Exact-NeRF: An Exploration of a Precise Volumetric Parameterization for Neural Radiance Fields - Supplementary Material

Brian K. S. Isaac-Medina[1], Chris G. Willcocks[1], Toby P. Breckon[1, 2]
Department of {[1]Computer Science, [2]Engineering}, Durham University, UK
{brian.k.isaac-medina, christopher.g.willcocks, toby.breckon}@durham.ac.uk

A. Additional Formulation

In this section, we will present some additional formulations used in our model that do not affect the results presented in the paper. Notwithstanding that our proposed EIPE works for any polyhedron (which is the reason why it can be used under the mip-NeRF 360 [2] architecture without further treatment), we also present an alternative formulation of the EIPE for the particular case of strictly square pyramids.

A.1. Indeterminate cases for the EIPE

By simplifying Eq. (22) we obtain:

$$\sigma_{x,\tau} = \frac{\begin{array}{r}(x_{\tau,2}-x_{\tau,1})\cos(2^l x_{\tau,0})\\ +(x_{\tau,0}-x_{\tau,2})\cos(2^l x_{\tau,1})\\ +(x_{\tau,1}-x_{\tau,0})\cos(2^l x_{\tau,2})\end{array}}{2^{2l}(x_{\tau,1}-x_{\tau,0})(x_{\tau,2}-x_{\tau,0})(x_{\tau,2}-x_{\tau,1})}\,. \tag{29}$$

From Eq. (29) we observe that an indetermination occurs for the case of two points in the triangle τ sharing the same coordinate, such that $x_{\tau,i} = x_{\tau,j}, i \neq j$. In order to get a valid value for these cases, we get the limit when those two coordinates approach. We can write Eq. (29) as:

$$\sigma_{x,\tau} = \frac{f(x_{\tau,0},x_{\tau,1},x_{\tau,2})}{2^{2l}g(x_{\tau,0},x_{\tau,1},x_{\tau,2})}\,. \tag{30}$$

Then, we obtain the value for the case of $x_{\tau,0} = x_{\tau,1}$ using *l'Hopital's rule*:

$$\lim_{x_{\tau,0}\to x_{\tau,1}} \frac{f(x_{\tau,0},x_{\tau,1},x_{\tau,2})}{2^{2l}g(x_{\tau,0},x_{\tau,1},x_{\tau,2})} = \tag{31}$$

$$\lim_{x_{\tau,0}\to x_{\tau,1}} \frac{\frac{\partial}{\partial x_{\tau,0}}f(x_{\tau,0},x_{\tau,1},x_{\tau,2})}{2^{2l}\frac{\partial}{\partial x_{\tau,0}}g(x_{\tau,0},x_{\tau,1},x_{\tau,2})} = \tag{32}$$

$$\lim_{x_{\tau,0}\to x_{\tau,1}} \frac{\begin{pmatrix}-2^l(x_{\tau,2}-x_{\tau,1})\sin(2^l x_{\tau,0})\\ +\cos(2^l x_{\tau,1})-\cos(2^l x_{\tau,2})\end{pmatrix}}{2^{2l}(x_{\tau,2}-x_{\tau,1})(2x_{\tau,0}-x_{\tau,1}-x_{\tau,2})} = \tag{33}$$

$$\frac{2^l(x_{\tau,2}-x_{\tau,1})\sin(2^l x_{\tau,1})-\cos(2^l x_{\tau,1})+\cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2}-x_{\tau,1})^2}\,. \tag{34}$$

Similarly, from Eq. (33), we evaluate the case $x_{\tau,0} = x_{\tau,2}$:

$$\frac{-2^l(x_{\tau,2}-x_{\tau,1})\sin(2^l x_{\tau,2})+\cos(2^l x_{\tau,1})-\cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2}-x_{\tau,1})^2}\,. \tag{35}$$

