ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.30, No-1-2, 2022

Mip-NeRF RGB-D: Depth Assisted Fast Neural Radiance Fields

Arnab Dey
I3S-CNRS/Université Côte d'Azur
Sophia-Antipolis, France
adey@i3s.unice.fr

Yassine Ahmine
I3S-CNRS/Université Côte d'Azur
Sophia-Antipolis, France
ahmine@i3s.unice.fr

Andrew I. Comport
I3S-CNRS/Université Côte d'Azur
Sophia-Antipolis, France
Andrew.Comport@cnrs.fr

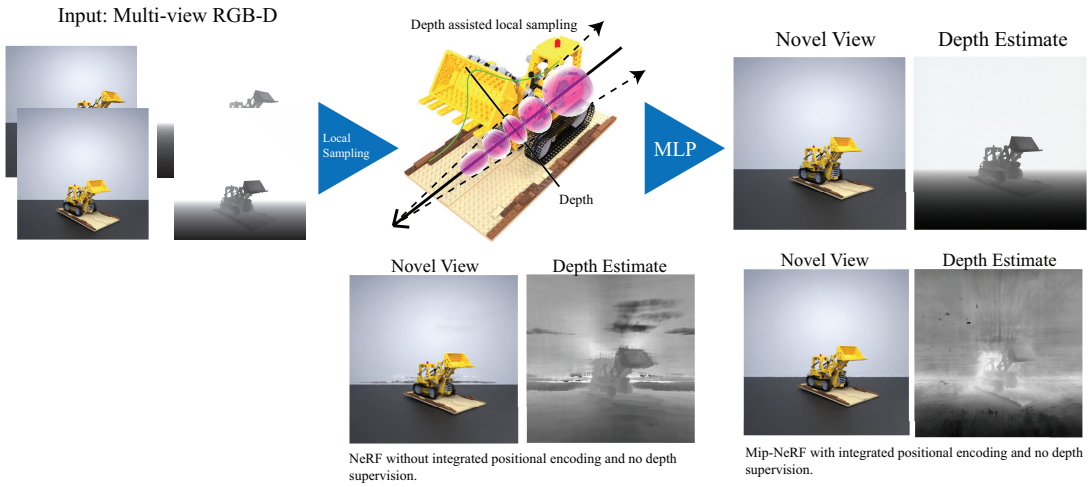

Figure 1: Mip-NeRF RGB-D uses RGB-D frames to represent 3D scenes using neural radiance fields. Depth information is used for local sampling and geometric loss. It produces significantly better photometry and geometry.

ABSTRACT

Neural scene representations, such as Neural Radiance Fields (NeRF), are based on training a multilayer perceptron (MLP) using a set of color images with known poses. An increasing number of devices now produce RGB-D(color + depth) information, which has been shown to be very important for a wide range of tasks. Therefore, the aim of this paper is to investigate what improvements can be made to these promising implicit representations by incorporating depth information with the color images. In particular, the recently proposed Mip-NeRF approach, which uses conical frustums instead of rays for volume rendering, allows one to account for the varying area of a pixel with distance from the camera center. The proposed method additionally models depth uncertainty. This allows to address major limitations of NeRF-based approaches including improving the accuracy of geometry, reduced artifacts, faster training time, and shortened prediction time. Experiments are performed on well-known benchmark scenes, and comparisons show improved accuracy in scene geometry and photometric reconstruction, while reducing the training time by 3 - 5 times.

Keywords

Computer vision, RGB-D NeRF, NeRF, Neural Scene Representation, Neural Rendering, Volume Rendering.

1 INTRODUCTION

Recent advances in neural scene representations [Sitzmann et al., 2019, Mildenhall et al., 2020] have demonstrated that neural networks can be used to represent 3D scenes as weights of a neural network for the purpose of rendering novel photorealistic views. Methods such as [Mildenhall et al., 2020, Saito et al., 2019, Lombardi et al., 2019] learned a volumetric representation from a sparse set of RGB images captured from color camera sensors. This method requires pre-

computation of camera poses and uses two multilayer perceptron networks to represent scene geometry and lighting effects. Although the NeRF models and their variants have shown impressive results, the underlying model is computationally inefficient, largely due to its volumetric search space for intersecting viewing rays, leading to extended training times. For example, volume rendering involves sampling points along each viewing ray (256 for NeRF) to calculate the color of the ray from the volume density and radiance of each sample point. Furthermore, the multiview triangulation problem is sometimes intractable from only images, which leads to artifacts and inaccurate geometry.

Although color-only approaches work well for applications that only have RGB images available, this approach can be improved by considering depth information alongside color. Many devices, including mobile phones, now include RGB-D sensors, and the aim of this paper is to investigate and devise a methodology to incorporate depth information into a neural scene representation.

Only a few methods have been proposed to take advantage of depth measurements simultaneously with color within the volumetric rendering pipeline [Neff et al., 2021, Deng et al., 2021]. However, these methods do not explicitly model the uncertainty of the sensor. To successfully incorporate noisy depth measurements into the volumetric rendering pipeline, the recent Mip-NeRF approach [Barron et al., 2021] provides a framework that accounts for the uncertainty of color pixel with varying depth by replacing classic 3D ray sampling with conic region sampling. This approach provides an elegant framework for including multivariate Gaussian uncertainty and will be extended in this paper to include depth uncertainty.

