NeRF-SR: High Quality Neural Radiance Fields using Supersampling

Chen Wang
BNRist, Department of Computer
Science and Technology, Tsinghua
University
Beijing, China

Xian Wu
Kuaishou Technology
Beijing, China

Yuan-Chen Guo
BNRist, Department of Computer
Science and Technology, Tsinghua
University
Beijing, China

Song-Hai Zhang
BNRist, Department of Computer
Science and Technology, Tsinghua
University
Beijing, China

Yu-Wing Tai[*]
Kuaishou Technology
Beijing, China
yuwing@gmail.com

Shi-Min Hu
BNRist, Department of Computer
Science and Technology, Tsinghua
University
Beijing, China

ABSTRACT

We present NeRF-SR, a solution for high-resolution (HR) novel view synthesis with mostly low-resolution (LR) inputs. Our method is built upon Neural Radiance Fields (NeRF) [32] that predicts per-point density and color with a multi-layer perceptron. While producing images at arbitrary scales, NeRF struggles with resolutions that go beyond observed images. Our key insight is that NeRF benefits from 3D consistency, which means an observed pixel absorbs information from nearby views. We first exploit it by a supersampling strategy that shoots multiple rays at each image pixel, which further enforces multi-view constraint at a sub-pixel level. Then, we show that NeRF-SR can further boost the performance of supersampling by a refinement network that leverages the estimated depth at hand to hallucinate details from related patches on only one HR reference image. Experiment results demonstrate that NeRF-SR generates high-quality results for novel view synthesis at HR on both synthetic and real-world datasets without any external information. Project page: https://cwchenwang.github.io/NeRF-SR

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Computer graphics**.

KEYWORDS

Neural Radiance Fields, Super-resolution

[*]Corresponding author

1 INTRODUCTION

Synthesizing photorealistic views from a novel viewpoint given a set of posed images, known as *novel view synthesis*, has been a long-standing problem in the computer vision community, and an important technique for VR and AR applications such as navigation, and telepresence. Traditional approaches mainly falls in the range of image-based rendering and follows the process of warping and blending source frames to target views [11, 21]. Image-based rendering methods heavily rely on the quality of input data and only produces reasonable renderings with dense observed views and accurate proxy geometry.

(a)   (b)

Figure 1: NeRF, the state-of-the-art novel view synthesis method, can synthesize photorealistic outputs at the resolution of training images but struggles at higher resolutions as shown in (a), while NeRF-SR produces high-quality novel views (b) even with low-resolution inputs.

Most recently, *neural rendering* has made significant progress on novel view synthesis by leveraging learnable components with 3D geometry context to reconstruct novel views with respect to input images. As the current state-of-the-art method, neural radiance fields (NeRF) [32] have emerged as a promising direction for neural

scene representation even on sparse image sets of complex real-world scenes. NeRF uses the weights of multi-layer perceptrons (MLPs) to encode the radiance field and volume density of a scene. Most importantly, the implicit neural representation is continuous, which enables NeRF to take as input any position in the volume at inference time and render images at any arbitrary resolution.

At the same time, a high-resolution 3D scene is essential for many real-world applications, e.g., a prerequisite to providing an immersive virtual environment in VR. However, a trained NeRF struggles to generalize directly to resolutions higher than that of the input images and generates blurry views (See Figure 1), which presents an obstacle for real-world scenarios, e.g., images collected from the Internet may be low-resolution. To tackle this problem, we present NeRF-SR, a technique that extends NeRF and creates high-resolution (HR) novel views with better quality even with low-resolution (LR) inputs. We first observe there is a sampling gap between the training and testing phase for super-resolving a 3D scene, since the sparse inputs are far from satisfying Nyquist view sampling rates [31]. To this end, we derive inspiration from traditional graphics pipeline and propose a supersampling strategy to better enforce the multi-view consistency embedded in NeRF in a sub-pixel manner, enabling the generation of both SR images and SR depth maps. Second, in the case of limited HR images such as panoramas and light field imaging systems that have a trade-off between angular and spatial resolutions, we find that directly incorporating them in the NeRF training only improves renderings *nearby the HR images* in a small margin. Thus, we propose a patch-wise warp-and-refine strategy that utilizes the estimated 3D geometry and propagate the details of HR reference to *all over the scene*. Moreover, the refinement stage is efficient and introduces negligible running time compared with NeRF rendering.

To the best of our knowledge, we are the first to produce visually pleasing results for novel view synthesis under mainly low-resolution inputs. Our method requires only posed multi-view images of the target scene, from which we dig into the internal statistics and does not rely on any external priors. We show that NeRF-SR outperforms baselines that require LR-HR pairs for training.

Our contributions are summarized as follows:

- the first framework that produces decent multi-view super-resolution results with mostly LR input images
- a supersampling strategy that exploits the view consistency in images and supervises NeRF in the sub-pixel manner
- a refinement network that blends details from any HR reference by finding relevant patches with available depth maps

2 RELATED WORK

Novel View Synthesis. Novel view synthesis can be categorized into image-based, learning-based, and geometry-based methods. Image-based methods warp and blend relevant patches in the observation frames to generate novel views based on measurements of quality [11, 21]. Learning-based methods predict blending weights and view-dependent effects via neural networks and/or other hand-crafted heuristics[5, 12, 41, 53]. Deep learning has also facilitated methods that can predict novel views from a single image, but they often require a large amount of data for training[34, 43, 47, 55, 60].

Different from image-based and learning-based methods, geometry-based methods first reconstruct a 3D model [46] and render images from target poses. For example, Aliev *et al.* [1] assigned multi-resolution features to point clouds and then performed neural rendering, Thies *et al.* [52] stored neural textures on 3D meshes and then render the novel view with traditional graphics pipeline. Other geometry representations include multi-planes images [9, 22, 23, 31, 50, 68], voxel grids [13, 17, 39], depth [9, 41, 42, 60] and layered depth [47, 56]. These methods, although producing relatively high-quality results, the discrete representations require abundant data and memory and the rendered resolutions are also limited by the accuracy of reconstructed geometry.

