Important Pixels Sampling for NeRF Training Based on Edge Values and Squared Errors Between the Ground Truth and the Estimated Colors

Kohei Fukuda, Takio Kurita and Hiroaki Aizawa
Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima, Japan

Keywords: Neural Radiance Fields, Novel View Synthesis.

Abstract: Neural Radiance Fields (NeRF) has impacted computer graphics and computer vision by enabling fine 3D representations using neural networks. However, depending on the data (especially on synthetic datasets with single-color backgrounds), the neural network training of NeRF is often unstable, and the rendering results become poor. This paper proposes a method to sample the informative pixels to remedy these shortcomings. The sampling method consists of two phases. In the early stage of learning (up to 1/10 of all iterations), the sampling probability is determined based on the edge strength obtained by edge detection. Also, we use the squared errors between the ground truth and the estimated color of the pixels for sampling. The introduction of these tweaks improves the learning of NeRF. In the experiment, we confirmed the effectiveness of the method. In particular, for small amounts of data, the training process of the neural network for NeRF was accelerated and stabilized.

1 INTRODUCTION

In the fields of computer graphics and computer vision, learning three-dimensional (3D) representations from 2D images has been a longstanding problem. To model a 3D structure of a scene and an object, various methods, such as point clouds, voxels, and implicit functions, have been proposed. A point cloud represents the surface and interior of the object or scene as a set of points. While it can represent arbitrary shapes, it is necessary to increase the number of points for complex shapes, leading to high storage requirements. On the other hand, voxels represent the object's surface and internal structure on regular grids in 3D space. Similar to point clouds, the drawback is high memory cost when high resolution is needed for complex shapes. Implicit function representation uses a mathematical equation to represent shapes. Unlike explicit representations such as point clouds and voxels, it can represent complex shapes with a fixed number of parameters, allowing for more efficient storage.

In the context of 3D modeling, neural fields are a type of neural network to represent a complex 3D structure of shapes or scenes. Instead of explicitly representing the 3D structure using point clouds and voxels, the neural fields take 3D coordinates as input and output 3D properties like color or occupancy.

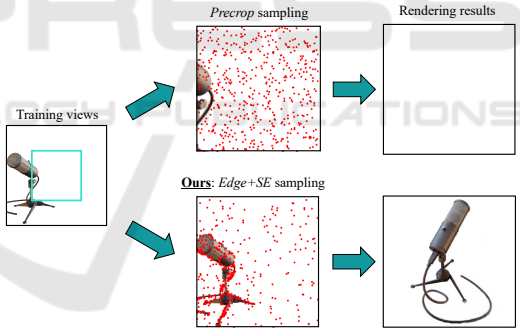

Figure 1: Our problem setting and rendering results.

The advantage of such an approach is to provide compact and learnable representations from given data. It also facilitates the manipulation of the 3D structure, such as interpolation, rendering, and transformation, through neural network inference.

Among them, Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) leverage the capability of the neural fields and a differentiable rendering technique to model the 3D structure and render novel views. NeRF predicts the colors of pixels sampled from known camera pose images using fully connected neural networks and volume rendering techniques, and solves a regression problem that approaches the ground truth colors, allowing the generation of fine-grained novel viewpoint

Fukuda, K., Kurita, T. and Aizawa, H.
Important Pixels Sampling for NeRF Training Based on Edge Values and Squared Errors Between the Ground Truth and the Estimated Colors.
DOI: 10.5220/0012346200003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 2: VISAPP*, pages 102-111
ISBN: 978-989-758-679-8; ISSN: 2184-4321

images. Thus, NeRF has made a great impact, and various follow-up works have been proposed (Yu et al., 2021; Jang and Agapito, 2021; Niemeyer et al., 2022; Wang et al., 2021; Lin et al., 2021; Meng et al., 2021; Metzer et al., 2023; Poole et al., 2022; Srinivasan et al., 2021; Yang et al., 2022; Barron et al., 2022; Deng et al., 2022; Oechsle et al., 2021; Park et al., 2021b; Du et al., 2021; Pumarola et al., 2021; Park et al., 2021a; Martin-Brualla et al., 2021; Ueda et al., 2022; Fukuda et al., 2023). Although many improvement methods have been proposed, the issue of training stability remains an open problem. Specifically, it is known that training NeRF on scenes with a single background color can lead to instability or a lack of convergence in the training process. We assume that the reason for the issue is due to the sampling of pixels to be trained. In the training phase, we randomly sample the pixels to be trained from the image in each iteration; hence, NeRF's model overfits the background color by sampling a large number of background pixels, resulting in the failure to render the object part. To solve this problem, the original NeRF employs a temporary technique called *Precrop*, which samples pixels from the center of the image only in the initial stage of training to be able to sample pixels in the object part. *Precrop* requires an object in the center of the training image. However, images in which objects are small or not centered cannot benefit from *Precrop*. In the first case, you have to set a hyperparameter to determine what ratio of the image to *Precrop*, and *Precrop* does not address the second case.

