RoGUENeRF: A Robust Geometry-Consistent Universal Enhancer for NeRF

Sibi Catley-Chandar[1,2], Richard Shaw[1],
Gregory Slabaugh[2], and Eduardo Pérez-Pellitero[1]

[1] Huawei Noah's Ark Lab
[2] Queen Mary University of London

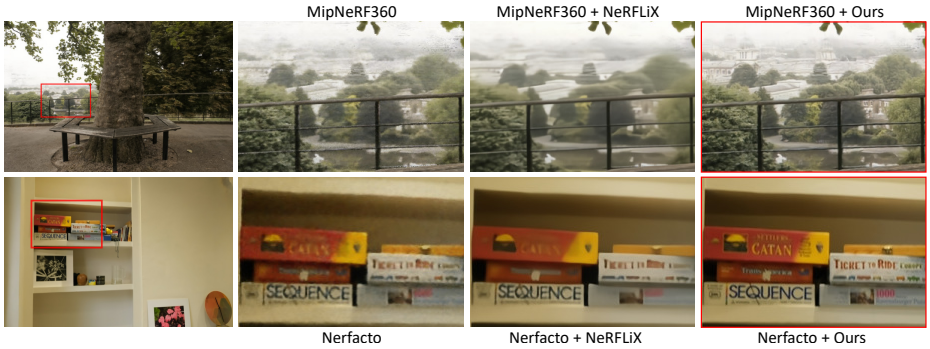

Fig. 1: Novel views from the MipNeRF360 dataset [3]. RoGUENeRF achieves noticeable qualitative improvements over state-of-the-art baselines and NeRF enhancers, especially in high-frequency regions such as trees, buildings and text.

Abstract. Recent advances in neural rendering have enabled highly photorealistic 3D scene reconstruction and novel view synthesis. Despite this progress, current state-of-the-art methods struggle to reconstruct high frequency detail, due to factors such as a low-frequency bias of radiance fields and inaccurate camera calibration. One approach to mitigate this issue is to enhance images post-rendering. 2D enhancers can be pre-trained to recover some detail but are agnostic to scene geometry and do not easily generalize to new distributions of image degradation. Conversely, existing 3D enhancers are able to transfer detail from nearby training images in a generalizable manner, but suffer from inaccurate camera calibration and can propagate errors from the geometry into rendered images. We propose a neural rendering enhancer, RoGUENeRF, which exploits the best of both paradigms. Our method is pre-trained to learn a general enhancer while also leveraging information from nearby training images via robust 3D alignment and geometry-aware fusion. Our approach restores high-frequency textures while maintaining geometric consistency and is also robust to inaccurate camera calibration. We show that RoGUENeRF substantially enhances the rendering quality

of a wide range of neural rendering baselines, e.g. improving the PSNR of MipNeRF360 by 0.63dB and Nerfacto by 1.34dB on the real world 360v2 dataset. Project page: `https://sib1.github.io/projects/roguenerf/`

1 Introduction

The seminal work of Mildenhall *et al.* [23] introduced an effective methodology to render highly photorealistic novel views of 3D scenes by means of Neural Radiance Fields (NeRFs). Given a set of posed multi-view images, NeRFs learn complex view-dependent effects via a learnable multilayer perceptron (MLP) which models the 3D radiance field of the scene, thanks in part to the input domain parameterization (3D coordinates + 2D viewing direction) and the direct pixel-wise photometric loss. The NeRF paradigm has been very popular in recent years [53], with active research in the field proposing new functionalities [27, 30, 31, 48], applications [10, 21, 24] and also tackling some of the open challenges present in [23]. A key aspect of subsequent literature is the successive improvement of the rendering fidelity, *i.e.* as measured by the Peak Signal-to-Noise Ratio (PSNR), producing higher quality rendered novel views [3, 4].

Nonetheless, an underlying challenge of these approaches is the shape-radiance ambiguity [57], *i.e.* training images can be explained with high accuracy by introducing inaccurate geometry, resulting in poor generalization outside of the training views. This is particularly problematic when the geometry around high-frequency details, such as textures, can not be resolved or disambiguated properly with the number of training views. The optimization process will generally either introduce inaccurate geometry with view-dependent radiance values overfitting each training view, *e.g.* view-dependent floater artifacts, or else converge to a mean radiance value, which then results in blurred renderings. Similarly, inaccurate camera calibration or missing lens distortion models also lead to blurred results and thus lack of fidelity in the high-frequency spectrum, mostly due to pixel and subpixel shifts among camera views. Further, analysis by Tancik *et al.* [43] describes a low-frequency bias of the standard MLP set-up.

In addition to these issues, the practical nature of data capture and camera pose estimation can introduce further error. For pseudo-static 3D scenes, small variations in the environment can occur during capture such as changes in lighting and small movements within the scene (e.g. foliage), which violate the 3D-consistent static scene assumption of NeRFs. This can also negatively affect the performance of camera pose estimation via COLMAP [37], which although mostly reliable is not infallible [12, 33], even when using carefully captured data.

In this work, we present RoGUENeRF, a NeRF enhancer which is designed to improve the image quality of NeRF renderings while maintaining geometric consistency and being robust to inaccurate camera calibration. Firstly, we propose a novel combined 3D + 2D alignment and refinement mechanism which accurately finds correspondences between images from different camera viewpoints, even when one input is severely degraded, and can also compensate for inaccurate estimates of scene geometry and camera poses. Secondly, we propose a novel

geometry-aware spatial attention module which regulates misaligned regions based on both camera distance and pixel-wise differences. Lastly, we propose a pre-training and fine-tuning strategy which learns a general geometry-consistent enhancement function that transfers well (*i.e.* fine-tunes in under 60 minutes per scene) to different distributions of rendering degradations. We thoroughly evaluate our proposed approach on six different NeRF baselines across real world bounded and unbounded scenes from three datasets: LLFF [22], DTU [11] and 360v2 [3]. We show consistent improvements in PSNR, SSIM and LPIPS over every baseline and qualitatively demonstrate substantial improvements in image quality over baselines and state-of-the-art NeRF enhancers.