Neural Radiance Fields. Implicit neural representation has demonstrated its effectiveness to represent shapes and scenes, which usually leverages multi-layer perceptrons (MLPs) to encode signed distance fields [8, 35], occupancy [4, 30, 38] or volume density [32, 33]. Together with differentiable rendering [18, 27], these methods can reconstruct both geometry and appearance of objects and scenes [26, 33, 44, 48, 49]. Among them, Neural Radiance Fields (NeRF) [32] achieved remarkable results for synthesizing novel views of a static scene given a set of posed input images. There are a growing number of NeRF extensions emerged, e.g., reconstruction without input camera poses[25, 59], modelling non-rigid scenes [28, 36, 37, 40], unbounded scenes[64] and object categories [15, 54, 63]. Relevant to our work, Mip-NeRF [2] also considers the issue of *resolution* in NeRF. They showed that NeRFs rendered at various resolutions would introduce aliasing artifacts and resolved it by proposing an integrated positional encoding that featurize conical frustums instead of single points. Yet, Mip-NeRF only considers rendering with downsampled resolutions. To our knowledge, no prior work studies how to increase the resolution of NeRF.

Image Super-Resolution Our work is also related to image super-resolution. Classical approaches in single-image super-resolution (SISR) utilize priors such as image statistics [19, 69] or gradients [51]. CNN-based methods aim to learn the relationship between HR and LR images in CNN by minimizing the mean-square errors between SR images and ground truths [6, 7, 57]. Generative Adversarial Networks (GANs) [10] are also popular in super-resolution which hallucinates high resolution details by adversarial learning [20, 29, 45]. These methods mostly gain knowledge from large-scale datasets or existing HR and LR pairs for training. Besides, these 2D image-based methods, especially GAN-based methods do not take the view consistency into consideration and are sub-optimal for novel view synthesis.

Reference-based image super-resolution (Ref-SR) upscales input images with additional reference high-resolution (HR) images. Existing methods match the correspondences between HR references and LR inputs with patch-match [66, 67], feature extraction [61, 62] or attention [62]. Although we also aim to learn HR details from given reference images, we work in the 3D geometry perspective and can bring details for all novel views instead of one image.

3 BACKGROUND

Neural Radiance Fields (NeRF) [32] encodes a 3D scene as a continuous function which takes as input 3D position $\mathbf{x} = (x, y, z)$ and

observed viewing direction $\mathbf{d} = (\theta, \phi)$, and predicts the radiance $\mathbf{c}(\mathbf{x}, \mathbf{d}) = (r, g, b)$ and volume density $\sigma(\mathbf{x})$. The color depends both on viewing direction $\mathbf{d}$ and $\mathbf{x}$ to capture view dependent effects, while the density only depends on $\mathbf{x}$ to maintain view consistency. NeRF is typically parametrized by a multilayer perceptron (MLP) $f : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$.

NeRF is an emission-only model (the color of a pixel only depends on the radiance along a ray with no other lighting factors). Therefore, according to volume rendering [16], the color along the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ that shots from the camera center $\mathbf{o}$ in direction $\mathbf{d}$ can be calculated as:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\mathrm{d}t \qquad (1)$$

where

$$T(t) = \exp\left(- \int_{t_n}^{t} \sigma(\mathbf{r}(t))\mathrm{d}t \right) \qquad (2)$$

is the accumulated transmittance that indicates the probability that a ray travels from t_n to t without hitting any particle.

NeRF is trained to minimize the mean-squared error (MSE) between the predicted renderings and the corresponding ground-truth color:

$$\mathcal{L}_{\text{MSE}} = \sum_{\mathbf{p} \in \mathcal{P}} \|\hat{\mathbf{C}}(\mathbf{r_p}) - \mathbf{C}(\mathbf{r_p})\|_2^2 \qquad (3)$$

where $\mathcal{P}$ denotes all pixels of training set images, $\mathbf{r_p}(t) = \mathbf{o} + t\mathbf{d_p}$ denotes the ray shooting from camera center to the corners (or centers in some variants [2]) of a given pixel $\mathbf{p}$. $\hat{\mathbf{C}}(\mathbf{r_p})$ and $\mathbf{C}(\mathbf{r_p})$ are the ground truth and output color of $\mathbf{p}$.

In practice, the integral in Equation (1) is approximated by numeric quadrature that samples a finite number of points along with the rays and computes the summation of radiances according to the estimated per-point transmittance. The sampling in NeRF follows a *coarse-to-fine* mechanism with two MLPs, *i.e.*, coarse network is queried on equally spaced samples whose outputs are utilized to sample another group of points for more accurate estimation and fine network is then queried on both groups of samples.

4 APPROACH

In this section, we introduce the details of NeRF-SR. The overall structure is presented in Figure 2. The supersampling strategy and patch refinement network will be introduced in Section 4.1 and Section 4.2.

4.1 Supersampling

NeRF optimizes a 3D radiance field by enforcing multi-view color consistency and samples rays based on camera poses and pixels locations in the training set. Although NeRF can be rendered at any resolution and retain great performance when the input images satisfy the Nyquist sampling rate, it is impossible in practice. Compared to the infinity possible incoming ray directions in the space, the sampling is quite sparse given limited input image observations. NeRF can create plausible novel views because the output resolution is the same as the input one and it relies on the interpolation property of neural networks. However, this becomes a problem when we render an image at a higher resolution than training images, specifically, there is a gap between the training and testing phase. Suppose a NeRF was trained on images of resolution $\text{H} \times \text{W}$,
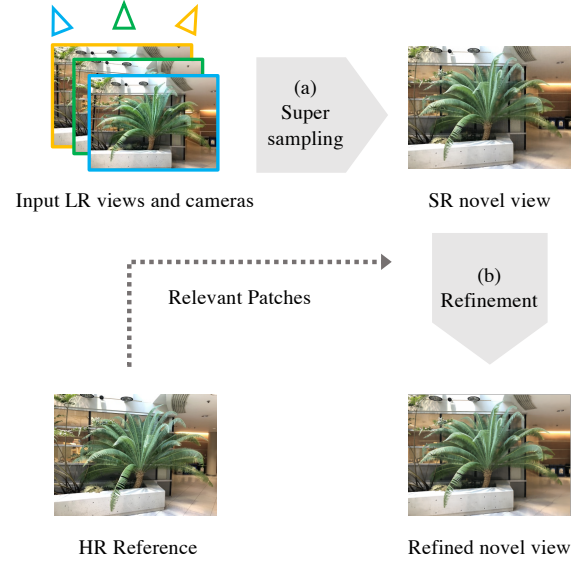

