

Speeding up pixelNeRF by Upsampling Low-Resolution Images

Felix Förster^{1,2}

Lars Schneider^{1,2}

Tobias Neumeier^{1,2}

Adrian Struzek^{1,2}

Abstract

In recent years, Neural Radiance Fields (NeRF) has advanced the field of novel view synthesis by enabling high-quality 3D scene reconstruction from sparse input images. However, the computational cost of traditional NeRF models remains a limitation which inhibits its usage in fields with high performance or real-time requirements, such as augmented and virtual reality.

PixelNeRF addresses some of these challenges by learning a scene prior, eliminating the need for per-scene optimization. We speed up pixelNeRF by training on full images instead of randomly sampled rays. To achieve this, we first render at a lower resolution, significantly reducing computational requirements. Afterward, a convolutional upsampler recovers high-resolution details, reducing inference times while maintaining as many details as possible.

We evaluate our approach on the SRN-Cars and CO3D-Apples datasets. Our results demonstrate substantial improvements in inference speed without losing output quality.

1. Introduction

Neural Radiance Fields (NeRF) have revolutionized novel view synthesis by using neural networks to reconstruct 3D scenes from a sparse set of images. However, per-ray volumetric sampling in traditional NeRF methods leads to slow inference speed, making real-time applications challenging. With pixelNeRF [16], Yu et al. introduced a scene prior to reduce the number of required views and compute time. The prior is learned from images in a fully convolutional manner without explicit 3D supervision, and the view synthesis is performed in a feed-forward manner from a sparse set of views. Despite these advances, inference time remains a bottleneck, particularly for high-resolution rendering.

In this work, we propose a novel approach to accelerate pixelNeRF inference while maintaining rendering quality.

Instead of processing randomly sampled rays, we train the model using full but downsampled images, reducing computational load during inference. To preserve output quality, we use a neural upsampling method based on a convolutional architecture proposed in [9]. During test time, our method renders low-resolution images and scales to the original input resolution using this learned super-resolution network. Additionally, we test the impact of using a VGG loss, aiming for further image quality improvements.

We evaluate our approach on the SRN-Cars [2] and CO3D-Apple datasets [8], with rendering quality and inference speed as key measures. CO3D-Apple examples can be seen in 1. Our method demonstrates significant inference acceleration while maintaining reasonable quality, showing the significance of our approach for real-time view synthesis applications as required in augmented and virtual reality applications.



Figure 1. Example data of CO3D-Apple from different scenes.

2. Related Work

Novel view synthesis: In recent years, NeRF has established its significance for novel view synthesis, motivating many research projects that aim to improve the problem of slow inference speeds. PixelNeRF [16] introduces a generalizeable NeRF framework that conditions on images using a convolutional neural network. Architectures proposed in MVNeRF [3], NPMVS [1] and IBNet [13] leverage multi-view stereo cues and image-based rendering to improve generalization and efficiency.

Methods like Instant-NGP [7] augment its network with multi-resolution hash encodings to improve rendering speed. Furthermore, FastNeRF [5] achieves significant

¹Authors are affiliated with the School of Computation, Information and Technology, Technical University of Munich.

Contact: {f.foerster, lars.schneider, tobias.neumeier, adrian.struzek}@tum.de

²These authors contributed equally.

gains in speed by employing factorized voxel grids. In contrast, our work builds on pixelNeRF but introduces a new approach: we render at a lower resolution, which allows processing the full image instead of batches of rays. Afterward, a super-resolution component scales the image to the original resolution, significantly reducing inference time while preserving quality.

Super-resolution: Super-resolution techniques aim to reconstruct high-resolution images from low-resolution inputs. Since traditional methods like bicubic interpolation struggle with loss of detail, deep learning-based super-resolution models were developed. [4] first demonstrated the usage of CNNs for this task, which was further improved by more advanced architectures such as EDSR [6] and SRGAN [14], employing deeper networks and generative adversarial training to increase quality.

In the context of NeRF, super-resolution has been explored by NeRF-SR [12] which applies a separate neural network to upsample rendered images, reducing the need for high-resolution sampling. In contrast to our work, NeRF-SR relies on super-sampling, shooting multiple rays at each image pixel. Furthermore, we use a perceptual loss (VGG [11]), aiming for even better quality improvements by sacrificing training speed.

3. Method

Our goal is to improve the inference time of pixelNeRF. To achieve this, we work with the pixelNeRF framework (3.1) on lower, downsampled resolutions (3.2) and then upsample the results (3.3) with a CNN. Due to compute limitations, the pixelNeRF network and the upsampler were **not** trained end-to-end.

3.1. pixelNeRF

PixelNeRF is a learning framework for novel view synthesis from a sparse set of views and was introduced by Yu et al. [16]. It predicts a neural scene representation from few or even a single input image. This is achieved by conditioning the network on the input images in a fully convolutional manner, unlike classical NeRF approaches. By training the convolutional network on multiple scenes, it can learn a scene prior, which is essential for achieving good quality results from sparse input.

