Hybrid Neural Rendering for Large-Scale Scenes with Motion Blur

Peng Dai[1*] Yinda Zhang[2*] Xin Yu[1] Xiaoyang Lyu[1] Xiaojuan Qi[1†]
[1]The University of Hong Kong [2] Google

Abstract

Rendering novel view images is highly desirable for many applications. Despite recent progress, it remains challenging to render high-fidelity and view-consistent novel views of large-scale scenes from in-the-wild images with inevitable artifacts (e.g., motion blur). To this end, we develop a hybrid neural rendering model that makes image-based representation and neural 3D representation join forces to render high-quality, view-consistent images. Besides, images captured in the wild inevitably contain artifacts, such as motion blur, which deteriorates the quality of rendered images. Accordingly, we propose strategies to simulate blur effects on the rendered images to mitigate the negative influence of blurriness images and reduce their importance during training based on precomputed quality-aware weights. Extensive experiments on real and synthetic data demonstrate our model surpasses state-of-the-art point-based methods for novel view synthesis. The code is available at `https://daipengwa.github.io/Hybrid-Rendering-ProjectPage/`.

1. Introduction

Novel-view synthesis of a scene is one critical feature required by various applications, *e.g.*, AR/VR, robotics, and video games, to name a few. Neural radiance field (NeRF) [23] and its follow-up works [3, 19, 24, 39, 43, 47] enable high-quality view synthesis on objects or synthetic data. However, synthesizing high-fidelity and view-consistent novel view images of real-world large-scale scenes remains challenging, especially in the presence of inevitable artifacts from the data-capturing process, such as motion blur (see Figure 1 & supplementary material).

To improve novel view synthesis, mainstream research can be mainly categorized into two lines. One line of methods directly resorts to features from training data to synthesize novel view images [4, 11, 29, 40], namely image-based rendering. By directly leveraging rich high-quality features from neighboring high-resolution images, these methods have a better chance of generating high-fidelity images with distinctive details. Nevertheless, the generated images often lack consistency due to the absence of global structural regularization, and boundary image pixels often contain serious artifacts. Another line of work attempts to equip NeRF with explicit 3D representations in the form of point cloud [28, 43], surface mesh [30, 44] or voxel grid features [9, 19, 46], namely neural 3D representation. Thanks to the global geometric regularization from explicit 3D representations, they can efficiently synthesize consistent novel view images but yet struggle with producing high-fidelity images in large-scale scenes (see the blurry images from Point-NeRF [43] in Fig. 1). This may be caused by low-resolution 3D representations [19], noisy geometries [1, 7], imperfect camera calibrations [2], or inaccurate rendering formulas [3], which make encoding a large-scale scene into a global neural 3D representation non-trivial and inevitably loses high-frequency information.

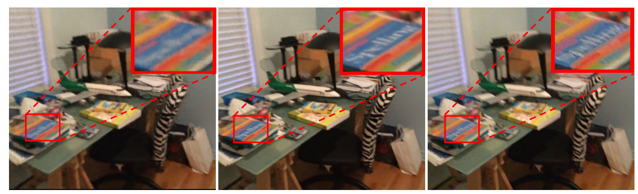

Figure 1. Our hybrid neural rendering model generates high-fidelity novel view images. Please note characters in the book where the result of Point-Nerf is blurry and the GT is contaminated by blur artifacts.

Albeit advancing the field, the above work all suffer immediately from low-quality training data, *e.g.*, blurry images. Recently, Deblur-NeRF [21] aims to address the problem of blurry training data and proposed a pipeline to simulate blurs by querying multiple auxiliary rays, which, however, is computation and memory inefficient, hindering their applicability in large-scale scenes.

In this paper, we aim at synthesizing high-fidelity and view-consistent novel view images in large-scale scenes using in-the-wild unsatisfactory data, *e.g.*, blurry data. First, to simultaneously address high fidelity and view consistency, we put forward a hybrid neural rendering approach that enjoys the merits of both image-based representation and neural 3D representation. Our fundamental design

*Equal contribution
†Corresponding author

centers around a 3D-guided neural feature fusion module, which employs view-consistent neural 3D features to integrate high-fidelity 2D image features, resulting in a hybrid feature representation that preserves view consistency whilst simultaneously upholding quality. Besides, to avoid the optimization of the hybrid representation being biased toward one modality, we develop a random feature drop strategy to ensure that features from different modalities can all be well optimized.

Second, to effectively train the hybrid model with unsatisfactory in-the-wild data, we design a blur simulation and detection approach to alleviate the negative impact of low-quality data on model training. Specifically, the blur simulation module injects blur into the rendered image to mimic the real-world blurry effects. In this way, the blurred image can be directly compared with the blurry reference image while providing blur-free supervisory signals to train the hybrid model. Besides, to further alleviate the influence of blurry images, we design a content-aware blur detection approach to robustly assess the blurriness scores of images. The calculated scores are further used to adjust the importance of samples during training. In our study, we primarily focus on the blur artifact due to its prevalence in real-world data (*e.g.*, ScanNet); however, our "simulate-and-detect" approach can also be applied to address other artifacts.

While our model is built upon the state-of-the-art 3D- and image-based neural rendering models, our contribution falls mainly on studying their combinatorial benefits and bridging the gap between NeRF and unsatisfactory data captured in the wild. Our major contributions can be summarized as follows.

