Few-Shot NeRF-Based View Synthesis for Viewpoint-Biased Camera Pose Estimation

Sota Ito[1](✉), Hiroaki Aizawa[2], and Kunihito Kato[1]

[1] Faculty of Engineering, Gifu University, 1-1 Yanagido, Gifu 501-1193, Japan
ito@cv.info.gifu-u.ac.jp, kato.kunihito.k6@f.gifu-u.ac.jp

[2] Graduate School of Advanced Science and Engineering, Hiroshima University,
1-4-1 Kagamiyama, Higashi-Hiroshima City, Hiroshima 739-8527, Japan
hiroaki-aizawa@hiroshima-u.ac.jp

Abstract. Recently, several works have paid attention to view synthesis by neural radiance fields (NeRF) to improve camera pose estimation. Among them, LENS and Direct-PoseNet synthesize novel views from pre-trained NeRF and then train the pose regression convolutional network using real observations and the augmented synthetic views for better localization. Therefore, the performance depends on the three-dimensional (3D) consistency and the image quality of novel views. Especially, localization tends to fail if a diverse and high-quality training set is unavailable. To solve this issue, we tackle the problem of learning camera pose regressor from the viewpoint-biased and limited training set. We propose augmenting the regressor's training set using a few-shot NeRF instead of an original NeRF, which is employed in the previous frameworks. We can render high-quality novel views with a consistent 3D structure for stable training of the regressor. The experiments show that few-shot NeRF is an effective data augmenter for camera pose estimation under the viewpoint-biased limited training set.

Keywords: Neural Radiance Fields · Camera Pose Estimation

1 Introduction

Camera pose estimation, the task of regressing a camera's relative position and rotation to an object in a given image, is a fundamental problem in computer vision and robotics. Given RGB or RGB-D images [13], we can estimate the camera parameters by reconstructing the target scene using SfM [12], regressing the camera pose [3], or iteratively optimizing the camera parameters [6]. Recently, many researchers have paid attention to the use of view synthesis by neural radiance fields (NeRF) [8] to improve camera pose estimation. Among them, LENS [9] and Direct-PoseNet [1] are practical and sophisticated approaches that utilize novel views from pre-trained NeRF for localization. Concretely, LENS utilizes the novel view rendered from the original NeRF [8] as data augmentation

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14255, pp. 308–319, 2023.
https://doi.org/10.1007/978-3-031-44210-0_25

to train the camera pose regressor, PoseNet [3], which directly estimates camera parameters from a given single image using a convolutional neural network. Direct-PoseNet has a similar approach. However, if a diverse and abundant set of multiview images is unavailable during the training of NeRF, it may not effectively render novel views. These novel views are crucial for training the camera pose regressor. In such circumstances, the quality of novel views could degrade, leading to suboptimal performance in camera localization.

Hence, in this paper, we tackle the problem of learning the pose regressor from the viewpoint-biased and limited training set. Because the training of NeRF tends to fail in such a situation, we propose augmenting the regressor's training set using a few-shot NeRF instead of an original NeRF, which is employed in the previous frameworks. Concretely, we adopt DietNeRF [2] as a few-shot NeRF for data augmentation. Using DietNeRF, we can render high-quality novel views with a consistent 3D structure for stable training of the regressor. In the training phase of the regressor, we learn the regressor to make it more stable using actual observed data and extended views rendered from the pre-trained DietNeRF.

In the experiments, to validate the effectiveness of the proposed method, we compared DietNeRF with the original NeRF using training data with a small number of shots and viewpoint bias. Our experiments demonstrated that the novel views by the DietNeRF further improve the camera pose estimation performance compared to the original NeRF.

2 Related Work

2.1 Neural Radiance Fields

Mildenhall *et al.* proposed neural radiance fields (NeRF) [8] for learning a multilayer perceptron (MLP) that represents the three-dimensional (3D) space of a target scene from multi-view images with camera pose. The learned 3D representation can be utilized to generate an unobserved scene.

