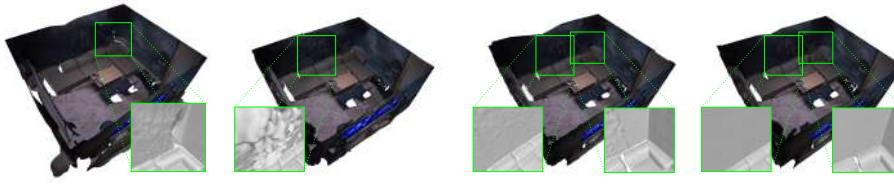
VF-NeRF: Learning Neural Vector Fields for Indoor Scene Reconstruction

Albert Gassol Puigjaner*[1], Edoardo Mello Rella*[1], Erik Sandström[1], Ajad Chhatkuli[1], and Luc Van Gool[1,2,3]

[1] Computer Vision Lab, ETH Zurich
[2] VISICS, KU Leuven
[3] INSAIT, Sofia



MonoSDF [58] F-score: 0.790 | ManhattanSDF [16] F-score: 0.757 | VF-NeRF (Ours) F-score: **0.894** | Ground Truth

Fig. 1: VF-NeRF . Using the recently proposed Vector Field (VF) [41] representation, our method reconstructs indoor scenes in the NeRF setting. Due to the planar inductive bias of VF, we can generally recover indoor scenes with high fidelity, providing State-of-the-Art (SOTA) performance.

Abstract. Implicit surfaces via neural radiance fields (NeRF) have shown surprising accuracy in surface reconstruction. Despite their success in reconstructing richly textured surfaces, existing methods struggle with planar regions with weak textures, which account for the majority of the indoor scenes. In this paper, we address indoor dense surface reconstruction by revisiting key aspects of NeRF in order to use the recently proposed Vector Field (VF) as the implicit representation. VF is defined by the unit vector directed to the nearest surface point. It therefore flips direction at the surface, and equals to the explicit surface normals. Except for this flip, VF remains constant along planar surfaces and provides a strong inductive bias in representing planar surface. Concretely, we develop a novel density-VF relationship and a training scheme that allows us to learn VF via volume rendering. By doing this, VF-NeRF can model large planar surfaces and sharp corners accurately. Additionally, we show that, when depth cues are available, our method further improves and achieves state-of-the-art results in reconstructing indoor scenes and rendering novel views. We extensively evaluate VF-NeRF on public datasets such as Replica and ScanNet and run comprehensive ablations of its components.

1 Introduction

Multi-view image-based 3D scene reconstruction is a cornerstone challenge in computer vision [17,43,46]. Traditional multi-view stereo (MVS) algorithms [11,

42,43,49,59] leverage matching and triangulation to derive 3D point coordinates from given input images. Nonetheless, they often struggle in regions characterized by uniform low-texture or repetitive patterns. Equipped with volume rendering, Neural Radiance Fields (NeRF) [34,51,56] and its variants [26,30,33] have established themselves as powerful alternatives to previous methods for surface reconstruction. However, NeRF methods still struggle with low-texture indoor surfaces, even when using Manhattan normal priors [8,16].

NeRF for indoor scene reconstruction has currently two significant challenges. The first is that the classical NeRF surface density [34], which provides high-quality view rendering, stumbles significantly when it comes to scene geometry reconstruction. Even when an SDF [51,56] representation is used for the geometry, any surface regularization for planar surfaces has to rely on the gradients of the SDF [16]. Note that these gradients are often noisy and unreliable for regularization. An additional downside of SDF is that its representation power is limited to water-tight surfaces. Therefore, it may not be able to faithfully reconstruct thin or open surfaces. The second challenge stems from poor texture in indoor surfaces, which provides weak multi-view constraints for the indirect triangulation in NeRF or direct triangulation in MVS approaches.

In this paper, we address the first challenge, that of the implicit scene representation in NeRF. In the process, we also push towards mitigating the challenge of weak texture through an improved inductive bias towards planar surfaces. To that end, we make use of the recently proposed Vector Field (VF) representation [41,53] in order to encode the scene geometry. This involves associating each position in the 3D space with a unit vector directed towards the nearest surface. It has been shown that VF may exhibit superior performance to SDF even on closed surfaces, particularly on sharp corners, thin objects and planar surfaces. This is due to the properties of VF and its inductive bias towards planar surfaces as they exhibit a constant normal along flat surfaces. However, the study confines itself to a supervised learning paradigm, and the self-supervised learning with NeRF poses significant challenges. Notably, without the ground-truth VF, a pair of points is required to compute the surface density given the VF predictions.

In the VF optimization, we use a dual MLP network, one to predict the VF and the other to predict the RGB color values. We learn the VF and the color through a training scheme via volume rendering on multi-view posed images similarly to VolSDF [51,56]. Specifically, we express the surface density via the cosine similarity of the VF predictions in the ray samples, which are obtained in a hierarchical manner. This novel VF-density relationship allows us to use neural volume rendering in order to train the VF as in [51,56]. As a first study on VF for NeRF, we consider its use for learning indoor scene geometry. We rigorously evaluate our method against leading benchmarks for indoor scenes, including ManhattanSDF [16], MonoSDF [58], and Neuralangelo [25], on indoor datasets such as Replica [47] and ScanNet [9], showing superior performance on both reconstruction and novel view rendering.