- We study a hybrid neural rendering model for synthesizing high-fidelity and consistent novel-view images.
- We design blur simulation and detection strategies that facilitate offering blur-free training signals for optimizing the hybrid rendering model.
- Extensive experiments on real (*i.e.*, ScanNet [5]) and synthetic data (*i.e.*, Habitat-sim [22,35]) showcase that our method outperforms state-of-the-art point-based methods designed for novel view synthesis.

2. Related Works

Neural Radiance Field NeRF [23] encodes the object or scene into an MLP and synthesizes novel view images through volume rendering [15]. Later works extend NeRF for object manipulation [14, 44, 45, 50] and dynamic scene modeling [18, 25, 26], etc. Recent work [19, 43, 51] has started incorporating explicit 3D representations into NeRF training to support large-scale scenes and improve rendering details and speed. For example, Liu et al. [19] enhance NeRF's capabilities by storing neural features in a voxel-based representation, which generates images with rich details. Similarly, Xu et al. [43] utilize a point-based neural radiance field in cooperation with point growing and pruning, which substantially speeds up training and improves the quality of the rendered image. Unlike the methods described above, we deliver a hybrid framework leveraging the advantages of neural 3D representation and image-based representation to yield high-quality images.

Image-Based Rendering Image-based rendering is a well-known and long-standing technique [8, 17] for generating novel view images. A typical pipeline is to identify a few nearby images, warp them to the target viewpoint, and then blend them to create the output [11, 12, 29]. Recently, image-based rendering methods collaborating with volume rendering have been developed for generalization across scenes [4, 40, 47]. For instance, IBRNet [40] employs extracted image features from neighboring images to directly predict target views without requiring per-scene optimization [23]. Since image-based rendering can directly use the rich textures from images, it typically converges faster. However, it generally suffers from temporal inconsistencies. Instead, we apply the globally consistent neural 3D feature to drive the blending process in this work, improving the consistency of rendered image sequences.

Rendering with Artifacts For the in-the-wild environments, it is almost impossible to capture artifact-free training data due to motion blurs, noise, and environmental factors, which can adversely affect rendering quality. One solution is to restore contaminated images first [6, 33, 34, 38, 41, 42, 48], and then use restored images for training. However, it is a challenging problem to maintain the view consistency of restored images [13] as a pre-trained network is used to process each frame independently. Recently, some works [10, 21, 31] have attempted to simulate the image degradation process for image restoration during training. For example, to remove reflections, Guo et al. [10] propose incorporating an auxiliary MLP to model the reflection effects, which is removed during inference. Rückert et al. [31] propose to learn exposure-related parameters and response functions for synthesizing HDR images from training images with various exposures. The work most related to us is Deblur-NeRF [21], which uses auxiliary rays to simulate blurs for each training image which, however, sacrifices computation efficiency. Instead of sampling extra rays, we propose to down-weight the importance of blurry images and design a simple and efficient blur simulation method, resulting in faster training and better results.

3. Method

Given RGB-D image sequences with inevitable in-the-wild artifacts, our approach aims to render high-quality and

