Generalizable Novel-View Synthesis using a Stereo Camera

Supplementary Material

A. Overview

In this supplementary document, we provide:
- additional implementation details (Sec. B),
- details on experimental settings (Sec. C),
- additional analyses (Sec. D),
- details on the StereoNVS dataset (Sec. E),
- additional quantitative results (Sec. F), and
- additional qualitative results (Sec. G).

B. Implementation Details for StereoNeRF

B.1. Stereo estimation network

We employ UniMatch [14] as the stereo estimation network in our framework. The stereo estimation network S outputs multiple depths during its refinement stage:

$$(d_{s,L}^{n,k}, d_{s,R}^{n,k}) = S(I_L^n, I_R^n), \quad \text{where } k \in \{1, \cdots, K\}, \tag{S1}$$

where K (set to 7) is the total number of outputs. Throughout the paper, we use $(d_{s,L}^{n}, d_{s,R}^{n})$ to indicate $(d_{s,L}^{n,K}, d_{s,R}^{n,K})$ for simplicity, where $(d_{s,L}^{n,K}, d_{s,R}^{n,K})$ are the final outputs. These multiple outputs are supervised using the stereo depth loss, which will be explained in Sec. B.4. Refer to [14] for more details about the stereo estimation network.

B.2. Stereo Feature Extractor

We provide additional details of the network architecture of the stereo feature extractor. To extract image features from input images, we modify the feature extractor from GeoNeRF [6] as follows: (1) We make the CNN encoder to project each of a stereo image pair (I_L^n, I_R^n) to feature maps. (2) We introduce the stereo attention module (SAM) [2] to fuse feature maps from the input stereo image pair. (3) We further extend SAM to aggregate stereo correlated features (t_L^n, t_R^n).

We utilize the Feature Pyramid Network (FPN) for both the CNN encoder and the CNN decoder within the stereo feature extractor, as done in GeoNeRF. The CNN decoder takes the fused feature maps from SAM as inputs and generates stereo image features with three different scales. We denote the multi-scale stereo image features as $(f_L^{n,l}, f_R^{n,l})$, where l is a scale index such that $l \in \{0, 1, 2\}$. $f_L^{n,2}$ has the same size as I_L^n, while $f_L^{n,0}$ and $f_L^{n,1}$ have resolutions of 1/4 and 1/2, respectively. These multi-scale stereo image features will be utilized to build multi-scale cost volumes in subsequent feature volume construction Sec. B.3.

B.3. Depth-guided Feature Volume Construction

We provide a detailed description for the depth-guided feature volume construction (Sec. 3.2). We adopt the cascaded cost volume strategy [4] of GeoNeRF [6] for constructing feature volumes. The feature volumes provide occlusion-aware geometric information for conditioning the neural renderer. To obtain fine and high-resolution feature volumes, cost volumes are built in a coarse-to-fine manner with three steps. Then, we introduce our depth-guided plane-sweeping (DGPS) in the initial step of the cascaded cost volume strategy. In the following, we explain the cascaded cost volume strategy first, followed by the incorporation of DGPS in this strategy.

The initial stage of the cascaded cost volume strategy begins by establishing the depth range using predefined near and far depths, denoted as d_{near} and d_{far}, respectively. Depth planes are then hypothesized by uniformly dividing the entire depth range by a predetermined number of depth planes, denoted as M. The interval between consecutive depth planes is determined as $\frac{d_{far}-d_{near}}{M}$ during this initial stage. On these depth planes, plane-sweeping aggregates the coarsest-scale multi-view features $(f_L^{n,0}, f_R^{n,0})$ to build the initial-stage cost volumes. Then, the MVS network produces the initial-stage feature volumes $(\phi_L^{n,0}, \phi_R^{n,0})$ and the initial-stage depth maps $(d_{m,L}^{n,0}, d_{m,R}^{n,0})$ from these initial-stage cost volumes.

Next, for the cost volume construction in the subsequent stages, we hypothesize depth planes around the depths obtained from the previous stage and perform plane-sweeping on these depth planes to aggregate finer-scale multi-view features $(f_L^{n,l}, f_R^{n,l})$. The plane interval of the finer-scale cost volume is reduced to $1/2$ compared to that of the previous stage. The feature volumes and depth maps for each stage are obtained via the MVS network, similar to the initial stage. We denote the resulting multi-scale feature volumes and depth maps as $(\phi_L^{n,l}, \phi_R^{n,l})$ and $(d_{m,L}^{n,l}, d_{m,R}^{n,l})$, respectively. We utilize individual UNet for the MVS network at each stage. The number of hypothesis depth planes M for each stage is set to 48, 32, and 8, respectively. The resolution of $d_{m,L}^{n,l}$ are the same as $f_L^{n,l}$.

However, using only the cascaded cost volume strategy might not ensure accurate geometry estimation, especially when the initial depth estimation is not precise. Inaccurate depth estimation may hinder the subsequent construction of feature volumes around the real geometry. Since the prediction of NeRF relies on these feature volumes, the neural renderer predicts inaccurate geometry when feature volumes are not constructed around the real geometry. To tackle this

issue, we introduce DGPS in the first stage of this strategy. Unlike the previous approach that hypothesizes depth planes across the entire depth range, DGPS hypothesizes depth planes around the stereo depth $(d^n_{s,L}, d^n_{s,R})$ obtained from the stereo estimation network. Leveraging the reliable stereo depth, DGPS ensures feature volume construction around the geometry, as explained in Sec. 5.3.

