Non-uniform Sampling Strategies for NeRF on 360° images

Takashi Otonari[1]
otonari@hal.t.u-tokyo.ac.jp
Satoshi Ikehata[1,2]
sikehata@nii.ac.jp
Kiyoharu Aizawa[1]
aizawa@hal.t.u-tokyo.ac.jp

[1] The University of Tokyo
Tokyo, Japan
[2] National Institute of Informatics
Tokyo, Japan

Abstract

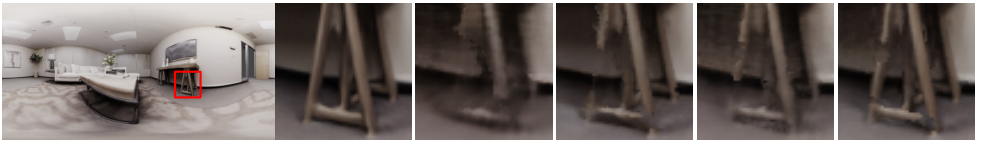
In recent years, the performance of novel view synthesis using perspective images has dramatically improved with the advent of neural radiance fields (NeRF). This study proposes two novel techniques that effectively build NeRF for 360° omnidirectional images. Due to the characteristics of a 360° image of ERP format that has spatial distortion in their high latitude regions and a 360° wide viewing angle, NeRF's general ray sampling strategy is ineffective. Hence, the view synthesis accuracy of NeRF is limited and learning is not efficient. We propose two non-uniform ray sampling schemes for NeRF to suit 360° images – distortion-aware ray sampling and content-aware ray sampling. We created an evaluation dataset *Synth360* using Replica and SceneCity models of indoor and outdoor scenes, respectively. In experiments, we show that our proposal successfully builds 360° image NeRF in terms of both accuracy and efficiency. The proposal is widely applicable to advanced variants of NeRF. DietNeRF, AugNeRF, and NeRF++ combined with the proposed techniques further improve the performance. Moreover, we show that our proposed method enhances the quality of real-world scenes in 360° images. Synth360: https://drive.google.com/drive/folders/1suL9B7DO2no21ggiIHkH3JF3OecasQLb.

1 Introduction

Synthesizing a view from other views is a long-standing problem in computer vision and graphics. With recent emerging interest in virtual and augmented reality, this technology is expected to support applications such as virtual tours and immersive 3-D games, where immersion in large, unbounded photorealistic virtual space is possible. In such applications, taking images of an entire scene with a camera of narrow field-of-view (FoV) is prohibitively tedious; therefore, view synthesis with 360° cameras of wide FoV is an attractive option.

Recently, neural radiance fields (NeRF) [28] has brought significant progress in photorealistic novel view synthesis. NeRF is an implicit MLP-based neural network trained on calibrated multi-view images, which maps 5-D vectors (3-D coordinates and 2-D viewing direction) to opacity and color values of the 3-D coordinates viewed from that direction. Using the NeRF model, the image pixels are independently synthesized by accumulating opacity and color values along the camera ray in continuous 3-D space.

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.



Reference | GT (cropped) | NeRF [28] 5,000 iterations | NeRF [28] 100,000 iterations | NeRF + Ours 5,000 iterations | NeRF + Ours 100,000 iterations

Figure 1: Näively sampling rays from wide FoV images causes biased, insufficient training of the NeRF [28] model due to the distorted sphere-to-plane projection and large less-textured regions. Our non-uniform sampling strategies directly tackle this problem.

Although NeRF [28] and its extensions (e.g., [16, 35, 37]) have attracted significant attention, to the best of our knowledge, no attempt has been made to train NeRF on images from 360° cameras where information at each viewpoint is stored in a single 360° image format (e.g., Equirectangular projection (ERP)).

