Pattern Recognition 151 (2024) 110419

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Structure-aware neural radiance fields without posed camera

Shu Chen [a,*], Yang Zhang [a], Yaxin Xu [a], Beiji Zou [b]

[a] The School of Computer Science & School of Cyberspace Security, Xiangtan University, Xiangtan, China
[b] The School of Computer Science and Engineering, Central South University, Changsha, China

ARTICLE INFO

Keywords:
Neural radiance fields
Depth consistent constraint
Novel view synthesis

ABSTRACT

The neural radiance fields (NeRF) for realistic novel view synthesis require camera poses to be pre-acquired by a structure-from-motion (SfM) approach. This two-stage strategy is not convenient to use and degrades the performance because the error in the pose extraction can propagate to the view synthesis. We integrate pose extraction and view synthesis into a jointly optimized process so that they can benefit from each other. For network training, only images are given without pre-known camera poses. The camera poses are obtained by the depth-consistent constraint in which the identical feature in different views has the same world coordinates transformed from the local camera coordinates according to the extracted poses. The depth-consistent constraint is jointly optimized with the pixel color constraint. The poses are represented by a CNN-based deep network, whose input is the related frames. This joint optimization enables NeRF to be aware of the scene's structure, resulting in improved generalization performance. Experiments on three datasets demonstrate the effectiveness of camera pose estimation and novel view synthesis. Code is available at https://github.com/XTU-PR-LAB/SaNerf.

1. Introduction

Volumetric neural rendering methods have been widely used in novel view synthesis and have achieved significant performance. Neural radiance fields (NeRF), introduced by Mildenhall et al. [1], implicitly model a static scene as a continuous five-dimensional (5D) function by training a multi-layer perceptron (MLP), and novel views are synthesized using volume rendering. Although NeRF and its variants have demonstrated unprecedented performance for view synthesis in a range of challenging scenes, most variants require accurate estimation of camera poses in advance. The camera poses are either directly accessible during training or extracted using a structure-from-motion (SfM) approach [2,3]. Accurate camera pose estimation is not a trivial task, and it can significantly impact the subsequent NeRF performance.

To eliminate the heavy dependence on having precise camera pose information, NeRF-- [4] jointly optimized both the camera poses and the parameters of the NeRF network by minimizing the photometric reconstruction error. However, the employed loss focuses on view synthesis, meaning the NeRF network could get stuck in local minima where the optimized camera poses are not optimal, leading to unrealistic synthesized views. Recently, GNeRF [5] attempted to train NeRF for complex scenarios without known camera poses by using generative adversarial networks (GAN); however, these networks work under the constraint that the camera sampling distribution must be close to the ground truth. In contrast, our approach does not assume a certain prior and can be optimized from randomly initialized camera poses.

NeRF is capable of generating high-quality photorealistic novel images of scenes because it relies on a sufficient number of input views to eliminate correspondence ambiguity [6], similar to direct methods in simultaneous localization and mapping (SLAM). For instance, LSD-SLAM [7] reconstructs the structure and appearance of scenes using a photometric consistent constraint. In contrast to LSD-SLAM, NeRF is an indirect method that implicitly constrains the structure of scenes by synthesizing different views. Its performance drops significantly when input views are sparse, as it is prone to finding a degenerate solution to the view reconstruction objective. Some researchers [6,8] have addressed this issue by leveraging the estimated depth as a prior to constrain the network optimization. However, this solution still treats structure reconstruction and view synthesis as two separate processes that cannot benefit from each other. In traditional SfM or SLAM [9], the structure and appearance of scenes are simultaneously reconstructed by optimizing an objective function. Inspired by the joint optimization approach derived from bundle adjustment in classical SfMs, we propose a structure-aware NeRF without a posed camera approach (SaNeRF).

To capitalize on the strengths of structure reconstruction and view synthesis, in this work, we jointly optimize the scene representation and camera poses during the training process. Our approach is to

* Corresponding author.
E-mail addresses: chenshu@xtu.edu.cn (S. Chen), 460086012@qq.com (Y. Zhang), 1253017164@qq.com (Y. Xu), bjzou@csu.edu.cn (B. Zou).

https://doi.org/10.1016/j.patcog.2024.110419
Received 11 July 2023; Received in revised form 31 January 2024; Accepted 4 March 2024
Available online 16 March 2024

incorporate a camera extrinsic parameters optimization solver into the MLP architecture, which implicitly represents the appearance of scenes and allows the solver to participate in guiding the updates of the MLP parameters. Specifically, we first extracted the SIFT pairs in two views. Then, the global 3D coordinates of each SIFT and its match in these views were optimized according to the depth-consistent constraint, ensuring that each SIFT and its correspondence must have the same 3D global coordinates. The depth-consistent constraint enables NeRF to be aware of the scene's structure. The global 3D coordinates of each SIFT were obtained by transforming the camera coordinates into the global coordinate system based on the camera's extrinsic parameters. The camera extrinsic parameters are extracted by a CNN in which the input is the related images.

To sum up, our main contributions include:

(1) Addressing the requirement that camera poses must be pre-known in NeRF by jointly optimizing the scene representation and the camera's extrinsic parameters. Our approach utilizes only randomly initialized poses for complex scenarios.

(2) Enabling NeRF to be aware of the scene's structure through the employed depth-consistent constraint, thereby improving the generalization ability of NeRF.

(3) Jointly optimizing the scene representation and camera pose extraction, freeing NeRF from the inaccurate estimation of camera poses and allowing mutual benefits between the two.

2. Related work

2.1. Scene representations

Scenes are traditionally represented using grids of voxels [10], meshes [11], and point clouds. Conventional signal representations are typically discrete, making them non-differentiable. In contrast, implicit neural representations leverage a 5D continuous function to depict a scene by mapping 3D coordinates to the corresponding values, such as a density value for shape or a pixel for texture. Park et al. [12] successfully learned a continuous Signed Distance Function (SDF) from ground-truth data to represent a class of shapes. This learned SDF offers several advantages, including high-quality shape representation and the ability to complete and interpolate 3D shapes from noisy input. Some researchers [13] have enhanced this approach by training SDFs directly from raw data without relying on 3D supervision, enabling unsupervised learning. Beyond implicit neural representations of geometry, certain methods simultaneously represent both geometry and appearance using a neural network [14]. However, it is worth noting that these techniques often require costly 3D supervision.

To achieve self-supervised implicit neural representation learning, 3D scenes are represented as 3D-structured neural scene representations formulated from 2D images [1,15]. Sitzmann et al. [16] proposed Scene Representation Networks (SRNs), which model scene representation as a continuous function that maps a spatial location to a feature of learned scene representation at that location. Instead of a long short-term memory network in SRNs, Niemeyer et al. [15] employed a five fully-connected ResNet to represent the scene geometry and appearance for easy geometry extraction in the final trained model. NeRF [1] combines neural implicit representations with classical volume rendering for photorealistic novel view synthesis of complex real-world scenes. NeRF works well when the images are from a roughly constant distance; otherwise, the synthesized novel views are prone to be blurred or contain aliasing artifacts. To address this problem, instead of rendering rays, Mip-NeRF [17] models the scene at multiple scales by rendering a series of conical frustums to anti-alias. NeRF has inspired many subsequent works: fast inference [18], deformable [19], and generalization [20]. However, all these approaches require the camera poses to be accurately estimated in advance.

2.2. Structure-aware view synthesis

Recently, depth priors have been incorporated into the optimization process of NeRF. Wei et al. [8] used pre-estimated depth priors to constrain NeRF optimization; the priors were obtained by training a monocular depth network based on sparse SfM estimation. Roessle et al. [6] employed a depth completion network to transform the 3D sparse points obtained by SfM into dense depth maps, then leveraged these dense depth priors to guide the radiance field optimization. DS-NeRF [21] proposed a similar approach, but the approach considered the reliability of estimated depth in the depth constraint. Because depth prior estimation and NeRF are two separate processes in which small camera calibration errors may impede photorealistic synthesis, they cannot benefit from each other. In comparison, our approach integrates sparse 3D structure estimation and view synthesis into joint optimization.

Several approaches have proposed exploiting generative three-dimensional models for structure-aware novel view synthesis [22]. Some works have employed explicit methods that either learn 3D feature representations or create voxelized 3D models [23]. Due to the learned neural projection function, these methods are prone to degrading the view-consistency of synthesized images or creating visible artifacts. In comparison, implicit methods [22] use an MLP network to create transformed images by rotating the extracted latent feature vectors. Schwarz et al. [20] proposed a generative radiance fields model for novel image synthesis by introducing a multi-scale patch-based discriminator; however, it can barely be used in single-object scenes. Niemeyer et al. [22] proposed a compositional 3D scene generative model to achieve controllable image synthesis; the proposed approach can be implemented in multi-object scenes. However, all the methods are only suitable for model learning in static scenes.