B.4. Stereo Depth Loss

We provide detailed explanation of Eq. 3 in the main paper: $\mathcal{L}^s_d$ and $\mathcal{L}^m_d$ are computed in a pixel-wise manner, and $\mathcal{L}^r_d$ is computed in a ray-wise manner. Specifically, $\mathcal{L}^s_d$ and $\mathcal{L}^m_d$ are defined as:

$$\mathcal{L}^s_d = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma^{K-k} \left\{ g(d^{n,k}_{s,L}, d^n_{gt,L}) + g(d^{n,k}_{s,R}, d^n_{gt,R}) \right\}, \quad \text{(S2)}$$

$$\mathcal{L}^m_d = \sum_{n=1}^{N}\sum_{l=0}^{2} 2^{-l} \left\{ g(d^{n,l}_{m,L}, d^{n,l}_{gt,L}) + g(d^{n,l}_{m,R}, d^{n,l}_{gt,R}) \right\}, \quad \text{(S3)}$$

where $g(d, d_{gt})$ represents the distance between the estimated depth maps d and the pseudo-ground-truth depth maps d_{gt}. K is the total number of depth predictions and γ (set to 0.9) is the weight to give higher weights for later depth prediction (Sec. B.1). $d^n_{gt,L}$ and $d^n_{gt,R}$ are pseudo-ground-truth depth maps of I^n_L and I^n_R, respectively. $d^{n,l}_{gt,L}$ and $d^{n,l}_{gt,R}$ are the downsampled versions of $d^n_{gt,L}$ and $d^n_{gt,R}$ according to the scale level l, respectively.

The distance function g is defined as:

$$g(d, d_{gt}) = \sum_{ij} m(i,j) \left\| \frac{1}{d(i,j)} - \frac{1}{d_{gt}(i,j)} \right\|, \quad \text{(S4)}$$

where (i, j) is a pixel coordinate, and $m(i, j)$ is a mask for the pixel (i, j). The distance function computes the difference between the inverses of depth values. The mask $m(i, j)$ is introduced to account for errors in the pseudo-ground-truth depth map. Specifically, we define $m(i, j) = 1$ if $d_{gt}(i, j)$ falls within the boundaries defined by the near and far depth range from COLMAP [10], and $m(i, j) = 0$ otherwise. For the norm $\| \cdot \|$, we use a smooth L_1 norm.

Among the loss terms of the stereo depth loss, $\mathcal{L}^r_d$ is defined as follow:

$$\mathcal{L}^r_d = \sum_{\mathbf{r}\in\mathcal{R}} m(\mathbf{r}) \left\| \frac{1}{d(\mathbf{r})} - \frac{1}{d_{gt}(\mathbf{r})} \right\|, \quad \text{(S5)}$$

where $\mathcal{R}$ denotes set of rays in each training batch, and $m(\mathbf{r})$ is a mask for ray $\mathbf{r}$. $m(\mathbf{r})$ is defined in the same way as $m(i, j)$.

B.5. Training Details for StereoNeRF

We set the balancing weights λ^{self}_d and λ^{stereo}_d in Eq. 2 as 0.1 and 1.0, respectively. For the stereo depth loss $\mathcal{L}^{stereo}_d$ in Eq. 3, we set λ_1, λ_2, and λ_3 as 0.1, 0.1, and 1.0, respectively. The image resolution for training and evaluation is 512×256 for both the StereoNVS-Real and the StereoNVS-Synthetic datasets. We use UniMatch [14] pre-trained on several mixed public datasets, which is used for all the mentioned pre-trained stereo estimation networks in our framework. Specifically, we employ the configuration of 'GMStereo-scale2-regrefine3-resumeflowthings-mixdata' from the official UniMatch website (https://github.com/autonomousvision/unimatch). We use CasMVSNet [4] pre-trained on BlendedMVS [15] for our MVS network, as done in GeoNeRF [6]. The training takes 5 days using one RTX3090 GPU.

B.6. Computing Depth from Disparity

We utilize the stereo estimation network to obtain pseudo-ground-truth depth d_{gt} and stereo depths d_s. Specifically, we employ UniMatch [14] for the stereo estimation network, which estimates disparities from input stereo-image pairs. If the focal length and the baseline length between the stereo image pair are known, depths can be computed from the estimated disparities as follows:

$$d = \frac{b \times f}{\text{disp}}, \quad \text{(S6)}$$

where d, b, f, and disp denote depth, baseline length, focal length, and disparity, respectively. We use the pre-computed values for the baseline length and the focal length (Sec. E).

C. Experimental Settings

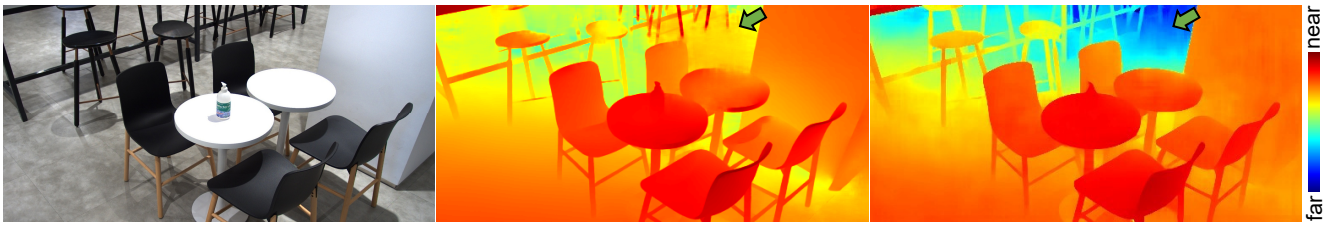
C.1. Training Details for Baselines

We train the baseline methods: SRF [1], IBRNet [12], GNT [11], GeoNeRF [6], and NeuRay [7] on the training set of the StereoNVS-Real dataset. We use 512 rays for the training batch except for GNT. All the models are trained for 250K iterations.

