Dense Depth Priors for Neural Radiance Fields from Sparse Input Views

Barbara Roessle[1]   Jonathan T. Barron[2]   Ben Mildenhall[2]   Pratul P. Srinivasan[2]   Matthias Nießner[1]

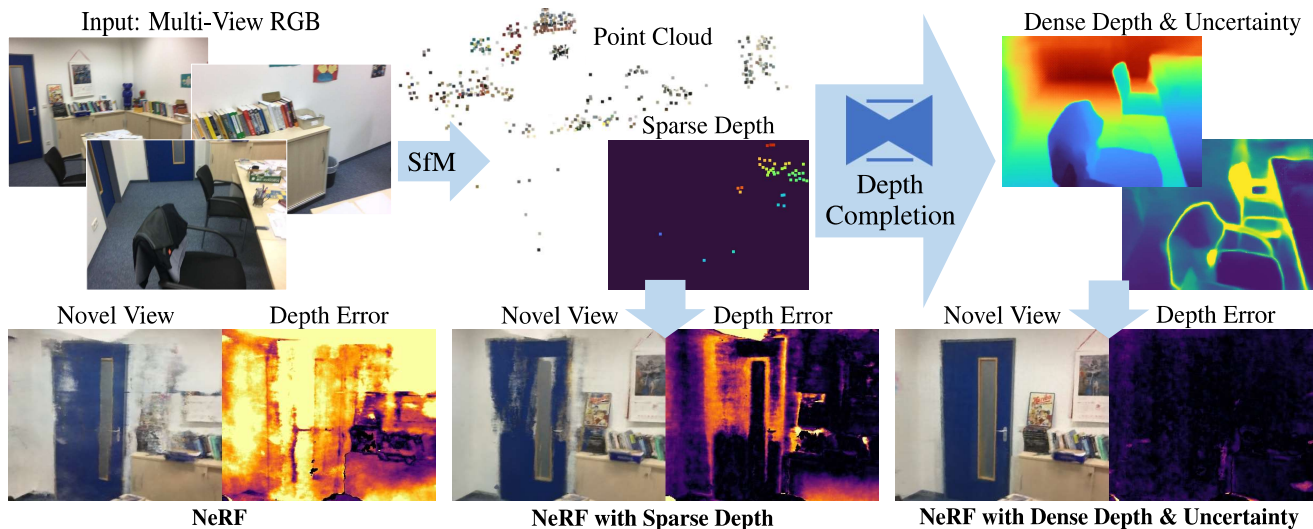[1]Technical University of Munich   [2]Google Research

Figure 1. We leverage dense depth priors for recovering neural radiance fields of complete rooms when only a handful of input images are available. To this end, we first utilize the sparse point cloud reconstructions from SfM preprocessing, which we feed into a depth completion network. We then impose these depth estimates as constraints to the NeRF optimization according to the estimated uncertainty. This facilitates novel view synthesis results at significantly higher image quality and lower depth error compared to NeRF.

Abstract

Neural radiance fields (NeRF) encode a scene into a neural representation that enables photo-realistic rendering of novel views. However, a successful reconstruction from RGB images requires a large number of input views taken under static conditions — typically up to a few hundred images for room-size scenes. Our method aims to synthesize novel views of whole rooms from an order of magnitude fewer images. To this end, we leverage dense depth priors in order to constrain the NeRF optimization. First, we take advantage of the sparse depth data that is freely available from the structure from motion (SfM) preprocessing step used to estimate camera poses. Second, we use depth completion to convert these sparse points into dense depth maps and uncertainty estimates, which are used to guide NeRF optimization. Our method enables data-efficient novel view synthesis on challenging indoor scenes, using as few as 18 images for an entire scene.

1. Introduction

Synthesizing realistic views from varying viewpoints is essential for interactions between humans and virtual environments, hence it is of key importance for many virtual reality applications. The novel view synthesis task is especially relevant for indoor scenes, where it enables virtual navigation through buildings, tourist destinations, or game environments. When scaling up such applications, it is preferable to minimize the amount of input data required to store and process, as well as its acquisition time and cost. In addition, a static scene requirement is increasingly difficult to fulfill for a longer capture duration. Thus, our goal is novel view synthesis at room-scale using few input views.

NeRF [21] represents the radiance field and density distribution of a scene as a multi-layer perceptron (MLP) and uses volume rendering to synthesize output images. This approach creates impressive, photo-realistic novel views of scenes with complex geometry and appearance. Unfortunately, applying NeRF to real-world, room-scale scenes given only tens of images does not produce desirable re-

sults (Fig. 1) for the following reason: NeRF purely relies on RGB values to determine correspondences between input images. As a result, high visual quality is only achieved by NeRF when it is given enough images to overcome the inherent ambiguity of the correspondence problem. Real-world indoor scenes have characteristics that further complicate the ambiguity challenge: First, in contrast to an "outside-in" viewing scenario of images taken around a central object, views of rooms represent an "inside-out" viewing scenario, in which the same number of images will exhibit significantly less overlap with each other. Second, indoor scenes often have large areas with minimal texture, such as white walls. Third, real-world data often has inconsistent color values across views, e.g., due to white balance or lens shading artifacts. These characteristics of indoor scenes are likewise challenging for SfM, leading to very sparse SfM reconstructions, often with severe outliers. Our idea is to use this noisy and incomplete depth data and from it produce a complete dense map alongside a per-pixel uncertainty estimate of those depths, thereby increasing its value for NeRF — especially in textureless, rarely observed, or color-inconsistent areas.