Figure 2: An overview of the proposed NeRF-SR that includes two components. (a), we adopt a super sampling strategy to produce super-resolution novel views from only low-resolution inputs. (b) Given an high-resolution reference at any viewpoint from which we utilize the depth map at hand to extract relevant patches, NeRF-SR generates more details for synthesized images.

the most straightforward way to reconstruct a training image on scale factor s, *i.e.*, an image of resolution $s\text{H} \times s\text{W}$ is sampling a grid of s^2 rays in an original pixel. Obviously, not only the sampled ray directions were never seen during training, but the pixel queried corresponds to a smaller region in the 3D space. Regarding this issue, we propose a supersampling strategy that tackles the problem of rendering SR images for NeRF. The intuition of supersampling is explained as follows and illustrated in Figure 3.

We start from the image formation process. The pixel values are mapped from scene irradiance through a *camera response function* (CRF). For simplification, we assume a pinhole camera model as in NeRF and consider ISO gain, shutter speed as implicit factors. Let $\mathcal{R}(\mathbf{p})$ denotes the set of all possible ray directions for pixel $\mathbf{p}$ from a training image, then:

$$C(\mathbf{p}) = f(E_{\mathcal{R}(\mathbf{p})}) \qquad (4)$$

where $C(\mathbf{p})$ indicates the color of $\mathbf{p}$, f is CRF, E is the incident irradiance over the area covered by $\mathbf{p}$, which is the integration of radiance over all incoming rays in $\mathbf{p}$. Although ideally the training ray directions should be sampled from $\mathcal{R}(\mathbf{p})$, it is both computational expensive and challenging for the network to fit this huge amount of data. Therefore, in our work, to super-resolve images at the scale of s, we first evenly split a pixel from training set into a $s \times s$ grid sub-pixels $\mathcal{S}(\mathbf{p})$. As in NeRF, we do not model CRF and output the color of each sub-pixel using a multi-layer perceptron directly. During training stage, ray directions for a pixel $\mathbf{p}$ will be sampled from the sub-pixels instead, denoted as $\mathcal{R}'(\mathbf{p}) = \{\mathbf{r_j} \mid \mathbf{j} \in \mathcal{S}(\mathbf{p})\} \subset \mathcal{R}(\mathbf{p})$.
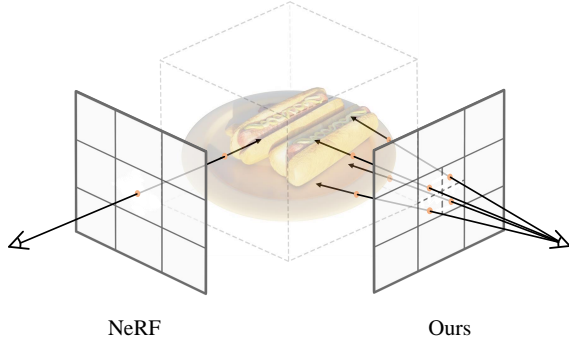
Figure 3: Original NeRF casts a single ray through a pixel (solid line) and performance MSE loss directly (left), while our method (right) splits a pixel into multiple sub-pixels (dash line) and draws a ray for each sub-pixel, then the radiances of sub-pixels will be averaged for MSE loss. Compared to vanilla NeRF, more 3D points in the scene can be corresponded and constrained in supersampling.

At inference stage, an $sH \times sW$ image can be directly obtained by directly rendering and organizing the sub-pixels, erasing the sampling gap between the training and testing phase.

Another concern is how to perform supervision with only ground truth images at dimension $H \times W$. Similar to the blind-SR problem, the degradation process from $sH \times sW$ is unknown and may be affected by many factors. Inspired by the graphics pipeline, we tackle this issue by compute the radiance for sub-pixels in $\mathcal{R}'(p)$ using Equation 1 and then average them to compare with the color of $\mathbf{p}$. Thus, Equation 3 can be extended as:

$$\mathcal{L}_{\text{MSE}} = \sum_{\mathbf{p} \in \mathcal{P}} \left\| \frac{1}{|\mathcal{R}'(\mathbf{p})|} \sum_{\mathbf{r}' \in \mathcal{R}'(\mathbf{p})} \hat{\mathbf{C}}(\mathbf{r}') - \mathbf{C}(\mathbf{r_p}) \right\|_2^2 \quad (5)$$

where $\mathcal{R}'(\mathbf{p})$ is the sub-pixel grid for pixel $\mathbf{p}$, $|\mathcal{R}'(\mathbf{p})|$ is the number of sub-pixels in $\mathcal{R}'(\mathbf{p})$, $\mathbf{r}'$ is the ray direction for a single sub-pixel, $\hat{\mathbf{C}}(\mathbf{r}')$ is the color of a sub-pixel predicted by the network. On the other hand, the LR images can be seen as downsampled from HR ones by averaging pixel color in a grid (We call it "average" kernel). This aborts any complex assumptions on the downsampling operation and make our method robust for various situations.

To summarize, supersampling extends original NeRF in two aspects: first it samples ray directions from $s \times s$ grid sub-pixels for pixel $\mathbf{p}$ instead of a single ray direction; second, it averages the color of the sub-pixels for supervision. In computer graphics, supersampling and averaging is often used in the rendering process to handle the problem of aliasing. In our work, we show that it fully exploits the cross-view consistency introduced by NeRF to a sub-pixel level, *i.e.*, a position can be corresponded through multiple viewpoints. While NeRF only shoots one ray for each pixel and optimizes points along that ray, supersampling constraints more positions in the 3D space and better utilize the multi-view information in input images. In other words, supersampling directly optimizes a denser radiance field at training time.

