

SceneRF:

Self-Supervised Monocular 3D Scene Reconstruction with Radiance Fields

Anh-Quan Cao

Raoul de Charette

Inria, Paris, France

<https://astra-vision.github.io/SceneRF>

Abstract

In the literature, 3D reconstruction from 2D image has been extensively addressed but often still requires geometrical supervision. In this paper we propose SceneRF, a self-supervised monocular scene reconstruction method with neural radiance fields (NeRF) learned from multiple image sequences with pose. To improve geometry prediction, we introduce new geometry constraints and a novel probabilistic sampling strategy that efficiently update radiance fields. As the latter are conditioned on a single frame, scene reconstruction is achieved from the fusion of multiple synthesized novel depth views. This is enabled by our spherical-decoder which allows hallucination beyond the input frame field of view. Thorough experiments demonstrate that we outperform all baselines on all metrics for novel depth views synthesis and scene reconstruction. Our code is available at <https://astra-vision.github.io/SceneRF>.

1. Introduction

Humans evolve in a 3D physical world where even the slightest motion requires a thorough understanding of their surroundings to plan displacements and avoid collisions. While binocular vision provides an evident edge, physiological studies suggest that humans can sense depth even with only monocular vision [28]. Despite a long-standing line of research [63, 68, 78] this ability is yet unequaled by computer vision algorithms, which mostly rely on multiple-views to reconstruct complex scenes [56]. However, estimating 3D from a single view would unveil novel applications in a world flooded with consumer cameras where mobile robots, like autonomous cars, still require costly depth sensors [3, 5].

From a 3D computer vision perspective, a small portion of the field addressed reconstruction of complex scenes from a single image [7, 11, 24, 79] although the latter still learns from supervision with depth data. Meanwhile, the

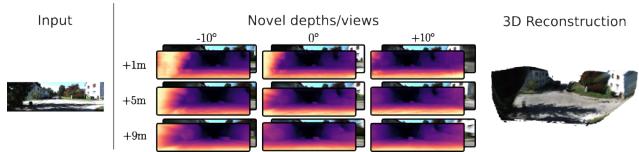


Figure 1. Self-supervised Monocular Scene Reconstruction. SceneRF trains self-supervisedly from sequences of images with poses. At inference, a *single input image* (left) suffices to synthesize highly divergent novel depth/views (middle), and obtain scene reconstruction of complex urban scenes (right).

recent advent of Neural Radiance Field [42] (NeRF) which optimizes a radiance field self-supervisedly from one or more views, unraveled many descendants for synthesizing novel views [72]. NeRF-based methods were for example applied to synthesis novel view, with unprecedented performance, but are usually limited to objects when it comes to single-view input [37, 44, 48]. Beside [32], to handle complex scene people train on synthetic data [60] or require additional geometrical cues to train on real data [12, 54, 56]. Reducing the need of supervision on complex scenes would lower dependency to costly-acquired datasets.

In this work, we address single-view reconstruction of complex and large 3D urban scenes, in a fully self-supervised manner. Our method, coined SceneRF, trains from sequences of posed images to optimize a neural radiance fields (NeRF), which is 100m deep, allowing at inference the use of a single RGB image from which it can predict novel depth/view synthesis and the complete scene reconstruction (Fig. 1). More in depth, we build upon PixelNeRF [75] that encodes NeRF conditioned on an image, generalizable to new scenes, and propose specific design choices to *explicitly* optimize depth. Because large scenes hold their own challenges, we also introduce a novel probabilistic ray sampling to efficiently choose the sparse location to optimize within the wide radiance volume, and introduce a spherical U-Net, which aims is to enable hallucination beyond the input image field of view.

- We build on custom design choices to explicitly optimize depth (Sec. 3.1) and propose an efficient spherical U-Net (SU-Net, Sec. 3.3) – altogether allowing use of our radiance field for scene reconstruction (Sec. 3.4).
- We introduce a probabilistic ray sampling strategy (PrSamp, Sec. 3.2) which learns to represent the continuous density volume with a mixture of gaussians – boosting both performance and efficiency,
- To the best of our knowledge, we propose the first self-supervised large scene reconstruction method using a single-view as input. Results on challenging driving scenes show that our method outperforms even baselines that are depth-supervised (Sec. 4).

2. Related work

With the seminal NeRF [42], the recent literature on 3D from images has gone wild, as surveyed in [72]. Hence, we limit this section to the smaller portion of works using **single view input at inference**, and study the literature along two axes related to our work: *novel views/depths synthesis* and *3D reconstruction*. For disambiguation, hereafter ‘novel view’ always refer to novel RGB view synthesis.

Novel views/depths synthesis. Rendering novel view from an image is a long-lasting research problem [22, 49, 66, 73] although most recent works rely on generalizable NeRFs like PixelNerf [75], MINE [32], or GRF [67] which learn a representation generalizable to unseen input images. The almost entire single-view literature however focuses on objects which hold specific challenges such as shape and appearance disentanglement [27, 55], exploiting symmetry priors [34], or category-centric/agnostic view synthesis [36, 53]. While most methods consider input image with a single object on a plain background, others like CO3D [53] handle objects on cluttered scenes, or large-scale scenes: synthetic as in SEE3D [60], or real as in MINE [32] or AutoRF [44]. With a single input view, only MINE [32] seems to address novel view/depth synthesis of real driving scenes – highly cluttered by nature.

In general, **depth supervision** is shown to improve quality and convergence speed [6, 12, 54, 56], leveraging for example structure from motion [12, 56] or Lidar data [54]. While all NeRF-based methods implicitly optimize depth, those doing it explicitly require supervision. Instead, we explicitly optimize depth in a self-supervised manner. Since all NeRF-based methods optimize radiance field only at sparse locations, **efficient sampling strategy** is needed to handle large scenes [45]. Departing from the original hierarchical sampling [42], a log warping strategy was proposed in DONeRF [45] using depth maps as supervision, while [30] uses a pretrained NeRF, and [30]

employs dual sampling-shading networks in a 4-stage training scheme. We inspire from above works but instead learn to approximate the continuous density volume as a mixture of Gaussians from which we can efficiently sample, without any complex setup.

3D reconstruction While early deep methods focused on reconstruction with explicit representations: like voxels [71], point clouds [1, 15, 74] or meshes [9, 35, 69], recently, implicit representations became more popular [25, 40, 50–52, 73]. A common practice, for 3D object reconstruction is to employ object detectors [17, 20, 26, 29, 80]. A number of works also addressed holistic 3D scene understanding seeking prediction of geometry and semantics for indoor [11, 14, 24, 31, 47, 64, 79, 82], outdoor scenes [76], and both [7]. If semantic and geometry are estimated jointly the task is referred as semantic scene completion, recently surveyed in [58], when image input is complemented with geometrical cues. Among above works, MonoScene [7] is the closest to us in spirit, but requires full supervision. There are few alternatives for **self-supervised 3D reconstruction**. The naive strategy leveraging monocular depth estimation, reviewed in [43], inherently restricts reconstruction to the visible surface. A popular self-supervised line of research uses differentiable renderers, trained with a set of views and poses [13, 48, 62]. To alleviate the need of color rendering, [21] optimizes silhouettes and [83] a 2D projection. Despite impressive visual results, these methods remain focused on objects, sometimes on cluttered background scenes. Instead, we learn scene reconstruction self-supervisedly by training only with images and poses.

3. SceneRF

SceneRF learns to infer the scene geometry from a single monocular RGB image, training in a self-supervised manner with image-conditioned Neural Radiance Fields (NeRFs) [42, 75]. Given a training set made of S sequences, each having m RGB images and their corresponding poses, denoted $\{(I_1^i, P_1^i), \dots, (I_m^i, P_m^i)\}_{i=1}^S$, we estimate a neural representation conditioned on each first frame $\{I_1^i\}_{i=1}^S$. The conditioning learned is shared across sequences and self-supervisedly optimized by the other frames (*i.e.*, $\{I_2^i, \dots, I_m^i\}_{i=1}^S$). Subsequently, it can be used for 3D reconstruction from a single RGB image.