In this paper, it will be demonstrated that considering depth information can improve geometry considerably compared to only color information at several different levels. First, this method shows how local sampling along the rays, guided by surface information from RGB-D frames, can reduce the number of samples along the ray and replace the coarse network of NeRF. Second, a joint color-and-depth-loss term will be shown to allow the network to learn the geometry and color of the scene from a limited number of input views. Third, the proposed method shows how depth uncertainty can be incorporated into a multivariate Gaussian method to query the MLP. Finally, an adaptive training method will be proposed that allows the network to learn multiple scales of uncertainty within the representation.

To sum up, the proposed method is based on a RGB-D neural radiance field combined with an implicit occupancy representation that takes into account both color and depth observations.

The key contributions of the article are summarized as follows:

- Depth information is used for efficient sampling.
- The representation is optimized simultaneously on scene geometry and photometry.
- Depth uncertainty is handled adaptatively via a new local sampling strategy.

2 RELATED WORK

The proposed Mip-NeRF RGB-D uses a set of RGB-D inputs to learn a volumetric scene representation of the observed scene using a multilayer perceptron by leveraging both depth and color information. In the following, related work to this research will be discussed.

2.1 *Novel view synthesis from images*

Image-based view synthesis uses a number of techniques to generate novel images, such as transforming or warping an existing set of images using estimated geometry and camera poses to create novel views [Hedman et al., 2016, Gortler et al., 1996]. [Heigl et al., 1999] used a sequence of images and directly rendered the views by projective mapping of all images to a common plane of mean geometry. To generate a novel view from a set of captured images of different poses requires blending them to target views; even though the geometry of the static objects is constant in different views, the appearance can change depending on lighting and object properties. To overcome these drawbacks [Hedman et al., 2018, Thies et al., 2020] used artificial neural networks to reduce artifacts and view-dependent effects in the generated novel views.

2.2 *Implicit neural surface representation*

These methods use neural networks to learn a neural surface representation of the object using voxels, meshes, and point cloud data. Although they are capable of achieving impressive results, they are limited by their internal resolution and high-frequency details. Mescheder et al. [Mescheder et al., 2019] used a neural network to learn a continuous 3D occupancy function; Given 3D points as input to an occupancy network, the network predicts binary occupancy at that 3D location. Later, [Chen and Zhang, 2019] used an MLP to predict occupancy from a feature vector and the 3D coordinates of the location. On the other hand, [Park et al., 2019] learned a signed distance function(SDF) instead of occupancy to improve the quality of the reconstruction. [Saito et al., 2019] showed that it is possible to infer 3D surfaces and texture from a single image using an implicit function.

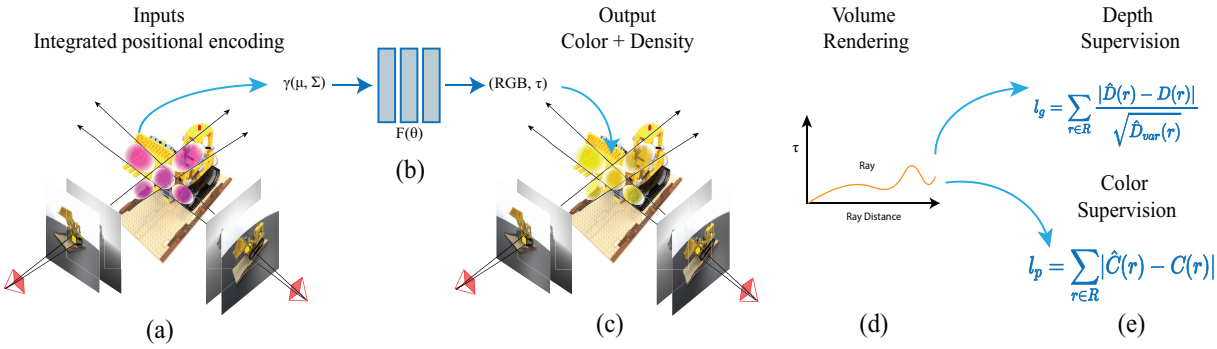

Figure 2: An overview of the proposed Mip-NeRF RGB-D. **(a)** The input to the network is the integrated positional encoding of a conical frustum segment. **(b)** The network outputs volume density and color. **(c)** The color and depth of a ray is generated using the classic volume rendering method. **(d)** The network is optimized using a color and depth loss.

2.3 *Neural volume rendering*

Lombardi et al. [Lombardi et al., 2019] initially introduced volume rendering for novel view synthesis using a CNN-based encoder and an MLP-based decoder to produce density and color for each point in space. The well-known Neural Radiance Fields approach was introduced in [Mildenhall et al., 2020] and demonstrated compelling results with a simple method that takes 3D points and the associated view direction as input to an MLP and outputs density and color. Some of the drawbacks of NeRF are its long training time, its long rendering time, the need to train a separate model for each scene, and it only works on static scenes. Various investigations have been conducted since the original NeRF to address these problems. [Liu et al., 2020, Neff et al., 2021, Garbin et al., 2021, Reiser et al., 2021, Yu et al., 2021] addressed the slow inference time of NeRF by using a tiny MLP and a better sampling strategy. [Deng et al., 2021] used sparse depth supervision during training to improve the training time of the NeRF model. [Pumarola et al., 2020, Gafni et al., 2020, Noguchi et al., 2021] address the problem of static scenes. [Yu et al., 2020, Schwarz et al., 2020, Tancik et al., 2020, Chan et al., 2020] have generalized NeRF models using fully convolutional image features, a generator discriminator, and meta-learning. Saito et al.[Barron et al., 2021] focused on NeRF aliasing and sampling problems. They proposed an integrated positional encoding, which uses a conical frustum defined by the mean and covariance of the rays, and the neural radiance field is integrated over the region represented by 3D Gaussian encoding.