SRF. For SRF [1], we use official PyTorch implementation from https://github.com/jchibane/srf. For training, we use the Adam optimizer with learning rate of 0.0005 and the exponential decay strategy for learning rate scheduling.

IBRNet. For IBRNet [12], we use official PyTorch implementation from https://github.com/googleinters/IBRNet. For training, we use the Adam optimizer with learning rates of 0.001 and 0.0005 for feature extraction and rendering network, respectively. we use the exponential decay strategy for learning rate scheduling.

GNT. For GNT [11], we use official PyTorch implementation from https://github.com/VITA-Group/GNT. For



(a) Scene (b) from off-the-shelf stereo estimator (c) from partially-trained stereo estimator

Figure S1. Improved depth estimation accuracy thanks to our partial training scheme for the stereo estimation network. Without partial training scheme, the off-the-shelf stereo estimation network [14] predicts depths with an incorrect scale.

training, we use the we use the Adam optimizer with learning rates of 0.001 and 0.0005 for feature extraction and rendering network. we use the exponential decay strategy for learning rate scheduling. Since GNT employs large transformer backbone, we use 2048 rays for its training batch.

GeoNeRF. For GeoNeRF [6], we use official PyTorch implementation from https://github.com/idiap/GeoNeRF. For training, we use the Adam optimizer with learning rates of 0.0005 and the cosine-annealing scheduling. We use Cas-MVSNet [4] pre-trained on BlendedMVS [15] for the MVS network of GeoNeRF.

NeuRay. For NeuRay [7], we use official PyTorch implementation from https://github.com/liuyuan-pal/NeuRay. For training, we use the Adam optimizer with learning rates of 0.0002 and the exponential decay strategy for learning rate scheduling.

C.2. Experimental Details for Analysis

We provide additional details for experiments in Sec. 5.3.2 and Sec. 5.3.3 in the main paper.

C.2.1 Effectiveness of Depth-Guided Plane-Sweeping

Tab. 4 in the main paper shows the results using more depth planes for cascaded cost volume construction. As DGPS is applied in the first stage of the cascaded cost volume approach, we utilize 96 depth planes in the first stage for the additional model. Note that we use 48 depth planes in the first stage for all the other models.

C.2.2 Benefit of Stereo Estimation in Depth Loss

Tab. 5 in the main paper shows the effectiveness of utilizing stereo estimation networks for depth supervision. To obtain d_{gt}^{mvs}, we use UniMVSNet [9], one of the state-of-the-art MVS network, which is trained on the DTU dataset [5] and the BlendedMVS dataset [15].

	PSNR (↑)	SSIM (↑)	LPIPS (↓)	ABS (↓)
Model w/o partial training scheme	32.52	0.9252	0.1299	0.1287
Our final model	33.45	0.9336	0.1203	0.1056

Table S1. Effectiveness of our partial training scheme for robust depth estimation of the stereo estimation network.

D. Additional Analyses on StereoNeRF

D.1. Robustness to Stereo Depth Error

In our experimental setup, we utilize pseudo-ground truth stereo depth maps estimated from an off-the-shelf model [14] for NeuRay [7], GeoNeRF$_{+D}$ [6], and our proposed method. Since NeuRay and GeoNeRF$_{+D}$ rely on depth without considering potential errors, they are not robust to depth inaccuracies. As shown in Tab. 1 in the main paper, these methods report significantly degraded results in real-world scenes where obtaining error-free depth proves challenging. However, our approach circumvents this issue by incorporating partial training of the stereo estimation network within the stereo feature extractor, leveraging multi-view supervision (Sec. 3.4 in the main paper). This training scheme ensures robust depth estimation of the stereo estimation network, subsequently utilized in our DGPS, resulting in consistent performance regardless of the dataset. In the following discussion, we present additional experiments to further demonstrate the robustness of our method to depth errors.

Fig. S1 (b) shows such estimation errors where an off-the-shelf depth estimation model inaccurately predicts depths with incorrect scales. For example, the depth of distant regions is often inaccurately estimated to be closer to the camera. This erroneous depth estimation negatively affects the performance of GeoNeRF$_{+D}$ and NeuRay, particularly in real-world scenes (Tab. 1 in the main paper). However, our proposed solution, explained in Sec. 3.4 in the main paper, effectively mitigates this issue by incorporating partial training of the stereo estimation network.

To demonstrate the effectiveness of our partial training scheme of the stereo estimation network, we conduct an additional experiment. Specifically, we train an additional model, which is our final model with frozen parameters of

Crop Ratio	Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
1	GeoNeRF$^{(6)}$	**28.77**	**0.907**	**0.139**
	Ours	28.44	0.900	0.140
1/2	GeoNeRF$^{(6)}$	**28.87**	0.923	0.094
	Ours	28.84	**0.924**	**0.085**
1/3	GeoNeRF$^{(6)}$	29.09	0.926	0.093
	Ours	**29.23**	**0.931**	**0.080**

Table S2. Comparison stereo-camera setting against monocular-view setting with center-cropping strategy.

the stereo estimation network. We then compare the additional model with our final model based on both image and shape qualities. Tab. S1 presents the results, indicating a slight decline in performance for the additional model compared to our final model employing the partial training scheme.

This slight performance decline can be attributed to two main factors. Firstly, the use of inaccurate depth in DGPS results in the misplacement of feature volumes, deviating from the actual geometry. This misalignment, in turn, leads to inaccurate geometry estimation by the neural renderer, negatively impacting view synthesis performance. Secondly, as we use rendered depth as ground truth for L_d^{self}, the depth error introduced by the neural renderer can destabilize our framework's training process. In contrast, our proposed partial training scheme enhances the stereo estimation network's ability to estimate depth with precise scale (Fig. S1 (c)). It is because this training scheme makes our stereo estimation network more robust to differences in dataset characteristics. Therefore, DGPS with more accurate depth ensures the construction of feature volumes around real geometry, ultimately leading to improved geometry estimation by the neural renderer. This enhanced accuracy contributes to stable training, leading to better performance.