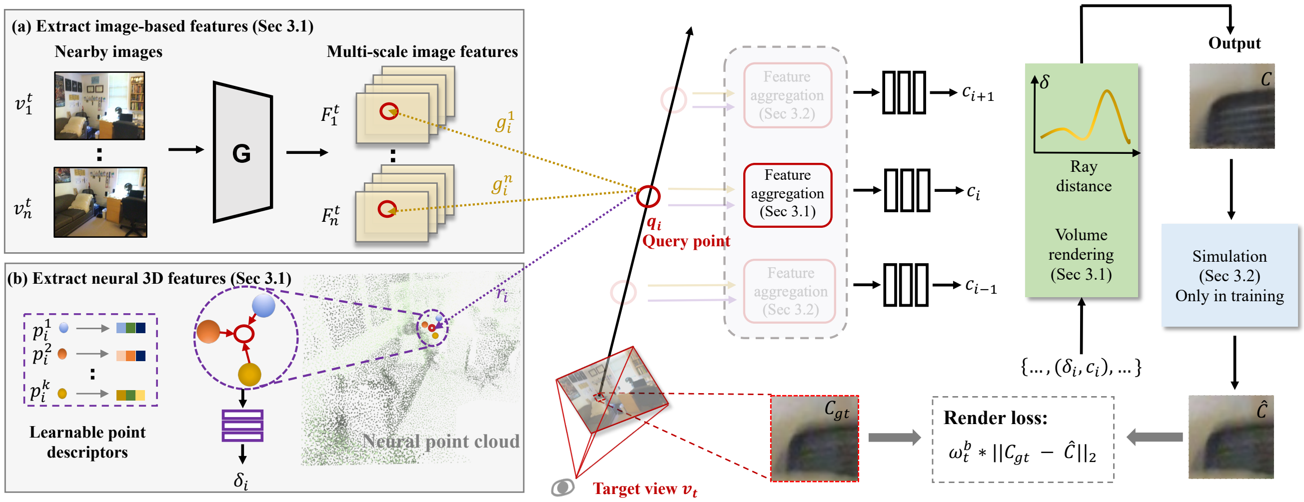

Figure 2. An overview of our hybrid neural rendering model $\mathcal{H}$. For each query point q_i on a ray cast from the target view v_t, it has two modalities of features, *i.e.*, (a) the image-based features $\{g_i^1, ..., g_i^n\}$ extracted from the n nearby images $\{v_1^t, ..., v_n^t\}$ and (b) the neural 3D feature r_i interpolated from k neighboring point descriptors $\{p_i^1, ..., p_i^k\}$. To generate high-quality and consistent novel view images, our hybrid neural rendering aggregates and benefits from both features from two modalities. To handle blur artifacts in the reference images, we simulate blur effects on the rendered image patch C to obtain $\hat{C}$, and then calculate the rendering loss with the ground-truth image patch C_{gt}. During training, we also down-weight the importance of images contaminated by artifacts according to the pre-computed quality-aware weights ω_t^b (see Sec. 3.2).

consistent novel view images. In our study, we consider motion blur as the major artifact due to its ubiquity in data captured with hand-held devices. An overview of our model is shown in Fig. 2. First, we put forward a hybrid neural rendering model $\mathcal{H}$ that incorporates neural features extracted from images and a geometry-aware neural radiance field (*e.g.*, Point-NeRF) for producing high-quality and view-consistent synthesis results (see Sec. 3.1). Then, to produce blur-free supervisory signals for training the hybrid model, we develop a blur simulation module and a content-aware blur detection strategy to alleviate the negative impacts of blurry ground-truth reference images (see Sec. 3.2). At last, we introduce the loss functions and optimization strategies for training our models (see Sec. 3.3).

3.1. Hybrid Neural Rendering Model

Our hybrid neural rendering model is designed to combine image-based representation and the geometry-based neural radiance field for faithful and view-consistent synthesis. It consists of a neural feature extraction module to harvest information from two kinds of representations, and a neural feature fusion module to aggregate extracted neural features in a data-driven manner. Given the aggregated features, our approach renders output images based on volume rendering. During training, we design a random drop strategy to avoid the optimization being dominated by one of the two representations.

Neural Feature Extraction As shown in Fig. 2, for each query point q_i on a ray cast from a target view v_t, we extract two modalities of features – image-based features and neural 3D features, described as follows.

Image-based features (Fig. 2 (a)): First, we use a lightweight CNN with down-sampling layers to extract multiscale image features $\{F_1^t, F_2^t, ..., F_n^t\}$ from n nearby views $\{v_1^t, v_2^t, ..., v_n^t\}$. Then, the query point q_i is projected to these nearby views, and features $\{F_1^t(q_i), F_2^t(q_i), ..., F_n^t(q_i)\}$ at the projected point location will be used to construct the image-based features for rendering. Following IBRNet [40], we additionally add image color $v_j^t(q_i)$ and deviations of view directions $\Delta d_j^t(q_i)$ to image-based features. As a result, for each query point q_i, its image-based feature representation is $g_i = \{g_i^1, g_i^2, ..., g_i^n\}$ where g_i^j is the combination of $F_j^t(q_i)$, $v_j^t(q_i)$, and $\Delta d_j^t(q_i)$.

Neural 3D features (Fig. 2 (b)): We adopt a point-based neural 3D representation [1, 7, 43] due to the wide application and high availability of point clouds. Following Point-NeRF [43], we aggregate features from multi-view depth maps to obtain point-based 3D representations, *i.e.* each point is described by a learnable descriptor. Then the neural 3D feature r_i is obtained by interpolating descriptors from its k-nearest neighborhoods $\{p_i^1, p_i^2, ..., p_i^k\}$. Note that the point-based representation can be replaced with other geometry-based representations, such as voxel-based or mesh-based representations [19, 44].

Neural Feature Aggregation As shown in Fig. 3, given

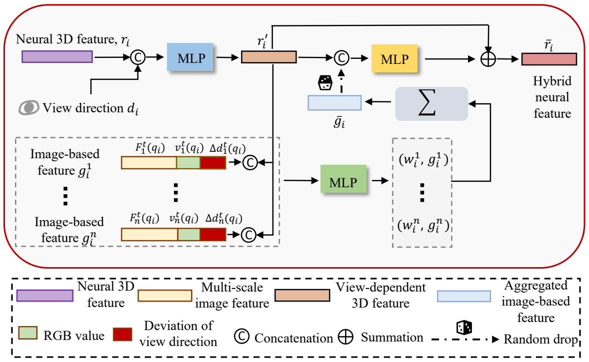

Figure 3. An overview of our feature aggregation module. The neural 3D feature r_i and multiple image-based features g_i^n are aggregated to generate a hybrid neural feature $\hat{r}_i$.

image-based features g_i from n nearby views and neural 3D features r_i, we design a learnable method to aggregate them to form a hybrid feature $\overline{r}_i$ for each query point q_i,

First, the neural 3D feature r_i combined with the view direction d_i is fed into an MLP to produce a view-dependent neural 3D feature $r_i' = \text{MLP}(r_i, d_i)$. Then, as the neural 3D feature consistently maintains global information and is free from view occlusions, we use it together with the image-based features to generate aggregation weights $\{\omega_i^1, \omega_i^2, ...\omega_i^n\}$ via an MLP layer (*i.e.*, $\omega_i^j = \text{MLP}(r_i', g_i^j)$). Further, the aggregation weights are used to combine $\{g_i^1, g_i^2, ..., g_i^n\}$ to form an aggregated image feature $\overline{g}_i$ following Eq. (1):

$$\overline{g}_i = \sum_{j=1}^{n} (\frac{\omega_i^j}{\gamma_i} \times g_i^j), \text{ where } \gamma_i = \sum_{j=1}^{n} \omega_i^j. \quad (1)$$