More concretely, the näive NeRF was implemented to uniformly sample rays from each pixel in all training images with equal probability. This is based on the assumption that rays passing through each pixel have equal coverage in 3D space and each pixel in the image has equal information. However, as projecting a spherical 360° information of a view to a planar image inevitably introduces projective distortion, the uniform sampling on image coordinates biases the distribution of camera rays in 3-D space. For instance, as shown in Fig. 1, high-latitude areas (*i.e.*, above and below the camera) in an ERP image occupy a much smaller 3-D space compared to low-latitude areas (front and side of the camera). Hence, a strategy is needed to take into consideration the projective distortion of sample rays for training NeRF model on 360° images. In addition, a wide FoV image is likely to contain large low-frequency textures (e.g., ceiling and floor, sky and ground) that need smaller samples than high-frequency textures; therefore, näively sampling rays from all the pixels is quite redundant.

The main contribution of this research is to raise these issues involved in applying NeRF to 360° images and to present the first effective ideas for addressing them. Concretely, we propose two non-uniform ray sampling strategies for efficiently training NeRF-based models on 360° images in the most standardized ERP format. First, we propose the *distortion-aware ray sampling*, which normalizes the sampling probability based on the per-pixel scaling factor of the area from the two-dimensional (2-D) plane to the sphere (i.e., the ratio of areas of the spherical surface to those of the corresponding ERP image). Intuitively, pixels at higher latitudes in ERP coordinates have smaller coverage in 3-D space; therefore, the lower sampling probability is allocated. Second, we propose the *content-aware ray sampling*, which adaptively updates the sampling probability based on pixel-wise reconstruction loss for each training step. Intuitively, the low-frequency textures such as sky and ground easily converge; therefore, we would lower the sampling probability of those regions as the training proceeds. Our method is surprisingly easy to implement and can therefore be easily incorporated into various NeRF-like models.

Our method is evaluated on both synthetic and real data. To evaluate the proposed method under ideal conditions, we create a synthetic evaluation benchmark of NeRF on 360 images using a physically-based renderer (*i.e.*, *Blender* [1]) with *Replica Dataset* [19] and *SceneCity* [3] Blender add-on. In our experiments, we will demonstrate that the proposed non-uniform sampling strategies are applicable to both the näive NeRF [28] and its variants such as DietNeRF [20], AugNeRF [11], and NeRF++ [43].

Since most consumer 360° cameras store information in an equirectangular projection (ERP) image, we assume that an image is represented in this format as in previous works (*e.g.*, [23, 41, 42]). However, it is straightforward to transfer our ideas to other spherical formats, such as cubic projection, which also causes distortions by sphere-to-plane projection.

2 Related Works

Novel View Synthesis: Before NeRF [28] emerged, discrete representations of scenes have been used in the novel view synthesis task. Several approaches with discrete representations of scenes employ point clouds [4], voxels [33], meshes [38], plane sweep volumes [14, 15], or multi-plane images [27, 34]. While effective, they have a disadvantage of limited resolution due to the large memory consumption.

Conversely, NeRF [28], a learning-based method based on continuous implicit representation, has achieved high-resolution rendering by taking advantage of volume rendering with continuous neural radiance fields. To improve the performance of NeRF, several studies have combined multiple representations (e.g., point clouds [43], voxels [26], and multi-plane images [21, 40]). In addition, NeRF opens up many new kinds of research using implicit neural representation [10, 24, 30, 36].

However, there have been few studies on novel view synthesis using multi-view 360° images [17, 22, 25]. To the best of our knowledge, this work is the first attempt to handle the projective distortion to apply NeRF for calibrated multi-view 360° images. Problems that arise when NeRF is applied to 360° images in ERP format include projective distortion, the presence of low-frequency texture regions resulting from wide FoV, and unbounded scenes. This study addresses two issues of projective distortion and the presence of regions of low-frequency texture.