While NeRF can learn the consistent 3D structure and generate realistic novel views, training NeRFs requires multi-view images with camera parameters, which is laborious. Moreover, when the training data is small or when the viewpoints are biased, the training tends to fail the training and generate poor-quality rendering images. Therefore, several studies have been proposed to reduce the number of training data [2,4,15]. pixelNeRF [15] is a method for learning NeRFs from a single image by conditioning the color and density of the 3D coordinates on the features extracted by the trained CNN. InfoNeRF [4] learns to minimize the density of sampling points on the ray except for high-density points where an object exists, thereby suppressing the effect of noise and improving the quality of the image generation. DietNeRF [2] is a method that uses CLIP [10] for training to prevent training collapse and improve the generation quality when the amount of training data is small. This is because CLIP's image encoder can extract semantic features to make the semantic features similar between viewpoints in the 3D space during training so that unobserved regions that do not appear in the training data can be made semantically consistent. As a result, even when the training data is small or the training viewpoints are biased, it is possible to learn so that unobserved regions are complemented plausibly.

DietNeRF. In this section, we describe the training phase of DietNeRF in detail. The DietNeRF model takes 3D coordinates $\mathbf{x}$ and view direction $\mathbf{d}$ as input and outputs the density σ and color $\mathbf{c}$ of the 3D coordinates. This mapping function is modeled by a multi-layer perceptron (MLP). Next, to calculate the pixel value, we sample a ray $\mathbf{r}$ on 3D space based on camera pose, aggregate these properties $(\sigma, \mathbf{c})$ for each ray through the MLP, and then calculate the pixel value $\mathbf{C}(\mathbf{r})$ based on a volume rendering approach. The MLP's trainable parameters are optimized by minimizing the following photometric loss function,

$$\mathcal{L}_{\mathrm{MSE}}(\mathcal{R}) = \frac{1}{N} \sum_{\mathbf{r} \in \mathcal{R}} ||\mathbf{C}(\mathbf{r}) - \hat{\mathbf{C}}(\mathbf{r})||_2^2, \tag{1}$$

where $\mathbf{C}(\mathbf{r})$ is a ground truth color and $\mathcal{R}$ is a set of N rays.

To hallucinate unseen regions, DietNeRF introduces the auxiliary semantic loss function, which aims to minimize the semantic distance between feature vectors of ground truth image $\mathbf{I}$ and synthesized image $\hat{\mathbf{I}}$. These feature vectors are extracted from CLIP's [10] image encoder ϕ. This process is formulated as

$$\mathcal{L}_{\mathrm{sc}}(\mathbf{I}, \hat{\mathbf{I}}) = \phi(\mathbf{I})\phi(\hat{\mathbf{I}})^{\top}. \tag{2}$$

The total loss function for training DietNeRF is described as

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{MSE}} \mathcal{L}_{\mathrm{MSE}} + \lambda_{\mathrm{sc}} \mathcal{L}_{\mathrm{sc}}, \tag{3}$$

where λ_{MSE} and λ_{sc} are hyperparameters that balance these loss function. Before training the pose regressor, we train DietNeRF according to the final loss function (Eq. (3)).

2.2 Camera Pose Estimation

Camera pose estimation is a key component for various applications. To achieve this task, several approaches have been proposed. Among them, absolute pose regression learns to regress the camera parameter from a given image by convolutional neural networks (CNN) from a pair of target scenes and the corresponding camera pose. PoseNet is one of the representative works. PoseNet regresses the parameters using MobileNet-V2 [11], enabling fast inference. However, since PoseNet is based on CNN, it easily overfits the training data and the camera distribution, resulting in poor performance. In addition, overfitting can be apparent when large-scale and diverse training data is unavailable.

For better estimation, many researchers have paid attention to the use of novel view synthesis techniques using NeRF [8]. LENS [9] augments the unseen views using NeRF-W [7] to enhance the pose regressor training. LENS generates a 3D grid based on density information obtained from NeRF-W and selects