4.2 Patch-Based Refinement

With supersampling, the synthesized image achieves much better visual quality than vanilla NeRF. However, when the images for a scene do not have enough sub-pixel correspondence, the results of supersampling cannot find enough details for high-resolution synthesis. Also, often there are limited high-resolution images from which HR content are available for further improving the results.

Here, we present a patch-based refinement network to recover high-frequency details that works even in the *extreme* case, *i.e.*, only one HR reference is available, as shown in Figure 4. Our system is though not limited to one HR reference and can be easily extended to multiple HR settings. The core design consideration focuses on how to "blend" details on the reference image $\mathcal{I}_{\text{REF}}$ into NeRF synthesized SR images that already captured the overall structure. We adopt a patch-by-patch refine strategy that turns an SR patch $\widetilde{P}$ into the refined patch P. Other than $\widetilde{P}$, the input should also include an HR patch from $\mathcal{I}_{\text{REF}}$ that reveals how the objects or textures in $\widetilde{P}$ presents in high-resolution. However, due to occlusion and inaccuracy of depth estimation, multiple HR patches are required to cover the region in $\widetilde{P}$ and we use K patches $\{P^{\text{REF}}\}_{k=1}^{K}$ for reference. Also, patches in $\{P^{\text{REF}}\}_{k=1}^{K}$ cover larger regions than $\widetilde{P}$ and contain less relevant information. The refinement stage aims at local detail enhancement and well preserve the view consistent structure from super-sampling with the spatial information of depth predictions.

We use a U-Net based convolutional architecture for the refinement network, which has demonstrated its efficacy in several existing novel view synthesis methods [5, 41, 42]. In earlier attempts, we model the refinement procedure as an image-to-image translation [14] and find channel-wise stack $\widetilde{P}$ and $\{P^{\text{REF}}\}_{k=1}^{K}$ were unable to fit the training set perfectly. Therefore, inspired by [5, 41], we instead encode each patch respectively with an encoder consisting of seven convolutional layers. The decoder of the network takes as input the nearest-neighbor upsampled features from previous layers concatenated with both the encoded features of $\widetilde{P}$ and max-pooled features of $\{P^{\text{REF}}\}_{k=1}^{K}$ at the same spatial resolution. All convolutional layers are followed by a ReLU activation.

Training The training of the refinement network requires SR and HR patch pairs, which are only available at the camera pose of $\mathcal{I}_{\text{REF}}$. Therefore, $\widetilde{P}$ is randomly sampled from the SR image and P is the patch on $\mathcal{I}_{\text{REF}}$ at the same location. We perform perspective transformations to $\widetilde{P}$ and P as during testing, the input patches are mostly from different camera poses. Moreover, to account for the inaccuracy of reference patches at testing time, we sample $\{P^{\text{REF}}\}_{k=1}^{K}$ within a fixed window around P. In order to preserve the spatial structure of $\widetilde{P}$ while improving its quality, our objective function combines reconstruction loss $\mathcal{L}_{rec}$ and perceptual loss $\mathcal{L}_{per}$, where

$$\mathcal{L}_{\text{refine}} = \mathcal{L}_{rec} + \mathcal{L}_{per} = ||\widetilde{P} - P||_1 + \Sigma_l \lambda_l ||\phi_l(\widetilde{P}) - \phi_l(P)||_1 \quad (6)$$

ϕ_l is a set of layers in a pretrained VGG-19 and λl is the reciprocal of the number of neurons in layer l. Note that we adopt l_1-norm instead of MSE in $\mathcal{L}_{rec}$ because it is already minimized in supersampling and l_1-norm will sharpen the results.

Testing At inference time, given a patch $\widetilde{P}$ on synthesized image $\mathcal{I}_n$, we can find a high-resolution reference patch on reference image
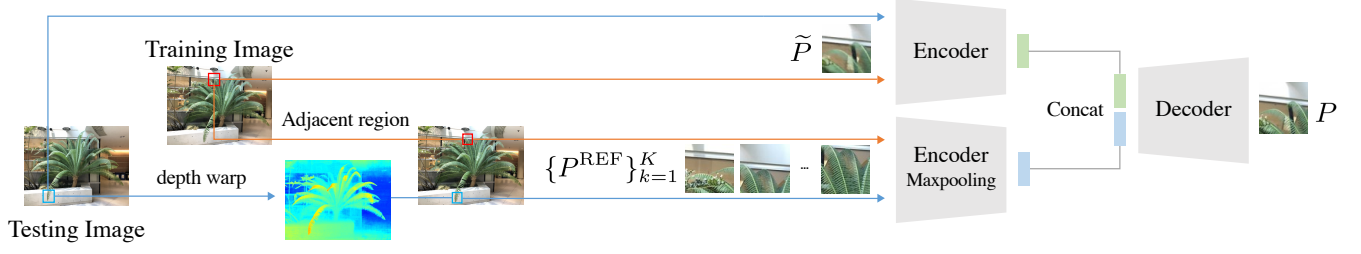
Figure 4: Our refinement module encodes synthesized patches $\widetilde{P}$ from images produced by supersampling and reference patches $\{P^{\text{REF}}\}_{k=1}^{K}$ from $\mathcal{I}_{\text{REF}}$. The encoded features of $\mathcal{I}_{\text{REF}}$ are maxpooled and concatenated with that of $\widetilde{P}$, which is then decoded to generate the refined patch. In the training phase, $\widetilde{P}$ is sampled from synthesized SR image at the camera pose of $\mathcal{I}_{\text{REF}}$ and $\{P^{\text{REF}}\}_{k=1}^{K}$ is sampled at adjacent regions. When testing, $\{P^{\text{REF}}\}_{k=1}^{K}$ is obtained via depth warping. (The input and output patches are zoomed for better illustration, zoom in to see the details on leaves after refinement)