Finally, we learn a residual term for r_i' to get the final hybrid neural feature $\overline{r}_i$. This is achieved by enhancing the neural 3D feature r_i' using the aggregated image features $\overline{g}_i$, which can be described as:

$$\overline{r}_i = r_i' + \text{MLP}(r_i', \overline{g}_i). \quad (2)$$

Volume Rendering As illustrated in Fig. 2, we use k nearby geometric-consistent point descriptors $\{p_i^1, p_i^2, ..., p_i^k\}$ to predict volume density δ_i considering the view-independent nature of 3D geometry. The radiance values c_i are estimated through our hybrid neural features $\overline{r}_i$, which contain rich details. Then, we apply the volume rendering [23] to get the output color c of each ray following Eq. 3:

$$\begin{aligned} c &= \sum_{i=1}^{M} \tau_i (1 - \exp(-\delta_i \Delta_i)) c_i, \\ \tau_i &= \exp(-\sum_{t=1}^{i-1} \delta_t \Delta_t). \end{aligned} \quad (3)$$

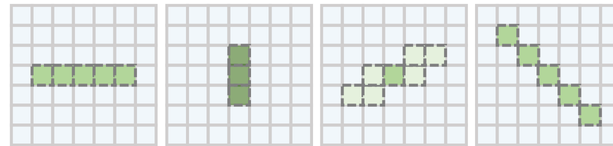

Figure 4. Examples of blur kernels. The pre-defined blur kernels have different moving directions and distances.

Here, M indicates the number of query points on a ray; Δ_i represents the distance between two adjacent query points along the ray, and the τ_i means volume transmittance.

Random Drop We develop two random drop strategies that randomly drop image features during optimization to ensure both modalities of features can be well-optimized: 1) the *ray-based random drop* will drop all image features on randomly selected rays; 2) the *query-point-based random drop* will randomly select query points on all rays and then remove all image features on them. The motivation behind the random drop is that we find the optimization of the hybrid representation can be easily dominated by image features, leaving neural 3D features poorly trained. This is because the image features are very similar to the reference images and are thus more easily optimized. Unless otherwise specified, we adopt the ray-based random drop during training. In the experiment part (see Fig. 10), we show the effects of the two strategies.

3.2. Blur Simulation and Detection

We propose two complementary strategies to address the negative influence of blurry reference images on optimizing the hybrid neural rendering model. First, we design a simulation method that simulates blur effects on the rendered image patch C to imitate the blur effects of the reference image patch C_{gt}. By comparing the blurred image patch $\hat{C}$ with the reference image patch during training, the sharpness of the rendered images can be preserved. Second, we develop a content-aware detection method to pre-compute the blurriness scores of reference images and down-weight the importance of blurry images based on the calculated scores. The two strategies work collectively to address the data quality challenge.

Blur Simulation To simulate motion blur, we assume that the camera moves in one direction through a certain distance while capturing high frame rate videos. Specifically, we take into account N_v directions and N_d distances for creating blur kernels $(B_i | i = 0, ..., N_v \times N_d)$ that are used to simulate blurs, and some examples of blur kernels are shown in Fig. 4. When $i = 0$, it means no blur simulation. To determine which blur kernel approximates the blur effects best, we first apply all blur kernels to the rendered results to obtain the blurred image patches $\hat{C}_i = \text{Conv}(C, B_i)$, and then choose the blur kernel i that yields an output patch $\hat{C}_i$ with the minimum photo-metric loss w.r.t the reference image patch C_{gt}. This process is described as:

$$L_i = ||\hat{C}_i - C_{\text{gt}}||_2, \\ L_{\text{color}} = \min\{L_i | i = 0, ..., N_v \times N_d\}. \tag{4}$$

Because our blur simulation does not need to render extra rays as in Deblur-NeRF [21], it runs faster and is more memory efficient. This blur simulation process is removed during inference to produce sharp images C.

Content-Aware Blur Detection In addition, we also down-weight the contribution of blurry images based on the blurriness score (a smaller value indicates more severe blur artifacts). However, we find that the "variation of the Laplacian" [27] method used to compute blurriness scores is prone to be influenced by image contents, thus unsuitable for scoring the reference images directly. As shown in the left of Fig. 5, the upper image is sharper than the bottom one but has a lower blurriness score. This is because the upper image contains more textureless contents (*i.e.*, the floor).