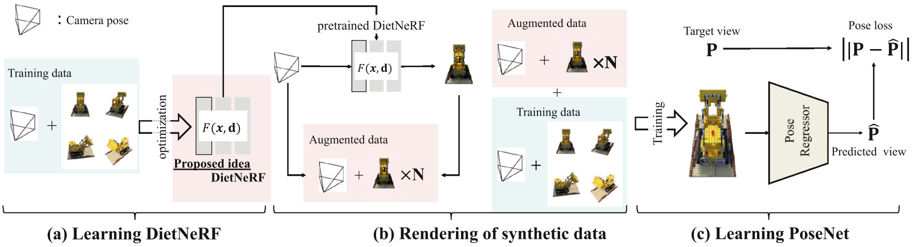


Fig. 1. Overview of the proposed method (a) Training DietNeRF from a small amount of training data. (b) Generating synthetic data for PoseNet using DietNeRF. (c) Training PoseNet using synthetic data and a small amount of real training data.

a viewpoint that is not too close to the object's location. The camera poses generated from the nearest camera pose from the selected viewpoint, and the camera pose generated from that viewpoint using NeRF-W are added to the training of the pose regressor. Direct-PoseNet [1] also uses pre-trained NeRF photometric errors for training. This has the advantage that unlabeled images can be used to train Pose Regressor.

However, the property of CNN-based regressors like PoseNet heavily depends on the view quality and the viewpoint distributions. In addition, building a training set for both the regressor and NeRF model is laborious. Therefore, in this paper, we use a few-shot NeRF, which can generate plausible unobserved views from the limited dataset, to augment training data for boosting the regressor's generalization ability.

3 Proposed Method

In this paper, we introduce an improved pipeline for few-shot and viewpoint-based camera pose estimation. As shown in the Fig. 1, the proposed method consists of three steps: training DietNeRF [2] as a view augmenter (Sect. 2.1 and Sect. 3.1), generating synthetic data for PoseNet [3] (Sect. 3.2), and training PoseNet for camera pose estimation (Sect. 3.3).

3.1 The Training of DietNeRF

To generate novel views for the pose regressor as shown in Fig. 1(a), we first train DietNeRF [2] from a given small dataset using the procedure in (Sect. 2.1).

3.2 View Synthesis for Data Augmentation

The camera pose regressor like PoseNet [3] has a problem of overfitting the training data when the limited and biased training data, results in poor performance for camera pose estimation. To solve this problem, our strategy is to utilize the novel views rendered by DietNeRF [2] as additional training data. The augmented data set consists of the image from the unseen viewpoint, and the corresponding camera poses because we can obtain the pairs from DietNeRF.

To sample viewpoints for training data augmentation, we assume that we are observing the target object from a hemispherical plane with a constant distance. Typically, such viewpoint distribution is based on a von Mises distribution in the directional statistic. The distribution changes depending on the parameters of the mean and concentration relative to the mean. When the concentration is zero, the von Mises distribution returns to a uniform distribution. Since the viewpoint of the composite data should have a viewpoint that captures a wide range of the target scene as in the uniform distribution, the azimuth and elevation angles are sampled from the von Mises distribution with mean 0 and concentration 0, and the 3D coordinates are determined. Following this sampling strategy, we sample N viewpoints consisting of azimuth and elevation angles from the von Mises distribution. Given sampled viewpoints, we generate additional view images from DietNeRF, as shown in Fig. 1(b).

3.3 The Training of Camera Pose Regressor

Finally, we train PoseNet [3] using real multi-view images with camera pose and synthetic additional images generated from DietNeRF [2] (Sect. 3.2), as shown in Fig. 1. The camera extrinsic parameters for camera pose estimation consist of the rotation and translation matrix. The following loss functions $\mathcal{L}_{\text{pose}}$ are defined based on the predicted camera pose $\hat{\mathbf{P}}$ and the ground-truth $\mathbf{P}$ of the training data.

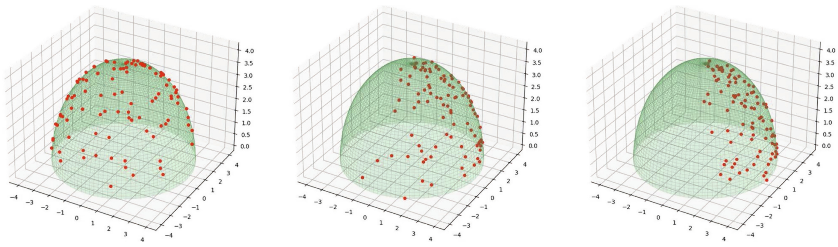
$$\mathcal{L}_{\text{pose}} = \frac{1}{|\mathbf{P}|}||\mathbf{P} - \hat{\mathbf{P}}||_2^2. \tag{4}$$

4 Evaluation

We perform experiments from two perspectives: (i) we quantitatively and qualitatively evaluate a novel view quality of the original NeRF and DietNeRF for view augmentation in a viewpoint-biased setting, (ii) we quantitatively compare our model with previous work for camera pose estimation task.