2.3. Camera pose estimation

Classical methods use template matching with a specific 3D model to estimate camera poses. However, the employed model is commonly inaccurate in that the performance of object pose estimation is significantly affected. In contrast, SfM [3] is able to simultaneously recover the 3D structure of scenes and camera poses. In SfM, the initial camera poses are estimated in two consecutive steps: (1) features such as SIFT are extracted from each image and the identical features belonging to the consecutive image frames are matched; (2) the camera poses are inferred by five-point or eight-point algorithms, according to the matches. The extracted camera poses are further refined by a bundle adjustment (BA) optimization. Instead of object tracking, some data-driven learning approaches [24] suggest to regress the camera poses directly from raw images based on a large scale of training dataset. However, non-linearity of the rotation space limits the generalization ability of DNN-based method. In contrast, keypoint-based approaches [25] detect the keypoints of objects and use the PnP-RANSAC approach to estimate camera poses. By inverting a trained model [5], NeRF has also been used for camera pose recovery.

3. Preliminary

NeRF is a continuous 5D function mapping a 3D location $\mathbf{x} = (x, y, z)$ and 2D viewing direction (θ, φ) to an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. The continuous function is implicitly represented by a multi-layer perceptron, and the weights of the multi-layer perceptron are optimized to synthesize the input images of a specific scene. Given n training images and the corresponding camera poses, the NeRF is optimized according to a photometric loss as

$$L = \frac{1}{n}\sum_{i=1}^{n}\left\|I_i - \hat{I}_i\right\|_2^2, \tag{1}$$

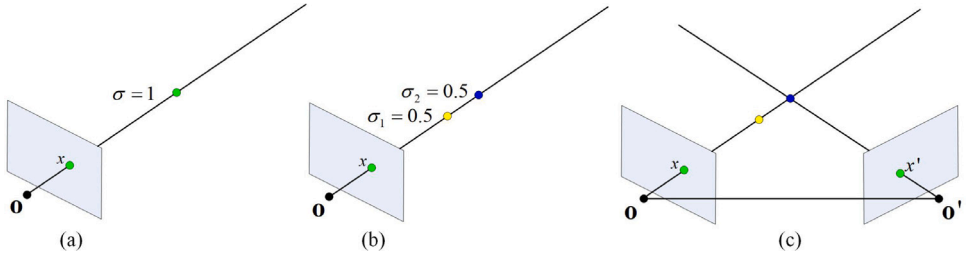

Fig. 1. (a) The correct solution; (b) one incorrect solution; (c) projecting the voxel to another view makes (b) impossible. $\mathbf{o}$ and $\mathbf{o}'$ are the two camera centers, respectively. x and x' are the projections of the voxel projecting onto two views, respectively. σ is the volume density.

where I_i is the ground-truth color of image i and $\hat{I}_i$ is the corresponding synthesized image by volume rendering.

For each pixel of $\hat{I}_i$, casting a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, $\mathbf{o} \in \mathbf{R}^3$, $\mathbf{d} \in S^2$, $t \in [t_n, t_f]$ from the camera center $\mathbf{o}$ through the pixel along direction $\mathbf{d}$, and its color renders as

$$\hat{\mathbf{c}}_\theta(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\,\sigma_\theta(\mathbf{r}(t))\mathbf{c}_\theta(\mathbf{r}(t), \mathbf{d})\,dt, \tag{2}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma_\theta(\mathbf{r}(s))\,ds\right)$, and $\sigma_\theta(\cdot)$ and $\mathbf{c}_\theta(\cdot,\cdot)$ indicate the volume density and color prediction of the radiance field, respectively.

There are two drawbacks in NeRF:

(1) Precomputed camera parameters are required to train a NeRF model. As pointed out by [26] that small camera calibration errors may impede photorealistic synthesis.

(2) NeRF's performance declines dramatically if the input scene is sparse [27]. NeRF tend to well synthesize the input views, however, due to the sparse input, the synthesized novel views may degenerate because NeRF is not biased to learn a structure-consistent model.

We explain the second drawback as follows:

Given an opaque voxel (in the surface) in the scene, the color of this voxel projected onto one view is $p(v)$; $\sigma_\theta(\cdot)$ must satisfy the following constraint, as shown in Fig. 1(a).

$$\begin{aligned} &\mathbf{c}(\mathbf{r}(t)) = p(v) \text{ and } \sigma(\mathbf{r}(t)) = 1, \\ &\text{while } \mathbf{r}(t) \text{ is the location of the voxel,} \\ &\sigma(\mathbf{r}(t)) = 0, \text{ otherwise.} \end{aligned} \tag{3}$$

However, according to (2), there are many solutions with different $\sigma(\mathbf{r}(t))$ and $\mathbf{c}(\mathbf{r}(t))$ that make $\hat{\mathbf{c}}_\theta(\mathbf{r})$ render as $p(v)$. Fig. 1(b) demonstrates one choice in which the pixel value (the green dot) is represented by the mixture of projections of two voxels (yellow dot and blue dot), and this is called shape-radiance ambiguity [26]. Fig. 1(c) shows that introducing more constraints can eliminate this ambiguity to some extent. In this situation, if the opaque voxel is projected to another view, that the photometric loss in the other view makes Fig. 1(b) impossible. Therefore, NeRF requires dense inputs to introduce more constraints to avoid degenerate solutions.

Our approach jointly optimizes the scene representations and camera poses during the training to address the aforementioned drawbacks. The combination strategy enables them in a mutually reinforcing manner.

4. SaNeRF

4.1. Overview

Fig. 2 shows the pipeline of our SaNeRF framework for novel view synthesis without a posed camera. Given a dataset with n images, we first extract SIFT matches between any three images selected from the dataset. Then, an image from the n images is determined as the reference image, and its camera coordinate system is set as the world coordinate system based on how many images it matched and the number of matches. We input any three images (with one being the reference image) into a pose network to estimate the extrinsic parameters of the other two images relative to the reference image. The estimated extrinsic parameters of the other images, along with the extrinsic parameters of the reference image ($\{\mathbf{I}, 0\}$, $\mathbf{I}$ is the identity matrix and 0 is the zero vector), are then input into the NeRF to synthesize their images. In NeRF, the input poses are used to transform the rays in the camera coordinate system into the world coordinate system by

$$\mathbf{d}' = \mathbf{R}\mathbf{d};\ \mathbf{o}' = \mathbf{t}, \tag{4}$$

where $(\mathbf{R}, \mathbf{t})$ are the estimated extrinsic parameters; $\mathbf{d}'$ and $\mathbf{o}'$ are the viewing direction and start point of a ray after transformation, respectively; $\mathbf{d}$ is the viewing direction of the ray before transformation, and the start point of the ray before transformation is $(0, 0, 0)$.

In addition, the 3D coordinates of the SIFT features in each image are estimated according to the volume densities output from the neural radiance fields, and formulated as a weighted sum of all samples volume densities σ_i along the ray, defined as

$$\mathbf{x}_s = \sum_{i=1}^{N_c} w_i\left(\mathbf{o}' + t_i\mathbf{d}'\right),\ w_i = T_i\left(1 - \exp\left(-\sigma_i\delta_i\right)\right), \tag{5}$$

where $\delta_i = t_{i-1} - t_i$ is the distance between adjacent samples.

Besides the photometric reconstruction loss (L_{pm}) introduced in NeRF, positions of matched features loss (L_{3D}) is employed in SaNeRF to optimize the pose network, according to the multi-view geometry in which the identical features in different views have the same 3D world coordinates.

4.2. Pre-processing

For each image in the dataset, we established a set of three images, where one image is assumed to be the reference image, and the other two images were selected from the dataset without replacement. We extracted SIFT features from each image and obtained the matched SIFT correspondences between them. If the number of matched SIFT correspondences was above three, then the other two images were denoted as the matched images for that particular image. By doing this, we were able to determine the number of matched images for each image and the number of matched correspondences.

The reference image was identified as the image with the maximum number of matched images and the maximum number of matched correspondences.