In Sec. 3.1 we first formulate our usage of NeRF for novel depth synthesis, detailing our strategy to explicitly optimize depth with a reprojection loss. We then detail two major components. In Sec. 3.2, we introduce a topology-preserving strategy to efficiently sample points close to the surface. And in Sec. 3.3, we detail our U-Net with a spherical decoder which goal is to hallucinate the scene *beyond* the input image field of view. Importantly, our choices are

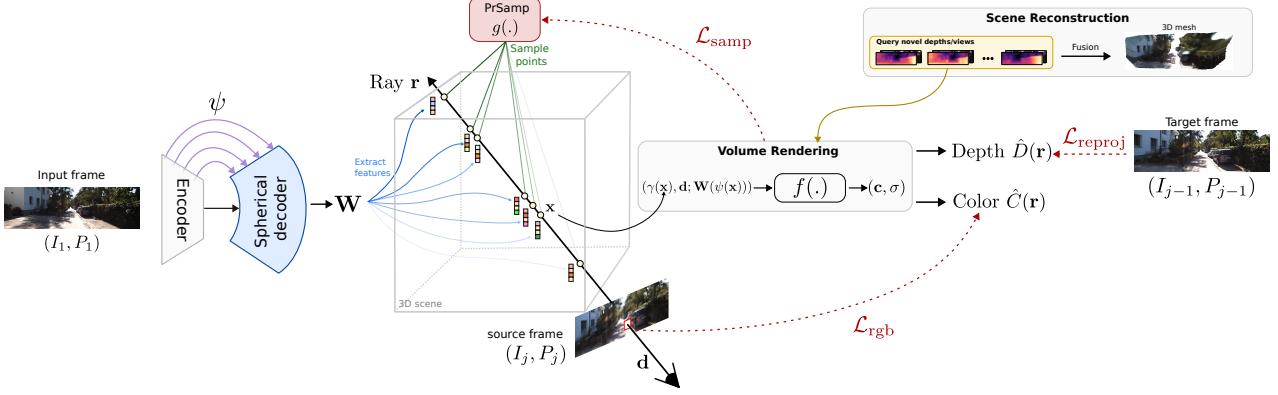


Figure 2. SceneRF leverages generalizable neural radiance field (NeRF) to generate novel depth views, conditioned on a single input frame. During training for each ray \mathbf{r} in addition to color \hat{C} , we explicitly optimize depth \hat{D} with a reprojection loss $\mathcal{L}_{\text{reproj}}$ (Sec. 3.1), introduce a Probabilistic Ray Sampling strategy (PrSamp, Sec. 3.2) to sample points more efficiently. To hallucinate features outside the input FOV, we propose a spherical U-Net (Sec. 3.3). Finally, the synthesized depths are used for scene reconstruction (Sec. 3.4).

driven by the need of good geometrical scene reconstruction, detailed in Sec. 3.4, though results are also on par with novel view synthesis.

3.1. NeRF for novel depth synthesis

In their original formulation, NeRFs [42, 75] optimize a continuous volumetric radiance field $f(\cdot) = (\sigma, \mathbf{c})$ such that for a given 3D point $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{R}^3$, it returns a density σ and RGB color \mathbf{c} . In the following, we build on PixelNeRF [75] to learn a generalizable radiance field across sequences, and introduce new design choices to efficiently synthesize novel depth views.

The training of SceneRF is illustrated in Fig. 2. Given the first *input* frame (I_1) of a sequence¹, we extract a feature volume $\mathbf{W} = E(I_1)$ with our SU-Net (Sec. 3.3). We then select randomly a *source* future frame I_j , $2 \leq j \leq m$, and randomly sample ℓ pixels from it. Given known *source* pose and camera intrinsics, we efficiently sample N points along the rays passing through these pixels (Sec. 3.2). Each sampled point \mathbf{x} is then projected on a sphere with $\psi(\cdot)$ so we can retrieve the corresponding *input* image feature vector $\mathbf{W}(\psi(\mathbf{x}))$ through bilinear interpolation. The latter is passed to the NeRF MLP $f(\cdot)$, along with viewing direction \mathbf{d} and positional encoding $\gamma(\mathbf{x})$, to predict the point density σ and RGB color \mathbf{c} in the input frame coordinates. Altogether, this writes:

$$f(\gamma(\mathbf{x}), \mathbf{d}; \mathbf{W}(\psi(\mathbf{x}))) = (\mathbf{c}, \sigma) \quad (1)$$

As in original NeRF [42], we apply quadrature to approximate the color $\hat{C}(\mathbf{r})$ of camera ray \mathbf{r} from colors sampled along the ray. For the sake of generality, we write it as:

¹For clarity from now on we drop the superscript i sequence index, but the process applies to all S sequences.

$$\hat{C}(\mathbf{r}) = \sum_i^N w_i \mathbf{c}_i \text{ where } w_i = T_i (1 - \exp(-\sigma_i \delta_i)), \quad (2)$$

with T_i the accumulated transmittance and δ_i is the distance to the previous adjacent point, as defined in [42].

3.1.1 Depth optimization

Unlike most NeRFs, we seek to unravel depth explicitly from the radiance volume and therefore define its estimation $\hat{D}(\mathbf{r})$ as:

$$\hat{D}(\mathbf{r}) = \sum_i^N w_i d_i, \quad (3)$$

where d_i is the distance of point i to the sampled position.

To optimize depth without ground-truth supervision, we inspire from self-supervised depth methods [18, 19], and apply a photometric reprojection loss between the warped *source* image I_j and its preceding frame I_{j-1} , referred as *target*. We choose consecutive frames to ensure maximum overlaps. Using sparse depth estimate \hat{D}_j , the photometric reprojection loss $\mathcal{L}_{\text{reproj}}$ writes:

$$\mathcal{L}_{\text{reproj}} = \frac{1}{\ell} \sum_{i=1}^{\ell} \| I_j(i) - I_{j-1} \left(\text{proj}(\hat{D}_j(i)) \right) \|_1, \quad (4)$$

with $\text{proj}(\cdot)$ the projection of 2d coordinates i in I_{j-1} using ad-hoc camera intrinsics and poses. Important note that while \hat{D}_j is sparse – since only estimated for *some* rays, Eq. (3) – the stochastic nature of these rays brings statistically dense supervision. To also account for moving objects, we apply the pixels auto-masking strategy from [19].

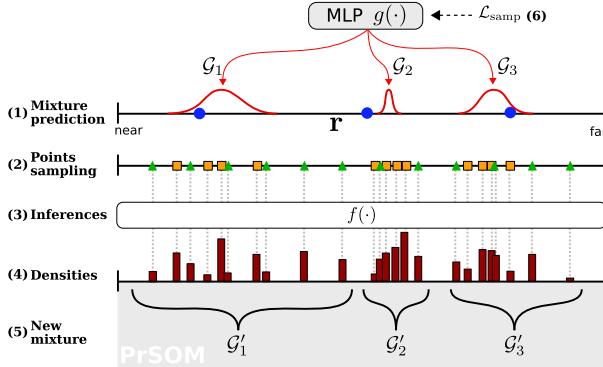


Figure 3. Steps of our Probabilistic Ray Sampling (for $k=3$ Gaussians and $m=4$ points per Gaussian). Refer to Sec. 3.2 for details.

3.2. Probabilistic ray sampling (PrSamp)

Prior works [23, 42, 45] show the importance of sampling points close to the surface during volume rendering as it boosts performance [45] and lowers computational cost with less $f(\cdot)$ inferences in Eq. (1). Arguably this is even more crucial for depth estimation. Yet, efficient sampling is non-trivial in large urban scenes without ground truth depth.

We address this with a probabilistic ray sampling strategy (PrSamp) which, in substance, approximates the continuous density along each ray as a mixture of 1D Gaussians guiding the points sampling. Because large values in the mixture correlate with surface locations, this allows better sampling with significantly fewer points per ray. In practice we use only 64 points to optimize a ray being 100m long.

