GeoAug: Data Augmentation for Few-Shot NeRF with Geometry Constraints

Di Chen[1], Yu Liu[1], Lianghua Huang[1], Bin Wang[1], and Pan Pan[1]

Alibaba Group
{guangpan.cd,ly103369,ly103369,ganfu.wb,panpan.pp}@alibaba-inc.com

Abstract. Neural Radiance Fields (NeRF) show remarkable ability to render novel views of a certain scene by learning an implicit volumetric representation with only posed RGB images. Despite its impressiveness and simplicity, NeRF usually converges to sub-optimal solutions with incorrect geometries given few training images. We hereby present GeoAug: a data augmentation method for NeRF, which enriches training data based on multi-view geometric constraint. GeoAug provides random artificial (novel pose, RGB image) pairs for training, where the RGB image is from a nearby training view. The rendering of a novel pose is warped to the nearby training view with depth map and relative pose to match the RGB image supervision. Our method reduces the risk of over-fitting by introducing more data during training, while also provides additional implicit supervision for depth maps. In experiments, our method significantly boosts the performance of neural radiance fields conditioned on few training views.

Keywords: Neural Radiance Fields, Few-Shot Learning, Unsupervised Depth Estimation

1 Introduction

To sense and infer our 3-dimensional world is a natural and fundamental ability of human beings. However, not until recently did we find how remarkable the ability is when creating virtual reality (VR) systems. We can memorize a scene from one perspective and imagine its appearance from another viewpoint with no effort, yet for a VR system, it is quite difficult to develop an automatic algorithm which is able to render photo-realistic images for a novel view. This task is challenging since it not only requires to understand the 3D scene geometry, but also needs to synthesize high-frequency textures with complex viewpoint-dependent effects.

Recently, real progress has been made on novel view synthesis. A representative work is Neural Radiance Fields (NeRF) [22], which learns an implicit scene representation and generate images with volume rendering. When trained on a specific scene, a Multi-layer Perceptron (MLP) is used to estimate the volume density and color for each point in the space. Volume rendering is then used to generate the RGB image, supervised by the ground truth image with a photometric reconstruction loss. NeRF has shown its exceptional ability on high-quality image synthesis, while being conceptually simple and easy to train.

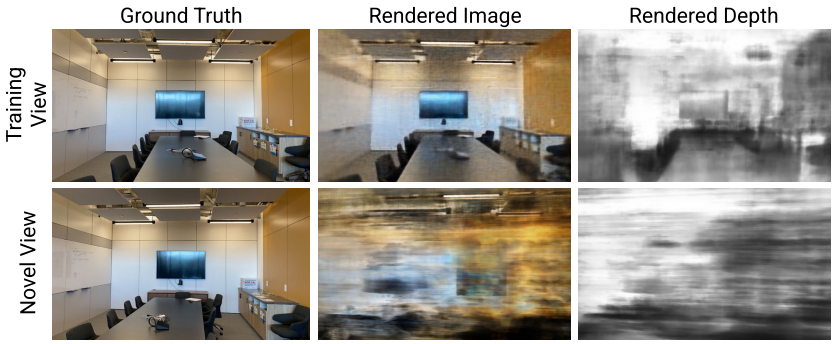

Fig. 1: NeRF overfits to few training views. NeRF produces favorable rendering result for training views, while fails to generalize to novel views. NeRF also struggles to learn the correct geometries with few training views.

Typically, NeRF needs a large amount of input views for simultaneous geometry and appearance reconstruction with high fidelity. If the training views are inadequate, NeRF tends to overfit the limited training samples. Satisfactory images could only be produced at observed poses, while the renderings under novel views are polluted with many artifacts. A comparative example is shown in Fig. 1. The reason is that NeRF cannot infer the correct 3D geometry from limited views with only RGB image supervision, resulting in a sub-optimal solution which cannot generalize to novel views [41]. Fig. 1 also illustrates that the depth maps for neither the training view nor the novel view are correctly inferred. Several works [40,33,35,12,6] have been proposed to address this problem, among which DSNeRF [6] shows its simplicity and effectiveness by adding sparse depth as an additional explicit supervision.

In this work, we aim to improve the performance of NeRF with few training views. A straightforward way is to increase the training samples by data augmentation. Meanwhile, DSNeRF [6] also shows that adding depth supervision is a valid strategy. To this end, we propose **GeoAug**, a data augmentation method for NeRF based on geometry constraint with implicit depth supervision. In addition to the original training views, we first generate camera poses under novel views and render the corresponding images. Since ground truth images are not available for novel views, we warp the rendered images to nearby training views based on the predicted depth maps and relative camera poses. Then we can impose a photometric loss between the warped images and training images. A comprehensive illustration is shown in Fig. 2. Since the warping operation is differentiable and involves depth information, the model is encouraged to learn depth estimation implicitly, which could be seen as a strong geometric constraint.

Our method is inspired by prior work [43] which aims for unsupervised depth estimation by means of view synthesis. In [43], view synthesis serves as a proxy task which forces the network to infer the depth map, based on the insight that a view synthesis system could only perform well across multiple views if

the scene geometry is modeled correctly. Our work follows the same insight, except that we focus on view synthesis as the primary task and uses the implicit depth estimation as an extra regularizer. Compared to DSNeRF [6] which uses *explicit* and *sparse* depth supervision, our method provides *implicit* and *dense* depth signal. Both GeoAug and DSNeRF discover supervision signal for free, while being complementary to each other and easy to integrate into other NeRF based models. Empirical evaluations on NeRF Real [22,21] and DTU [13] datasets demonstrate the effectiveness of our approach on improving the synthesis quality with few training views.