2 Related Work

2.1 High Fidelity NeRF

Since the seminal work of Mildenhall *et al.* [23], there have been several works which aim to improve the fidelity of NeRF-based models. Some methods [12, 34] incorporate additional processing layers after the NeRF model, coupled with image quality specific loss functions. Roessle *et al.* [34] proposed GANeRF, using a 2D conditional generator trained adversarially to refine the rendered output. To maintain view-consistency, a discriminator is trained end-to-end together with the underlying NeRF model, requiring computationally expensive patch-based training. Combined with the time required to train the generator, this significantly increases optimization time per scene from 15 minutes for the underlying NeRF model to 58 hours. AlignNeRF [12] also incorporates additional processing layers trained end-to-end with the NeRF model by introducing a shallow convolutional network coupled with an alignment aware loss and relies on the shallowness of the enhancement network to maintain view consistency. Some methods tackle the problem of NeRF super-resolution [9, 49] by using relevant patches from a HR reference frame. Other approaches attempt to improve the underlying differentiable rendering algorithm directly [2, 3, 47, 54] to better tackle anti-aliasing effects, unbounded scenes or reflections and can better model the characteristics of the scene. A further approach is to jointly optimize camera poses together with the NeRF to reduce geometric errors from incorrect poses [5, 17, 20, 28, 45].

Another popular line of work has been to address the slow training and inference time of NeRF by changing from an MLP to a faster representation, e.g. voxels [36], SDFs [46], MPIs [41, 42], hash encodings [26, 44, 50], tensor decomposition [7], octrees [56], reducing training time per scene to as fast as a few seconds. ZipNeRF [4] marries the advantages of both high-fidelity and fast-training by combining voxel-grid data structures with cone-based rendering. Dynamic scenes captured as volumetric videos also pose a significant challenge for many existing neural rendering models which otherwise perform well on static fixed scenes. Many methods model the additional time dimension [8, 10, 16, 18, 25, 31, 32, 39] and are able to render short free viewpoint videos effectively but at the cost of lower image quality and temporal consistency. Despite huge progress in improving the fidelity of NeRFs, these models remain

constrained by errors in training data and camera pose estimation leading to rendering artifacts in real world scenes.

2.2 NeRF Enhancers

Another approach to improving NeRF fidelity is to apply a NeRF-specific enhancer as a post-rendering step, which assumes the NeRF has already been trained and does not backpropagate gradients to the underlying NeRF model. Boosting View Synthesis [35] aims to improve image quality by transferring colour residuals from training views to novel views inspired by concepts from classic image-based rendering methods. The use of residuals depends on a pixel perfect alignment between the rendered and ground truth training views, however this is often not the case due to inaccurate camera pose estimation [12] thus limiting the possibility of accurate residual transfer and requiring a handcrafted weighting strategy to alleviate ghosting artifacts. NeRFLiX [60] on the other hand eschews classic 3D models entirely and instead proposes to learn a general 2D viewpoint mixer which is trained via simulated image degradation. This is reasonably effective provided the testing domain is well represented in the simulated degradation and neighbouring images are close enough in camera pose and image content. However if the distribution of the rendering artifacts shifts from the simulated data, the performance degrades. NeRFLiX++ [60] significantly improves the computational efficiency of NeRFLiX while also increasing the realism of image degradations by introducing a GAN-based degradation simulator, further boosting performance. Existing NeRF enhancers either use a 2D approach to learn a general function which is agnostic to scene geometry, or they use a 3D approach which suffers from inaccurate camera calibration and can propagate errors from the geometry into image renderings. In contrast, our method is robust to errors in camera poses and maintains view-consistency while also being pre-trained to learn a general enhancement function, effectively combining the advantages of 2D and 3D approaches.

3 Method

Overview. We present RoGUENeRF, a geometry-consistent enhancer for NeRF models which substantially improves the visual quality and fidelity of rendered images. We show an overview of our method in Figure 2. Our proposed approach consists of three core elements: 3D Alignment, Non-Rigid Refinement and Geometric Attention. We leverage the fact the NeRF model has learned an estimate of the scene geometry and can render depth maps as well as RGB images. For a novel test view, we use depth maps and camera poses to 3D-align training image features to the novel camera viewpoint. To compensate for any slight inaccuracies in estimated geometry, we improve the alignment further with non-rigid refinement by means of a lightweight iterative optical flow network. We then regulate the contribution of any remaining misaligned regions with a geometry-aware attention module. Finally the image features are fused and processed with

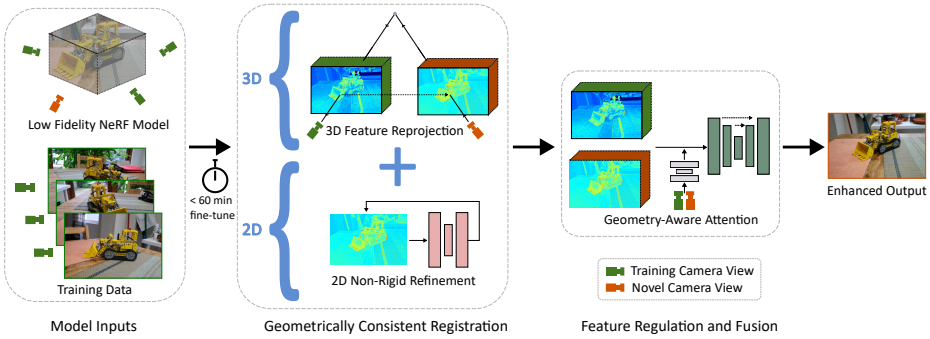

Fig. 2: RoGUENeRF Overview: Given a trained NeRF model and corresponding training data, our method substantially enhances the rendering quality of the NeRF while maintaining view-consistency.

a Uformer [51], a 2D enhancer on which our method is based. We pre-train our model on a small dataset of render-GT image pairs and show that we can quickly fine-tune on a novel scene to achieve a substantial improvement in image quality.

