

MonoPatchNeRF: Improving Neural Radiance Fields with Patch-based Monocular Guidance

Yuqun Wu^{1*}, Jae Yong Lee^{1*}, Chuhang Zou²,
Shenlong Wang¹, and Derek Hoiem¹

¹ University of Illinois at Urbana-Champaign
² Amazon Inc.

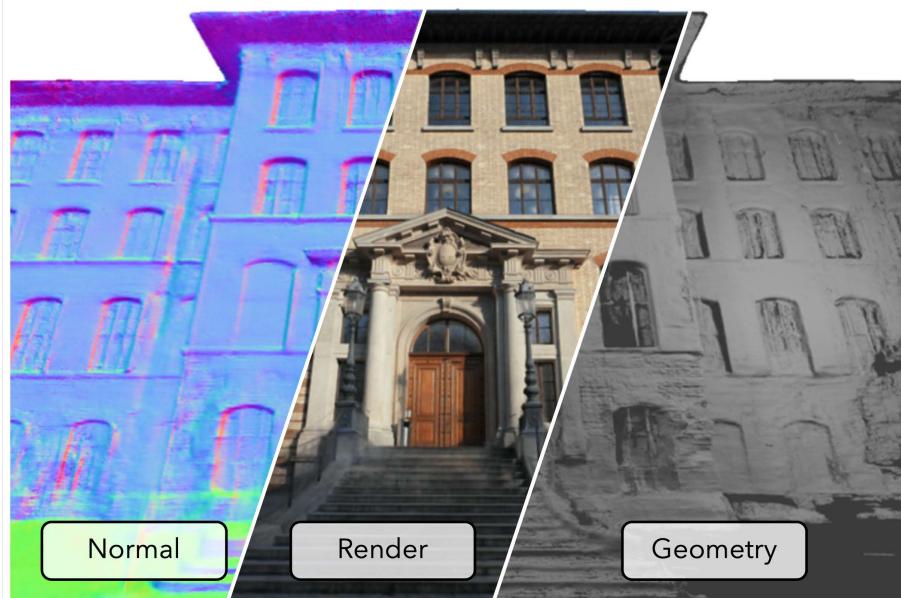


Fig. 1: We present **MonoPatchNeRF** on a sparse-view scene *facade*. Our method renders realistic images and accurate normals from the test view and reconstructs the complete mesh.

Abstract. The latest regularized Neural Radiance Field (NeRF) approaches produce poor geometry and view extrapolation for multiview stereo (MVS) benchmarks such as ETH3D. In this paper, we aim to create 3D models that provide accurate geometry and view synthesis, partially closing the large geometric performance gap between NeRF and traditional MVS methods. We propose a patch-based approach that effectively leverages monocular surface normal and relative depth predictions. The patch-based ray sampling also enables the appearance regularization of normalized cross-correlation (NCC) and structural similarity (SSIM) between randomly sampled virtual and training views. We further show that “density restrictions” based on sparse structure-from-motion points can help greatly improve geometric accuracy with a slight drop in novel

*Equal contribution

[†]Project page: <https://yuqunw.github.io/MonoPatchNeRF/>

view synthesis metrics. Our experiments show 4x the performance of RegNeRF and 8x that of FreeNeRF on average F1@2cm for ETH3D MVS benchmark, suggesting a fruitful research direction to improve the geometric accuracy of NeRF-based models, and sheds light on a potential future approach to enable NeRF-based optimization to eventually outperform traditional MVS.

1 Introduction

Modeling 3D scenes from imagery is a core problem in computer vision, directly useful for mapping, facility assessment, robotics, construction monitoring, and more. These applications require both accurate geometry and realistic visualization from novel views, and may involve large scenes with sparse views.

Neural Radiance Field (NeRF) [23] approaches can synthesize novel views with high fidelity, especially when interpolating training views. The NeRF approach also has great potential for 3D geometry estimation, as a single model can be optimized according to many losses and constraints, encoding photometric consistency, single-view predictions, and material properties. However, this potential is currently unrealized — even the latest regularized approaches like RegNeRF [24] and FreeNeRF [34] produce poor geometry and view extrapolation for multiview stereo (MVS) benchmarks such as ETH3D [29]. While MVS methods typically produce superior geometry in numerous contexts, they often result in noisy and incomplete models, and they do not support high-quality rendering. Furthermore, enhancing MVS methods by introducing additional cues and priors is challenging due to the inherent heuristics and algorithmic complexity.

Recent neural field based works have led to some improvements in geometric fidelity and the capability to handle sparse views [10, 24, 34, 40]. MonoSDF [38] incorporates monocular surface normal and depth predictions in a signed distance function (SDF) model. This works well for objects or small indoor scenes, but due to the initialization with a sphere, smoothness priors, and volumetric representation, this approach does not readily provide good results for large scenes, open scenes, or those with fine structures like bicycle racks. In more recent SDF based work, Neuralangelo [16], smoothness priors are learned as a curvature loss with coarse-to-fine scheduled learning, which works well on challenging thin structures, but our experiments indicate limited effectiveness in a sparse view setup. RegNerf [24] incorporates losses of appearance likelihood and smooth depth for patches on randomly sampled virtual views, improving results when sparse input views are available. Yet, as our experiments show, this virtual view patch-based regularization is not sufficient to achieve geometrically accurate models. While not complete solutions, these approaches contain two critical ideas that we will build on: (1) leverage monocular surface normal and depth predictions to guide the model; (2) encourage virtual view appearance consistency to further regularize the model.

In this paper, we aim to create 3D models that provide accurate geometry and view synthesis, using a patch-based approach that effectively leverages monocular depth/normal predictions and virtual view appearance consistency priors. We

have found that density-based models are better suited for encoding details in large scenes compared to SDF-based models. However, integrating monocular predictions of surface normal and up-to-scale depth into the ray-based sampling of density models is difficult because surface normal and relative depth are not well-defined along a ray in the density model. With our patch-based sampling technique, we use these monocular cues more effectively. Additionally, patch-based sampling allows us to take advantage of structural losses for regularization, such as normalized cross-correlation (NCC) and structural similarity (SSIM). These are robust to varying lighting conditions, hence extensively employed in classic MVS. Using these losses, we boost patch-level photometric consistency across randomly sampled virtual views, significantly constrain the 3D structure, and reduce artifacts. We further leverage monocular depth predictions by translating and scaling them based on sparse structure-from-motion (SfM) points and restricting surfaces to lie within a dilation of the union of all thus aligned points. This “density restriction” greatly improves geometric accuracy, with a slight drop in novel view synthesis metrics. With these improvements, we achieve promising results on the ETH3D benchmark, outperforming other NeRF-based methods in novel view synthesis and by a large margin in geometry. Results fall well short of state-of-art MVS methods for geometric accuracy, but we believe this work supports the potential of NeRF-based optimization to eventually surpass traditional MVS approaches.