4.3. Pose network

Following [28], we employ fully convolutional architecture to model the pose network. The input to the pose network is the reference image concatenated with the other two images (along the color channels), and the outputs are the relative poses between the reference image and each of the other two images. The network consists of seven stride-2 convolutions. Except for the kernel sizes of the first and the second which are seven and five, respectively, all other convolutions are three. The last convolutional layer is followed by a 1×1 convolution with

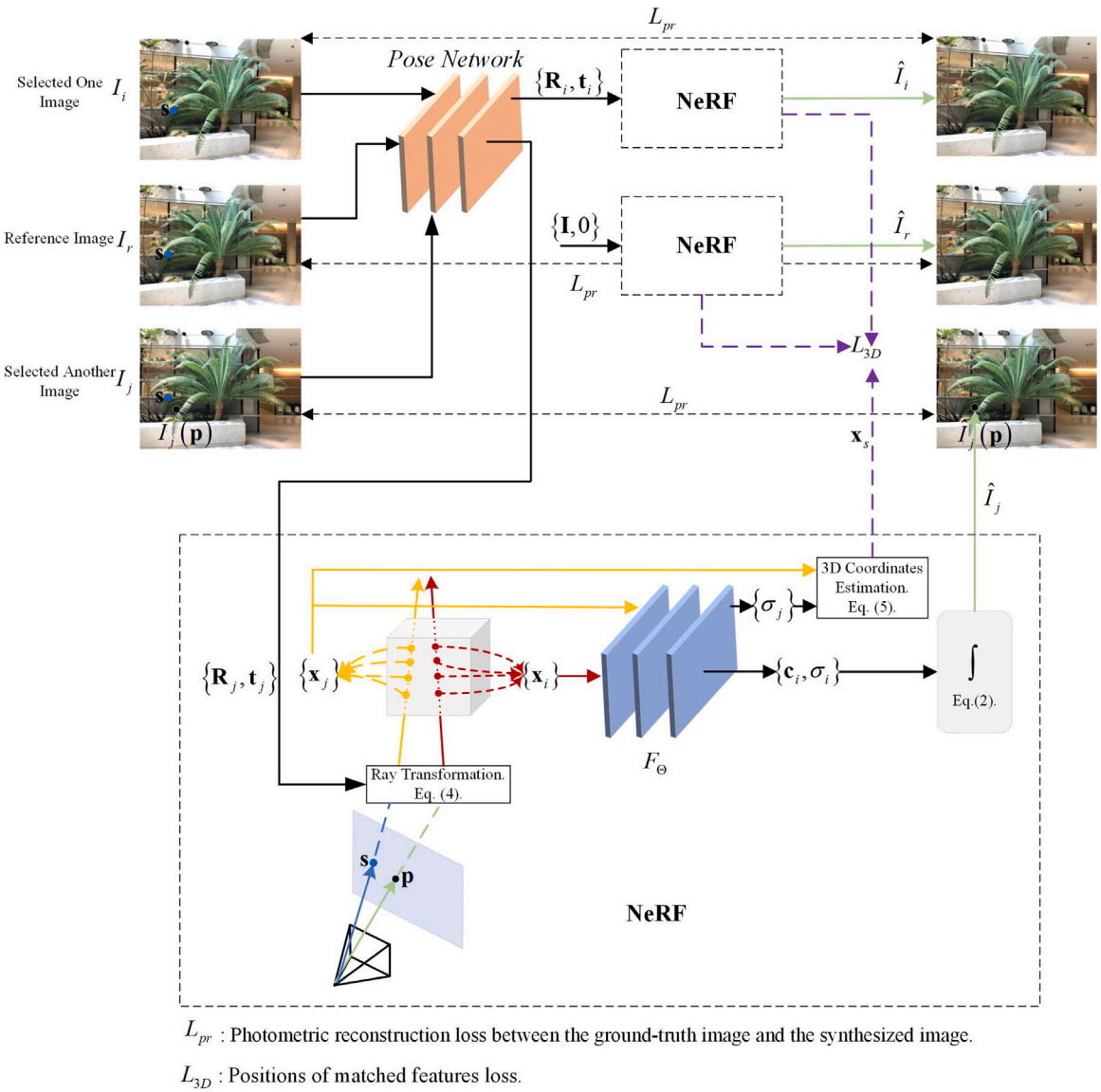

Fig. 2. Overview of SaNeRF. The blue points in the selected images are the matched features of the blue point in the reference image. F_Θ is the neural radiance field which represented by a MLP. **s** represents the location of one matched SIFT correspondence in the image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6×2 output channels (corresponding to three Euler angles and 3D translation for each image). Finally, global average pooling is applied to aggregate predictions at all spatial locations.

4.4. Training loss

We extend the loss defined in (1) to include the photometric reconstruction error of SIFT features, and the final photometric reconstruction loss is defined as

$$L_{pr} = L + \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\mathbf{c}_i^j - \hat{\mathbf{c}}_i^j\right\|_2^2, \tag{6}$$

where $\mathbf{c}_i^j$ is the ground-truth color of the matched SIFT feature j in the image i which is obtained by the bilinear interpolation at the location of the SIFT j from the image i. $\hat{\mathbf{c}}_i^j$ is the corresponding rendered color from NeRF.

At the pre-processing, we extracted SIFT features at each frame and matched them between three images. We denote the 3D coordinates of matched correspondences as $\left\{\left[\mathbf{x}_r^k, \mathbf{x}_i^k, \mathbf{x}_j^k\right] | k = 1, \ldots, m\right\}$, and the positions of matched features loss is defined as

$$L_{3D} = \frac{1}{nml}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l}\left\|\mathbf{x}_r^k - \mathbf{x}_i^k\right\|_2^2 + \left\|\mathbf{x}_r^k - \mathbf{x}_j^k\right\|_2^2 + \left\|\mathbf{x}_i^k - \mathbf{x}_j^k\right\|_2^2, \tag{7}$$

where $\mathbf{x}_r^k$, $\mathbf{x}_i^k$, and $\mathbf{x}_j^k$ are the estimated 3D coordinates of the matched SIFT features k in the reference image and another two images according to (5), respectively.

The total loss is formulated as a combination of the aforementioned losses; each loss is controlled by a factor.

$$L_{total} = \alpha L_{pr} + \beta L_{3D}. \tag{8}$$

5. Experiment

5.1. Experimental details

We used the publicly available Pytorch framework to implement our approach. The performance of our system is evaluated on the LLFF-NeRF [1,30], ScanNet [31] and DTU [32] datasets.

Table 1
Quantitative comparisons for novel view synthesis on LLFF-NeRF dataset.

Methods	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns
PSNR↑								
NeRF [1] ECCV20	**32.70**	25.17	20.92	31.16	20.36	27.40	26.80	27.45
Plenoxels [24] arXiv21	30.22	25.46	21.41	31.09	20.24	27.83	26.48	27.58
TensoRF [13] ECCV22	32.35	25.27	21.30	**31.36**	19.87	**28.60**	**26.97**	**28.14**
BARF [29] ICCV21	31.95	23.79	18.78	29.08	19.45	23.37	22.55	22.78
NeRF-- [4] arXiv21	25.73	21.83	18.73	26.55	16.50	25.34	22.49	24.35
Ours	28.09	**25.95**	**21.87**	27.51	**21.40**	26.98	20.15	26.29
SSIM↑								
NeRF [1] ECCV20	0.948	0.792	0.690	0.881	0.641	0.827	0.880	0.828
Plenoxels [24] arXiv21	0.937	0.832	0.760	0.885	0.687	0.862	0.890	0.857
TensoRF [13] ECCV22	**0.952**	0.814	0.752	**0.897**	0.649	**0.871**	**0.900**	**0.877**
BARF [29] ICCV21	0.940	0.710	0.537	0.823	0.574	0.698	0.767	0.727
NeRF-- [4] arXiv21	0.83	0.62	0.52	0.67	0.38	0.71	0.72	0.63
Ours	0.817	**0.837**	**0.783**	0.806	**0.747**	0.858	0.702	0.857
LPIPS $_{vgg}$↓								
NeRF [1] ECCV20	0.178	0.280	0.316	0.171	0.321	0.219	0.249	0.268
Plenoxels [24] arXiv21	0.192	0.224	0.198	0.180	0.242	0.179	0.238	0.231
TensoRF [13] ECCV22	0.167	0.237	0.217	0.148	0.278	0.169	0.221	0.196
BARF [29] ICCV21	**0.099**	0.311	0.353	0.132	0.291	0.211	**0.206**	0.298
NeRF-- [4] arXiv21	0.44	0.49	0.47	0.44	0.56	0.37	0.44	0.49
Ours	0.281	**0.098**	**0.117**	**0.064**	**0.102**	**0.065**	0.358	**0.091**

Datasets: The LLFF Dataset consists of eight scenes captured with a handheld cellphone, with 20–62 images each. The resolution of each image in the dataset is 4032 × 3024. Due to the limited capacity of NVIDIA RTX 2080Ti, we downsized each image 1/8 scale to 504 × 378 dimensions in pixels, and held out 1/8 of these as the test set for novel view synthesis.

