RGB-D NeRF: Depth supervised NeRF on synthetic depth maps

Markus Pobitzer
ETH Zürich
pobmarku@student.ethz.ch
Supervised by Dr. Sergey Prokudin

Abstract

Virtual reality (VR) and augmented reality (AR) immerse the user in a new digital world. However, representing real-world scenes and objects digitally is very challenging. Realistic lighting and high details are hard to model. An approach that solves some of the mentioned shortcomings was introduced with Representing Scenes as Neural Radiance Fields for View Synthesis (NeRF). NeRF can produce photorealistic novel views but needs many RGB input images to train. In this work, we explore how NeRF can be extended with synthetic depth information to reduce the needed number of input images.

1. Introduction

The utilization of NeRF [6] as a technique for creating implicit representations of a scene has gained significant attention in the field of computer vision. With sufficient training data, NeRF is capable of generating highly detailed and realistic representations of a scene, complete with accurate lighting. These representations can then be utilized to digitally reconstruct and render novel views of the scene. This approach has the potential to facilitate a wide range of novel interactions with real-world environments.

As an illustrative example, consider an operating room scenario, where a digital representation of the operating room and surgical procedures can help medical students to better understand and engage with the work of surgeons. Or it can also be used to enable remote assistance in surgeries.

However, there still remain several significant challenges that must be overcome to make real-world NeRF-based applications a reality. One hurdle is the large number of input images required for training a NeRF model. While it is ideal to have as many views as possible, in practice it can be infeasible to gather the large number of images typically required for optimal results. In this work, we investigate whether the incorporation of depth maps into the training process can serve to mitigate this challenge by enabling the use of fewer training views.

2. Related Work

Novel view synthesis has seen a drastic increase in interest mainly due to the proposed method called NeRF which produces photorealistic images. The applications of such a method are of particular interest for virtual reality and augmented reality use cases where we want to render a scene or object from every possible view. NeRF still suffers from some shortcomings that followup-work tried to address. To mention some, a reduced amount of input images are needed in [11, 2, 3], and faster training time was achieved in [2, 1, 3, 9].

The authors of NeRF suggest taking around 100 input images for the best performance. Even though the method works also with 25 input images [7] it completely fails when we further reduce it [11].

DS-NeRF (Depth-supervised NeRF): Fewer Views and Faster Training for Free [2]. In DS-NeRF the authors propose a depth-supervised method that builds on Sparse 3D Points obtained from an SfM (Structure from Motion) pipeline. While estimating the camera poses with SfM it is possible to simultaneously obtain sparse 3D Points from the scene. Given the depth estimation from SfM D and the rendered ray termination distance $w(t)$ they compute the KL divergence between these two parts. The depth estimation $N(D, \hat{\sigma})$ is modeled as a Normal distribution around D and the corresponding reprojection error $\hat{\sigma}$. The final loss is a combination of the color loss introduced in [6] and the KL divergence. They have shown that consistent RGB and depth renderings were achievable with only 2 training views for real forward-facing scenes.

RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs [8]. In RegNeRF no ground truth depth prior is used for training. However, the authors propose to sample unobserved views from the scene and regularize the estimated geometry. Their assumption is that in the real world, most surfaces are smooth and

therefore neighboring pixels have similar depth values. To make sure that the network outputs plausible images for these unobserved viewpoints, they also introduce an appearance regularization. The final loss is a combination of the color loss on the training images as introduced in NeRF and the introduced geometry plus appearance loss on unobserved poses. In the work they have shown, that floating artifacts that appear in NeRF when using sparse input views are drastically reduced. That is a direct benefit of the more consistent geometry.

Neural Radiance Fields from Sparse RGB-D Images. Neural Radiance Fields from Sparse RGB-D Images for High-Quality View Synthesis [12]. In this work, the authors propose to first reconstruct a coarse mesh of the scene from sparse RGB-D inputs. From which synthetic images can be produced to pre-train a NeRF network. In this manner, many training images can be rendered from sparse input views. The initial RGB images are used to fine-tune the network. In their experiment, they have shown, that 3D scenes can be captured well starting with only 6 RGB-D input images.