In summary, this paper offers the following **contributions**:

- We demonstrate the effective use of monocular surface normal and relative depth predictions for density-based NeRF models, proposing losses based on patch-based ray sampling optimization and occupancy constraints.
- Using patch-based ray sampling, we demonstrate the effectiveness of NCC and SSIM photometric consistency losses between patches from virtual views and training views, creating stronger constraints than the appearance likelihood and depth smoothness constraints of RegNeRF.
- Our method significantly improves geometric accuracy, achieving **4x the performance of RegNeRF and 8x that of FreeNeRF** on average F1@2cm for the ETH3D MVS benchmark and produces competitive results in novel view synthesis, ranking **best in terms of SSIM and LPIPS**. These findings indicate a fruitful research direction to improve the geometric accuracy of NeRF-based models.

2 Related Works

Reconstructing the 3D scene from a set of images is a fundamental problem in computer vision. Multi-view Stereo (MVS), aiming to reconstruct geometry given images with known poses, is a well-studied field and develops from early works [18] with pure photometric scoring to more recent approaches that incorporate deep learning techniques [9, 13, 35, 36]. While there are various formulations of scene representation, state-of-the-art methods typically predict the depth map of each image based on photometric consistency with a set of source views [12, 13, 19, 28],

and use geometric consistency across views to fuse the depths together into a single point cloud [6, 28]. The fused point clouds are evaluated against the ground truth geometry for precision-and-recall [11], or accuracy-and-completeness [29], with F_1 score combining the harmonic mean of the two values. Different from MVS, our method constructs a volumetric density representation and extracts depth maps using volumetric rendering of expected depth at source views, followed by a conventional depth-map fusion pipeline [6].

While MVS approaches solve for precise geometries [11, 29], the novel view rendering of meshes or point clouds generated by MVS is somewhat inferior compared to Neural Radiance Field (NeRF), a recently popular alternative for 3d reconstruction. NeRF [23] provides an elegant and effective solution for novel view synthesis. While NeRF has been shown to reliably estimate geometry and appearance in dense captures, multiple papers [10, 24, 26, 34, 37] show that it often fails to converge correctly in outward-facing, sparse, or wide-baseline inputs. PixelNeRF [37] and DietNeRF [10] propose learning based feed-forward solutions that utilize prior knowledge for sparse input scenario. PixelNeRF represents the scene as extracted CNN features, and DietNeRF samples random poses and supervises random views using consistency with CLIP [26]. Both methods manage to render smooth novel views for scenes with sparse inputs, but suffer from degradation in rendering quality due to the low-resolution CNN features or indirect supervision.

Extending on PixelNeRF and DietNeRF, RegNeRF [24] proposes to use stronger regularization on randomly sampled patches via a geometric smoothness loss and an adversarial photometric loss. SPARF [30] jointly optimizes the NeRF models and camera poses with extracted pixels matches on input views, and improves performance given sparse inputs. Meanwhile, FreeNeRF [34] which modulates frequency in positional-encoded space shows that a simple coarse-to-fine approach is able to alleviate object-centric data-sparsity [1] and front-facing [21] scenes. DeLiRa [7] samples synthesized view points and warps the dilated rays for the photometric consistency loss. Previous methods target at solving the sparse views problem with more regularization through learned prior or virtual view smoothing geometry, but do not fully utilize the underlying geometry consistency. Our work extends RegNeRF [24] for sampling virtual views, but apply a photometric loss between virtual and training patches through learned geometry consistency for a better per scene regularization.

Using image-based priors in regularization has also been shown effective in training Neural SDF and NeRF based models. Roessle et al. [27] and Nerfing-MVS [32] use a pretrained network to estimate dense depth given sparse SfM points, and supervise the NeRF model with the estimated depth. However, the sparse depth from SfM often contains noise that can be passed down to the dense prediction [3, 25]. MonoSDF [38] proposes to use depth and normals estimated with RGB images and significantly improve the geometric quality of SDF models. NeuralWarp [4] proposes cross view photometric consistency for training patches as MVS with visibility information from SfM. While also using patch warping during optimization, our method enforces a consistency between virtual patches

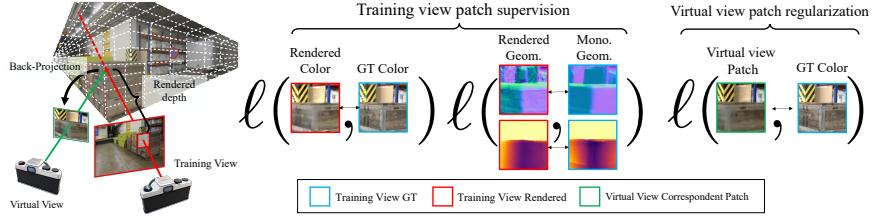


Fig. 2: Overview of our architecture. Our MonoPatchNeRF contains three major types of losses: 1) color supervision of RGB images, 2) geometric supervision of monocular depth and normal maps, and 3) virtual view patches regularization between randomly sampled patches and corresponding ground truth RGB pixels. We sample the virtual view pose via random translations from the training view camera center, and obtain the virtual view corresponding patch by rendering along the back-projected ray that is unprojected with the rendered depth from the training view (Figure 3). Additionally, we limit the density search space by pruning out the regions using the monocular geometry (Figure 4).

and training patches, which fits better for sparse view scenario, and we apply occlusion mask with geometry consistency determined by network outputs, as virtual patches do not have the visibility prior (Figure 3). Though with better geometry, SDF based approaches do not perform well in inferring high-frequency details such as bicycles and chair arms. Neuralangelo [16] partly resolves this problem using coarse-to-fine optimization and improves the geometry by using numerical gradient for higher derivatives, but our experiments indicate limited effectiveness in a sparse view scenario. While monocular cues have been widely applied, it is well known that the monocular depth is consistent locally, and the global inconsistency issue can lead to a worse geometry. We instead incorporate patch based monocular priors with a local scale-and-shift transformation, to better utilize the strength of monocular depth.

3 Method

Given a collection of posed images capturing a large unbounded scene, our goal is to construct a 3D neural radiance field representation that not only renders high-quality images from any viewpoint but also provides accurate and complete surface geometry. We achieve this by introducing three novel components: 1) patch-based monocular geometry cues, allowing more accurate and structured geometry; 2) patch-level photometric consistency regularization across virtual and training viewpoints, enhancing the overall rendering and geometry quality 3) density restriction through monocular cues and sparse geometry, inhibiting NeRF from predicting opacity on obvious empty space. Figure 2 provides a summary of our approach.

3.1 Background: Neural Radiance Fields (NeRF)

The Neural Radiance Field (NeRF) [22] establishes a parametric representation of the scene, enabling realistic rendering of novel viewpoints from a given image collection. NeRF is defined by $\mathbf{c}, \sigma = F_\theta(\mathbf{x}, \mathbf{v})$, where \mathbf{c} is output color, σ represents opacity, \mathbf{x} is the 3D point position, and \mathbf{v} denotes the viewing direction. The variable θ represents learnable parameters optimized for each scene.