In summary, our contributions are threefold:

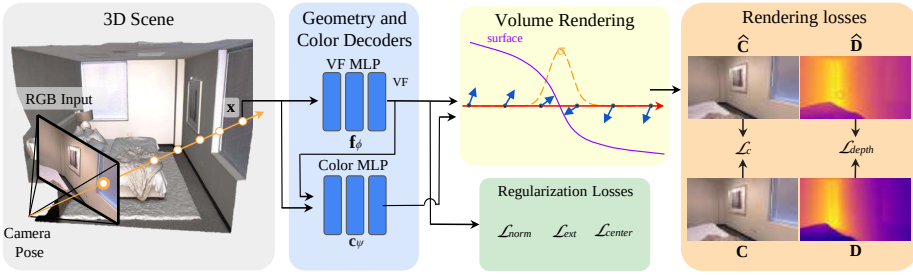


Fig. 2: VF-NeRF overview. We use VF to represent the geometry of a scene. Specifically, given an input image taken from the camera view position, we shoot a batch of rays onto the 3D scene. We predict the VF and color of the points along the ray using geometry and color decoders, two sets of MLPs. By computing the cosine similarity between neighboring points on the ray, we can identify the surface as the locations where the value equals -1; this happens when the two predicted vectors have opposing directions. From the cosine similarity, we then differentiably compute the surface density required for volume rendering. We render the RGB and depth in order to compute the re-rendering losses. We then optimize them together with the regularization terms for the network parameters.

– We propose to learn the VF representation of 3D scenes with multi-view images via volume rendering.
– We develop an efficient hierarchical ray sampling approach, which allows us to sample more densely near surfaces.
– We demonstrate the effectiveness of our method on different indoor scene datasets, showing state-of-the-art results.

2 Related work

Multi-view Surface Reconstruction. Traditional MVS approaches have often relied on feature matching for depth estimation [2–4, 12, 24, 42–44]. These classical methods extract image features, match them across views for depth estimation, and then fuse the obtained depth maps to form dense point clouds. Voxel-based representations [1, 11, 45] rely on color consistency between the projected images to generate an occupancy grid of voxels. Subsequently, meshing techniques, like Poisson surface reconstruction [21, 22] are applied to delineate the surface. However, these methods typically fail to reconstruct low-textured regions and non-Lambertian surfaces. Additionally, the reconstructed point clouds or meshes are often noisy and may fail to reconstruct some surfaces.

Recently, learning-based methods have gained attention, offering replacements for classic MVS methods. Methods like [5, 18, 54, 55] leverage 3D CNNs to extract features and predict depth maps, while others [6, 15] construct cost volumes hierarchically, yielding high-resolution results. However, these methods

often fail to accurately reconstruct the scene geometry due to the limited resolution of the cost volume.

Neural Radiance Fields (NeRF). In recent studies [30,33,34,40] the potential of MLPs to represent scenes both in terms of density and appearance has been explored. While these techniques can produce photorealistic results for novel view synthesis, determining an isosurface for the volume density to reconstruct scene geometry remains a challenge. Commonly, NeRF uses thresholding techniques to derive surfaces from the predicted density. However, these extracted surfaces can often exhibit noise and inaccuracies.

Neural Scene Representations. Approaches based on neural scene representations employ deep learning to learn properties of 3D points and to generate geometry. Traditional methods like point clouds [10, 28] and voxel grids [7, 52] have been primary choices for representing scene geometry. More recently, implicit functions, such as occupancy grids [36, 37], SDF [19, 25, 31, 38, 51, 56, 57], and VF [41, 53] have gained popularity due to their precision in capturing scene geometry. For instance, in [31, 36] a novel differentiable renderer to learn the scene geometry from images is proposed, while [57] focuses on modeling view-dependent appearance, which proves successful on non-Lambertian surfaces. [29], instead, utilizes 2D silhouettes from single images to reconstruct their underlying 3D shape. However, these methods rely on masks to accurately reconstruct the geometry from multi-view images. Consequent works, such as VolSDF [56] and NeuS [51] introduce a second MLP in the NeRF context to represent the geometry as the SDF, further leveraging volume rendering to learn the geometry from images. Building upon these methods, Neuralangelo [25] takes inspiration from Instant Neural Graphics Primitives (Instant NGP) [35] to introduce hash encodings in neural SDF models, enhancing surface reconstruction resolution. However, a challenge persists as these methods tend to fail in large indoor planar scenes with low-texture regions, leading to inaccurate surface reconstructions.

Priors for Neural Scene Representations. Several works have explored the integration of priors during optimization to improve the reconstruction of indoor scenes. For instance, Manhattan-SDF [16] suggests incorporating dense depth maps from COLMAP [43] to facilitate the learning of 3D geometry and employs Manhattan world [8] priors to address the challenges posed by low-textured planar surfaces. A limitation of this approach is its reliance on semantic segmentation masks to pinpoint planar regions, adhering to the Manhattan world assumption. This dependency can lead to added complexity and potential inaccuracies in regions where segmentations are less accurate. More recently, NeuRIS [50] proposes to use normal priors to guide the reconstruction of indoor scenes. Expanding on this work, MonoSDF [58] introduces both normal and depth monocular cues into the optimization.

3 Method

Given a set of posed images of an indoor scene, our goal is to reconstruct the dense scene geometry. We represent the surface geometry in NeRF with VF [41,

53], and describe its properties in Sec. 3.1. We then introduce the surface density as a parametrization of the VF in Sec. 3.2 and describe our hierarchical ray sampling method in Sec. 3.3. Finally, in Sec. 3.4, we formulate the optimization problem and introduce the loss terms. We provide an overview in Fig. 2.