3.1 Preliminaries

NeRFs are trained on a set of ground truth RGB training images $\{H_i\}_{i=1}^{M}$ which capture a real world scene from set of known camera poses $\{C_i\}_{i=1}^{M}$, where M is the size of the training set. After training, NeRFs are able to freely render RGB images I_i and depths D_i of the scene from any camera viewpoint, including novel camera poses C_k not in $\{C_i\}_{i=1}^{M}$. NeRFs can interpolate well between camera poses, but the rendered images I_i are typically degraded in quality compared to the ground truth H_i especially in parts of the scene that contain high-frequency textures.

3.2 Nearest Neighbour Selection

The high-frequency textures lost during the NeRF optimization process are still present in the set of training images used to train the NeRF. We take advantage of this by finding the set of training images which have the largest overlap of image content with the novel rendered image I_k. Given a novel camera pose C_k and training poses $\{C_i\}_1^M$, we compute the nearest camera poses to C_k. We define a camera pose C to be the concatenation of a rotation matrix R and a translation vector $\boldsymbol{t}$ which describe the orientation and position of the camera respectively. To find the distance between rotations of different camera poses, we compute the three dimensional Euler angles of the rotation matrices and compute the mean of the L1 norm of the differences as follows:

$$dist_{ang}^{i}(R_k, R_i) = \frac{1}{3}|\pi(R_k) - \pi(R_i)|_1, \tag{1}$$

where $\pi(R_i)$ are the Euler angles of the rotation matrix of C_i and $dist^i_{ang}$ is the angular distance between C_k and C_i, and $|\ |_1$ is the L1 norm. We also compute the distance between the camera positions as follows:

$$dist^i_{pos}(\boldsymbol{t}_k, \boldsymbol{t}_i) = \frac{1}{3}|\boldsymbol{t}_k - \boldsymbol{t}_i|_1, \tag{2}$$

where $dist^i_{pos}$ is the positional distance between C_k and C_i. Empirically and qualitatively, we find $dist^i_{pos}$ to be the strongest indicator for image content overlap with a novel view, followed second by $dist^i_{ang}$. We first choose the 5 camera poses which have the smallest $dist^i_{pos}$ values, $\{C_p\}^5_{p=1}$, as follows:

$$\{C_p\}^5_{p=1} = \{C_j\} \mid\ j \in min_5\{dist^j_{pos}(\boldsymbol{t}_k, \{\boldsymbol{t}_j\}^M_{j=1})\}, \tag{3}$$

where $min_5\{\}$ are the 5 camera poses corresponding to the smallest distance values. Of these we choose the n with the smallest $dist^i_{ang}$ values, $\{C_i\}^n_{i=1}$, as follows:

$$\{C_i\}^n_{i=1} = \{C_p\} \mid\ p \in min_n\{dist^p_{ang}(R_k, \{R_p\}^5_{p=1})\}, \tag{4}$$

where $min_n\{\}$ are the n camera poses corresponding to the smallest distance values. The n training images corresponding to these camera poses, $\{H_i\}^n_{i=1}$ are considered to be the closest neighbouring training images with respect to our novel camera view.

3.3 3D Alignment

Once nearest neighbours are selected, we extract full resolution 64-dimensional image features using a small convolutional encoder block and reproject the features into the novel camera view using the pinhole camera model. This allows our enhancer to leverage relevant information from neighbours even if the cameras poses have very different orientation and position. Given a rendered novel view I_k and a set of neighbouring training images $\{H_i\}^n_{i=1}$, we extract image features with two separate convolutional encoder blocks as follows:

$$I^f_k = conv_I(I_k), \tag{5}$$

$$\{H^f_i\}^n_{i=1} = conv_H(\{H_i\}^n_{i=1}), \tag{6}$$

where I^f_k and $\{H^f_i\}^n_{i=1}$ are image features extracted from I_k and $\{H_i\}^n_{i=1}$ respectively and $conv_I()$ and $conv_H()$ are small convolutional blocks. We reproject the neighbouring image features into the novel camera view C_k using the pinhole camera model. Given a 3D coordinate $\boldsymbol{x}_k = (x_k, y_k, z_k)$ in the novel camera view, where (x_k, y_k) are the pixel coordinates in the image, and z_k is the depth value at that pixel coordinate, we reproject the coordinate into a neighbouring camera view as follows:

$$\boldsymbol{x}_{k\to i} = K_i C_i C_k^{-1} K_k^{-1}[\boldsymbol{x}_k, 1], \tag{7}$$

where $\boldsymbol{x}_{k\to i} = (x_{k\to i}, y_{k\to i}, z_{k\to i}, 1)$ is the 3D coordinate reprojected from camera k to camera i, K_i is the camera intrinsic matrix, C_i^{-1} is the inverse of the camera pose and $[\ ,\ ]$ denotes concatenation. To ensure geometric consistency, we conduct visibility testing by comparing the reprojected depth value $z_{k\to i}$ with the depth value computed by NeRF, z_i, as follows:

$$vis_i = \mathcal{H}\left(1 - \frac{z_{k\to i}}{z_i} + l\right) \tag{8}$$

where vis_i is the visibility score for the given pixel, $\mathcal{H}()$ denotes the Heaviside function, and l is a leniency threshold to account for a degree of inaccuracy in depth and camera pose estimates. The reprojected feature map is formed by copying the values from the reprojected coordinates $\boldsymbol{x}_{k\to i}$ into the original coordinates $\boldsymbol{x}_k$, weighted by the visibility score:

$$H^f_{i\to k}\langle\phi(\boldsymbol{x}_k)\rangle = vis_i \times H^f_i\langle\phi(\boldsymbol{x}_{k\to i})\rangle, \tag{9}$$

where $H^f_{i\to k}$ are the image features reprojected from camera i to camera k, $\phi(\boldsymbol{x}_k) = (\frac{x_k}{z_k}, \frac{y_k}{z_k})$ and $\langle\ \rangle$ denotes 2D coordinate indexing.