2.4 *Neural radiance field with depth*

To solve the problem of incorrect geometry prediction when a limited number of input views are given, [Deng et al., 2021] proposed using depth as alternate supervision. Ds-NeRF uses a sparse 3D point cloud and then reprojects the errors between the detected 2D keypoints and projected 3D points, generated by commonly used structure-from-motion (SfM) algorithms which are error-prone. They optimize the model over a combined color and depth loss function. Similarly, NerfingMVS [Wei et al., 2021] uses a monocular depth network to generate depth prior from SfM reconstruction of the scene. The adapted depth priors are used to guide the sampling process of points along the ray. Unlike DS-NeRF, it generates a dense depth prior from sparse SfM points using a pretrained depth network. Azinovic et al.[Azinović et al., 2021] also demonstrated the incorporation of depth with NeRF to produce a better and more detailed reconstruction than simply using color or depth alone. Unlike others, it uses a truncated signed distance function(TSDF) instead of volume density to represent the underlying geometry. It still uses two networks that significantly affect training and prediction time. On the other hand, iMap [Sucar et al., 2021] shows that NeRF can be used to represent scenes in a real-time SLAM system. It jointly optimizes the 3D map and camera pose using keyframes. iMap uses a smaller MLP (4 layers) than NeRF and does not consider the viewing direction to model lighting effects. DONeRF [Neff et al., 2021] proposed a compact dual network design to reduce evaluation cost, leading to a faster prediction time. The coarse NeRF network is replaced by a *depth oracle network* based on a classification network. To reduce the number of samples along the rays, they suggested nonlinear transformation and a local sampling strategy, which helped them to achieve a similar result to NeRF with a fraction of the samples, but the method is limited to forward facing scenes.

3 METHODS

Now the proposed Mip-NeRF RGB-D will be presented. First, the implicit scene representation used by

NeRF based methods will be over-viewed, followed by the explanation of the rendering process. After that, an efficient network architecture will be proposed and a joint optimization method using RGB-D data will be presented. Finally, the local sampling strategy used to reduce the number of samples along the rays and reduce training time will be described.

3.1 Implicit scene representation

The proposed system is based on the Mip-NeRF [Barron et al., 2021] method which is an extension of NeRF for handling anti-aliasing. Vanilla NeRF based methods use a set of images and corresponding poses to train a MLP network that represents the scene by outputting the emitted radiance and volume density of 3D locations. Given 5D coordinates(3D location + viewing direction) as input, the network F_Θ learns an implicit function that estimates color $C = (r,g,b)$ and volume density τ as:

$$F_\Theta : (x,y,z,\theta,\phi) \rightarrow (C,\tau). \tag{1}$$

First, rays $r(t) = \mathbf{o} + t\mathbf{d}$ passing through each pixel of the image are generated, where the ray origin $\mathbf{o}$ is the camera center and $\mathbf{d}$ is the ray direction. Then N sample points are placed along the ray stratified manner between predefined near and far bounds. The color of each pixel is computed using a radiance and a volume density along the ray. In Mip-NeRF the rays are replaced with cones generated using the camera center and the pixel size. The cone is split into N intervals $\mathcal{T}_) = [t_i, t_i + 1)$ and for each interval the integrated positional encoding of the mean and the covariance (μ,Σ) of the corresponding conical frustum is computed. Integrated positional encoding encodes the Gaussian approximation of the conical frustum as follows:

$$\gamma(\mu,\Sigma) = \left\{ \begin{bmatrix} \sin(2^l\mu) exp(-2^{l-1} diag(\Sigma)) \\ \cos(2^l\mu) exp(-2^{l-1} diag(\Sigma)) \end{bmatrix} \right\}_{l=0}^{L-1}, \tag{2}$$

where Σ is the covariance of the Gaussian approximation:

$$\Sigma = \sigma_t^2 (\mathbf{d}\cdot\mathbf{d}^{\mathrm{T}}) + \sigma_r^2 \left(I - \frac{\mathbf{d}\cdot\mathbf{d}^{\mathrm{T}}}{||\mathbf{d}||_2^2} \right). \tag{3}$$

The variance along the ray is denoted by σ_t^2 and the variance perpendicular to the ray is σ_r^2. Mip-NeRF uses this integrated positional encoding instead of the frequency positional encoding as input to the neural network. One of the key difference between Mip-NeRF and the proposed method is the local sampling strategies, which will be discussed in the subsequent part of this article.