The näive NeRF assumes that the entire scene can be packed into a bounded volume, so is problematic for unbounded scenes. NeRF++ [43] and mip-NeRF 360 [8] improved the performance of unbounded scenes in perspective images. NeRF++ [43] separates the scene into foreground and background and parameterizes the background by inverted sphere parameterization. Mip-NeRF 360 [8] proposes a parameterization to handle unbounded scenes under the conical frustum proposed in mip-NeRF [7], which is a NeRF variant that addresses sampling and aliasing. As a result, the mean squared error is reduced by 54% compared to mip-NeRF for unbounded scenes in perspective images. However, mip-NeRF 360 is not suitable for images with large projective distortion, such as ERP format. As described by the authors, this is because mip-NeRF assumes a small difference between the base and top radii of the frustum [7]. Our proposed method does not solve the problem caused by this intra-pixel distortion and therefore mip-NeRF 360 is not suitable. Our experiments show that NeRF++ combined with our method makes possible more effective novel view synthesis of 360° images for unbounded scenes.

We should note that some previous works controlled the sampling probability of 3D points along rays (*e.g.*, TermiNeRF [31] and NeRF-ID [5]), however, our proposal is different from them in that we control the *pixel-wise* sampling probability.

Spherical Novel View Synthesis: Spherical novel view synthesis is the task of synthesizing a novel 360° view from a set of multi-view 360° images. In this field, multi-sphere images (MSI)-based methods, which are spherical extensions of multi-plane images (MPI), are the most commonly used representation [6, 9, 18]. MSI-based methods have a fast rendering

time, but provide a discrete representation of the scene and have limited accuracy and large memory consumption to represent large scenes. In contrast, NeRF-based methods have their advantages in their accuracy and practical memory consumption with continuous neural representations even though the large training/test time is still a big challenge. So far, there have been very few attempts to apply NeRF-based models to 360° images. If any, OmniNeRF [17] handled more spherical information from a fisheye camera, and to the best of our knowledge, there was no attempt to explicitly tackle challenges that arise when NeRF-based models are trained from 360° information which is projected onto 2-D images.

Hard Example Mining: In the image understanding tasks such as image classification and object detection, hard example mining is a bootstrapping technique to solve the imbalance problem of training samples [12, 32]. In hard example mining, the sampling probability is non-uniformly assigned according to its current loss for enhancing the neural network to learn from harder and more important examples. Our content-aware ray sampling strategy introduced the idea of hard example mining in training NeRF model.

3 Preliminaries

Given calibrated multi-view images, NeRF [28] learns implicit 3-D volumes of opacity σ and color $\mathbf{c}$ of each 3-D coordinate by minimizing the pixel-wise discrepancy between the actual observation and the volume rendering result. The opacity σ is a function of 3-D position, independent of the viewing direction, and the color $\mathbf{c}$ is a function of both spatial position and viewing direction. NeRF trains neural networks for both opacity and color volumes based on the inverse volume rendering using regularly sampled 3-D points on rays $\mathbf{r}$ passing through pixels with a ray origin $\mathbf{o}$ and direction $\mathbf{d}$ in each input image as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ in the range $t_n \leq t \leq t_f$. Specifically, we partition $[t_n, t_f]$ into N evenly-spaced bins and then draw one sample $t_i (1 \leq i \leq N)$ uniformly at random from within each bin as

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right]. \tag{1}$$

Using all samples along a ray, the color $\hat{C}(\mathbf{r})$ at a corresponding pixel is computed using the volume rendering principle as

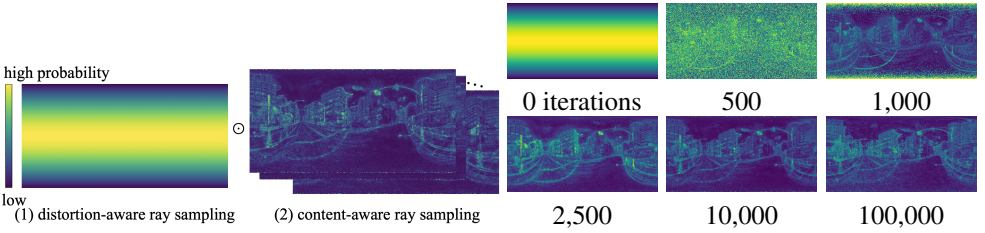
$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i, \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{2}$$

where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. NeRF has two levels of coarse and fine MLPs and learns for the loss function $\mathcal{L}$, that is, the squared error between the color $\hat{C}_c(\mathbf{r})$, $\hat{C}_f(\mathbf{r})$ synthesized by each ray $\mathbf{r} \in \mathcal{R}$ in each batch and the ground truth color $C(\mathbf{r})$.