3.1 Vector Field Representation

In VF-NeRF , the scene geometry is defined using unit vectors that point towards the nearest surface. Let $\Omega \subset \mathbb{R}^3$ be the surface of an object in $\mathbb{R}^3$ and $\Gamma \subset \mathbb{R}^3$ be the set of unit norm 3-vectors. We make use of the VF definition [41]: VF is a function $\mathbf{f} : \mathbb{R}^3 \to \Gamma$ that maps a point in space to a unit vector directed to the closest surface point of Ω:

$$\mathbf{f}(\mathbf{x}) = \begin{cases} \dfrac{\mathbf{x}_S - \mathbf{x}}{||\mathbf{x}_S - \mathbf{x}||_2} & \text{if } \mathbf{x} \notin \Omega \\ \dfrac{\widehat{\mathbf{x}}_S - \widehat{\mathbf{x}}}{||\widehat{\mathbf{x}}_S - \widehat{\mathbf{x}}||_2} & \text{if } \mathbf{x} \in \Omega \end{cases} \tag{1}$$

where $\mathbf{x}_S = \arg\min_{\mathbf{s}\in\Omega} ||\mathbf{x} - \mathbf{s}||_2$ is the closest surface point with respect to $\mathbf{x}$, and $\widehat{\mathbf{x}} = \lim_{||\epsilon||_2 \to 0} \mathbf{x} + \epsilon$ is a point close to the surface, with $\epsilon \in \mathbb{R}^3$ being an infinitesimal 3D vector.

Given the definition of the VF representation, we identify a surface Ω between a point $\mathbf{x} \in \mathbb{R}^3$ and an infinitesimally close neighbor using the cosine similarity between the VF at the two points. When the two points are on opposite sides of the surface Ω, their cosine similarity approaches -1. Conversely, it is close to 1 everywhere else, except at diverging discontinuities of the field.

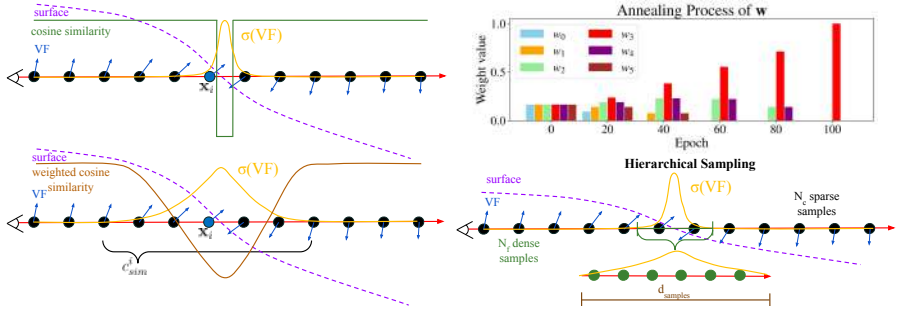
$$\Omega = \{\mathbf{x}_1, \mathbf{x}_2 = \mathbf{x}_1 + \epsilon | \cos(\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2)) < \tau\}, \quad \cos(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{||\mathbf{v}_1||_2 ||\mathbf{v}_2||_2} \tag{2}$$

where $\epsilon \in \mathbb{R}^3$ is an infinitesimal displacement and $|\tau| \leq 1$ is a cosine similarity threshold. Ideally, $\tau = -1$ for infinitesimally close neighbors.

From these definitions, we notice a similarity to the surface density $\sigma : \mathbb{R}^3 \to \mathbb{R}_{\geq 0}$, a function that indicates the rate at which a ray is occluded at location $\mathbf{x}$. Ideally, for non-translucid surfaces, $\sigma(\mathbf{x})$ behaves as a delta function, being zero everywhere except at the surface. To model this function typically used in volume rendering [20, 34], a simple transformation of the cosine similarity can be used. In fact, the cosine similarity between the VF of infinitesimally close neighbors is a delta function itself, yielding approximately 1 everywhere and -1 at the surface. However, as we show, a smooth function is necessary in order to ease the learning of VF through volume rendering.

3.2 Density as Transformed VF

We draw inspiration from existing methods [51, 56], which use neural volume rendering to learn the geometry of a scene as an implicit function. Contrary to



(a) Density and weighted cosine similarity

(b) Sliding window weights and hierarchical sampling

Fig. 3: (a) Density using non-averaged and averaged cosine similarity. The figures show the VF, cosine similarity and density of a ray crossing a surface. Top: density as a transformation of the cosine similarity. This yields a sharp function similar to the delta function centered at the surface. Bottom: Density as a transformation of the weighted average cosine similarity. This produces a smoother function with the maximum centered at the surface.
(b) Sliding window weights annealing example and hierarchical sampling. **Top:** Example of weights at different stages of the annealing. In this case, the sliding window contains 6 weights. At the beginning of the training (epoch 0), the weights for each neighbor are equal. At the end of the training (epoch 100) the cosine similarity is computed only with respect to the closest next neighbor. **Bottom:** Initially, we sample uniform points along the ray and compute the surface density through the predicted VF. We then densely sample points within a range $d_{samples}$ centered at the maximum of the surface density.

these previous methods that use SDF, we propose to model the surface density as a function of the learnable VF. Given a viewing ray and the VF sampled at multiple points along the ray, we use a differentiable process to estimate the surface density. As previously highlighted, the cosine similarity of the VF at neighboring points along the ray can be used to indicate whether there is a surface between them. The resulting surface density function, showcased in Fig. 3a top row, closely resembles a delta function. This behavior is desirable in order to obtain sharp reconstructions; however, due to its discontinuity, the desired convergence is hard to achieve. In order to make the gradient-based optimization tractable, we first need a smoothing transformation. To this end, we adopt a sliding window approach and compute a weighted average cosine similarity. We thus smooth the function at points near the surface. The effect of the sliding window can be seen in the bottom row of Fig. 3a.

Given a set of samples in a ray, we initially predict the VF at each point of the ray. We then define the weights of a sliding window of size M, where the size is an even number, as $\mathbf{w} = [w_0, w_1, ..., w_{M-1}]$. The sliding window and the predicted VF are used to compute the weighted average cosine similarity associated with each point. The smoothed cosine similarity of a point is computed as the weighted average of the cosine similarities using multiple forward and backward neighbors

of the ray. Therefore, given a ray of $N+1$ points $\mathbf{r} = [\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_N]$, we can compute N averaged cosine similarities as:

$$c^i_{sim}(\mathbf{r}) = \sum_{j=0}^{M/2-1} \left[w_j \cos\left(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_{i-j-1})\right) + w_{j+M/2} \cos\left(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_{i+j+1})\right) \right] \quad (3)$$

For simplicity, in Eq. (3) we do not consider the boundary cases of computing the weighted average cosine similarity of the first and last samples along rays. However, note that the cosine similarity of the first and last points of the ray is not smooth because the sliding window would go out of range. Additionally, given N+1 points, we can only compute N cosine similarities since the last point of the ray does not have a successor.