We propose a method that guides the NeRF optimization with dense depth priors, without the need for additional depth input (e.g., from an RGB-D sensor) of the scene. Instead, we take advantage of the sparse reconstruction that is freely available as a byproduct of running SfM to compute camera pose parameters. Specifically, we complete the sparse depth maps with a network that estimates uncertainty along with depth. Taking uncertainty into account, we use the resulting dense depth to constrain the optimization and to guide the scene sampling. We evaluate the effectiveness of our method on complete rooms from the Matterport3D [2] and ScanNet [6] datasets, using only a handful of input images. We show that our approach improves over recent and concurrent work that uses sparse depth from SfM or multi-view stereo (MVS) in NeRF [10, 26].

In summary, we demonstrate that dense depth priors with uncertainty estimates enable novel view synthesis with NeRF on room-size scenes using only 18–36 images, enabled by the following contributions:

1. A data-efficient approach to novel view synthesis on real-world scenes at room-scale.
2. An approach to enhance noisy sparse depth input from SfM to support the NeRF optimization.
3. A technique for accounting for variable uncertainty when guiding NeRF with depth information.

2. Related Work

The ability to synthesize novel views of a scene from a set of observed images and corresponding camera viewpoints is necessary for enabling virtual experiences of real-world environments. In situations where it is feasible to

densely sample images of the scene, novel viewpoints can be synthesized with simple light field interpolation [12, 17]. However, when fewer observed views of the scene are available, it becomes increasingly necessary to use information about the scene's geometry to render new views. A common paradigm for geometry-based view synthesis is to use a triangle mesh representation of scene geometry to reproject observed images into each novel viewpoint and combine them using either heuristic [1, 9, 27] or learned [14, 22] blending algorithms. More recently, these mesh-based geometry models have been replaced by volumetric scene representations such as voxel grids [18] or multiplane images [11, 20, 24, 29]. NeRF [21] popularized an approach that avoids the steep scaling properties of discrete voxel representations by representing a scene as a continuous volume, parameterized by a MLP that is optimized to minimize the loss of re-rendering all observed views of a scene. Since its introduction, NeRF has become the dominant scene representation for view synthesis, and many recent works are built on top of NeRF's neural volumetric model.

However, in situations where the scene is observed from very few sparsely-sampled viewpoints, NeRF's high capacity to model detailed geometry and appearance can result in various artifacts, such as "floaters", i.e., artifacts caused by a flawed density distribution. In this work, we directly address the few-input setting, proposing a strategy that takes advantage of sparse depth to constrain NeRF's scene geometry and improve rendering quality. This depth data is freely available as a byproduct of running SfM to compute camera poses from the input images (e.g., by using COLMAP [23]). Our method takes inspiration from techniques that generate complete dense depth maps from sparse depth inputs. These include classic techniques that fuse observed depths into a single 3D reconstruction, typically in the form of a truncated signed distance function [5, 15], as well as more recent techniques that train deep networks to operate over the sparsely observed geometry in 3D [7, 8]. Although these methods are effective for dense 3D scene reconstruction, the resulting geometry is not ideal for view synthesis since its edges frequently do not align with edges in the observed images. Instead, we leverage recent work on 2D sparse depth completion that directly completes depth maps in image space [3, 4] and extend it to also predict uncertainty.

A few recent works have also proposed incorporating depth observations into NeRF reconstruction. Nerfing-MVS [26] uses depth from MVS to overfit a depth predictor to the scene. The resulting depth prior guides the NeRF sampling. In comparison, our method employs depth completion on the SfM depth and additionally employs a depth loss to supervise the geometry recovered by NeRF. This way, our novel views achieve significantly better color and depth quality in the few-input setting without relying on the computationally more expensive MVS preprocessing. Con-

current work on depth-supervised NeRF [10] directly uses sparse depth information from SfM in the NeRF optimization. To handle inaccuracy in the sparse reconstruction, the 3D points are weighted according to their reprojection error. In contrast, we learn dense depth priors with uncertainty to more effectively guide the optimization, leading to more detailed novel views, as well as more accurate geometry and higher robustness to SfM outliers.