To exclude the influence of image contents, we develop a content-aware blur detection approach, which outputs accurate blurriness scores by scoring the overlapping regions. As shown in Fig. 5 right, our method first takes two neighboring images $\{I_t, I_{t+1}\}$ as inputs and estimates their overlapping regions (blue areas in Fig. 5) using optical flow [37]. Then, it returns two images' blurriness scores $\{S_t^1, S_{t+1}^1\}$ calculated from the overlapping regions. Next, to compute the blurriness score of image I_{t+2}, we use another image pair $\{I_{t+1}, I_{t+2}\}$ and repeat the process above to obtain two new blurriness scores $\{S_{t+1}^2, S_{t+2}^1\}$. Considering different overlapping regions in an image (*e.g.*, blue and red regions of I_{t+1} in Fig. 5) will lead to different blurriness scores S_{t+1}^1 and S_{t+1}^2, we align them by scaling S_{t+1}^2 to S_{t+1}^1. Correspondingly, the blurriness score of I_{t+2} is scaled following $S_{t+2}^1 = S_{t+1}^1 / S_{t+1}^2 \times S_{t+2}^1$. Similarly, the blurriness scores of other images can be computed. Please refer to the supplementary file for details. Finally, we convert blurriness scores into quality-aware weights ω_t^b following:

$$\omega_t^b = (\frac{N \times S_t^1}{\sum_{t=0}^{N} S_t^1})^{\alpha}, \tag{5}$$

where N represents the number of images, and $\alpha \geq 0$ is a hyper-parameter used to adjust the distribution of quality-aware image weights. These weights are further applied to the training objective in Sec. 3.3 to down-weight the importance of blurry images. Alternatively, you can use ω_t^b as sampling probabilities to sample training images.

3.3. Optimization

Our training objective consists of a photometric loss $\mathcal{L}_{\text{color}}$ in Eq. (4) that requires the rendered image patch C after blur simulation $\hat{C}$ to have the same appearance as the reference image patch C_{gt}; and a sparsity loss $\mathcal{L}_{\text{sparse}}$ [20,43] that encourages each point to have a confidence of 0 or 1 for the follow-up point pruning and growing operations. Following Point-NeRF [43], the point growing and pruning operations are applied every 10k iterations. After incorporating the quality-aware design (ω_t^b in Sec. 3.2), the final training objective is defined as:

$$\mathcal{L}_t = \omega_t^b(\mathcal{L}_{\text{color}} + \beta\mathcal{L}_{\text{sparse}}), \tag{6}$$

where $\beta = 0.002$ is used to balance different loss terms and ω_i^b is the estimated blurriness score to down weight blurry images (see Section 3.2).

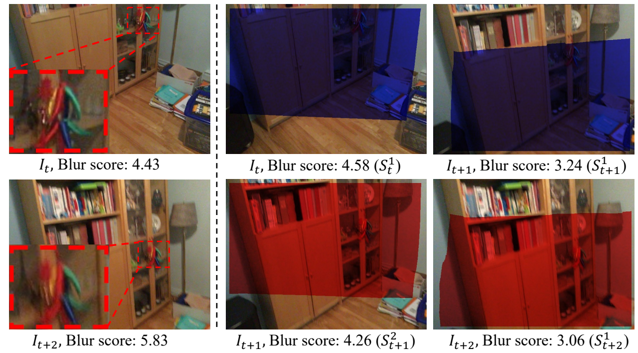

Figure 5. Content-aware blur detection. Left: the blurriness score [27] (large is sharper) is highly affected by image contents, and is usually low when the image contains textureless contents (*e.g.*, floor). Right: the content-aware blur detection computes blurriness scores on overlapping regions of two images, thus obtaining more accurate scores.

4. Experiments

4.1. Implementation Details

Network and Training The 2D CNN (G) used to extract image features has three down-sampling layers, and the point-based neural 3D representation is constructed following Point-NeRF [5]. We select four neighboring frames ($n = 4$) and eight nearest point descriptors ($k = 8$) to extract neural features. We train our models using the Adam [16] optimizer with an initial learning rate of 0.0005. A total of 200k iterations are used for training.

Blur Simulation We build our blur kernels considering $N_v = 4+8$ directions (*i.e.*, 'left-right', 'up-down', 'top left-bottom down', and 'bottom left-top right'; both symmetrical and asymmetrical) and three moving distances $N_d = 3$ (*i.e.*, 1, 2, 4). To apply blur simulation, we sample 8×8 patches with dilations [32] during the training, and the α in Eq. (5) is set as 1.

Dataset We conduct our experiments on ScanNet [5] and synthetic data generated from Habitat-sim [22]. 1) Scan-

Net [5] contains RGB-D image sequences captured in large-scale indoor scenes with handheld sensors. Following Point-NeRF, we conduct experiments on "Scene0101_04" and "Scene0241_01" and select every fifth image for training and the remaining images for testing. Note that images in the ScanNet are blurry, which is not suitable for quantitatively evaluating the sharpness of rendered images. Thus, we additionally evaluate our method using synthetic data. 2) Habitat-sim is a simulator [22,36] that synthesizes blur-free RGB-D sequences of large-scale scenes (*i.e.*, 'VangoRoom' and 'LivingRoom' [35]). We then add motion blurs to the synthesized training sets. Please see the supplementary file for details.

Baselines We compare our method with other representative image-based and neural-radiance-based novel view synthesis approaches, including: 1) NeRF [23]; 2) IBRNet [40] which combines image-based rendering with volume rendering and generates high-quality novel view images without using depth; 3) NPBG [1] which renders images using a U-Net-like design by rasterizing point descriptors onto the image plane 4) Point-NeRF [43], which is the state-of-the-art point-based method for novel view synthesis combining point-based neural representation and neural radiance field with volume rendering; and 5) Deblur-NeRF [21] which improves the sharpness of rendered images by simulating the blurring process with a deformable sparse kernel module.