The effect of the weighted sliding window can be changed by modifying its weights. We start with a uniform distribution where all the weights are equal and sum up to 1. We introduce an annealing process to progressively add more weight to the closest neighbors with the final objective to end with a one-hot vector where all the weight is located at the closest next neighbor. Thanks to this approach, the network can be easily optimized, while preserving the desired sharpness during inference. The annealing process is depicted in Fig. 3b. This process is linear and depends on the training epoch. Specifically, the weights of the sliding window are computed at the beginning of every epoch using the following equation:

$$\widehat{w}_i = \frac{M}{2} \mathrm{ReLU}\left(1 - \frac{n|i - M/2|}{N_{epochs}}\right), \quad w_i = \frac{\widehat{w}_i}{||\widehat{\mathbf{w}}||}. \quad (4)$$

Using sliding window cosine similarity, we redefine the surface density as a transformation that maps a point in the ray $\mathbf{r} \in \mathbb{R}^{(N+1)\times 3}$ to a scalar value, $\sigma : \mathbb{R}^{(N+1)\times 3} \times \mathbb{N}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$. Leveraging the cosine similarity, we define the surface density as follows:

$$\sigma(\mathbf{r}, i) = \mathrm{ReLU}(\alpha \Psi_{\mu,\beta}(-c^i_{sim}(\mathbf{r})) - \alpha \Psi_{\mu,\beta}(\xi)). \quad (5)$$

where $\alpha, \mu, \beta > 0$ are learnable parameters and ξ is a cosine similarity threshold value left as a hyperparameter. ReLU is the rectified linear unit and $\Psi_{\mu,\beta}$ represents the Cumulative Distribution Function (CDF) of the Laplace distribution. μ denotes the Laplacian mean, while β is Laplacian "diversity" and α is a scaling factor. Formally, the Laplacian CDF is defined as follows:

$$\Psi_{\mu,\beta}(x) = \begin{cases} 1 - \exp\left(-\frac{|x-\mu|}{\beta}\right) & \text{if } x > \mu \\ \exp\left(-\frac{|x-\mu|}{\beta}\right) & \text{if } x \leq \mu. \end{cases} \quad (6)$$

With this definition of the density function, we can accumulate the densities and colors using numerical quadrature [34] to render the color and depth of the pixel associated with the ray:

$$C(\mathbf{p}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i \tag{7}$$

$$D(\mathbf{p}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))t_i \tag{8}$$

where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$ is the accumulated transmittance and $\delta_i = t_{i+1} - t_i$ is the distance between samples along a ray. Note that Eqs. (7) and (8) can be seen as traditional alpha compositing with alpha values $\alpha_i = 1 - \exp(\sigma_i \delta_i)$.

3.3 Hierarchical Ray Sampling

Sampling rays densely in a uniform manner proves highly inefficient due to the prevalence of free space and occluded regions along the ray, which do not contribute significantly to volume rendering. To address this challenge, similarly to prior works [16,25,34,50,51,56,58], we propose a hierarchical sampling strategy to allocate samples selectively in regions likely to contain surfaces. Initially, we sparsely sample $N_c = 100$ points along a ray and predict their corresponding VFs. Given these predictions, we compute the surface density σ along the ray and densely resample around its maximum. As shown in Fig. 3b bottom row, our dense sampling approach involves uniformly sampling N_f points in a window of size $d_{samples} = 30$cm centered at the point yielding maximum surface density σ. The number of points N_f sampled during this step increases every $n^{inc} = 50$ epochs using a fixed step size $N_f^{inc} = 5$ until reaching a maximum $N_f^{max} = 100$. Consequently, after some epochs we make use of a total of $N_c + N_f^{max} = 200$ points to render the predicted color $C(\mathbf{p})$ and depth $D(\mathbf{p})$. Note that while some works optimize coarse and fine networks simultaneously to predict the surface density [34,40], our approach employs a single network to predict the VF. Hierarchical sampling allows the network to progressively refine the 3D representation of the scene in a coarse-to-fine manner.

3.4 Training

Our approach leverages a dual-MLP structure. First, $\mathbf{f}_\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^{3+256}$ predicts the VF of the scene alongside a global geometry feature vector $\mathbf{z} \in \mathbb{R}^{256}$. Here, ϕ represents the network learnable parameters. Second, $\mathbf{c}_\psi : \mathbb{R}^{3+3+3+256} \rightarrow \mathbb{R}^3$ approximates the radiance field color values based on a given spatial point, viewing direction, VF, and global feature vector. Here, ψ represents the radiance field network learnable parameters. Consequently, for a specific point on a ray $\mathbf{x}$ and its viewing direction $\mathbf{d}$, we can predict the VF as $(\mathbf{v}, \mathbf{z}) = \mathbf{f}_\phi(\mathbf{x})$ and the radiance field as $\mathbf{c} = \mathbf{c}_\psi(\mathbf{x}, \mathbf{v}, \mathbf{d}, \mathbf{z})$. Our model also incorporates three learnable parameters for the density function as described in Eq. (5), namely α, μ and β.