Figure 2 shows the pixelNeRF architecture for the single input image case. First, a CNN encoder computes a pixel-aligned feature grid W for the input image. Then, for each pixel in the output image, a ray is cast through the pixel into the volume of the neural scene representation and sampled at a certain number of points. The sample points are projected into the input image space, and a feature vector is extracted for the respective sample point by interpolating between the pixel-wise features.

The sample point and viewing direction are then passed into

the NeRF network, and the feature vector is fed into each layer as a residual. With this, the network predicts the color and density values for the current sample point. The output RGB and density values are finally integrated into the final pixel color.

In the multi-view case, the approach is similar. Each input image is encoded into a latent representation separately, and the query points are projected into each view frame to retrieve a feature vector for each view. The coordinates and features for each view are processed separately and only aggregated in the NeRF network by average pooling to obtain the output density and color from the final layers.

3.2. Downsample

To achieve faster inference times compared to pixelNeRF, we render lower-resolution images. Since considerably fewer rays are needed on lower resolutions, this results in a substantial speedup. For our approach, we downsample the images to 1/16th of their input size, which results in 32x32 images instead of the original 128x128. The output image is also rendered at the same resolution. To accommodate the lower resolution, we need to retrain the network. During training, the original implementation sampled only 128 rays per view. When using all image pixels during training, pixelNeRF can be trained end-to-end with the upsampler. This approach can lead to better results.

3.3. Upsample

The goal of the upsampler is to increase the resolution for the output of pixelNeRF, which has a lower spatial resolution than the original. The upsampler needs to be fast to decrease inference time while using the combination of pixelNeRF and the upsampler. We used the architecture of [10], which only uses convolutional layers in combination with a sub-pixel convolutional layer at the end to increase the spatial dimensions of the input image. The model used in [10] has 37k parameters. We also used an adapted model using more convolutional layers, to make the upsampler more powerful while keeping the inference times low; this bigger model has 200k parameters. The upsampler is trained on the flicker30 [15] dataset, using Mean-Squared-Error (MSE) on images in the YCbCr domain as proposed in [10]. The luminance channel of YCbCr has the most impact on the human eye in terms of structure. At the same time, using the chromatic channels can avoid color shifts in the results. A perceptual loss further improves the results. We use VGG-16 [11], in combination with the Mean-Squared-Error (MSE) loss. The outputs of the model have to be rescaled to a spatial size of [244, 244]. In addition, we applied normalization using a mean of [0.485, 0.456, 0.406] and std of [0.229, 0.224, 0.225] provided by [11]. The downside of using a perceptual loss like VGG is a significantly increased training time. This problem is caused by the size of the

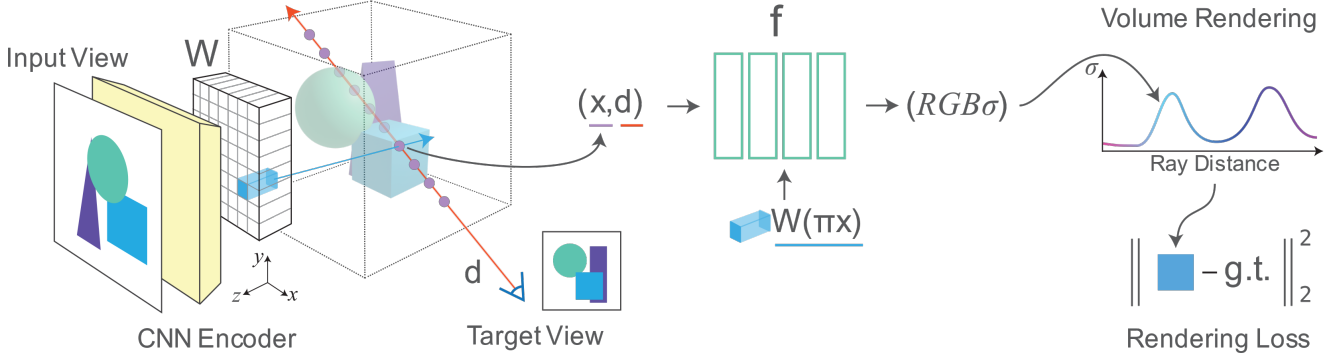


Figure 2. Architecture of pixelNeRF for single image input.

VGG model, which has about 136M parameters, being significantly larger than the model of the upsampler.

4. Results

4.1. Upsampler Results

Version	PSNR	SSIM	Parameter Count
large-mse	26.01	0.793	200992
large-vgg	28.41	0.823	200992
small-mse	26.42	0.786	37200
small-vgg	26.71	0.735	37200
bicubic	26.85	0.781	-

Table 1. Evaluation results of the upsampler on Set5 dataset for a rescaling ratio of 4.



Figure 3. Image shows a comparison between the upsampler results with MSE and VGG. The model input is was downsampled by a factor of 4.