D.2. Comparison Between Monocular-Camera and Stereo-Camera Settings

We conduct further evaluation of the effectiveness of the stereo-camera setting by comparing our method with a baseline model [6], which is trained on six views (i.e., six images) in a monocular setting. Tab. 2 in the main paper already demonstrates that our proposed method, trained on three stereo-camera pairs (i.e., six images), outperforms the baseline method trained on three views (i.e., three images) in the monocular setting. To provide a more comprehensive comparison, we present an additional experiment comparing the monocular-camera and stereo-camera settings using the same number of images (i.e., six images).

In this experiment, we compare our method against the baseline method, GeoNeRF [6], specifically a variant denoted as GeoNeRF$^{(6)}$, which utilizes six monocular images in the inference. These six monocular images are obtained by selecting the left-side images from six stereo-camera image pairs. Note that leveraging six monocular views significantly broadens the observation of spatial views. To mitigate the impact of unobserved spatial regions in the evaluation, we compute errors by comparing center-cropped regions of both synthesized and ground-truth images.

As shown in Tab. S2, leveraging additional spatial views, GeoNeRF$^{(6)}$ achieves better results than our method. However, employing a 1/2 center-cropping strategy in the comparison, our method shows comparable or slightly better results than GeoNeRF$^{(6)}$. In addition, employing a 1/3 center-cropping strategy, our method surpasses GeoNeRF$^{(6)}$. These results underscore the effectiveness of our method, particularly in the observed region.

D.3. Choice of Stereo Estimation Network

In our framework, we adopt UniMatch [14] as the stereo estimation network due to its generalization capability. This capability is attributed to its training on a large-scale stereo-image dataset, which endows UniMatch with the ability to generalize well across various datasets. Consequently, our framework benefits from this high generalization capacity, leading to superior performance. To validate the relationship between the performance of our framework and the generalization capability of the stereo estimation network, we conduct an experiment comparing our final model with another model employing a different stereo estimation network.

This alternative model utilizes a stereo estimation network that has the identical network architecture to the stereo estimation network in our final model but is trained on the SceneFlow dataset [8]. Specifically, we utilize the "GMStereo-scale2-regrefine3-sceneflow" configuration from the official UniMatch website (https://github.com/autonomousvision/unimatch) for the stereo estimation network. Since the SceneFlow dataset is notably smaller in size compared to the large-scale dataset used to train our final model, this alternative model lacks the generalization ability demonstrated by our final model.

We compare the performances of our final model and this alternative model on the StereoNVS-Synthetic dataset. As shown in Tab. S3, our final model outperforms this alternative model across various metrics. This result underscores the correlation between the generalization ability of the stereo estimation network and the performance of our framework. Furthermore, we anticipate that incorporating a more refined stereo estimation network into our framework has the potential to further enhance performance.

E. StereoNVS dataset

Fig. S2 shows example scenes in the StereoNVS-Real and the StereoNVS-Synthetic datasets.