$\mathcal{I}_{\text{REF}}$ for each pixel on $\widetilde{P}$:

$$P_{i,j}^{\text{REF}} = K_{\text{REF}} T(K_n^{-1} d_{i,j} \widetilde{P}_{i,j}) \tag{7}$$

where i, j denotes a location on patch $\widetilde{P}$, d is the estimated depth, T is the transformation between camera extrinsic matrices from $\mathcal{I}_n$ to $\mathcal{I}_{\text{REF}}$, and K_{REF} and K_n refer to the camera intrinsic matrices of $\mathcal{I}_{\text{REF}}$ and $\mathcal{I}_n$. Therefore, Equation (7) computes the 3D world coordinate of i, j based on $d_{i,j}$ and camera parameters, then backproject it to a pixel on $\mathcal{I}_{\text{REF}}$ and extract the corresponding patch at that location (points fall out of $\mathcal{I}_{\text{REF}}$ are discarded). In summary, to obtain the refined P, we first sample K patches from $\{P_{i,j}^{\text{REF}}\}$ to construct the set $\{P^{\text{REF}}\}_{k=1}^{K}$ and then input them together with $\widetilde{P}$ into the network. More details of the refinement network can be found in the supplementary material.

The training of NeRF requires correspondences of input images in the 3D space. As long as the HR reference falls in the camera frustum of input images, it can be easily wrapped to other views and bring enough details. Therefore, our refinement network is well-suited for any NeRF compatible dataset.

5 EXPERIMENTS

In this section, we provide both quantitative and qualitative comparisons to demonstrate the advantages of the proposed NeRF-SR. We first show results and analysis of super-sampling, and then demonstrate how the refinement network adds more details to it. Our result only with super-sampling is denoted as Ours-SS and our result after patch-based refinement is denoted as Ours-Refine.

5.1 Dataset and Metrics

To evaluate our methods, we train and test our model on the following datasets. We evaluate the quality of view synthesis with respect to ground truth from the same pose using three metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [58] and LPIPS[65].

Blender Dataset The Realistic Synthetic 360° of [31] (known as Blender dataset) contains 8 detailed synthetic objects with 100 images taken from virtual cameras arranged on a hemisphere pointed

inward. As in NeRF[32], for each scene we input 100 views for training and hold out 200 images for testing.

LLFF Dataset LLFF dataset[31, 32] consists of 8 real-world scenes that contain mainly forward-facing images. We train on all the images and report the average metrics on the whole set.

5.2 Training Details

In super-sampling, we implement all experiments on top of NeRF [32] using PyTorch. As we train on different image resolutions independently, for fair comparison we train blender dataset and LLFF dataset for respectively 20 epochs and 30 epochs, where each epoch contains an iteration of the whole training set. We choose Adam as the optimizer (with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$) with batch size set to 2048 (2048 rays a batch for all experimented scales) and learning rate decayed exponentially from $5 \cdot 10^{-4}$ to $5 \cdot 10^{-6}$. Following NeRF, NeRF-SR also uses a hierarchical sampling with the same size "coarse" and "fine" MLP. The number of coarse samples and fine samples are both set to 64.

5.3 Comparisons

Since there are no previous work that deals with super-resolving NeRF, we devise several reasonable baselines for comparisons, detailed as the following:

NeRF Vanilla NeRF is already capable of synthesising images at any resolution due to its implicit formation. Therefore, we train NeRF on LR inputs using the same hyperparameters in our method and directly render HR images.

NeRF-Bi aims to super-resolve a trained LR NeRF. We use the same trained model in the NeRF baseline, but render LR images directly and upsample them with the commonly used bicubic upsampling.

NeRF-Liif Liif [3] achieves state-of-the-art performance on continuous single image super-resolution. Similar to the NeRF-Bi baseline, we super-resolve LR images using pretrained liif model instead. Note that to the training process of liif requires LR-HR pairs, therefore it introduces external data priors.