3.4 Non-Rigid Refinement

In real world data, there are often errors in camera pose estimation due to the limitations of COLMAP [12, 38], and also in the geometry estimated by NeRF, hence our 3D alignment is unlikely to find perfect correspondences. To account for this, we introduce a lightweight iterative optical flow network which further refines the alignment between the neighbouring images and the novel view image. Typically optical flow methods expect two clean images to accurately find correspondences but the domain gap between the blurry rendered image and the neighbouring images violates this assumption. Our choice however is motivated by the fact that optical flow methods can produce reasonable results based on global structures and shapes alone [12]. We use the flow network presented in [6] as it has been shown to work well even with domain gaps. We perform the iterative refinement in feature space and learn a network trained end-to-end which is optimized for our specific task instead of general purpose alignment [6,13]. Given the 3D aligned neighbouring features $H^f_{i\to k}$, we refine the alignment further as follows:

$$f_{H^f_{i\to k}} = \mathcal{F}(H^f_{i\to k}, I^f_k), \tag{10}$$

$$H^{f'}_{i\to k} = warp_{2D}(H^f_{i\to k}, f_{H^f_{i\to k}}), \tag{11}$$

where $f_{H^f_{i\to k}}$ is the estimated flow field, $H^{f'}_{i\to k}$ are the reprojected and warped neighbouring image features, $\mathcal{F}()$ is our lightweight optical flow network and $warp()_{2D}$ denotes the function for warping an image with a 2D flow field.

3.5 Geometry-Aware Attention

Any regions which remain misaligned after our combined 3D and 2D alignment can feed through to the final enhanced results as ghosting artifacts. Spatial attention has been shown to be effective at reducing such artifacts [55]. We propose a learnable combined spatial and geometric attention module to regulate misaligned regions. Our geometry-aware attention is based on both 2D and 3D spatial information. This allows our enhancer to regulate contributions from neighbours based on similarities in pixel content and also geometric distance based on camera orientation and depths. Given the neighbouring and novel view image features, as well as the neighbour depth projected to the novel view $D_{i\to k}$ and the novel view depth D_k, we compute the pixel attention weights as follows:

$$\psi^i_{pix} = \mathcal{A}_{pix}(H^{f'}_{i\to k}, I^f_k, D_{i\to k}, D_k), \tag{12}$$

where $\psi^i_{pix} \in \mathbb{R}^{w\times h}$ are the pixel attention weights, w and h are the width and height of the image and $\mathcal{A}_{pix}()$ is the pixel attention module which is composed of two convolutional layers and a sigmoid activation. In the second stage, the camera attention weights are computed using the Euler angles and positions of neighbour and novel view camera poses:

$$\psi^i_{cam} = \mathcal{A}_{cam}(\pi(R_i), \pi(R_k), \mathbf{t}_i, \mathbf{t}_k) \tag{13}$$

where $\psi^i_{cam} \in \mathbb{R}^1$ are the camera attention weights and $\mathcal{A}_{cam}()$ is the camera attention module which is composed of two fully connected layers and a sigmoid activation. Finally both sets of the weights are applied to the neighbour image features. ψ_{pix} is applied at a per-pixel level while ψ_{cam} is applied at a per-image level:

$$H^{f^a}_{i\to k} = \psi^i_{cam} \times \psi^i_{pix} \times H^{f'}_{i\to k}, \tag{14}$$

where $H^{f^a}_{i\to k}$ are the attention regulated image features.

3.6 Feature Fusion

We use the maxpooling approach described in [1] to combine our set of attention regulated neighbouring features, $\{H^{f^a}_{i\to k}\}^n_{i=1}$ into a single feature map H^f_{pool}. This approach has the advantages of outperforming concatenation [6] and also defining a flexible architecture which can accept any number of input neighbours. This gives us the ability to use fewer or more neighbouring images depending on factors such as availability of data, image resolution and GPU memory, ensuring our enhancer is practical and can be applied in multiple settings. Finally we process the pooled features and novel view features together with a convolutional layer and enhance them further using a 2D enhancer, Uformer [51], which we modify to accept image features instead of RGB inputs:

$$\hat{H}_k = \mathcal{U}(conv_{merge}([I^f_k, H^f_{pool}])), \tag{15}$$

where $\mathcal{U}$ is the Uformer, and $conv_{merge}()$ denotes a convolutional layer.

3.7 Pre-training and Implementation Details

Our enhancer is first pre-trained using a single NeRF baseline and dataset, specifically NeRF [23] and LLFF [22]. We then fine-tune for 1 hour on each new scene or novel NeRF baseline method. To generate the training and fine-tuning data for our enhancer, we first train a given baseline NeRF model on a scene and render all images from the training set, which generates a set of render-GT pairs. We pre-train our model on all scenes from the LLFF dataset using the renders generated by NeRF [23] for 3000 epochs (approximately 5 days) and fine-tune on novel scenes and NeRF models for one hour per scene, which is comparable to the per-scene training time of state-of-the-art NeRF methods [4]. We reduce L1 and perceptual losses between the enhanced image and ground truth as follows:

$$L = |\hat{H}_i - H_i|_1 + 10^{-3}|\omega(\hat{H}_i) - \omega(H_i)|_1, \tag{16}$$

where $\hat{H}_i$ is our predicted enhanced image, H_i is the GT, L is our loss function and $\omega()$ is a pre-trained VGG-19 [40]. We use random crops of size 512×512 with a batch size of 4 and a learning rate of 1×10^{-4} with the Adam optimizer [15]. We use a leniency threshold of 0.25 for visibility testing and we use 5 neighbours for the LLFF and 360v2 datasets, and 2 neighbours for the DTU dataset. We train our model using PyTorch [19, 29] on $4\times$ NVidia V100 GPUs. For all baselines and enhancers, we use the official code and checkpoints provided by authors when available. For reproducibility, we provide full implementation details of each component of our method in the supplementary.

4 Results

4.1 Datasets and Metrics

We evaluate our method on three varied real world multi-view datasets, LLFF (8 scenes) [22], 360v2 (9 scenes) [3] and DTU (124 scenes) [11]. Together, these datasets contain a mixture of front-facing and 360° scenes, both indoor bounded and outdoor unbounded, with complex geometries and a range of high-frequency textures. The total number of images per scene varies from 20 to 311. Note, we evaluate all scenes from LLFF and 360v2 at a consistent $4\times$ downsampled resolution, unlike previous works which evaluate the indoor and outdoor scenes from 360v2 at different downsampling rates. We evaluate all DTU scenes at full resolution and use the diffuse light setting. For each scene, every eighth image is held out for testing and the remaining images are used to train the NeRF baselines and fine-tune our enhancer. Following previous works, we use PSNR, SSIM [52] and LPIPS (VGG) [58] to evaluate our method.