Therefore, we propose pixels sampling method that can stably train NeRF for any image (whether the object is in the center or not). The proposed method consists of a two-stage sampling technique. The first is to introduce pixel sampling based on edge detection as an alternative to *precrop* for stability in the early stages of training. The edge detection is performed on the training images as prior knowledge, and each pixel is weighted and sampled using the edge detection. Second, we introduce pixel sampling based on the squared error between the ground truth and the predicted color to make the subsequent training more stable. Pixels that are twice the size of the pixel to be trained are sampled randomly, weighted based on the squared error of each pixel, and sampled.

Our contributions are summarized as follows:

- We propose pixel sampling methods for NeRF to be trained with higher stability than the original NeRF.
- Our proposed method succeeded in eliminating the hyperparameters that were present in *precrop*.

2 RELATED WORK

2.1 Neural Radiance Fields

Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) is the method of 3D structural representation. NeRF can learn the 3D structure of the objects or the scene and render novel-view images by using multi-view images with camera poses. In this method, by approximating fully connected neural networks to radiance fields, which are vector fields each coordinate is given a color and a volume density, and 3D structural representation is made possible.

In this training flow, one image is sampled from the training images randomly, some pixels in this image are sampled randomly, their mean squared error (MSE) is calculated, and the parameters of NeRF's networks are optimized. However, one problem arises when learning images with a single-color background (Blender). This problem is that the training loss does not converge and rendering fails. This training instability is caused by network dependence and data dependence. For network dependencies, (Mildenhall et al., 2020) introduces the network structure to positional encoding, and (Barron et al., 2022) uses softplus function (Glorot et al., 2011). However, few methods have been proposed to improve data dependent problems. In the usual random sampling, the NeRF's model overfits the background color. This phenomenon is especially noticeable when there is only a small amount of training data. In contrast, the original NeRF employs *Precrop* as a temporary technique, which crops only the center of the image and samples pixels from there.

2.2 Edge Detection

Edge detection is a type of image-processing technique that finds the edges of objects where the brightness changes significantly in the image. By using edge detection, we can understand the contours and shapes of the object in the image.

There are some methods of edge detection. The Sobel method (Sobel et al., 1968) uses a gradient detection filter with first-order differentiation to detect edges. The Canny method (Canny, 1986) uses thresholds to detect edges. The Laplacian method uses a gradient detection filter with second-order differentiation to detect edges. These edge detection methods are used in many previous works related to images (Singh et al., 2016; Ali and Clausi, 2001).

Edges in an image give rich information about the shape and structure of an object. In practice, the edge information is used for face emotion recognition

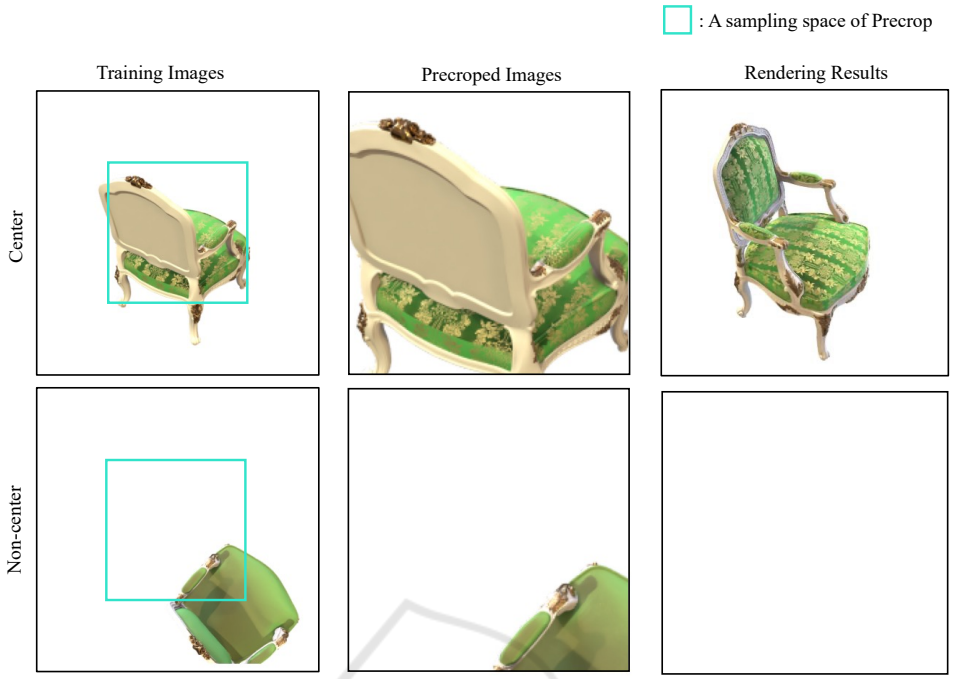

Figure 2: The comparisons between training center and non centered images. This figure shows the results of training with NeRF for images with and without an object in the center, respectively. In the upper, an object is centered in the left image. In this image, the *Precrop* range captures the object and the rendering result is clear (bottom). In contrast, an object is not centered in the right image. In this, the *Precrop* space does not capture the object and the rendering result overfits this white background.

