Lidar and Additional Constraints for NeRF on Outdoor Scenes

Kiray, Mert
mert.kiray@tum.de

Tunali, Yigit Aras
yigitaras.tunali@tum.de

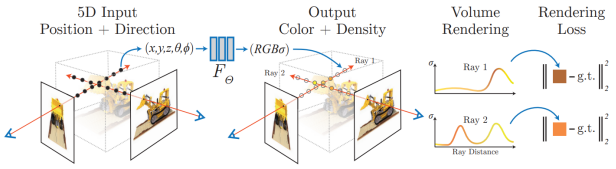


Figure 1: Complete pipeline of the NeRF [8]

Abstract

Neural Radiance Fields [8] are used to achieve state-of-the-art novel view synthesis of complex scenes with a fully connected non-convolutional deep network. One of the main disadvantages of NeRFs is that they require many images of the scene to be synthesised, which are preferably egocentric. This is usually not the case for many outdoor scenarios, especially for autonomous driving related datasets. In this work, we address the lack of imagery by introducing additional constraints on the optimisation of the Neural Radiance Field by using depth information provided by a LIDAR sensor.From experiments and various ablation studies we have conducted, we have concluded that using the constraints of a LIDAR sensor alone is not sufficient to achieve the desired results. In addition to depth constraints, we extend and experiment with the notions of extrapolated Lidar depth, feature loss, semantic loss, GAN-based loss and nverse-depth smoothing loss. We compare the results, improvements and shortcomings of all the methods applied above and present the best combination that our research has produced.

1. Introduction

The goal of Neural Radiance Field is to create a novel synthesis of a static scene from a given set of egocentric images. For this purpose, a Neural Radiance Field, $\mathbf{F_\Theta}$, is optimised. $\mathbf{F_\Theta}$ can be thought of as a 5D function that takes as input the 3D position (x, y, z) and the viewing direction (θ, ϕ) and gives as output the RGB colour and density (R, G, B, σ) at that point. For training, a random batch of rays are shot from pixels of input images, using the regular volume rendering method discussed in [8], the colour at that pixel is calculated and compared to the ground truth. This process is trivially differentiable and can ultimately be reduced to traditional alpha compositing.

To achieve the state of the art, the NeRFs take two other important steps. The first is position coding of the (x, y, z) coordinates to prevent bias towards learning lower frequency features. The second is hierarchical volume sampling, which trains a coarse and a fine mesh, achieving a similar goal to importance sampling. Although NeRFs produce impressive results, they struggle with some issues that arise when applied on outdoor scenarios, such as the lack of required image volume and apperances of dynamic objects. Combined with the lack of egocentric data in our autonomous driving scenario, the quality of the resulting synthesis drops immensely.

We modify the structure shown in Figure 1 so that the LIDAR depth information is also used as input and influences the optimisation procedure. We took the approach from DSNeRF [4] as a basis. In addition to using the depth information, we extend the model with a feature loss and additional depth operations to further improve the performance and complete the missing information from the LIDAR scans. In addition to the methods mentioned above, we have also experimented with and implemented some methods from previous works to further improve the performance of our approach. Thus, both Semantic Loss [12] and a GAN-based loss were implemented. Apart from the GAN-based loss, which was abandoned due to time constraints and complexity, all additional constraints either reduce the time required for training or directly improve the quality of the synthesised images.

2. Related Work

2.1. Depth-supervised NeRF: Fewer Views and Faster Training for Free

One of the researches that comes closest to what we want to achieve is depth-supervised NeRF [4]. The basic idea is to augment regular NeRFs with depth monitoring. Using the additional depth signals, the authors were able to reduce the number of images required while optimizing the Neural Radiance Field to be suitable for indoor static scenes. The main difference in our work is the use of outdoor scenes

and mainly non-egocentric data. As a starting point, we will adapt the depth monitoring part of this work to synthesize plausible novel views with fewer images.

