3D Scene Reconstruction with Neural Radiance Fields (NeRF) Considering Dynamic Illumination Conditions

Olena Kolodiazhna[1], Volodymyr Savin[1], Mykhailo Uss[2] and Nataliia Kussul[1,3,4]
[1]*Institute of Physics and Technology, Igor Sikorsky Kyiv Polytechnic Institute, Peremohy Avenue 37, Kyiv, Ukraine*
[2]*Department of Information-Communication Technologies, National Aerospace University, Chkalova Str. 17, Kharkiv, Ukraine*
[3]*Department of Space Information Technologies and System, Space Research Institute National Academy of Science of Ukraine an State Space Agency of Ukraine, Glushkov Avenue 40, Kyiv, Ukraine*
[4]*Anhalt University of Applied Sciences, Bernburger Str. 57, Köthen, Germany*
kolodyazhna.lena@gmail.com, vladimyr.savin@gmail.com, mykhail.uss@gmail.com, nataliia.kussul@gmail.com

Keywords: Computer Vision, Neural Radiance Fields, Dynamic Illumination, View Synthesis, 3D Scene Reconstruction.

Abstract: This paper addresses the problem of novel view synthesis using Neural Radiance Fields (NeRF) for scenes with dynamic illumination. NeRF training utilizes photometric consistency loss that is pixel-wise consistency between a set of scene images and intensity values rendered by NeRF. For reflective surfaces, image intensity depends on viewing angle and this effect is taken into account by using ray direction as NeRF input. For scenes with dynamic illumination, image intensity depends not only on position and viewing direction but also on time. We show that this factor affects NeRF training with standard photometric loss function effectively decreasing quality of both image and depth rendering. To cope with this problem, we propose to add time as additional NeRF input. Experiments on ScanNet dataset demonstrate that NeRF with modified input outperforms original model version and renders more consistent 3D structures. Results of this study could be used to improve quality of training data augmentation for depth prediction models (e.g. depth-from-stereo models) for scenes with non-static illumination.

1 INTRODUCTION

3D scene reconstruction is a long-standing problem in computer vision consisting in understanding 3D structure of a scene given its 2D images. It is applied to diverse domains, including Augmented Reality (AR) and Virtual Reality (VR). For example, 3D scene reconstruction enables occlusions and collision processing between augmented content and the physical world for natural, seamless, and realistic interactions in AR. Multiple methods and tools, including Markov Random Fields [1], local stereo matching algorithms [2, 3] and deep neural networks [4], are applied to solve 3D scene reconstruction task. The complexity of this task is due to the need for simultaneous consistent reconstruction of global scene structures and their local details that require massive computations and a large amount of data. Availability of accurate and reliable data is crucial in deep neural networks training.

Manual data collection with further annotation and synthetic data generation are common approaches for training datasets acquisition. The process of manual collection is very time-consuming and expensive: one need to collect GT depth data, image data along with accurate camera poses. Consequently, this process also requires additional specific equipment, for example, depth cameras. Fully synthetic data cannot completely replace real data in neural networks training. In order to guarantee reasonable performance in real operation conditions, models obtained after fitting on such fully synthetic data should be finetuned on real data from the target domain [5].

One of the recent advances in view synthesis with reliable results is Neural Radiance Fields (NeRF) [6]. It is an MLP (multilayer perceptron) network that can generate novel views of the scene given a limited amount of pictures of the scene with corresponding camera poses during the training process. NeRF optimizes underlying continuous volumetric function using a sparse set of input views [6]. This method may be used for new data generation and augmenting training datasets for depth prediction neural networks. In, for example, [7] authors uses synthetic im-

ages generated by NeRF to solve localization tasks. They demonstrate that additional synthetic data improves the accuracy of regression of the camera's position. One of the advantages of this MLP network is that it can synthesize not only RGB images but also depth maps that are important for depth estimation networks. And as mentioned before, 3D scene reconstruction should be close to the original environmental structures, so it requires reliable depth maps.

During training NeRF optimizes photometric loss function that is equal to pixel-wise difference between intensities of original and generated images. However, this loss function has its limitations in dynamic scenes and scenes with illumination changes: such data violate the brightness consistency assumption important for such photometric loss functions due to its dependence on image intensity. Discussed loss function is also a common loss function that is minimized in depth estimation networks that are trained in unsupervised mode (e.g. in [8, 9]). Authors of [10] analyze the problem of using photometric consistency loss functions for datasets with bad or dynamic illumination. They demonstrate that standard photometric loss function fails for such data.