3. Method

Our method facilitates room-scale novel view synthesis from a small collection of RGB images $\{I_i\}_{i=0}^{N-1}$, $I_i \in [0, 1]^{H \times W \times 3}$ (see Fig. 2). As a preprocessing step (e.g., using SfM), the camera pose $\mathbf{p}_i \in \mathbb{R}^6$, intrinsic parameters $K_i \in \mathbb{R}^{3 \times 3}$, and a sparse depth map $Z_i^{\mathrm{sparse}} \in [0, t_f]^{H \times W}$ are computed for each image. 0 values of the sparse depth indicate invalid pixels, and t_f is the far plane of the volume rendering. Our approach builds upon NeRF [21]. Prior to the NeRF optimization, a network estimates depth with uncertainty from the sparse depth input (Sec. 3.1). We incorporate the resulting dense depth prior into the NeRF optimization by adding a depth constraint and a guided sampling approach (Sec. 3.2).

3.1. Depth Completion with Uncertainty

Network Architecture With the goal of completing sparse depth from SfM, two challenges presented by this input data play a key role in designing the depth prior network. First, sparse reconstructions are noisy and have outliers. As a consequence, dense depth predictions are expected to have spatially varying accuracy, which makes it crucial to know the uncertainty at a per-pixel level. Second, the density of SfM point clouds varies significantly across space, depending on the number of image features. E.g., SfM reconstructions from 18–20 images per ScanNet scene lead to sparse depth maps with 0.04% valid pixels on average. Hence, depth completion must be able to predict dense depth even from largely empty sparse depth maps.

In order to address the first challenge, we construct our depth prior network D_{θ_0} to predict dense depth maps $Z_i^{\mathrm{dense}} \in [0, t_f]^{H \times W}$ along with pixelwise standard deviations $S_i \in [0, \infty)^{H \times W}$ from the sparse depth maps:

$$\left[Z_i^{\mathrm{dense}}, S_i \right] = D_{\theta_0}(I_i, Z_i^{\mathrm{sparse}}) , \tag{1}$$

where D_{θ_0} is a convolutional network with ResNet [13] downsampling and skip connections to two upsampling branches to predict depth Z_i^{dense} and standard deviation S_i. To address the second challenge of extremely sparse input depth, we employ a Convolutional Spatial Propagation Network (CSPN) [4] in each branch. This refinement block locally and iteratively applies a kernel with weights given by a learned affinity matrix. This refines the often blurry

depth output to become more detailed and sharp. Equally important, this process spreads information to neighboring pixels; i.e., information propagates further with each iteration. An increased number of iterations in the depth and the uncertainty head prove helpful to handle very sparse input.

Network Training Though we evaluate on RGB-only data using SfM, we train our model on RGB-D data from ScanNet [6] and Matterport3D [2]. To avoid both the effort of running SfM on a large dataset and the possibility of SfM failures in the training data, the sparse depth input is sampled from the range sensor depth. As such, it is critical to subsample and perturb the dense depth from the sensor in a way that creates realistic sparse training depth to enable the network to generalize to real SfM input at test time. Specifically, a SIFT feature extractor, e.g., from COLMAP [23], is used to determine locations where sparse depth points would exist in a SfM reconstruction. We sample the sensor depth at these points and perturb it with Gaussian noise $\mathcal{N}(0, s_{\mathrm{noise}}(z)^2)$, where the standard deviation s_{noise} increases with depth. The function $s_{\mathrm{noise}}(z)$ is determined by fitting a second-order polynomial to the depth deviation between the sparse SfM reconstructions and sensor depth.

Under the assumption that the output is normally distributed, we supervise the network by minimizing the negative log likelihood of a Gaussian:

$$\mathcal{L}_{\theta_0} = \frac{1}{n} \sum_{j=1}^{n} \left(\log(s_j^2) + \frac{(z_j - z_{\mathrm{sensor},j})^2}{s_j^2} \right) , \tag{2}$$

where z_j and s_j are the predicted depth and standard deviation of pixel j, $z_{\mathrm{sensor},j}$ is the sensor depth at j, and n is the number of valid pixels in the dense sensor depth map.

3.2. Radiance Field with Dense Depth Priors

Scene Representation Following NeRF [21], we encode the radiance field of the scene into a MLP F_{θ_1} that predicts color $\mathbf{c} = [r, g, b]$ and volume density σ from a position $\mathbf{x} \in \mathbb{R}^3$ and a unit-norm viewing direction $\mathbf{d} \in \mathbb{S}^2$. γ applies a positional encoding with 9 frequencies on the position. Because our scenes are angularly undersampled, we minimize the capacity of our view-dependent network by omitting positional encoding for the viewing direction.

$$[\mathbf{c}, \sigma] = F_{\theta_1}(\gamma(\mathbf{x}), \mathbf{d}, \boldsymbol{\ell}_i) . \tag{3}$$

As an additional input to F_{θ_1}, we generate a per-image embedding vector $\boldsymbol{\ell}_i \in \mathbb{R}^e$ [19]. This allows the network to compensate for view-specific phenomena such as inconsistent lighting or lens shading, which can cause severe artifacts in novel views, particularly with few input images.