3. Method

3.1. Neural Radiance Field revisited

In a Neural Radiance Field (NeRF) a scene is represented implicitly through a function. The input of the function is a 5D vector, representing the 3D coordinate $x = (x, y, z)$ in the scene and a 2D viewing direction d that is relevant to the lightning in the scene. The function outputs a color c and a volume density σ. As a result, we have a function $f(x, d) = (c, \sigma)$.

This method can be used to implicitly represent a complete scene. Given several training images and the corresponding camera parameters it is possible to train the network by shooting rays through the scene and comparing the color from the image with the predicted color from a ray.

Estimating Color. To estimate the color of a ray r we have to integrate over all points t along the ray, where t corresponds to the distance from the ray origin, $r(t)$ corresponds then to a 3-D location in the scene determined by the ray r and the distance t from the rays origin. We infer the network about the density of the current point $\sigma(r(t))$ as also the color at the position $c(r(t), d)$. Where d corresponds to the viewing direction that is important for the lightning effects in the scene. The color of a ray gets computed by

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt \tag{1}$$

Where $T(t)$ corresponds to the probability that a ray travels from t_n to t without hitting an object.

$$T(t) = exp\left(-\int_{t_n}^{t} \sigma(r(s))ds\right) \tag{2}$$

Since we have to work in discrete space we approximate the integral by a sum and sample a finite number of points N along the ray.

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))c_i \tag{3}$$

where

$$T_i = exp\left(-\sum_{j=1}^{i-1} \sigma_i\delta_j\right). \tag{4}$$

$\delta_j = t_{j+1} - t_j$ corresponds to the distance between two adjacent samples where t_j is the distance of the $j - th$ sample from the ray origin.

Ray termination. With function T_i and the estimated density σ_i of a point i on the ray, we can estimate the probability that the ray terminates at point i

$$w_i = T_i(1 - \exp(-\sigma_i\delta_i)). \tag{5}$$

The color estimate is than $\hat{C}(r) = \sum_{i=1}^{N} w_i c_i$.

Estimating depth. The depth of a ray corresponds to the distance from the ray's origin to the point where the ray terminates, i.e. the first time it hits an object in the scene. The depth estimate can be written as

$$\hat{z}(r) = \sum_{i=1}^{N} w_i t_i. \tag{6}$$

3.2. Supervision

Given RGB-D input views we can supervise the training of the network as follows. We create the set of possible rays R that is defined by the training views. For the color we have

$$L_C = \frac{1}{|R|} \sum_{r\in R} ||\hat{c}(r) - c_{gt}(r)||^2$$

where c_{gt} corresponds to the ground truth color value of the ray. For the depth supervision, we have

$$L_D = \frac{1}{|R|} \sum_{r\in R} ||\hat{z}(r) - z_{gt}(r)||^2$$

where c_{gt} corresponds to the depth of the ray. The final loss is the combination of color and depth supervision

$$L = L_C + \lambda_d L_D$$

In our experiments, we set $\lambda_d = 0.1$.

3.3. Synthetic Depth Maps

Since the used datasets did not contain ground truth depth maps we created synthetic depth maps. To obtain the depth maps used for our RGB images we let a NeRF model train on the full training set and used the predicted depth as our depth map. The NeRF model was trained for 200k iterations.

4. Experiments

Dataset. We report results from the scene "Fern" of the real-world forward-facing dataset LLFF and "Ship" of the synthetic dataset Blender, both introduced in [5]. For LLFF we follow the community standards [6, 8] of using every 8-th image for our test set and selecting the training views evenly from the remaining images. The Blender dataset has already a train/test split and we choose the training images evenly from the training set.

Metrics To evaluate the RGB images we use the PSNR, SSIM [10], and LPIPS [13] scores. To evaluate the depth outputs we use the mean absolute relative difference

$$\frac{1}{|N|}\sum_{z\in N}\frac{|z-z_{gt}|}{z_{gt}}. \tag{7}$$

And the mean squared error (MSE)

$$\frac{1}{|N|}\sum_{z\in N}||z-z_{gt}||^2. \tag{8}$$

Baseline. The baseline is NeRF where we used the base code from DS-NeRF to train the model.

Training. Our experiments were conducted on an NVIDIA GeForce RTX 2080 Ti. We trained the networks for 50k iterations with a learning rate of $5e-4$ and Adam [4] as our optimizer. The batch size is set to 1024 (fetched rays per gradient step), 64 samples per ray, and 128 additional samples around the estimated ray termination point.