For the case when $x_{\tau,1} = x_{\tau,2}$, we differentiate with respect to $x_{\tau,1}$ to obtain the corresponding value:

$$\lim_{x_{\tau,1}\to x_{\tau,2}} \frac{f(x_{\tau,0},x_{\tau,1},x_{\tau,2})}{2^{2l}g(x_{\tau,0},x_{\tau,1},x_{\tau,2})} = \tag{36}$$

$$\lim_{x_{\tau,1}\to x_{\tau,2}} \frac{\begin{pmatrix}-\cos(2^l x_{\tau,0})+\cos(2^l x_{\tau,2})\\ +2^l(x_{\tau,2}-x_{\tau,0})\sin(2^l x_{\tau,1})\end{pmatrix}}{2^{2l}(x_{\tau,2}-x_{\tau,0})(x_{\tau,0}+x_{\tau,2}-2x_{\tau,1})} = \tag{37}$$

$$\frac{-2^l(x_{\tau,2}-x_{\tau,0})\sin(2^l x_{\tau,1})+\cos(2^l x_{\tau,0})-\cos(2^l x_{\tau,2})}{2^{2l}(x_{\tau,2}-x_{\tau,0})^2}\,. \tag{38}$$

Finally, when $x_{\tau,0} = x_{\tau,1} = x_{\tau,2}$, we use again the *l'Hopital's rule* on Eq. (33) and differentiate again with respect to $x_{\tau,0}$ to obtain:

$$\lim_{x_{\tau,0}\to x_{\tau,1}\to x_{\tau,2}} \sigma_{x,\tau} = -\frac{1}{2}\cos(2^l x_{\tau,0})\,. \tag{39}$$

Using the same approach, we can find the following expressions for $\xi_{x,\tau}$ (Eq. (27)):

$$\lim_{x_{\tau,0}\to x_{\tau,1}} \xi_{x,\tau} = \frac{\begin{pmatrix}2^l(x_{\tau,2}-x_{\tau,1})\cos(2^l x_{\tau,1})\\ +\sin(2^l x_{\tau,1})-\sin(2^l x_{\tau,2})\end{pmatrix}}{2^{2l}(x_{\tau,2}-x_{\tau,1})^2} \tag{40}$$

$$\lim_{x_{\tau,0}\to x_{\tau,2}} \xi_{x,\tau} = \frac{\begin{pmatrix}-2^l(x_{\tau,2}-x_{\tau,1})\cos(2^l x_{\tau,2})\\ -\sin(2^l x_{\tau,1})+\sin(2^l x_{\tau,2})\end{pmatrix}}{2^{2l}(x_{\tau,2}-x_{\tau,1})^2} \tag{41}$$

$$\lim_{x_{\tau,1}\to x_{\tau,2}} \xi_{x,\tau} = \frac{\begin{pmatrix}-2^l(x_{\tau,2}-x_{\tau,0})\cos(2^l x_{\tau,2})\\ -\sin(2^l x_{\tau,0})+\sin(2^l x_{\tau,2})\end{pmatrix}}{2^{2l}(x_{\tau,2}-x_{\tau,0})^2} \tag{42}$$

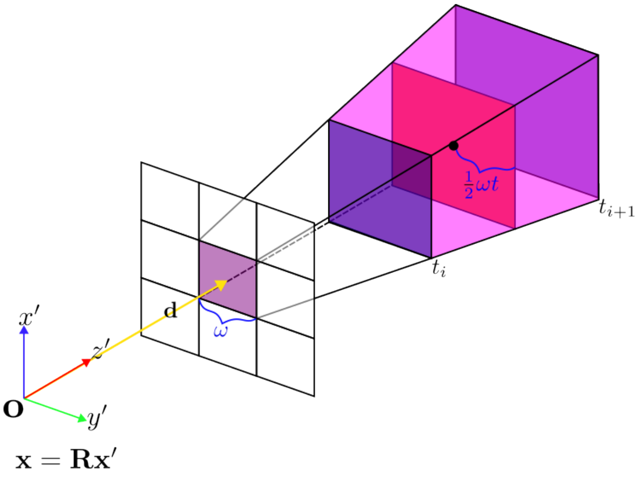


Figure 7. Parameterization of the square pyramid using the pixel width ω.

$$\lim_{x_{\tau,0} \to x_{\tau,1} \to x_{\tau,2}} \xi_{x,\tau} = \frac{1}{2} \sin(2^l x_{\tau,0}) \,. \tag{43}$$

Similar expressions can be obtained for the y and z coordinates.