3.2 Volume rendering

Similarly to NeRF, a volume rendering formula was used to calculate the color and the depth of pixels from radiance and volume density of the conical frustum. The volume density $\tau(P)$ at location $P = (x,y,z)$ can be interpreted as the differential probability of ray termination. The expected color $C(r)$ of a camera ray $r(t) = \mathbf{o} + t\mathbf{d}$ with near and far bounds t_n and t_f is:

$$C(r) = \int_{t_n}^{t_f} T(t)\tau(r(t))c(r(t),\mathbf{d})dt, \tag{4}$$

where

$$T(t) = exp(-\int_{t_n}^{t} \tau(r(s))ds). \tag{5}$$

The function $T(t)$ denotes the accumulated transmittance along the ray from t_n to t, i.e., the probability that the ray travels from t_n to t without hitting any other particle. In the stratified sampling approach $[t_n, t_f]$ is partitioned into N evenly-spaced bins and then one sample is drawn uniformly at random from within each bin. The samples are used to estimate predicted color $\widehat{C}(r)$ as:

$$\widehat{C}(r) = \sum_{i=1}^{N} T_i (1 - exp(-\tau_i\delta_i))c_i, \tag{6}$$

where

$$T_i = exp(-\sum_{j=1}^{i-1} \tau_j\delta_j). \tag{7}$$

Here $\delta_j = t_{j+1} - t_j$ is the distance between adjacent samples. Similarly [Wei et al., 2021], the depth can be represented with volume density using:

$$\widehat{D}(r) = \sum_{i=1}^{N} T_i (1 - exp(-\tau_i\delta_i))t_i, \tag{8}$$

where T_i is the accumulated transmittance. To optimize the network, NeRF uses a squared error between the rendered and true pixel colors.

3.3 Optimization

The network parameters θ are optimized using a set of RGB-D frames, each of which has a color, depth, and camera pose information. The proposed method minimizes the geometric and photometric loss together on a set of frames as the rendering functions are completely differentiable. The photometric loss l_p is the absolute difference (L1-norm)[Sucar et al., 2021] between the predicted color and the ground truth color of the ray. The photometric loss over a set of rays is defined as:

$$l_p = \sum_{r\in R} |\hat{C}(r) - C(r)|. \tag{9}$$

The geometric loss is the absolute difference between predicted and true depths, normalized by the depth variance [Sucar et al., 2021] to discourage weights with high uncertainty:

$$l_g = \sum_{r\in R} \frac{|\hat{D}(r) - D(r)|}{\sqrt{\hat{D}_{var}(r)}}, \tag{10}$$

where $\hat{D}_{var}(r) = \sum_{i=1}^{N} T_i(1 - exp(-\tau_i\delta_i))(\hat{D}(r) - t_i)^2$ depth variance of the image. The neural network can be optimized by combining photometric, and geometric losses together using empirically chosen scale factors λ_p:

$$min_\theta(l_g + \lambda_p l_p). \quad (11)$$

3.4 Network architecture

The network architecture is similar to the original NeRF with some modifications. The proposed method uses only one network with 4 hidden layers of feature size 256. The skip connection is used in layer 3. The viewing direction is concatenated to the fourth layer before the color and volume density are output. The integrated positional encoding was applied to the conical frustum and a positional encoding of frequency 4 is applied to the viewing directions as was done in Mip-NeRF. By decreasing the network size, faster training and prediction time were achieved without significantly compromising the novel view quality.

3.5 Local sampling

NeRF based methods estimate pixel color by placing samples on viewing rays traced through the pixels. The final color of the pixels is calculated by the alpha composition [Max, 1995] of the volume density and the radiances of the samples along the ray. Samples relevant to the volume produce higher volume density, so samples close to the surface are more relevant to the pixel's final color. NeRF uses 256 samples in a stratified manner and 2 networks to ensure that samples are placed on relevant parts of the ray. To compute a pixel color, each sample on that ray needs a full network evaluation, so the training time increases exponentially with the number of samples on the ray. Although Mip-NeRF uses a conical frustum instead of viewing rays, it still requires 2 network passes and a large number of bins to create integrated positional encoding. In reality, the majority of the scene volume is empty space (for 360 scenes), and the samples placed on the empty space have less contribution to the final color. Therefore, given the depth information of an image, it is possible to place fewer samples and to place them directly on the relevant parts of the ray, while achieving similar quality results. Finally, in this case, it is also possible to eliminate the coarse network with local sampling, which NeRF uses to find important sampling locations along the ray. Various depth-guided sampling strategies have been considered. Figure:3 shows the comparison between the proposed local sampling and the baseline approaches.

3.5.1 Stratified sampling

The so-called stratified sampling strategy is very similar to the original Mip-NeRF sampling, but the conical

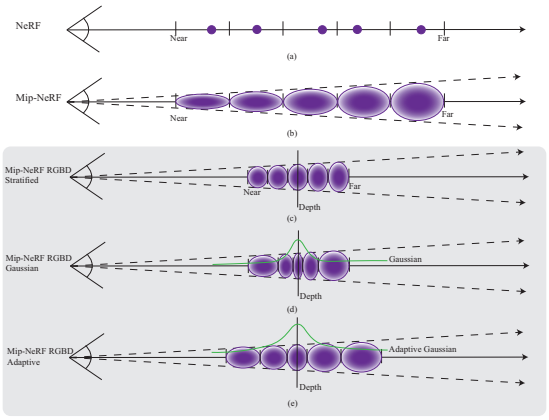

Figure 3: Visualization of the proposed sampling strategies(c, d, e) compared to NeRF(a) and Mip-NeRF(b). Black arrows represent ray direction and purple ellipsoids represent a Gaussian approximation of the conical frustum.

frustums are generated only close to the surface based on depth information(Figure:3(c)). Here the space between the near and far bounds $[t_n, t_f]$ is divided into N evenly spaced bins and a sample is drawn uniformly at random from each bin where $t_n = D - \alpha_n$ and $t_f = D + \alpha_f$, α_n and α_f are empirically chosen based on the depth uncertainty. The samples are then used as the bounds of the conical frustum. This allows the network to avoid empty space and eventually decrease the number of bins needed for each ray.