Referring to symbols and steps in Fig. 3, for a given ray \mathbf{r} we first uniformly sample k points (●) between *near* and *far* bounds. (1) Taking as input the points ● and their corresponding features, a dedicated MLP $g(\cdot)$ predicts a mixture of k 1D Gaussians $\{\mathcal{G}_1, \dots, \mathcal{G}_k\}$. (2) We then sample m points per Gaussian (■) and 32 more points uniformly (▲) along the ray; which amounts to $N=k \times m \blacksquare + 32 \blacktriangle$ points. Note that the additional uniform points sampling enforces exploration of the volume, thus avoiding $g(\cdot)$ from falling into local minima. (3) All points are passed to $f(\cdot)$ in Eq. (1) for NeRF volume rendering of color $\hat{C}(\mathbf{r})$ and depth $\hat{D}(\mathbf{r})$. (4) Subsidarily, the densities $\{\sigma_1, \dots, \sigma_N\}$ inferred during rendering act as cues for 3D surface locations that we use to get a new mixture of Gaussians. Doing so requires solving a points-Gaussians assignment problem, (5) which we solve using Probabilistic Self-Organizing Maps (PrSOM) from [2]. In a nutshell, PrSOM assigns points to Gaussians from the likelihood of the former to be observed by the latter, while strictly preserving the underlying mixture topology. For each Gaussian \mathcal{G}_i and its assigned points \mathcal{X}_i , the new Gaussian \mathcal{G}'_i is an average of all points $j \in \mathcal{X}_i$ weighted by the conditional probability $p(j|\mathcal{G}_i)$ defined in [2] and α_j the occupancy probability of j . In practice, we use alpha

values from [42] which are good enough occupancy estimators: $\alpha_j = 1 - \exp(-\sigma_j \delta_j)$ with δ_j the distance to previous point.

Finally, (6) the Gaussians predictor $g(\cdot)$ is updated from the mean of KL divergences between current and new Gaussians:

$$\mathcal{L}_{\text{gauss}} = \frac{1}{k} \sum_i^k \text{KL}(\mathcal{G}_i || \mathcal{G}'_i). \quad (5)$$

To further enforce one Gaussian *on the visible surface*, we also minimize distance between depth and closest Gaussian:

$$\mathcal{L}_{\text{surface}} = \min_i (\|\mu(\mathcal{G}'_i) - \hat{D}(\mathbf{r})\|_1). \quad (6)$$

The complete loss is the sum: $\mathcal{L}_{\text{samp}} = \mathcal{L}_{\text{gauss}} + \mathcal{L}_{\text{surface}}$.

In practice, we use $k = 4$ Gaussians and $m = 8$ points per Gaussians, leading to only $N = 64$ points per ray. More details are in Appendix A.1.

3.3. Spherical U-Net (SU-Net)

By definition, the validity domain of $f(\cdot)$ is restricted to the feature volume $\mathbf{W}(\cdot)$ which for a simple U-Net is within image FOV, thus preventing estimation of color and depth (Eqs. 2,3) outside of the FOV where features cannot be extracted. This is unsuitable for scene reconstruction.

Instead, we equip our SU-Net with a decoder convolving in the spherical domain. Because spherical projection has less distortion than its planar counterpart [59], we can enlarge the FOV (typically, approx. 120°) to enable hallucination of color and depth outside of the source image FOV.

At the bottleneck, the encoder features are mapped to an arbitrary sphere with $\psi(\cdot)$ and passed to our spherical decoder. Given the wide feature space, we employ lightweight dilated convolutions in the spherical decoder in order to increase the receptive field at low cost. As in standard U-Net, we use multi-scale skip connections to enhance gradient flow simply by mapping features with $\psi(\cdot)$.

In practice, we map a 2D pixel $[x, y]^T$ to its *normalized* latitude-longitude spherical coordinates $[\theta, \phi]$. Considering $[\nabla_x, \nabla_y, 1]^T \sim \mathbf{K}^{-1} [x, y, 1]^T$ a ray passing through said pixel and the camera center. The projection writes:

$$\psi \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \pi - \arctan(\nabla_x^{-1}) \\ \arccos(-\nabla_y/r) \end{pmatrix} \quad (7)$$

where $r = \sqrt{\nabla_x^2 + \nabla_y^2 + 1}$. When inputted in the decoder, $[\theta, \phi]$ are discretized uniformly and features stored in a tensor that covers an arbitrary large FOV.

3.4. Scene reconstruction

With prior sections, SceneRF is now equipped with novel depth synthesis capability that allows us to synthesize depth that significantly diverges from the source input

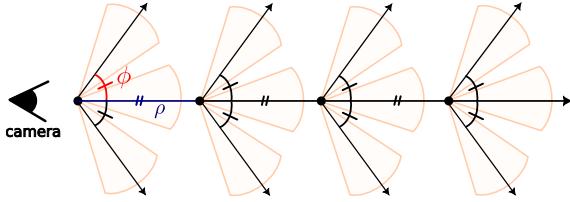


Figure 4. Given an input image, we synthesize novel depth views uniformly along an imaginary straight path and for varying angles. The novel depths are later composed into a single scene volume.

position. We use this ability to frame scene reconstruction as the composition of multiple novel depth views.

As illustrated in Fig. 4, given an input frame we synthesize novel depths along an imaginary straight path, uniformly every ρ meters for up to 10 meters. At each position, we also vary the horizontal viewing angles $\Phi = \{-\phi, 0, \phi\}$.

Synthesized depths are then converted to TSDF using [77] and the overall scene TSDF for voxel v is obtained using the minimum of all: $V(v) = \text{TSDF}_{\arg\min_i |\text{TSDF}_i(v)|}(v)$, where i spans all synthesized depths. Traditionally, a voxel TSDF is the weighted average of all TSDFs [10, 46], but we empirically show (see Appendix A.2) that using the minimum leads to better results. We conjecture that this relates to the linearly increasing depth error with distance.

4. Experiments

We evaluate SceneRF on two primary tasks: novel depth synthesis and scene reconstruction, and a subsidiary task: novel view synthesis. For the main tasks, we outperform all baselines on all metrics, while for novel view synthesis we remain competitive with recent single-image NeRFs baselines [32, 36, 75]. Finding datasets meeting our requirements is a challenge so we report results on SemanticKITTI [3, 16] for the three tasks, but considering exhaustive supervisions setups. Since we first address *self-supervised* monocular scene reconstruction from RGB images, we detail our non-trivial adaptation of monocular reconstruction baselines [7, 8, 33] (Sec. 4.1). Bare in mind that all baselines are more supervised than us.

Unless mentioned otherwise, we use $k = 4$ gaussians and $m = 4$ points per Gaussians for PrSamp (Sec. 3.2) and render depths every $\rho = 0.5m$ at 3 angles $\Phi = \{-10, 0, +10\}$ for scene reconstruction (Sec. 3.4).

Dataset. SemanticKITTI [3] contains pairs of outdoor geolocalized images with lidar scans voxelized as 256x256x32 grid of 0.2m voxels, with semantic labels. Since we only consider geometry, regardless of their semantics we regard all voxels *not* free as occupied. We use the standard train/val split as in [3, 7], and left-crop RGB images to 1220x370. To train SceneRF, we extract sequences

of successive frames spanning $\approx 10m$ while ensuring a minimum of 0.4m distance between two frames of our sequence. Poses are from IMU/GPS data, relative to the first frame of each sequence. This results in 10,270 training sequences. For evaluation, the synthesized RGB images are evaluated at 1:3 resolution, and novel depth at 1:2 against lidar data projected as sparse ground-truth depths.

Metrics. To evaluate scene reconstruction quality, we compute the intersection over union (IoU), precision, and recall of occupied voxels. For novel depth estimation, we follow common practice, *i.e.*, capping depth to 80m, and computing usual metrics [19]: relative error absolute (Abs Rel) or squared (Sq Rel), root mean squared error (RMSE), mean \log_{10} error (RMSE log), threshold accuracies ($\delta_1, \delta_2, \delta_3$). Following [32], we measure quality of synthesized RGB images with: Structural Similarity Index (SSIM) [70], PSNR, and LPIPS perceptual similarity [81].

Training setup. Ultimately, SceneRF is trainable end-to-end using the loss: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reproj}} + \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{samp}}$ where \mathcal{L}_{rgb} is the standard L2 photometric reconstruction loss used in NeRFs [42, 54, 75]. We train on 4 Tesla v100 GPU with AdamW batch size of 4. Main results were trained for 50 epochs (≈ 5 days) and all ablations for 20 epochs (≈ 2 days). The initial learning rate is set to 1e-5 and exponentially decayed at each epoch with gamma 0.95.

4.1. Baselines

Novel depth/views. Despite the buzzing NeRF field, there are in fact few *single-view* NeRFs. We select 3 of them among the best open-sourced ones for novel depths/views synthesis: PixelNeRF [75], VisionNeRF [36], MINE [32]. Similar to us, all train with images and poses.