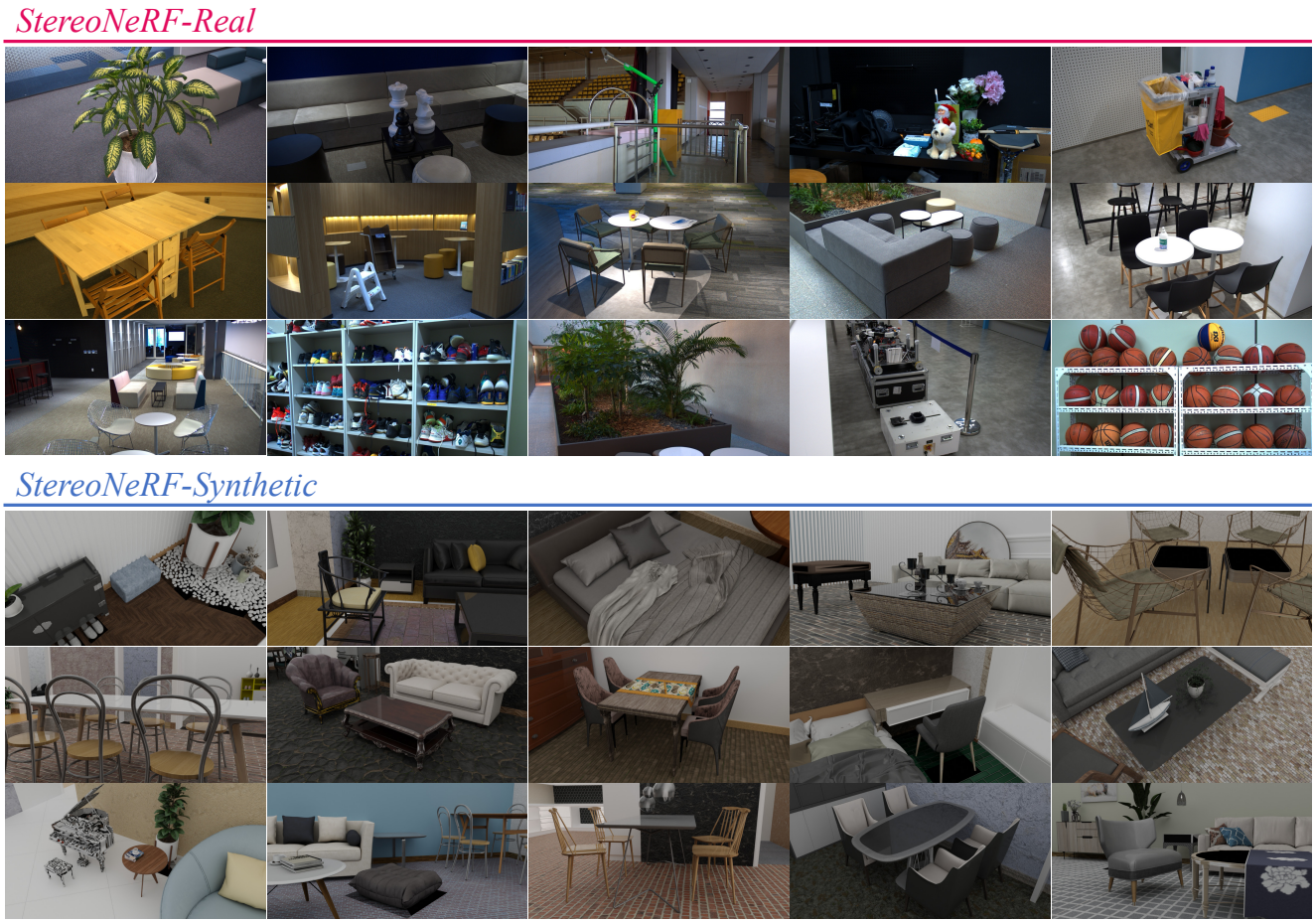


Figure S2. The StereoNVS datasets. The StereoNVS dataset presents multi-view stereo-pair images for both real-world and synthetic scenes. These images are part of the entire dataset.

	PSNR (↑)	SSIM (↑)	LPIPS (↓)	ABS (↓)
Model w/ small-dataset-trained stereo network	31.73	0.9026	0.1637	0.2207
Our final model	**33.45**	**0.9336**	**0.1203**	**0.1056**

Table S3. Experiment on the choice of the stereo estimation network in our framework.

For the StereoNVS-Real, the baseline length is around 8.8cm and the focal length is around 1568 pixels. These values vary slightly for each scene due to the scale ambiguity induced by the structure-from-motion approach of COLMAP for camera pose acquisition. When capturing the real-world scenes, we manually fixed the stereo camera's focal length, exposure time, gain, gamma correction, and white balance for each scene. In addition, since acquired images have a stereoscopic constraint with fixed camera positions and orientations, we employed the "RigBundleAdjuster" configuration in COLMAP [10] to ensure the stereoscopic constraint during the camera pose acquisition.

For the StereoNVS-Synthetic, the baseline length and the focal length are 8cm and 711 pixels, respectively. We randomly sampled 50 synthetic scenes from the 3D-Front dataset [3]. For each scene, we generated 200 viewpoints of stereo image pairs, which are constrained to look at a specific object like furniture. Then, we filtered camera viewpoints placed too close to the objects in the scene.

F. Additional Quantitative Results

F.1. Depth Quality Assessment

We assess the depth quality using additional depth metrics such as absolute error (ABS), absolute relative error (ARE), and root mean square error (RMSE). Tab. S4 presents depth quality evaluation for our method and all the baseline methods [6, 7, 11, 12] on the StereoNVS-Synthetic dataset. The result reveals similar trends to those using ABS in the Tab. 1 in the main paper. Our method shows comparable depth quality to GeoNeRF$_{+D}$, although GeoNeRF$_{+D}$ explicitly uses depth maps in the inference.

Method	ABS (↓)	ARE (↓)	RMSE (↓)
SRF	0.8125	0.4673	1.1347
IBRNet	0.2628	0.1408	0.5137
GeoNeRF	0.1577	0.0933	0.3631
GNT	0.4512	0.1945	0.6593
GeoNeRF$_{+D}$	0.0782	0.0576	0.2301
NeuRay	0.1571	0.0852	0.3636
Ours	0.1056	0.0725	0.2656

Table S4. Depth quality assessment using additional depth metrics. We compare our method against the baseline methods [1, 6, 7, 11, 12]. Our method shows comparable depth quality to GeoNeRF$_{+D}$, although GeoNeRF$_{+D}$ explicitly uses depth maps in the inference.

F.2. Evaluation on the BlendedMVS Dataset

We evaluate our model on the BlendedMVS dataset [15]. Since BlendedMVS do not provide stereo-camera images, we were not able to evaluate our model directly on BlendedMVS, which is released on the official website (https://github.com/YoYo000/BlendedMVS). Nonetheless, while BlendedMVS does not directly provide stereo-camera images, it still provides 3D meshes. Thus, we conducted an additional evaluation by rendering stereoscopic images from the meshes. For this evaluation, we classify large scenes based on the information available on the website (https://github.com/kwea123/BlendedMVS_scenes). Then, we sample several scenes from these categorized scenes. The list of the names for the sampled scenes is as follows:

- '5aa515e613d42d091d29d300',
- '5bf18642c50e6f7f8bdbd492',
- '5af02e904c8216544b4ab5a2',
- '5b69cc0cb44b61786eb959bf',
- '5bfc9d5aec61ca1dd69132a2',
- '5b08286b2775267d5b0634ba',
- '5ba75d79d76ffa2c86cf2f05',
- '58eaf1513353456af3a1682a', and
- '5af28cea59bc705737003253'.

Tab. S5 shows that our method outperforms other baseline methods [1, 6, 7, 11, 12] on the BlendedMVS dataset. Given the complex structures in scenes from BlendedMVS, these results highlight the effectiveness of our method, particularly in such settings. In addition, SRF [1] and GNT [11] demonstrate significantly degraded performances on BlendedMVS due to their limited generalization capabilities.