Optimization with Depth Constraint To optimize the radiance field, the color $\hat{\mathbf{C}}(\mathbf{r})$ of each pixel in the batch R
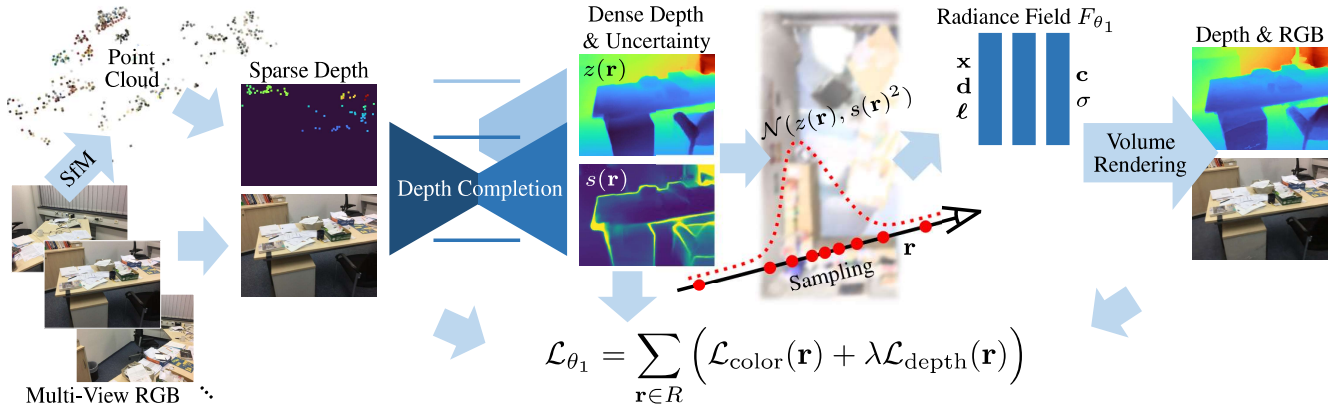
Figure 2. Overview of our radiance field optimization pipeline. Given a small set of RGB images of a room, we run SfM to obtain camera parameters and a sparse reconstruction, from which a sparse depth map is rendered for each input view. A depth completion network predicts dense depth and standard deviation, which is used to focus the scene sampling on surfaces. The samples on a ray, its viewing direction and a per-camera latent code are input to the radiance field. The output color and density are integrated to obtain the pixel's color and the ray's expected termination depth. The radiance field is supervised using the input RGB and the depth completion output.

is computed by evaluating a discretized version of the volume rendering integral (Eq. (4) [21]). Specifically, a pixel determines a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ whose origin is at the camera's center of projection $\mathbf{o}$. Rays are sampled along their traversal through the volume. For each sampling location $t_k \in [t_n, t_f]$ within the near and far planes, a query to F_{θ_1} provides the local color and density.

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{K} w_k \mathbf{c}_k , \tag{4}$$

$$\text{where} \quad w_k = T_k \left(1 - \exp(-\sigma_k \delta_k)\right) , \tag{5}$$

$$T_k = \exp\left(-\sum_{k'=1}^{k} \sigma_{k'} \delta_{k'}\right) , \tag{6}$$

$$\delta_k = t_{k+1} - t_k . \tag{7}$$

Besides the predicted color of a ray, a NeRF depth estimate $\hat{z}(\mathbf{r})$ and standard deviation $\hat{s}(\mathbf{r})$ are needed to supervise the radiance field according to the learned depth prior (Sec. 3.1). The NeRF depth estimate and standard deviation are computed from the rendering weights w_k:

$$\hat{z}(\mathbf{r}) = \sum_{k=1}^{K} w_k t_k , \quad \hat{s}(\mathbf{r})^2 = \sum_{k=1}^{K} w_k (t_k - \hat{z}(\mathbf{r}))^2 . \tag{8}$$

The network parameters θ_1 are optimized using a loss function $\mathcal{L}_{\theta_1}$ composed of a mean squared error (MSE) term on the color output $\mathcal{L}_{\text{color}}$ and a Gaussian negative log likelihood (GNLL) term on the depth output $\mathcal{L}_{\text{depth}}$:

$$\mathcal{L}_{\theta_1} = \sum_{\mathbf{r} \in R} \left(\mathcal{L}_{\text{color}}(\mathbf{r}) + \lambda \mathcal{L}_{\text{depth}}(\mathbf{r})\right), \tag{9}$$

$$\mathcal{L}_{\text{color}}(\mathbf{r}) = \left\| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2 , \tag{10}$$

$$\mathcal{L}_{\text{depth}}(\mathbf{r}) = \begin{cases} \log\left(\hat{s}(\mathbf{r})^2\right) + \frac{(\hat{z}(\mathbf{r}) - z(\mathbf{r}))^2}{\hat{s}(\mathbf{r})^2} & \text{if } P \text{ or } Q \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

$$\text{where} \quad P = |\hat{z}(\mathbf{r}) - z(\mathbf{r})| > s(\mathbf{r}) , \tag{12}$$

$$Q = \hat{s}(\mathbf{r}) > s(\mathbf{r}) . \tag{13}$$

Here $z(\mathbf{r})$ and $s(\mathbf{r})$ are the target depth and standard deviation from the corresponding Z_i^{dense} and S_i. The depth loss is applied to rays where at least one of the following conditions is true: 1) the difference between the predicted and target depth is greater than the target standard deviation Eq. (12), or 2) the predicted standard deviation is greater than the target standard deviation Eq. (13). This way, the loss encourages NeRF to terminate rays within one standard deviation of the most certain surface observation in the depth prior. At the same time, NeRF retains some freedom to allocate density to best minimize the color loss. The effectiveness of this depth loss in contrast to MSE is shown in the ablation study (Sec. 4.4).