3.5.2 Gaussian sampling

In this strategy, instead of placing the bins equidistantly around the surface, the limits of the conical segments are selected from a normal distribution where the mean is the depth and the standard deviation ς is empirically chosen based on the depth uncertainty. In this way, it ensured that the conical frustums are smaller (toward the ray direction) on the relevant part of the ray(close to the true depth as in Figure:3(d)) to emphasize the high-frequency details on the surface. This allows the network to handle the generalized uncertainty present in the depth estimate.

3.5.3 Adaptive sampling

The adaptive sampling strategy uses a normal distribution with a varying standard deviation $\varsigma(r)$ based on the number of epochs and the depth of the ray. Therefore, the normal distribution (the mean is the depth measurement) is used to define the limits of the conical frustums(Figure:3(e)). $\varsigma(r)$ varies during training according to the number of epochs in a coarse to fine manner to improve the fine photometric details. Additionally, this sampling strategy takes into account the depth uncertainty, which increases with distance. The standard deviation of each ray is calculated as follows:

$$\varsigma(r) = \frac{D(r)}{4}(\exp^{-\lambda_r i} + \lambda_m), \quad (12)$$

where i is the epoch number, λ_r is the rate of decrease, λ_m is minimum standard deviation, and $D(r)$ is the true depth of the ray. λ_r and λ_m are empirically chosen based on dataset and depth uncertainty.

4 EXPERIMENTAL RESULTS

In this section, the proposed methods are evaluated on various datasets and compared with other state-of-the-art NeRF based methods.

4.1 Experimental Setup

4.1.1 Datasets

Simulated datasets were used for all experiments. Each data set contained RGB images, depth maps, and their corresponding camera poses. All poses in the datasets belong to an upper hemisphere, where the object is placed in the center. Four different scenes were considered for the experiments: Lego[1], Cube, Human [2], and Drums[3]. The input images have 800×800 resolution, and the depth measurements are in meters. Each of the datasets has three versions, in which the number of training images is 8, 30, and 100.

4.1.2 Implementation Details

The proposed method is implemented using a combination of PyTorch and CUDA. The ADAM optimizer with a learning rate of 5×10^{-4} and an exponential decay of the learning rate of 5×10^{-1} in every five epoches has been used. A batch size of 2048 on 2 Nvidia Rtx 3090 GPUs was used for all experiments. 16 frequency bands were used for integrated positional encoding of the conical frustum and 4 frequency bands to encode viewing directions with positional encoding.

For all experiments, the following parameters are used as default: the photometric scale factor for the loss function is $\lambda_p = 100$, standard deviation for Gaussian sampling is 0.3, $\lambda_r = 0.09$ and $\lambda_m = 0.1$ for adaptive sampling.

4.1.3 Metrics

Four metrics are used to evaluate the predicted RGB image quality and depth map: *Peak Signal-to-Noise Ratio (PSNR in dB)*: to compare the quality of the RGB reconstruction, the higher is better; *Absolute Relative distance* (Abs Rel in m): to compare the quality of the generated depth map, the lower is better; *Structural Similarity Index* (SSIM in %) [Wang et al., 2004]: quantifies the degradation of image quality in the reconstructed image, the higher is better; *Learned Perceptual Image Patch Similarity* (LPIPS) [Zhang et al., 2018]: the distance between the patches of the image, the lower means that the patches are more similar.

[1] https://www.blendswap.com/blend/11490
[2] https://renderpeople.com/free-3d-people/
[3] https://www.blendswap.com/blend/13383

4.2 Comparison

First, local sampling strategies are compared in Section 4.2.1. Then, the effect of a different number of samples on the proposed model is discussed. After that, the proposed method is applied in different scenes. Finally, in Section 4.2.5, the proposed method is compared with other NeRF-based methods that use depth supervision.

4.2.1 Comparison between different local sampling strategies

Instead of placing samples over the entire ray, local sampling places samples only on the relevant regions of the ray using depth information. In this section, proposed local sampling strategies are compared. Table 1 shows the quantitative performance of the three sampling strategies mentioned.

	Metrics			
Strategy	PSNR ↑	SSIM ↑	Abs Rel↓	LPIPS ↓
Equidistant	19.86	0.87	0.02	0.0023
Gaussian	20.75	0.88	0.04	0.0022
Adaptive	21.09	0.89	0.04	0.0024
NeRF	19.21	0.88	0.34	0.0036

Table 1: Comparison of three different sampling strategies. The Lego spherical dataset containing 8 training images has been used for all experiments. All experiments used 16 sampling points per ray. Best values are highlighted by green, significant wrose values by red, and darker shades represent best values.