The main contribution of this work is using a time variable as a sequential image index added to the NeRF model input parameters for static scenes but with dynamic illumination. This modification considers some illumination changes across all datasets and allows their compensation. To evaluate its influence on scene reconstruction, the depth generation quality of NeRF was measured by calculating relative depth errors on two ScanNet data [11] scenes with some changes of illumination and with the presence of reflections.

The rest of this paper is organized as follows. Section 2 reviews related works in generating image and depth data and scene reconstruction with illumination changes. Section 3 discusses the problem of NeRF training for scenes with dynamic illumination and proposes modifications of NeRF architecture and loss function. Section 4 discusses experiments that demonstrate the effect of the proposed modifications. Finally, Section 5 summarizes main article's contributions and future research directions.

2 RELATED WORK

Having a certain number of images of the scene captured from different positions, NeRF reconstructs its entire 3D structure and enables the synthesis of novel views. This method is not the first to address this problem. Other variants of generating novel views include, for example, 3D grid-based optimization of the representation of the scene [12] or methods optimized by neural networks that map XYZ coordinates into sign distance functions (SDF) [13, 14]. However, due to their computational complexity, these methods require high-quality ground truth 3D data and have low-scaled capability focused on generating high-resolution images. On the contrary, NeRF optimizes the representation of the scene in the form of a continuous differentiable function that allows training model in an end-to-end manner.

However, the original NeRF model [6] also has drawbacks due to, for example, time consuming and requirement of accurate camera poses. Authors of BARF model [15] mitigate exact camera poses requirement by adding them to the optimization process. Both NeRF and BARF models have common limitation of supporting only a single scene. They both fail to generalize other environments and allow image synthesis only for the specific scene used in the training process. There are some NeRF models such as [16, 17] that overcome the mentioned problem by using MVS-based (multi-view stereo) approaches.

In general, models that are based on NeRF assume that scenes are static, but it is not always true. Neural Scene Flow Fields (NSFF) [18] is one of the NeRF representatives that enables the optimization for dynamic scenes. The authors of this model modified the basic representation of the scene as NeRF, considering the dynamic conditions of the environment. As a result, obtained new representation of the scene, NSFF, simulates the dynamic scene as a time variable continuous function of the environment representation, geometry, and movement of the 3D scene. Such an approach enables the interpolation of changes both in space and time. In contrast to this work, in this paper we consider time dependence NeRF for static scenes with dynamic illumination.

3 METHODOLOGY

3.1 NeRF Training Loss

NeRF has recently gained success in generating novel views for complex scenes [6]. It represents a scene using the fully-connected deep network. The input to the model is a 5D vector-valued function which arguments are spatial location (x, y, z) and viewing direction (θ, ϕ). The output is a volume density σ and RGB color $\vec{c}$. The model can be written as [6]:

$$F_w : (\vec{x}, \vec{d}) \rightarrow (\vec{c}, \sigma) \quad (1)$$

To train NeRF, we need a dataset with RGB images of the scene, camera poses, and camera intrinsic parameters. The training process involves rendering corresponding views of the scene and minimization of the photometric loss between observed and synthesized images.

The rendering process is illustrated in Figure 1. First, we march camera rays through each pixel of the image and sample some points. Then, these points are fed to the MLP network that predicts color and density for each of them. At the last stage, classical volumetric rendering [19] is used to aggregate all colors and densities for each sampled point and get the final result for the pixel.

During training, NeRF minimizes photometric loss function. Generally, it uses two networks: coarse and fine [6]. But for simplicity in our work, we only consider the first coarse subnetwork. Given M images $(I_1,...,I_M)$, the goal of NeRF training is optimize the following synthesis-based objective:

$$L_F = \sum_{i=1}^{M} \sum_{u} ||\hat{I}_i(u;w) - I_i(u)||_2^2, \tag{2}$$

where w is the network parameters that also depend on the view directions, u denotes pixels coordinates, $\hat{I}(u;w)$ is the synthesized RGB value at pixel u.