During training, a batch of pixels $\mathcal{P}$ and their corresponding rays are sampled to minimize the difference between the rendered images $\widehat{C}(\mathbf{p})$ and the reference images $C(\mathbf{p})$:

$$\mathcal{L}_c = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} ||\widehat{C}(\mathbf{p}) - C(\mathbf{p})||_1 \tag{9}$$

Learning the geometry of indoor scenes solely from images presents a challenge in reconstructing accurate geometries, even in textured regions. To address this, we enhance the learning of scene representation by introducing a depth consistency loss similarly to [50, 58]. This loss compares the rendered depth, $\widehat{D}(\mathbf{p})$, with a reference depth, symbolized as $D(\mathbf{p})$. Depending on the availability of data, the depth $D(\mathbf{p})$ can be derived from multi-view stereo methods [42, 43, 59] or by monocular depth estimation [13, 14].

$$\mathcal{L}_{depth} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} ||\widehat{D}(\mathbf{p}) - D(\mathbf{p})||_1 \tag{10}$$

In addition to the rendering losses, we add three regularization terms to impose the known properties of VF. First, we impose that the VF has a unit vector property by applying the unit norm loss $\mathcal{L}_{norm}$:

$$\mathcal{L}_{norm} = \frac{1}{N+1} \sum_{i=0}^{N} (||\mathbf{f}(\mathbf{x}_i)||_2 - 1)^2 \tag{11}$$

Additionally, in object-centric scenes, the VF at outer, distant points resembles a vector directed toward the center. Therefore, we incorporate a loss that guides the VF for points outside the scene, denoted as $\mathcal{P}_{ext}$, to point towards the object's center, represented by $\mathbf{c}_{scene}$.

$$\mathcal{L}_{ext} = \frac{1}{|\mathcal{P}_{ext}|} \sum_{\mathbf{x} \in \mathcal{P}_{ext}} \left|\left| \mathbf{f}(\mathbf{x}) - \frac{\mathbf{c}_{scene} - \mathbf{x}}{||\mathbf{c}_{scene} - \mathbf{x}||_2} \right|\right|_2 \tag{12}$$

Finally, considering that in indoor scenes, images are captured from within the scene's geometry, we introduce a loss function that guides points near the scene's center, represented as $\mathcal{P}_{cen}$, to point outwards:

$$\mathcal{L}_{cen} = \frac{1}{|\mathcal{P}_{cen}|} \sum_{\mathbf{x} \in \mathcal{P}_{cen}} \left|\left| \mathbf{f}(\mathbf{x}) - \frac{\mathbf{x} - \mathbf{c}_{scene}}{||\mathbf{x} - \mathbf{c}_{scene}||_2} \right|\right|_2 \tag{13}$$

The overall loss is defined as a weighted sum of the individual losses:

$$\mathcal{L} = w_c \mathcal{L}_c + w_{norm} \mathcal{L}_{norm} + w_{ext} \mathcal{L}_{ext} + w_{depth} \mathcal{L}_{depth} + w_{cen} \mathcal{L}_{center} \tag{14}$$

where w_c, w_{norm}, w_{ext}, w_{depth} and w_{cen} are hyperparameters.

Table 1: Quantitative results. Our method outperforms all baselines in terms of the averaged F-score and median Chamfer Distance. On novel view synthesis, VF-NeRF outperforms all baselines on ScanNet, and renders high-quality images on Replica, being second only to MonoSDF by a small margin. P stands for precision, R for recall and F-1 for F1-score. **Best result**. **Second best result**.

	Replica					ScanNet				
	PSNR ↑	CD ↓	P ↑	R ↑	F-1 ↑	PSNR ↑	CD ↓	P ↑	R ↑	F-1 ↑
COLMAP [43]	-	-	0.760	0.403	0.527	-	-	0.604	0.485	0.538
NeRF [34]	-	-	0.153	0.295	0.201	-	-	0.085	0.166	0.112
UNISURF [37]	-	-	0.195	0.338	0.247	-	-	0.298	0.335	0.315
NeuS [51]	-	-	0.524	0.465	0.493	-	-	0.406	0.437	0.421
VolSDF [56]	-	-	0.317	0.442	0.369	-	-	0.489	0.546	0.516
N-Angelo [25]	31.44	611	0.243	0.323	0.262	17.83	103	0.269	0.188	0.220
M-SDF [16]	27.48	5.6	0.723	**0.856**	0.779	20.78	1.45	0.778	0.694	0.730
NeuRIS [50]	-	-	-	-	-	**24.40**	1.71	0.773	0.682	0.723
MonoSDF [58]	**32.25**	**0.37**	**0.906**	**0.889**	**0.897**	23.84	**1.42**	**0.863**	**0.730**	**0.788**
VF-NeRF	**31.49**	**0.13**	**0.976**	0.842	**0.904**	**26.21**	**0.258**	**0.928**	**0.821**	**0.865**

4 Experiments

Implementation details. Our method is developed using PyTorch [39] and trained using the Adam optimizer [23]. The VF and color functions are designed as MLPs consisting of 8 and 4 hidden layers, respectively. Positional encodings [34] are used for the spatial positions $\mathbf{x}$ and viewing directions $\mathbf{d}$ to address the challenge of learning high-frequency details of the scene. Furthermore, we find that initializing the VF network by pretraining it to point toward the center of the scene eases the training process. The learning rate is initialized at 5×10^{-4} and is decreased using an exponential decay approach [27]. The training process spans 3000 epochs. Notably, weight annealing for the sliding window technique is executed between the 700th and 1400th epochs. Each epoch's iteration count is equivalent to the dataset's training image count, and 1024 rays are sampled during each iteration. For each ray, we make use of our hierarchical sampling strategy. Additionally, we use Truncated Signed Distance Function (TSDF) fusion to extract the surface mesh from the predicted depth maps and images. We set the following weights of the multi-objective loss function: $w_c = 1.0$, $w_{norm} = 0.05$, $w_{ext} = w_{cen} = 0.5$, $w_{depth} = 0.25$. Regarding the density function parameters, we set the cosine similarity threshold to $\xi = -0.5$ and initialize the learnable parameters to $\mu = 0.7$, $\beta = 0.5$ and $\alpha = 100$.