4.2 Baseline Methods

To demonstrate the general applicability of our proposed enhancer, we extensively evaluate our method using six different baseline NeRF methods which

Table 1: Quantitative evaluation of our enhancer applied to six different NeRF baselines. Results are averaged across all scenes for each dataset. Red and orange highlights indicate 1st and 2nd best performing methods. Our model consistently outperforms all baselines and other enhancers across all metrics. †Results as reported by authors.

Model	Dataset	PSNR (dB) ↑	SSIM↑	LPIPS↓
ZipNeRF	360v2	28.90	0.8367	0.1779
ZipNeRF + NeRFLiX	360v2	29.00 (↑ 0.10)	0.8317 (↓ 0.005)	0.2045 (↑ 15.0%)
ZipNeRF + Ours	360v2	29.23 (↑ 0.33)	0.8465 (↑ 0.098)	0.1662 (↓ 6.6%)
MipNeRF360	360v2	28.26	0.8050	0.2297
MipNeRF360 + NeRFLiX	360v2	28.44 (↑ 0.18)	0.8036 (↓ 0.001)	0.2441 (↑ 6.3%)
MipNeRF360 + Ours	360v2	28.89 (↑ 0.63)	0.8302 (↑ 0.025)	0.1987 (↓ 13.5%)
Nerfacto	360v2	26.11	0.7157	0.3266
Nerfacto + NeRFLiX	360v2	26.92(↑ 0.81)	0.7410 (↑ 0.025)	0.3044 (↓ 6.8%)
Nerfacto + Ours	360v2	27.45(↑ 1.34)	0.7700 (↑ 0.054)	0.2670 (↓ 18.2%)
NeuS2++	DTU	27.35	0.7587	0.4386
NeuS2++ + NeRFLiX	DTU	27.40 (↑ 0.05)	0.7752 (↑ 0.017)	0.4167 (↓ 5.0%)
NeuS2++ + Ours	DTU	28.46 (↑ 1.11)	0.8237 (↑ 0.065)	0.3939 (↓ 10.2%)
NeRF	LLFF	26.57	0.8170	0.2389
NeRF + Boosting View Synthesis†	LLFF	27.08 (↑ 0.51)	0.8371 (↑ 0.020)	-
NeRF + NeRFLiX	LLFF	27.17 (↑ 0.60)	0.8552 (↑ 0.038)	0.1695 (↓ 29.0%)
NeRF + NeRFLiX++†	LLFF	27.25 (↑ 0.68)	0.8580 (↑ 0.041)	-
NeRF + Ours	LLFF	27.67 (↑ 1.10)	0.8713 (↑ 0.054)	0.1495 (↓ 37.4%)
TensoRF	LLFF	26.88	0.8432	0.1829
TensoRF + NeRFLiX	LLFF	27.38 (↑ 0.50)	0.8652 (↑ 0.022)	0.1514 (↓ 17.2%)
TensoRF + NeRFLiX++†	LLFF	27.38 (↑ 0.50)	0.8660 (↑ 0.023)	-
TensoRF + Ours	LLFF	27.58 (↑ 0.70)	0.8670 (↑ 0.024)	0.1494 (↓ 18.3%)

together represent the evolution of neural rendering over the last few years. These include the current state-of-the-art with respect to image fidelity (MipNeRF360 [3], ZipNeRF [4]), training speed (Nerfacto [44], NeuS2++ an unbounded variant of NeuS2 [50]) and older seminal works (NeRF [23], TensoRF [7]). We compare to state-of-the-art NeRF enhancers including Boosting View Synthesis [35], NeRFLiX [60] and NeRFLiX++ [59], an extension of NeRFLiX using a GAN-based degradation simulator. For each dataset, we report results averaged across all scenes in Table 1. We provide per-scene results of each method and also compare to AligNeRF [12] on outdoor scenes in the supplementary. As there is no available code, we report results directly from the original works for [12], [35] and [59].

4.3 Quantitative and Qualitative Evaluation

We present quantitative results in Table 1. We show that our model achieves improvements in all metrics for all six NeRF baselines. On the 360v2 dataset, RoGUENeRF improves the PSNR of Nerfacto by 1.34dB, MipNeRF360 by 0.63dB and ZipNeRF by 0.33dB, and achieves corresponding reductions in LPIPS of 18.2%, 13.5% and 6.6% respectively. On the DTU and LLFF datasets, RoGUEN-

Table 2: Evaluation of robustness to camera pose noise. We present MUSIQ [14] scores when Gaussian noise is added to training camera poses on the Garden scene from the 360v2 dataset. Small, medium and large noise settings correspond to standard deviations of 6.25e-2/3.125e-4 | 12.5e-2/6.25e-4 | 25e-2/12.5e-4 with respect to camera rotation/position.

Model	No Noise	Small Noise	Medium Noise	Large Noise
Nerfacto	66.50	53.33	41.65	32.43
Nerfacto + Ours	72.32 (↑5.8)	62.59 (↑9.3)	58.94 (↑17.3)	51.60 (↑19.2)
ZipNeRF	71.38	63.29	58.67	49.50
ZipNeRF + Ours	72.06 (↑0.7)	69.76 (↑6.5)	65.03 (↑6.4)	52.39(↑2.9)

eRF improves the PSNR of Neus2++ by 1.11dB, NeRF by 1.10dB and TensoRF by 0.70dB, with corresponding reductions in LPIPS of 10.2%, 37.4% and 18.3% respectively. We consistently improve SSIM scores for all baselines and datasets. We also outperform state-of-the-art NeRF enhancers NeRFLiX and NeRFLiX++ in every setting. These results are reflected in the qualitative evaluation shown in Figures 1 and 3. We note that ZipNeRF already achieves very high fidelity in high-frequency regions, so the improvements from our enhancer are largely from denoising without over-smoothing. We show noticeable improvements over all other NeRF baseline models and NeRFLiX in restoring high-frequency details. We also present video results in the supplementary which demonstrate that RoGUENeRF has view-consistency in line with the baseline NeRF models, as opposed to NeRFLiX which is geometry-agnostic and suffers from flickering in high-frequency regions.

