency, we only train a 16-frames version of the model, and we train each model for 70K training steps. We find that adding highly reflective assets significantly improves the model's performance on shiny assets, and surprisingly that it also provide a benefit to standard assets. Intuitively, this can be attributed to the fact that shiny assets are significantly more challenging to relight, and are more beneficial to the improvement of the model's performance than diffuse objects.

Shading Embedding Ablations. While in Section C we motivate shading embeddings visually, we demonstrated its importance by comparing our novel shading embeddings

and the standard appearance embeddings that prior work used. As we also show in Section 5 of the main paper and in the supplementary webpage, we show that the typical appearance embeddings are worse than not using any appearance embeddings and training on the relit images directly. This can be attributed to the fact that appearance embeddings can absorb any view-dependent changes that are necessary for realistic reflections, and render a mostly-diffuse object. On the other hand, not using any appearance embeddings can allow the reflections to move naturally along the object, but also include the flickering from the inconsistencies. Our shading embeddings resolve these issues: they can explain away any inconsistencies due to the reflections warping, while preserving the view-dependent effects necessary to render realistic reflections.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2

[2] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, Andre Araujo, Ricardo Martin-Brualla, Kaushal Patel, Daniel Vlasic, Vittorio Ferrari, Ameesh Makadia, Ce Liu, Yuanzhen Li, and Howard Zhou. NAVI: Category-agnostic image collections with high-quality 3D shape and pose annotations. *NeurIPS*, 2023. 2

[3] Dor Verbin, Pratul P. Srinivasan, Peter Hedman, Ben Mildenhall, Benjamin Attal, Richard Szeliski, and Jonathan T. Barron. NeRF-Casting: Improved View-Dependent Appearance with Consistent Reflections. *SIGGRAPH Asia*, 2024. 2

[4] Xiaoming Zhao, Pratul P. Srinivasan, Dor Verbin, Keunhong Park, Ricardo Martin Brualla, and Philipp Henzler. IllumiNeRF: 3D Relighting Without Inverse Rendering. *NeurIPS*, 2024. 1