Datasets. We test the performance on Replica [47] and ScanNet [9]. The Replica dataset consists of 18 synthetic indoor scenes, where each scene contains a dense ground truth mesh, and 2000 RGB and depth images. Similarly to MonoSDF [58], we focus on only seven scenes from this dataset for comparison purposes. The ScanNet dataset contains 16113 indoor scenes with 2.5 million views, with each view containing RGB-D images. Additionally, a fused mesh is provided for each scene. We select the four scenes from this dataset used by ManhattanSDF [16]

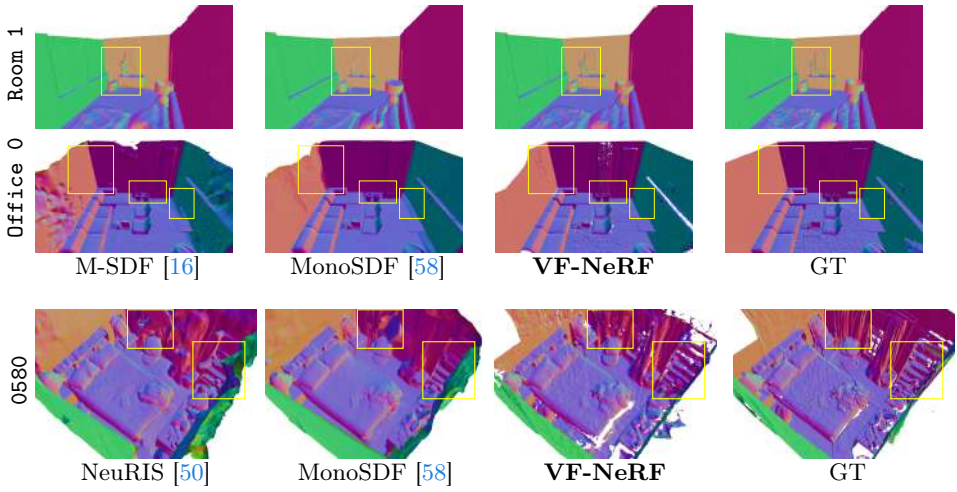


Fig. 4: 3D reconstruction qualitative results. VF-NeRF outperforms the SOTA in planar regions of the scenes such as walls and floors as well as in several details. We highlight regions where VF-NeRF outperforms the other methods with yellow boxes. We note that these include both planar and detailed regions.

and MonoSDF to evaluate our method. For replica, we sample 1 of every 20 posed images for training, while in Scannet we sample 1 of every 40.

Metrics. We evaluate following standard protocol [58]. For 3D surface reconstruction, we focus on evaluating our method with median Chamfer distance (CD) and F1-score [48] with a threshold of 5cm. We also provide the peak signal-to-noise ratio (PSNR) to evaluate view synthesis. The detailed definitions of these metrics are included in the supplementary material.

Baselines. We compare our method against the State of the Art (SOTA), which use volume rendering for indoor scene reconstruction: ManhattanSDF [16], MonoSDF [58], NeuRIS [50] and Neuralangelo [25]. We use Marching Cubes [32] to extract the meshes rendered by the baselines.

4.1 Comparisons with baselines

3D reconstruction. We evaluate our method on the Replica and ScanNet datasets. The qualitative results on Replica and ScanNet are illustrated in Fig. 4. Quantitative results on both datasets are shown in Tab. 1. Additional detailed qualitative and quantitative results are included in the supplementary material. Our method outperforms volume rendering based benchmarks in terms of F1-score and median CD on both datasets. Most interestingly, the gap in performance is significantly higher in ScanNet, a more challenging dataset containing noisy depth maps. The ability to perform extremely well on real depths/images might be explained by the strong inductive bias offered by VF which allows it to learn planar regions even in the presence of noisy data.

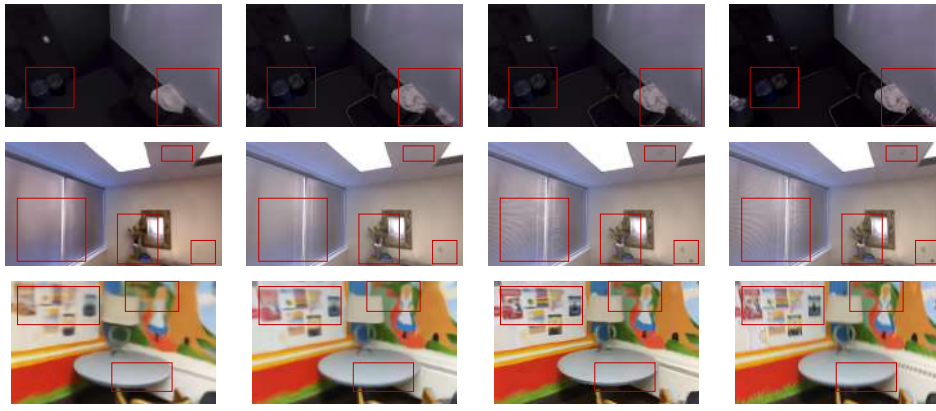


Fig. 5: Novel view synthesis qualitative results. Our method renders accurate images with high-frequency details. Compared to ManhattanSDF, our method is more accurate and introduces less smoothness in both datasets. Additionally, VF-NeRF is more effective than MonoSDF in preserving high frequency details (e.g. the blinds). Interestingly, we observe that in the bottom example, VF-NeRF renders an image that, at spots (e.g. the drawing on the wall), is sharper than the GT image which suffered from motion blur.