Scene reconstruction. For monocular scene reconstruction, we consider 4 baselines being: MonoScene [7], LMSCNet^{rgb} [57], 3DSketch^{rgb} [8], AICNet^{rgb} [33]. The three latter baselines^{rgb} are *RGB-inferred version* from [7]. Unlike us all baselines require geometric supervision so for completeness we report 2 types of setups: (i) The standard ‘3D’ supervision setup where baselines are trained with the original 3D ground-truth. (ii) ‘Depth’ supervision where 3D labels come from the fusion of *supervised depth sequences* relying on [4] depthmaps – a lidar/stereo *supervised* method. Setups are described in Sec. 4.3 and more implementations details are in Appendix C. Importantly, **all baselines are more supervised than us**.

4.2. Novel depth synthesis

We first evaluate our NeRF ability to generate novel depth and novel RGB views. To do so, given an input image we synthesize novel depths and views at the position of any future frames within at most 10 meters from input frame.

Method	Novel depth synthesis								Novel view synthesis		
	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta 1\uparrow$	$\delta 2\uparrow$	$\delta 3\uparrow$	LPIPS↓	SSIM↑	PSNR↑	
PixelNeRF [75]	0.2364	2.080	6.449	0.3354	65.81	85.43	92.90	0.489	0.466	15.80	
MINE [32]	0.2248	1.787	6.343	0.3283	65.87	85.52	93.30	0.448	0.496	16.03	
VisionNerf [36]	<u>0.2054</u>	<u>1.490</u>	<u>5.841</u>	<u>0.3073</u>	<u>69.11</u>	<u>88.28</u>	<u>94.37</u>	<u>0.468</u>	<u>0.483</u>	16.49	
SceneRF (ours)	0.1681	1.291	5.781	0.2851	75.07	89.09	94.50	0.476	0.482	16.46	

Table 1. Novel depth/view synthesis evaluation on SemanticKITTI (val. set). We outperform all baselines with SceneRF on our main task of novel depth synthesis, and perform on par on novel views.

	Supervision			in-FOV			Whole Scene		
	3D	Depth	Image	IoU	Prec.	Rec.	IoU	Prec.	Rec.
LMSCNet ^{rgb} [57]	✓			38.36	59.12	52.21	35.84	55.70	57.12
3DSketch ^{gb} [8]	✓			36.69	40.10	81.17	34.42	37.85	79.12
AICNet ^{rgb} [33]	✓			33.11	34.95	86.23	30.38	32.27	84.59
MonoScene [7]	✓			39.06	51.91	61.20	37.14	49.90	59.24
LMSCNet ^{rgb} [57]		✓		12.75	13.09	83.44	12.08	13.00	63.16
3DSketch ^{gb} [8]		✓		12.65	13.01	82.18	12.01	12.95	62.31
AICNet ^{rgb} [33]		✓		11.78	11.93	90.57	11.28	11.84	70.89
MonoScene [7]		✓		14.80	17.04	53.04	13.53	16.98	40.06
SceneRF (ours)			✓	15.18	17.67	51.87	13.84	17.28	40.96

Table 2. Scene reconstruction on SemanticKITTI [3] (val. set). Despite being the *only* self-supervised method, we outperform all ‘Depth’ supervised baselines. The type of supervision refers to: ‘3D’ ground-truth, ‘Depth’ sequences from the *supervised depth method* AdaBins [4], and self-supervised ‘Image’ sequence. We do not highlight best results purposely as the supervision differs.

From Tab. 1, we outperform all baselines on *novel depth synthesis* with a comfortable margin. In particular, we note the significant gap with the second best method, VisionNerf [36], on AbsRel (0.1681 vs. 0.2054) and $\delta 1$ (75.07 vs 69.11) which are two challenging metrics. The smaller gap in RMSE shows our depth improvement is better for close distances. Results also advocate that our design choices always improve over PixelNeRF from which we depart, and benefit depth generation which is not a natural NeRF capability. We also evaluate on the subsidiary task of *novel views synthesis*, showing in Tab. 1 that SceneRF is roughly on par with others. In particular, we outperform PixelNeRF on 2 out of 3 metrics – which notably highlight that our geometric objectives does not degrade view synthesis.

In Fig. 5 we primarily show novel depths (our main task) and novel views for varying input frames, multiple positions and angles w.r.t. the input frame position. For all, novel depths are visually outperforming the baselines. In particular, we note the sharper depth edges and the better quality at far when zooming in. When varying the viewing angle (*i.e.*, -10° or $+10^\circ$) we note also fewer edge artefacts than baselines, which is even more striking for the 2nd and 3rd examples. Please also refer to the supplemental video.

4.3. 3D reconstruction results.

We evaluate 3D reconstruction on SemanticKITTI (val set) in Tab. 2 by comparing outputs with the voxelized ground-truth, and reporting performance for both the complete scene (‘Whole Scene’) or only the volume within the input camera frustum (‘in-FOV’).

Since supervision affects significantly the performance, each of our 4 baselines uses 2 supervision setups: ‘3D’ where baselines are trained with full 3D ground truth coming from the accumulation of lidar scans, and ‘Depth’ using as supervision the TSDF fusion [77] of depth sequences from the *supervised AdaBins* method [4]. SceneRF is the only one to train self-supervised from ‘Image’ sequences. It is important to note that all baselines incorporate some sense of ground truth depth which we do not.

From Tab. 2, ‘3D’ baselines perform twice better than any others — a logical outcome given the importance of geometrical cues for scene reconstruction. The storyline however evolves when comparing SceneRF, ‘Image’ supervised, versus ‘Depth’ supervised baselines as we outperform all. This is surprising given the additional geometrical supervision of ‘Depth’ methods, and this advocates that our pipeline can efficiently learn geometrical cues from images sequence. An interesting remark is that the IoU gap between ‘in-FOV’ and ‘Whole scene’ is larger for SceneRF (15.18 vs 13.84) compared to ‘Depth’ baselines (<1). This seems to indicate that SceneRF is having harder time at hallucinating reconstruction besides input FOV. Interestingly, the low numbers for all methods advocate for the task complexity, showing room for future research.

Fig. 5 also shows some qualitative reconstruction. Overall, SceneRF produces better reconstruction results with less artefacts, especially on vegetation and sidewalk in the 1st example, house in the 2nd, and general scene structure in the 3rd.

4.4. Ablation studies

We now ablate SceneRF, first by removing each of our components and then by studying in details our Probability sampling, Spherical decoder and Scene reconstruction.

Architectural components. Tab. 3 reports novel depth/view synthesis of SceneRF when removing the *rgb*

Method	Novel depth synthesis					Novel view synthesis				
	Abs Rel \downarrow	Sq Rel \downarrow	RMSE \downarrow	RMSE log \downarrow	$\delta 1 \uparrow$	$\delta 2 \uparrow$	$\delta 3 \uparrow$	LPIPS \downarrow	SSIM \uparrow	PSNR \uparrow
SceneRF	0.1717	1.309	5.696	0.2809	75.01	89.35	94.76	0.490	0.475	16.29
w/o \mathcal{L}_{rgb}	0.1911	1.639	6.826	0.3730	69.76	85.99	92.78	-	-	-
w/o $\mathcal{L}_{\text{reproj}}$	0.1926	1.471	5.890	0.2949	71.82	88.64	94.49	0.492	0.477	16.29
w/o SU-Net	0.1766	1.379	5.897	0.2943	73.78	88.26	94.08	0.478	0.474	16.36
w/o PrSamp	0.1845	1.318	5.763	0.2880	71.60	89.25	94.71	0.513	0.461	16.43

Table 3. Architecture ablation for novel depth/view synthesis. We ablate losses: the standard L1 reconstruction loss \mathcal{L}_{rgb} and our reprojection loss ($\mathcal{L}_{\text{reproj}}$). For architectural components, in ‘w/o SU-Net’ we replace our spherical U-Net (Sec. 3.3) by a standard U-Net of similar capacity, while in ‘w/o PrSamp’ we replace our probabilistic sampling (Sec. 3.2) with a standard hierarchical sampling as in [36, 75] with the same number of inferences. All components contribute to the significantly better results for our primary task of novel depth synthesis, while little degrading novel view synthesis which still compete with the best methods.