2.2. pixelNeRF: Neural Radiance Fields from One or Few Images

A final work we examined that addresses the lack of enough images is the pixelNerf approach [11]. The authors combine the regular NeRF approach and additionally train a CNN to learn certain scene priors to reduce the number of images needed for synthesis. Since our data will mainly be outdoor scenes in an autonomous driving scenario, learning scene priors could improve our performance and quality immensely.

2.3. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis

In the DIET-NeRF [5] approach, the authors try to improve the few-shot capabilities of NeRFs by introducing additional constraints to the optimisation process. They extract high-level semantic attributes from arbitrary poses to use in a loss term for monitoring the optimisation from different poses. The semantic attributes are extracted using a pre-trained CLIP[10] model, and the semantic loss significantly increases the performance of the model even with only a few input images.

3. Datasets

For this project, we are mainly using KITTI-360. Since our goal is to improve the view synthesis of outdoor scenes and also use LIDAR data, KITTI was a natural choice. As it contains many sensors and labelled data, we hoped to use these additional inputs to further improve our synthesis.

4. Methods

4.1. LIDAR Depth Loss

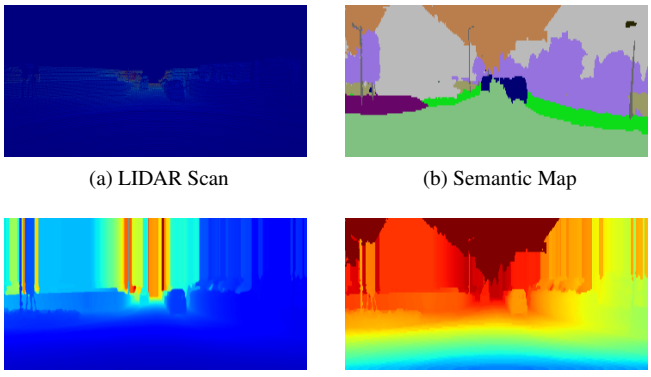
We took DSNeRF as a basis and modified it to work with the pose and depth data from the KITTI-360 dataset. Since DSNeRF was not designed to handle outdoor scenes, its performance drops significantly when trained on unconstrained scenes. We solved this problem by projecting our scene and LIDAR depths into an NDC space. We convert our LIDAR depths to the range $[0, 1]$, where 0 is the near plane and 1 is the far plane. To convert our depth map from Euclidean space to NDC space, we use the formula $ndc\_depth = 1 - (1/real\_depth)$.

4.2. Feature Loss

We follow [5] to force our network to have a better perceptual similarity with the GT images. We use a VGG19 network pretrained on Imagenet to extract features from the rendered and GT images. Then we use a L1 loss between these two feature vectors to guide our optimization process. We experimented with different layers of the VGG19 network for feature extraction, but ultimately decided on using the CONV1-1, CONV2-1, CONV3-4, CONV4-4 and CONV5-4 layers with weights $[0.1, 0.1, 1, 1, 1]$. Although we were inspired by the DIET-NeRF approach, we struggled with memory issues that led us to change their method. We selected a random section of the rendered image and shot rays from those pixels. We modified the training to randomly sample some rays to have gradient flow so that the backpropagation operation would fit in the memory we were working with.

4.3. Semantic Loss

We follow the work of Semantic-NeRF [12] to create an additional fully connected head that predicts the semantic class of the queried pixels. This head is created before the viewing-direction information is fed into the network, since the semantic segmentation of the view does not depend on the direction of sight. To extract semantic groundtruth classes for training, we use DeepLab v3 [2] pretrained on Cityscapes dataset [3]. We use this additional segmentation information to improve the synthesis quality and create a semantic map for unseen camera poses.