4.4 Robustness To Inaccurate Camera Calibration

We evaluate the robustness of our method to inaccurate camera calibration in Table 2. To simulate the effects of inaccurate camera pose estimation, we apply increasing levels of additive zero mean Gaussian noise to the camera poses estimated by COLMAP. The rotation standard deviation is expressed in degrees while position standard deviation is expressed as a unit of physical distance, with total scene size set to a bounding cube of length 2. For each noise level, we train Nerfacto and ZipNeRF using the noisy camera poses and evaluate the ability of our enhancer to improve the perceptual quality of the noisy NeRF baselines. Training with incorrect camera poses introduces pixel shifts between rendered images and ground truth, so we assess performance with MUSIQ [14], a SOTA no-reference image quality assessment metric. We show that our improvements over Nerfacto and ZipNeRF are larger when using noisy camera poses compared to the no noise setting. This is reflected in Figure 4 where we qualitatively evaluate the medium noise setting. The baselines suffer a large drop in image quality while RoGUENeRF is robust to inaccurate poses and is able to achieve noticeable improvements over the noisy baselines.

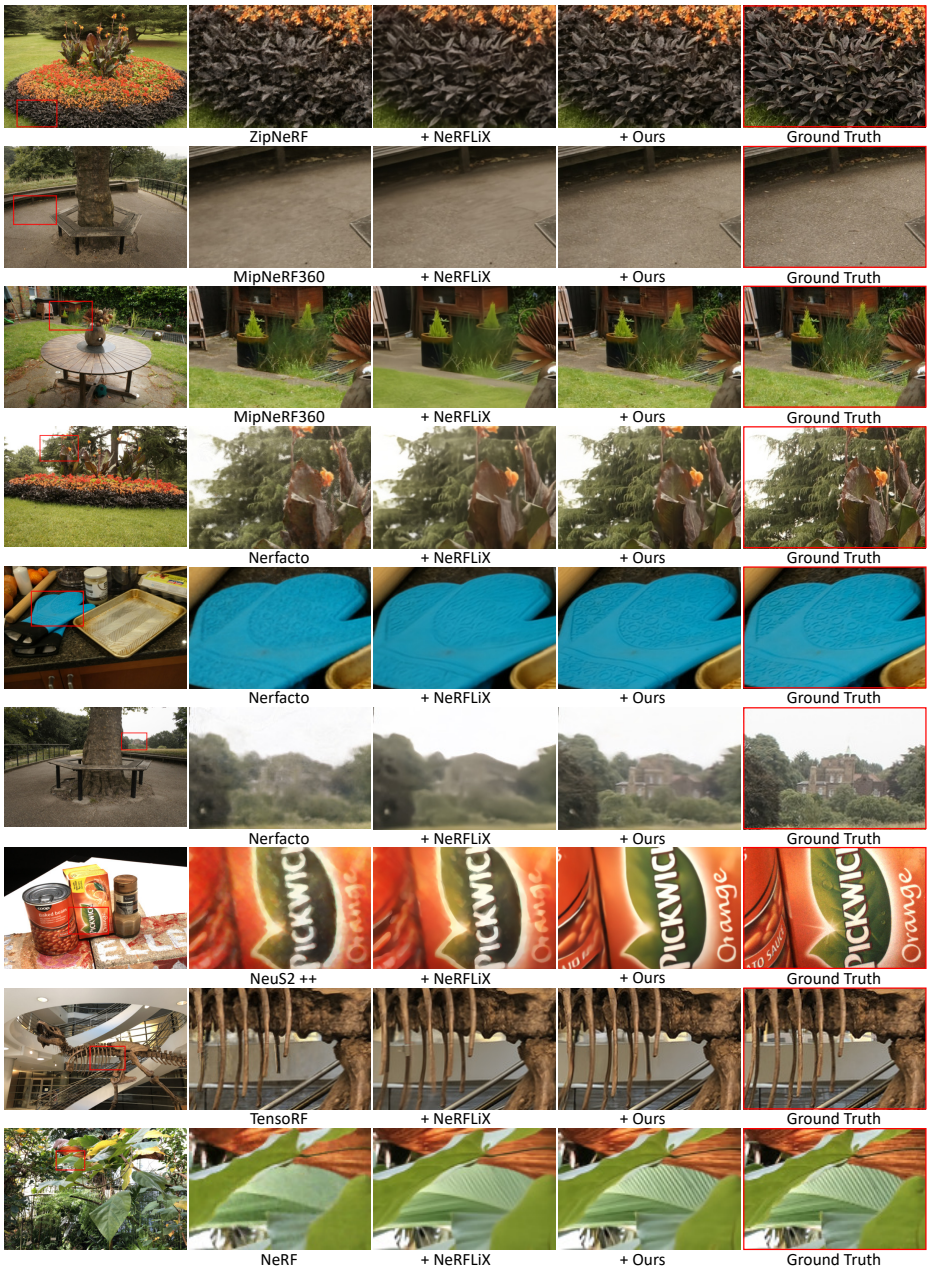

Fig. 3: Qualitative comparisons with six different NeRF baseline models across three datasets. Our method recovers more detail in high-frequency regions such as foliage, tarmac floor, patterns on the glove, the edges of the building and text.

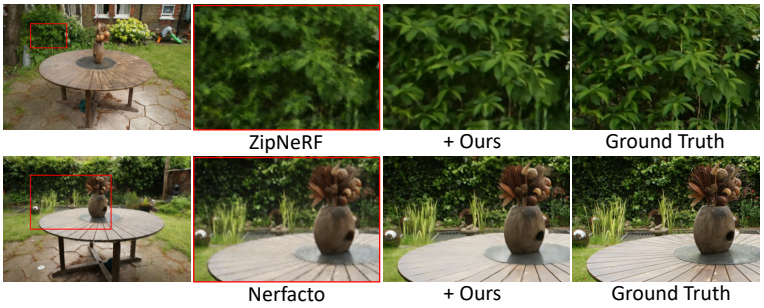

Fig. 4: Qualitative results when adding medium pose noise. The quality of the baseline suffers greatly while our enhancer is robust and achieves noticeable improvements.