Following the experimental setup in [30], we selected eight scenes in the ScanNet dataset to evaluate our method. In each scene, 40 images were selected to cover a local region, and all images were resized to 648 × 484. Similar to NeRF [1], we held out 1/8 of these as the test set for novel view synthesis.

Similar to the approach proposed by Meng et al. [5], we used four scenes in the DTU dataset to evaluate our method. In each scene, every 8-th image was taken as the test image, the rest images were used for training, and all images were resized to 500 × 400. The scenes used for evaluation were selected based on diversity consideration: scan4 is the real scene with rich texture, scan63 is the synthetic scene, scan48 and scan104 are captured in the real scene and have less texture.

Training details: We employed a similar MLP in NeRF [1] for each of the emitted RGB radiance and volume density predictions. The hierarchical sampling strategy in NeRF [1] was adopted and numbers of sampled points of both coarse sampling and importance sampling were set to 64. We optimized our model with the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ and the learning rate was 5e-4. The loss weights were set as $\alpha = 1.0$ and $\beta = 1.0$. One NVIDIA RTX 2080Ti was used for training and testing and SaNeRF took about 20–60 h to train a single scene of the LLFF dataset.

Evaluation Metrics: We employed two kinds of metrics to evaluate the proposed approach: For the quality of novel view rendering measurement, we used the common metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [33] and Learned Perceptual Image Patch Similarity (LPIPS) [34]; For the accuracy of the optimized camera parameters evaluation, we computed the Absolute Trajectory Error (ATE) [35], which first aligns two sets of pose trajectories globally using a similarity transformation Sim(3) and reports the absolute distance between two translation vectors. Since the ground-truth was not available, we ran COLMAP [3] to obtain the extrinsic parameters of each camera, and used ATE to evaluate the accuracy by computing the difference between our optimized camera and the estimations from COLMAP.

5.2. Comparison with state-of-the-art

5.2.1. Novel view synthesis comparison

For evaluation, we need to estimate the camera poses of the test view images. We trained our approach on each scene two times. The first time, we trained SaNeRF on all images in the scene, in order to estimate the cameras' poses in the test images; The second time, we trained SaNeRF only on the training images to evaluate the quality of the novel view rendering from the cameras in the test images in which the poses of the test images were obtained from the first training.

Results on LLFF-NeRF dataset.

We compared SaNeRF to NeRF [1] as well as recent and four recently proposed works: Plenoxels [24], BARF [29], NeRF-- [4] and TensoRF [13]. The quantitative comparisons for novel view synthesis are shown in Table 1, and the visual results are illustrated in Fig. 3. Some detail comparison results are presented in Fig. 4. Our method achieved the best performance measured by PSNR and SSIM in scenes (*fern*, *leaves* and *orchids*) because the rich texture in these scenes can guarantee enough reliable keypoints for precise camera poses estimation. As compared to BARF [29], which trains NeRF from imperfect camera poses, our approach outperformed it in five scenes. As compared to NeRF-- without known camera parameters, our approach outperformed it in almost all of the scenes except for *scene T-Rex*. Our approach achieved the best performance in six scenes on LPIPS because LPIPS is free from the errors that are caused by the inaccurately estimated camera poses. We also noticed that our approach yields inferior performance in some metrics, attributed to the presence of rich textures in the LLFF-NeRF dataset-essential for NeRF and its derivatives, but not fully adhered to by our methodology.

Results on ScanNet dataset.

Table 2 shows the quantitative comparisons for novel view synthesis on the ScanNet dataset. The qualitative results are illustrated in Fig. 5. Our approach achieved the best performance in scenes (*scene 0079*, *scene 0158* and *scene 0553*). With other challenging scenes that did not have enough keypoints for pose estimation, our approach failed to synthesis good results. Our approach outperformed the original NeRF [1] in five scenes which benefitted from the joint optimization.

Results on DTU dataset.

We present the quantitative comparisons for novel view synthesis on the DTU dataset in Table 3, and the visual results are shown in Fig. 6. From Table 3, we notice that our approach achieved the best

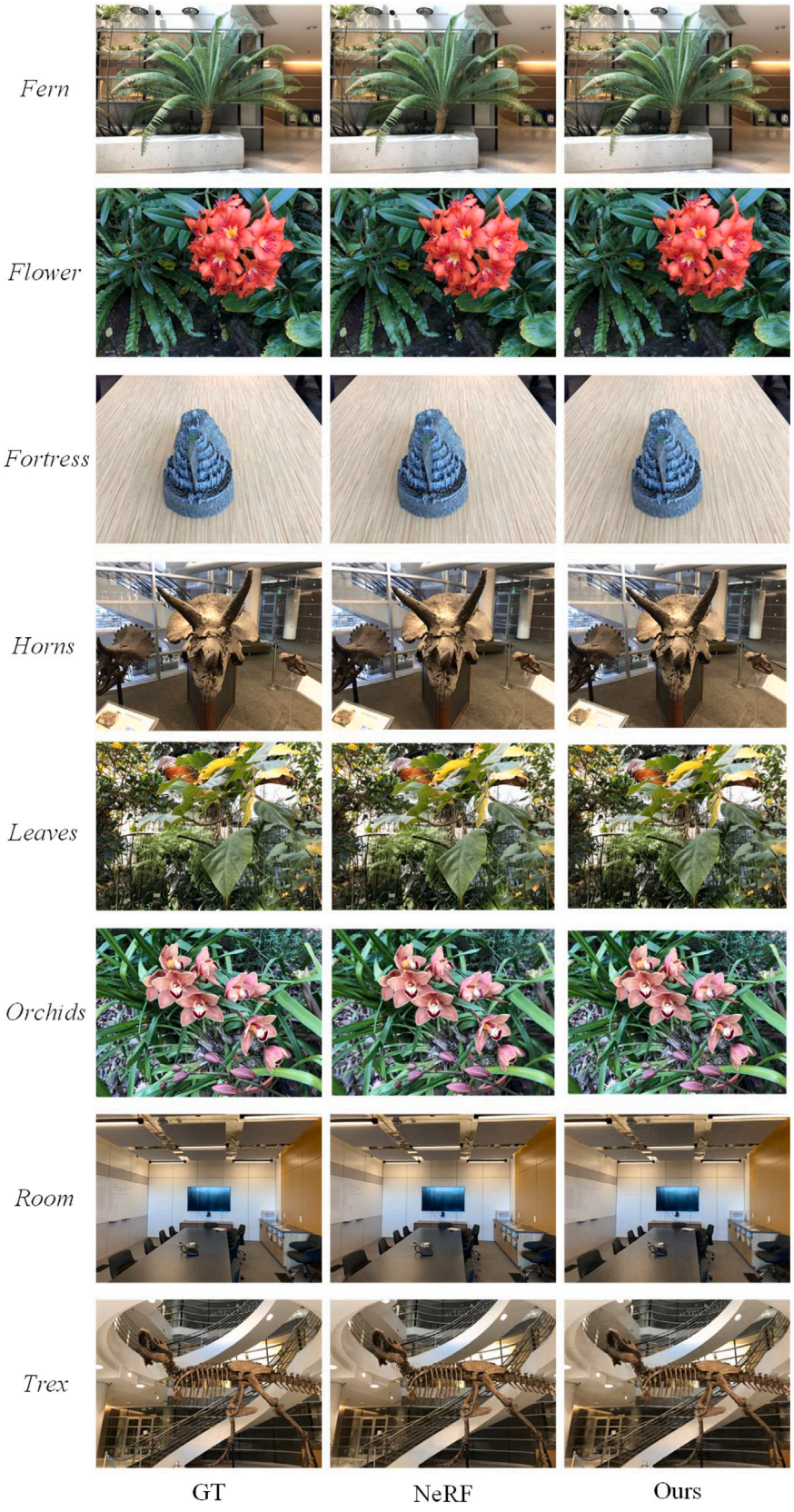

Fig. 3. Qualitative comparison between our SaNeRF with unknown cameras and other approaches on the LLFF-NeRF dataset.

Fig. 4. Some detail comparison results from Fig. 3.

Table 2
Quantitative comparisons for novel view synthesis on ScanNet dataset.