(a) LIDAR Scan (b) Semantic Map

(c) Depths filled + filtered (d) Segmentation info leveraged

Figure 2: Depth Extrapolation

4.4. Depth Extrapolation

One of the biggest problems we faced with LIDAR data was that depth values were only recorded up to a certain height of the image. This resulted in us obtaining depth maps that were only half filled and mostly lacked information above a certain line, as seen in 2a. We used an approach similar to [6] to extrapolate the missing depth information, and also used a bilateral filter to eliminate generated outlier values as can be seen from 2c. We also include the segmentation map in 2b created for the segmentation loss to

fill pixels corresponding to a sky with nearly infinite depth values. With these steps, we are able to improve the contribution of LIDAR depth data to our network and create a smoother and more accurate depth map for novel views as can be seen in 2d.

4.5. Inverse Depth Smoothness Loss

Because of the way we handled the depth extrapolation step, the extrapolated depth values had to be adjusted. Simply filling in the missing values and using filters to limit the noise still resulted in unwanted artefacts. To smooth the depth values and regulate the structure of the predicted depths, we use the image-aware inverse depth smoothing loss as described in [1] with the following formula.

$$\text{Smoothness Loss} = |\partial_x d_{ij}|\, e^{-\|\partial_x I_{ij}\|} + |\partial_y d_{ij}|\, e^{-\|\partial_y I_{ij}\|}$$

4.6. Adversarial Loss

Following [9], we use an adversarial loss to guide the network to produce higher quality images. We adapted the discriminator from [7] to compare the image patches we created in section 4.2 with the patches we extracted from the GT images. We assume that our NeRF network is a generator and use the discriminator to calculate an adversarial loss between patches. However, we discarded the adversarial loss for now because the NeRF network was not powerful enough to fool the discriminator and we had problems with mode collapse during training.

5. Results

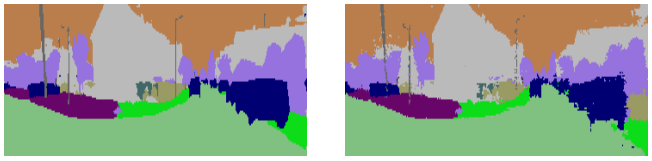
Method	PSNR↑	SSIM ↑	LPIPS ↓
NeRF	23.28	0.9437	0.2127
DSNeRF	22.62	0.9319	0.2595
Ours	**23.46**	**0.9443**	**0.2089**

Table 1: Quantitative results

For all results, we use every loss we have discussed so far, with the exception of Adversarial Loss, which we have abandoned. Figure 2 shows the simple LIDAR scan input and the steps of our depth completion approach. The resulting output Figure 2d is significantly better than using half-filled depth information from the LIDAR input data. This additional informative depth map helps us to further utilise the dense depth information from LIDAR. The differences between a normal NeRF, DS-NeRF and our approach can be seen in the Figure 4. One of the reasons why DSNeRF performs worse is that it does not work in NDC space and does not give correct results for an unconstrained scene. Our approach and regular NeRF use NDC space for the calculations and we also use the semantic segmentation map to assign almost infinite values to the sky. All these steps together improve our results both visually and quantitatively.

We also show that NeRF can be used for semantic segmentation of outdoor scenes. This is especially important for autonomous driving tasks. The semantic loss added to NeRF also helps to get better metrics and better novel views. An example of semantic segmentation of a new scene can be seen in Figure 3.

Finally, we show that the feature loss can also be used to improve synthesis quality. The performance of feature loss is highly dependent on the size of the patch to which feature extraction is applied. Also, we can only backpropagate through a limited number of pixels from these patches during training, again due to memory constraints. In our tests, we have seen that including feature loss improves some metrics, but ultimately slows down the training process. If the memory problem could be solved or a more efficient method to calculate and use this loss could be implemented, we believe that the contribution of this loss could be improved even further.