tasks (Zhang et al., 2019), face detection tasks (Singh et al., 2016), and remote sensing images processing (Ali and Clausi, 2001). For NeRF, (Gai et al., 2023) was improved by using canny edge detection. We guess that edge information is effective for NeRF's training because the base of objects or scenes can be represented more accurately. Therefore, we propose a sampling method using this edge information for NeRF's training.

2.3 Data Sampling

Machine learning can create a model to predict accurately by using large amounts of training data and optimizing this parameter. However, there are many kinds of training data (e.g., data that is valid for learning and data that is not, Data with and without noise). In previous studies, a number of methods have been proposed to train machine learning models by taking into account the features of the data. Dataset distillation methods (Nguyen et al., 2020; Nguyen et al., 2021) reduce computational cost by distilling large amounts of datasets into smaller synthetic datasets. Curriculum learning methods (Soviany et al., 2021) are techniques that mimic the human learning process, learning difficult data from easy-to-learn data. For a large dataset, statistical sampling methods are often used. Sampling is a method to extract a subset from a large training dataset, namely random sampling, cluster sampling, and hierarchical sampling.

In NeRF, random sampling is used to select pixels to be trained from images. However, as mentioned in Sec. 2.1, random sampling results in unstable learning depending on the data. Therefore, we propose a new sampling method.

3 PROPOSED APPROACH

3.1 Problem Setting

The goal of this study is to improve the stability of the training of NeRF. While NeRF can represent novel 3D fields, the training may not converge, especially for images with a single-color background (Blender). We assume that the training is affected by the background in the early stages of training when using training data with a single-color background, therefore, the error does not converge, resulting in failure to learn a 3D structural representation of the object. That's why we think that if the important pixels to the learning process are sampled at the beginning of the training,

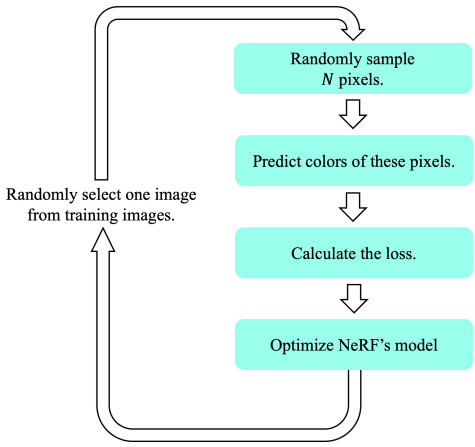

Figure 3: Training Flow of NeRF.

NeRF's training could go well.

As mentioned in Sec. 2.1, the original NeRF employs *Precrop* as a temporary technique. However, this technique is effective only when the rendering target occupies most of the image and is centered. If the object is centered but too small, the cropping rate must be manually changed. If the object is not in the center, it is overfitted with the background color. The actual rendering result is shown in Fig 2.

We propose a simple yet effective pixels sampling method that can stably train NeRF's model for any images (whether object-centered or not).

3.2 The Training Flow of NeRF

NeRF is a 3D representation method of objects or scenes in 2D training images by training multi-view images with camera poses and optimizing the parameters of a fully connected network. The training flow per one iteration of the original NeRF consists of (1) data sampling and pixel sampling, (2) rendering the color in the image, and (3) calculation of the loss value and update of the network parameters, as shown in Fig 3.

Data Sampling and Pixel Sampling. In this step, we decide the pixels to predict colors and calculate loss values. Firstly, we select one image randomly from the training dataset. Next, we sample pixels for NeRF's training from the given image. In the original NeRF, the initial stage of training uses the *Precrop* technique, which samples pixels in the center of the image, to avoid the biased sampling of the background color. After that, pixels are randomly sampled from the whole image (*Random sampling*). As mentioned in Sec. 2.1, *Precrop* and *Random sampling* remain the issue for training stability in the case where a single-color background dominates a large area of the image. Therefore, we propose a novel pixel sampling strategy that enables stable learning even in such cases.