A.2. Alternative EIPE for Squared Pyramids

As mentioned earlier, our EIPE in Eq. (28) can be used for any shape whose vertices are known. However, the computational cost increases if the 3D shape is complex since a larger number of triangular faces will need to be processed. For more efficient methods, we can focus our analysis on specific shapes. Particular to our scenario, we can obtain an alternative EIPE exclusively for a square pyramid (note that this will not be the case for the contraction function in mip-NeRF 360) with a known camera pose $[\mathbf{R}|\mathbf{o}]$ and pixel width ω (similar to $\dot{r}$ in mip-NeRF). From Fig. 7, we calculate the volume of the frustum as

$$V = \int_{t_i}^{t_{i+1}} \int_{-\omega z/2}^{\omega z/2} \int_{-\omega z/2}^{\omega z/2} dx' dy' dz' \tag{44}$$

$$V = \frac{\omega^2}{3} \left(t_{i+1}^3 - t_i^3 \right) \,. \tag{45}$$

The numerator in Eq. (6) for the x coordinate can be obtained in the same way:

$$I_x = \int_{t_i}^{t_{i+1}} \int_{-\omega z/2}^{\omega z/2} \int_{-\omega z/2}^{\omega z/2} \sin(2^l x) dx' dy' dz' \,. \tag{46}$$

Since the camera pose is known, we can express x as

$$x = r_{11} x' + r_{12} y' + r_{13} z' + o_1 \,, \tag{47}$$

where r_{ij} is an element of the rotation matrix $\mathbf{R}$ and o_1 is the first element of $\mathbf{o}$. Substituting Eq. (47) in Eq. (46) (and omitting the integration limits for clarity):

$$I_x = \iiint \sin(2^l (r_{11} x' + r_{12} y' + r_{13} z' + o_1)) dx' dy' dz' \,. \tag{48}$$

The solution to the integral in Eq. (48) is then:

$$I_x = \frac{1}{2^{3l} r_{11} r_{12}} \left[\frac{C_1}{\zeta_1} - \frac{C_2}{\zeta_2} - \frac{C_3}{\zeta_3} + \frac{C_4}{\zeta_4} \right] \,, \tag{49}$$

$$C_j = \cos\left(2^l (t_{i+1} \zeta_j + o_1)\right) - \cos\left(2^l (t_i \zeta_j + o_1)\right) \,, \tag{50}$$