(a) Semantic GT (b) Semantic Prediction

Figure 3: Semantic Segmentation

Figure 4: Results

6. Ablation Study

We have conducted ablation studies using different combinations of our approaches to better analyse their performance quantitatively. As shown in Figure 2, extrapolating the LIDAR scan without applying a depth smoothing loss degrades the performance of the reconstructed scene, although it significantly improves the estimated depth of the scene. Adding a depth smoothing loss to our extrapolated LIDAR scan to correct for incorrectly filled depth values and fill in the holes in the estimated depth of the scene significantly improves all three metrics. Adding feature loss to our overall loss also appears to improve the LPIPS score as expected. While semantic loss improves the PSNR metric, it worsened the LPIPS score. Since our final method takes all losses into account, the feature loss and semantic loss seem to balance each other in terms of LPIPS and PSNR scores.

As explained in the previous sections, due to memory constraints, we had to use downsized images as input for most of our runs (usually by a factor of 4), which affects both the performance of the network and some of our losses. The patch-based approaches we had to implement for some of our losses also reduce their performance, as they do not work as well when they receive partial areas of the image as input. Figure 5 shows the error comparison with the LPIPS metric between NeRF, DSNerF and our approach. As can be seen, our method causes fewer errors overall, especially in areas where regular NeRF and DSNeRF have quite high errors.

Method	PSNR↑	SSIM ↑	LPIPS ↓
LIDAR	23.14	0.9446	0.2200
Extrapolated	23.11	0.9463	0.2217
Extrap.+Smooth.	23.44	**0.9465**	0.2089
Extrap.+Smooth.+ Feat.	23.40	0.9459	**0.2018**
Extrap.+Smooth.+Semant.	23.38	0.9435	0.2274
Extrap.+Smooth+Feat.+Semant.	**23.46**	0.9443	0.2089

Table 2: Ablation Results

7. Conclusion

In summary, our main contribution is the inclusion of LIDAR depth values in the NeRF optimisation procedure. We also address the problem that LIDAR data from autonomous driving scenarios are incomplete by using depth extrapolation and inverse depth smoothing losses. Using these two approaches and exploiting the results from semantic segmentation, we can further improve the quality of our depth map by dividing the scene into foreground and background/sky so that the depth values can be improved. Finally, we also included a feature loss term corresponding to the L1 distance between the feature vector extracted from the GT image and the synthesis image, further increasing the image quality of the output.

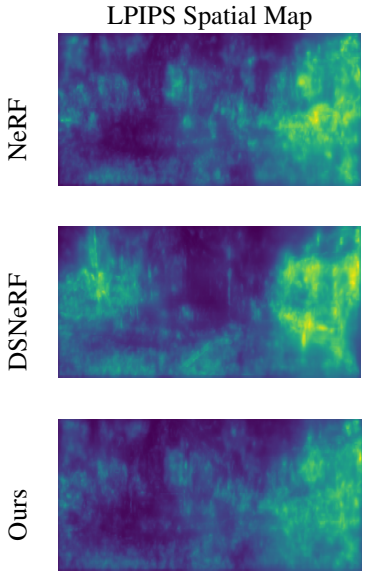


Figure 5: LPIPS Error Analysis, Lighter Green = Higher Error

There are still many improvements possible for the methods tried. One of them is to improve the segmentation quality. Since the segmentation results are used to change the depth values, incorrect or faulty predictions of classes can lead to unwanted artefacts. Another approach would be to improve the extrapolation performance by using different kernel schemes or even fully integrating it into the network and training it in an end-to-end fashion. There is also room for improvement for feature loss. Since we use patch-based feature extraction, different extraction methods could be used to work better in these patches due to memory constraints. Other networks such as a CLIP ViT could be used for feature extraction process as well.

References

[1] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019. 3

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[4] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free, 2021. 1

[5] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. *CoRR*, abs/2104.00677, 2021. 2

[6] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22. IEEE, 2018. 2

[7] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016. 3

[8] 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, 2020. 1

[9] Rohit Pandey, Sergio Orts-Escolano, Chloe LeGendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. Total relighting: Learning to relight portraits for background replacement. volume 40, August 2021. 3

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 2

[11] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021. 2

[12] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1, 2