F.3. Evaluation Using Per-scene Metrics

Tab. S6 presents per-scene evaluaion using mean and variance of PSNR, SSIM [13], and LPIPS [16] on the StereoNVS-Real test set. Our method mostly shows better performance than the baseline methods [6, 7, 11, 12].

Method	PSNR(↑)	SSIM(↑)	LPIPS(↓)	ABS (↓)	ARE (↓)	RMSE (↓)
SRF	13.93	0.2251	0.6866	39.74	0.5952	50.96
IBRNet	21.68	0.6758	0.3191	7.726	0.1539	15.86
GeoNeRF	23.31	0.7635	0.2426	4.552	0.0769	11.45
GNT	11.58	0.2459	0.7675	23.31	0.4636	32.04
GeoNeRF$_{+D}$	23.67	0.7390	0.2618	3.434	**0.0473**	**8.011**
NeuRay	23.29	0.7271	0.2700	3.358	0.0634	8.861
Ours	**24.08**	**0.7842**	**0.2301**	**3.275**	0.0572	8.452

Table S5. Evaluation on the BlendedMVS dataset [15]. We quantitatively compare our method against the baseline methods [1, 6, 7, 11, 12].

G. Additional Qualitative Results

We present an additional qualitative comparison of StereoNeRF with other baseline methods: IBRNet [12], GNT [11], GeoNeRF [6], and NeuRay [7].

Fig. S3 presents novel-view rendering results with whole images and depths for the StereoNVS-Real dataset. Our method shows the best image quality, along with the accurately estimated depth maps. While GeoNeRF$_{+D}$ seems to produce depth results comparable to our method, their synthesized images show artifacts, especially for the object boundary regions. It is because they utilize the pseudo-ground-truth depths in its inference without considering the error. Additionally, we present the rendering results of diverse scenes in the both StereoNVS-Real and StereoNVS-Synthetic datasets. Fig. S4 shows zoomed-in patches, where other baseline methods synthesize degenerated images, especially for thin structures and textureless regions.

(mean,var)	Scene #3	Scene #11	Scene #13	Scene #25	Scene #26	Scene #30	Scene #38	Scene #40
IBRNet	(26.58, 1.5)	(27.19, 1.4)	(21.32, 5.6)	(28.52, 3.5)	(28.41,2.8)	(23.32, 4.7)	(27.50, 1.6)	(26.41, 1.7)
GeoNeRF	(28.40, 2.7)	(28.81, 1.8)	(**22.88**, 4.5)	(30.46, 2.6)	(30.23, 3.2)	(24.93, 4.6)	(30.05, 2.2)	(28.60, 2.6)
GNT	(26.00, 1.2)	(27.17, 1.5)	(21.38, 3.5)	(28.86, 2.9)	(28.61, 1.7)	(23.30, 4.5)	(27.51, 1.1)	(25.96, 2.6)
GeoNeRF$_{+D}$	(27.12, 2.4)	(26.50, 1.0)	(21.54, 3.2)	(28.26, 3.4)	(29.00, 1.9)	(23.32, 2.8)	(27.64, 1.1)	(27.59, 0.8)
NeuRay	(27.73, 2.2)	(27.61, 1.8)	(21.54, 3.2)	(28.26, 3.4)	(29.00, 1.9)	(23.32, 2.8)	(27.64, 1.1)	(27.59, 0.8)
Ours	(**28.87**, 2.6)	(**29.35**, 2.2)	(22.85, 4.5)	(**30.82**, 2.4)	(**30.76**, 3.8)	(**25.37**, 5.0)	(**30.50**, 2.2)	(**29.33**, 2.4)

(a) PSNR

(mean,var)	Scene #3	Scene #11	Scene #13	Scene #25	Scene #26	Scene #30	Scene #38	Scene #40
IBRNet	(0.8133, 0.0007)	(0.8953, 0.0004)	(0.8386, 0.0038)	(0.8729, 0.0010)	(0.8239, 0.0009)	(0.8491, 0.0018)	(0.7850, 0.0007)	(0.8675, 0.0005)
GeoNeRF	(0.8669, 0.0005)	(0.9210, 0.0002)	(**0.8704**, 0.0019)	(0.9173, 0.0003)	(0.8801, 0.0005)	(0.8897, 0.0013)	(0.8967, 0.0011)	(0.9034, 0.0005)
GNT	(0.8077, 0.0005)	(0.8977, 0.0002)	(0.8516, 0.0018)	(0.8802, 0.0006)	(0.8363, 0.0007)	(0.8518, 0.0016)	(0.7999, 0.0009)	(0.8205, 0.0009)
GeoNeRF$_{+D}$	(0.8283, 0.0006)	(0.8728, 0.0003)	(0.7371, 0.0023)	(0.7944, 0.0015)	(0.8042, 0.0013)	(0.7946, 0.0019)	(0.8533, 0.0008)	(0.8761, 0.0004)
NeuRay	(0.8515, 0.0006)	(0.8993, 0.0003)	(0.8369, 0.0019)	(0.8736, 0.0011)	(0.8534, 0.0004)	(0.8570, 0.0011)	(0.7860, 0.0007)	(0.8864, 0.0002)
Ours	(**0.8726**, 0.0005)	(**0.9254**, 0.0002)	(0.8699, 0.0021)	(**0.9240**, 0.0002)	(**0.8917**, 0.0006)	(**0.8993**, 0.0010)	(**0.9056**, 0.0009)	(**0.9135**, 0.0004)