Considering an input query viewpoint with camera parameters \mathbf{K} , \mathbf{R} , and \mathbf{t} (representing intrinsic parameters, rotation, and translation), a pixel's camera viewing ray can be calculated as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{v}$, where \mathbf{o} is the camera center and \mathbf{v} is the ray direction. The pixel color is then computed through volume rendering as $\mathbf{c}(\mathbf{r}) = \sum_{i=1}^N w_i \mathbf{c}_i$, with \mathbf{c}_i being the NeRF color evaluated from the i -th sampled point $\mathbf{x}_i = \mathbf{o} + t_i \mathbf{v}$ along the ray. Weight w_i is calculated through alpha composition as $w_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) (1 - \exp(-\sigma_i \delta_i))$, where σ_i defines sampled point's opacity and $\delta_i = t_i - t_{i-1}$ defines the distance between the i -th sample and its preceding point. The depth of each pixel can also be rendered through volume rendering: $\mathbf{d}(\mathbf{r}) = \sum_{i=1}^n \hat{w}_i t_i$, where \hat{w}_i is the weight normalized along the ray.

NeRF is trained by minimizing the difference between the rendered pixel value \mathbf{c} and the observed pixel value $\hat{\mathbf{c}}$. We apply per-pixel Huber Loss: $L_{\text{huber}}(\mathbf{c}, \hat{\mathbf{c}})$ during the training. Despite its exceptional performance in synthesizing nearby novel views, NeRF falls short in capturing high-quality geometry and rendering extrapolated viewpoints (Figure 6). To overcome these challenges, we propose patch-based supervision and regularization, both in the form of additional loss functions for training NeRF.

3.2 Distillation of Patch-based Monocular Cues

Learning-based networks for single-image normal and depth prediction can provide robust cues for the geometry of a scene. Taking inspiration from this, prior works [8, 38] exploit monocular depth and normal supervision to determine the geometry of neural fields. However, these works introduce monocular depth supervision on a per-pixel basis, which does not account for that monocular methods tend to only be accurate locally up to a translation and scale. To address this challenge, it is typically necessary to incorporate either per-batch scale estimation or relative depth supervision. We choose to employ a robust scale-and-shift invariant loss function, which provides superior scale invariance.

Given a patch $P = \{p\}$ and the rendered depths $\{d_p\}$ and normals $\{\mathbf{n}_p\}$, our patch-based monocular supervision loss is defined as:

$$L_{\text{mono}} = L_{\text{depth}} + L_{\nabla \text{depth}} + L_{\text{normal}} + L_{\nabla \text{normal}}. \quad (1)$$

The depth loss L_{depth} penalizes the discrepancy between the rendered depth $\{d_p\}$ and the monocular estimation depth $\{\hat{d}_p\}$, whereas $L_{\nabla \text{depth}}$ minimizes their gradient discrepancy. Nevertheless, due to the inherent scale and shift ambiguities, directly supervising depth by minimizing the absolute difference will not yield

the best results. We instead optimize the difference between depth images with normalized scale and shift:

$$L_{\text{depth}} = \sum_{p \in P} \|\hat{d}_p^\dagger - d_p\| \quad (2)$$

$$L_{\nabla \text{depth}} = \sum_{p \in P} \|\nabla \hat{d}_p^\dagger - \nabla d_p\|, \quad (3)$$

where $\hat{d}_p^\dagger = s\hat{d}_p + t$ represents the normalized depth using the optimal scale s and shift t estimated from pixels $\{p\}$ with a least-squares criterion [20], and ∇d_p is the gradient value of depth image at pixel p .

Our key insight is that having per-patch scale and shift makes the optimization process more robust to the monocular depth local scale drifts.

We also apply a normal loss with the monocular estimation normals $\hat{\mathbf{n}}$ to regularize our rendered density field, following RefNeRF [31]. Like their approach, we define two separate formulations of rendered surface normals. First, we introduce an additional normal rendering MLP head on NeRF. Using this MLP head our NeRF can produce per-ray normal estimation with volume rendering: $\mathbf{n}^\theta = \sum_i w_i \mathbf{n}_i^\theta$. Next, we estimate surface normal as the negative direction of the gradient of the optical density, denoted as $\mathbf{n}_i^\nabla = -\nabla \sigma_i / \|-\nabla \sigma_i\|$, with rendered form as $\mathbf{n}^\nabla = \sum_i w_i \mathbf{n}_i^\nabla$. We directly supervise $\mathbf{n}^\theta, \mathbf{n}^\nabla$ with the monocular surface normal estimate $\hat{\mathbf{n}}$ as:

$$\begin{aligned} L_{\text{normal}} &= \sum_{p \in P} (1 - \cos(\hat{\mathbf{n}}_p, \mathbf{n}_p^\nabla) + |\hat{\mathbf{n}}_p - \mathbf{n}_p^\nabla|) \\ &\quad + \sum_{p \in P} (1 - \cos(\hat{\mathbf{n}}_p, \mathbf{n}_p^\theta) + |\hat{\mathbf{n}}_p - \mathbf{n}_p^\theta|) \end{aligned} \quad (4)$$

Finally, we also apply the gradient loss $L_{\nabla \text{normal}}$ over \mathbf{n}^∇ :

$$L_{\nabla \text{normal}} = \sum_{p \in P} (|\nabla \hat{\mathbf{n}}_p - \nabla \mathbf{n}_p^\nabla|) \quad (5)$$

3.3 Patch-based Photometric Consistency over Virtual Views

NeRF models tend to be under-constrained when the input images are sparse, which often leads to artifacts in novel view renderings despite a low training view loss. To mitigate this, we employ patch-based photometric consistency regularization.

As visualized in Fig 3, for each patch P with associated colors $\{\hat{\mathbf{c}}_p\}$, pose \mathbf{o}_p , directions $\{\mathbf{v}_p\}$ and rendered depth $\{d_p\}$ in the training set, we randomly sample a corresponding virtual patch P^* from a nearby virtual view. The correspondence is determined by unprojecting the pixel from the training view into 3D points, denoted as $\mathbf{X}_p = \mathbf{o}_p + d_p \mathbf{v}_p$, and then back-projecting \mathbf{X}_p into the pose \mathbf{o}_{p^*} of

P^* for the corresponding direction $\mathbf{v}_{p^*} = (\mathbf{X}_{p^*} - \mathbf{o}_{p^*}) / \|(\mathbf{X}_{p^*} - \mathbf{o}_{p^*})\|$. Given \mathbf{o}_{p^*} and \mathbf{v}_{p^*} , we predict the color \mathbf{c}_{p^*} and depth d_{p^*} , and define our occlusion aware patch-based virtual view consistency loss as follows:

$$L_{\text{virtual}} = L_{\text{SSIM}}(\{\mathbf{c}_{p^*}\}, \{\hat{\mathbf{c}}_p\}, \{M_{p \rightarrow p^*}\}) + L_{\text{NCC}}(\{\mathbf{c}_{p^*}\}, \{\hat{\mathbf{c}}_p\}, \{M_{p \rightarrow p^*}\}) \quad (6)$$