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]. \tag{3}$$

4 Method

Basically, NeRF is trained on the batch-wise reconstruction loss where each batch contains multiple rays whose sampling probability is uniform at all the pixels in all the images. However, as has already been stated, when NeRF model is trained on 360° images in ERP for-



(a) two non-uniform sampling strategies (b) transition in sampling probability

Figure 2: Ray sampling of the proposed method. The proposed method is divided into two components: (1) distortion-aware ray sampling and (2) content-aware ray sampling.

mat, this uniform sampling strategy theoretically becomes problematic as the 3-D coverage of each ray passing through each pixel is *not* uniform due to the projective distortion. In addition, a wide FoV image basically contains large low-frequency texture regions such as sky and ground, ceiling and floor, and it is wasteful to keep spending the same amount of learning resources there as are spent on the high-textured areas. Therefore, we propose two *non-uniform* ray sampling strategies that individually control the pixel-wise sampling probability considering the geometric distortion (*i.e.*, distortion-aware ray sampling) and image content (*i.e.*, content-aware ray sampling).

Distortion-aware Ray Sampling: To reduce the sphere-to-plane projective distortion bias by uniform sampling of rays for the inverse volume rendering in NeRF [28], we control the sampling probability pixel-by-pixel considering the coverage of each pixel in 3-D space. Concretely, we compute the area S_d where a pixel of an ERP image occupies the unit spherical surface as

$$\begin{aligned} S_d &= \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} \cos\theta d\theta d\phi \\ &= (\phi_2 - \phi_1)(\sin\theta_2 - \sin\theta_1). \end{aligned} \tag{4}$$

Here $\theta \in [\theta_1, \theta_2]$ and $\phi \in [\phi_1, \phi_2]$ are latitude and longitude respectively, that bound each pixel in the ERP coordinates. Simply speaking, the higher latitude regions have smaller 3-D coverage and lower latitude regions have larger 3-D coverage. We calculate S_d for all pixels in all training images and normalize all of the values so that they add up to one. This result is used as the probability of sampling each pixel (P_d) during training as shown in Fig. 2-(a)-left. Intuitively, the higher sampling probabilities are assigned to lower latitude regions as each ray has to cover the larger space in 3-D.

Content-aware Ray Sampling: To avoid taking redundant samples from low-frequency texture (texture-less) regions in a 360° image with wide FoV, we further control the sampling probability so that the probability around the texture-less regions decreases. However, if no samples were assigned to the low-frequency region at all, learning would not proceed in that region; therefore, it is desirable to take samples from the entire image in the early stages of learning, then samples gradually be concentrated in more challenging regions. Based on this observation, we are inspired by the online hard example mining [12, 32], which is a bootstrapping technique that adaptively samples examples in a non-uniform way depending on the current loss of each example. We assume that the ℓ_2 reconstruction loss around the low-frequency texture regions decreases faster than that around the high-frequency texture

regions; hence, we decrease the sampling probability at pixels with smaller reconstruction loss at the *last* iteration. Concretely, we define S_c as a collection of pixel-wise inverse reconstruction loss at all pixels in all the training images that are uniformly initialized by one. At each iteration, a batch of rays passing through m pixels (*i.e.*, $m = 2048$ in our implementation) are stochastically sampled using the sampling probability at the current iteration and only S_c values at sampled pixels are updated based on the reconstruction loss. Then, the content-aware sampling probability (P_c) of each pixel is updated by normalizing S_c so that they add up to one (See Fig. 2-(a)-right). In Fig. 2-(b), we illustrate how the content-aware sampling probabilities are updated through iterations.