4.2. Results on ScanNet

Quantitative comparisons with other baselines in terms of PSNR, SSIM, and LPIPS [49] are reported in Table 1. Our hybrid neural rendering design "Ours (H)" outperforms previous methods by enhancing the quality of neural 3D representations. However, the PSNR and SSIM drop in the full version of our method "Ours". This is because our blur-handling modules mimic blurriness effects and down weight blur images, enabling the model learn from clean supervision. However, since this differs from the original training data distribution, the model may not fit the evaluation metric well. Moreover, Deblur-NeRF delivers a low PSNR because it tends to introduce misalignment between rendered and reference images.

We show qualitative comparisons in Fig. 6. Our method can render high-quality novel view images while other baselines suffer severely from blurriness and distortions. For example, the clock on the wall is distorted with IBRNet, and the book generated by Point-NeRF is blurry. In contrast, our model produces results with clear characters in the book (Fig. 6 "Ours (H)"), validating the efficacy of our hybrid representation. Further, the rendered images become sharper when using our design to handle blur artifacts; please notice the human face on the poster (Fig. 6 "Ours"). To better demonstrate the efficacy of our approach

	Scene101_04			Scene241_01		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Point-NeRF [43]	29.88	0.913	0.203	30.54	0.910	0.236
IBRNet [40]	29.55	0.811	0.307	21.49	0.755	0.368
NPBG [1]	26.33	0.871	0.187	27.34	0.841	**0.188**
Deblur-NeRF [21]	24.55	0.693	0.308	20.66	0.652	0.401
NeRF [23]	27.16	0.730	0.350	21.69	0.610	0.494
Ours (H)	**30.33**	**0.919**	0.186	**31.25**	**0.918**	0.218
Ours	29.33	0.909	**0.181**	30.78	0.914	0.206

Table 1. Quantitative comparisons on ScanNet. "Ours (H)": use hybrid neural rendering without handling blur artifacts. We use PSNR, SSIM, and LPIPS to evaluate the rendering quality (↓: small is better; ↑: large is better). Our method outperforms all other baselines by a large margin, especially on PSNR. Note that the full version of our method ("Ours") is worse on the PSNR and SSIM, this is because the reference images in ScanNet are blurry.

in rendering consistent results, we provide videos in the supplementary file: our results are more temporally consistent than the image-based rendering (*i.e.*, IBRNet), thanks to the globally consistent neural 3D features.

4.3. Results on Synthetic Data

We conduct experiments on the synthetic data to validate our designs to handle blurriness. In particular, we incorporate our designs into two different frameworks (*i.e.*, NeRF and Point-NeRF) to show its generalization ability. Here, we remove the image-based rendering branch on the NeRF-based framework for fair comparisons. Fig. 7 shows that our method significantly enhances the sharpness of rendered images compared to NeRF and Point-NeRF, which are also confirmed in Table 2. Notably, images from Deblur-NeRF contain more details than NeRF but suffer from distorted image structures, such as the blinds and the table leg. This is because the learning of ray deformation is under-constrained with too many degrees of freedom and thus prone to corrupting original structures. Our easy-to-plug-in method outperforms Deblur-NeRF on PSNR and SSIM and delivers competitive performance on LPIPS. It is worth noting that to achieve the above results, NeRF takes 4.5 hours, while our method takes 4.6 hours. Thus, the increase in training time brought by blur simulation is negligible. However, Deblur-NeRF needs 8.5 hours which incurs much more overheads. The time is reported with training the model on a single NVIDIA 3090 GPU.

	VangoRoom			LivingRoom		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF	28.83	0.769	0.339	29.73	0.848	0.215
Deblur-NeRF [21]	29.30	0.793	**0.247**	31.82	0.895	0.132
Ours+NeRF	**30.26**	**0.805**	0.259	**32.70**	**0.912**	**0.124**
Point-NeRF	31.24	0.950	0.152	32.20	0.959	0.109
Ours+Point-NeRF	**33.27**	**0.966**	**0.097**	**35.30**	**0.980**	**0.051**

Table 2. Quantitative comparisons on the synthetic data. We apply our design used to handle blur artifacts to two different frameworks, *i.e.*, NeRF and Point-NeRF. With our design, the values of all three metrics receive significant improvements.