Rendering the Color in the Image: In the next step, by shooting rays to these pixels from the camera position and inputting the coordinates of multiple points on the rays and the ray direction vectors to the neural network, we obtain the color of each point and the volume density, which indicates the probability that the ray hits an object as outputs. Then, we predict the colors of the N pixels with these outputs of the network and volume rendering techniques.

Optimizing the NeRF's Model: The loss values are calculated between these predicted and the ground truth colors to update the network's parameters. The loss function of the original NeRF is described as follows:

$$\mathcal{L}_{mse} = \frac{1}{N}\sum_{n=1}^{N} \|\hat{C}_n - C_n\|^2, \tag{1}$$

where $\hat{C}$ is the predicted color and C is the ground truth color. Finally, we perform error backpropagation and update the network parameters.

3.3 The Sampling Method Based on Edge Detection

In this section, we explain a new pixels sampling method based on edge detection instead of *Precrop* method. This method achieves stability in learning NeRF by simply detecting edges from training images and using this information for sampling in the first phase.

For all training images, we detect edges by using edge filters (e.g., Laplacian, Sobel). These calculated values represent the intensity of edges, with pixels with higher values indicating stronger edges, and conversely, pixels with lower values indicating weaker edges. By using these values, we calculate the sampling probability of each pixel and perform weighted sampling based on this probability.

We consider a pixel $I_{k,i,j}$ in a training image I_k where $0 \leq i < H$, $0 \leq j < W$, H is the height and W is width of the image I_k. In the original random sampling method, the probability that $I_{k,i,j}$ is chosen is as follows:

$$P_{\text{random}}[I_{k,i,j}] = \frac{1}{H \times W}, \tag{2}$$

where $H \times W$ means the number of pixels in one image. *Precrop* sampling, the pixels sampling method from images cropped out the central part of the full images, has been used in the original NeRF's implementation. Then, the probability that $I_{k,i,j}$ is sampled

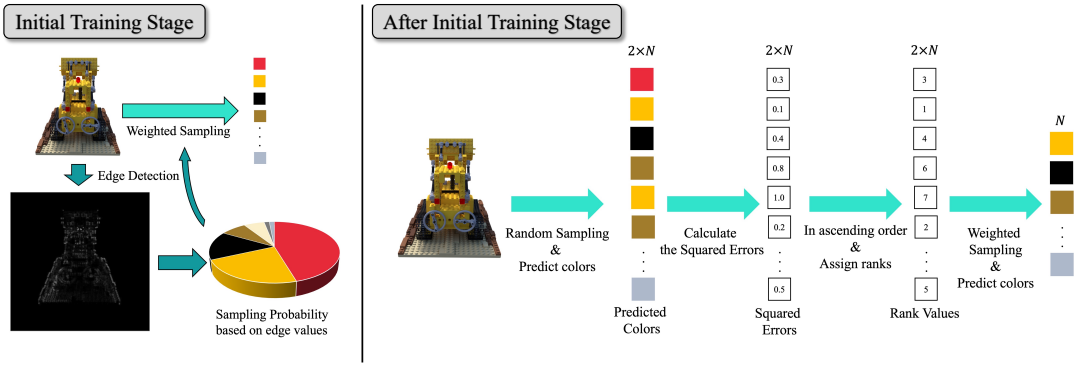

Figure 4: An overview of our proposed method. This figure shows our pixels sampling method. In the initial training stage (before about 1/10 of all epochs), we use our *Edge sampling* method which calculates the sampling probability of each pixel from the values obtained by edge detection of the original images and performs weighted sampling based on it. After the initial training stage, we use our *SE sampling* method which randomly samples twice the number of pixels used for training, computes their squared errors, and performs weighted sampling based on their ranking.

is as follows:

$$P_{\text{precrop}}[I_{k,i,j}] = \frac{4}{H \times W}, \tag{3}$$

where $\frac{H}{4} \leq i < \frac{3H}{4}$ and $\frac{W}{4} \leq i < \frac{3W}{4}$. However, there is not always an object in the center of the image, in which case the *Precrop* sampling method will have no effect. Then, our new sampling method is effective.

Firstly, edge detection is performed for I_k as follows:

$$E_k = \phi(G(I_k)), \tag{4}$$

where $\phi(\cdot)$ is the edge detection and $G(\cdot)$ is the gray scale transformation function.

By using Eq (4), we define the sampling probability of pixels in the image as follows:

$$P_{\text{edge}}[I_{k,i,j}] = \frac{|E_{k,i,j}| + \varepsilon}{\sum_{h}^{H-1} \sum_{w}^{W-1} (|E_{k,h,w}| + \varepsilon)}, \tag{5}$$

where $\varepsilon > 0$. In Eq (6), we use absolute values of E because E includes negative numbers. Also, in order to target non-edge areas to sampling areas, ε is added to the numerator and the denominator and defined so that the sampling probabilities of all pixels are greater than 0. Based on this probability P_{edge}, we perform weighted-pixel sampling at the beginning of training instead of *Precrop* sampling. The effect of this method is to enable sampling pixels, which is important for NeRF training, and to improve stability in the early stages of training.