Sampling Strategy Details: We multiply and normalize the distortion-aware probabilities (P_d) and content-aware probabilities (P_c) so that they sum up to one. At each training iteration, we stochastically pick m pixels using this sampling probability, and a NeRF model is trained on the rays that pass through those pixels. We perturb the center of rays within the pixels to augment their coverage.

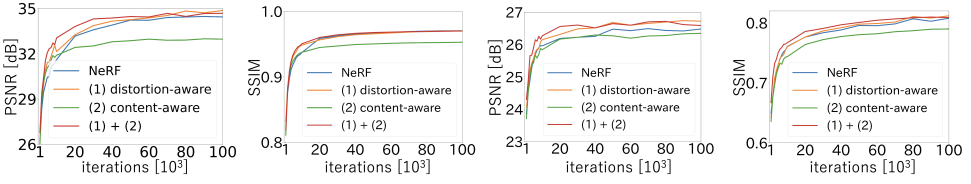
5 Results

To validate the effectiveness of our non-uniform ray sampling strategies, we implemented our method on NeRF [28] and its variants (*i.e.*, DietNeRF [20], AugNeRF [11], and NeRF++ [43]), and trained each model on both synthetic and real datasets. To exclude the effects other than different sampling strategies, only the minimal changes for the adaptation to ERP images were made (*i.e.*, a ray is defined on the spherical coordinates, rather than Cartesian coordinates). All the models were trained on a single NVIDIA Tesla A100 machine with 2048 samples per iteration. We used the Adam optimizer with default hyperparameters (*i.e.*, β_1 =0.9, β_2 =0.999, and $\varepsilon = 10^{-7}$) and a learning rate of 5×10^{-4} which was linearly decayed so that it became 5×10^{-5} at the 100k-th iteration. For each target scene, models were trained during 100k iterations taking approximately 7∼8 hours. In all scenes, the ray's near values were set to 0. In synthetic scenes, the ray's far values were set to reach the area excluding the sky. In real-world scenes, sufficient far value to reach the point clouds was set based on the OpenSfM camera position and point clouds.

5.1 Synth360 Dataset

It is known that the evaluation of NeRF-based models using real-world datasets is inevitably affected by camera calibration errors by structure-from-motion (SfM) tools, which may negatively affect the theoretical analysis [39]. Furthermore, a real 360° image often suffers from stitching errors which destroy the precise geometric consistency. Thus, we firstly evaluated our method on the ideal synthetic images. Since there was little synthetic dataset available for novel-view synthesis with multiple 360° images, we synthesized 360° images in the ERP format with Blender's Cycles renderer [1] using highly photorealistic 3D scenes, Replica Dataset [29], for indoor scenes and the city generator add-on for Blender, SceneCity [3], for outdoor scenes. Using these resources, we randomly picked 8 indoor scenes from Replica Dataset and generated 2 city scenes with SceneCity. For each indoor and outdoor scene, we rendered images by placing 5 to 9 cameras uniformly for training images and 100 cameras for test images whose resolutions are all 320×640. Examples of rendered images are illustrated in Fig. 3

Figure 3: Examples of synthesized scenes (*i.e.*, 6 of 10 scenes) in Synth360 dataset.



Indoor scenes (from Replica Dataset [19]) Outdoor scenes (from SceneCity [3])

Figure 4: Effect of each sampling strategy. The proposed method is divided into two parts: (1) distortion-aware ray sampling and (2) content-aware ray sampling. PSNR/SSIM were calculated for the images at each test pose.