(b) SSIM

(mean,var)	Scene #3	Scene #11	Scene #13	Scene #25	Scene #26	Scene #30	Scene #38	Scene #40
IBRNet	(0.2812, 0.0007)	(0.1811, 0.0008)	(0.1744, 0.0020)	(0.1729, 0.0009)	(0.2400, 0.0013)	(0.1889, 0.0011)	(0.2144, 0.0009)	(0.2349, 0.0009)
GeoNeRF	(**0.1931**, 0.0010)	(0.1221, 0.0001)	(0.1343, 0.0009)	(0.1139, 0.0003)	(0.1658, 0.0009)	(0.1513, 0.0011)	(0.1160, 0.0003)	(0.1832, 0.0006)
GNT	(0.3189, 0.0007)	(0.2026, 0.0008)	(0.1741, 0.0012)	(0.1881, 0.0008)	(0.2672, 0.0011)	(0.2043, 0.0014)	(0.2262, 0.0012)	(0.2643, 0.0014)
GeoNeRF$_{+D}$	(0.2305, 0.0014)	(0.1626, 0.0002)	(0.2691, 0.0013)	(0.1988, 0.0009)	(0.2100, 0.0006)	(0.2324, 0.0017)	(0.1502, 0.0004)	(0.2079, 0.0003)
NeuRay	(0.2389, 0.0011)	(0.1581, 0.0006)	(0.1770, 0.0012)	(0.1675, 0.0010)	(0.1914, 0.0006)	(0.1771, 0.0009)	(0.2154, 0.0008)	(0.1868, 0.0006)
Ours	(0.1963, 0.0012)	(**0.1173**, 0.0002)	(**0.1329**, 0.0012)	(**0.1081**, 0.0003)	(**0.1482**, 0.0007)	(**0.1428**, 0.0007)	(**0.1094**, 0.0004)	(**0.1749**, 0.0005)

(c) LPIPS

Table S6. Evaluation using per-scene mean and variance of PSNR, SSIM, and LPIPS on the StereoNVS-Real test set.

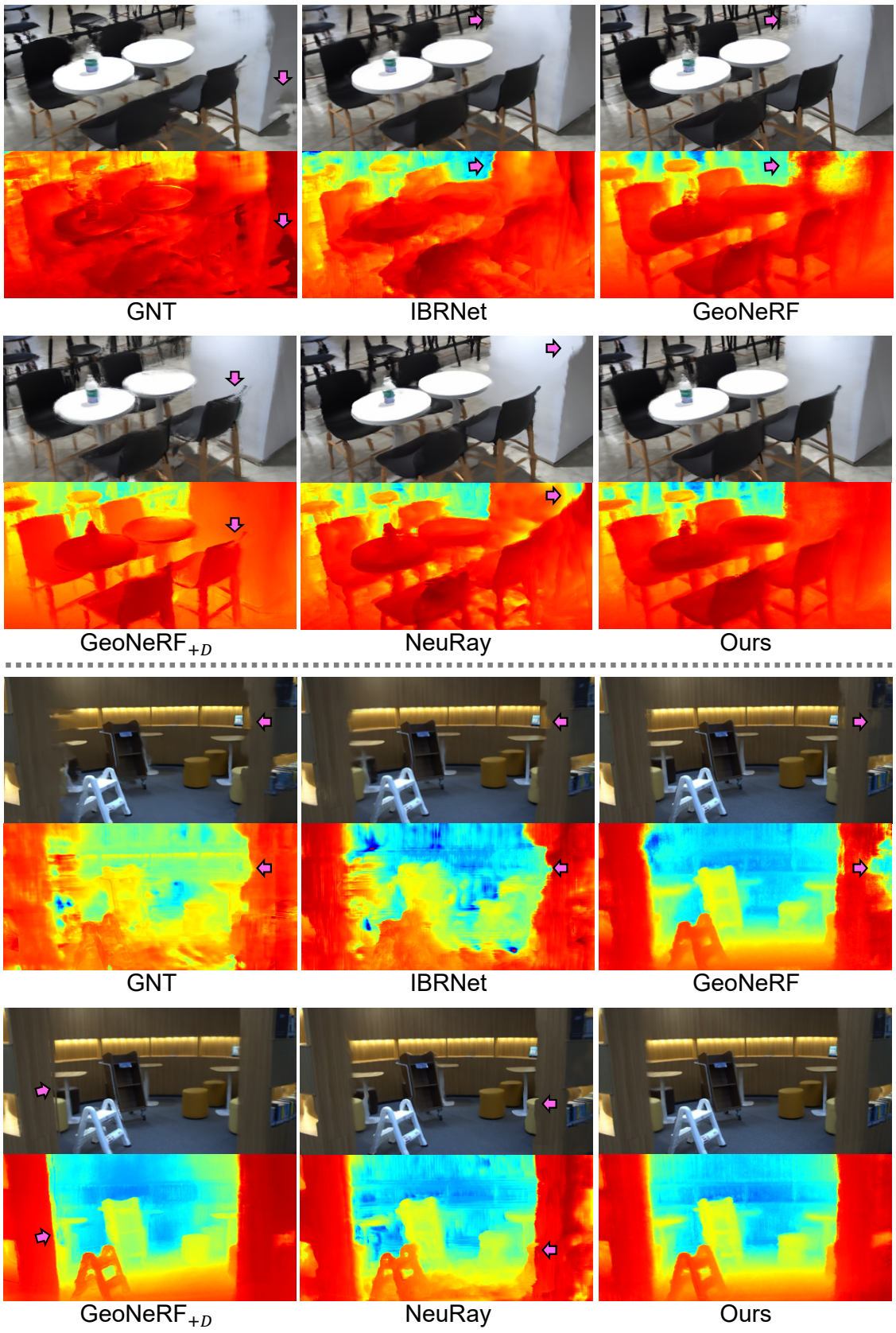


Figure S3. Qualitative comparison of novel-view synthesis on the StereoNVS-Real dataset. All models are trained using stereo images. Our method outperforms the baseline methods [6, 7, 11, 12] for both image and depth qualities.