3.4 The Sampling Method Based on Squared Errors

We propose a method to stabilize not only the learning stability in the early stages of training but also the training thereafter. After the first phase of training, we use a new pixels sampling method based on the squared error between the predicted and the ground truth colors instead of *Random sampling* in Sec. 3.2.

Firstly, we sample $2 \times N$ pixels randomly from a training image when the number of pixels, that are used for training, is N. Next, the colors of these pixels are predicted, and each squared error value is calculated based on Eq (1). If a pixel in an image ($I_{k,l}$) is sampled at random, let $\text{SE}_{k,l}$ be this squared error, where $0 \leq l < 2 \times N$. Then, we sort these values in ascending order and assign a rank to each pixel as follows:

$$Rank_{k,n} = \underset{0 \leq n < 2 \times N}{\text{argsort}}\ \text{SE}_{k,n}. \tag{6}$$

Based on these rank values, the sampling probability is determined, and weighted sampling of N pixels is performed from $2 \times N$ pixels as follows:

$$P_{\text{SE}}[I_{k,l}] = \frac{Rank_{k,l}}{\sum_{n}^{2 \times N} (Rank_{k,n})}. \tag{7}$$

4 EXPERIMENT

To evaluate the effectiveness of our proposed method, experiments comparing the original random sampling method with our proposed sampling method were conducted with the synthetic dataset. In the following sections, we explain the evaluation metrics, the dataset, and the experimental results.

4.1 Experimental Settings

4.1.1 Datasets

In these experiments, we used three datasets. The first dataset is the synthetic dataset created in Blender (Chair, Drums, Ficus, Hotdog, Lego, Materials, and Mic), used in the original NeRF paper (Mildenhall et al., 2020). This dataset is designed so that the object is the center of the image. We call it centered Blender. The second is the dataset created based on the first, in which the object is not centered in the image. We call it non-centered Blender.

4.1.2 Evaluation Metrics

For quantitative evaluation, we used Peak Signal to Noise Ratio (PSNR). PSNR can reflect the ratio of the maximum possible power of a signal to the power of corrupting noise that affects the fidelity of its representation. The larger value of PSNR represents less image degradation.

4.1.3 Baseline Methods

In this paper, we introduce our sampling method to the original NeRF (Mildenhall et al., 2020) and InfoNeRF (Kim et al., 2022). InfoNeRF is a method for mitigating inconsistencies in reconstruction and overfitting to the training images taken from a few viewpoints. These two introduction experiments to NeRF show whether the effect depends on the number of data.

4.1.4 Sampling Methods

We evaluated the following pixels sampling methods. Since we changed only the pixels sampling method in the experiments, the main algorithms of NeRF and InfoNeRF were not changed. In the following experiments, we defined $1/10$ of all iterations as the initial training stage from our experience.

Random Sampling: we sample the training pixels from all pixels of the image in all iterations as described in Eq (2).

Precrop + Random Sampling: we sample the training pixels from pixels in the center of the images up to $1/10$ of the total number of iterations and use *Random* method

SE Sampling: we sample the training pixels from all pixels randomly up to $1/10$ of the total number of iterations and use our sampling method based on the squared errors

Precrop + SE Sampling: we use *Precrop* method up to $1/10$ and use *SE sampling*

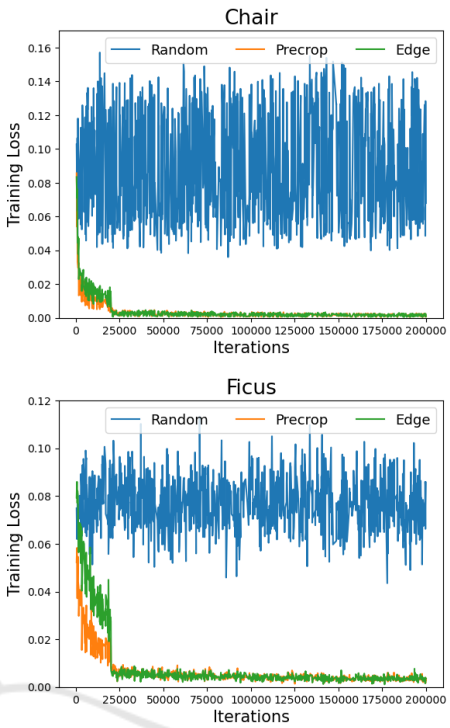

Figure 5: Training loss comparisons. These figures show training loss comparisons of *Random*, *Precrop*, and *Edge sampling*. In Chair (left) and Ficus (right), *Random*'s training loss (blue) has not dropped. On the other hand, *Precrop*'s (orange) and *Edge sampling*'s (green) have done. These results indicate that the sampling method in the initial training stage has a significant impact on the training of NeRF.