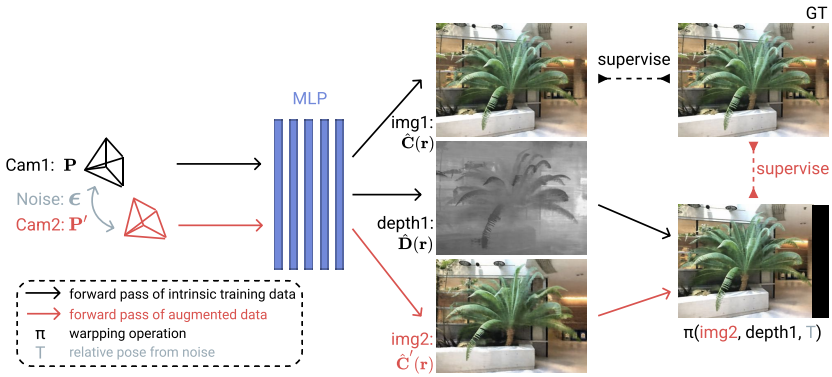

Fig. 2: Overall pipeline of training NeRF with GeoAug. We first sample a random camera pose (termed as Cam1) from training set, render its appearance image (img1) and depth map. A novel camera pose Cam2 is then created by adding noise to Cam1. The appearance image (img2) of Cam2 is rendered and warped to Cam1 with depth map and the relative pose between Cam1 and Cam2. Both img1 and the warped-img2 are supervised by the ground truth image of Cam1.

2 Related Work

Novel View Synthesis. The literature of novel view synthesis could be roughly categorized into two classes: 1) *explicit* 3D reconstruction and 2) *implicit* representations. For approaches in the first category, the 3D geometry and appearance is explicitly represented by point clouds [37], voxels [20,30], meshes [29,11,15], or multi-plane images [44,34,21,7,5]. Once the 3D scene is reconstructed, it is trivial to render the 2D image under arbitrary view. These approaches are computationally efficient, while having the merit of straightforward to check and modify the 3D structure. However, these methods are typically difficult to optimize due to the discontinuous nature.

On the other hand, implicit approaches[25,39,8,22,31] directly model the appearance of 3D scene. Without the need to represent geometry explicitly, means of discretization, *i.e.* creating voxel, mesh or multi-plane, are not adopted. Therefore, images under arbitrary views could be synthesized continuously with high definition. The representative work is Neural Radiance Fields (NeRF) [22], which maps from camera pose to color and volume density of each location in the space with an MLP. RGB images are then produced with differentiable volume rendering. Due to its simplicity and exceptional rendering quality, recent works adopt NeRF for various extensions such as generative adversarial networks [4,28,24], video synthesis [18,38], relighting [1,32], scene editing [19,14], *etc.*

Few-Shot Neural Radiance Fields. NeRF-based methods usually require a lot of images from different views for training. Several works have been proposed to address the data-hungry problem of NeRF by exploiting training data [40,35], meta learning [33] and additional supervision [12,6]. PixelNeRF [40] takes advantage of the training images during test time rendering, which is ignored by vanilla NeRF. Convolutional feature of the training image is projected onto the ray of novel view, which is later used as a conditional embedding for MLP inference. IBRNet [35] adopts a similar strategy and adds an additional ray transformer for better density estimation. Instead of randomly initialize the weights of MLP, MetaNeRF [33] propose to pre-train the network on a large-scale dataset before fine-tuning on each scene. DietNeRF [12] adds a pair-wise loss which regularizes multi-view consistency by pulling the cosine distance between high-level semantic features of different views. RegNeRF [23] renders image patches from unseen camera views and regularize the RGB values with a trained normalizing flow model. The density values are also regularized by a smoothness loss. DSNeRF [12] utilizes the sparse depth information generated by COLMAP [27] as an explicit supervision for the rendered depth map. Our work shares similar insight to DSNeRF, *i.e.* provide supervision for depth map, except that our approach is unsupervised, requiring no additional data nor annotations. In this paper, we mainly apply our data augmentation method upon DSNeRF, yet it is worth to notice that our method is compatible to all NeRF-based models.

Unsupervised Depth Estimation. Our GeoAug method is closely related to works on unsupervised depth estimation [43,10,3,2], which utilize the geometric constraints between frames as supervisory signal. During training, adjacent video frames, denoted as source/target frames, are sampled as inputs to a depth network and a pose network respectively. The output depth map and relative camera pose are used to warp the source frame to the viewpoint of target frame. The photometric error between the warped source frame and target frame is minimized during training, which represents the geometric constraint. During inference, the depth network could be used separately for depth estimation.

For NeRF models, camera poses are estimated beforehand by SfM [27], thus no pose network is needed. The depth map could be rendered in a similar way to RGB images. Therefore, the warping-based geometric constraint could be used as an additional supervision for NeRF, encouraging a better understanding of scene geometry.

3 Methodology

We now present our geometry-aware data augmentation method in this section. We begin by revisiting the classic NeRF method and volumetric rendering, with an important baseline: DSNeRF [6], which adds additional sparse depth supervision to NeRF. Then we introduce our data augmentation method as well as the adaptive noise module. Finally, we summarize the overall training procedure.

3.1 Revisiting Volume Rendering

NeRF [22] is originally proposed for the task of novel view synthesis, which aims at rendering an RGB image given a camera pose $\mathbf{P}$. This is implemented by 1) shooting rays from the center of camera pose $\mathbf{P}$, 2) predicting the radiance intensity at each point along all the rays and 3) rendering each pixel by accumulating the radiance of all points along the ray.