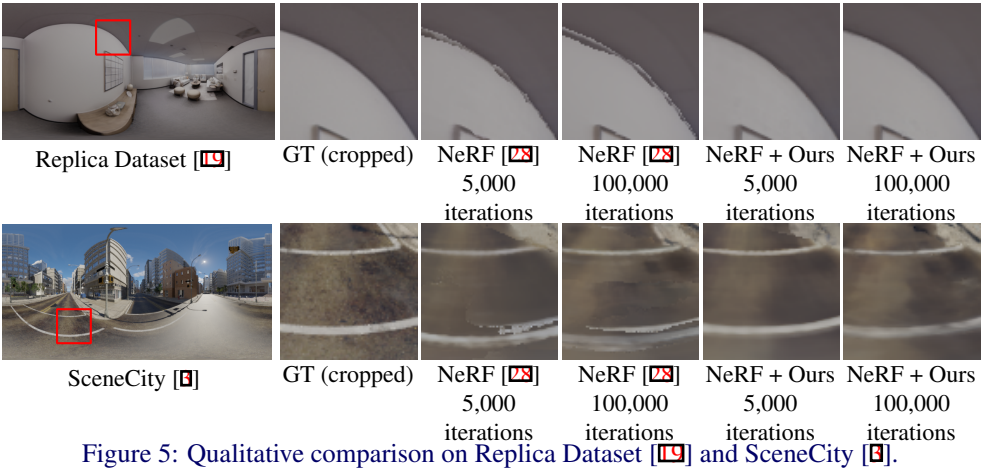


Figure 5: Qualitative comparison on Replica Dataset [19] and SceneCity [3].

5.2 Evaluation on Synth360

For evaluating the contribution of individual components, we firstly ablated each of our non-uniform sampling strategies implemented upon the näive NeRF model [28]. PSNR/SSIM curves during training are shown in Fig. 4. By comparing curves between the näive NeRF model and that with both distortion-aware and content-aware sampling strategies, we observe the obvious performance improvement by our method on both indoor (Replica) and outdoor (SceneCity) scenes; not only did our method improved the learning speed, but also PSNR/SSIM at convergence. Fig. 5 illustrates the qualitative comparisons at 5,000-th iter-

Table 1: Quantitative comparison using NeRF [28], DietNeRF [20], AugNeRF [11], and NeRF++ [43] on Replica Dataset [19] and SceneCity [3]. The best is highlighted.

	Replica Dataset [19]		SceneCity [3]	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NeRF [28]	34.44	0.970	26.48	0.808
NeRF [28]+Ours	34.68	0.970	26.59	0.809
DietNeRF [20]	36.34	0.974	31.86	0.889
DietNeRF [20]+Ours	**37.69**	**0.977**	**32.26**	**0.892**
AugNeRF [11]	36.80	0.982	27.53	0.826
AugNeRF [11]+Ours	37.07	0.983	28.61	0.844
NeRF++ [43]	12.39	0.386	26.62	0.799
NeRF++ [43]+Ours	14.39	0.521	28.33	0.832

ation and 100,000-th iteration, which show obvious advantages of our method over näive NeRF to recover details around the boundaries between walls and ceilings (Fig. 5-top) and around the white centerline (Fig. 5-bottom). Looking at the results in Fig. 4 in more details, we also observe that the content-aware sampling alone rather degraded the performance. We further analyzed this failure case and found that the model trained based solely on the content-aware sampling without the distortion-aware sampling had been overfitted to high-frequency details at high-latitude regions which were severely distorted due to the projective distortion. This result suggests that a proper consideration of the projective distortion is critical for NeRF on 360° images. In the supplementary material, we provide a deeper analysis of how our distortion-aware sampling contributed to different latitude regions and how our content-aware sampling contributed to a different amount of textures.

To validate that the benefit of our non-uniform sampling strategies is not limited to the näive NeRF model, we applied our method to other NeRF-like models such as DietNeRF [20], AugNeRF [11], and NeRF++ [43] with only changes about sampling strategies.[1] A quantitative comparison of PSNR/SSIM at 100,000-th iteration is shown in Table 1. We observe that our non-uniform sampling strategy consistently improved the performance of DietNeRF, AugNeRF, and NeRF++. Please note that the reconstruction accuracy of NeRF++ is much lower than others because NeRF++ failed to properly decompose the foreground and background on our Synth360 dataset. In the supplementary, we also visualize PSNR/SSIM curves and rendered images for each method for further discussion.