Edge Sampling: we use our sampling method based on edge detection up to $1/10$ and use *Random* method. Although the Canny edge detector is used in (Gai et al., 2023), we didn't use it because it has two threshold parameters. Also, since the PSNR of using Laplacian filter and Sobel filter for edge detection in Lego showed 31.76 and 31.97, respectively, Sobel was used for edge sampling in this paper.

Edge sampling + SE sampling: we use *Edge sampling* method up to $1/10$ and use *SE sampling* method.

4.2 Results on the Original NeRF

4.2.1 Centered Blender Dataset

This experiment is the original NeRF trained by 7 blender data in which the object is centered. Table 1 shows these results. *Random* results show training instability depending on the data (especially Chair, Ficus, and Mic). The reason for this is that the image has a large background area, which causes overfitting

Table 1: PSNR of the original NeRF using Centered Blender Dataset.

Sampling Methods	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Avg.
Random	13.69	23.94	14.26	36.63	31.97	29.82	13.12	23.34
Precrop+Random	33.42	26.36	29.36	36.87	31.85	29.65	32.88	31.48
SE sampling	13.69	24.07	14.26	37.09	32.26	**29.99**	13.12	23.49
Precrop+SE sampling	33.73	**26.62**	29.61	**37.36**	**32.27**	29.89	**33.33**	**31.83**
Edge sampling	33.47	26.38	29.35	36.7	31.97	29.71	32.72	31.47
Edge sampling+SE sampling	**33.9**	26.55	**29.7**	37.14	32.18	29.84	33.19	31.78

to the background area in the early stages of training. For this problem, the original NeRF introduces *Precrop*, and the results can be seen to be improved in *Precrop+Random* results. *Edge sampling* results show that our proposed *Edge sampling* method effect achieves the same training stability as *Precrop* and there is no significant difference in the effectiveness between *Precrop* and *Edge sampling* for these datasets. Comparing *Random* and *SE sampling*, *Precrop+Random* and *Precrop+SE sampling*, *Edge sampling* and *Edge sampling+SE sampling*, the value of PSNR increases regardless of the sampling method used in the initial training stage. These results indicate that *SE sampling* has the effect of accelerating learning.

4.2.2 Non-Centered Blender Dataset

This experiment is the original NeRF trained by 7 Blender datasets in which the object is not centered. Table 2 shows these results. In Chair, *Precrop* stabilized the training when the object was at the center of the image (Table 1). However, the training was not successful in this case. From this result, *Precrop* may also have unstable learning depending on the data. On the other hand, when *Edge sampling* is used, the training is stable regardless of the data.

4.3 Results on Few-Shot NeRF

4.3.1 Centered Blender Dataset

This experiment is InfoNeRF (Kim et al., 2022) by 7 blender data in which the object is centered. This result is shown in Table 3. Since only 10 training images are used in InfoNeRF, the performance of each sampling method is different from Table 1. Comparing *Precrop* and *Edge sampling*, the PSNR values in Table 1 are almost the same, while those of *Edge sampling* are better in Table 3. This indicates that the edge information is more important for training because the number of training images is smaller. In addition, *SE sampling* accelerates learning in all combinations, which is the same as the result of the original NeRF.

4.3.2 Non-Centered Blender Dataset

This experiment is InfoNeRF by 7 blender data in which the object is not centered. This result is shown in Table 4. The results of Chair and Ficus show that when the object is in the center of the image, the training is successful by using *Precrop* in Table 3, but when the object is not in the center, it is not successful. However, the use of *Edge sampling* solves this problem. We also find that *SE sampling* accelerates the training process more than the use of ordinary *Random* sampling.

5 DISCUSSIONS

From the experiments in Section 4, we found that *Edge sampling* stabilizes the initial stage of training, independent of the data. In this section, we examine the effect of *Edge sampling* on the part of the image.

Fig 6 shows where *Edge sampling* shows its effects for Chairs and Drums. This figure compares *Precrop* and *Edge sampling* rendering accuracy by using the images cropped from the full-image images with many edges. In Chair, the mean squared error between the ground truth and predicted colors (MSE) of *Precrop* is 0.00096. On the other hand, the MSE of *Edge sampling* is 0.00093. In the case of Drums, the MSE of *Precrop* is 0.005306. On the other hand, the MSE of *Edge sampling* is 0.00413. Also, the red circles in Fig 6 show that the rendering results of *Edge sampling* are cleaner than those of *Precrop*. Based on these results, we think that *Edge sampling* is effective not only in improving the stability of training but also in improving the rendering accuracy of the edges. In addition to this, the edge parts in the images are very important for NeRF's rendering.