4.5 Ablation Study

We validate the importance of individual components of our method in Table 3. We conduct the study across all scenes in the LLFF dataset and present average results. Our combined contributions show an improvement of 1.1dB PSNR over NeRF and 0.93dB over a strong 2D baseline enhancer, Uformer [51] on which our proposed approach is based. We show a sharp improvement in LPIPS perceptual quality from our 3D alignment module (7.4%), and a further improvement from our non-rigid refinement (13.4%), which greatly improves the ability of our model to find accurate correspondences and correct errors in geometry. We show qualitative results of our ablation in Figure 5. We evaluate the following model components: **UF**: Baseline 2D enhancer Uformer which our approach is based on. **NN**: Nearest Neighbour Selection. **3D-A**: Our 3D Alignment module comprising of feature reprojection using depths and camera poses. **NRF**: Our Non-Rigid Refinement module comprising of an iterative lightweight optical flow network which further aligns the reprojected features. **GA**: Our Geometric Attention module comprising a geometry-aware feature regulation mechanism. **PT**: The effect of pre-training our model. We conduct a more detailed ablation study on pre-training and fine-tuning in Table 4. Here we show two things; that our method can achieve large improvements in as little as 1 minute of fine-tuning; and that it is our complete combined contributions which allow our method to learn novel image degradations so quickly. Uformer+NN also has access to nearest neighbours from the training set, but as it is unaware of geometry, it is not capable of quickly leveraging the information from nearby viewpoints.

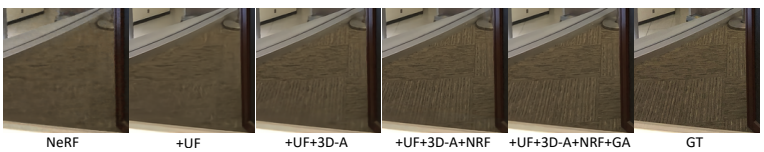

Fig. 5: Qualitative results of the ablation study. Settings correspond to Table 3.

Table 3: Ablation on the performance of our individual contributions averaged across all eight scenes from the LLFF dataset with NeRF as the baseline model.

Model	UF	NN	3D-A	NRF	GA	PT	PSNR	SSIM	LPIPS
NeRF	☐	☐	☐	☐	☐	☐	26.57	0.8170	0.2389
+ Uformer	☑	☐	☐	☐	☐	☐	26.74	0.8314	0.2245
+ 3D Alignment	☑	☑	☑	☐	☐	☐	26.84	0.8397	0.2078
+ Non-Rigid Refinement	☑	☑	☑	☑	☐	☐	26.93	0.8467	0.1799
+ Geometric Attention	☑	☑	☑	☑	☑	☐	27.00	0.8501	0.1755
+ Pre-training	☑	☑	☑	☑	☑	☑	27.67	0.8713	0.1495

Table 4: Ablation on fine-tuning time on the Garden scene from the 360v2 dataset with Nerfacto as the baseline model. Column headers on the right indicate time spent fine-tuning each model. Results presented are PSNR (dB).

Model	UF	NN	3D-A	NRF	GA	PT	1 Min	10 Minutes	1 Hour
Nerfacto	☐	☐	☐	☐	☐	☐	25.28	25.28	25.28
+ Uformer	☑	☐	☐	☐	☐	☑	25.43	25.52	25.73
+ Uformer+NN	☑	☑	☐	☐	☐	☑	25.39	25.60	25.67
+ Ours	☑	☑	☑	☑	☑	☑	26.08	26.19	26.24

Limitations A limitation of our method is the requirement to store all relevant training images. This could be prohibitive for very large scenes, especially at higher image resolutions. Secondly, although we are several times faster than the baseline NeRF methods, our approach does not yet achieve real time (*i.e.* 30fps) inference. **Societal Impact** As NeRFs become more editable, our method could be misused to improve the photorealism of generated videos of people, places and objects. Currently our method requires ground truth images with a dense 3D coverage of the scene, so it is not trivial to use in the wild.

5 Conclusion

We have presented RoGUENeRF, a geometry-consistent NeRF enhancer which combines concepts from 3D and 2D vision to substantially improve the image quality of NeRF renderings in real world settings. Our model accurately finds correspondences between different camera views by performing 3D alignment and non-rigid refinement, while also being robust to errors in camera pose estimation and reducing reprojection artifacts with geometry-aware attention. RoGUENeRF achieves consistent improvements in image quality over six varied NeRF baselines and existing NeRF enhancers across three challenging real world datasets. Our model demonstrates wide applicability and strong generalization, fine-tuning on a novel scene in under 60 minutes to learn the distribution of image degradations.

References

1. Aittala, M., Durand, F.: Burst image deblurring using permutation invariant convolutional neural networks. In: ECCV (September 2018)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5835–5844 (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: ICCV (2023)
5. Bian, W., Wang, Z., Li, K., Bian, J.W., Prisacariu, V.A.: Nope-nerf: Optimising neural radiance field with no pose prior. In: CVPR. pp. 4160–4169 (June 2023)
6. Catley-Chandar, S., Tanay, T., Vandroux, L., Leonardis, A., Slabaugh, G., Pérez-Pellitero, E.: FlexHDR: Modeling alignment and exposure uncertainties for flexible HDR imaging. IEEE TIP **31** (2022)
7. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022)
8. Dhamo, H., Nie, Y., Moreau, A., Song, J., Shaw, R., Zhou, Y., Pérez-Pellitero, E.: Headgas: Real-time animatable head avatars via 3d gaussian splatting. arXiv preprint arXiv:2312.02902 (2023)
9. Huang, X., Li, W., Hu, J., Chen, H., Wang, Y.: Refsr-nerf: Towards high fidelity and super resolution view synthesis. In: CVPR. pp. 8244–8253. IEEE Computer Society, Los Alamitos, CA, USA (jun 2023)
10. Işık, M., Rünz, M., Georgopoulos, M., Khakhulin, T., Starck, J., Agapito, L., Nießner, M.: Humanrf: High-fidelity neural radiance fields for humans in motion. ACM TOG **42**(4), 1–12 (2023)
11. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR. pp. 406–413. IEEE (2014)
12. Jiang, Y., Hedman, P., Mildenhall, B., Xu, D., Barron, J.T., Wang, Z., Xue, T.: Alignerf: High-fidelity neural radiance fields via alignment-aware training. In: CVPR. pp. 46–55 (2023)
13. Kalantari, N.K., Ramamoorthi, R.: Deep HDR Video from Sequences with Alternating Exposures. Comput. Graph. Forum (2019)
14. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: Multi-scale image quality transformer. In: ICCV. pp. 5148–5157 (October 2021)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
16. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R.A., Lv, Z.: Neural 3d video synthesis from multi-view video. In: CVPR. pp. 5511–5521 (2021)
17. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: ICCV (2021)
18. Liu, Y.L., Gao, C., Meuleman, A., Tseng, H.Y., Saraf, A., Kim, C., Chuang, Y.Y., Kopf, J., Huang, J.B.: Robust dynamic radiance fields. In: CVPR (2023)
19. maintainers, T., contributors: TorchVision: PyTorch's Computer Vision library (Nov 2016)
20. Meuleman, A., Liu, Y.L., Gao, C., Huang, J.B., Kim, C., Kim, M.H., Kopf, J.: Progressively optimized local radiance fields for robust view synthesis. In: CVPR. pp. 16539–16548 (June 2023)

21. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: NeRF in the dark: High dynamic range view synthesis from noisy raw images. In: CVPR (2022)
22. 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)
23. 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)
24. Moreau, A., Piasco, N., Tsishkou, D.V., Stanciulescu, B., de La Fortelle, A.: Lens: Localization enhanced by nerf synthesis. In: Conference on Robot Learning (2021)
25. Moreau, A., Song, J., Dhamo, H., Shaw, R., Zhou, Y., Pérez-Pellitero, E.: Human gaussian splatting: Real-time rendering of animatable avatars. In: CVPR (2024)
26. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG **41**(4) (2022)
27. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: CVPR (2021)
28. Park, K., Henzler, P., Mildenhall, B., Barron, J.T., Martin-Brualla, R.: Camp: Camera preconditioning for neural radiance fields. ACM Trans. Graph. (2023)
29. 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: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) NeurIPS. pp. 8024–8035. Curran Associates, Inc. (2019)
30. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human bodies. In: ICCV (2021)
31. Peng, S., Yan, Y., Shuai, Q., Bao, H., Zhou, X.: Representing volumetric videos as dynamic mlp maps. In: CVPR (2023)
32. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: Neural Radiance Fields for Dynamic Scenes. In: CVPR (2020)
33. Raoult, V., Reid-Anderson, S., Ferri, A., Williamson, J.E.: How reliable is structure from motion (sfm) over time and between observers? a case study using coral reef bommies. Remote Sensing **9**(7) (2017)
34. Roessle, B., Müller, N., Porzi, L., Bulò, S.R., Kontschieder, P., Nießner, M.: Ganerf: Leveraging discriminators to optimize neural radiance fields. ACM TOG (2023)
35. Rong, X., Huang, J.B., Saraf, A., Kim, C., Kopf, J.: Boosting view synthesis with residual transfer. In: CVPR (2022)
36. Sara Fridovich-Keil and Alex Yu, Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)
37. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
38. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
39. Shaw, R., Song, J., Moreau, A., Nazarczuk, M., Catley-Chandar, S., Dhamo, H., Perez-Pellitero, E.: Swags: Sampling windows adaptively for dynamic 3d gaussian splatting. arXiv preprint arXiv:2312.13308 (2023)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
41. Tanay, T., Leonardis, A., Maggioni, M.: Efficient view synthesis and 3d-based multi-frame denoising with multiplane feature representations. In: CVPR (2023)

42. Tanay, T., Maggioni, M.: Global latent neural rendering. In: CVPR (2024)
43. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In: NeurIPS (2020)
44. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23 (2023)
45. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: CVPR. pp. 4190–4200 (June 2023)
46. Turki, H., Agrawal, V., Bulò, S.R., Porzi, L., Kontschieder, P., Ramanan, D., Zollhöfer, M., Richardt, C.: Hybridnerf: Efficient neural rendering via adaptive volumetric surfaces. In: Computer Vision and Pattern Recognition (CVPR) (2024)
47. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In: CVPR (2022)
48. Wang, C., Chai, M., He, M., Chen, D., Liao, J.: Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In: CVPR. pp. 3835–3844 (2022)
49. Wang, C., Wu, X., Guo, Y.C., Zhang, S.H., Tai, Y.W., Hu, S.M.: Nerf-sr: High quality neural radiance fields using supersampling. In: ACM MM. p. 6445–6454. MM '22, Association for Computing Machinery, New York, NY, USA (2022)
50. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: ICCV (2023)
51. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: CVPR. pp. 17683–17693 (June 2022)
52. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
53. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond. Comput. Graph. Forum (2022)
54. Xu, L., Agrawal, V., Laney, W., Garcia, T., Bansal, A., Kim, C., Rota Bulò, S., Porzi, L., Kontschieder, P., Božič, A., Lin, D., Zollhöfer, M., Richardt, C.: VR-NeRF: High-fidelity virtualized walkable spaces. In: SIGGRAPH Asia Conference Proceedings (2023)
55. Yan, Q., Gong, D., Shi, Q., Hengel, A.v.d., Shen, C., Reid, I., Zhang, Y.: Attention-guided network for ghost-free high dynamic range imaging. In: CVPR. pp. 1751–1760 (2019)
56. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: ICCV (2021)
57. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv:2010.07492v2 (2020)
58. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
59. Zhou, K., Li, W., Jiang, N., Han, X., Lu, J.: From nerflix to nerflix++: A general nerf-agnostic restorer paradigm. IEEE TPAMI pp. 1–17 (2023)
60. Zhou, K., Li, W., Wang, Y., Hu, T., Jiang, N., Han, X., Lu, J.: Nerflix: High-quality neural view synthesis by learning a degradation-driven inter-viewpoint mixer. In: CVPR. pp. 12363–12374 (2023)