Methods	Scene 0000	Scene 0079	Scene 0158	Scene 0316	Scene 0521	Scene 0553	Scene 0616	Scene 0653
PSNR↑								
NSVF [36] NeurIPS20	**23.36**	26.88	31.98	**22.29**	27.73	31.15	15.71	28.95
SVS [37] CVPR21	21.39	25.18	29.43	20.63	27.97	30.95	**21.38**	27.91
NeRF [1] ECCV20	18.75	25.48	29.19	17.09	24.41	30.76	15.76	30.89
NerfingMVS [8] ICCV21	22.10	27.27	30.55	20.88	**28.07**	32.56	18.07	**31.43**
Ours	20.22	**28.65**	**33.12**	16.06	25.89	**33.69**	14.56	28.55
SSIM↑								
NSVF [36] NeurIPS20	0.823	0.887	0.951	0.917	0.892	0.947	0.704	0.929
SVS [37] CVPR21	**0.914**	0.923	0.953	**0.941**	**0.924**	0.968	**0.899**	0.965
NeRF [1] ECCV20	0.751	0.896	0.928	0.828	0.871	0.950	0.699	0.953
NerfingMVS [8] ICCV21	0.880	0.916	0.948	0.899	0.901	0.965	0.748	**0.964**
Ours	0.638	**0.917**	**0.955**	0.716	0.783	**0.969**	0.598	0.899

performance in scene scan4. Since no enough keypoint matches can extracted in scan 48, scan 63, and scan 104 to guarantee a precise pose, our approach outperformed the approach proposed by Meng et al. [5]. NeRF [1] fails in almost all scenes except scan4 because it is optimized based on photometric reconstruction constraint and cannot synthesize good novel views in the scenes with big photometric ambiguity. In contrast, our approach can eliminate the ambiguity to some extent because it considerate the 3D structure of scenes during training and achieved better performance than NeRF.

5.2.2. Extrinsic parameters comparison

The performance of the camera pose estimation is evaluated on the LLFF-NeRF dataset. Since the ground-truth camera poses are not available, we first ran COLMAP [3] on all images to obtain camera poses as references, then present the difference between the predicted camera poses from SaNeRF and the corresponding ones from COLMAP on the training set.

Table 4 shows the ATE comparison results on the translation between our approach and other approaches. We aligned our optimized camera trajectories by ATE, and compared them with the estimated trajectories from COLMAP in Fig. 7. From Table 4 and Fig. 7, we notice that the camera parameters estimated from the proposed approach are very close to the estimations from COLMAP, demonstrating the advantages of our joint optimization strategy.

To better understand the optimization process, we visualize the estimated camera parameters at some epochs during training for the scene *flower* in Fig. 8. The camera parameters are randomly initialized at the beginning of training, and the estimations are converged after about 100,000 epochs.

5.2.3. Novel view synthesis with sparse inputs

To assess the efficacy of our proposed joint optimization framework for pose extraction and view synthesis, we conducted experiments on the ScanNet dataset utilizing sparse inputs, the input views were selected from the neighborhood of the mid image in the train set. The quantitative comparisons are presented in Table 5, with the values representing the average estimated results across all scenes in the ScanNet dataset. Analysis of Table 5 reveals a significant performance advantage for our approach over other state-of-the-art methods in PSNR. This superiority is attributed to the presence of contaminated images in the

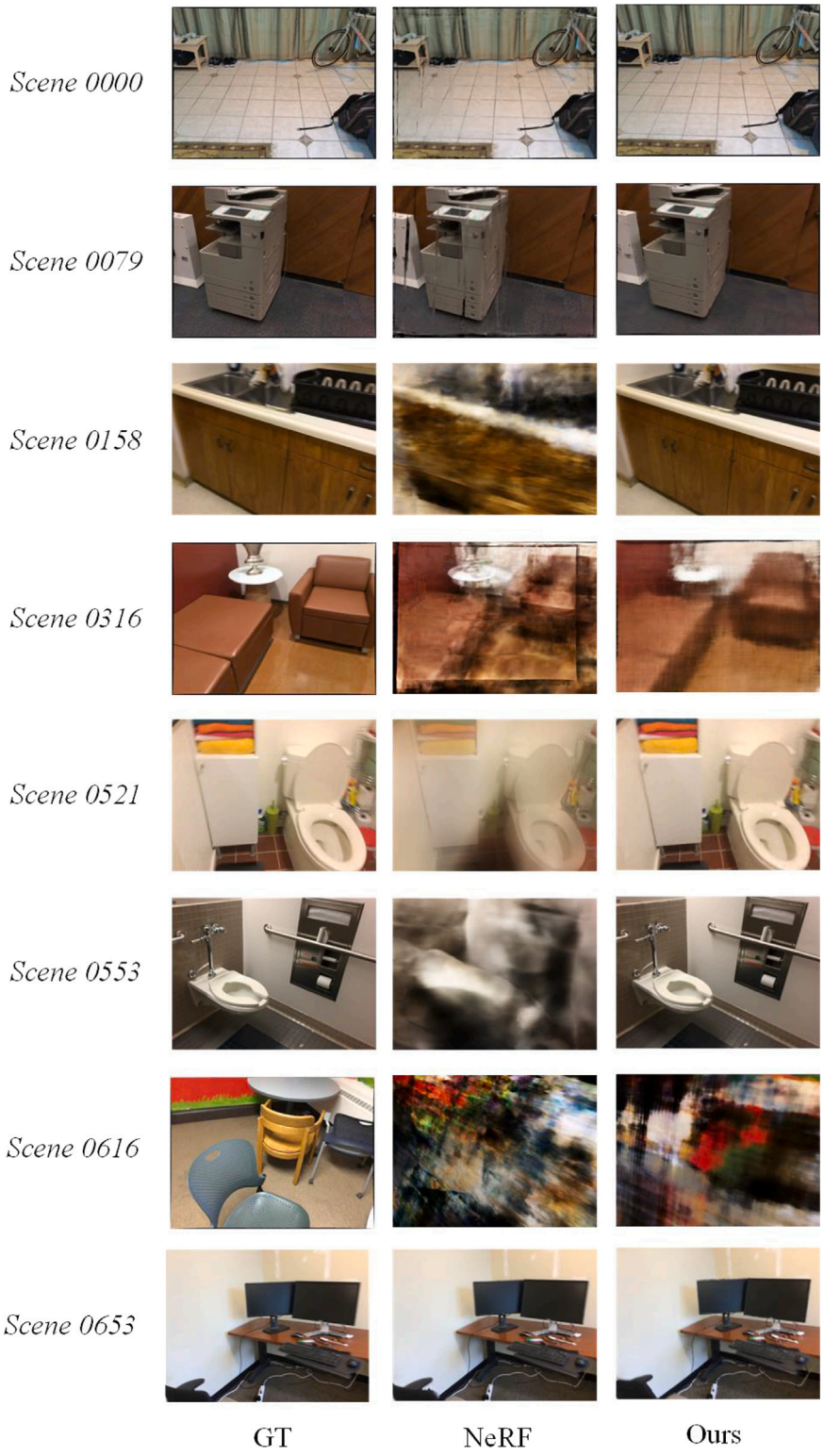

Fig. 5. Qualitative comparison between our SaNeRF with unknown cameras and other approaches on the ScanNet dataset.

Table 3
Quantitative comparisons for novel view synthesis on DTU dataset.

Methods	Scan4	Scan48	Scan63	Scan104
PSNR↑				
NeRF [1] ECCV20	22.05	6.718	27.80	10.52
GNeRF [5] ICCV21	22.88	23.25	25.11	21.40
Ours	27.50	21.02	22.65	13.95
SSIM↑				
NeRF [1] ECCV20	0.69	0.52	0.90	0.48
GNeRF [5] ICCV21	0.82	0.87	0.90	0.76
Ours	0.76	0.78	0.87	0.45
LPIPS $_{vgg}$↓				
NeRF [1] ECCV20	0.32	0.65	0.21	0.60
GNeRF [5] ICCV21	0.37	0.21	0.29	0.44
Ours	0.25	0.34	0.31	0.66

Table 4
Quantitative evaluation of our optimized camera translations on LLFF-NeRF dataset measured by ATE.