There are two limitations of our method. The first is that, as with *Precrop*, we have to set a hyperparameter to determine how far to perform *Edge sampling*. The second is that *SE sampling* requires more memory than normal *Random* sampling.

Table 2: PSNR of the original NeRF using Non-centered Blender Dataset.

Sampling Methods	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Avg.
Random	13.69	11.12	14.26	29.4	25.6	22.82	13.12	18.57
Precrop+Random	13.69	18.46	28.71	29.56	25.61	23.13	13.12	21.75
SE sampling	13.69	11.12	14.26	29.07	25.73	22.76	13.12	18.53
Precrop+SE sampling	13.69	18.55	28.97	29.25	25.64	**23.21**	13.12	21.77
Edge sampling	**27.13**	**19.38**	28.88	**29.72**	29.56	23.02	**29.38**	**26.72**
Edge sampling+SE sampling	27.12	19.15	**29.27**	29.07	**29.89**	22.82	29.25	26.65

Table 3: PSNR of Few-shot NeRF using Centered Blender Dataset.

Sampling Methods	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Avg.
Random	14.03	10.93	14.22	26.63	25.14	22.14	13.04	18.01
Precrop+Random	25.45	19.19	22.87	26.4	24.91	22.07	26.91	23.97
SE sampling	14.03	19.13	14.22	**27.71**	25.37	**22.28**	13.04	19.39
Precrop+SE sampling	25.68	18.89	22.86	27.22	24.89	22.05	27.12	24.10
Edge sampling	25.52	**19.94**	**22.89**	26.71	25.77	22.03	27	24.26
Edge sampling+SE sampling	**25.75**	19.83	**22.89**	27.55	**26.03**	22.05	**27.38**	**24.49**

Table 4: PSNR of Few-shot NeRF using Non-centered Blender Dataset.

Sampling Methods	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Avg.
Random	14.03	**17.5**	14.22	22.2	**21.84**	8.73	13.04	15.93
Precrop+Random	14.03	16.71	14.22	21.76	20.99	18.14	20.02	17.98
SE sampling	14.03	17.44	14.22	22.25	21.82	8.73	13.04	15.93
Precrop+SE sampling	14.03	16.92	14.22	21.97	21.71	18.2	20.08	18.16
Edge sampling	22.38	**17.5**	**21.87**	22.13	21.3	18.21	20.69	20.58
Edge sampling+SE sampling	**22.43**	17.4	21.8	**22.31**	21.42	**18.44**	**20.73**	**20.64**

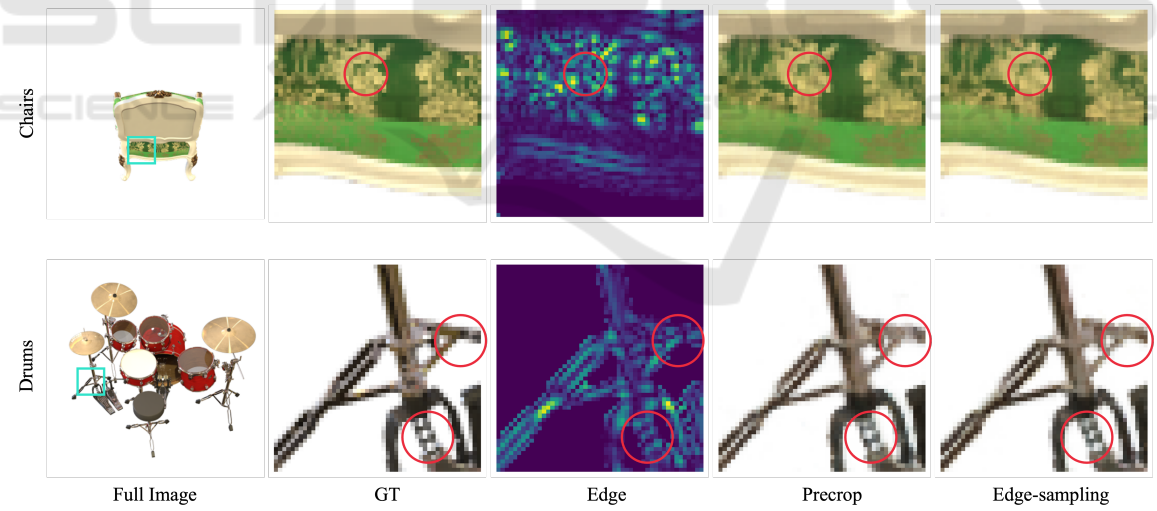

Figure 6: The effects of *Edge sampling* other than training stability. This figure shows the comparison of rendering accuracy between *Precrop* and *Edge sampling* of Chair (upper) and Drums (lower). We consider the red circles to be microscopic but clean.