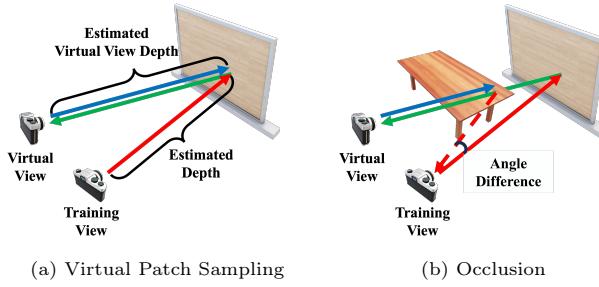


Fig. 3: Virtual View Patch sampling and occlusion visualization. (a) We first sample a virtual view near the training view. Then, we render a patch from the training view to estimate the depth (Red line), which is projected to the virtual view (Green line). Finally, we render the virtual patch (Blue line) and compare the rendered RGB to the ground truth RGB in the training patch. (b) A pixel is marked as occluded and excluded from the NCC and SSIM losses, if the projection of its rendered virtual view depth is inconsistent with the training view depth based on the angle difference.

where L_{SSIM} measures the structural similarity between two patches, L_{NCC} is the normalized cross-correlation between two corresponding patches, and $\{M_{p \rightarrow p^*}\}$ is the occlusion mask from patch P to patch P^* . Both losses have been shown to be robust to view-dependent effects, such as changes in illumination. However, viewing the same point from different perspectives can result in varying occlusion patterns. To ensure the occlusion does not negatively impact our loss functions, we additionally apply mask $\{M_{p \rightarrow p^*}\}$ to remove any clearly occluded pixels. We compute $M_{p \rightarrow p^*}$ by comparing an angle threshold θ_{thresh} and the angle $\theta_{p^* \rightarrow p}$ between rays from \mathbf{o}_p to \mathbf{X}_p and \mathbf{o}_{p^*} to \mathbf{X}_{p^*} :

$$\theta_{p^* \rightarrow p} = \arccos(\mathbf{v}_{p^* \rightarrow p} \cdot \mathbf{v}_p), \quad M_{p \rightarrow p^*} = \begin{cases} 1 & \text{if } \theta_{p^* \rightarrow p} \leq \theta_{\text{thresh}} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$\text{where } \mathbf{v}_{p^* \rightarrow p} = (\mathbf{X}_{p^*} - \mathbf{o}_p) / \|(\mathbf{X}_{p^*} - \mathbf{o}_p)\|, \quad \mathbf{X}_{p^*} = \mathbf{o}_{p^*} + d_{p^*} \mathbf{v}_{p^*}$$

3.4 Density Restriction by Empty Space Pruning with Sparse SfM Geometry

One significant challenge in NeRF is the occurrence of floaters and background collapse [2]. The challenge arises because NeRF fails to predict correct geometry for surfaces with low texture and view-dependent effects, or tends to overfit in

near-camera regions that are unseen from other views during training. We address this problem by limiting the domain of density distributions using monocular geometric prior and sparse multi-view prior (Figure 4).

Monocular depth provides useful relative distance information, and SfM points provide metric depth that can be utilized for aligning the monocular depth with 3D space. For each view, we use RANSAC to solve a scale and shift for monocular depth with the projected sparse points to minimize the influence of noise in the points. We then constrain the grid density distribution to a specified interval surrounding the optimal monocular depth maps for each view. This hard restriction effectively prunes out empty space, thereby eliminating the floaters and improving the overall geometry estimation.

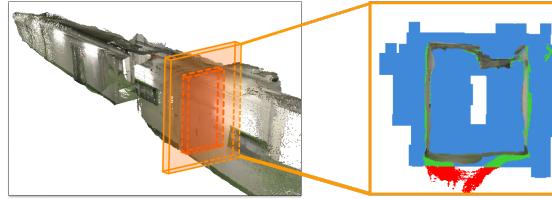


Fig. 4: Visualization of density restrictions. We show on the left the point cloud reconstruction of our model trained under density restrictions. On the right, we capture a vertical slice of the scene to show how density restrictions improve the geometry. The blue area indicates density-restricted region, and colored area represents the point cloud reconstruction of our method trained under density restrictions. Green and red regions indicate the point cloud of our model trained without density restrictions, where green indicates that the region is inside the density-restricted area, and red indicates that the region is outside the density-restricted area.

3.5 Training

We start by estimating monocular geometric cues with images using the pre-trained Omnidata [5] model. With the sparse points from SfM, we use RANSAC to search for the optimal shift and scale for the monocular depth. We initialize the density restriction by voxelizing the space and labeling each center of a voxel within 20% of the projected monocular depth map, and then exclude sampling outside of labeled voxels. During training, we iterate through images, sample a bunch of patches per reference image, sample one virtual patch for each training patch, and evaluate the loss terms for all patches. We use NerfAcc [15] with modified QFF [14] as our base model for faster training and inference without loss of accuracy, and train with the unified loss $L = \sum \lambda_i L_i$, with $\lambda_{huber}, \lambda_{depth}, \lambda_{\nabla depth}, \lambda_{normal}, \lambda_{\nabla normal}, \lambda_{SSIM}, \lambda_{NCC}$ being 1.0, 0.05, 0.025, $1E - 3$, $5E - 4$, $1E - 4$, $1E - 4$. Please see our supplementary material for more details on parameters, model architecture, and training and inference time.

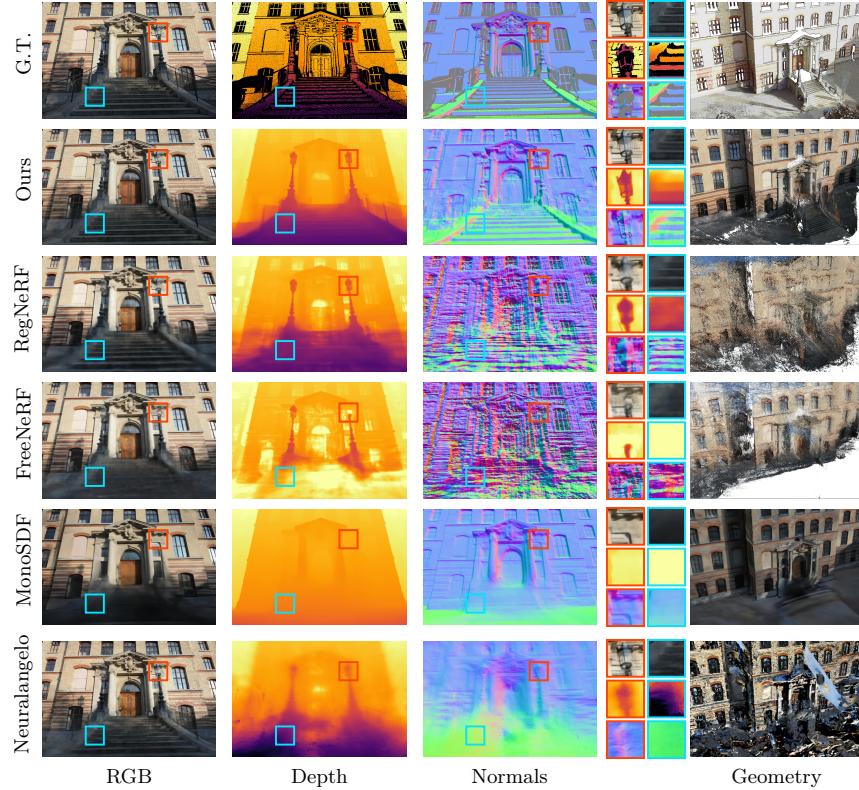


Fig. 5: Qualitative results on ETH3D [29]. We visualize the rendered RGB, depth and normal map of the test views and the complete geometry reconstruction on the *facade* of ETH3D [29] for our method and baselines [16, 24, 34, 38]. We zoom in on challenging areas such as lamps and stairs to highlight the difference. The depths of patches are re-normalized for visualization purposes. The geometry of MonoSDF and Neuralangelo is a mesh, and the geometry of other methods is a projected point cloud. Best viewed when zoomed in.