Depth-Guided Sampling In addition to the depth loss function, the depth prior contains valuable signal to guide sampling along a ray. To render one pixel of a room-scale scene, we require the same number of MLP queries as the original NeRF; however, we replace the coarse network used for hierarchical sampling. During optimization, half of the samples are distributed between the near and far planes and the second half are drawn from the Gaussian distribution determined by the depth prior $\mathcal{N}(z(\mathbf{r}), s(\mathbf{r})^2)$. At test time, when the depth is unknown, the first half of the samples are used to render an approximate depth $\hat{z}(\mathbf{r})$ and standard deviation $\hat{s}(\mathbf{r})$ that is then used to sample the second half according to $\mathcal{N}(\hat{z}(\mathbf{r}), \hat{s}(\mathbf{r})^2)$.

4. Results

We evaluate our method with a baseline comparison (Sec. 4.3) and an ablation study (Sec. 4.4) on the Scan-Net [6] and Matterport3D [2] datasets.

4.1. Experimental Setup

ScanNet   We run COLMAP SfM [23] to obtain camera parameters and sparse depth. Specifically, we run SfM on all images to obtain camera parameters. To ensure a clean split between train and test data, we withhold the test images when computing the point cloud used for rendering the sparse depth maps. On average, the resulting depth maps have 0.04% valid pixels. We use three sample scenes each with 18 to 20 train images and 8 test images. This set of images results from excluding video frames with motion blur while ensuring that surfaces are observed from at least one input view (see details in the supplement).

Matterport3D   Using RGB images from the PrimeSense camera, COLMAP SfM struggled to reconstruct complete rooms in Matterport3D, hence, we mimic sparse depth from SfM by sampling and perturbing the sensor depth as described for depth prior training in Sec. 3.1. Sparse depth maps rendered from a SfM point cloud are by nature 3D-consistent. While consistency in 3D is irrelevant for training 2D depth completion, it plays a critical role when optimizing a 3D scene representation with NeRF. Hence, we ensure 3D-consistent sparse depth on the scenes used for NeRF by projecting the sampled and perturbed 3D points to all other views. On average the resulting depth maps are 0.1% complete. The impact of the sparse depth density is studied in the supplement. We evaluate three example rooms each with 24 to 36 train images and 8 test images.

NeRF Optimization   We process rays in batches of 1024 and use the Adam optimizer [16] with learning rate 0.0005. For fairness, all approaches in the ablation and baseline experiments are configured to use 256 MLP evaluations per pixel, independent of the used sampling approach. The radiance fields are optimized for 500k iterations. Further NeRF and depth prior implementation details are available in the supplement.

Evaluation Metrics   For quantitative comparison, we compute the peak signal-to-noise ratio (PSNR), the Structural Similarity Index Measure (SSIM) [25] and the Learned Perceptual Image Patch Similarity (LPIPS) [28] on the RGB of novel views as well as the root-mean-square error (RMSE) on the expected ray termination depth of NeRF against the sensor depth in meters. By comparing color values directly, PSNR has limited expressiveness, when the images of the scene have inconsistent color. As shown in

| Dataset | RMSE [m] ↓ | |
	Sparse depth	Dense depth
ScanNet	0.261	0.268
Matterport3D	0.041	0.135

Table 1. Accuracy of depth priors.

Sec. 4.4, the latent codes used to represent view-specific appearance largely help to produce consistent colors across the scene. Still, the color of a rendered image will not necessarily be similar to that of the test view against which it is evaluated. To compensate for this difference, we report an additional PSNR value, which is computed after optimizing for the latent codes on the entire test views. We are unable to use the left/right image split evaluation procedure from NeRF-W [19], because appearance changes too drastically across the image, so these numbers should be considered an upper bound on performance. This additional metric is listed in parentheses (Tabs. 2 and 3) for all approaches that use a latent code. All other metrics as well as all renderings in the paper are computed by setting the latent code to zero, given that the codes are unknown at test time.

4.2. Depth Priors

Tab. 1 shows the depth prior accuracy on the three Scan-Net and three Matterport3D scenes used for NeRF. These scenes are part of the test sets during depth completion training. The higher quality, generated sparse depth on Matterport3D leads to more accurate dense depth priors. However, the network interpolates the more noisy sparse depth from SfM on ScanNet without a relevant drop in accuracy.

4.3. Baseline Comparison

We compare our method to NeRF [21] as well as recent and concurrent work that equally uses sparse depth input in NeRF, namely Depth-supervised NeRF (DS-NeRF) [10] and NerfingMVS [26]. Since DS-NeRF and NerfingMVS rely on SfM and MVS depth, respectively, they are run on ScanNet. NeRF and our method are run on both datasets. The quantitative results (Tab. 2) show that our method outperforms the baselines in all metrics.