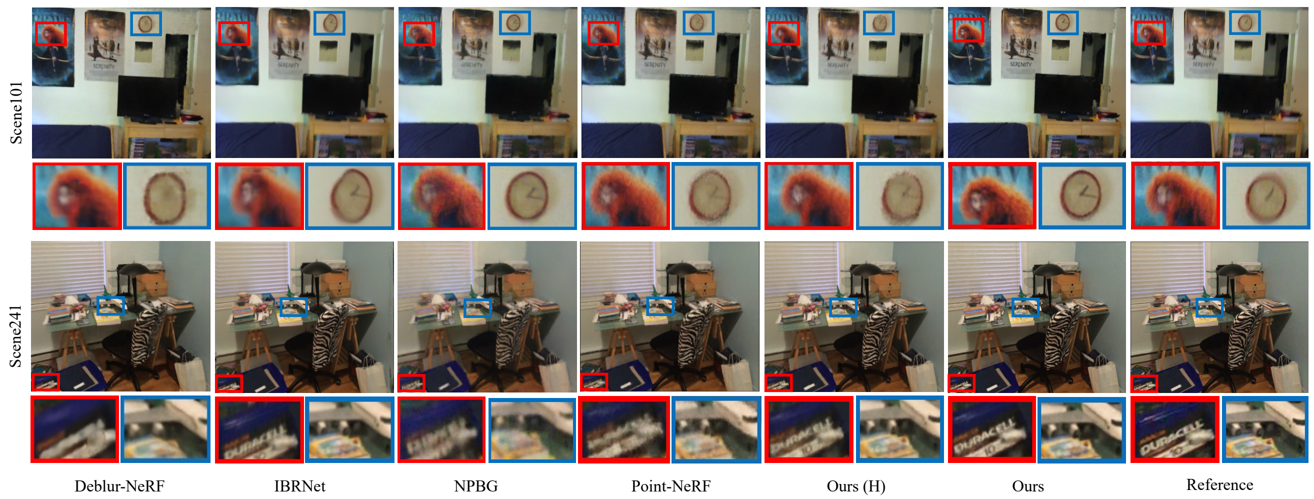

Figure 6. Qualitative comparisons on ScanNet. The highlighted regions are zoomed-in and placed at the bottom for better comparisons. From the results, our method can synthesize sharper images than other approaches that are suffering from blurriness, distortions, and jagged edges. Moreover, the sharpness is further improved after applying our design to handle blur artifacts (*i.e*., Ours vs. Ours (H)).

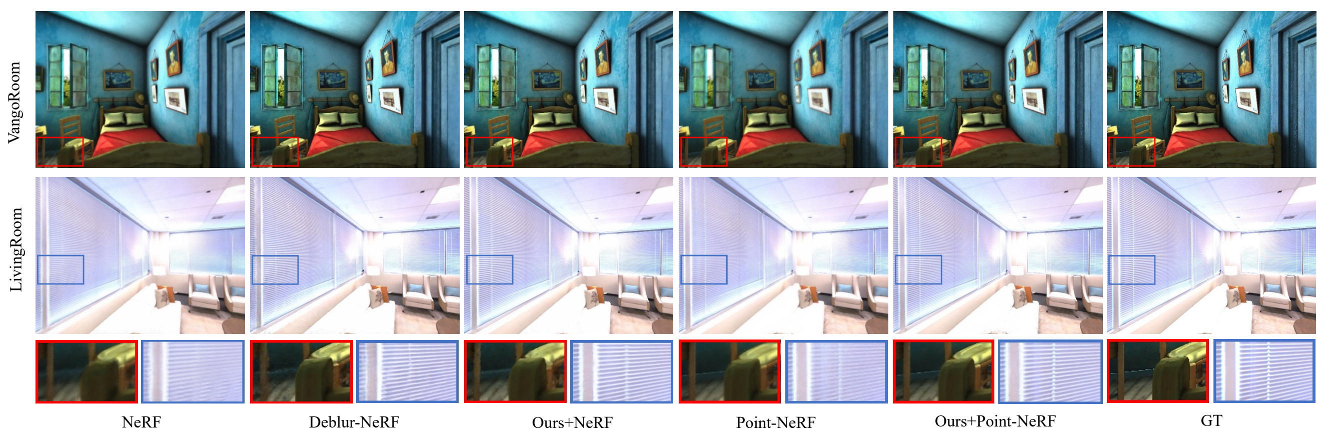

Figure 7. Qualitative comparisons on the synthetic data. We validate our designs to deal with blur artifacts on two different frameworks, *i.e*., NeRF and Point-NeRF. By applying our design, the details of rendered images become sharper. Besides, the Deblur-NeRF can also improve the sharpness, but the structure (*e.g*., blinds) is distorted.

4.4. Ablation Studies

In this section, we conduct comprehensive ablations of the proposed designs in our method.

Advantages of Image Features We first assess the contribution of image features in our system by comparing "Ours (H)" and Point-NeRF (*i.e*., without using image features). As shown in Fig. 6 and Table 1, our method benefits from image features and outperforms Point-NeRF. Moreover, our hybrid model converges faster. For example, we achieve PSNR 31.0 after 20k iterations (80 minutes) on "Scene241_01", whereas Point-NeRF delivers 29.3 PSNR after 40k iterations (84 minutes). (Please refer to supplementary material for more results.) This is because compressing all information into a neural 3D representation is difficult since it requires accurate camera poses, high-resolution 3D representations, etc. On the contrary, high-fidelity image features can directly compensate for defective neural 3D features and enable synthesizing high-quality results with fewer training iterations.

Advantages of Neural 3D Features We then show the value of the neural 3D feature by comparing it with IBRNet, which uses only image features. From the results in Fig. 6 and the video in the supplementary material, the rendered images from IBRNet are often distorted and inconsistent due to the lack of global 3D constraints. To further investigate the efficacy of learned neural 3D representations, we replace the neural 3D features with the mean and vari-

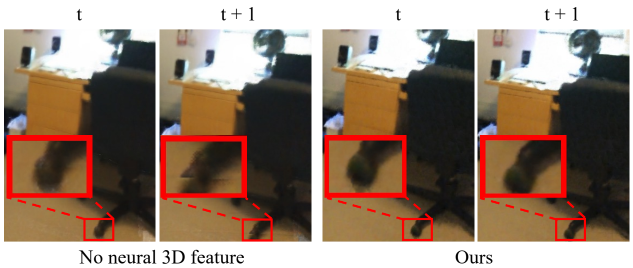

Figure 8. Advantages of neural 3D features. Without using the neural 3D feature, the rendered images will be inconsistent at different times. This can be observed in the example of the chair leg.