NeRF-Swin SwinIR [24] is the start-of-the-art method on single image super-resolution. Like NeRF-Bi and NeRF-Liif, NeRF-Swin
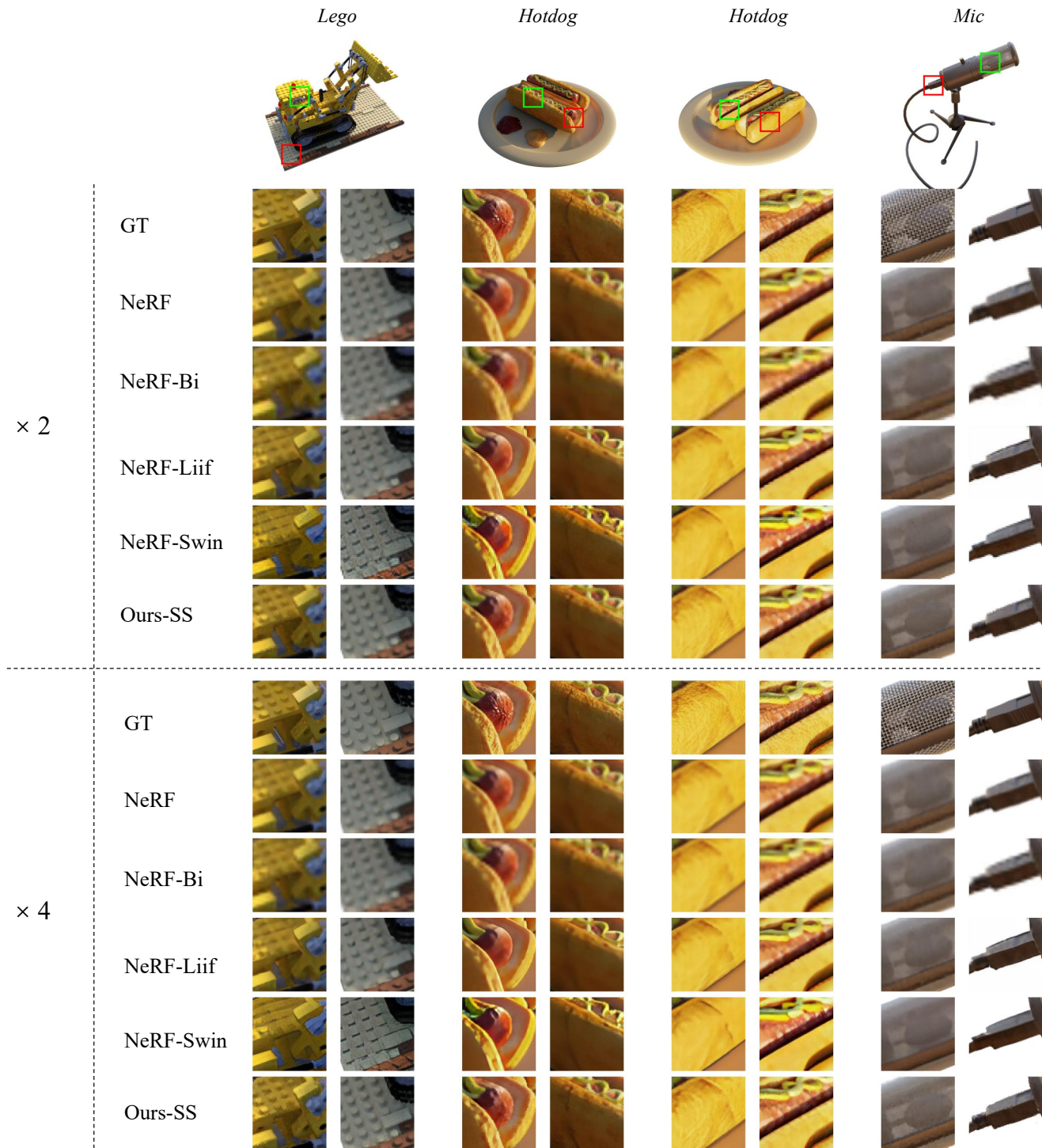
Figure 5: Qualitative comparison of blender dataset when the input images are 200×200 and upscale by 2 and 4. Note how NeRF-SR recovers correct details through super-sampling even when inputting low-resolution images, such as *Lego*'s gears, *Hotdog*'s sausage and sauce, *Mic*'s magnets and shiny brackets. Note NeRF-SR is able to synthesize consistently over different viewpoints, here we provide two for *Hotdog,* videos can be found on our website. Please zoom in for a better inspection of the results.

Method	Blender×2 (100 × 100)			Blender×4 (100 × 100)			Blender×2 (200 × 200)			Blender×4 (200 × 200)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [32]	<u>27.54</u>	<u>0.921</u>	0.100	<u>25.56</u>	0.881	0.170	<u>29.16</u>	<u>0.935</u>	0.077	<u>27.47</u>	0.910	0.128
NeRF-Bi	26.42	0.909	0.151	24.74	0.868	0.244	28.10	0.926	0.109	26.67	0.900	0.175
NeRF-Liif	27.07	0.919	<u>0.067</u>	25.36	<u>0.885</u>	0.125	28.81	0.934	<u>0.058</u>	27.34	<u>0.912</u>	0.096
NeRF-Swin	26.34	0.913	0.075	24.85	0.881	<u>0.108</u>	28.03	0.926	<u>0.058</u>	26.78	0.906	<u>0.086</u>
Ours-SS	**29.77**	**0.946**	**0.045**	**28.07**	**0.921**	**0.071**	**31.00**	**0.952**	**0.038**	**28.46**	**0.921**	**0.076**

Table 1: Quality metrics for novel view synthesis on blender dataset. We report PSNR/SSIM/LPIPS for scale factors ×2 and ×4 on two input resolutions (100 × 100 and 200 × 200) respectively.
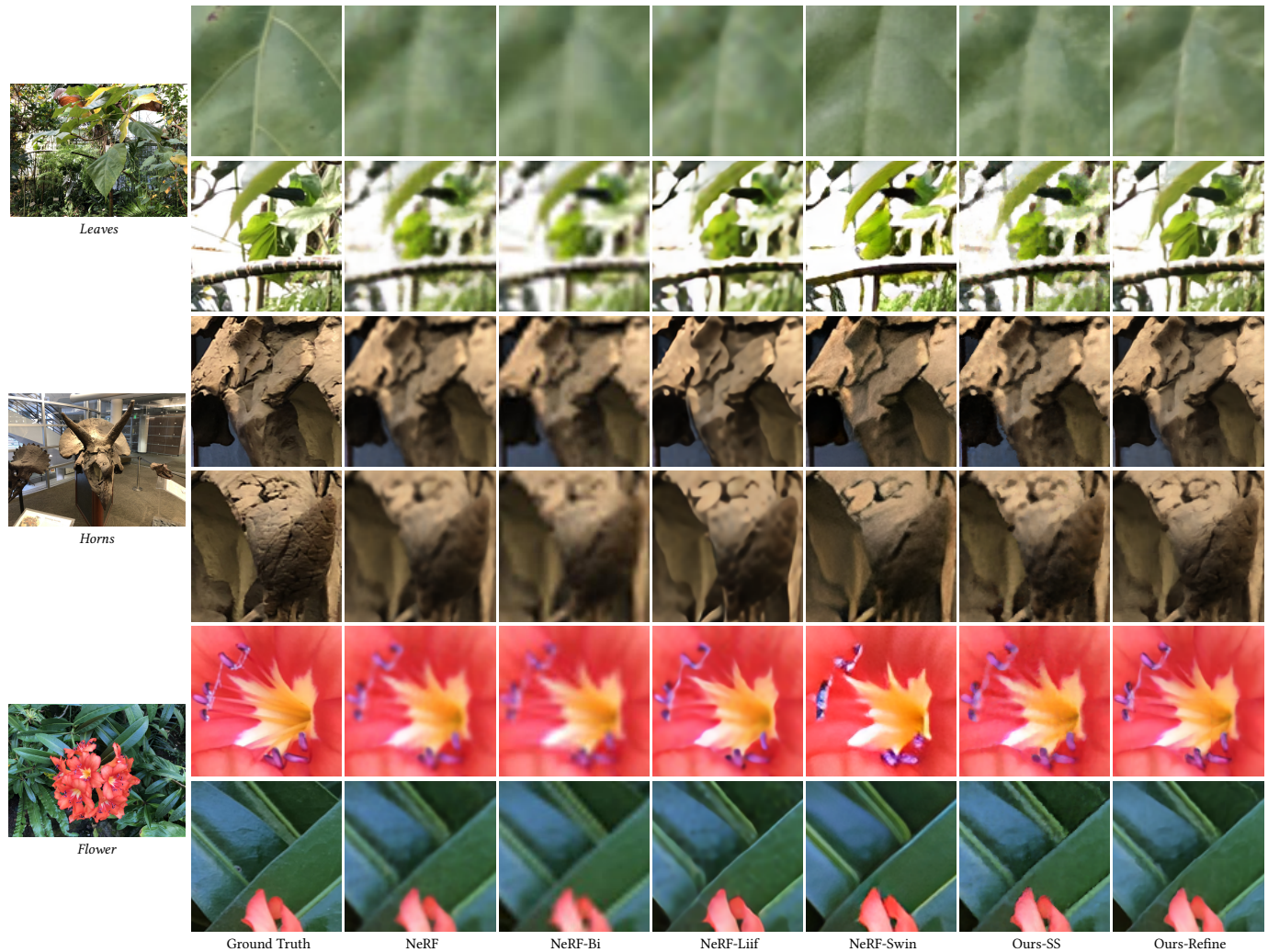