"Floaters" are a common problem when applying NeRF approaches in a setting with few input views. By using dense depth priors with uncertainty, our method strongly reduces these artifacts compared to the baselines (example 2 Fig. 3). This contributes to far more accurate depth output and greater detail in color, e.g., visible in the books and the door handle (example 3 Fig. 3). We found that our method is more robust to outliers in the sparse depth input. E.g., erroneous SfM points in the area of the sofa back (example 5 Fig. 3) cause much larger deficiencies in geometry and color of the other approaches. This suggests that dense depth priors with uncertainty focus the optimization

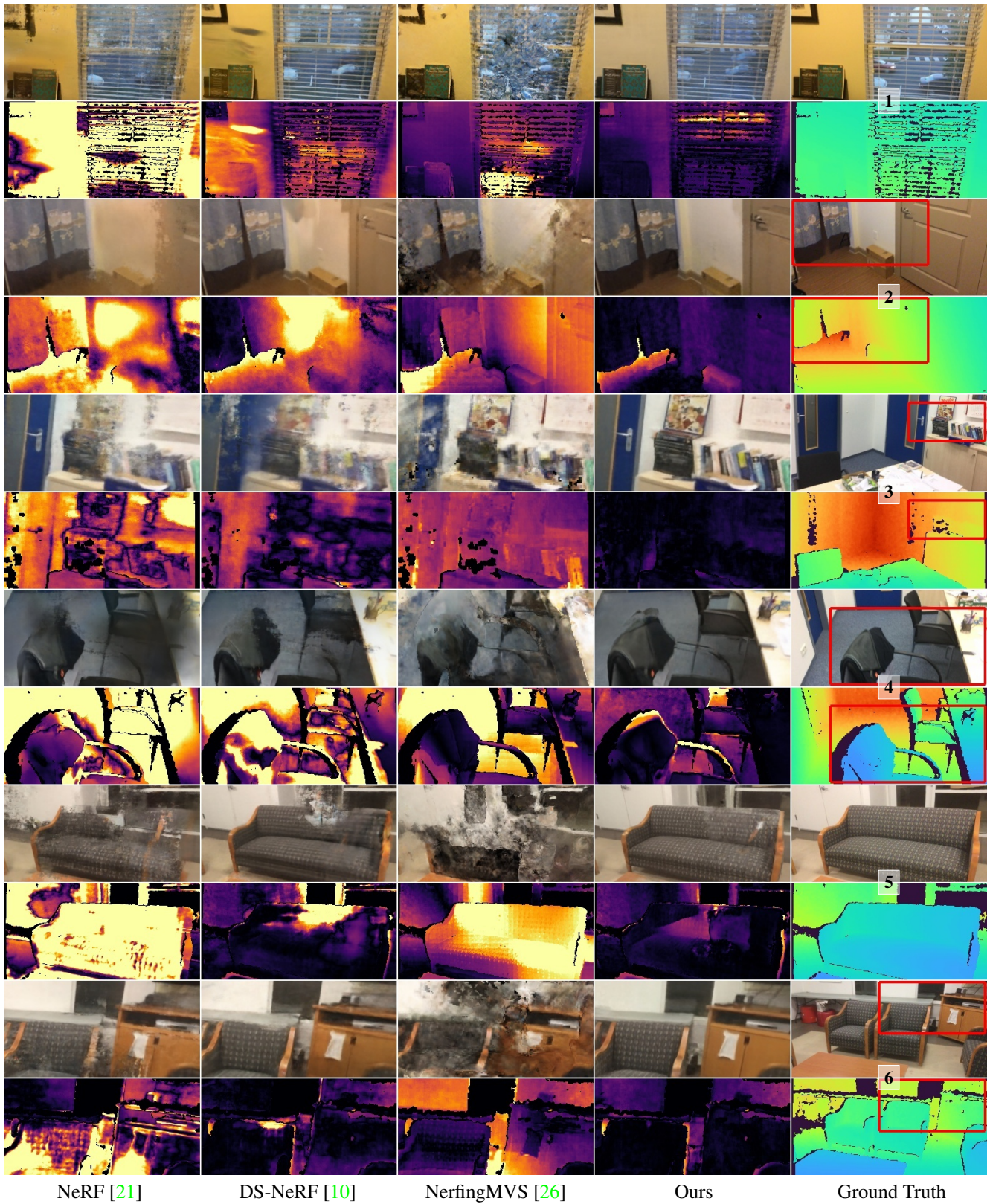| NeRF [21] | DS-NeRF [10] | NerfingMVS [26] | Ours | Ground Truth |

Figure 3. Rendered RGB and depth error for test views from three ScanNet rooms next to the ground truth RGB and depth.

on more certain and accurate views, while direct incorporation of sparse depth, as in DS-NeRF, is more error-prone. Besides greater robustness to outliers, dense depth guides NeRF better at object boundaries that are not represented in the sparse depth input. This is observable in example 6 (Fig. 3), where a part of the chair back is missing in DS-NeRF, while it is complete using our method.

