Differentiable Neural Surface Refinement for Modeling Transparent Objects

Supplementary Material

1. Derivation of the Analytic Derivative

Consider the optimization problem that finds the distance to the first surface intersection (zero level crossing) along the ray from $\mathbf{x}_i$ with direction $\mathbf{d}_i$

$$\begin{aligned} \delta_i(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{d}_i) \in \arg\min_{\delta\in\mathbb{R}} \quad & \delta \\ \text{subject to} \quad & \phi_{\text{SDF}}(\mathbf{x}_i + \delta\mathbf{d}_i; \boldsymbol{\theta}) = 0 \\ & \delta \geqslant \epsilon \end{aligned} \tag{1}$$

for the small constant ϵ, to avoid trivial solutions. Assuming δ_i exists, and noting that the inequality $\delta \geqslant \epsilon$ is inactive, then by Proposition 4.6 from Gould et al. [1] we can compute the derivative as

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}t} = H^{-1}A^{\mathsf{T}}\left(AH^{-1}A^{\mathsf{T}}\right)^{-1}\left(AH^{-1}B - C\right) - H^{-1}B \tag{2}$$

for any parameter t. For scalar A, denoted a, many of the terms cancel out, simplifying the expression to

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}t} = -a^{-1}C_t \tag{3}$$

where

$$a = \frac{\partial}{\partial\delta}\phi_{\text{SDF}}(\mathbf{x}_{i+1}(\delta); \boldsymbol{\theta}) \tag{4}$$

$$= \frac{\partial}{\partial\mathbf{x}_{i+1}}\phi_{\text{SDF}}(\mathbf{x}_{i+1}; \boldsymbol{\theta})\frac{\mathrm{d}\mathbf{x}_{i+1}}{\mathrm{d}\delta} \tag{5}$$

$$= \mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i \tag{6}$$

$$C_{\boldsymbol{\theta}} = \frac{\partial}{\partial\boldsymbol{\theta}}\phi_{\text{SDF}}(\mathbf{x}_{i+1}; \boldsymbol{\theta}) \tag{7}$$

$$C_{\mathbf{x}_i} = \frac{\partial\phi_{\text{SDF}}(\mathbf{x}_{i+1}; \boldsymbol{\theta})}{\partial\mathbf{x}_{i+1}}\frac{\mathrm{d}\mathbf{x}_{i+1}}{\mathrm{d}\mathbf{x}_i} = \mathbf{n}_{i+1} \tag{8}$$

$$C_{\mathbf{d}_i} = \frac{\partial\phi_{\text{SDF}}(\mathbf{x}_{i+1}; \boldsymbol{\theta})}{\partial\mathbf{x}_{i+1}}\frac{\mathrm{d}\mathbf{x}_{i+1}}{\mathrm{d}\mathbf{d}_i} = \mathbf{n}_{i+1}\delta_i, \tag{9}$$

where the intersection point is $\mathbf{x}_{i+1} = \mathbf{x}_i + \delta_i\mathbf{d}_i$ and its normal vector is $\mathbf{n}_{i+1} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\phi_{\text{SDF}}(\mathbf{x}_{i+1})$. Then the required derivatives are given by

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\boldsymbol{\theta}} = -\frac{1}{\mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i}\frac{\partial}{\partial\boldsymbol{\theta}}\phi_{\text{SDF}}(\mathbf{x}_{i+1}; \boldsymbol{\theta}) \tag{10}$$

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\mathbf{x}_i} = -\frac{\mathbf{n}_{i+1}}{\mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i} \tag{11}$$

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\mathbf{d}_i} = -\frac{\delta_i\mathbf{n}_{i+1}}{\mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i}, \tag{12}$$

which amounts to implicit differentiation of $\phi_{\text{SDF}} = 0$. For ease of implementation, we define the function value of the SDF as $s_i = \phi_{\text{SDF}}(\mathbf{x}_i + \delta_i\mathbf{d}_i; \boldsymbol{\theta})$ and note that

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}\delta_i}{\mathrm{d}s_i}\frac{\partial s_i}{\partial\boldsymbol{\theta}} \tag{13}$$

$$\therefore \frac{\mathrm{d}\delta_i}{\mathrm{d}s_i} = -\frac{1}{\mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i}, \tag{14}$$

where Eq. (14) follows from Eqs. (10) and (13). Using this derivative with respect to the SDF outputs means that PyTorch Autograd can handle backpropagation through the SDF itself.

These derivatives work for any number of sequential refractions, where the next direction vector is computed by Snell's Law

$$\mathbf{d}_{i+1} = \frac{\eta_i}{\eta_{i+1}}\mathbf{d}_i - \mathbf{n}_{i+1}\left(\frac{\eta_i}{\eta_{i+1}}\mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i + \right. \tag{15}$$

$$\left.\sqrt{1 - \frac{\eta_i^2}{\eta_{i+1}^2}\left(1 - (\mathbf{n}_{i+1}^{\mathsf{T}}\mathbf{d}_i)^2\right)}\right), \tag{16}$$

where η_i is the refractive index of the material at point $\mathbf{x}_i + \epsilon\mathbf{d}_i$ (perturbed to be off the surface), which is 1.0003 for air. This can be implemented in PyTorch by defining an Autograd function, shown in the code listing below, which allows PyTorch to handle backpropagation through the SDF parameters.