Specifically, given a camera center $\mathbf{o} \in \mathbb{R}^3$ and viewing direction $\mathbf{d}$, the corresponding ray is represented as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, parameterized by t. For a specific point $\mathbf{x} \in \mathbb{R}^3$ on the ray, NeRF uses an Multi-Layer Perceptron (MLP) to predict the color $\mathbf{c} \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}^+$. The MLP could be seen as a function f_θ that maps from spatial location and direction to radiance field: $f_\theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$.

Once the entire radiance field is available, RGB images could be rendered by integrating along rays with volume rendering:

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)\,\mathrm{d}t\,, \quad \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(s)\,\mathrm{d}s\right) \tag{1}$$

where t_n and t_f represent the near and far bounds of the rays. $T(t)$ is a transmittance term which measures the probability that light could travel from t_n to t without being obstructed.

During training, NeRF model is supervised by a reconstruction loss:

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

where $\mathcal{R}$ is the set of rays randomly sampled during training. $\mathbf{C}$ is the ground truth RGB image under camera pose $\mathbf{P}$.

DSNeRF: A Supervised Baseline. Given enough training images captured under various camera poses, NeRF is effective at representing the scene implicitly and thus rendering satisfying images at novel views. However, under few-shot settings, NeRF is prone to overfit the available scenes with incorrect geometry [41], *e.g.* a plain canvas at the camera's near bound filled with the pixels from training images [6]. Adding images from more diverse views could alleviate this problem, but they are not always available in real-world applications. To this end, DSNeRF [6] is proposed to improve NeRF under few-shot settings.

Generally, DSNeRF adds a depth reconstruction loss $\mathcal{L}_d$ alongside the RGB loss $\mathcal{L}_c$:

$$\mathcal{L}_d = \frac{1}{|\mathcal{R}_d|} \sum_{\mathbf{r} \in \mathcal{R}_d} w_{\mathbf{r}} \|\hat{\mathbf{D}}(\mathbf{r}) - \mathbf{D}(\mathbf{r})\|_2^2 \tag{3}$$

where the depth ground truth $\mathbf{D}(\mathbf{r})$ and the corresponding confidence $w_{\mathbf{r}}$ are side products of camera pose estimation with structure-from-motion (SfM) [27], which is the standard preprocess for training NeRF. $\hat{\mathbf{D}}(\mathbf{r})$ is the depth map rendered in a similar way to rendering RGB images:

$$\hat{\mathbf{D}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)t \, \mathrm{d}t \tag{4}$$

Note that SfM only produces a sparse set of points with depth, thus the sampling set $\mathcal{R}_d$ for depth supervision is different from the one for RGB supervision $\mathcal{R}$.

DSNeRF has shown it's superior performance over other NeRF variants for few-shot settings [40,33,35]. Therefore, we choose it as our baseline and test our data augmentation method upon DSNeRF.

3.2 Geometry-Aware Data Augmentation

Fig. 2 shows the overview of our GeoAug method. During training, we first add random noise to the 6 degree-of-freedom (6-DoF) representation of a camera pose $\mathbf{P}$ in the training set:

$$\mathbf{P}' = \mathbf{P} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \delta) \tag{5}$$

where the noise vector $\boldsymbol{\epsilon}$ is sampled from a Gaussian distribution with 0 mean and δ standard deviation. We then render the RGB image $\hat{\mathbf{C}}'(\mathbf{r})$ under camera view $\mathbf{P}'$ using volume rendering described in Sec. 3.1. Since there is no ground truth for $\hat{\mathbf{C}}'(\mathbf{r})$, we warp $\hat{\mathbf{C}}'(\mathbf{r})$ from $\mathbf{P}'$ to $\mathbf{P}$ and supervise the warped image with $\mathbf{C}$:

$$\mathcal{L}_a = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \|\pi(\hat{\mathbf{C}}'(\mathbf{r}), \hat{\mathbf{D}}(\mathbf{r}), T_{\mathbf{p} \to \mathbf{p}'}) - \mathbf{C}(\mathbf{r})\| \tag{6}$$

In Eq. 6, $\pi(\cdot)$ denote the differentiable image warping function. Let p' and p be the homogeneous coordinates of pixels in $\hat{\mathbf{C}}'(\mathbf{r})$ and $\hat{\mathbf{C}}(\mathbf{r})$ respectively, K the camera intrinsics matrix and $T_{\mathbf{p} \to \mathbf{p}'}$ the relative pose between $\mathbf{P}$ and $\mathbf{P}'$. The warping function $\pi(\cdot)$ is described as

$$p' \sim K T_{\mathbf{p} \to \mathbf{p}'} \hat{\mathbf{D}}(p) K^{-1} p \tag{7}$$

Since the projected coordinates p' are continuous values, we use nearest sampling to get the pixel value from $\hat{\mathbf{C}}'(\mathbf{r})$. We also ignore the pixels if p' is outside the image bound of $\hat{\mathbf{C}}'(\mathbf{r})$.