4.1. LLFF - Fern

On this dataset, we trained on the scene "Fern". A qualitative comparison of sparse input views with NeRF can be seen in Figure 1. The evaluation based on the RGB output can be seen in Table 4.1 for the PSNR score, Table 4.1 for the SSIM score, and Table 4.1 for the LPIPS metric. In Table 4.1 and Table 4.1 the comparison of the depth output can be found.

RegNeRF. For the comparison with RegNeRF, we evaluated the outputs provided by the authors for 3 and 6 input views.

DS-NeRF. For the comparison to DS-NeRF, we evaluated the network with 2 input views where the training data was given by the authors. More training samples were not given by the authors and due to time reasons, we did not evaluate other input views. In the paper, there are comparisons for an average based on all LLFF scenes.

PSNR ↑

# Views	2	3	5	6	12
NeRF	16.99	20.33	23.47	24.17	25.31
DS-NeRF	<u>20.62</u>	-	-	-	-
RegNerf	-	19.62	-	24.28	-
Ours	19.10	<u>21.06</u>	<u>23.55</u>	<u>24.41</u>	<u>25.43</u>

Table 1. PSNR ↑: evaluation of LLFF - Fern. Comparison of the PSNR scores of the test set on 2, 3, 5, 6, and 12 training views.

SSIM ↑

# Views	2	3	5	6	12
NeRF	0.438	0.628	0.723	0.752	0.786
DS-NeRF	<u>0.579</u>	-	-	-	-
RegNerf	-	0.595	-	<u>0.776</u>	-
Ours	0.569	<u>0.646</u>	0.724	0.757	0.786

Table 2. SSIM ↑: evaluation of LLFF - Fern. Comparison of the SSIM scores of the test set on 2, 3, 5, 6, and 12 training views.

LPIPS ↓

# Views	2	3	5	6	12
NeRF	0.494	0.354	<u>0.287</u>	0.258	<u>0.244</u>
DS-NeRF	0.461	-	-	-	-
RegNerf	-	0.346	-	<u>0.201</u>	-
Ours	<u>0.394</u>	<u>0.342</u>	0.294	0.261	0.246

Table 3. LPIPS ↓: evaluation of LLFF - Fern. Comparison of the LPIPS scores of the test set on 2, 3, 5, 6, and 12 training views.

Abs relative difference depth ↓

# Views	2	3	5	6	12
NeRF	0.241	0.114	0.063	<u>0.034</u>	<u>0.031</u>
Ours	<u>0.064</u>	<u>0.049</u>	<u>0.042</u>	0.040	0.039

Table 4. Abs relative difference depth ↓: evaluation of LLFF - Fern. Comparison of the absolute relative depth difference of the test set on 2, 3, 5, 6, and 12 training views.

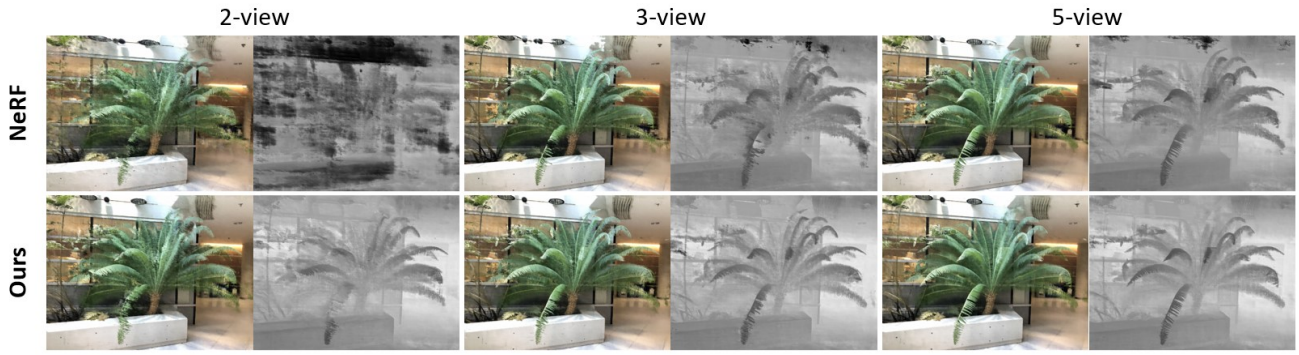

Figure 1. Comparison of the Results of the LLFF - Fern scene between NeRF and our approach. Shown are the RGB and depth map outputs for 2, 3, and 5 input views.