k	m	Abs Rel \downarrow	Sq Rel \downarrow	RMSE \downarrow	RMSE log \downarrow	$\delta 1 \uparrow$	$\delta 2 \uparrow$	$\delta 3 \uparrow$
1	32	0.1850	1.358	5.956	0.2940	71.38	88.73	94.51
2	16	0.1788	1.327	5.889	0.2878	72.68	88.90	94.70
4	4	0.1845	1.371	5.878	0.2940	71.62	88.59	94.51
	8	0.1717	1.309	5.696	0.2809	75.01	89.35	94.76
	16	0.1664	1.319	5.980	0.2894	74.58	88.48	94.17
8	2	0.1832	1.333	5.863	0.2934	71.60	88.61	94.50
	4	0.1768	1.311	5.824	0.2910	72.86	88.60	94.42
	8	0.1697	1.311	5.794	0.2873	74.59	88.71	94.34

Table 4. Ablation of the PrSamp (Sec. 3.2) with varying the number of Gaussians (k) and point sampled per Gaussian (m).

loss (\mathcal{L}_{rgb}), reprojection loss ($\mathcal{L}_{\text{reproj}}$, Eq. (4)), spherical U-Net (SU-Net, Sec. 3.3), or Probabilistic Sampling (PrSamp, Sec. 3.2). Without SU-Net, our spherical decoder is replaced with a standard decoder while setting $\psi(\cdot)$ to simple cartesian projection. Without PrSamp, we fallback to standard hierarchical inferences [42, 75] using the same number of inferences for fair comparison.

In a nutshell, our components all contribute to the best novel depth synthesis metrics. In particular, both $\mathcal{L}_{\text{reproj}}$ and PrSamp improve significantly the absolute relative error and the $\delta 1$ showing a beneficial effect on close range depth estimation. For the subsidiary task of novel view synthesis, our components have mixed effects showing that depth improvements comes at slightly lower reconstruction capacity.

Probabilistic Ray Sampling (Sec. 3.2). A simple assumption is that sampling more Gaussians, or more points, would better approximate the underlying density volume and thus yield better results. This is proven wrong in Tab. 4 where we vary the number of Gaussians (k) and points sampled per Gaussian (m). The best results are in fact obtained with $k = 4$ and $m = 8$. We conjecture this relates to the radiance field not being able to optimize too many surfaces per ray. To preserve computation cost also, more Gaussians implies less points per Gaussians which introduces noise.

Spherical U-Net (Sec. 3.3). Tab. 3 highlights the benefit of our SU-Net (cf. ‘w/o SU-Net’). We com-

Method	Sampling	in-FOV			Whole Scene			
		step	rot.	IoU	Prec.	Rec.	IoU	Prec.
AdaBins [4] (<i>supervised</i>)				18.15	26.00	35.38	15.37	27.33
Monodepth2* [19]				12.41	18.32	27.82	10.76	18.28
SceneRF (ours)				11.84	16.32	30.17	11.80	19.91
	0.25	-10 / 0 / +10		15.03	17.36	52.83	13.73	16.98
	0.5	-10 / 0 / +10		15.18	17.67	51.87	13.84	17.28
	1.0	0		14.69	18.58	41.20	13.08	18.56
	1.0	-10 / 0 / +10		14.80	17.68	47.65	13.40	17.27
	1.0	-20 / 0 / +20		14.64	17.30	48.84	13.37	16.73
	1.0	-30 / 0 / +30		14.38	17.17	46.99	13.24	16.40
	2.0	-10 / 0 / +10		14.79	17.84	46.40	13.35	17.41

* We train Monodepth2 with GT poses for fair comparison with our setting

Table 5. Scene reconstruction using either a single depth input (top 3 rows), or using our novel depth views (bottom) with varying steps (ρ) and angles (Φ) in our reconstruction scheme Sec. 3.4. Refer to text for details.

plement this study, by studying planar (*i.e.*, standard decoder) and spherical decoder of different horizontal FOV. We experiment with planar-80°, planar-120°, spherical-80°, spherical-120°, getting respectively 17.66/17.25/17.67/**17.17** for Abs Rel metric (lower is better) and 73.78/74.23/73.46/**75.01** for $\delta 1$ (higher is better). Larger FOV seems to always improve, but our spherical decoder reaches best results – presumably because it induces less projection distortion.

Scene reconstruction (Sec. 3.4). We study variations of our scene reconstruction scheme in Tab. 5. In the top 3 rows, we first evaluate reconstruction using *a single depth map at the input frame* with the best monocular depth estimation methods being: AdaBins [4], Monodepth2 [19], and Ours. As expected here, AdaBins – the only supervised method – outperforms all. However, the bottom part of Tab. 2 shows that when SceneRF is complemented with our unique synthesis of novel depths we reach the best performance among self-supervised methods with a 3 points margin in IoU compared to Monodepth2 [19]. More in-depth, referring to Sec. 3.4 we vary depth steps (ρ) and rotations (Φ). Looking at in-FOV performance, novel depths always improve by at least 4% IoU, showing the benefit of

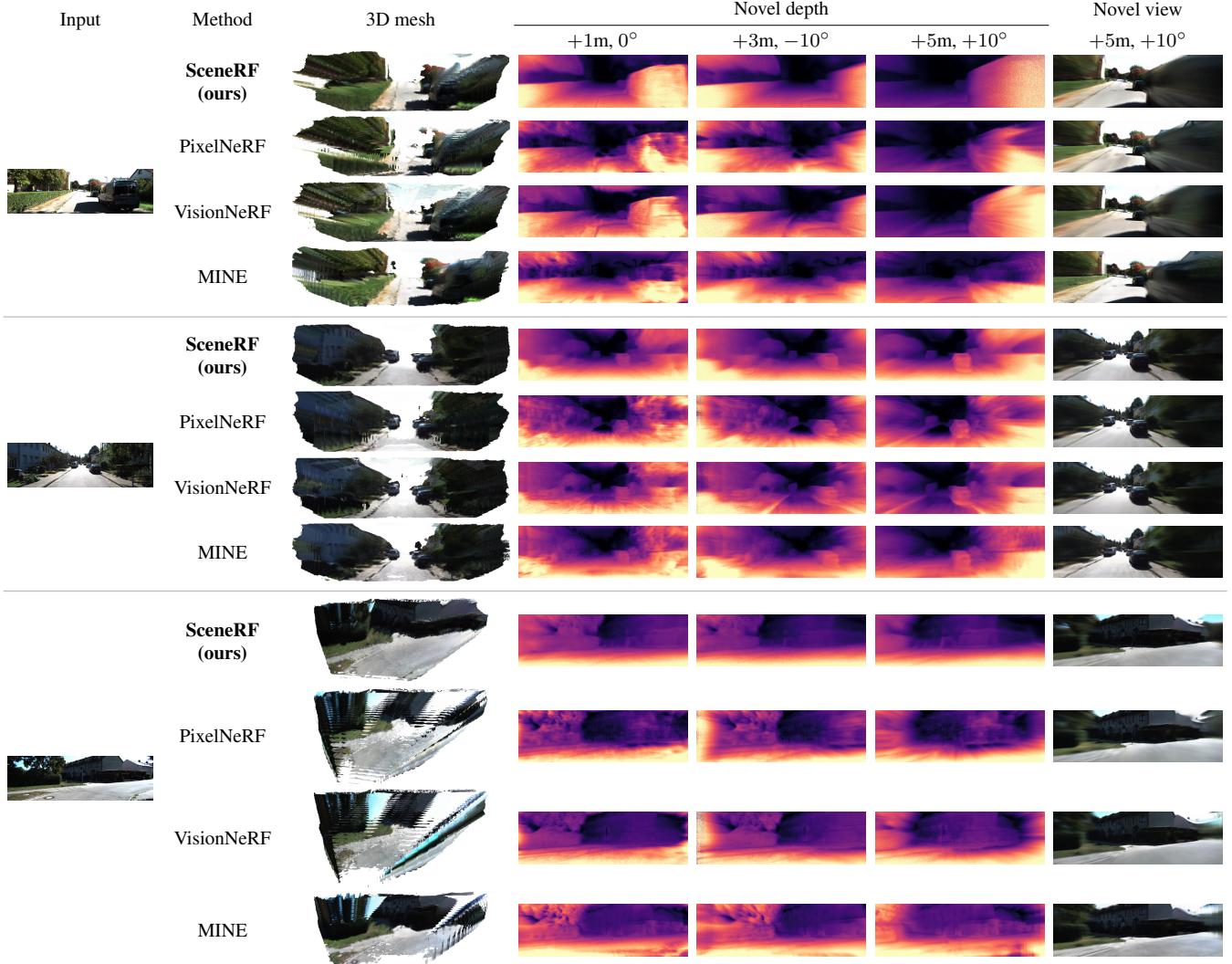


Figure 5. Qualitative results on SemanticKITTI [3] (val set). For each input frame, we report novel depth and view synthesis at varying positions and viewing angles w.r.t. the input frame. In particular we note our depths is sharper and better at-far distances. This advocates for our design choices which comes at little, if any, qualitative degradation of novel views. The 3D mesh shows reconstruction with our scheme for all methods (Sec. 3.4) colorized with volume rendering Eq. (2). Our reconstruction appears evidently better than others, which is evident when drastically varying the viewpoint (bottom row). Please refer to video in supplementary for better qualitative judgement.

our reconstruction scheme. In general, we also note that synthesizing more depths (*i.e.*, smaller ρ steps) with varying angles (*i.e.*, $\Phi \notin 0$) boost all IoU, though large angle variations degrade slightly the results. We conjecture this results from our autonomous driving setup where camera is always front-facing, thus providing little peripheral supervision.