Methods	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns
BARF [29] ICCV21	0.270	0.192	0.249	0.364	0.404	0.224	0.720	0.222
NeRF-- [4] arXiv21	**0.013**	0.007	0.006	0.041	0.018	0.011	**0.013**	0.015
Ours	0.019581	**0.005418**	**0.001678**	**0.002514**	**0.001096**	**0.000788**	0.069863	**0.001357**

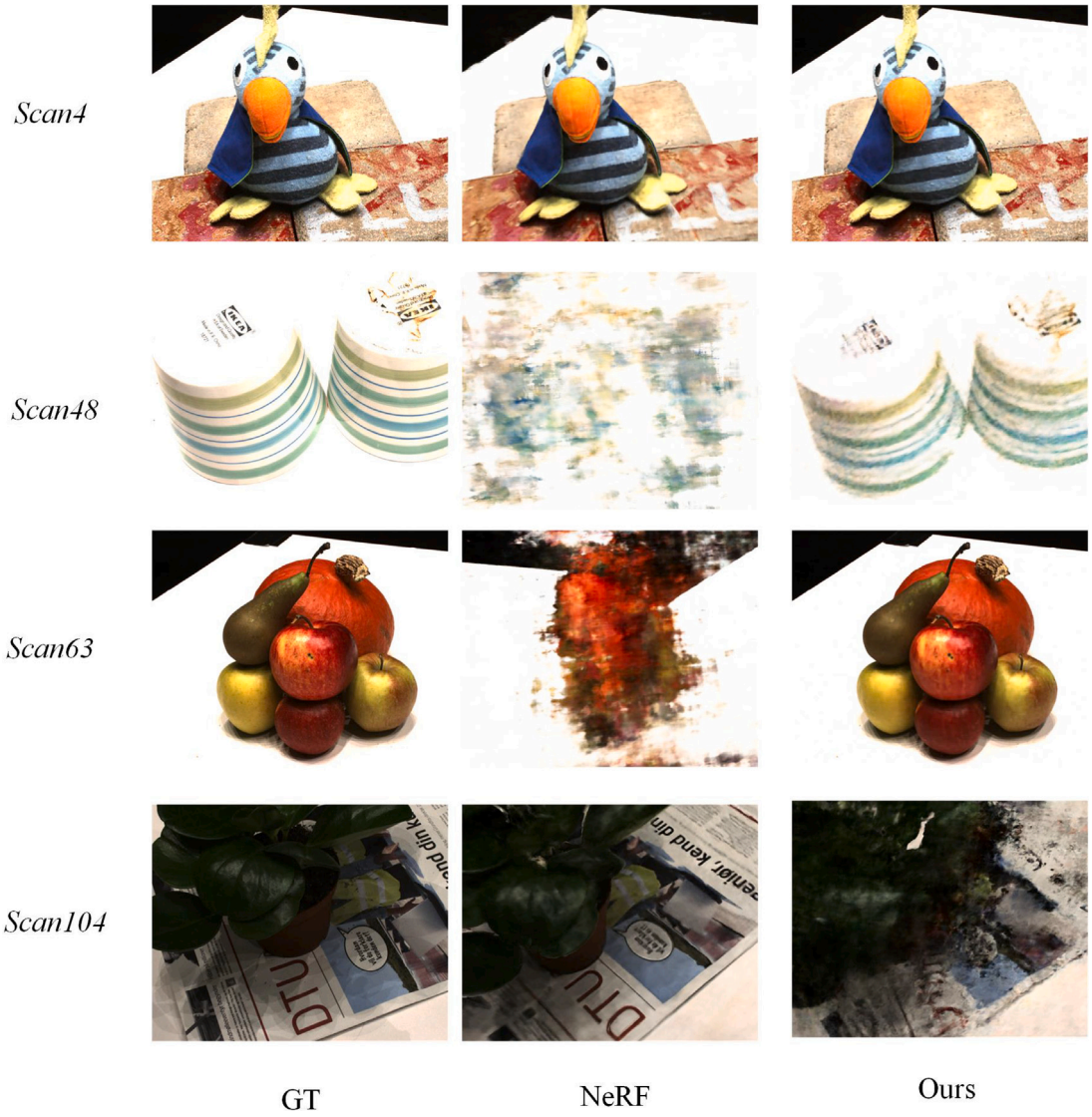

Fig. 6. Qualitative comparison between our SaNeRF with unknown cameras and other approaches on the DTU dataset.

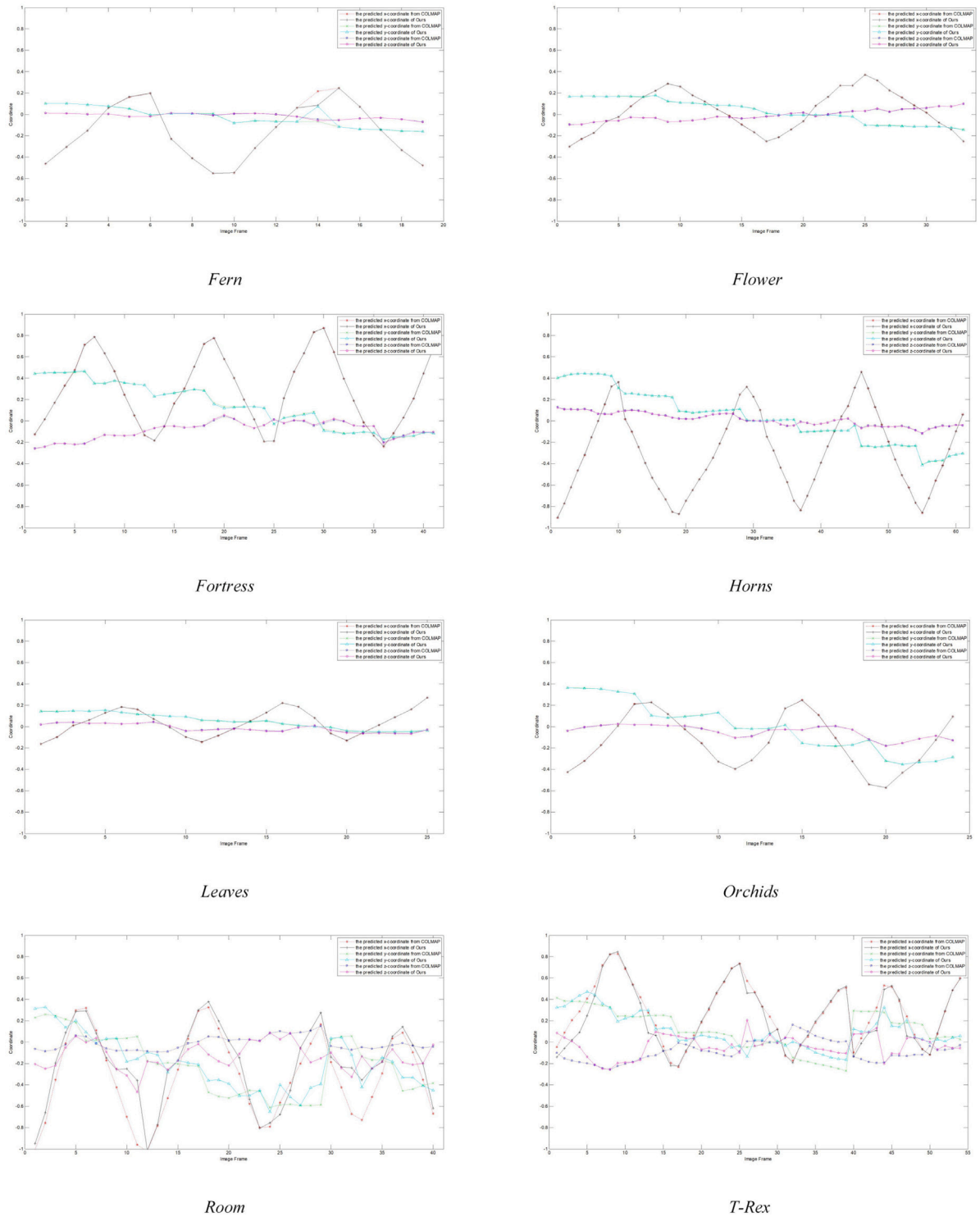

Fig. 7. Comparison of camera trajectories between the optimized translations and the ones estimated from COLMAP.

ScanNet dataset, resulting in unexpected black color regions within the images. The impact of these contaminated images manifests in two key outcomes: (1) SfM approaches struggle to accurately estimate camera poses, thereby negatively affecting the performance of NeRF and its derivatives; (2) the contaminated regions prove unsuitable for the rendering loss in NeRF, compromising the fidelity of the synthesized results.

In contrast, our approach is not reliant on accurately estimated camera poses, thus avoiding the performance pitfalls associated with SfM. Furthermore, the employed loss based on the positions of matched features remains unaffected by contaminated images, mitigating the adverse effects of rendering loss in NeRF. Qualitative comparisons, depicted in Fig. 9 using synthesized results from the third image in the evaluation set, illustrate that our approach can precisely synthesize the novel views across all scenes.