6 CONCLUSION

In this paper, we propose a new pixels sampling method to stabilize NeRF training instead of *Precrop* sampling. This method consists of two steps. In the first stage, instead of *Precrop* in the early stage of learning, we use a pixels sampling method based on edge detection to sample the pixels to be trained. This stabilizes the learning of NeRF. In the second stage, after the initial learning stage, pixel sampling is performed based on the value of the squared error between the predicted pixel and the correct pixel. Here,

the learning of NeRF is accelerated. The combination of these two sampling methods makes the learning of NeRF more stable.

In our experiments, we compare each sampling method with conventional NeRF and InfoNeRF, which corresponds to a small amount of data. By using *Edge sampling*, we have achieved stable learning for any training data. *SE sampling* also achieves acceleration of learning. In the case of the small amount of data, *Edge sampling* not only stabilizes the learning but also tends to improve the PSNR results. These results indicate that edges are very important for NeRF training.

Our proposed method has two limitations. The first is that the hyperparameter for how far *Edge sampling* should be performed must be set appropriately. Research will be needed on pixels sampling methods that can learn NeRF for any data without setting this hyperparameter. The second is that *SE sampling* requires a large amount of memory space because the squared error of twice as many pixels used for training must be calculated. So, we need to consider pixels sampling methods that are more efficient with the number of pixels used for training.

REFERENCES

Ali, M. and Clausi, D. (2001). Using the canny edge detector for feature extraction and enhancement of remote sensing images. In *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*, volume 5, pages 2298–2300. Ieee.

Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., and Hedman, P. (2022). Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.

Deng, K., Liu, A., Zhu, J.-Y., and Ramanan, D. (2022). Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891.

Du, Y., Zhang, Y., Yu, H.-X., Tenenbaum, J. B., and Wu, J. (2021). Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society.

Fukuda, K., Kurita, T., and Aizawa, H. (2023). Neural radiance fields with regularizer based on differences of neighboring pixels. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Gai, Z., Liu, Z., Tan, M., Ding, J., Yu, J., Tong, M., and Yuan, J. (2023). Egra-nerf: Edge-guided ray allocation for neural radiance fields. *Image and Vision Computing*, 134:104670.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Jang, W. and Agapito, L. (2021). Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12949–12958.

Kim, M., Seo, S., and Han, B. (2022). Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921.

Lin, C.-H., Ma, W.-C., Torralba, A., and Lucey, S. (2021). Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2021). Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219.

Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., and Yu, J. (2021). Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6361.

Metzer, G., Richardson, E., Patashnik, O., Giryes, R., and Cohen-Or, D. (2023). Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673.

Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*.

Nguyen, T., Chen, Z., and Lee, J. (2020). Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*.

Nguyen, T., Novak, R., Xiao, L., and Lee, J. (2021). Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198.

Niemeyer, M., Barron, J. T., Mildenhall, B., Sajjadi, M. S., Geiger, A., and Radwan, N. (2022). Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490.

Oechsle, M., Peng, S., and Geiger, A. (2021). Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 5589–5599.

Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R. (2021a). Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874.

Park, K., Sinha, U., Hedman, P., Barron, J. T., Bouaziz, S., Goldman, D. B., Martin-Brualla, R., and Seitz, S. M. (2021b). Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. (2022). Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Pumarola, A., Corona, E., Pons-Moll, G., and Moreno-Noguer, F. (2021). D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327.

Singh, A., Singh, M., and Singh, B. (2016). Face detection and eyes extraction using sobel edge detection and morphological operations. In *2016 Conference on Advances in Signal Processing (CASP)*, pages 295–300. IEEE.

Sobel, I., Feldman, G., et al. (1968). A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. (2021). Curriculum learning: A survey. *CoRR*, abs/2101.10382.

Srinivasan, P. P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., and Barron, J. T. (2021). Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504.

Ueda, I., Fukuhara, Y., Kataoka, H., Aizawa, H., Shishido, H., and Kitahara, I. (2022). Neural density-distance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 53–68. Springer.

Wang, Z., Wu, S., Xie, W., Chen, M., and Prisacariu, V. A. (2021). Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Yang, W., Chen, G., Chen, C., Chen, Z., and Wong, K.-Y. K. (2022). Ps-nerf: Neural inverse rendering for multi-view photometric stereo. In *European Conference on Computer Vision*, pages 266–284. Springer.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2021). pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587.

Zhang, H., Jolfaei, A., and Alazab, M. (2019). A face emotion recognition method using convolutional neural network and image edge computing. *IEEE Access*, 7:159081–159089.