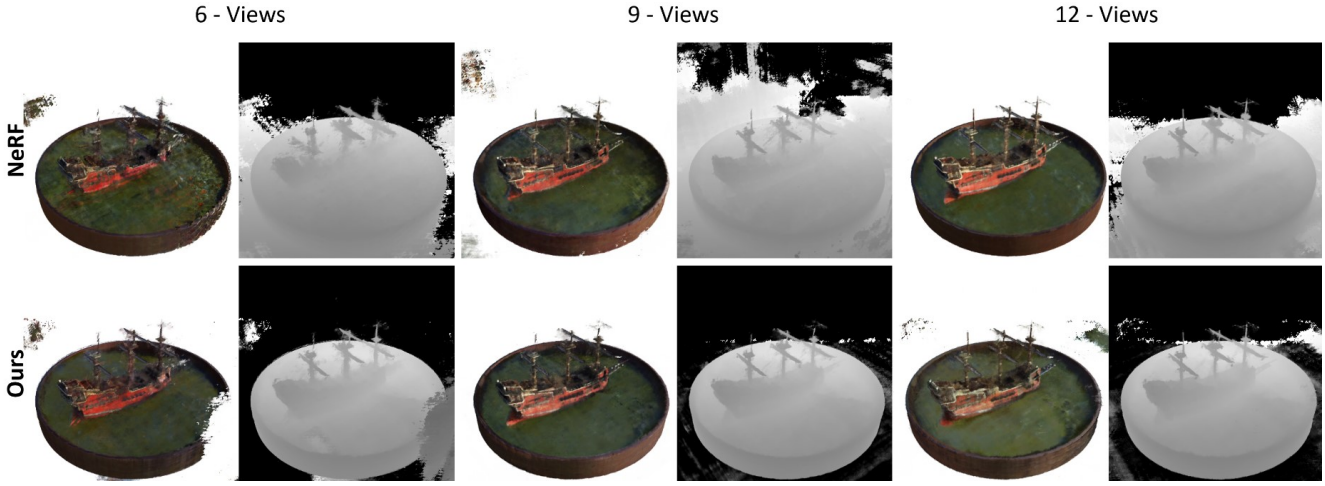

Figure 2. Comparison of the Results of the Blender - Ship scene between NeRF and our approach. The RGB and depth map outputs are shown for 6, 9, and 12 input views.

MSE depth ↓

# Views	2	3	5	6	12
NeRF	0.0495	0.0100	0.0036	0.0014	0.0010
Ours	0.0040	0.0026	0.0017	0.0012	0.0011

Table 5. MSE depth ↓: evaluation of LLFF - Fern. Comparison of the mean squared error of the test set on 2, 3, 5, 6, and 12 training views.

4.2. Blender - Ship

We also looked at the 360° scene "ship" and the results can be seen in Figure 2. Here we used more input views since the scene is more complex and 4 or fewer views did not produce a usable output. We make depth supervision only on the object and not the transparent background. Table 4.2 and 4.2 show the results of the RGB evaluation and Table 4.2 the results for the depth evaluation.

5. Discussion

Sparse input views. From the evaluation, we have seen that depth supervision can drastically increase scene completeness. As seen in Figure 1 the output of NeRF with only two views is lacking compared to our output. The main cause for this is floating artifacts in the scene that come from inconsistent depth maps. Inconsistencies are still visible with 3 and 5 input views, see the depth map of NeRF in Figure 1, where there are black patches on the top of the depth map. Indicating that an object would be very close to the camera but in reality, is not. This is one if not the greatest strength we observed when using depth supervision, representing better geometry in the scene that directly leads to fewer artifacts in the RGB output. Similar observations can be made on the Ship dataset where the background contains artifacts.

However, it does not take many input views for NeRF to get a similar or better color output than our method.