Figure 6: Qualitative comparison on LLFF dataset at an upscale of 4 between bicubic, NeRF, Ours-SS and Ours-Refine. NeRF-SR presents correct and clear texture in the leaves of *Leaves* and *Flowers* and fissures on *Horns*' ears and noses, which can further be enhanced using the refinement network. Please zoom in for better inspection of the results.

performs super-resolution on a LR NeRF with the released SwinIR models under the "Real-World Image Super-Resolution" setting, which has a training set of more than 10k LR-HR pairs.

5.4 Effectiveness of supersampling

For blender dataset, we super-sample on two resolutions: 100 × 100 and 200 × 200, and test scales ×2 and ×4. For the LLFF dataset, the input resolution is 504 × 378 and we also upscale by ×2 and ×4. The

downscaling of images in the dataset from original resolution to training resolution is done by the default Lanczos method in the Pillow package.

Figure 5 shows qualitative results for all methods on a subset of blender scenes. Renderings from NeRF-Bi exhibit correct global shapes but lack high-frequency details. Vanilla NeRF produces renderings that have more details than NeRF-Bi if the scene is already well-reconstructed at input resolution. However, it is still restricted by the information in the input image. NeRF-Liif can recover some details, but lacks enough texture. NeRF-SR find sub-pixel level correspondence through supersampling, which means missing details in the input can be found from other views that lie in the neighboring region in 3D space.

Quantitative results of blender dataset are summarized in Table 1. NeRF-SR outperforms other baselines in all scenarios. NeRF-Liif or NeRF-Swin have the second best LPIPS, providing good visual quality but cannot even compete with NeRF in PSNR and SSIM. The reason is maybe the blender dataset is synthetic and has different domain than the dataset it is trained on, resulting false prediction (see NeRF-Swin on *Lego* and *Hotdog*).

The qualitative and quantitative results for LLFF dataset are demonstrated in Figure 6 and Table 2 respectively. NeRF and NeRF-Bi suffers from blurry outputs. While NeRF-Liif and NeRF-Swin recovers some details and achieve satisfying visual quality (comparable LPIPS to Ours-SS) since they are trained on external datasets, they tend to be oversmooth and even predicts false color or geometry (See the leaves of *Flower* in Figure 6). NeRF-SR fill in the details on the complex scenes and outperforms other baselines significantly. Therefore, we can conclude that learning-based 2D baselines struggle to perform faithfully super-resolution, especially in the multi-view case.

In Section 4.1, we mentioned that the supervision is performed by comparing the average color of sub-pixels due to the unknown nature of the degradation process (We call it "average kernel"). However, in our experiments, the degradation kernel is actually Lanczos, resulting an asymmetric downscale and upscale operation. We further experiment on the condition that the degradation from high-resolution to input images is also "average kernel" for blender data at the resolution 100×100. Results show this symmetric downscale and upscale operation provides better renderings than asymmetric one. PSNR, SSIM, LPIPs are all improved to 30.94 dB, 0.956, 0.023 for scale $\times 2$ and 28.28 dB, 0.925 and 0.061 for $\times 4$ respectively. The sensitivity to the degradation process is similar to that exhibited in single-image super-resolution. Detailed Rendering can be found in the supplementary.

5.5 Refinement network

LLFF dataset contains real-world pictures that have a much more complex structure than the blender dataset, and super-sampling isn't enough for photorealistic renderings. We further boost its outputs with a refinement network introduced in Section 4.2. We use a fixed number of reference patches ($K = 8$) and the dimensions of patches are set to 64×64. While inferencing, the input images are divided into non-overlapping patches and stitched together after refinement. Without the loss of generosity, we set the reference image is to the first image in the dataset for all scenes, which

is omitted when calculating the metrics. The inference time of the refinement stage is neglibile compared to NeRF's volumetric rendering: for example, it takes about 48 seconds for NeRF's MLP to render a 1008×756 image, and it only takes another 1.3 seconds in the refinement stage on a single 1080Ti.

The quantitative results of refinement can be found in Table 2. After refinement, metrics are improved substantially at the scale of 4. For the scale of 2, PSNR increases only a bit after refining, a possible reason is that supersampling already learns a decent high-resolution neural radiance fields for small upscale factors and the refinement only improves subtle details (Please refer the supplementary for an example). However, we can see that LPIPS is still promoted, meaning the visual appearance improves. The problem doesn't occur for larger magnifications such as 4 since supersampling derives much fewer details from low-resolution inputs, making the refinement process necessary.