One prerequisite for successful warping between different views is that there should be no occlusion or disocclusion. However, it is not always true if the scene geometry is complex, *i.e.* a scene containing a lot of discontinuous depths. Regions with complex geometry would induce outliers in Eq. 6, corrupting the gradients and harm the training process. In our experiments, a simple L2 loss for $\mathcal{L}_a$ works just fine, which means that the outlier points are not dominating the loss. To improve the robustness of our method, we propose to designate the metric $\|\cdot\|$ in Eq. 6 as smooth L1 loss instead of the commonly used L2 loss, since smooth L1 loss is less sensitive to outliers [9]. Our experiments also show that smooth L1 loss improves the performance over L2 loss. Smooth L1 loss is defined as below:

$$\|\cdot\|_{\text{smooth L1}} = \begin{cases} 0.5 \times (\cdot)^2 & \text{if } |\cdot| < 1.0 \\ |\cdot| - 0.5 & \text{otherwise} \end{cases} \tag{8}$$

3.3 Adaptive Noise

One important hyper-parameter in our GeoAug method is the standard deviation δ of camera pose noise. δ controls the offset magnitude between $\mathbf{P}'$ and $\mathbf{P}$, *i.e.* the larger the δ, the more $\mathbf{P}'$ is deviated from $\mathbf{P}$. It could be tedious to tune δ since we have no prior knowledge of the pose offset. If $\mathbf{P}'$ is too far away from $\mathbf{P}$, the warping operation could be highly unreliable. If the offset is otherwise too small, the efficacy of data augmentation is diminished. On the other hand, using the same noise magnitude through out the entire training process may not be optimal.

Inspired by the adaptive augmentation methods used in GANs [16], we propose to tune δ *adaptively* instead of picking the appropriate δ manually through exhaustive experiments. Our method is based on a heuristic rule regarding the discrepancy between the loss values of $\mathcal{L}_c$ and $\mathcal{L}_a$. Typically, an ideal augmentation method should keep the loss of augmented samples a little higher than the loss of intrinsic training samples. In our case, we first set an initial δ_0 as a base, multiply/divide δ_0 with a factor γ ($\gamma > 1$) if $\mathcal{L}_a$ smaller/larger than a margin m over $\mathcal{L}_c$:

$$\delta_t = \begin{cases} \delta_0, & \text{if } t = 0 \\ \delta_{t-1} * \gamma, & \text{if } 2\bar{\mathcal{L}}_a < \bar{\mathcal{L}}_c + m \\ \delta_{t-1}/\gamma, & \text{if } 2\bar{\mathcal{L}}_a > \bar{\mathcal{L}}_c + 2m \\ \delta_{t-1}, & \text{if } \bar{\mathcal{L}}_c + m \leq 2\bar{\mathcal{L}}_a \leq \bar{\mathcal{L}}_c + 2m \end{cases} \tag{9}$$

where $\bar{\mathcal{L}}$ is the averaged loss value of the most resent 100 training steps. In other words, the rule defined by Eq. 9 aims to keep the loss value of $2\bar{\mathcal{L}}_a$ between $\bar{\mathcal{L}}_c + m$ and $\bar{\mathcal{L}}_c + 2m$.

Although our adaptive noise rule brings additional hyper-parameters, *i.e.* δ_0, γ and m, we find through experiments that the choice of δ_0 and γ does not affect the performance too much, since δ_t would converge to the same range quickly.

Algorithm 1: Training DSNeRF with GeoAug

Data: Training set $\{\mathbf{P}, \mathbf{C}, \mathbf{D}\}$, initial noise standard deviation δ_0, noise growing factor γ, loss margin m, number of augmented sample per-iteration N, learning rate η, loss weights λ_d, λ_a
Result: Trained radiance field function f_θ
Initialize MLP parameters of f_θ ;
$t \leftarrow 0, \delta_t \leftarrow \delta_0$;
for $t \leftarrow 1$ **to** *NumIters* **do**
  Sample rays $\mathbf{r} \in \mathcal{R}$ and ground truth $\mathbf{C}(\mathbf{r})$;
  Render $\hat{\mathbf{C}}(\mathbf{r})$ and $\hat{\mathbf{D}}(\mathbf{r})$ under view $\mathbf{P}$;
  Calculate loss $\mathcal{L}_c$ with Eq. 2 ;
  Sample rays $\mathbf{r}_d \in \mathcal{R}_d$ and depth map $\mathbf{D}(\mathbf{r})$;
  Render depth value $\hat{\mathbf{D}}(\mathbf{r}_d)$ under view $\mathbf{P}$;
  Calculate loss $\mathcal{L}_d$ with Eq. 3 ;
  $\mathcal{L}_a \leftarrow 0$;
  for $n \leftarrow 1$ **to** N **do**
    Draw noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \delta_t)$;
    $\mathbf{P}' \leftarrow \mathbf{P} + \boldsymbol{\epsilon}$;
    Render $\hat{\mathbf{C}}'(\mathbf{r})$ under view $\mathbf{P}'$;
    Warp $\hat{\mathbf{C}}'(\mathbf{r})$ with Eq. 7 ;
    Calculate loss with Eq. 6 and add to $\mathcal{L}_a$;
  end
  $\mathcal{L}_a \leftarrow \mathcal{L}_a / N$;
  Update averaged loss value $\bar{\mathcal{L}}_c$ and $\bar{\mathcal{L}}_a$;
  Update δ_t with Eq. 9 ;
  $\mathcal{L} \leftarrow \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_a \mathcal{L}_a$;
  Update parameters $\theta \leftarrow \text{Adam}(\theta, \eta, \nabla_\theta \mathcal{L})$;
end

As for the margin m, it's much easier to set than setting δ directly, since the magnitude of m is strongly related to the loss value of $\mathcal{L}_c$, which is intuitive to find an appropriate magnitude through experiments. Therefore, we set δ_0, γ and m to 5e−5, 1.01 and 3e−3 respectively in all the experiments without further tuning. Note that $\bar{\mathcal{L}}_a$ is re-scaled by a factor of 2 in Eq. 9. It is because that we have to align the magnitude of $\bar{\mathcal{L}}_a$ and $\bar{\mathcal{L}}_c$ when compared directly, on account of the fact that smooth L1 loss [9] scales the L2 loss part by 0.5.