	PSNR ↑						SSIM ↑					
# Views	5	6	9	12	15	20	5	6	9	12	15	20
NeRF	16.65	17.65	13.72	22.98	24.60	25.52	0.677	0.684	0.684	0.785	0.807	0.813
Ours	19.87	16.49	21.15	23.05	24.86	25.54	0.724	0.709	0.758	0.782	0.801	0.809

Table 6. PSNR and SSIM evaluation of Blender - Ship. Comparison of the PSNR and SSIM of the test set on 5, 6, 9, 12, 15, and 20 training views.

LPIPS ↓						
# Views	5	6	9	12	15	20
NeRF	0.281	0.272	0.290	0.207	0.188	0.187
Ours	0.258	0.276	0.225	0.211	0.212	0.198

Table 7. LPIPS evaluation of Blender - Ship. Comparison of the LPIPS score of the test set on 5, 6, 9, 12, 15, and 20 training views.

Geometry. The geometry in the scene seems to greatly benefit from the depth supervision. With only two views our method is able to create a good depth map. NeRF needs 5 input views to create a compatible result, see Table 4.1 and Table 4.1. It is even more noticeable in the Ship scene where our method produces good geometry in the scene from the beginning on. Also here we observe that this holds for sparse input views and NeRF is already a strong method for predicting a good geometry with more views.

5.1. Limitations

A significant limitation of the current approach is the utilization of synthetic depth maps for training. When considering real-world scenes where depth maps are obtained through sensor measurements, it is necessary to account for the presence of noise in the data. As such, the simple depth loss function employed in this approach is not optimal, as it fails to incorporate any form of noise modeling. To address this limitation, one potential solution is to utilize the KL-Divergence loss function, as introduced in the DS-NeRF method, which has the capability to consider uncertainties in the data. Our assessment of related work is somewhat deficient, as we did not conduct a sufficient number of scene evaluations for the DS-NeRF method.

One of the fundamental challenges associated with utilizing sparse input views for image representation is the difficulty in capturing the entirety of a scene, leading to an incomplete representation of details. When maintaining a close resemblance to the real object, this issue can only be partially addressed. An obvious solution is to increase the number of input views.

However, when utilizing data that is not carefully selected, the training views may not provide comprehensive coverage of the entire scene, resulting in inferior representations compared to using fewer views. This phenomenon was observed in our experimentation with the Ship scene, where utilizing 9 input views with NeRF did not yield superior results compared to utilizing 6 or 5 views, as demonstrated in Table 4.2.

6. Conclusion

In this work, we have demonstrated that utilizing depth supervision with NeRF can lead to improved results when utilizing sparse input views. Notably, this approach resulted in a significant reduction in floating artifacts caused by incorrect modeling of geometry. Through our experimentation on the Fern scene, we have established that utilizing only 2 views is sufficient to achieve a realistic geometric representation. This effect also carried over to the RGB outputs, leading to improved detail representation.

As future research in the field of RGB-D-based NeRF approaches, it would be interesting to explore the potential of combining ideas from related work. Specifically, pre-training a coarse network using synthetically generated images of a constructed mesh of the scene in conjunction with geometry regularization, as seen in the RegNerf method, followed by fine-tuning using depth supervision could be a promising avenue for investigation.

In conclusion, depth-supervised NeRF is an effective approach for training on sparse input views and can lead to improved results compared to traditional NeRF, particularly in terms of geometry. As the number of training views increases, the performance of NeRF can keep up with our method.

7. Acknowledgement

We thank ETH Computer Vision and Learning Group for providing the materials and hardware for this project. Special thanks go out to the supervisor of this work, Dr. Sergey Prokudin for proposing this interesting topic and for the kind guidance throughout the project.

	abs relative difference depth ↓						MSE depth ↓					
# Views	5	6	9	12	15	20	5	6	9	12	15	20
NeRF	15.00	11.43	25.01	6.51	5.80	4.36	0.353	0.291	0.592	0.221	0.242	0.175
Ours	5.91	10.65	4.98	4.18	4.56	3.73	0.152	0.280	0.175	0.101	0.090	0.077

Table 8. Absolute relative difference depth and MSE depth evaluation of Blender - Ship. On the test set on 5, 6, 9, 12, 15, and 20 training views.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[2] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022.

[3] Arnab Dey, Yassine Ahmine, and Andrew I Comport. Mip-nerf rgb-d: Depth assisted fast neural radiance fields. *arXiv preprint arXiv:2205.09351*, 2022.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[6] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[7] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In *Computer Graphics Forum*, volume 40, pages 45–59. Wiley Online Library, 2021.