We demonstrate the renderings qualitatively before and after refining in Figure 6. It is clear to see that the refinement network boosts supersampling with texture details and edge sharpness.

Method	LLFF×2			LLFF×4		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [32]	26.36	0.805	0.225	24.47	0.701	0.388
NeRF-Bi	25.50	0.780	0.270	23.90	0.676	0.481
NeRF-Liif	<u>26.81</u>	<u>0.823</u>	<u>0.145</u>	<u>24.76</u>	<u>0.723</u>	0.292
NeRF-Swin	25.18	0.793	0.147	23.26	0.685	<u>0.247</u>
Ours-SS	**27.31**	**0.838**	0.139	**25.13**	**0.730**	0.244
Ours-Refine	**27.34**	**0.842**	**0.103**	**25.59**	**0.759**	**0.165**

Table 2: Quality metrics for view synthesis on LLFF dataset. We report PSNR/SSIM/LPIPS for scale factors ×2 and ×4 on input resolutions (504×378).

6 LIMITATIONS AND CONCLUSION

A major limitation of NeRF-SR is that it does not enjoy the nice arbitrary-scale property. It also introduces extra computation efficiency, albeit it consumes no more time than training a HR NeRF.

In conclusion, we presented NeRF-SR the first pipeline of HR novel view synthesis with mostly low resolution inputs and achieve photorealistic renderings without any external data. Specifically, we exploit the 3D consistency in NeRF from two perspectives: supersampling strategy that finds corresponding points through multi-views in sub-pixels and depth-guided refinement that hallucinates details from relevant patches on an HR reference image. Finally, region sensitive supersampling and generalized NeRF super-resolution may be explored for future works.

ACKNOWLEDGMENTS

REFERENCES

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. 2020. Neural point-based graphics. In *Computer Vision–ECCV 2020:*

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. Springer, 696–712.

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021.* IEEE, 5835–5844. https://doi.org/10.1109/ICCV48922.2021.00580

[3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. 2021. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8628–8638.

[4] Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5939–5948.

[5] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. 2019. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 7781–7790.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision.* Springer, 184–199.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38, 2 (2015), 295–307.

[8] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. 2020. Curriculum deepsdf. In *European Conference on Computer Vision.* Springer, 51–67.

[9] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2367–2376.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[11] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. 1996. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.* 43–54.

[12] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

[13] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. 2020. Learning a neural 3d texture space from 2d exemplars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8356–8364.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1125–1134.

[15] Wonbong Jang and Lourdes Agapito. 2021. CodeNeRF: Disentangled Neural Radiance Fields for Object Categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 12949–12958.

[16] James T Kajiya and Brian P Von Herzen. 1984. Ray tracing volume densities. *ACM SIGGRAPH computer graphics* 18, 3 (1984), 165–174.

[17] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–10.

[18] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 3907–3916.

[19] Kwang In Kim and Younghee Kwon. 2010. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence* 32, 6 (2010), 1127–1133.

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 4681–4690.

[21] Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.* 31–42.

[22] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. 2021. MINE: Towards Continuous Depth MPI with NeRF for Novel View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 12578–12588.

[23] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. 2020. Crowdsampling the plenoptic function. In *European Conference on Computer Vision.* Springer, 178–196.

[24] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 1833–1844.

[25] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. *arXiv preprint arXiv:2104.06405* (2021).

[26] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems* 33 (2020), 15651–15663.

[27] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. 2019. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 7708–7717.

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7210–7219.

[29] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition.* 2437–2445.

[30] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4460–4470.

[31] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision.* Springer, 405–421.

[33] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3504–3515.

[34] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 2019. 3d ken burns effect from a single image. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–15.

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 165–174.

[36] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 5865–5874.

[37] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. 2021. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *arXiv preprint arXiv:2106.13228* (2021).

[38] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. 2020. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16.* Springer, 523–540.

[39] Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

[40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 10318–10327.

[41] Gernot Riegler and Vladlen Koltun. 2020. Free view synthesis. In *European Conference on Computer Vision.* Springer, 623–640.

[42] Gernot Riegler and Vladlen Koltun. 2021. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 12216–12225.

[43] Chris Rockwell, David F Fouhey, and Justin Johnson. 2021. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 14104–14113.

[44] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2304–2314.

[45] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. 2017. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision.* 4491–4500.

[46] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 4104–4113.

[47] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3d photography using context-aware layered depth inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8028–8038.

[48] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. 2019. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2437–2446.

[49] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618* (2019).

[50] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 175–184.

[51] Jian Sun, Zongben Xu, and Heung-Yeung Shum. 2008. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

[52] Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

[53] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. 2020. Image-guided neural object rendering. In *8th International Conference on Learning Representations*. OpenReview. net.

[54] Alex Trevithick and Bo Yang. 2021. GRF: Learning a General Radiance Field for 3D Representation and Rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15182–15192.

[55] Richard Tucker and Noah Snavely. 2020. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 551–560.

[56] Shubham Tulsiani, Richard Tucker, and Noah Snavely. 2018. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 302–317.

[57] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. 2015. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE international conference on computer vision*. 370–378.

[58] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. Ieee, 1398–1402.

[59] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. 2021. NeRF−−: Neural Radiance Fields Without Known Camera Parameters. *arXiv preprint arXiv:2102.07064* (2021).

[60] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7467–7477.

[61] Yanchun Xie, Jimin Xiao, Mingjie Sun, Chao Yao, and Kaizhu Huang. 2020. Feature representation matters: End-to-end learning for reference-based image super-resolution. In *European Conference on Computer Vision*. Springer, 230–245.

[62] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. 2020. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5791–5800.

[63] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4578–4587.

[64] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492* (2020).

[65] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

[66] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. 2019. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7982–7991.

[67] Haitian Zheng, Minghao Guo, Haoqian Wang, Yebin Liu, and Lu Fang. 2017. Combining exemplar-based approach and learning-based approach for light field super-resolution using a hybrid imaging system. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2481–2486.

[68] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

[69] M Zontak and M Irani. 2011. Internal statistics of a single natural image. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. 977–984.