Evaluation Setup: To evaluate the upsampler performance, we used 1. peak signal-to-noise ratio (PSNR) and 2. structural similarity index measure (SSIM) on images from

the Set5 dataset. Both metrics are calculated on the generated image from the upsampler in combination with the ground truth label.

Result: Table 1 shows different variations of the trained upsampler evaluated on the Set5 dataset, which is commonly used for testing super-resolution models. VGG loss improves the PSNR and SSIM of the models compared to the respective version only trained with MSE. While the increased parameter count did not improve model performance significantly in combination with MSE, the bigger model performs better when using VGG loss. The evaluation shows that our trained upsampler can outperform bicubic interpolation results with fast inference times. Similar results can be seen when using a rescaling factor of two.

4.2. PixelNeRF + Upsampling Result

Evaluation Setup: For the evaluation of our results, we use the same metrics as for the upsampler. In addition, we analyzed the inference time. In the following, we show qualitative and quantitative results on the SRN-Cars and CO3D-Apple datasets. We compare three versions: the original pixelNeRF architecture, the pixelNeRF architecture with downsampled images, and the combination of downsampled images and upsampler. Note that we also state the original pixelNeRF checkpoint performance for the SRN-Dataset. We calculate the results of the downsampled pixelNeRF by upscaling the result with nearest neighbor interpolation and comparing it with the



Figure 4. Qualitative results on the SRN-Cars dataset. The respective images were cropped for more space efficiency.

original-sized image.

Qualitative Results: Figure 4 shows different example results of the different approaches on the SRN-Cars dataset. We can see nearly identical results from the original checkpoint and our reproduced one. While looking closer at the downsampled result, it is clear that there is a massive loss of detail. This is due to the low resolution of the images. This is also the reason why the result of the upsampled version is very blurry. Another factor for this lack of detail is the simple architecture of the upsampler. A similar blurry behavior is also observable when trained and applied to the CO3D-Apple dataset.

Version	PSNR	SSIM	Inference Time
original	21.95	0.8849	2.44s
reproduced	20.21	0.8474	2.38s
downsampled	13.24	0.4773	0.09s
down-/upsampled	19.21	0.7619	0.13s

Table 2. Evaluation results on the SRN-Cars dataset. The results of the downsampled pixelNeRF are compared to the full-resolution images by nearest neighbor interpolation. The original data point refers to the results of the checkpoint from [16].

Quantative Results: Table 2 shows the PSNR, SSIM, and inference time of each version. There are three key takeaways.

1. First of all, we were able to reproduce the checkpoint of the original pixelNeRF paper. The small remaining gap could be explained by shorter training, variance, or pretraining.
2. Furthermore, we can observe a noticeable improvement with the upsampler. But there is still a gap between the upsampled and original performance. This is most likely due to the simple upsampler and extreme downsampling.
3. The Inference time of the original pixelNeRF architecture was cut down by roughly 19 times. This is a signif-

icant improvement and opens new areas, like real-time applications.

Version	PSNR	SSIM	Inference Time
default	20.58	0.8988	2.66s
downsampled	13.53	0.4218	0.10s
down-/upsampled	18.93	0.7372	0.12s

Table 3. Evaluation results on the CO3D-Apple dataset. The results of the downsampled pixelNeRF are compared to the full-resolution images by nearest neighbor interpolation.

Table 3 displays the respective results for the evaluation of the CO3D-Apple dataset. We are able to achieve similar results as before. Note that worse results are expected on other CO3D datasets, which are potentially smaller or contain more complex objects.

5. Future Work

There are many possible improvements to the presented approach. In our opinion, there are two promising modifications:

1. To improve the resulting quality while maintaining a high inference speed, a rescaling factor of two could be a good trade-off between better quality while maintaining high speed.
2. Replacing the upsampler with a more capable generative network could increase the details and quality of the upsampled results.

6. Conclusion

We present a method to speed up pixelNeRF and related methods. We achieve real-time inference speed with the same output resolution by running pixelNeRF on lower resolutions and using a fully convolutional upsampler. Although considerably faster than pixelNeRF, our work suffers from loss of quality.

References

- [1] Jiakai Cao, Zhenlong Yuan, Tianlu Mao, Zhaoqi Wang, and Zhaoxin Li. Nerf-based polarimetric multi-view stereo. *Pattern Recognition*, 158:111036, 2025. 1
- [2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1
- [3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo, 2021. 1
- [4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks, 2015. 2
- [5] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps, 2021. 1
- [6] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution, 2017. 2
- [7] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1
- [8] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 1
- [9] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016. 1
- [10] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016. 2
- [11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2
- [12] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. Nerf-sr: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, page 6445–6454. ACM, 2022. 2
- [13] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering, 2021. 1
- [14] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks, 2018. 2
- [15] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 2
- [16] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *CoRR*, abs/2012.02190, 2020. 1, 2, 4