4 Experiments

Our experiments investigate: (1) the geometric accuracy of existing NeRF methods and our method on the challenging MVS benchmarks, as measured by point cloud metrics and view extrapolation; (2) how each of our contributions and system components affect performance.

We use the training scenes from the ETH3D High-Resolution dataset (ETH3D) [29] and selected scenes from TanksAndTemples (TnT) [11] as challenging large-scale natural scenes. ETH3D consists of 7 large-scale indoor scenes and 6 outdoor scenes, each sparsely capturing about 35 images on average, which is especially challenging for NeRF. In addition, we select large-scale indoor scenes (*Church*, *Meetingroom*) and outdoor scenes (*Barn*, *Courthouse*), as well as the advanced testing scenes used in the [38] from TnT to validate our method on densely captured scenes.



Fig. 6: Qualitative comparison of novel view images and meshes. We provide test view rendered images and meshes on the ETH3D dataset [29]. The mesh of Ours, RegNeRF [24] and FreeNeRF [34] are generated via TSDF fusion given predicted RGBD sequence. Best viewed when zoomed in.

Evaluation Protocols: We evaluate the methods on novel view synthesis and geometric inference. For novel view synthesis, we train the model with 90% of the images and treat the remaining 10% images as test views for evaluation. We report Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [39] as evaluation metrics. For geometric inference, we train on all images, and evaluate using the provided 3D geometry evaluation pipeline [11, 29] on point clouds. To generate point clouds, for radiance based method, we render the expected depth map of each training view, and fuse the depth maps with the scheme proposed by Galliani et al [6]; for SDF based methods, we render the mesh with SDF values using marching cube [17], and sample points from the rendered mesh. For TnT dataset, we only report the geometry inference as all views are densely captured.

Table 1: Quantitative evaluation on ETH3D [29]. We report baselines, our results with and without MVS depth based guidance, and reference MVS results on ETH3D [29]. We denote NVS as the model’s ability to perform novel-view synthesis, and the indoor and outdoor scenes in the ETH3D dataset as **In**, **Out**. The top rows show the baselines and our methods without using additional multi-view supervision, and the bottom rows show the reference MVS results and our method supervised with ACMMP [33] depth. We use author provided codes to evaluate the baselines, and ETH3D webpage provided results for MVS. We mark the top methods in blue and green. (□ best, □ second best)

Method	NVS	PSNR ↑ SSIM ↑ LPIPS ↓	Prec. _{2cm} ↑	Recall _{2cm} ↑	F-score _{2cm} ↑	F-score _{5cm} ↑
			Mean / In / Out	Mean / In / Out	Mean / In / Out	Mean / In / Out
RegNeRF [24]	✓	20.90 / 0.707	0.439 / 7.3 / 11.1 / 2.8	6.0 / 9.5 / 1.9	6.4 / 10.0 / 2.2	15.5 / 22.4 / 7.4
FreeNeRF [34]	✓	17.24 / 0.590	0.581 / 7.5 / 10.4 / 4.1	2.6 / 3.8 / 1.3	3.3 / 4.7 / 1.7	8.5 / 10.8 / 5.7
MonoSDF [38]	✓	18.85 / 0.679	0.498 / 25.2 / 26.2 / 24.0	19.3 / 28.5 / 8.5	20.1 / 26.9 / 12.1	41.1 / 45.2 / 36.4
Neuralangelo [16]	✓	19.53 / 0.696	0.414 / 3.3 / 3.4 / 3.2	2.1 / 3.4 / 0.6	2.3 / 3.4 / 1.0	7.2 / 8.0 / 6.2
Ours	✓	20.12 / 0.720	0.379 / 36.2 / 45.6 / 25.2	24.4 / 29.8 / 18.2	28.8 / 35.6 / 20.9	46.9 / 52.9 / 40.0
Ours (MVS-Depth)	✓	20.48 / 0.742	0.341 / 70.2 / 71.6 / 68.4	53.6 / 58.5 / 47.9	60.4 / 64.0 / 56.3	80.7 / 81.7 / 75.3
Gipuma [6]	✗	- / -	- / 86.5 / 89.3 / 83.2	24.9 / 24.6 / 25.3	36.4 / 35.8 / 37.1	49.2 / 47.1 / 51.7
COLMAP [28]	✗	- / -	- / 91.9 / 95.0 / 88.2	55.1 / 52.9 / 57.7	67.7 / 66.8 / 68.7	80.5 / 78.5 / 82.9
ACMMP [33]	✗	- / -	- / 90.6 / 92.4 / 88.6	77.6 / 79.6 / 75.3	83.4 / 85.3 / 81.3	92.0 / 92.2 / 91.9

Qualitative Results: Figure 5 shows the qualitative results on the ETH3D dataset. Compared to RegNeRF [24] and FreeNeRF [34], our approach provides noise-free, more accurate color, depth and normal maps. Compared to MonoSDF [38] and Neuralangelo [16], we produce high-frequency details (e.g., stairs, lamps) on appearance and geometry. Our fused point cloud shows that our NeRF geometry is more consistent and detailed across multiple views compared to the baselines. Figure 6 provides additional novel view rendered images and meshes among different scenes. Our method estimates clear appearance and accurate geometry, including texture-less, semi-transparent, or reflective surfaces (e.g., tables, glass doors, and windows) where RegNeRF [24] and FreeNeRF [34] fail. Our method is robust to scenes with challenging shape and initialization (*Relief_2*), while Neuralangelo [16] and original MonoSDF [38] hugely suffer from the setup. For better performance, we reduce the radius of initialized sphere surface (bias parameter) for MonoSDF and achieves much better results, but it still fails to extend to the far end of the gallery. See supplementary material for more comparisons and the local minimum of MonoSDF [38] with original parameters.

Comparison on ETH3D Dataset: We report the quantitative comparison against the baselines on Table 1. Our method achieves the best results for all metrics of geometry and novel view synthesis, except for the second best PSNR. RegNeRF [24] achieves better NVS results than other baselines, while predicting inaccurate geometry. MonoSDF [38] outperforms other baselines for geometry metrics, but the novel view rendering is of low quality. Neuralangelo [16] fails to generate high-quality reconstruction due to the sparsity of the views and not using geometric cues, but it achieves better novel view synthesis results than MonoSDF [38], which shows the advantage of using numerical gradients for hash grid to compute higher-order derivatives (surface normals). These results indicate that current approaches have difficulty rendering high quality images while maintaining correct geometry, and our method significantly outperforms

RegNeRF on geometry and MonoSDF on image rendering, showing that our method is effective for both purpose. For a better performance on ETH3D, we modify MonoSDF with a reduced bias term by the authors' suggestion. Due to GPU limitations, we use a smaller batch size for Neuralangelo than author-provided parameters, which may slightly reduce performance. We provide one comparison of meshes from the two parameters in supplementary material.