3.4 Training

During training, we can augment each sample by N times and average the losses of all augmented samples as $\mathcal{L}_a$. For convenience, we set $N = 1$ throughout the paper. The final loss $\mathcal{L}$ is a linear combination of the original NeRF loss $\mathcal{L}_c$, sparse depth loss $\mathcal{L}_d$ and the loss of augmented samples $\mathcal{L}_a$:

$$\mathcal{L} = \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_a \mathcal{L}_a \tag{10}$$

The loss weight λ_d and λ_a are set to 0.01 and 0.1 throughout the paper. The complete training process is summarized in **Algorithm 1**, where the gray background marks the sparse depth supervision for DSNeRF. Our GeoAug procedures are marked with green.

4 Experiments

For experiments, we first introduce the implementation details and the datasets in Sec. 4.1. We then compare the view synthesis performance with other methods in Sec. 4.2. Finally, we conduct ablation study in Sec. 4.3, including the metric choice of $\mathcal{L}_a$, efficacy of adaptive noise and performance under multiplicity settings.

4.1 Settings

Implementation Details. Our models are implemented with PyTorch [26]. To improve computational efficiency, we made several modifications to the original NeRF model. 1) Instead of using two-stage MLPs with hierarchical sampling, we only use a single MLP with stratified sampling; 2) The network width is reduced from 256 to 128; 3) Each ray is discretized uniformly into 128 points. We keep using this configuration throughout the paper. Moreover, we also change the random pixel sampling to patch sampling [28], in order to conduct valid warping operation. We also apply patch sampling to the baseline methods for fair comparison.

During training, we sample 4096 rays under a single view for each iteration. The model is trained for 10000 epochs with a learning rate of 0.001, which is exponentially decayed with a rate of 0.9954 every 10 epochs. We use the Adam optimizer [17] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for all models. All the experiments are conducted on a single NVIDIA Tesla V100 GPU.

Dataset and Evaluation Protocol. NeRF Real-world Data (NeRF Real) [21,22] is a real-world dataset containing 8 forward-facing scenes. We use the official test split for each scene, *i.e.* test image is sampled every 8-th image. For training images, we randomly sample 2, 5 and 10 views for three different few-shot settings.

DTU MVS Dataset (DTU) [13] is a large-scale multi-view stereo dataset captured in a controlled environment. The complete dataset contains 80 scenes, from which we choose 15 scenes for testing following the configuration of [6]. For each scene, we reduce the image resolution to 400×300 and randomly sample 3, 6, 9 views for training. We use the ground truth camera poses provided by the dataset and run COLMAP [27] by initializing the camera poses with ground truths for sparse depth information.

To evaluate the synthesis quality, we report PSNR, SSIM [36] and LPIPS [42] calculated against the corresponding ground truth.

Scene	PSNR ↑			SSIM ↑			LPIPS ↓		
	NeRF	DSNeRF	+GeoAug	NeRF	DSNeRF	+GeoAug	NeRF	DSNeRF	+GeoAug
Fern	18.08	18.34	**19.36**	0.50	0.51	**0.54**	0.60	0.60	**0.55**
Flower	18.31	19.07	**19.64**	0.46	0.51	**0.51**	0.60	0.57	**0.54**
Fortress	20.24	19.86	**20.33**	0.54	0.53	**0.57**	0.59	0.59	**0.54**
Horns	**16.23**	15.42	15.76	**0.44**	0.41	0.41	0.65	0.67	**0.64**
Leaves	**15.33**	14.89	15.17	0.32	0.32	**0.34**	0.61	0.62	**0.56**
Orchids	14.05	14.41	**14.62**	0.30	0.31	**0.32**	0.62	0.61	**0.60**
Room	20.23	20.89	**22.39**	0.72	0.73	**0.77**	0.57	0.58	**0.50**
Trex	17.26	17.49	**17.91**	0.54	0.52	**0.55**	0.59	0.61	**0.58**
Mean	17.47	17.55	**18.15**	0.48	0.48	**0.50**	0.60	0.61	**0.56**

Table 1: View synthesis results on NeRF Real dataset [21,22]. The numbers are averaged over three few-shot settings, *i.e.* 2-view, 5-view and 10-view. Our GeoAug method effectively improves DSNeRF on all three metrics.

Method	PSNR ↑			SSIM ↑			LPIPS ↓		
	3-view	6-view	9-view	3-view	6-view	9-view	3-view	6-view	9-view
NeRF [22]	11.26	13.00	15.71	0.43	0.47	0.59	0.61	0.61	0.47
DSNeRF [6]	13.47	14.82	18.81	0.49	0.57	**0.69**	0.58	0.52	0.43
DSNeRF +GeoAug	**14.91**	**17.12**	**19.57**	**0.52**	**0.64**	0.68	**0.54**	**0.47**	**0.43**

Table 2: View synthesis results on DTU dataset [13]. The numbers are averaged over 15 test scenes following the setting of DSNeRF [6].