4.1 Evaluation Setting

Dataset. We used NeRF synthetic dataset proposed in the original NeRF paper [8]. This dataset is rendered from a high-quality 3D model using Blender. Because we aim to improve the performance of the few-shot and viewpoint-biased settings in the experiments, we created subsets of 10 images as a training set from NeRF synthetic for training models. Following Sect. 3.2, we sampled augmented unseen viewpoints from a von Mises distribution with a concentration of 0. To evaluate our model in a viewpoint-based setting, we controlled the mean parameter of the von Mises distribution. The viewpoint-biased data we created are categorized into three types: `random`, `side`, and `front`. These viewpoint distributions cover the hemisphere, the target object from the side, and the



(a) `low` (b) `middle` (c) `high`

Fig. 2. `low`, `middle`, and `high`-concentrated viewpoint distribution for evaluation in the viewpoint-biased setting

Table 1. Number of training successes of `random`

Scene	Model	Successes rate
Lego	NeRF	2/5
	DietNeRF	**5/5**
Hotdog	NeRF	0/5
	DietNeRF	**5/5**
Drums	NeRF	**5/5**
	DietNeRF	**5/5**

Table 2. Number of training successes of `side`/`front`

Scene	Model	Successes rate
Lego	NeRF	1/5 / 2/5
	DietNeRF	**5/5 / 5/5**
Hotdog	NeRF	2/5 / 1/5
	DietNeRF	**5/5 / 5/5**
Drums	NeRF	**5/5 / 5/5**
	DietNeRF	**5/5 / 5/5**

target object from the front, respectively. Therefore, because `side` and `front` include largely invisible regions due to self-occlusion, we investigated variations of the `side` and `front` viewpoints. By controlling the azimuth of von Mises distribution, we additionally created `high`, `middle`, and `low` concentrated viewpoint datasets for `side` and `front` viewpoints. These `high`, `middle`, and `low` concentrated viewpoints differ in the degree of observation regions, as shown in Fig. 2.

Evaluation Metrics. We quantitatively evaluated the image completion quality in an invisible region using Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM) [14], and Learned Perceptual Image Patch Similarity (LPIPS) [16]. To quantitatively evaluate the camera pose estimation, we used translation error and rotation error as metrics. The translation error indicates the error of the camera position and is calculated from the mean-squared error of the translation matrix between the ground truth and prediction. On the other hand, the rotation error indicates the mean-squared error of the rotation angle between the ground truth and prediction.

Network Details. We optimized the trainable parameters of DietNeRF using Adam [5], where the batch size is 1,024 and the initial learning rate is set to

Table 3. Rendering quality for `random`

Scene	Viewpoint	Model	PSNR	SSIM	LPIPS
Lego	`low`	NeRF	15.72	.688	.304
		DietNeRF	**24.10**	**.866**	**.109**
Hotdog		NeRF	20.14	.837	.179
		DietNeRF	**26.94**	**.928**	**.066**
Drums		NeRF	19.17	.812	.161
		DietNeRF	**19.66**	**.830**	**.124**