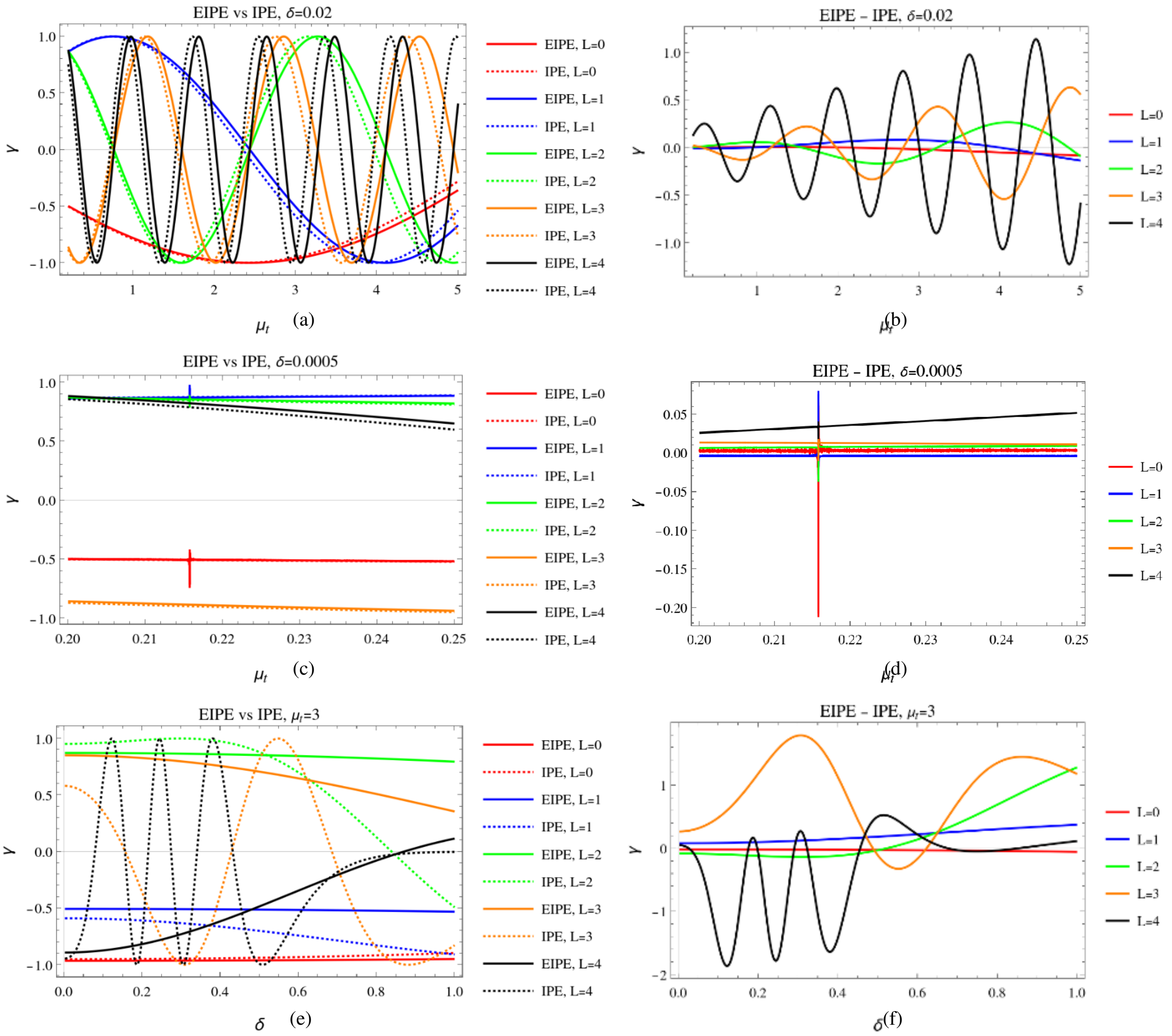
$$\zeta_j = \boldsymbol{\eta}_j^\top \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \end{bmatrix} \,, \tag{51}$$

$$\boldsymbol{\eta}_1 = \begin{bmatrix} \frac{\omega}{2} \\ \frac{\omega}{2} \\ 1 \end{bmatrix}, \boldsymbol{\eta}_2 = \begin{bmatrix} -\frac{\omega}{2} \\ \frac{\omega}{2} \\ 1 \end{bmatrix}, \boldsymbol{\eta}_3 = \begin{bmatrix} \frac{\omega}{2} \\ -\frac{\omega}{2} \\ 1 \end{bmatrix}, \boldsymbol{\eta}_4 = \begin{bmatrix} -\frac{\omega}{2} \\ -\frac{\omega}{2} \\ 1 \end{bmatrix} \,. \tag{52}$$

Similarly to the EIPE in Eq. (28), an indeterminate value arises in Eq. (49) for $r_{11} = 0$ and $r_{12} = 0$. For these cases, *l'Hopital's rule* can be used as in Sec. A.1 or Eq. (48) can be solved by substituting $r_{11} = 0$ and $r_{12} = 0$. We omit these calculations for brevity.

B. Numerical Analysis between IPE and EIPE

We compare the exact value of the EIPE with the approximation in Eq. (7) used by mip-NeRF [1]. In Fig. 8a we contrast the value of the EIPE vs the IPE for frustums of length $\delta_i = 0.02$ at different positions along the ray $\mathbf{d}$ and at different positional encoding frequencies L. The values of $\mathbf{d}$, $\mathbf{o}$ and $\mathbf{R}$ correspond to a random pixel of a random image of the blender dataset. It is seen that the approximation is precise for frustums that are near the camera (small μ_t), but it degrades the further it gets. It is also observed that this effect grows faster for larger values of L. This trend is more noticeable in the plot of the error between the EIPE and IPE (Fig. 8b), where the magnitude of the error is a periodic function approximately bounded by two lines whose pendant seems to grow proportional with L. Furthermore, it is observed that the frequency of the error is also proportional to L. Figs. 8c and 8d show a similar analysis for small values of μ_t and $\delta_i = 5 \times 10^{-4}$, which correspond to small frustums. In these instances, it is observed that numerical errors occur, which is consistent with the analysis of the *Impact of Numerical Underflow* in Sec. 5. A similar analysis for a fixed value of $\mu_t = 3$ and varying δ_i is shown in Figs. 8e and 8f. Here, a more drastic error is seen when δ_i increases, which is consistent with the observation made in [2] that the IPE does not approximate well for very elongated Gaussians. Additionally, rapid changes in the IPE are observed for small variations in the length of the frustum (see Fig. 8e IPE $L = 3$ and IPE $L = 4$), which might not be desired. On the other hand, our EIPE is more robust to