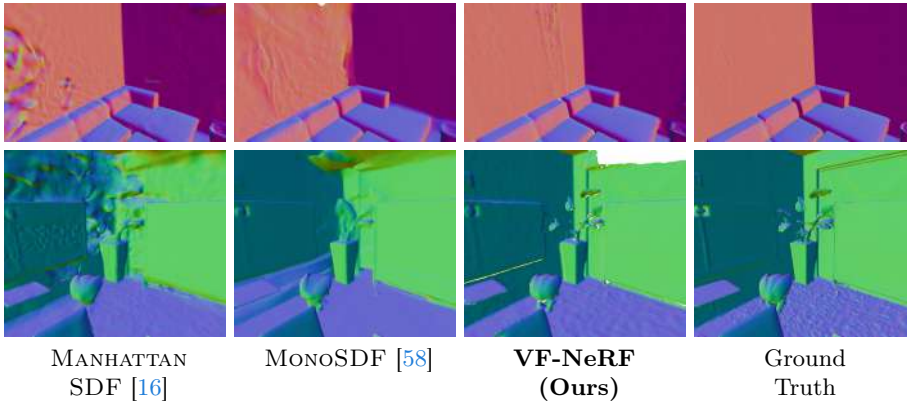
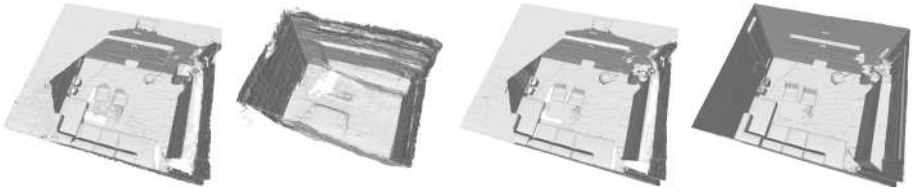


Fig. 6: 3D reconstruction of planar regions. VF-NeRF represents planar surfaces with higher accuracy and fewer artifacts compared to the SOTA. Besides planar regions, note that high frequency details are still preserved in our method (see plant in second row meshes).

The performance of Neuralangelo drops due to the lack of texture in indoor scenes, since it only makes use of color images as supervision signal throughout the optimization. ManhattanSDF can generally recover high-quality scenes,



w\o hierarchical sampling | w\o $\mathcal{L}_{depth}$ | **VF-NeRF (Ours)** | Ground Truth

Fig. 7: Ablations. Removing the hierarchical sampling generates holes in the reconstructed surfaces and artifacts (see table in the meshes). Our method without $\mathcal{L}_{depth}$ is less accurate, as most regions of the scene are low-textured. Nonetheless, it still captures the overall scene coarse geometry.

although it struggles in some planar areas due to its dependency on semantic segmentation masks. MonoSDF generally achieves impressive results, although several artifacts appear in some planar regions. In contrast, our method can recover planar surfaces with great fidelity as well as many fine details (the flowers and picture on the wall or the trash bins in Fig. 4). The capacity of our method to represent planar surfaces compared to the baselines is depicted in Fig. 6.

Novel view synthesis. We present qualitative and quantitative novel view synthesis comparisons in Fig. 5 and Tab. 1. VF-NeRF generally renders high-quality views that preserve high-frequency details and outperforms ManhattanSDF and Neuralangelo in terms of PSNR in both datasets. Additionally, VF-NeRF outperforms all baselines in ScanNet, a more realistic setup with real data, and comes second to MonoSDF in Replica by a small margin. More qualitative results can be found in the supplementary material.

4.2 Ablations

Loss terms. We analyze the impact of different loss terms on the surface representation and provide quantitative results. Specifically, we remove the following losses and study their effects: center and exterior supervision $\mathcal{L}_{center}$, $\mathcal{L}_{ext}$, unit norm $\mathcal{L}_{norm}$ and depth $\mathcal{L}_{depth}$. We demonstrate that removing these losses decreases the performance of our method, as presented in Tab. 2. Additionally, removing the depth loss significantly decreases the performance, although the coarse geometry is still preserved, as demonstrated in Fig. 7 since most of the surfaces of the scene do not have enough texture.

Sliding window annealing and initialization. Tab. 2 presents the results of ablating the sliding window cosine similarity and the custom VF network initialization. In the case of the sliding window annealing, we use cosine similarity with the next point of the ray instead of the weighted average. This is the same as just using the sliding window as a one-hot vector where all the weight is located at the closest next neighbor. The results show that both elements enhance the surface reconstruction.

Table 2: Ablations quantitative results. Removing loss terms, sliding window annealing, the VF network initialization or the hierarchical sampling decreases our method's performance.

	Precision↑	Recall↑	**F1-score↑**	**CD (mm)↓**
w\o annealing	0.964	0.809	0.880	0.10
w\o initialization	0.901	0.793	0.844	0.13
Uniform sampling	0.942	0.818	0.876	0.14
w\o $\mathcal{L}_{center}, \mathcal{L}_{ext}$	0.952	0.804	0.872	0.11
w\o $\mathcal{L}_{norm}$	0.928	0.801	0.860	0.11
w\o $\mathcal{L}_{depth}$	0.421	0.263	0.324	63.5
VF-NeRF	**0.986**	**0.817**	**0.894**	**0.09**

Sampling. We investigate the importance of our sampling strategy introduced in Sec. 3.3. We showcase the results achieved when using only uniform sampling in Tab. 2. As expected, we find that using a hierarchical sampling strategy enhances the performance of our method. Additionally, we find that removing the hierarchical sampling and just using uniform sampling generates many artifacts as depicted in Fig. 7.

4.3 Limitations

One limitation of our method is its inherent smoothing bias, which sometimes makes it hard to represent high-frequency details by self-supervised learning. Although our method is generally capable of representing high-frequency details, it struggles in some cases. Additionally, VF-NeRF assumes homogeneous density with three hyperparameters α, μ, β. Future works could explore using different hyperparameters depending on the geometric characteristics.