5.2.4. Ablation study

In Table 6, we present the model size and Floating Point Operations per Second (FLOPs) for both our proposed model and NeRF. This analysis aims to assess the performance improvement relative to the

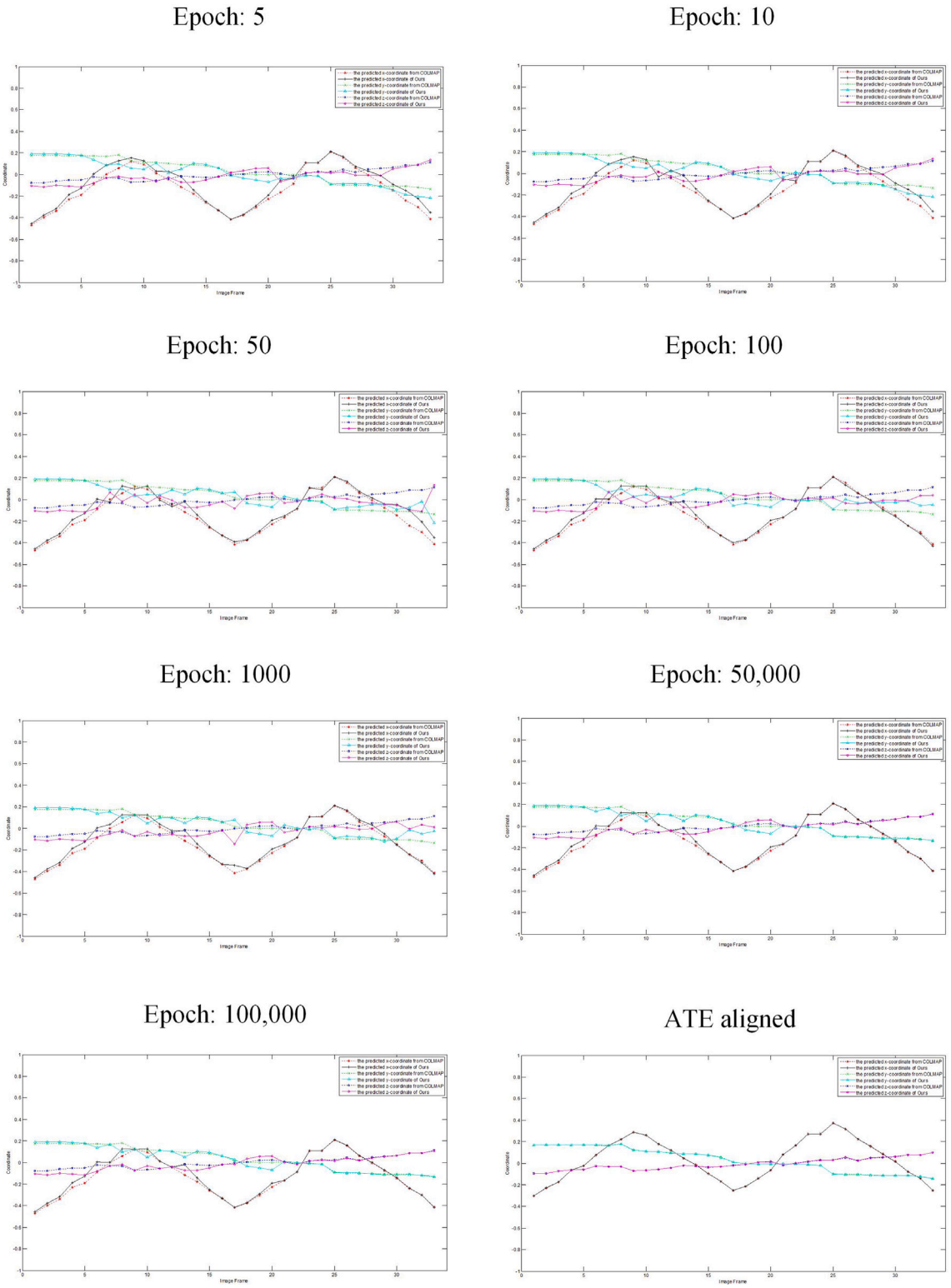

Fig. 8. History of camera pose optimization during training. As each epoch goes by, our optimized camera translations gradually converge towards COLMAP estimations.

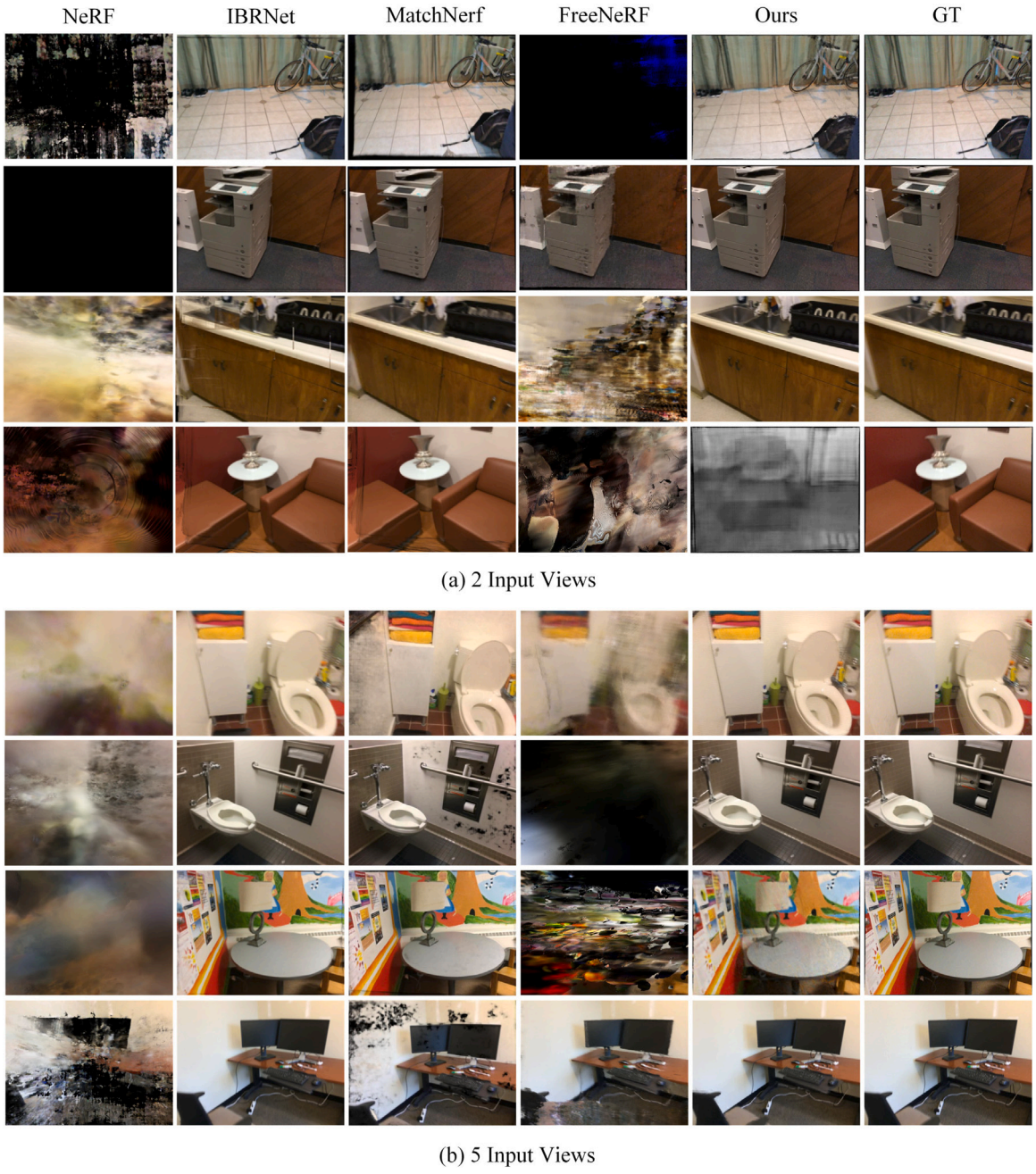

Fig. 9. Qualitative comparison between our SfMNeRF and other approaches with sparse inputs on the ScanNet dataset.

Table 5
Quantitative comparisons for novel-view synthesis with spare inputs on ScanNet dataset. Best results shown in bold.

Method	PSNR↑		SSIM↑		LPIPS $_{vgg}$↓	
	2-view	5-view	2-view	5-view	2-view	5-view
NeRF [1] ECCV20	11.0	12.6	0.425	0.528	0.711	0.693
IBRNet [38] CVPR21	27.0	17.1	**0.880**	**0.885**	**0.175**	**0.171**
MatchNerf [39] arXiv23	**29.3**	27.3	0.831	0.831	0.261	0.270
FreeNeRF [40] CVPR23	12.6	14.2	0.377	0.418	0.589	0.543
Ours	**29.3**	**28.4**	0.791	0.791	0.385	0.373

associated increase in computational cost. As depicted in Table 6, our model exhibits a larger size compared to NeRF. However, the increment in FLOPs is relatively marginal. This discrepancy arises from the fact that the MLP in NeRF undergoes multiple calculations, whereas the PoseNet in our methodology is computed only once.