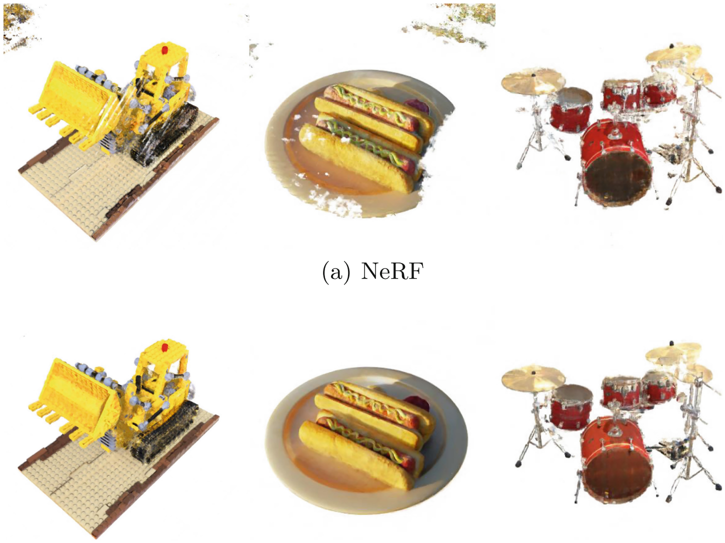
Table 4. Rendering quality for `side`/`front`

Scene	Viewpoint	Model	PSNR	SSIM	LPIPS
Lego	`low`	NeRF	19.23/**22.07**	.800/.837	.178/.132
		DietNeRF	**22.27**/22.02	**.836/.841**	**.114/.117**
	`middle`	NeRF	26.83/**25.45**	**.917/.903**	.064/**.073**
		DietNeRF	**27.21**/25.26	**.916**/.900	**.061**/.076
	`high`	NeRF	28.89/**27.23**	**.945/.936**	.043/**.048**
		DietNeRF	**29.33**/27.04	**.946**/.931	**.042**/.055
Hotdog	`low`	NeRF	**24.02**/23.63	**.882**/.877	**.113**/.123
		DietNeRF	20.47/**25.55**	.858/**.902**	.122/**.087**
	`middle`	NeRF	**32.18**/30.00	**.957**/.941	**.041**/.057
		DietNeRF	29.48/**30.12**	.942/**.945**	.056/**.053**
	`high`	NeRF	**35.01/32.50**	**.974/.965**	**.025/.033**
		DietNeRF	33.16/32.37	.966/**.965**	.035/.036
Drums	`low`	NeRF	18.34/18.07	.809/.811	.171/.179
	`middle`	DietNeRF	**19.41/19.84**	**.827/.833**	**.124/.120**
		NeRF	21.14/21.55	.862/.861	.115/.119
		DietNeRF	**21.62/22.02**	**.873/.871**	**.089/.090**
	`high`	NeRF	22.56/22.95	.891/.886	.088/.094
		DietNeRF	**23.13/23.14**	**.900/.892**	**.069/.076**

0.0005. For stability training, we applied an exponentially decreasing scheduler, increasing the learning rate by 0.1 over 250,000 iterations. Following the paper [2], we minimized Eq. (3) until 200,000 iterations and then minimized Eq. (1) from 200,000 to 250,000 iterations for better generalization.

4.2 The Completion Performance of DietNeRF

Quantitative Comparison. Quantitative completion results for NeRF and DietNeRF and the number of successful studies are shown in Table 1, 2, 3 and 4. From the `random` distribution results as shown in Table 3, we confirmed that



(a) NeRF

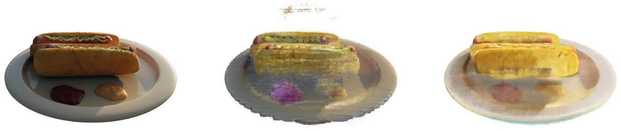
(b) DietNeRF

Fig. 3. Visual comparison between NeRF and DietNeRF. NeRF produces artifacts in an unseen viewpoint when the training dataset is small

DietNeRF did not tend to overfit training data, and the rendering quality was slightly better than that of NeRF. Especially, for the HotDog target, DietNeRF outperformed NeRF in terms of rendering quality and training stability. These results are similar to those reported in the DietNeRF paper [2] and indicate that DietNeRF's generalization ability is superior to the vanilla NeRF model. On the other hand, when the training viewpoint distribution is biased to `side`, the PSNR scores of NeRF and DietNeRF for Hotdog target were 24.02 and 20.47, respectively. In the case of `front`, the PSNR was 23.63 and 25.55, the opposite results. From these comparison results in the viewpoint-biased settings, we found that the rendering performance of DietNeRF tends to depend on the training viewpoint distribution and target object.