5.3 Evaluation on Real 360° Images

We also evaluated our non-uniform sampling strategies on two real scenes; one is indoor and the other is outdoor. We used a consumer 360° camera to capture each scene in the ERP format and calibrated extrinsic parameters using OpenSfM [2]. A quantitative comparison of PSNR/SSIM at 100,000-th iteration among ours (with both strategies), NeRF [28], DietNeRF [20], AugNeRF [11] and NeRF++ [43] on these two scenes is shown in Table 2. While PSNR/SSIM scores of real images are lower than those of synthetic ones as expected, our proposed method also consistently improved the reconstruction quality. We illustrate the qualitative comparison at 5,000-th iteration and 100,000-th iteration in Fig. 6.[2] It is interesting to see that our sampling strategies did not significantly improve both PSNR and SSIM

[1] More implementation details are presented in the supplementary.

[2] Due to the space limit, we only show the rendered images of NeRF and NeRF + Ours. Please refer to the supplementary for other results.

Table 2: Quantitative comparison using NeRF [28], DietNeRF [20], AugNeRF [1], and NeRF++ [43] on our real-world indoor and outdoor scenes. The best is highlighted.

	indoor scene		outdoor scene	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NeRF [28]	22.84	0.760	24.14	0.723
NeRF [28]+Ours	23.28	**0.805**	24.16	0.739
DietNeRF [20]	21.50	0.771	23.58	0.733
DietNeRF [20]+Ours	22.89	0.794	23.84	0.735
AugNeRF [1]	18.71	0.677	22.44	0.682
AugNeRF [1]+Ours	20.60	0.686	22.54	0.724
NeRF++ [43]	22.20	0.769	23.90	0.786
NeRF++ [43]+Ours	**23.37**	0.801	**24.32**	**0.803**

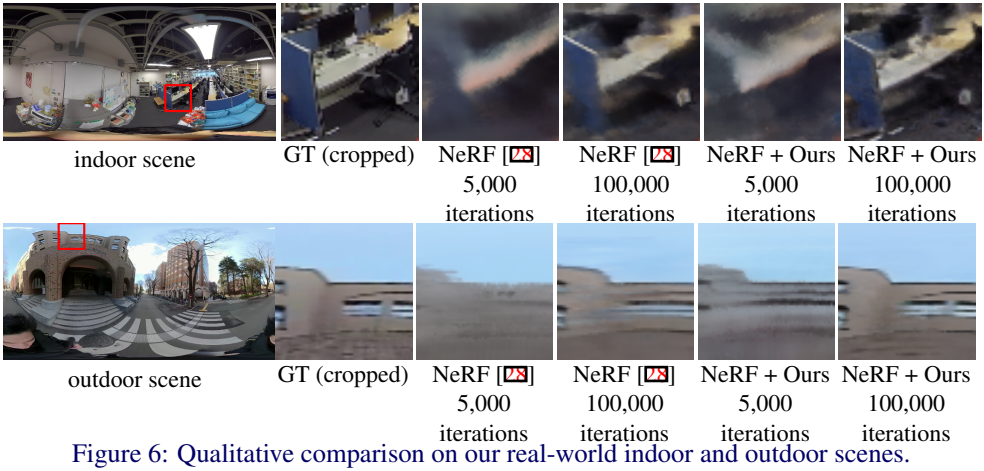


Figure 6: Qualitative comparison on our real-world indoor and outdoor scenes.

scores from the näive NeRF, however, we can observe the clear advantages of our method on the visual comparison. In 360° images with a wide field of view and many flat areas, non-uniform sampling seemed to contribute to the visual quality of high-frequency regions more than PSNR/SSIM scores.