The results show that adaptive sampling performs best among the proposed sampling strategies. All local sampling strategies improve the underlying geometry compared to NeRF.

4.2.2 Effect of ray sample size

NeRF based methods use a large number of ray samples to estimate volume density and produce fine output details. The number of samples is arguably the most important parameter for any NeRF model because it is directly related to the training time, the prediction time, and the quality of the novel view. The following experiments in Table 2 demonstrate that the proposed method performs significantly well even when the number of samples is low. The 64 samples provide the best compromise between novel view quality and training time.

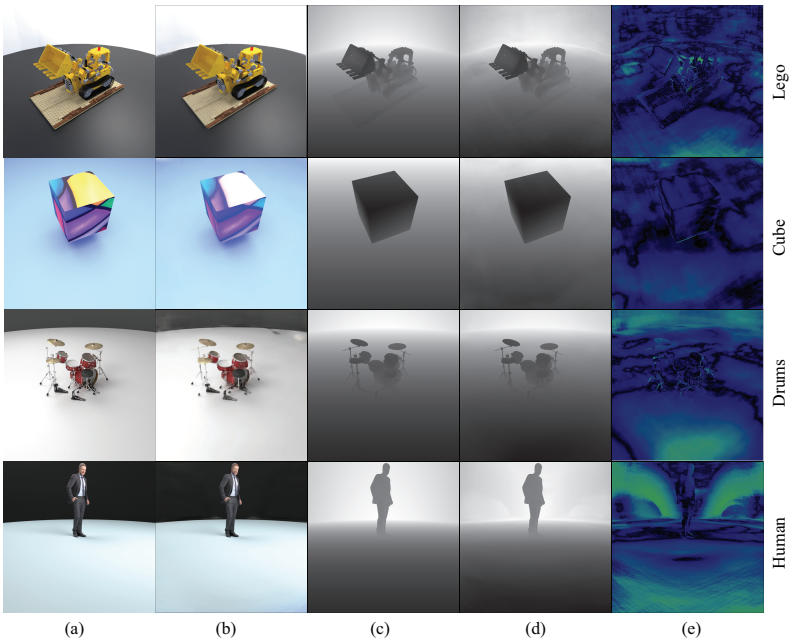

Figure 4: Qualitative comparison on blender scenes: Visual comparison between generated RGB ground truth images and the true depth maps generated by the proposed method, where (a) Ground truth RGB images; (b) Predicted RGB images; (c) True depth maps; (d) Predicted depth maps; (e) Absolute error between predicted and true depth maps.

	Number of samples		
Metrics	16	64	128
PSNR ↑	19.4	21.18	22.13
SSIM ↑	0.86	0.89	0.9
Abs Rel ↓	0.05	0.04	0.05
LPIPS ↓	0.0025	0.0023	0.0018
Training Time ↓	38m	44m	1.54h

Table 2: A comparison between training time and novel view quality based on the number of samples per ray.

4.2.3 *Different datasets*

The proposed method was evaluated with 4 different datasets with different characteristics to demonstrate its robustness in different types of scenes. The cube has a simple geometry but a complicated texture. Alternatively, the drums have a complicated and very detailed 3D structure. The Lego scene is a good mix of photometric and geometric details. The human scene mimics some real-world applications. Quantitative results are shown in Table 3 and qualitative results are shown in Figure 4.

	Metrics			
dataset	PSNR↑	SSIM↑	AbsRel↓	LPIPS↓
Lego	28.21	0.93	0.02	0.0013
Cube	22.78	0.95	0.01	0.0001
Human	37.7	0.97	0.02	0.00008
Drums	27.85	0.9	0.02	0.0012

Table 3: Performance of the proposed method on 4 different simulated datasets.

4.2.4 *Fewer input views*

To demonstrate that the proposed method can perform well even when the number of training images is limited, three different datasets with different numbers of training images have been considered for experiments. Table 4 shows a comparison between different datasets. Although increasing the number of inputs increases the quality of the novel view, the training time also increases significantly.

	Number of input views		
Metrics	8	30	100
PSNR ↑	21.18	25.25	28.21
SSIM ↑	0.89	0.92	0.93
Abs Rel ↓	0.04	0.04	0.02
LPIPS ↓	0.0023	0.0017	0.0013
Training Time ↓	44m	2.15h	6.45h

Table 4: More samples in the training set provides more supervision for the network to learn the scene representation. With an increasing number of input views, geometric and photometric metrics improve. Subsequently, training time increases significantly. A dataset with 100 images takes 11 times longer to train than dataset with 8 images.

4.2.5 *Scene representation*

In this section, the proposed method is compared with other state-of-the-art NeRF-based methods.

NeRF [Mildenhall et al., 2020]: The implementation of PyTorch Lighting of the NeRF by [Quei-An, 2020]

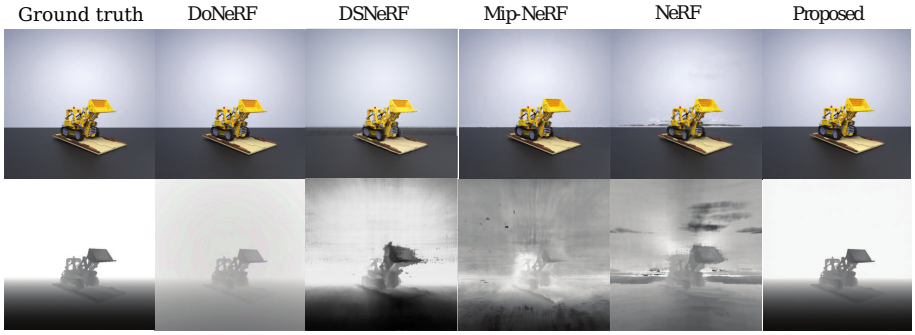

Figure 5: Qualitative comparison: Visual comparison results between the proposed method and other state-of-the-art methods.