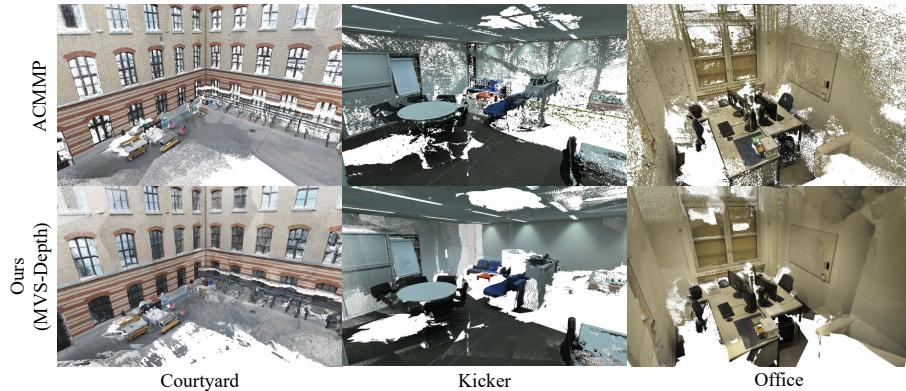


Fig. 7: Point cloud visualization. We visualize the point clouds of ACMMMP [33] and our method with MVS depth supervision on ETH3D [29]. Our method is able to complete textureless and reflective surfaces.

Integration of MVS Depth: We find the volumetric (NeRF) based methods provide less accurate geometry than the MVS based methods, especially on the sparse-view setup. Hence, we additionally use ACMMMP [33] inferred depth (denoted as \hat{d}_p^{mvs} for simplicity) to supervise our rendered depth with an additional L1 Loss $L_{mvs} = |\hat{d}_p^{mvs} - d_p|$ and weight $\lambda_{mvs} = 0.1$. We do not apply scale and shift for \hat{d}_p^{mvs} because MVS depth is metric depth. Marked as “Ours (MVS-Depth)” in Table 1, these additional MVS depths significantly boost the geometry and slightly improve the rendering. However, when supervised with MVS depth, the resulting geometry is less accurate than the MVS fused point cloud. We believe this is because NeRF methods have trouble fitting the MVS depth while

Table 2: Comparisons on TnT [11]. We report the F -score of each method on Tanks and Temples Dataset. * indicates results from original papers. For ease of comparison against Neuralangelo [16], we exclude scene *Church* which is not provided by the authors for comparison. *NeuralA.* refers to *Neuralangelo*. Advanced scene results are from the official online evaluation site. We mark the best scoring methods with **bold**.

	Scene	Ours	MonoSDF	NeuralA.*	NeuralWarp*	Scene	Ours	MonoSDF*	
Training	Meetingroom	22.0	27.2	32.0	8.0	Advanced	Auditorium	8.0	3.2
	Barn	49.4	6.0	70.0	22.0		Ballroom	26.6	3.7
	Courthouse	38.3	6.1	28.0	8.0		Courtroom	17.2	13.8
	Church	20.3	21.8	-	-		Museum	21.4	5.7
	Mean	36.6	13.1	43.3	12.7		Mean	18.3	6.5

Table 3: Ablations on ETH3D [29]. We report evaluations of different combinations of our components on the ETH3D dataset [29]. **Patch** denotes using patch-based training, **Mono.** denotes using monocular geometric cues, **Virtual** denotes using virtual view-based regularization, and **Restr.** denotes density restriction. We mark the best-performing combinations for each criterion in **bold**.

Patch	Mono.	Virtual	Restr.	PSNR ↑ SSIM ↑ LPIPS↓			F-score _{2cm} ↑			F-score _{5cm} ↑		
				Mean	In	Out	Mean	In	Out	Mean	In	Out
	✓			18.8	0.695	0.397	6.2 / 10.1 / 1.6	14.2 / 21.6 / 5.7				
		✓		19.6	0.695	0.393	6.7 / 11.4 / 1.3	15.1 / 24.3 / 4.4				
✓	✓		✓	18.2	0.618	0.484	10.3 / 15.0 / 4.8	23.1 / 30.1 / 15.0				
✓	✓	✓		20.0	0.723	0.388	8.5 / 11.2 / 5.3	18.3 / 20.8 / 15.5				
✓	✓	✓	✓	21.4	0.745	0.382	18.3 / 24.7 / 11.0	33.7 / 39.2 / 27.3				
✓	✓	✓	✓	20.9	0.742	0.372	23.1 / 29.5 / 15.7	38.7 / 43.8 / 32.7				
✓	✓	✓	✓	20.1	0.720	0.379	28.8 / 35.6 / 20.9	46.9 / 52.9 / 40.0				

Table 4: Ablations on TnT [11] dataset Training scenes. We show evaluations of different configurations on the selected TnT scenes. We report the *F*-score for each scene and average precision, recall and F-score for all scenes. We mark the best performing configuration in **bold** for each evaluation.

Patch	Mono.	Virtual	Restr.	Indoor			Outdoor			Mean		
				Church	Meetingroom	Barn	Courthouse	Prec.	Rec.	F-score	Prec.	Rec.
	✓			1.6	4.7	16.1	4.5	9.7	6.2	6.7		
		✓		1.7	7.7	22.9	6.2	10.2	10.4	9.6		
✓	✓		✓	10.2	10.6	24.7	16.4	15.2	17.7	15.5		
✓	✓	✓		7.8	13.0	35.7	11.9	15.0	22.2	17.1		
✓	✓	✓	✓	2.3	13.7	38.0	22.4	18.7	21.4	19.1		
✓	✓	✓	✓	20.3	22.0	49.4	38.3	28.4	38.6	32.5		

also minimizing photometric loss, as per-view photometric loss does not enforce correct geometry as well as MVS cross-view photometric consistency. See Figure 7 for qualitative comparison. We note that “Ours” without additional notation refers to our model with monocular cues but not MVS depth for the whole paper.

Comparison on TnT Dataset: In addition to the sparse views setup, we report the quantitative comparison against the baselines [4, 16, 38] on Table 2. Our method slightly underperforms Neuralangelo [16], indicating the strength of Neuralangelo for scenes with denser captures. We note that Neuralangelo initializes with ground truth points for scene scaling (and is sensitive to this initialization). However, in practice, ground truth is not available, so the scaling of our method is set based on automatically obtained SfM sparse points. In the advanced scenes, we significantly outperform MonoSDF on all four scenes, demonstrating our method’s capability on densely sampled large scale scenes.

Ablation Study: We ablate each component of our method in Tables 3 and 4. The tables show that including all of patch-based training, monocular cues, virtual view-based regularization, and density restriction greatly improves the geometry estimation. Table 3 indicates that monocular cues and the density restriction very lightly decrease the novel view synthesis quality, likely because they prevent the model from using erroneous geometry to create some view-dependent effects that it otherwise has trouble modeling. When using monocular supervision without patch-based training, loss for gradient of depth $L_{\nabla \text{depth}}$ and normals $L_{\nabla \text{normal}}$ are not applied as gradients are less accurate among randomly sampled pixels.