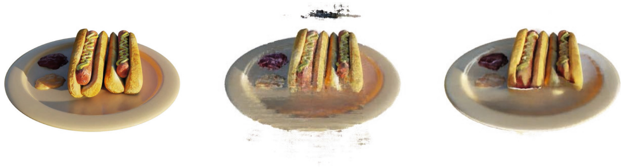
Visual Results of `random`. We closely looked at the rendering quality of NeRF and DietNeRF for boosting camera pose estimation performance. The rendering images of NeRF in the viewpoint-based setting are shown in Fig. 3. The figure clearly showed that while NeRF's PSNR score was partially competitive to DietNeRF, the rendering results in unseen viewpoints collapsed and had artifacts. On the other hand, DietNeRF can complete the unseen region even if the scene was not observed in the training phase. This is because CLIP's semantic feature enhances the viewpoint generalization of DietNeRF.

Visual Results of `side` and `front`. Figures 4 and 5 show the rendering results of `side` and `front` settings, respectively. Interestingly, we found that DietNeRF's completion ability depends on not only the training viewpoint distribution but



(a) Ground Truth (b) NeRF (c) DietNeRF

Fig. 4. Rendering results outside of `Side`'s training viewpoint



(a) Ground Truth (b) NeRF (c) DietNeRF

Fig. 5. Rendering results outside of `front`'s training viewpoint

also the **symmetric property** of the target object. Specifically, we found that DietNeRF tends to be able to complement invisible regions when the object has a symmetrical structure (Lego, Drums, and Hotdog) and the learning perspective captures one side of the symmetry.

When the training viewpoints are biased (`side` and `front`) and when DietNeRF is superior to NeRF, NeRF is sometimes superior to DietNeRF in the quality of the validation data (`middle` and `high`) for the vicinity of the training viewpoint. This indicates that while DietNeRF performs well in complementing unseen regions, it may not be as good as NeRF in producing quality for visible regions.

4.3 The Performance of Camera Pose Estimation

Quantitative Comparison. The results of camera pose estimation in the `random`, `side`, and `front` are shown in Tables 5 and 6, respectively. These scores were obtained from the PoseNet trained on real data and synthetic data by NeRF and DietNeRF. When training data was sampled from `random` distribution, DietNeRF was able to generate more high-quality novel views than NeRF, and the rendering images could enhance PoseNet's generalization, as shown in Table 5. When the training view was biased to `side` or `front`, Among NeRF and DietNeRF, higher generation quality had better camera pose estimation accuracy.

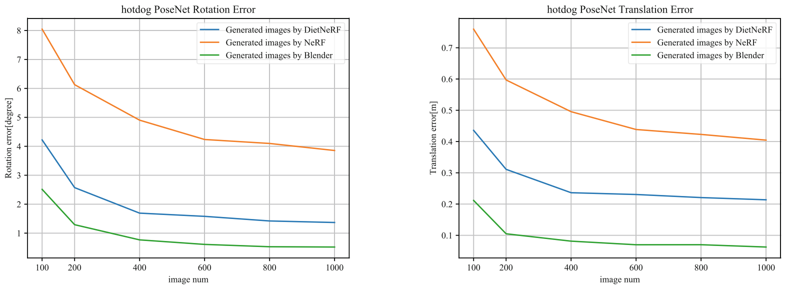
The Effect of Viewpoint Augmentation Scale. Figure 6 shows the results when changing the number of additional data generated by Blender, NeRF, and DietNeRF. From the figure, we found that the performance was significantly

Table 5. Camera pose estimation result of `random`.

Scene	Viewpoint	Synthetic data	Translation error [m]	Rotational error [°]
Lego	`low`	None	2.1019	31.34
		NeRF	1.1960	15.77
		DietNeRF	**0.3682**	**3.37**
Hotdog		None	1.9318	26.14
		NeRF	0.8784	10.15
		DietNeRF	**0.4227**	**4.04**
Drums		None	2.3129	40.71
		NeRF	0.5479	**4.46**
		DietNeRF	**0.4777**	**4.78**