5. Conclusion

Self-supervised monocular scene reconstruction is yet in its early steps, though our proposed method is already able to reconstruct complex 3D scene leveraging image-

conditioned NeRF. Our custom designs for explicit depth, our novel ray sampling and our spherical decoder, altogether enable novel applications to scene reconstruction. We highlight the complexity of our setup w.r.t. to prior works, as we handle complex cluttered scenes in challenging environments.

Limitations. Despite good results, we note that SceneRF suffers from two main limitations. First, the time-consuming depth/view synthesis due to per-point inference which could be solved with ray inference [61]. Second, SceneRF is yet poorly resistant to rotation, which we at-

tribute to our training set being mostly front-facing. A data-oriented solution, is to use other datasets [5] with more diverse views. Finally, our works open an interesting avenue for novel research to scene reconstruction directly from volume density.

Broader impact, Ethics. The promotion of self-supervised monocular 3D reconstruction contributes to alleviating the needs of costly data acquisition and labeling campaigns. On the long term, this also paves the way to 3D algorithms training directly on video sequences – easier to collect and significantly more diverse than existing 3D datasets. A by-product is that it would contribute to improving generalization of 3D reconstruction. While there are no ethical concerns specific to our proposed method, we note that all methods estimating 3D from 2D are far less precise than those leveraging depth sensors (*e.g.*, lidar, depth cameras, stereo, etc.). When it comes to safety-critical applications, like autonomous driving, we argue for use of redundant sensors.

Acknowledgment. The work was partly funded by French project SIGHT (ANR-20-CE23-0016) and was performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011012808 and 2022-AD011012808R1). We thank Fabio Pizzati and Ivan Lopes for their kind proofreading.

Appendices

We provide additional implementation details of SceneRF in Appendix A, the effect of $\mathcal{L}_{\text{reproj}}$ on baselines in Appendix B, baselines implementation details in Appendix C, and additional qualitative results in Appendix D.

The supplementary video allows better evaluation of our method is available at: <https://youtu.be/tRah87F4Gdk>.

A. Additional implementation details

A.1. Probabilistic ray sampling (PrSamp) details

For clarity, in Algorithm 1, we detail the pseudocode of the Probabilistic Ray Sampling (Sec. 3.2).

A.2. 3D reconstruction details

Fusing TSDFs. From Sec. 3.4, we fuse individual TSDFs by taking the minimum of their absolute values ('TSDF min') instead of the more standard average of all TSDFs ('TSDF avg'). We justify this choice, in Tab. 6 showing that using 'TSDF min' leads to +2.77 IoU for the whole scene. We argue to the varying viewpoints of our depth synthesis, which induce that some surfaces are better estimated by specific depth viewpoints. Averaging all ('TSDF avg') has a smoothing effect on $V(\cdot)$ which subsequently reduces accuracy.

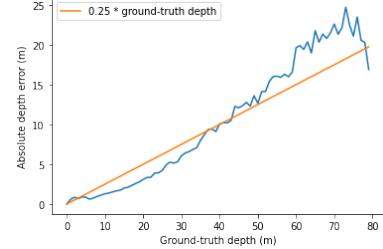


Figure 6. **Absolute depth error w.r.t. ground-truth depth.** We compute error from 100 randomly selected scenes in the training set, and observe a linear relation between error and distance.

Occupancy grid. To convert the scene TSDF volume $V(\cdot)$ (cf. Sec. 3.4) into an occupancy grid, we first study the depth estimation error. Comparing 100 frames in Fig. 6 with sparse Lidar ground truth, we note a linear relation between error estimation and ground truth depth. This motivated us to model the occupancy grid $O(\cdot)$ as an adaptive depth threshold:

$$O(v) = 1 \iff V(v) < \min(0.25d_v, 4.0), \quad (8)$$

with v a voxel in V , and d_v its distance to the camera origin. We arbitrarily cap the threshold to 4 meters to avoid considering all far voxels as occupied.

A.3. Network architecture

For 2D features extraction, the encoder is similar to [7], which is based on a pre-trained EfficientNetB7 [65]. The spherical decoder has 5 layers, each of which doubles the input resolution and halves the feature dimension. To make up for the large amount of empty space that comes with increasing the field of view, we augment the receptive field by putting three ResNet blocks with dilation sizes of 1, 2, and 3 in each layer. The skip connections (described in Sec. 3.3) are used between the encoder and decoder at the corresponding scale.

B. Effect of $\mathcal{L}_{\text{reproj}}$ on baselines performance

In Tab. 7, we apply the reprojection loss $\mathcal{L}_{\text{reproj}}$ (Sec. 3.1.1) to all baselines, showing that it improves significantly all baselines performance.

C. Baselines details

We re-train all baseline networks, including the novel depth/view synthesis (Appendix C.1) and scene reconstruction baselines (Appendix C.2). We provide the reader with additional details about our baselines.

C.1. Novel depth/views baselines

We train SceneRF and baselines using AdamW [39] optimizer on 4 Tesla V100 32g with learning rate of 1e-5 for

Algorithm 1: Probabilistic Ray Sampling.

Input : Ray \mathbf{r} .

Param: Number of Gaussians k , and m number of points per Gaussian.
 Near and far bounds: $t_n = 0.2\text{m}$ and $t_f = 100\text{m}$.
 Learning rate lr of gradient descend (GD).

Result: Points sampled \mathcal{P}

```

1  $\mathbf{d} \leftarrow \text{dir}(\mathbf{r})$ 
   // Uniform sampling (●)
2  $\mathcal{I} \leftarrow \{\text{uniform-samp}(\text{num}=k, \text{start}=t_n, \text{end}=t_f) \times \mathbf{d}\}$ 
   // (1) Predicts Gaussians ( $\mathcal{G}$ ) with MLP  $g(\cdot)$ 
3  $\mathcal{G} \leftarrow g(\{(\mathbf{x}, \mathbf{W}(\psi(\mathbf{x}))) \mid \forall \mathbf{x} \in \mathcal{I}\})$ 
   // (2) Sample  $m$  points from Gaussians (■)
4  $\mathcal{P} \leftarrow \emptyset$ 
5 for  $i \leftarrow 1$  to  $k$  do
6   |  $\mathcal{P} \leftarrow \mathcal{P} \cup \text{gauss-sampling}(\mathcal{G}_i, m)$ 
7 end
   // Sample 32 points uniformly (▲)
8  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\text{uniform-samp}(\text{num}=32, \text{start}=t_n, \text{end}=t_f) \times \mathbf{d}\}$ 
   // (3) - (4) NeRF inference to compute densities
9  $\sigma \leftarrow \{f(\gamma(\mathbf{x}), \mathbf{d}; \mathbf{W}(\psi(\mathbf{x})))_\sigma \mid \forall \mathbf{x} \in \mathcal{P}\}$ 
   // (5) PrSOM point-Gaussian assignment
10  $\alpha \leftarrow \{\text{alpha-value}(s, \dots) \mid \forall s \in \sigma\}$ 
11  $\mathcal{X} \leftarrow \{\text{PrSOM}(\mathcal{G}, \mathcal{P}, \alpha)\}$ 
   // (6) Compute new Gaussians from assigned points and update  $g(\cdot)$ 
12  $\mathcal{G}' \leftarrow \{(\mu(\mathcal{X}_i), \text{std}(\mathcal{X}_i)) \mid \forall i \in N, 1 \leq i \leq k\}$ 
13  $\mathcal{L}_{\text{gauss}} \leftarrow \frac{1}{k} \sum_i^k \text{Kullback-Leibler}(\mathcal{G}_i \parallel \mathcal{G}'_i)$ 
14  $\mathcal{L}_{\text{surface}} \leftarrow \min_i (\|\mu(\mathcal{G}'_i) - \hat{D}(\mathbf{r})\|_1)$ 
15  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{gauss}} + \mathcal{L}_{\text{surface}}$ 
16  $g \leftarrow \text{GD}^{lr}(g, \nabla \mathcal{L}_{\text{total}})$ 