(a) (b) (c) (d) (e) (f)

Figure 8. Numerical comparison between the IPE and our EIPE. (a) EIPE vs IPE for different values of μ_t and (b) their difference. (c) EIPE vs IPE with respect to the length of the frustum δ_i and (d) their difference.

these elongations, meaning that it could be a more reliable parameterization for distant objects.

Despite the increasing error in the approximation of the IPE for larger values of L, this effect gets mitigated by the nature of the IPE itself, which gives more importance to the components of the positional encoding with smaller frequencies. However, in scenarios with distant backgrounds where more elongated frustum arise, such as in the bicycle scene, Exact-NeRF seems to perform better (Sec. 5). Given that the scenes in the blender and mip-NeRF 360 datasets are composed of one central object only, it is difficult to evaluate the performance of the IPE and EIPE formulations for distant objects or scenarios with several objects.

C. Additional Results on the Blender Dataset

We present more qualitative comparisons in Fig. 9 between different scenes of the blender dataset. The reconstructions of both mip-NeRF and Exact-NeRF are almost identical, but a few differences can be noted, *e.g.*, the apron of the chair and the holes in the lego scene are slightly sharper in our reconstruction; the details in the cymbals of the drum are more similar with the ground truth; the reconstruction of the water in the ship scene is more accurate with our method. Besides these minimal differences, our exploratory work demonstrates that analytical solutions to a volumetric positional encoding exist if the shape of the frustum is changed.