4.2 Benchmark Comparison

Comparison on NeRF Real. In this section, we inspect the perceptual quality of novel view synthesis on NeRF Real dataset [21,22]. Three NeRF variants are compared, namely, 1) the basic NeRF described in Sec. 3.1, 2) basic NeRF with sparse depth supervision, denoted as DSNeRF [6], and 3) DSNeRF with our proposed GeoAug method. We average the experiment results under three few-shot settings and present them in Tab. 1. We can see from Tab. 1 that DSNeRF improves the mean PSNR of basic NeRF by 0.08 dB, yet did not increase SSIM or LPIPS. One explanation for the limited improvement is that the noisy depth information provided by SfM is less reliable. For instance, 'Fortress', 'Horns' and 'Leaves' are the three scenes with lowest SfM confidence for depth among all scenes. Therefore, DSNeRF performs even worse than the basic NeRF on the three scenes. In contrast, our GeoAug method does not rely on external depth information, thus won't be affected by the noise from SfM. We can see that our GeoAug boosts the performance of DSNeRF in all scenes. The three metrics are consistently better than the basic NeRF and DSNeRF.

We also present qualitative comparison in Fig. 5. Video visualizations are available at `bit.ly/3wMX1Sb`. When the training images are relatively abundant, *e.g.* under 10-view setting, DSNeRF already renders satisfying images.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
DSNeRF	21.27	0.61	0.52
w. GeoAug, L2	22.07	0.64	0.48
w. GeoAug, L1	21.37	0.61	0.52
w. GeoAug, SmoothL1	**22.65**	**0.66**	**0.44**

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Noise $\delta = 5e{-}5$	22.09	0.64	0.48
Noise $\delta = 2e{-}4$	21.84	0.64	0.47
Adaptive Noise	**22.65**	**0.66**	**0.44**

Table 3: Ablation study conducted on the 'fern' scene under 10-view setting. Left: Different metric choices of augmentation loss $\mathcal{L}_a$. **Right:** GeoAug with different noise level.

Therefore, our GeoAug method only brings slight improvement w.r.t. image details like the leaves of fern and telephone wires on the desk. When the number of training images decreases, our GeoAug method shows greater improvement on preserving the overall image structure. Two representative samples are the "fern" and "room" scene in the 2-view setting, where GeoAug preserves the shape structure of fern, TV and long desk, while DSNeRF completely fail to reconstruct. For scenes with extremely complex geometries, *e.g.* stacks of petals and leaves, our method produces clearer basic structures such as flower edges and stems.

Comparison on DTU We present the numerical performance on DTU in Tab. 2. Different to the conclusion on NeRF Real dataset [21,22] where DSNeRF only brings limited improvement over NeRF, Tab. 2 shows that DSNeRF consistently improves the basic NeRF on all settings by a large margin. This is mainly due to the better point cloud estimation quality, *i.e.* depth information generated by COLMAP is more reliable since the camera poses are initialized with ground truths. As a result, DSNeRF learns better geometry since the explicit depth supervision is hindered less by noise.

Our GeoAug method does not rely on SfM estimations. Therefore, the performance of GeoAug does not depend on the dataset. Moreover, our GeoAug method provides *dense* depth supervision upon the *sparse* point depth of DSNeRF. We can see in Tab. 2 that our GeoAug method improves the performance of DSNeRF on DTU, especially under 6 and 3 view settings. The lower block of Fig. 5 shows the qualitative comparisons. GeoAug helps to render clearer local details such as the characters on the bottle and building windows. On the other hand, NeRF and DSNeRF struggle to preserve large-scale structures, *e.g.* buildings under 6 view and 3 view settings. Our method enhances DSNeRF with the ability of inferring better structures, thanks to the additional geometry constraints provided by GeoAug.

4.3 Ablation Study

Smooth L1 Loss. As discussed in Sec. 3.2, outliers are usually inevitable during the warping process. Unsupervised depth estimation methods like [43] use an explanatory mask for each frame to cast out the outliers. However, since the augmented views in our method are randomly sampled, it is implausible to maintain a mask pool for every training view. Therefore, we choose to leave the

outliers and use a robust function, *i.e.* smooth L1 loss [9] to measure the loss of augmented samples. In Tab. 3 (left), we compare different choices for augmented loss $\mathcal{L}_a$. For L2 loss, the re-scale factor for $\bar{\mathcal{L}}_a$ of adaptive noise is changed from 2 to 1, in order to match the L2 loss of original training samples. For L1 loss, we cannot compare its magnitude directly to L2 loss. Therefore, we remove adaptive noise and set the noise arbitrarily to 5e−5. We can see from Tab. 3 (left) that using L2 loss for $\mathcal{L}_a$ improves DSNeRF by 0.8 dB w.r.t. PSNR, which means that the outlier problem is not strong enough to counteract the effectiveness of our GeoAug method. Furthermore, replace L2 loss with smooth L1 loss brings an additional 0.58 dB improvement on PSNR. Therefore, smooth L1 loss is a better choice than L2 loss for handling warping outliers. Besides, we also tried to use standard L1 loss as $\mathcal{L}_a$. It only brings minor improvement to PSNR, while SSIM and LPIPS are not better. We assume it is due to its over-tolerance to large errors and the lack of adaptive noise.

Adaptive Noise. Our adaptive noise chooses the suitable noise standard deviation δ automatically for our GeoAug method, reduces the need for exhaustive parameter tuning. In Fig. 3, we demonstrate how adaptive noise works during the training process. As expected, we can see that the augmentation loss $2\mathcal{L}_a$ is kept above the reconstruction loss $\mathcal{L}_c$ with a reasonable margin. It ensures that the augmented samples are neither too easy nor too noisy. The noise standard deviation δ is initially set to 5e−5, which quickly converges to the range between 1.8e−4 and 2.2e−4.