5 Conclusion

In this work, we presented VF-NeRF, a novel NERF approach for multiview surface reconstruction, utilizing Vector Fields (VFs) to encapsulate the scene's geometry. By transforming the VF, we can represent the volume density of each point in the scene. The key idea is to learn the VF of the scene through volume rendering. Additionally, we proposed a hierarchical sampling approach which enables us to sample more densely near surfaces, improving the efficiency and precision of our method. The experiments demonstrate the performance of our method to reconstruct indoor scenes, outperforming state-of-the-art methods on indoor datasets. Furthermore, our method is capable of rendering novel views that preserve high-frequency details and outperforms several baselines in Replica and ScanNet. Finally, we showcased the effectiveness of our method to reconstruct planar surfaces, while preserving details present in the scenes.

References

1. Agrawal, M., Davis, L.: A probabilistic framework for surface reconstruction from multiple images. CVPR (2001) 3
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. SIGGRAPH (2009) 3
3. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo - stereo matching with slanted support windows. BMVC (2011) 3
4. Broadhurst, A., Drummond, T., Cipolla, R.: A probabilistic framework for space carving. ICCV (2001) 3
5. Chen, R., Han, S., Xu, J., Su, H.: Point-based multi-view stereo network. ICCV (2019) 3
6. Cheng, S., Xu, Z., Zhu, S., Li, Z., Li, L.E., Ramamoorthi, R., Su, H.: Deep stereo using adaptive thin volume representation with uncertainty awareness. CVPR (2020) 3
7. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. ECCV (2016) 4
8. Coughlan, J., Yuille, A.: Manhattan world: compass direction from a single image by bayesian inference. ICCV (1999) 2, 4
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. CVPR (2017) 2, 10
10. Fan, H., Su, H., Guibas, L.: A point set generation network for 3d object reconstruction from a single image. CVPR (2017) 4
11. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE TPAMI (2010) 1, 3
12. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. ICCV (2015) 3
13. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency (2017) 9
14. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation (2019) 9
15. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. CVPR (2020) 3
16. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. CVPR (2022) 1, 2, 4, 8, 10, 11, 12
17. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004) 1
18. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: End-to-end deep plane sweep stereo. ICLR (2019) 3
19. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. CVPR (2019) 4
20. Kajiya, J.T., Herzen, B.V.: Ray tracing volume densities. SIGGRAPH (1984) 5
21. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson Surface Reconstruction. Symposium on Geometry Processing (2006) 3
22. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. SIGGRAPH (2013) 3
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2015) 10
24. Kutulakos, K., Seitz, S.: A theory of shape by space carving. ICCV (1999) 3

25. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. CVPR (2023) 2, 4, 8, 10, 11
26. Li, Z., Wang, Q., Cole, F., Tucker, R., Snavely, N.: Dynibar: Neural dynamic image-based rendering (2023) 2
27. Li, Z., Arora, S.: An exponential learning rate schedule for deep learning. ICLR (2020) 10
28. Lin, C.H., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3d object reconstruction. AAAI Conference on Artificial Intelligence (2018) 4
29. Lin, C.H., Wang, C., Lucey, S.: Sdf-srn: Learning signed distance 3d object reconstruction from static images. Advances in Neural Information Processing Systems **33**, 11453–11464 (2020) 4
30. Liu, L., Gu, J., Lin, K.Z., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. NeurIPS (2020) 2, 4
31. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. CVPR (2019) 4
32. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH (1987) 11
33. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. CVPR (2021) 2, 4
34. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. ECCV (2020) 2, 4, 5, 7, 8, 10
35. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. SIGGRAPH (2022) 4
36. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. CVPR (2020) 4
37. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. ICCV (2021) 4, 10
38. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. CVPR (2019) 4
39. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library (2019) 10
40. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: Neural Radiance Fields for Dynamic Scenes. CVPR (2020) 4, 8
41. Rella, E.M., Chhatkuli, A., Konukoglu, E., Gool, L.V.: Neural vector fields for implicit surface representation and inference (2022) 1, 2, 4, 5
42. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. ECCV (2016) 1, 3, 9
43. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. CVPR (2016) 1, 3, 4, 9, 10
44. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. CVPR (2006) 3
45. Seitz, S., Dyer, C.: Photorealistic scene reconstruction by voxel coloring. CVPR (1997) 3

46. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. IJCV (2008) 1
47. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C.Y., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.A.: The replica dataset: A digital replica of indoor spaces. ArXiv (2019) 2, 10
48. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: Neuralrecon: Real-time coherent 3d reconstruction from monocular video. CVPR (2021) 11
49. Tola, E., Strecha, C., Fua, P.V.: Efficient large-scale multi-view stereo for ultra high-resolution image sets. Machine Vision and Applications (2011) 1
50. Wang, J., Wang, P., Long, X., Theobalt, C., Komura, T., Liu, L., Wang, W.: Neuris: Neural reconstruction of indoor scenes using normal priors. ECCV (2022) 4, 8, 9, 10, 11
51. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. NeurIPS (2021) 2, 4, 5, 8, 10
52. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2vox: Context-aware 3d reconstruction from single and multi-view images. ICCV (2019) 4
53. Yang, X., Lin, G., Chen, Z., Zhou, L.: Neural vector fields: Implicit representation by explicit learning (2023) 2, 4
54. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. CVPR (2019) 3
55. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. ECCV (2018) 3
56. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. NeurIPS (2021) 2, 4, 5, 8, 10
57. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. NeurIPS (2020) 4
58. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. NeurIPS (2022) 1, 2, 4, 8, 9, 10, 11, 12
59. Zheng, E., Dunn, E., Jojic, V., Frahm, J.M.: Patchmatch based joint view selection and depthmap estimation. CVPR (2014) 1, 9