6 Conclusion

In this work, we proposed two non-uniform ray sampling strategies for effectively training NeRF-based models from 360° images specifically in the ERP format: distortion-aware ray sampling and content-aware ray sampling. Based on our Synth360 dataset which rendered synthetic indoor and outdoor scenes, we showed that the proposed method consistently improved the training curves of näive NeRF and its variants. Because of its simplicity, the proposed method is highly compatible with other efficient methods (*e.g.*, [29]), and we believe that the most important contribution of this research is that it shows the importance of non-uniform sampling of rays in distorted image representations, such as 360° images. For future work, we should integrate our non-uniform sampling strategies into more diverse variants of NeRF model (*e.g.*, [10, 24, 30, 36]).

Acknowledgement

This work is partially supported by JSPS KAKENHI 21H03460 and JST-Mirai Program JPMJMI21H1.

References

[1] Blender. https://www.blender.org/.

[2] OpenSfM. https://github.com/mapillary/OpenSfM.

[3] SceneCity. https://www.cgchan.com/store/scenecity.

[4] K. A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky. Neural point-based graphics. *ECCV*, 2020.

[5] Relja Arandjelović and Andrew Zisserman. Nerf in detail: Learning to sample for view synthesis. 2021.

[6] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *ECCV*, 2020.

[7] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.

[8] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

[9] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. 39(4):86:1–86:15, 2020.

[10] R. M. Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[11] Tianlong Chen, Peihao Wang, Zhiwen Fan, and Zhangyang Wang. Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations. In *CVPR*, 2022.

[12] J. Chu, Z. Guo, and L.Leng. Object detection based on multi-layer convolution feature fusion and online hard example mining. *IEEE access*, 6:19959–19967, 2018.

[13] K. Deng, A. Liu, J. Y. Zhu, and D. Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *CVPR*, 2022.

[14] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016.

[15] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019.

[16] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[17] Kai Gu, Thomas Maugey, Sebastian Knorr, and Christine Guillemot. Omni-nerf: neural radiance field from 360° image captures. *ICME*, 2022.

[18] Tewodros Habtegebrial, Christiano Gava, Marcel Rogge, Didier Stricker, and Varun Jampani. Somsi: Spherical novel view synthesis with soft occlusion multi-sphere images. In *CVPR*, 2022.

[19] T. Whelan J. Straub, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv:1906.05797*, 2019.

[20] A. Jain, M. Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021.

[21] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, 2021.

[22] J. Li, Y. He, Y. Hu, Y. Han, and J. Wen. Learning to compose 6-dof omnidirectional videos using multi-sphere images. In *ICIP*, 2021.

[23] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *CVPR*, 2022.

[24] Z. Li, S. Niklaus, N. Snavely, and O. Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[25] K-E. Lin, Z. Xu, B. Mildenhall, P. P. Srinivasan, H. Geoffroy, S. DiVerdi, Q. Sun, K. Sunkavalli, and R. Ramamoorthi. Deep multi depth panoramas for view synthesis. In *ECCV*, 2020.

[26] L. Liu, J. Gu, K. Z. Lin, T. S. Chua, and C. Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020.

[27] B. Mildenhall, P. P. Srinivasan, R. O. Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[28] 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.

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4): 102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

[30] M. Niemeyer and A. Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021.

[31] Martin Piala and Ronald Clark. Terminerf: Ray termination prediction for efficient neural rendering. *3DV*, 2021.

[32] A. Shrivastava, A. Gupta, and R. B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

[33] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, 2019.

[34] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, 2019.

[35] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021.

[36] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021.

[37] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. *CVPR*, 2022.

[38] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *ECCV*, 2014.

[39] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv:2102.07064*, 2021.

[40] S. Wizadwongsa, P. Phongthawee, J.Yenphraphai, and S. Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021.

[41] Youngho Yoon, Inchul Chung, Lin Wang, and Kuk-Jin Yoon. Spheresr: 360° image super-resolution with arbitrary projection via continuous spherical image representation. In *CVPR*, 2022.

[42] Jiaming Zhang, Kailun Yang, Chaoxiang Ma, Simon Reiß, Kunyu Peng, and Rainer Stiefelhagen. Bending reality: Distortion-aware transformers for adapting to panoramic semantic segmentation. In *CVPR*, 2022.

[43] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020.