[8] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

[9] Cheng Sun, Min Sun, and Hwann-Tzong 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*, pages 5459–5469, 2022.

[10] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[11] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[12] Yu-Jie Yuan, Yu-Kun Lai, Yi-Hua Huang, Leif Kobbelt, and Lin Gao. Neural radiance fields from sparse rgb-d images for high-quality view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

8. Appendix

For a better comparison between the evaluation in NeRF and our approach see Figure 3 and Figure 4.

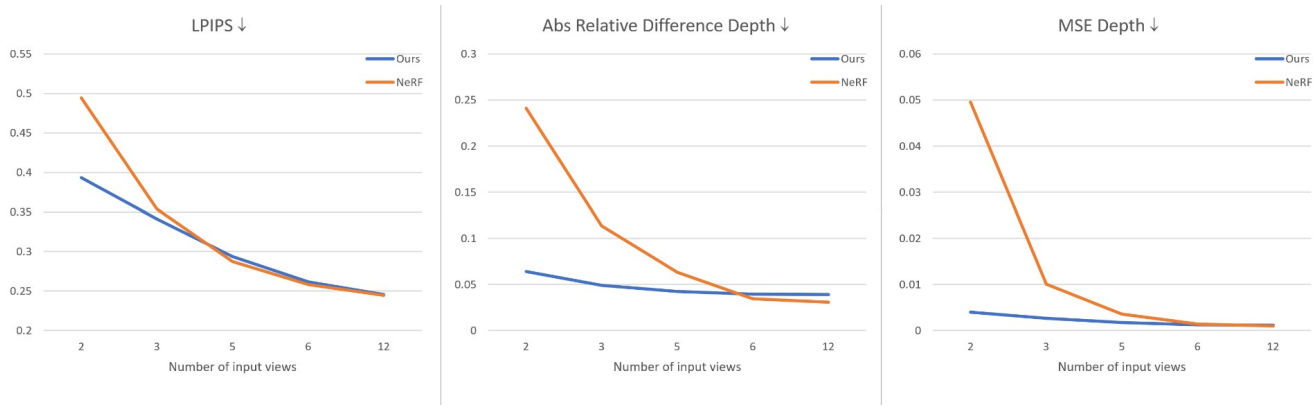

Figure 3. Scene Fern: Visual representation of the metrics shown in Table 4.1, 4.1, and 4.1

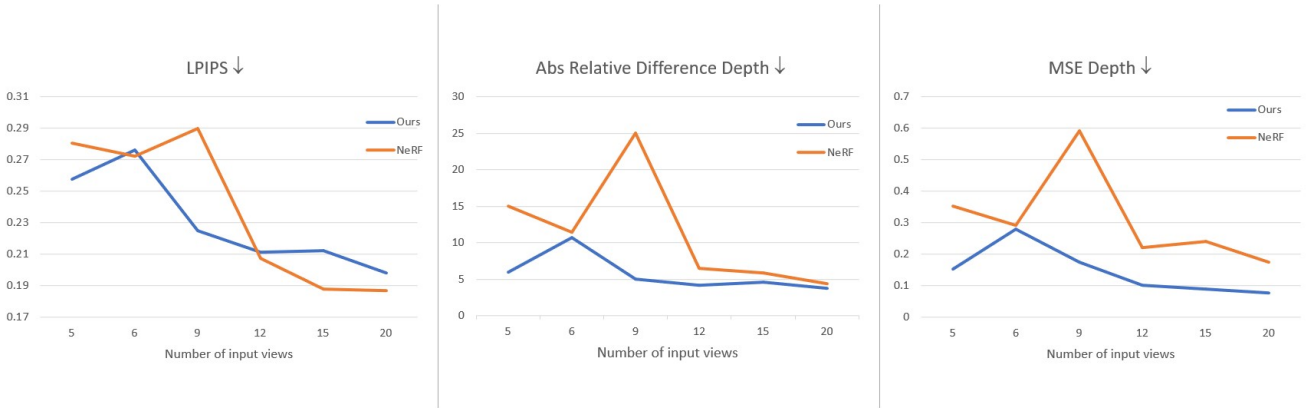

Figure 4. Scene Ship: Visual representation of the metrics shown in Table 4.2, and 4.2