3.2 Photometric Loss Limitations

One of the characteristics of real data is dynamic illumination. Usually, existing datasets have static scenes without changes of lighting. Illumination changes may be caused by some reasons e.g., from external sources such as the sun or from car lights. Also, these changes may be due to camera exposures. NeRF models take into account positions and input ray directions that can compensate effects caused by reflective surfaces but they do not consider time variable. Time dependence is important for scenes with dynamic illumination. Consequently, standard NeRF models may fail in rendering a correct color for images from such datasets. Also, as it shown in [10], depth estimation networks that use photometric-consistency loss functions fail in recovering 3D scene structure for datasets with bad or dynamic illumination. Given this, NeRF-based models may have the same problems as they also use photometric loss function.

3.3 Model Modifications

3.3.1 Depth Loss

The original NeRF model and all its modifications predict color along with the density σ that can be interpreted as an opacity of the objects. Using the obtained density, we can calculate distances to objects. During NeRF training, the predicted depth maps are not further optimized. This leads to the incorrectly predicted distances to the objects and, thus, low quality of the 3D scene reconstruction. There are some datasets that have (incomplete) depth maps that can be used to improve NeRF quality. To do this, we propose to add an additional loss function, which is defined as the MSE error between the GT (ground truth) values of the distance maps and the predicted ones:

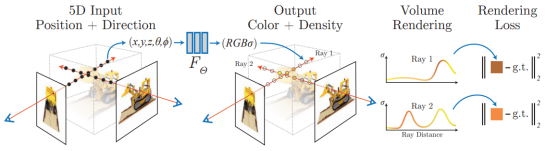

Figure 1: NeRF rendering process [6].

$$L_D = ||D - \hat{D}||_2^2, \tag{3}$$

where D - GT depth maps, $\hat{D}$ - predicted depth maps. The overall loss function takes the following form:

$$L = \sum_{i=1}^{M} \sum_{u} ||\hat{I}_i(u;w) - I_i(u)||_2^2 + ||D - \hat{D}||_2^2, \tag{4}$$

where M denotes number of images, u is pixel coordinates, $\hat{I}_i(u;w)$ is synthesized RGB color in pixel u for image i, $I_i(u)$ is GT RGB color in pixel u for image i, D is GT depth maps, $\hat{D}$ is predicted depth maps by NeRF.

3.3.2 Time as Additional Variable of NeRF Model

In order to model dynamic illumination, we propose to use time t as supplemental input variable to the network. Each of the elements of t corresponds to the sequential index of the input image for NeRF training. Time and images indices have linear dependence. This is due to the fact that the processed dataset is a video sequence with a fixed value of frames per second (fps). Variable t is additionally normalized to the $[0,1]$ interval, and positional encoding is applied. This variable can be added in two variants to the model. The first one corresponds to time t added with 3D coordinates denoted as $(\vec{x},t)$ and the second one is time t added with input rays directions denoted as $(\vec{d},t)$.

The modified NeRF model is defined as:

$$F_w : (\vec{x},\vec{d},t) \rightarrow (\vec{c},\sigma) \tag{5}$$

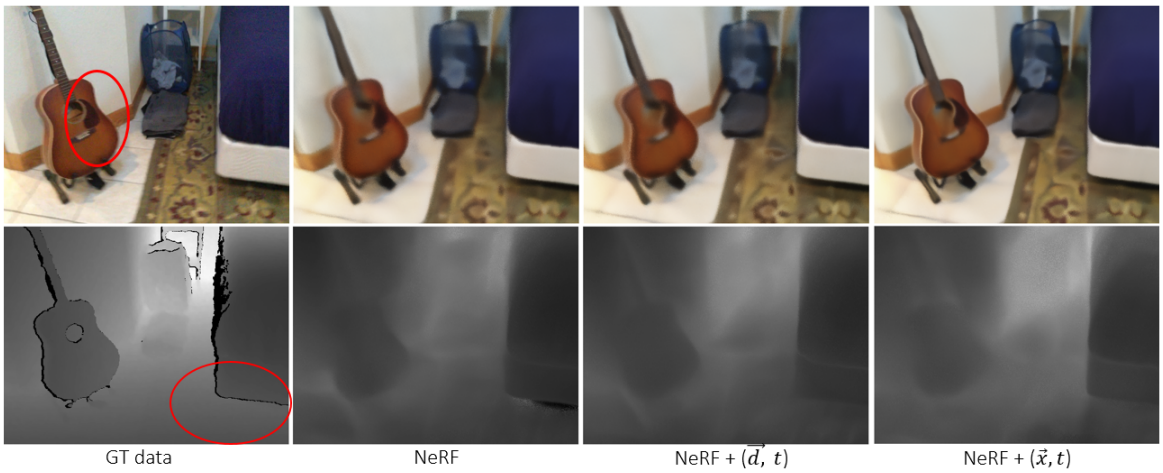

Figure 2: Qualitative comparisons for scene0000_00. First line corresponds to RGB images and second line to depth maps. Adding time to the NeRF model improve both quality of RGB images and depth maps as shown in highlighted regions.