NerfingMVS Details The error map calculation in NerfingMVS fails when applied to an entire room as opposed to a local region, causing invalid sampling ranges. The issue is solved as detailed in the supplement. To improve the performance of this baseline, we train its depth predictor 10 epochs longer than was done in the paper. Still, the depth priors on the ScanNet scenes remain at RMSE
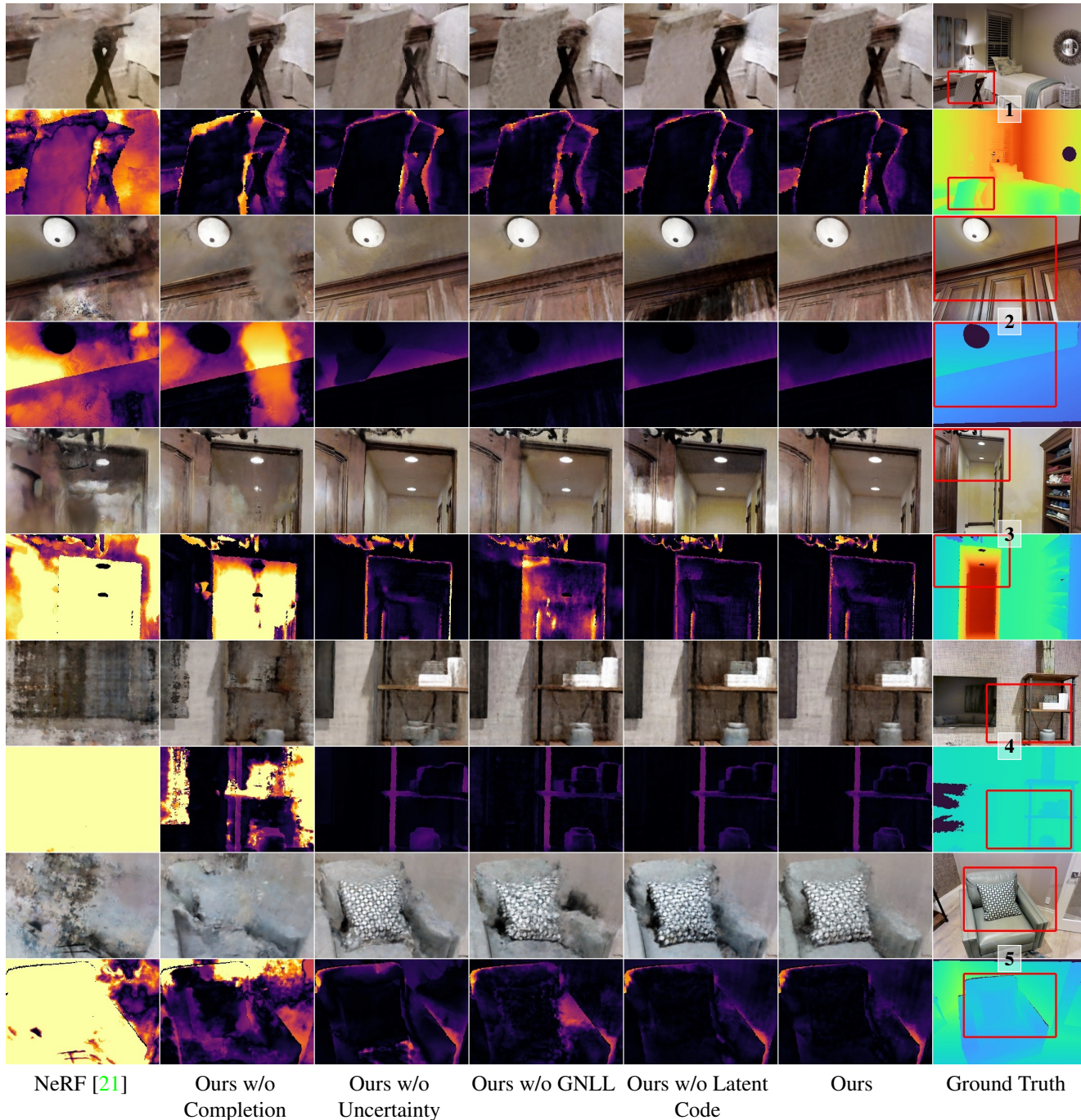

| NeRF [21] | Ours w/o Completion | Ours w/o Uncertainty | Ours w/o GNLL | Ours w/o Latent Code | Ours | Ground Truth |

Figure 4. Rendered RGB and depth error for test views from three Matterport3D rooms next to the ground truth RGB and depth.