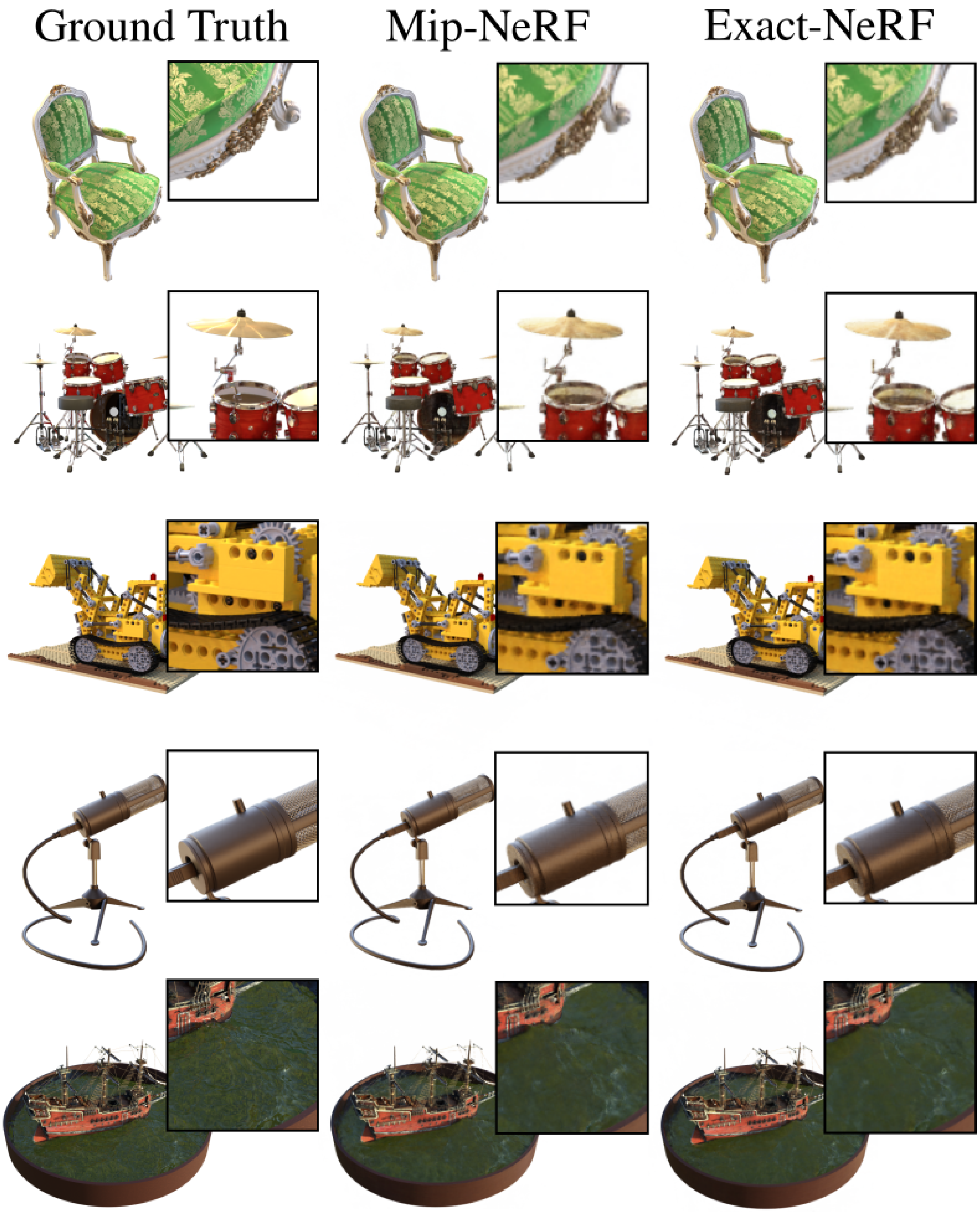


Figure 9. Additional results of Exact-NeRF for the blender dataset.

D. Limitation of Existing Metrics

Following the approach from previous NeRF research, we report PSNR, SSIM and LPIPS as our evaluation metrics. PSNR and SSIM are two of the first evaluation metrics for image reconstruction. Traditionally, PSNR (based on the MSE metric) has been used to assess the quality of lossy compression algorithms. Since the PSNR is obtained via the pixel-wise absolute error, it cannot measure the structural and/or perceptual similarity between the reconstructed and reference images. SSIM was proposed as an alternative metric since it quantifies the relation between the pixels and their neighbourhood (*i.e*., the structural information). Several works have focused on the weakness of these metrics [S3–S7], where the main criticism is that images subject to different compression artifacts and distortion effects (such as additive Gaussian blurring) exhibit similar PSNR and SSIM values. Additional work [S2] has shown analytical and experimental relations between both metrics, meaning that they are not independent. In order to overcome these effects, recent image quality assessment methods have been proposed. Ding *et al*. [S1] have carried out a comprehensive comparison between different metrics, where deep neural networks-based metrics such as LPIPS [33] and DISTS [7] showed to be the most reliable quality metrics for perceptual similarity. These metrics compare two images by measuring the distance of their feature maps from a pretrained neural network. These results motivated us to include the DISTS metric in our experiments (Tabs. 1 and 2). Our method obtains a better performance in the LPIPS and DISTS metrics, thus improving the perceptual quality.

Supplementary Material References

[S1] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Comparison of full-reference image quality models for optimization of image processing systems. *International Journal of Computer Vision*, 129(4):1258–1281, 2021. 5

[S2] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. 5

[S3] Zoran Kotevski and Pece Mitrevski. Experimental comparison of psnr and ssim metrics for video quality estimation. In Danco Davcev and Jorge Marx Gómez, editors, *ICT Innovations 2009*, pages 357–366, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 5

[S4] Peter Ndajah, Hisakazu Kikuchi, Masahiro Yukawa, Hidenori Watanabe, and Shogo Muramatsu. Ssim image quality metric for denoised images. In *Proc. 3rd WSEAS Int. Conf. on Visualization, Imaging and Simulation*, pages 53–58, 2010. 5

[S5] Mahmood Sharif, Lujo Bauer, and Michael K Reiter. On the suitability of lp-norms for creating and preventing adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1605–1613, 2018. 5

[S6] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[S7] Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. 5