6. Conclusion and future work

Our approach allows for novel view synthesis without posed camera. The joint optimization improves the accuracy of both view synthesis and camera pose estimation by taking advantage of the merits of each other. Our approach does not acquire to estimate the camera poses by a likely erroneous SfM in advance. The experimental results show that SaNeRF can effectively estimate the camera poses and learn neural implicit scene representations at the same time. Particularly,

Table 6
The model size and FLOPs of ours and NeRF.

Methods	Model size (# Params)	GFLOPs
NeRF [1] ECCV20	1.97428M	98.146712
Ours	3.56495M	99.3008697

our approach excels in synthesizing accurate novel views from sparse inputs. However, our SaNeRF heavily relies on the SIFT matches; therefore, it is not suitable for novel view synthesis in the scenes with many low-textured areas, just like the synthetic dataset in [1]. On the other hand, as the original NeRF, SaNeRF suffers from the scenes with repeated structures which are likely to be caused by photometric ambiguity. For instance, for the novel view synthesis of *scene 0316* from the ScanNet dataset, SaNeRF performs better than NeRF but the joint optimization struggles to converge. This is likely to be caused by the photometric ambiguity from the repeated structures. In the future, we plan on integrating the knowledge proposed in SfM/SLAM into SaNeRF to achieve robust novel view synthesis and self-supervised dense three-dimensional reconstruction.

CRediT authorship contribution statement

Shu Chen: Conceptualization, Methodology. **Yang Zhang:** Data curation, Writing – original draft. **Yaxin Xu:** Software, Validation. **Beiji Zou:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research is supported in part by the Research Foundation of Education Bureau of Hunan Province (No. 22A0124) and National Key R&D Program of China (No. 2018AAA0102102).

References

[1] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis, in: Proceedings of the European Conference on Computer Vision, ECCV, 2020, pp. 405–421.
[2] Y. Chen, S. Shen, Y. Chen, G. Wang, Graph-based parallel large scale structure from motion, Pattern Recognit. 107 (2020) 107537, 1–11.
[3] J.L. Schonberger, J. Frahm, Structure from motion revisited, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 4104–4113.
[4] Z. Wang, S. Wu, W. Xie, M. Chen, V.A. Prisacariu, Nerf–: Neural radiance fields without known camera parameters, 2021, arXiv preprint arXiv:2102.07064.
[5] Q. Meng, A. Chen, H. Luo, M. Wu, H. Su, L. Xu, X. He, J. Yu, GNeRF: GAN-based neural radiance field without posed camera, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6331–6341.
[6] B. Roessle, J.T. Barron, B. Mildenhall, P.P. Srinivasan, M. Niesner, Dense depth priors for neural radiance fields from sparse input views, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12882–12891.
[7] J. Engel, T. Schps, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, in: Proceedings of the European Conference on Computer Vision, ECCV, 2014, pp. 834–849.
[8] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, J. Zhou, NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5590–5599.
[9] M. You, C. Luo, H. Zhou, S. Zhu, Dynamic dense CRF inference for video segmentation and semantic SLAM, Pattern Recognit. 133 (2023) 109023:1–17.
[10] G. Windreich, N. Kiryati, G. Lohmann, Voxel-based surface area estimation: From theory to practice, Pattern Recognit. 36 (2003) 2531–2541.
[11] E.M. Thompson, S. Biasotti, Description and retrieval of geometric patterns on surface meshes using an edge-based LBP approach, Pattern Recognit. 82 (2018) 1–15.
[12] J.J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, Deepsdf: Learning continuous signed distance functions for shape representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 165–174.
[13] A. Chen, Z. Xu, A. Geiger, J. Yu, H. Su, TensoRF: Tensorial radiance fields, in: Proceedings of the European Conference on Computer Vision, ECCV, 2020, pp. 333–350.
[14] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, A. Geiger, Texture fields: Learning texture representations in function space, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4530–4539.
[15] M. Niemeyer, L. Mescheder, M. Oechsle, A. Geiger, Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4455–4465.
[16] V. Sitzmann, M. Zollhöfer, G. Wetzstein, Scene representation networks: Continuous 3D-structure-aware neural scene representations, in: Proceedings of Advances in Neural Information Processing Systems, 2019, pp. 1121–1132.
[17] J.T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P.P. Srinivasan, Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5835–5844.
[18] D.B. Lindell, J.N.P. Martel, G. Wetzstein, AutoInt: Automatic integration for fast neural volume rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14551–14560.
[19] A. Pumarola, E. Corona, G. Pons-Moll, F. Moreno-Noguer, D-NeRF: Neural radiance fields for dynamic scenes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10313–10322.
[20] K. Schwarz, Y. Liao, M. Niemeyer, A. Geiger, GRAF: Generative radiance fields for 3D-aware image synthesis, in: Proceedings of Advances in Neural Information Processing Systems, 2020, pp. 20154–20166.
[21] K. Deng, A. Liu, J. Zhu, D. Ramanan, Depth-supervised NeRF: Fewer views and faster training for free, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12872–12881.
[22] M. Niemeyer, A. Geiger, Giraffe: Representing scenes as compositional generative neural feature fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11448–11459.
[23] P. Henzler, N.J. Mitra, T. Ritschel, Escaping plato's cave: 3d shape from adversarial rendering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9983–9992.
[24] A. Yu, S. Fridovich-Keil, M. Tancik, Q. Chen, B. Recht, A. Kanazawa, Plenoxels: Radiance fields without neural networks, 2021, arXiv preprint arXiv:2112.05131.
[25] X. Liu, R. Jonschkowski, A. Angelova, K. Konolige, Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11602–11610.
[26] K. Zhang, G. Riegler, N. Snavely, V. Koltun, NeRF++: Analyzing and improving neural radiance fields, 2020, arXiv preprint arXiv:2010.07492.
[27] M. Niemeyer, J.T. Barron, B. Mildenhall, M.S.M. Sajjadi, A. Geiger, N. Radwan, RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5470–5480.
[28] S. Chen, Z. Pu, X. Fan, B. Zou, Fixing defect of photometric loss for self-supervised monocular depth estimation, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2022) 2043–2055.
[29] C. Lin, W. Ma, A. Torralba, S. Lucey, BARF: Bundle-adjusting neural radiance fields, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5721–5731.
[30] B. Mildenhall, P.P. Srinivasan, R. Ortiz-Cayon, N.K. Kalantari, R. Ramamoorthi, R. Ng, A. Kar, Local light field fusion: Practical view synthesis with prescriptive sampling guidelines, ACM Trans. Graph. 38 (4) (2019) 1–14.
[31] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, M. Niesner, Scannet: Richly-annotated 3d reconstructions of indoor scenes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 5828–5839.
[32] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, H. Aanas, Large scale multi-view stereopsis evaluation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2014, pp. 406–413.
[33] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, 'Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[34] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.
[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-d SLAM systems, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 573–580.
[36] L. Liu, J. Gu, K. Lin, T. Chua, C. Theobalt, Neural sparse voxel fields, in: Proceedings of Advances in Neural Information Processing Systems, 2020, pp. 15651–15663.
[37] G. Riegler, V. Koltun, Stable view synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12211–12220.
[38] Q. Wang, Z. Wang, K. Genova, P. Srinivasan, H. Zhou, J.T. Barron, R. Martin-Brualla, N. Snavely, T. Funkhouser, IBRNet: Learning multi-view image-based rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4688–4697.
[39] Y. Chen, H. Xu, Q. Wu, C. Zheng, T. Cham, J. Cai, Explicit correspondence matching for generalizable neural radiance fields, 2023, arXiv preprint arXiv: 2304.12294.
[40] J. Yang, M. Pavone, Y. Wang, FreeNeRF: Improving few-shot neural rendering with free frequency regularization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 8254–8263.

Shu Chen received the Ph.D. degree from Central South University, China. He is currently an Associate Professor with the School of Computer Science, Xiangtan University, China, and the School of Cyberspace Security, Xiangtan University. His research interests include computer vision, human motion tracking, and 3D scene reconstruction.

Yang Zhang is currently pursuing the M.S. degree in computer application technology with Xiangtan University, China. His research interests include 3D depth reconstruction and scene reconstruction.

Yaxin Xu received the M.S. degree in computer application technology with Xiangtan University, China. Her research interests include 3D depth reconstruction and scene reconstruction.

Beiji Zou received the B.S. degree in computer science from Zhejiang University, China, in 1982, the M.S. degree from Tsinghua University, specializing in CAD and computer graphics, in 1984, and the Ph.D. degree in control theory and control engineering from Hunan University, in 2001. He is currently a Professor with the School of Computer Science and Engineering, Central South University, China. His research interests include computer graphics, CAD technology, and image processing.