```

Method	in-FOV			Whole Scene		
	IoU	Prec.	Rec.	IoU	Prec.	Rec.
SceneRF (TSDF avg)	12.38	13.67	56.75	11.07	11.81	63.98
SceneRF (TSDF min)	15.18	17.67	51.87	13.84	17.28	40.96

Table 6. **TSDF fusion strategy comparison.** We show that our way of extracting the TSDF described in section 3.4 is better than the traditional way of using the weighted average of TSDFs.

50 epochs. For each baseline, we rely on the recommended learning rate scheduler and number of positional encoding frequencies. For our network, since we build on PixelNeRF [75], we use its scheduler and number of frequencies. The ray batch size was 1200 and the training time was around 5 days per network. Additional information about the baselines implementations is provided below.

PixelNeRF [75]. We use the official implementation². Following the official sampling strategy, we sample 96 points per ray, consisting of 64 coarse points, which are used to sample 16 fine points hierarchically and 16 points around the estimated depth.

²<https://github.com/sxyu/pixel-nerf>

MINE [32]. We use the official implementation³. To balance memory cost, we use the 32 planes version.

VisionNeRF [36]. We use the official implementation⁴. To balance memory cost again, we sample 96 points (32 coarse, 64 fine) which is more than for ours.

C.2. Scene reconstruction baselines

Only MonoScene⁵ is a monocular baseline. To better compare with the literature, we follow the recommendation of MonoScene authors [7] and compare against the ^{rgb} versions of popular semantic scene completion baselines: LMSCNet⁶ [57], 3DSketch⁷ [8] and AICNet⁸ [33]. More in depth, to convert the sequence of depths into 3D label to train the scene reconstruction baselines, we use the Adabin [4] model to predict the depth for each image and fuse all depths into a single TSDF volume, then turned into an occupancy grid with the same reconstruction scheme as for ours (see Appendix A.2). For all, the mesh is obtained with the traditional marching cubes [38].

³<https://github.com/vincentfung13/MINE>

⁴<https://github.com/ken2576/vision-nerf>

⁵<https://github.com/cv-rits/MonoScene>

⁶<https://github.com/cv-rits/LMSCNet>

⁷<https://github.com/charlesCXK/TorchSSC>

⁸<https://github.com/waterljwant/SSC>

Method	$\mathcal{L}_{\text{reproj}}$	Novel depth synthesis						Novel view synthesis			
		Abs Rel \downarrow	Sq Rel \downarrow	RMSE \downarrow	RMSE log \downarrow	$\delta1\uparrow$	$\delta2\uparrow$	$\delta3\uparrow$	LPIPS \downarrow	SSIM \uparrow	PSNR \uparrow
PixelNeRF [75]	✓	0.2364 0.1986	2.080 1.544	6.449 5.963	0.3354 0.3093	65.81 70.30	85.43 87.19	92.90 93.82	0.489 0.488	0.466 0.481	15.80 16.11
MINE [32]	✓	0.2248 0.2003	1.787 1.599	6.343 6.023	0.3283 0.3070	65.87 70.22	85.52 86.98	93.30 93.89	0.448 0.445	0.496 0.497	16.03 15.96
VisionNerf [36]	✓	0.2054 0.1749	1.490 1.380	5.841 5.643	0.3073 0.2841	69.11 75.77	88.28 89.25	94.37 94.58	0.468 0.432	0.483 0.488	16.49 16.39

Table 7. The reprojection loss $\mathcal{L}_{\text{reproj}}$ improves the performance of all baselines.

D. Additional results

We show additional qualitative results in Fig. 7 and Fig. 8. Overall, SceneRF predicts smoother and finer depth maps, especially at far, which leads to a better-structured 3D scene with fewer artifacts than the baselines. When synthesizing RGB images, our approach achieves comparable results to other baseline methods.

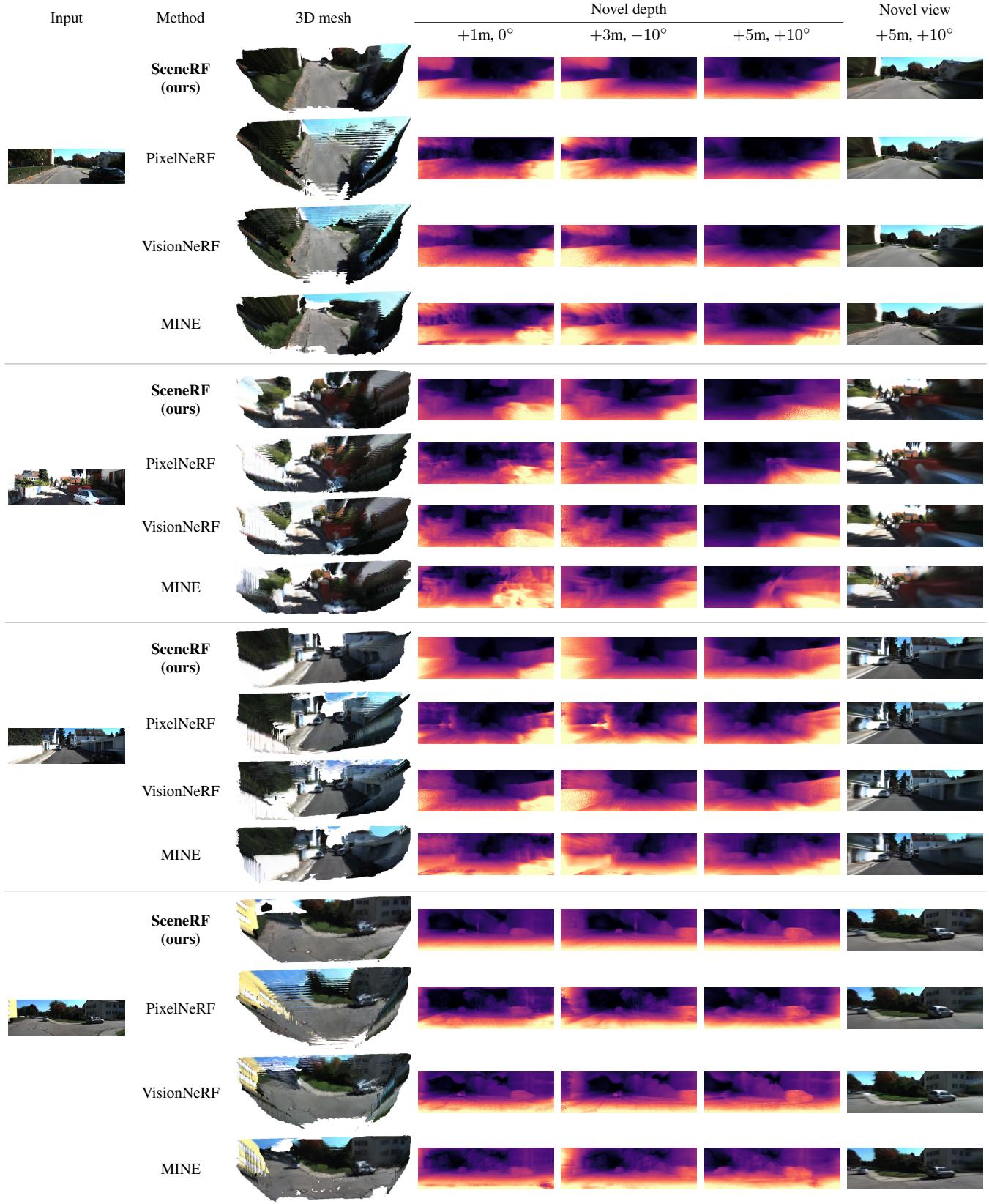


Figure 7. Additional qualitative results on SemanticKITTI [3] (val set). Please refer to the supplementary video for more results.