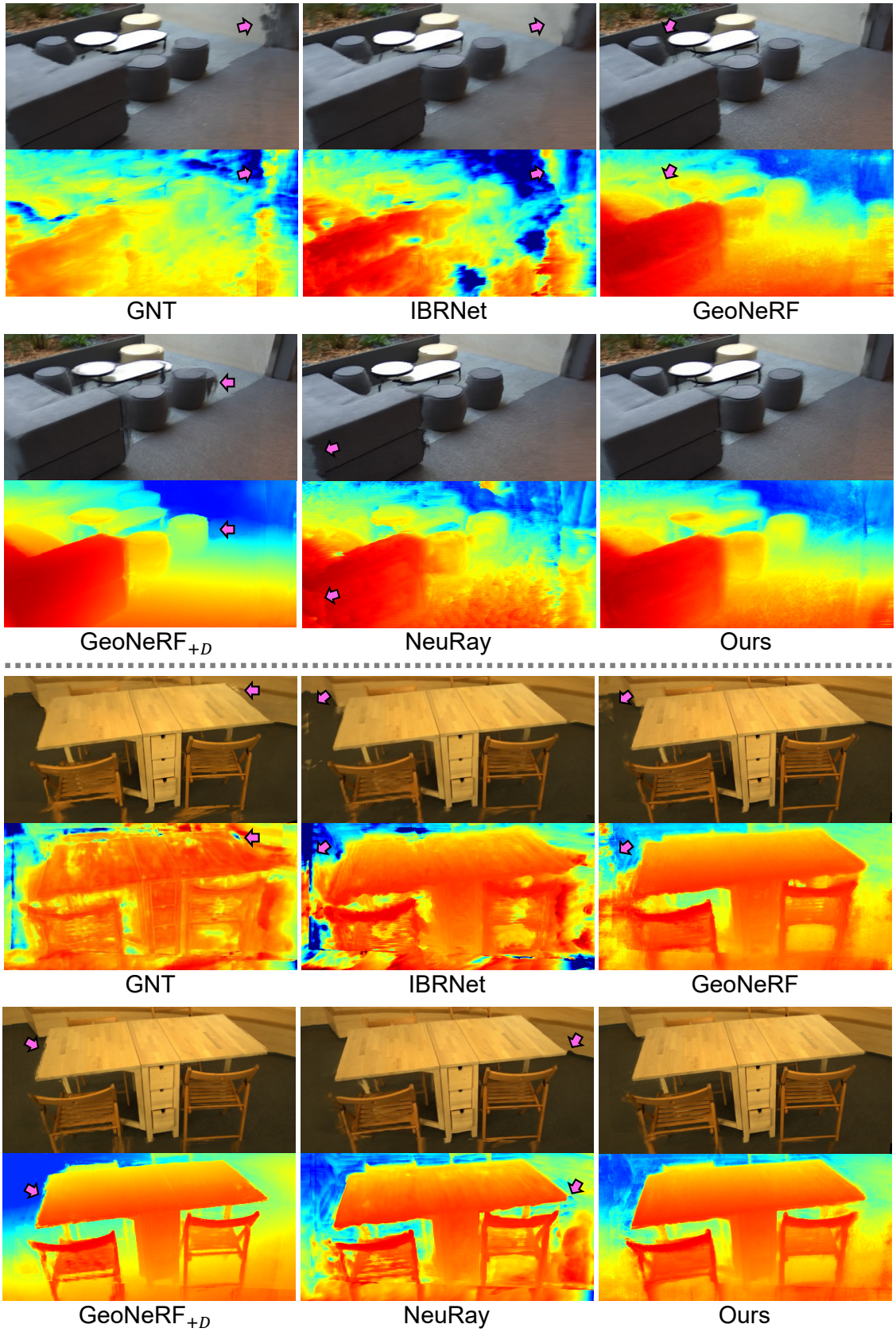


Figure S3. Qualitative comparison of novel-view synthesis on the StereoNVS-Real dataset. All models are trained using stereo images. Our method outperforms the baseline methods [6, 7, 11, 12] for both image and depth qualities.

Figure S4. Qualitative comparison of novel-view synthesis on the StereoNVS-Real and StereoNVS-Synthetic datasets. All models are trained using stereo images. Our method surpasses the baseline methods [6, 7, 11, 12], especially for thin structures and textureless regions.

References

[1] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *CVPR*, pages 7911–7920, 2021. 2, 6

[2] Xiaojie Chu, Liangyu Chen, and Wenqing Yu. Nafssr: Stereo image super-resolution using nafnet. In *CVPR*, pages 1239–1248, 2022. 1

[3] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 5

[4] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, pages 2495–2504, 2020. 1, 2, 3

[5] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, pages 406–413, 2014. 3

[6] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *CVPR*, pages 18365–18375, 2022. 1, 2, 3, 4, 5, 6, 8, 9, 10

[7] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, pages 7824–7833, 2022. 2, 3, 5, 6, 8, 9, 10

[8] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 4

[9] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *CVPR*, pages 8645–8654, 2022. 3

[10] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 2, 5

[11] Mukund Varma, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that nerf needs? In *ICLR*, 2022. 2, 5, 6, 8, 9, 10

[12] 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 *CVPR*, pages 4690–4699, 2021. 2, 5, 6, 8, 9, 10

[13] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[14] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE TPAMI*, 2023. 1, 2, 3, 4

[15] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, pages 1790–1799, 2020. 2, 3, 6

[16] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6