4 EXPERIMENTS

4.1 Dataset Description

ScanNet [11] is an RGB-D dataset that consists of 2.5 million images collected in more than 1500 different indoor locations. Camera intrinsics and extrinsics (camera poses) are provided for each scene. In this paper, we use two ScanNet scenes: scene0000_00 and scene0005_01. For these scenes illumination changes with viewing angle because of reflective surfaces and also with time. Time dynamics is most probably caused by the camera auto exposure. Scene0000_00 consists of 5577 images and scene0005_01 consists of 1449 images. For experiments we use only a part of the scene0000_00 that contains 500 images. We select only a part that contains illumination changes.

4.2 Experiment Description. Metrics

To demonstrate the effect of our proposed modifications we conducted the following experiments:

1) Train original NeRF model.
2) Train original NeRF model with added loss function.
3) Train original NeRF model with both added depth loss function and time variable.

Training configuration including train/validation split, number of sampled points, number of random rays is the same as in [15].

We evaluate our models using two metrics: MAE (mean absolute error) and MARE (mean absolute relative error). We measure errors between GT depth and predicted one.

MAE can be calculated as:

$$MAE = \frac{1}{N}\frac{1}{n}\frac{1}{m}\sum_{k=1}^{N}\sum_{i,j}^{n,m}|y_{k,i,j}) - \hat{y}_{k,i,j}|, \quad (6)$$

where $y_{k,i,j}$ - GT depth value for (i, j) pixel, $\hat{y}_{k,i,j}$ - predicted depth value for (i, j) pixel, N - number of images in the dataset, n - image height, m - image width.

And MARE can be calculated as:

$$MARE = \frac{1}{N}\frac{1}{n}\frac{1}{m}\sum_{k=1}^{N}\sum_{i,j}^{n,m}\frac{|y_{k,i,j} - \hat{y}_{k,i,j}|}{y_{k,i,j}} * 100\%, \quad (7)$$

where $y_{k,i,j}$ - GT depth value for (i, j) pixel, $\hat{y}_{k,i,j}$ - predicted depth value for (i, j) pixel, N - number of images in the dataset, n - image height, m - image width.

4.3 Experiment Results

The performance metrics of trained models in different modes are shown in Table 1. For the scene0000_00 NeRF models were trained without adding depth loss function. Base model results in low performance, achieving 28.1% relative depth error. Adding time variable allow us to improve the model quality by 8-11%. Qualitative results for this dataset can be found in Figure 2. For dataset scene0005_01 we analyze both modifications: depth loss and additional time t variable. Modification with depth loss reduces the relative depth error from 30% to 1.88%. Adding time variable further improves quality of synthesized color images as well as depth maps. It is especially shown for case of adding time variable with the input 3D coordinates $(\vec{x}, t)$. Relative depth error in this mode is equal to 0.93%. Qualitative results for dataset

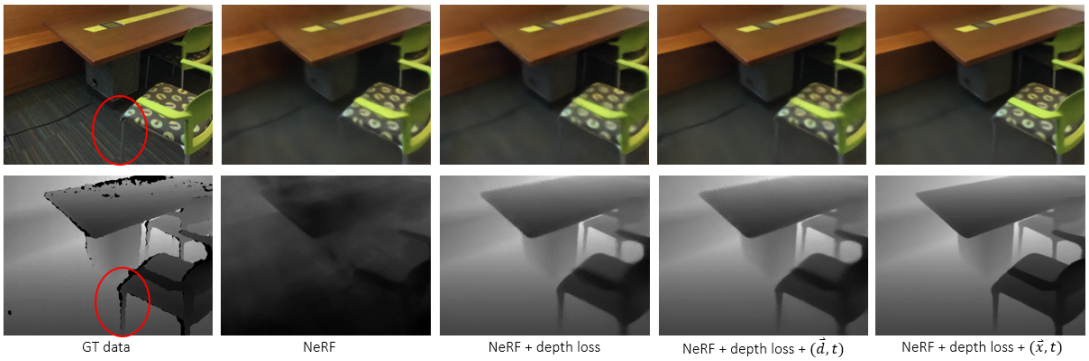

Figure 3: Qualitative comparisons for scene0005_01. First line corresponds to RGB images and second line to depth maps. Adding depth loss function as well as time variable to NeRF model improve synthesis quality and allow model to better generate thin structures of the scene (highlighted region).