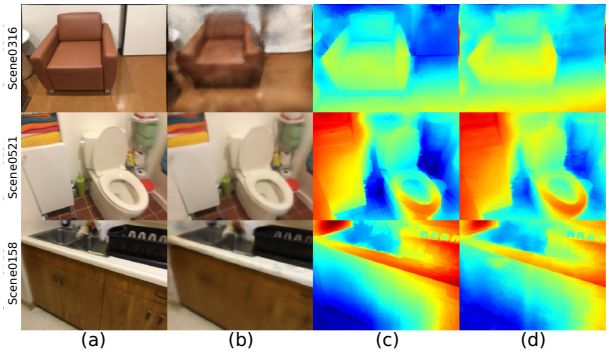

Figure 6: Qualitative result of the proposed method on 3 different real sensor datasets. (a) Ground truth RGB images; (b) Predicted RGB images; (c) Ground-truth depth maps; (d) Predicted depth maps.

	Metrics				
Method	PSNR ↑	SSIM ↑	Abs Rel↓	LPIPS ↓	Time ↓
DSNeRF	29.31	0.87	0.489	0.003	3:37h
DONeRF	39.23	0.98	0.008	0.00001	5:08h
NeRF	27.36	0.94	0.34	0.0015	3.40h
MipNeRF	30.66	0.95	0.334	0.006	1:27h
Proposed	32.72	0.95	0.001	0.0004	1.15h

Table 5: Quantitative comparison for novel view synthesis and depth estimation between the proposed method and state-of-the-art methods. The Lego dataset is used for all these experiments.

	Metrics			
dataset	PSNR↑	SSIM↑	AbsRel↓	LPIPS↓
scene0521	25.48	0.724	0.025	0.0004
scene0316	16.99	0.57	0.05	0.001
scene0158	24.93	0.74	0.02	0.0007

Table 6: Performance of the proposed method on 4 different real RGB-D datasets.

has been considered for the experiments. NeRF can be trained using simulated 360-degree Blender data or real data. For these particular experiments, simulated Blender scenes were used.

DSNeRF [Deng et al., 2021]: DSNeRF works only on the forward facing scenes where depth supervision data is generated using Colmap [Schonberger and Frahm, 2016]. The official implementation of DSNeRF was used for these experiments.

DONeRF [Neff et al., 2021]: DONeRF works only on forward-facing datasets where all poses belong to a view cell. This method works only with simulated data with a dense depth map. The official implementation of the DONeRF was used for the experiments.

Mip-NeRF [Barron et al., 2021]: The official Mip-NeRF implementation on JAX was converted to PyTorch for convenience of comparison.

All experiments were carried out on the same Lego scene dataset that contains 30 training images with resolution 800×800. The quantitative results in Table: 5 and the qualitative results in Figure: 5 show that DONeRF [Neff et al., 2021] can produce the best photometric quality, but is limited to forward-facing scenes, a longer training time, and oracle network-based depth prediction, where the proposed method uses only one smaller network (less space requirement), trains faster (4 times faster), and produces more accurate geometry.

Three different real-world datasets have been used from the NerfingMVS [Wei et al., 2021] for experimenting on real acquired depth images. Pre-processed depth maps were used instead of using raw depth maps because the raw depth maps contain areas without depth information (holes where sensors cannot estimate depth). NerfingMVS uses a monocular depth

prediction network to complete the missing depths. Alternatively, holes in the raw depth maps could be handled as in the classic RGB NeRF implementation, however, this randomly affects computational performance and comparisons. Table:6 shows the quantitative results of the proposed method on 3 different datasets. The adaptive sampling strategy with 16 samples was used for all of these experiments. The qualitative results of the experiments are shown in the Figure:6. The ground-truth depth shows that it is not very detailed and that some areas have wrong depth measurements, which results in some artifacts in the predicted image generated by the proposed method.

4.3 Analysis

Fewer views: The proposed method can learn a scene representation from fewer views, as depth supervision provides additional supervision and effective sampling. Depth supervsion allows the network to learn scene geometry and multi-view constancy from a very limited number of views.

Faster training: The results show a quantifiable speed improvement in training time with the proposed method compared to other state-of-the-art methods. Faster training was achieved using fewer samples, a smaller network architecture, and local sampling. The Mip-NeRF RGB-D method is $3-5\times$ faster compared to other similar NeRF-based methods.

Accurate depth estimation: The proposed method is capable of producing a more accurate geometry compared to other state-of-the-art methods. The network can learn accurate geometry from small number of inputs as few as 8 frames.