Python Code

```python
class DistToIntersection(torch.autograd.Function):
 def forward(ctx, si, xi, di, nj, deltai):
  ctx.save_for_backward(di, nj, deltai)
  return deltai
 def backward(ctx, grad_output):
  di, nj, deltai = ctx.saved_tensors
  return -grad_output / dot(nj, di),
   -grad_output * nj / dot(nj, di),
   -grad_output * deltai * nj / dot(nj, di),
   None, None
```

Note that we do not backpropagate through the normal vectors, since this involves the computation of second derivatives, which are very noisy and numerically unstable. This is implemented by a stop gradient on the normal tensors to remove them from the computation graph.

2. Comparison against SampleNeRFRO

We compare our approach with SampleNeRFRO [3] using *Optical Ball*. SampleNeRFRO, under the assumption of known geometry and refractive index, employs a voxelization technique to discretize the scene, storing the refractive index within each voxel. Subsequently, it relies on the Eikonal equation [2] for computing the refracted ray

Figure 1. Qualitative Comparison of View Synthesis Results on *Optical Ball*. Our method (middle) demonstrates superior alignment with ground truth compared to SampleNeRFRO [3] (right).

paths. It is important to note that SampleNeRFRO, unlike our method, does not explicitly account for reflection. While SampleNeRFRO is capable of modeling refraction to some extent, it is apparent that it falls short in delivering sharp and clear novel views. In contrast, our method consistently generates visually plausible and more coherent novel view results. Moreover, our method gains higher PSNR than SampleNeRFRO (24.07 versus 19.07).

Table 1 provides a direct comparison between our proposed model and the SampleNeRFRO approach, employing PSNR as the metric to assess the quality of novel view synthesis. The results demonstrate that our model outperforms SampleNeRFRO across four datasets. A key distinction of our model is its ability to attain superior performance without relying on known geometric or refractive index. This is a notable enhancement compared to the SampleNeRFRO model, which necessitates such information to perform piecewise-linear curved ray calculations as per the Eikonal equation, referenced in [2].

Model	↑ PSNR			
	Optical Ball	Bottle	Ball	Glass
SampleNeRFRO [3]	19.07	12.90	21.49	21.11
Ours	**24.07**	**23.20**	**21.70**	**21.30**

Table 1. Comparison with SampleNeRFRO. Our method achieves higher performance across four datasets without the need for known geometry or refractive index assumptions, unlike SampleNeRFRO [3] which relies on such information for curved ray calculations based on the Eikonal equation [2].

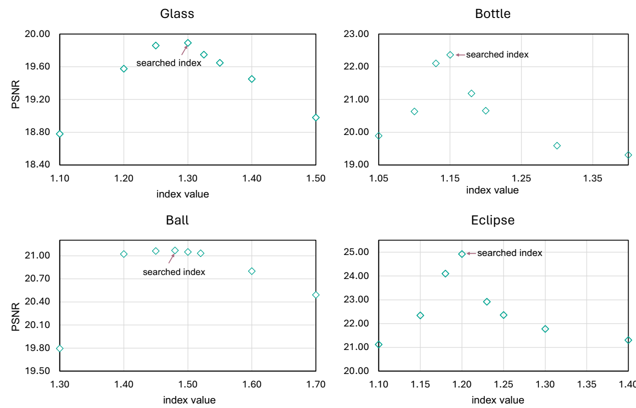

Figure 2. Refractive index search. We present search results from four transparent datasets. We search for the refractive index that enhances PSNR in novel views. Note that, this process demands no extra training, as it relies solely on rendering images with the pre-trained NeuS model while adjusting refractive indices.

3. Refractive Index Search

In addition to the search results showcased for *Optical Ball* in the main paper, we provide results for four other datasets in Figure 2. Our selection process involves identifying the refractive index that optimizes the PSNR for novel views, utilizing the pre-trained NeuS model. Notably, the refractive indices determined for the *Bottle* and *Eclipse* datasets correspond with the ground-truth values. However, for the *Glass* and *Ball* datasets, which are real-world datasets, the ground-truth refractive indices remain unknown.

References

[1] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):3988–4004, 2021. 1

[2] Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Eikonal rendering: Efficient light transport in refractive objects. *ACM Trans. Graph.*, 26(3):59–es, 2007. 1, 2

[3] Jen-I Pan, Jheng-Wei Su, Kai-Wen Hsiao, Ting-Yu Yen, and Hung-Kuo Chu. Sampling neural radiance fields for refractive objects. In *SIGGRAPH Asia 2022 Technical Communications*, pages 1–4. 2022. 1, 2