scene0005_01 can be found in Figure 3. Visual comparison of generated image and depth data shows us quality improvement with our mode modifications.

Table 1: Metrics for ScanNet dataset.

Dataset	Train. mode	MAE	MARE
scene0000_00	base model	0.686m	28.1%
scene0000_00	$(\vec{d})$	0.498m	20%
scene0000_00	$(\vec{x},t)$	0.429m	17.3%
scene0005_01	base model	0.577m	30%
scene0005_01	depth loss	0.03m	1.881%
scene0005_01	depth loss + $(\vec{d})$	0.03m	1.875%
scene0005_01	depth loss + $(\vec{x},t)$	0.015m	0.93%

5 CONCLUSIONS

In this paper, we study how dynamic illumination affect quality of a scene representation by NeRF model. Dynamic illumination can be caused by illumination sources with power changing in time (sunlight in cloudy weather), light sources switched on or off during scene acquisition, or by camera automatic exposure. We argue that such changes cannot be modelled by standard NeRF using position and viewing angle direction as inputs and lead to rendering quality degradation. To cope with this problem, we propose to extend NeRF input with additional time variable. This idea was previously used for scenes with dynamic objects, we demonstrate that same approach is useful for static scenes with dynamic illumination. Experiments on ScanNet dataset show that extending NeRF input with time variable leads to improvement of quality of synthesized images (e.g. for small structures) and to relative depth error decreasing by 10-28%. From the practical point of view, results of this work can be used to improve quality of data augmentation for training depth prediction models where quality of both image and depth rendering is highly important.

REFERENCES

[1] L. Tardon, I. Barbancho, and C. Alberola-López, Markov Random Fields in the Context of Stereo Vision, 01 2011.

[2] S. Zhu and L. Yan, "Local stereo matching algorithm with efficient matching cost and adaptive guided image filter," vol. 33, no. 9, 2017. [Online]. Available: https://doi.org/10.1007/s00371-016-1264-6

[3] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in BMVC, January 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/patchmatch-stereo-stereo-matching-with-slanted-support-windows/

[4] H. Laga, L. V. Jospin, F. Boussaid, and M. Bennamoun, "A survey on deep learning techniques for stereo-based depth estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 4, pp. 1738–1764, apr 2022.

[5] J. Watson, O. M. Aodha, D. Turmukhambetov, G. J. Brostow, and M. Firman, "Learning stereo from single images." Berlin, Heidelberg: Springer-Verlag, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-58452-8_42

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing

scenes as neural radiance fields for view synthesis," in ECCV, 2020.

[7] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Lens: Localization enhanced by nerf synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2110.06558

[8] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," 2016. [Online]. Available: https://arxiv.org/abs/1609.03677

[9] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," in Proceedings of the 2020 Conference on Robot Learning, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 1908–1917. [Online]. Available: https://proceedings.mlr.press/v155/li21a.html

[10] K. Wang, Z. Zhang, Z. Yan, X. Li, B. Xu, J. Li, and J. Yang, "Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 16 055–16 064.

[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2017.

[12] M. Waechter, N. Moehrle, and M. Goesele, "Let there be color! large-scale texturing of 3d reconstructions," in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 836–850.

[13] C. M. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, and T. Funkhouser, "Local implicit grid representations for 3d scenes," in Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2020.

[14] E. Penner and L. Zhang, "Soft 3d reconstruction for view synthesis," vol. 36, no. 6, 2017.

[15] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in IEEE International Conference on Computer Vision (ICCV), 2021.

[16] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14 124–14 133.

[17] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo," in ICCV, 2021.

[18] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[19] J. T. Kajiya and B. V. Herzen, "Ray tracing volume densities," Proceedings of the 11th annual conference on Computer graphics and interactive techniques, 1984.