Limitations: Our method is more geometrically accurate than NeRF/SDF based methods for the tested scenes, but still falls short of MVS systems, even with the MVS supervision. Potential directions for improvements include: guided sampling of virtual view patches; joint inference of geometry with single-view predictions; including per-image terms to better handle lighting effects; expanding material models, e.g. with both diffuse and specular terms per point; and incorporating semantic segmentation. Further, geometrically accurate NeRF-based methods are slow compared to MVS systems, which is a barrier to research and practice.

5 Conclusion

We propose MonoPatchNeRF, a patch-based regularized NeRF model that aims to produce geometrically accurate models. We demonstrate the effective use of monocular geometry estimates with patch-based ray sampling optimization and density constraints, as well as the effectiveness of NCC and SSIM photometric consistency losses between patches from virtual and training views. Our method significantly improves geometric accuracy, ranking top in terms of F_1 , SSIM, and LPIPS compared to state-of-the-art regularized NeRF methods on the challenging ETH3D MVS benchmark.

Acknowledgement This work is supported in part by NSF IIS grants 2312102 and 2020227. SW is supported by NSF 2331878 and 2340254, and research grants from Intel, Amazon, and IBM. Thanks to Zhi-Hao Lin for sharing the code for pose interpolation and video generation.

References

1. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. International Journal of Computer Vision (IJCV) (2016) [4](#)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022) [8](#)
3. Cheng, X., Wang, P., Yang, R.: Learning depth with convolutional spatial propagation network. IEEE transactions on pattern analysis and machine intelligence **42**(10), 2361–2379 (2019) [4](#)
4. Darmon, F., Basclé, B., Devaux, J.C., Monasse, P., Aubry, M.: Improving neural implicit surfaces geometry with patch warping. In: CVPR. pp. 6250–6259 (2022) [4](#), [14](#)
5. Eftekhar, A., Sax, A., Bachmann, R., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: ICCV (2021) [9](#)
6. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 873–881 (2015) [4](#), [11](#), [12](#)
7. Guizilini, V.C., Vasiljevic, I., Fang, J., Ambrus, R., Zakharov, S., Sitzmann, V., Gaidon, A.: Delira: Self-supervised depth, light, and radiance fields. ICCV pp. 17889–17899 (2023) [4](#)
8. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) [6](#)
9. Hanley, D., Bretl, T.: An improved model-based observer for inertial navigation for quadrotors with low cost imus. In: to appear in AIAA Guidance, Navigation, and Control Conference (AIAA-GNC) (2016) [3](#)
10. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5885–5894 (2021) [2](#), [4](#)
11. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017) [4](#), [10](#), [11](#), [13](#), [14](#), [2](#), [3](#), [5](#), [6](#)
12. Kuhn, A., Sormann, C., Rossi, M., Erdler, O., Fraundorfer, F.: Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In: 2020 International Conference on 3D Vision (3DV). pp. 404–413. Ieee (2020) [3](#)
13. Lee, J.Y., DeGol, J., Zou, C., Hoiem, D.: Patchmatch-rl: Deep mvs with pixelwise depth, normal, and visibility. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6158–6167 (2021) [3](#)
14. Lee, J.Y., Wu, Y., Zou, C., Wang, S., Hoiem, D.: Qff: Quantized fourier features for neural field representations. arXiv preprint arXiv:2212.00914 (2022) [9](#), [1](#), [2](#)
15. Li, R., Tancik, M., Kanazawa, A.: Nerface: A general nerf acceleration toolbox. arXiv preprint arXiv:2210.04847 (2022) [9](#)
16. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [2](#), [5](#), [10](#), [12](#), [13](#), [14](#), [3](#), [4](#)
17. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH '87, Association for Computing Machinery, New York, NY, USA (1987). <https://doi.org/10.1145/37401.37422> [11](#)
18. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th international joint conference on Artificial intelligence. vol. 2, pp. 674–679 (1981) [3](#)

19. Ma, X., Gong, Y., Wang, Q., Huang, J., Chen, L., Yu, F.: Epp-mvsnet: Epipolar-assembling based depth prediction for multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5732–5740 (2021) [3](#)
20. Manivasagam, S., Wang, S., Wong, K., Zeng, W., Sazanovich, M., Tan, S., Yang, B., Ma, W.C., Urtasun, R.: Lidarsim: Realistic lidar simulation by leveraging the real world. In: CVPR (2020) [7](#)
21. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) (2019) [4](#)
22. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV. Springer, Cham (2020) [6](#)
23. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) [2](#), [4](#)
24. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5480–5490 (2022) [2](#), [4](#), [10](#), [11](#), [12](#)
25. Park, J., Joo, K., Hu, Z., Liu, C.K., So Kweon, I.: Non-local spatial propagation network for depth completion. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. pp. 120–136. Springer (2020) [4](#)
26. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021) [4](#)
27. Roessle, B., Barron, J.T., Mildenhall, B., Srinivasan, P.P., Nießner, M.: Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12892–12901 (2022) [4](#)
28. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016) [3](#), [4](#), [12](#)
29. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3260–3269 (2017) [2](#), [4](#), [10](#), [11](#), [12](#), [13](#), [14](#), [3](#), [7](#), [8](#)
30. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: CVPR. pp. 4190–4200 (2022) [4](#)
31. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5481–5490. IEEE (2022) [7](#)
32. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5610–5619 (2021) [4](#)
33. Xu, Q., Kong, W., Tao, W., Pollefeys, M.: Multi-scale geometric consistency guided and planar prior assisted multi-view stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022) [12](#), [13](#)
34. Yang, J., Pavone, M., Wang, Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8254–8263 (2023) [2](#), [4](#), [10](#), [11](#), [12](#)

35. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018) [3](#)
36. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5525–5534 (2019) [3](#)
37. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR (2021) [4](#)
38. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in neural information processing systems **35**, 25018–25032 (2022) [2](#), [4](#), [6](#), [10](#), [12](#), [14](#), [1](#)
39. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) [11](#)
40. Zhang, X., Bi, S., Sunkavalli, K., Su, H., Xu, Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) [2](#)

Supplementary Material

6 Training Details

Table 5: Network Architecture Details. We have a spatial feature extractor that uses implementation of QFF [14] to extract features for each point. The extracted features are passed into the Density MLP to obtain per-point density and geometric features of length 15. The extracted geometry features are passed into Color MLP (along with the direction) and Surface Normal MLP to extract color \mathbf{c} and surface normal \mathbf{n}_θ respectively.