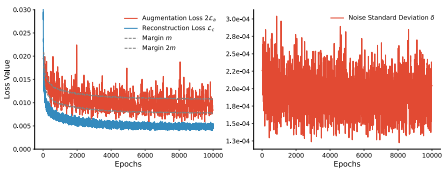

Fig. 3: Inspection on adaptive noise. Left: Reconstruction loss $\mathcal{L}_c$ and Augmentation loss $2\mathcal{L}_a$. **Right:** Magnitude of the noise standard deviation δ. Adaptive noise is designed to ensure that $2\bar{\mathcal{L}}_a$ is roughly between $\bar{\mathcal{L}}_c + m$ and $\bar{\mathcal{L}}_c + 2m$.

We also conduct an ablative experiment to show the difference between our adaptive noise and setting a global fixed noise. The results are gathered in Tab. 3 (right). We can see that arbitrarily setting δ to the initial value 5e−5 or the converged mean value 2e−4 is not optimal. Both of their performance are inferior to our adaptive noise strategy. Therefore, our adaptive noise not only reduces the need for parameter tuning, but also improves the performance of GeoAug.

Multiplicity Setting. In this section, we investigate our data augmentation method under multiplicity settings where the training images are relatively abundant. In Tab. 4, we report the synthesis result on the 'fern' scene with all 17 images for training. Different from the results under few-shot settings, the basic NeRF model performs better than DSNeRF given enough training images. This is because NeRF could avoid shape-radiance-ambiguity [41] when the training views are dense, thus inferring the correct geometry and generalizing well to

NeRF	**PSNR↑**	**SSIM ↑**	**LPIPS ↓**
NeRF	**23.09**	**0.68**	*0.44*
NeRF + GeoAug	*23.03*	**0.68**	**0.42**
DSNeRF	**PSNR↑**	**SSIM ↑**	**LPIPS ↓**
$\lambda_d = 0.001$	22.85	0.66	0.47
$\lambda_d = 0.01$	22.84	0.66	0.47
$\lambda_d = 0.1$	22.60	0.65	0.47
$\lambda_d = 0.01$ + GeoAug	**22.88**	**0.67**	**0.43**

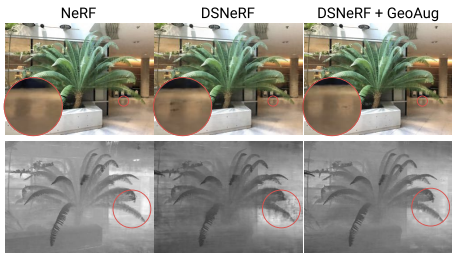

Table 4 & Fig. 4: View Synthesis with *all* training images on the 'fern' scene. Left: Given dense training view, the basic NeRF model avoids shape-radiance-ambiguity and renders high-quality novel views. DSNeRF shows inferior performance since it introduces extra noise through explicit depth supervision. Our GeoAug method harms less to the synthesis quality of NeRF. **Right:** The depth noise brought by DSNeRF causes stains on the rendering image. Our GeoAug method alleviates the negative effect of depth noise.

novel views. Under this circumstance, the depth supervision of DSNeRF only brings very limited hint on the geometry information, but instead involves additional noise resulted from imperfect SfM estimation. We can see in Tab. 4 that as the weight of depth regression loss λ_d increases, the performance of DSNeRF gets worse, since the model is forced to fit the noise. Fig. 4 also shows that the depth map and image produced by DSNeRF show more stain-like artifacts.

Similarly, the performance improvement of our GeoAug method is also diluted as more training views are available. However, our method does not rely on external depth information, thus free from SfM noises. When applied in companion with the basic NeRF model, GeoAug won't bring too much negative effect. The PSNR is only 0.06 dB lower, while the LPIPS is better than NeRF by 0.02. Meanwhile, GeoAug could also compensate the degradation of DSNeRF both quantitatively (Tab. 4) and qualitatively (Fig. 4).

5 Conclusion

In this paper, we present GeoAug: a data augmentation method for Neural Radiance Fields which alleviates the over-fitting problem under few-shot settings. During training, camera poses of random novel views are generated with an adaptive noise method, which are later used as inputs for the NeRF model. For each novel pose, the output rendering is warped to a nearby intrinsic training view and supervised by the corresponding ground truth image. In this way, our method enriches training data by leveraging geometry constraints through the warping operation, thereby posing implicit supervision on the rendered depth map. Experiments shows the effectiveness of GeoAug on improving the rendering quality for NeRF.