Table 6. Camera pose estimation results of `side`/`front`

Scene	Viewpoint	Synthetic data	Translation error [m]	Rotational error [°]
Lego	`low`	None	3.0805/3.2724	88.07/86.82
		NeRF	1.5110/**0.7490**	33.35/**10.55**
		DietNeRF	**1.1234**/0.8608	**20.92**/13.14
	`middle`	None	1.8646/2.2959	36.59/47.11
		NeRF	**0.2384**/**0.2710**	**3.91**/**4.44**
		DietNeRF	0.2564/0.2919	4.31/5.32
	`high`	None	1.5803/1.9970	22.07/29.79
		NeRF	**0.1524**/0.2050	**2.64**/3.57
		DietNeRF	0.1795/**0.1828**	3.10/**3.18**
Hotdog	`low`	None	3.2312/2.9927	93.55/81.33
		NeRF	**1.3533**/1.3833	**15.41**/20.09
		DietNeRF	1.4252/**0.8760**	17.54/**10.28**
	`middle`	None	2.0696/1.7145	38.40/39.13
		NeRF	**0.3128**/0.4053	**3.86**/5.06
		DietNeRF	0.3399/**0.3543**	4.19/**4.32**
	`high`	None	1.8319/1.3600	24.10/21.76
		NeRF	**0.1518**/0.2698	**1.90**/3.19
		DietNeRF	0.1648/**0.2645**	2.05/**3.11**
Drums	`low`	None	2.9115/3.0163	83.87/84.22
		NeRF	1.4975/1.7794	28.62/38.94
		DietNeRF	**1.1673**/**1.2222**	**17.62**/**18.02**
	`middle`	None	2.1315/2.0425	38.18/44.91
		NeRF	0.4134/0.6202	6.00/11.24
		DietNeRF	**0.3774**/**0.5446**	**5.80**/**8.64**
	`high`	None	1.9176/1.7196	24.15/26.76
		NeRF	**0.2350**/**0.2865**	**3.06**/**4.49**
		DietNeRF	0.2480/0.3323	3.69/4.94



(a) Rotation error

(b) Translation error

Fig. 6. Error of camera pose estimation when the scale of synthetic data is changing. Blue: camera pose estimation error of PoseNet trained on DietNeRF synthetic images, orange: NeRF synthetic images, green: ground truth images rendered by Blender. (Color figure online)

improved by DietNeRF, and increasing the number of additional data resulted in the improvement of camera pose estimation in all synthesizers (Blender, NeRF, and DietNeRF). When the number of synthetic data was set to the number of images that minimized the error of camera pose estimation using DietNeRF trained with 10 real images, we confirmed that the error of camera pose estimation was equivalent to PoseNet trained with 100 images generated by Blender for the translation error and 150 images generated by Blender for the rotation error.

5 Conclusion

In this paper, we proposed a view augmentation technique for learning a camera pose estimation model, PoseNet, from a small amount of training data. The proposed method improves the performance of camera pose estimation by generating synthetic data from DietNeRF trained with a small amount of data by generating new viewpoint images and training PoseNet using the synthetic data and a small amount of training data. In addition, we validated the improvement of camera pose estimation by increasing the number of synthetic data and confirmed that the performance improves by augmenting training data. In future work, it is necessary to verify whether the proposed method is effective in more realistic scenes.

References

1. Chen, S., Wang, Z., Prisacariu, V.: Direct-posenet: absolute pose regression with photometric consistency. In: 2021 International Conference on 3D Vision (3DV), pp. 1175–1185. IEEE (2021)

2. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: semantically consistent few-shot view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5885–5894 (2021)
3. Kendall, A., Grimes, M., Cipolla, R.: Posenet: a convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
4. Kim, M., Seo, S., Han, B.: Infonerf: ray entropy minimization for few-shot neural volume rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12912–12921 (2022)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Lin, Y.C., Florence, P.R., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: inverting neural radiance fields for pose estimation. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1323–1330 (2020)
7. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7210–7219 (2021)
8. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021)
9. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: Lens: localization enhanced by nerf synthesis. In: Conference on Robot Learning, pp. 1347–1356. PMLR (2022)
10. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
12. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2D-to-3D matching. In: 2011 International Conference on Computer Vision, pp. 667–674. IEEE (2011)
13. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2930–2937 (2013)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
15. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4578–4587 (2021)
16. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)