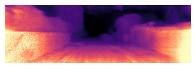
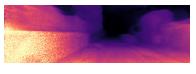
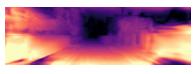
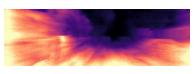
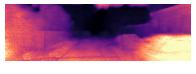
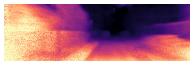
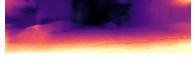
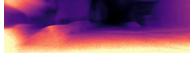
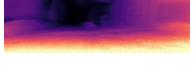
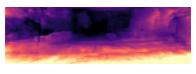
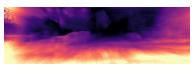
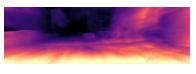
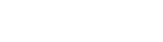
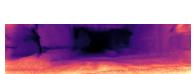
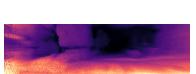
Input	Method	3D mesh	Novel depth			Novel view +5m, +10°
			+1m, 0°	+3m, -10°	+5m, +10°	
	SceneRF (ours)					
	PixelNeRF					
	VisionNeRF					
	MINE					
	SceneRF (ours)					
	PixelNeRF					
	VisionNeRF					
	MINE					
	SceneRF (ours)					
	PixelNeRF					
	VisionNeRF					
	MINE					

Figure 8. Additional qualitative results on SemanticKITTI [3] (val set). Please refer to the supplementary video for more results.

References

- [1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, 2018. 2
- [2] Fatiha Anouar, Fouad Badran, and Sylvie Thiria. Probabilistic self-organizing map and radial basis function networks. *Neurocomputing*, 1998. 4, 10
- [3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019. 1, 5, 6, 8, 12, 13
- [4] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *CVPR*, 2021. 5, 6, 7, 10
- [5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giacomo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1, 9
- [6] Ang Cao, C. Rockwell, and Justin Johnson. Fwd: Real-time novel view synthesis with forward warping and depth. In *CVPR*, 2022. 2
- [7] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 1, 2, 5, 6, 9, 10
- [8] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*, 2020. 5, 6, 10
- [9] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *CVPR*, 2020. 2
- [10] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996. 5
- [11] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. In *NeurIPS*, 2021. 1, 2
- [12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. 1, 2
- [13] Shivam Duggal and Deepak Pathak. Topologically-aware deformation fields for single-view 3d reconstruction. In *CVPR*, 2022. 2
- [14] Sayna Ebrahimi, Angjoo Kanazawa, and Trevor Darrell. Differentiable gradient sampling for learning implicit 3d scene reconstructions from a single image. In *ICLR*, 2022. 2
- [15] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 2
- [16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5
- [17] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *ICCV*, 2019. 2
- [18] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 3
- [19] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 3, 5, 7
- [20] Alexander Grabner, Peter M Roth, and Vincent Lepetit. Location field descriptors: Single image 3d model retrieval in the wild. In *3DV*, 2019. 2
- [21] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. Drwr: A differentiable renderer without rendering for unsupervised 3d structure learning from silhouette images. In *ICML*, 2020. 2
- [22] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *SIGGRAPH*, 1997. 2
- [23] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf efficient neural radiance fields. In *CVPR*, 2022. 4
- [24] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *ECCV*, 2018. 1, 2
- [25] Zixuan Huang, Stefan Stojanov, Anh Thai, Varun Jampani, and James M. Rehg. Planes vs. chairs: Category-guided 3d shape learning without any 3d cues. In *ECCV*, 2022. 2
- [26] Hamid Izadinia, Qi Shan, and Steven M. Seitz. Im2cad. In *CVPR*, 2017. 2
- [27] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *ICCV*, 2021. 2
- [28] Jan J Koenderink, Andrea J van Doorn, and Astrid ML Kappers. Depth relief. *Perception*, 1995. 1
- [29] Abhijit Kundu, Yin Li, and James M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 2
- [30] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhofer, and Markus Steinberger. Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. In *ECCV*, 2022. 2
- [31] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *ICCV*, 2017. 2
- [32] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, 2021. 1, 2, 5, 6, 10, 11
- [33] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *CVPR*, 2020. 5, 6, 10
- [34] Xingyi Li, Chaoyi Hong, Yiran Wang, Zhiguo Cao, Ke Xian, and Guosheng Lin. Symmnerf: Learning to explore symmetry prior for single-view view synthesis. In *ACCV*, 2022. 2
- [35] Yiyi Liao, Simon Donné, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *CVPR*, 2018. 2
- [36] Kai-En Lin, Yen-Chen Lin, Wei-Sheng Lai, Tsung-Yi Lin, Yichang Shih, and Ravi Ramamoorthi. Vision transformer for nerf-based view synthesis from a single input image. In *WACV*, 2023. 2, 5, 6, 7, 10, 11

- [37] Feng Liu and Xiaoming Liu. 2d gans meet unsupervised single-view 3d reconstruction. In *ECCV*, 2022. 1
- [38] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, 1987. 10
- [39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 9
- [40] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2
- [41] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *TOG*, 2019. 10
- [42] 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, 7
- [43] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 2021. 2
- [44] Norman Müller, Andrea Simonelli, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Autorf: Learning 3d object radiance fields from single view observations. In *CVPR*, 2022. 1, 2
- [45] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONERF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *CGF*, 2021. 2, 4
- [46] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011. 5
- [47] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 2
- [48] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 1, 2
- [49] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D ken burns effect from a single image. *TOG*, 2019. 2
- [50] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2
- [51] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 2
- [52] S. Popov, Pablo Bauszat, and V. Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In *ECCV*, 2020. 2
- [53] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021. 2
- [54] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Tom Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. 1, 2, 5
- [55] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. Sharf: Shape-conditioned radiance fields from a single view. In *ICML*, 2021. 2
- [56] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, 2022. 1, 2
- [57] Luis Roldão, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *3DV*, 2020. 5, 6, 10
- [58] Luis Roldao, Raoul De Charette, and Anne Verroust-Blondet. 3d semantic scene completion: a survey. *IJCV*, 2022. 2
- [59] David Salomon. *Transformations and projections in computer graphics*. Springer, 2006. 4
- [60] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Ambrus, Adrien Gaidon, William T Freeman, Fredo Durand, Joshua B Tenenbaum, and Vincent Sitzmann. Seeing 3d objects in a single image via self-supervised static-dynamic disentanglement. *arXiv preprint arXiv:2207.11232*, 2022. 1, 2
- [61] Vincent Sitzmann, Semon Rezhikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 8
- [62] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2
- [63] Peter Sturm and Steve Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *BMVC*, 1999. 1
- [64] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *CVPR*, 2019. 2
- [65] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 9
- [66] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Single-view to multi-view: Reconstructing unseen views with a convolutional network. *CoRR*, 2015. 2
- [67] Alex Trevithick and Bo Yang. GRF: Learning a general radiance field for 3D scene representation and rendering. In *ICCV*, 2021. 2
- [68] Frank A Van den Heuvel. 3d reconstruction from a single image using geometric constraints. *ISPRS*, 1998. 1
- [69] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 2

- [70] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 5
- [71] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *IJCV*, 2020. 2
- [72] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *EUROGRAPHICS*, 2022. 1, 2
- [73] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019. 2
- [74] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *ICCV*, 2019. 2
- [75] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7, 10, 11
- [76] Sergey Zakharov, Rares Ambrus, Vitor Campagnolo Guizilini, Dennis Park, Wadim Kehl, Frédéric Durand, Joshua B. Tenenbaum, Vincent Sitzmann, Jiajun Wu, and Adrien Gaidon. Single-shot scene reconstruction. In *CoRL*, 2021. 2
- [77] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 5, 6
- [78] M. Zerroug and R. Nevatia. Part-based 3d descriptions of complex objects from a single image. *TPAMI*, 1999. 1
- [79] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *CVPR*, 2021. 1, 2
- [80] Jason Y Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020. 2
- [81] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5
- [82] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *CVPR*, 2018. 2
- [83] Nikola Zubić and Pietro Lio'. An effective loss function for generating 3d models from single 2d image without rendering. In *AIAI*, 2021. 2