5 DISCUSSION

In this article, a new method was presented for representing 3D scenes from RGB-D data using recent neural radiance fields. Instead of learning the radiance field from RGB images, the proposed method uses RGB-D frames, which allows achieving better underlying geometry and faster training and prediction times. Additional depth supervision of dense depth maps is shown to have a significant improvement on the training time through local sampling. The proposed method trains $3-5\times$ faster and improves the novel view and depth quality. The experiments show significant improvements over the state-of-the-art methods, both quantitatively and qualitatively. Future perspective will be focused on extending this approach to dynamic scenes.

6 ACKNOWLEDGEMENTS

This project has received funding from the H2020 COFUND program BoostUrCareer under Marie Sklodowska-Curie grant agreement no. 847581. It also received funding from the EU H2020 MEMEX research project under grant agreement No. 870743.

7 REFERENCES

[Azinović et al., 2021] Azinović, D., Martin-Brualla, R., Goldman, D. B., Nießner, M., and Thies, J. (2021). Neural rgb-d surface reconstruction. *arXiv preprint arXiv:2104.04532*.

[Barron et al., 2021] Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., and Srinivasan, P. P. (2021). 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.

[Chan et al., 2020] Chan, E., Monteiro, M., Kellnhofer, P., Wu, J., and Wetzstein, G. (2020). pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. *https://arxiv.org/abs/2012.00926*.

[Chen and Zhang, 2019] Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948.

[Deng et al., 2021] Deng, K., Liu, A., Zhu, J.-Y., and Ramanan, D. (2021). Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*.

[Gafni et al., 2020] Gafni, G., Thies, J., Zollhöfer, M., and Nießner, M. (2020). Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. *https://arxiv.org/abs/2012.03065*.

[Garbin et al., 2021] Garbin, S. J., Kowalski, M., Johnson, M., Shotton, J., and Valentin, J. (2021). Fastnerf: High-fidelity neural rendering at 200fps. *https://arxiv.org/abs/2103.10380*.

[Gortler et al., 1996] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The lumigraph. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54.

[Hedman et al., 2018] Hedman, P., Philip, J., Price, T., Frahm, J.-M., Drettakis, G., and Brostow, G. (2018). Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15.

[Hedman et al., 2016] Hedman, P., Ritschel, T., Drettakis, G., and Brostow, G. (2016). Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):1–11.

[Heigl et al., 1999] Heigl, B., Koch, R., Pollefeys, M., Denzler, J., and Van Gool, L. (1999). Plenoptic modeling and rendering from image sequences taken by a hand-held camera. *Mustererkennung 1999, Springer*, pages 94–101.

[Liu et al., 2020] Liu, L., Gu, J., Lin, K. Z., Chua, T.-

S., and Theobalt, C. (2020). Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33.

[Lombardi et al., 2019] Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., and Sheikh, Y. (2019). Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*.

[Max, 1995] Max, N. (1995). Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108.

[Mescheder et al., 2019] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470.

[Mildenhall et al., 2020] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer.

[Neff et al., 2021] Neff, T., Stadlbauer, P., Parger, M., Kurz, A., Mueller, J. H., Chaitanya, C. R. A., Kaplanyan, A. S., and Steinberger, M. (2021). DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum*, 40(4).

[Noguchi et al., 2021] Noguchi, A., Sun, X., Lin, S., and Harada, T. (2021). Neural articulated radiance field. *arXiv preprint arXiv:2104.03110*.

[Park et al., 2019] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174.

[Pumarola et al., 2020] Pumarola, A., Corona, E., Pons-Moll, G., and Moreno-Noguer, F. (2020). D-NeRF: Neural radiance fields for dynamic scenes. *https://arxiv.org/abs/2011.13961*.

[Quei-An, 2020] Quei-An, C. (2020). Nerf_pl: a pytorch-lightning implementation of nerf. `https://github.com/kwea123/nerf_pl/`.

[Reiser et al., 2021] Reiser, C., Peng, S., Liao, Y., and Geiger, A. (2021). Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps.

[Saito et al., 2019] Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314.

[Schonberger and Frahm, 2016] Schonberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113.

[Schwarz et al., 2020] Schwarz, K., Liao, Y., Niemeyer, M., and Geiger, A. (2020). Graf: Generative radiance fields for 3D-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33.

[Sitzmann et al., 2019] Sitzmann, V., Zollhöfer, M., and Wetzstein, G. (2019). Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*.

[Sucar et al., 2021] Sucar, E., Liu, S., Ortiz, J., and Davison, A. J. (2021). imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238.

[Tancik et al., 2020] Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P., Barron, J. T., and Ng, R. (2020). Learned initializations for optimizing coordinate-based neural representations. *https://arxiv.org/abs/2012.02189*.

[Thies et al., 2020] Thies, J., Zollhöfer, M., Theobalt, C., Stamminger, M., and Nießner, M. (2020). Image-guided neural object rendering. *8th International Conference on Learning Representations*.

[Wang et al., 2004] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

[Wei et al., 2021] Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., and Zhou, J. (2021). Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619.

[Yu et al., 2021] Yu, A., Li, R., Tancik, M., Li, H., Ng, R., and Kanazawa, A. (2021). Plenoctrees for real-time rendering of neural radiance fields. *arXiv preprint arXiv:2103.14024*.

[Yu et al., 2020] Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2020). pixelNeRF: Neural radiance fields from one or few images. *https://arxiv.org/abs/2012.02190*.

[Zhang et al., 2018] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.