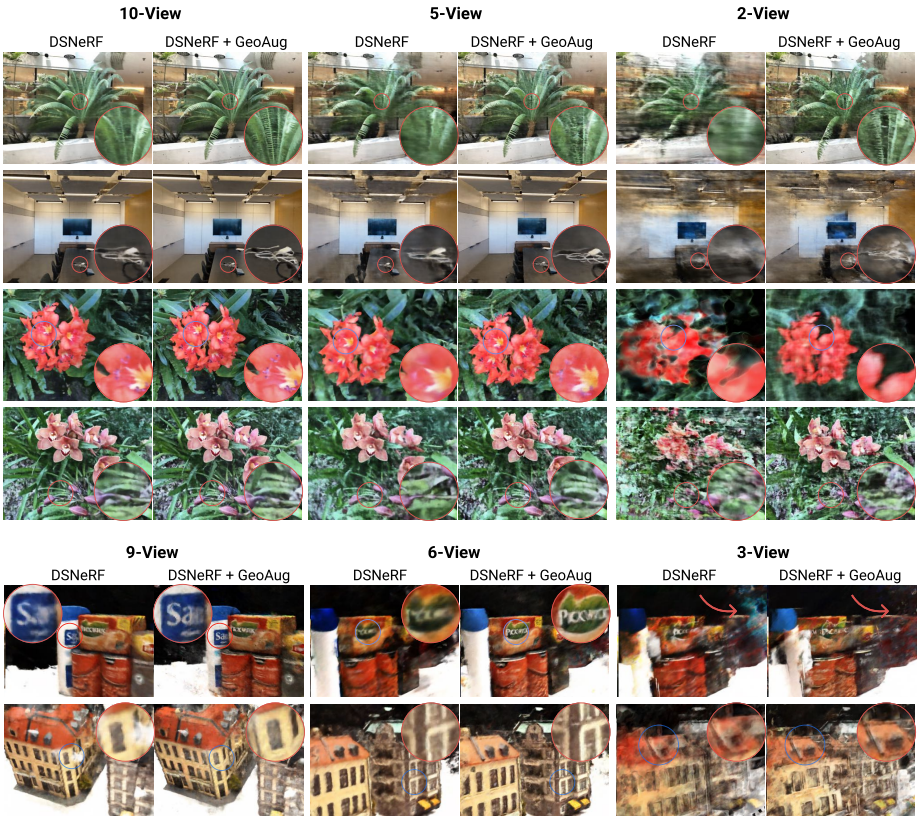

Fig. 5: Qualitative results on NeRF Real (upper block) and DTU (lower block) datasets. RGB images are rendered under different few-shot settings. Our GeoAug method helps to preserve image structure and retain more details.

Despite good performance on standard datasets, there are challenges yet to be explored: **1)** The warping outliers caused by camera movement and scene occlusions are not handled explicitly. This is the main reason why GeoAug won't improve NeRF under multiplicity settings, *i.e.* given dense training views. Although this problem is bypassed with a robust loss function, we believe that more improvement could be harvested if outliers could be managed properly. **2)** NeRF models and our GeoAug method assume that the camera pose for each training view is already known. However, under few-shot settings where the camera poses are so diverse that even SfM fails to estimate the camera pose and sparse depth, it is unlikely for NeRF models to fit training views or synthesis novel views. Therefore, extending GeoAug for NeRF models to the general purpose of multi-view stereo and structure-from-motion would be an interesting direction for future work.

References

1. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: Nerd: Neural reflectance decomposition from image collections. In: ICCV (2021) 4
2. Bozorgtabar, B., Rad, M.S., Mahapatra, D., Thiran, J.P.: Syndemo: Synergistic deep feature alignment for joint learning of depth and ego-motion. In: ICCV (2019) 4
3. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI (2019) 4
4. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: CVPR (2021) 4
5. Choi, I., Gallo, O., Troccoli, A., Kim, M.H., Kautz, J.: Extreme view synthesis. In: CVPR (2019) 3
6. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. arXiv:2107.02791 (2021) 2, 3, 4, 5, 9, 10
7. Flynn, J., Broxton, M., Debevec, P., DuVall, M., Fyffe, G., Overbeck, R.S., Snavely, N., Tucker, R.: Deepview: High-quality view synthesis by learned gradient descent. In: CVPR (2019) 3
8. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: CVPR (2016) 4
9. Girshick, R.: Fast r-cnn. In: ICCV (2015) 7, 8, 12
10. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019) 4
11. Hu, R., Ravi, N., Berg, A.C., Pathak, D.: Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In: ICCV (2021) 3
12. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: ICCV (2021) 2, 4
13. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR (2014) 3, 9, 10
14. Jiakai, Z., Xinhang, L., Xinyi, Y., Fuqiang, Z., Yanshun, Z., Minye, W., Yingliang, Z., Lan, X., Jingyi, Y.: Editable free-viewpoint video using a layered neural representation. In: SIGGRAPH (2021) 4
15. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV (2018) 3
16. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. NeurIPS (2020) 7
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014) 9
18. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: CVPR (2021) 4
19. Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.Y., Russell, B.: Editing conditional radiance fields. arXiv:2105.06466 (2021) 4
20. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. In: SIGGRAPH (2019) 3
21. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM TOG (2019) 3, 9, 10, 11

22. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 1, 3, 4, 5, 9, 10, 11
23. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022) 4
24. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: CVPR (2021) 4
25. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: CVPR (2020) 4
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 9
27. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 4, 6, 9
28. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: NeurIPS (2020) 4, 9
29. Shih, M.L., Su, S.Y., Kopf, J., Huang, J.B.: 3d photography using context-aware layered depth inpainting. In: CVPR (2020) 3
30. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: CVPR (2019) 3
31. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: NeurIPS (2019) 4
32. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: CVPR (2021) 4
33. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: Learned initializations for optimizing coordinate-based neural representations. In: CVPR (2021) 2, 4, 6
34. Tucker, R., Snavely, N.: Single-view view synthesis with multiplane images. In: CVPR (2020) 3
35. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021) 2, 4, 6
36. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP (2004) 9
37. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. In: CVPR (2020) 3
38. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: CVPR (2021) 4
39. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: NeurIPS (2020) 4
40. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021) 2, 4, 6
41. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv:2010.07492 (2020) 2, 5, 12

42. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 9
43. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017) 2, 4, 11
44. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. In: SIGGRAPH (2018) 3