Method	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE↓
NeRF [21]	19.03	0.670	0.398	1.163
DS-NeRF [10]	20.85	0.713	0.344	0.447
NerfingMVS [26]	16.29	0.626	0.502	0.482
Ours w/o Completion	20.43 (22.10)	0.707	0.366	0.526
Ours w/o Uncertainty	20.09 (22.21)	0.714	0.308	0.279
Ours w/o GNLL	20.80 (22.23)	0.733	0.312	0.275
Ours w/o Latent Code	20.87	0.726	**0.293**	0.243
Ours	**20.96 (22.30)**	**0.737**	0.294	**0.236**

Table 2. Quantitative results on ScanNet. Parentheses indicate PSNRs obtained after optimizing a latent code, when applicable.

0.379m. While our method uses only train images to compute the sparse depth input, NerfingMVS runs COLMAP MVS on train and test images together, which gives them an advantage.

4.4. Ablation Study

To verify the effectiveness of the added components, we conduct ablation experiments on the ScanNet and Matterport3D scenes. The quantitative results (Tabs. 2 and 3) show that the full version of our method achieves the best performance in image quality and depth estimates. This is consistent with the qualitative results in Fig. 4.

Without Completion Omitting depth completion and supervising with sparse depth only leads to inaccurate depth and color due to "floaters" in areas without depth input. Even in areas with sparse depth points, the results are less sharp than in versions that use completed depth.

Without Uncertainty Removing uncertainty from the optimization causes problems in resolving inconsistency in overlapping areas of the 2D depth priors. This results in wrong edges in RGB and depth (example 2 Fig. 4), duplication artifacts (example 4 Fig. 4) or lacking detail, e.g., in the patterns on the back of the chair (example 1 Fig. 4). The quantitative results on ScanNet (Tab. 2) show that considering uncertainty becomes even more important when using the lower quality sparse depth from SfM.

Without GNLL In this experiment, we replace GNLL with MSE in our depth loss (Eq. (11)), and observe that MSE struggles to constrain density behind surfaces. The lack of sharp edges in the density distribution is most visible for novel views looking in tangential direction of a surface, e.g., looking into the corridor (example 3 Fig. 4).

Without Latent Code Omitting the latent code that models per-camera information, leads to incapability to produce smooth and consistent color output across the scene. When rendering novel views, the frustums of training images are clearly visible by causing severe shifts in color intensity (examples 2 and 3, Fig. 4).

Method	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE↓
NeRF [21]	15.24	0.531	0.610	1.362
Ours w/o Completion	16.90 (18.84)	0.615	0.521	0.427
Ours w/o Uncertainty	17.95 (20.37)	0.658	0.413	0.115
Ours w/o GNLL	18.00 (20.65)	0.669	0.423	0.133
Ours w/o Latent Code	17.42	0.654	0.410	**0.110**
Ours	**18.33 (20.82)**	**0.673**	**0.402**	0.114

Table 3. Quantitative results on Matterport3D, using the same format as Tab. 2.

4.5. Limitations

Our method allows for a significant reduction in the number of input images for NeRF-based novel view synthesis while at the same time applying it to larger room-size scenes. However, other NeRF limitations such as long optimization times and slow rendering remain. As a consequence of the drastic reduction in the number of input images, surfaces are typically not observed by more than two other views, hence view-dependent effects are limited. While our approach optimizes NeRF given as few as 18 images, the depth prior network requires a larger training dataset. Although these priors generalize well and only need to be trained once, it would be beneficial if the depth reconstruction could be also learned from a sparse setting.

5. Conclusion

We have presented a method for novel view synthesis using neural radiance fields (NeRF) that leverages dense depth priors, thus facilitating reconstructions with only 18 to 36 input images for a complete room. By learning a depth prior that generalizes across scenes, our method takes advantage of depth information without requiring depth sensor input of the scene. Instead, the depth prior network relies on the sparse reconstruction, which is available for free after structure from motion (SfM) on the input images. With only a few input views available, we show that our dense depth priors with uncertainty effectively guide the NeRF optimization, thus leading to significantly higher image quality of novel views and more accurate depth estimates compared to other approaches using SfM or multi-view stereo output in NeRF. Overall, we believe that our method is an important step towards making NeRF reconstructions available in commodity settings.

References

[1] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432, 2001. 2

[2] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *3DV*, 2017. 2, 3, 5

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 2

[4] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE TPAMI*, 42, 2020. 2, 3

[5] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CVPR*, 2017. 2, 3, 5

[7] Angela Dai, Christian Diller, and Matthias Niessner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[8] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018. 2

[9] 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*, SIGGRAPH '96, page 11–20, 1996. 2

[10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *ArXiv*, abs/2107.02791, 2021. 2, 3, 5, 6, 8

[11] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proc. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2

[12] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54, 1996. 2

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 3

[14] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. 2018. 2

[15] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, October 2011. 2

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[17] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. 2

[18] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (SIGGRAPH)*, 2019. 2

[19] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 3, 5

[20] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima K. Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (SIGGRAPH)*, 2019. 2

[21] 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. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[22] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *ECCV*, 2020. 2

[23] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 5

[24] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, 2019. 2

[25] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 5

[26] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 2, 5, 6, 8

[27] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 287–296, 2000. 2

[28] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5

[29] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (SIGGRAPH)*, 2018. 2