Figure 9. Efficacy of random drop. Without random drop, areas around image boundary (*e.g.*, the sofa in the left image) not covered by image features are bad.

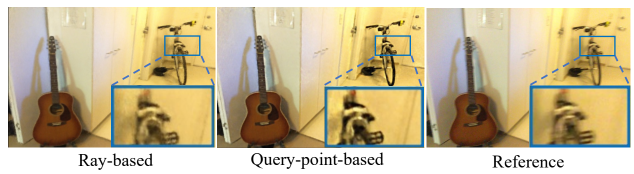

Figure 10. Different random drop methods. Query-point-based random drop automatically enhances the rendered images, but it tends to change the color tone. Please note the bicycle.

ance of image features extracted from nearby frames in the feature aggregation module (see Fig. 3). The corresponding results are displayed in Fig. 8, our approach preserves the shape consistency of the chair leg.

Blur Simulation and Quality-aware Design We further show the effect of blur simulation and detection by removing each of them at a time, and the results on "LivingRoom" are shown in Table 3. According to Table 3, both components contribute to the final performance, and the performance is improved when they are combined.

	NeRF-Based			Point-NeRF-Based		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Baseline	29.73	0.848	0.215	32.20	0.959	0.109
+ Blur simulation	32.24	0.905	0.133	34.32	0.974	0.065
+ Quality-aware weight	31.81	0.900	0.148	34.48	0.977	0.065
Full	32.70	0.912	0.124	35.30	0.980	0.051

Table 3. We validate our designs to handle blur artifacts on the 'LivingRoom'. Each component improves the performance and works better when combined.

Random Drop Methods The random drop strategy is to avoid the optimization being dominated by image features. As shown in Fig. 9, without using random drop in the training process, the results are poor in areas not covered by image features (the right side of the sofa). This region can only rely on neural 3D representation for rendering; thus, the poor results imply that the neural 3D representations are not well optimized. In contrast, our method with the random drop strategy produces high-quality images. Moreover, we observe that the results are slightly different when using different variants of the random drop strategy. For example, the rendered image is automatically enhanced using query-point-based random drop, as displayed in Fig. 10. One possible explanation is that, during training, the use of volume rendering for aggregation automatically enhances query points with image features to compensate for query points with low-quality neural 3D features on the same ray. However, this enhancement tends to change the color tone, as demonstrated in the bicycle example in Fig. 10. Thus, we currently adopt the ray-based random drop to render images having a closer appearance to the reference images.

5. Conclusion

In this paper, we present an approach to render high-fidelity and view-consistent images in large-scale scenes from sources contaminated by motion blurs. We develop a hybrid neural rendering model that makes use advantages of both image-based representation and neural 3D representation to render high-quality and view-consistent results. We also propose to efficiently simulate blur effects on the rendered image and design a quality-aware training strategy to down-weight blurry images, which helps the hybrid neural rendering model learn from blur-free supervisions and generate high-fidelity images. We conduct experiments on both real and synthetic data and obtain superior performance over previous methods.

Limitations Our method focuses on dealing with simple motion blurs in the training data, and defocus blur is not considered. Moreover, there are many other in-the-wild challenges, such as images captured under different exposure times and light conditions that require further research.

Acknowledgement This work has been supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant no. 17202422), and RGC matching fund scheme (RMGS).

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision*, pages 696–712. Springer, 2020.

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022.

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[6] Peng Dai, Xin Yu, Lan Ma, Baoheng Zhang, Jia Li, Wenbo Li, Jiajun Shen, and Xiaojuan Qi. Video demoireing with relation-based temporal consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17622–17631, 2022.

[7] Peng Dai, Yinda Zhang, Zhuwen Li, Shuaicheng Liu, and Bing Zeng. Neural point cloud rendering via multi-plane projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7830–7839, 2020.

[8] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996.

[9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[10] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022.

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

[12] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[13] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022.

[14] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. *arXiv preprint arXiv:2203.12575*, 2022.

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

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[18] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022.

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

[20] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019.

[21] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12870, 2022.

[22] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[24] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

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

[26] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[27] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martinez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 314–317. IEEE, 2000.

[28] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022.

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

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

[31] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022.

[32] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

[33] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2642–2650, 2021.

[34] Hyeongseok Son, Junyong Lee, Jonghyeop Lee, Sunghyun Cho, and Seungyong Lee. Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. *ACM Transactions on Graphics (TOG)*, 40(5):1–18, 2021.

[35] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[36] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[37] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[38] Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11956–11965, 2021.

[39] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[40] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021.

[41] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[42] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 157–170. Springer, 2010.

[43] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022.

[44] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *European Conference on Computer Vision*, pages 597–614. Springer, 2022.

[45] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.

[46] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.

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

[48] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Jiajun Shen, Jia Li, and Xiaojuan Qi. Towards efficient and scale-robust ultra-high-definition image demoiréing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 646–662. Springer, 2022.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 586–595, 2018.

[50] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021.

[51] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.