Name	# Parameters	Input	Output Size
Spatial Features			
QFF [14]	32x80x80x80	$(x, y, z) \in \mathbb{R}^3$	32
Density MLP			
$D0_\theta$	32x16	QFF	16
$D1_\theta$	16x15	$ReLU(D0_\theta)$	15
$D2_\theta$	16x1	$ReLU(D0_\theta)$	$\sigma \in \mathbb{R}^1$
Color MLP			
$C0_\theta$	18x16	$D1_\theta + (\mathbf{d} \in \mathbb{R}^3)$	16
$C1_\theta$	16x3	$ReLU(C0_\theta)$	$\mathbf{c} \in \mathbb{R}^3$
Surface Normal MLP			
$S0_\theta$	15x16	$D1_\theta$	16
$S1_\theta$	16x3	$ReLU(S0_\theta)$	$\mathbf{n}_\theta \in \mathbb{R}^3$

Throughout our experiments, we use $\lambda_{huber}, \lambda_{depth}, \lambda_{\nabla depth}, \lambda_{normal}, \lambda_{\nabla normal}, \lambda_{SSIM}, \lambda_{NCC}, \lambda_{MVS}$ set to 1.0, 0.05, 0.025, 0.001, 0.0005, $1E - 4$, $1E - 4$, 0.1. We use L2 norm for L_{depth} and L1 norm for L_{normal} . We compute the novel view occlusion mask by comparing the angle difference between the rays from the training origin to the training unprojected pixels and the novel view unprojected pixels. Pixels with angles greater than 10 degrees are considered occluded, and are excluded when calculating the novel patches loss L_{novel} . We also provide the detailed network architecture in Table 5. Our model contains a total of 16,385,392 parameters and is trained for 50,000 steps on ETH3D in 2.25 hours and 200,000 steps on TanksandTemples in 9 hours using a single Nvidia A40 GPU. We compare our statistics with other baseline models in Table 6.

7 Baseline Training Details

MonoSDF: We show how varying the initialization bias [?] of MonoSDF [38] affects its reconstruction quality. We used the author provided configs of TnT on

Table 6: Training and Inference Time. We report the number of steps and time taken for optimization and rendering time for one image with resolution of 640,960 in ETH3D [29] training scenes. All timings are measured with the same device using single NVidia A40 GPU. The fast training and the rendering time comes from the architectural choice [14].

Model	# Steps	Training (hh:mm)	Inference (ss)
FreeNeRF [34]	200,000	14:26	36.4
RegNeRF [24]	200,000	07:00	30.4
MonoSDF [38]	200,000	20:54	136.0
Neuralangelo [16]	200,000	19:58	61.9
Ours	50,000	02:15	4.6

ETH3D, but found that MonoSDF suffer from local minimum in some challenging scenes with original bias parameters due to the scene scale. We therefore contacted the MonoSDF authors and were advised to use a small bias for initialization. Figure 8 compares the reconstruction of MonoSDF given different bias parameters in a challenging scene *relief_2* of ETH3D [29].

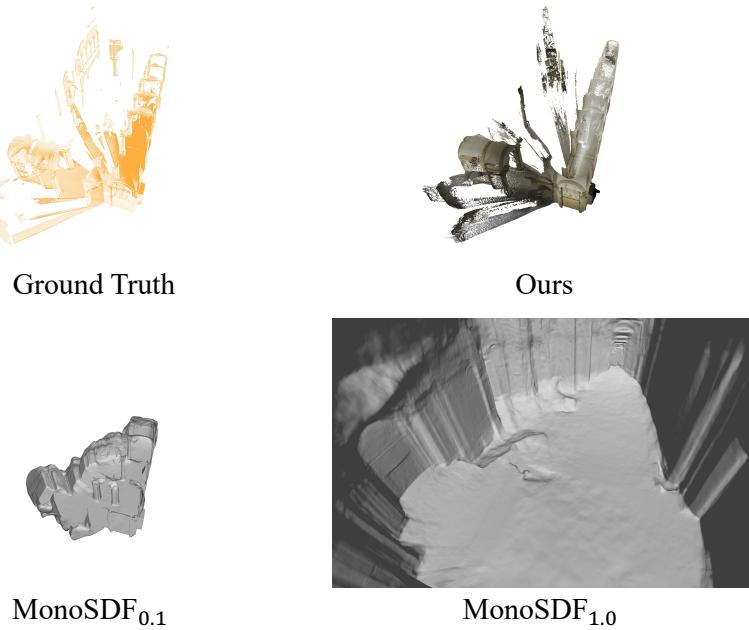


Fig. 8: Visualization of MonoSDF with different parameters in *Relief_2*. MonoSDF_{1.0} denotes MonoSDF trained with the default bias parameter (1.0) in the code provided by the authors on the large scale TnT [11] evaluation. MonoSDF_{0.1} denotes MonoSDF trained with the parameter suggested by the authors (0.1) for large-scale scenes specifically. MonoSDF [38] reconstructs better mesh given smaller bias, and falls into local minimum with original bias.

Neuralangelo: For Neuralangelo [16] experiment on ETH3D [29], we follow author provided setup on TanksAndTemples [11], but use a batch size of 4 instead of 16 to run on the same device settings. We additionally disable image embedding features, as we empirically found it to worsen the results. One visualization of results with different batch size is present in Figure 9. The $F\text{-score}_{2cm}$, $F\text{-score}_{5cm}$ of high and low batch size results are 1.53, 11.5 and 1.46, 11.8.

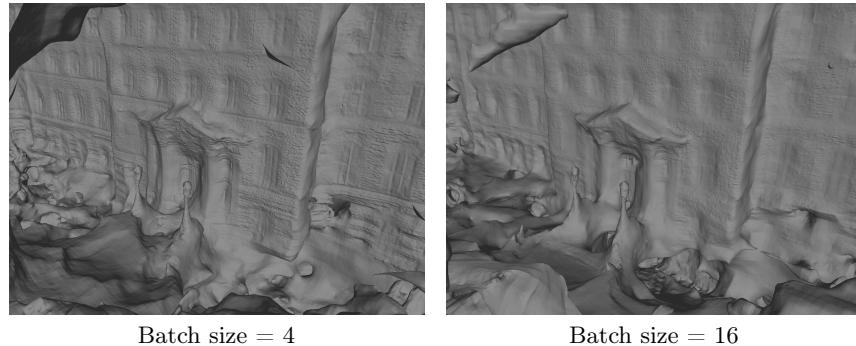


Fig. 9: Visualization of Neuralangelo with different batch size in *Facade*.

8 Foreground Fattening

In patch-based MVS, Foreground flattening can happen due to the plane-based propagation of depth candidates and a patch-wise planar assumption in computing photometric scores. Our method does not suffer from foreground flattening because we do not make planar assumptions (current depth estimates are projected into other views to compute photometric scores), and the rendering loss discourages such artifacts. Figure 10 shows the alignment of RGB image and the depth images. The image and the depth are aligned precisely, indicating that our method does not suffer from the foreground flattening.

9 Additional Qualitative Results

Images: Additional visualization of mesh and novel view synthesis is shown in Figure 11. We provide additional visualizations of our method on subsets of TanksAndTemples [11] advanced scenes in Figure 12 and Figure 13, and on ETH3D [29] in Figure 14 and Figure 15. All our results, except for ones annotated with (Ours-MVS-Depth) in Figure 7 and Table 1 of the main paper, are from our model with monocular cues.



Fig. 10: Image and Depth overlay visualization. From the left to right, we overlay the RGB image and the rendered depth map with varying fade thresholds. We show that our method does not experience foreground flattening as the images and the depths are precisely overlapped.

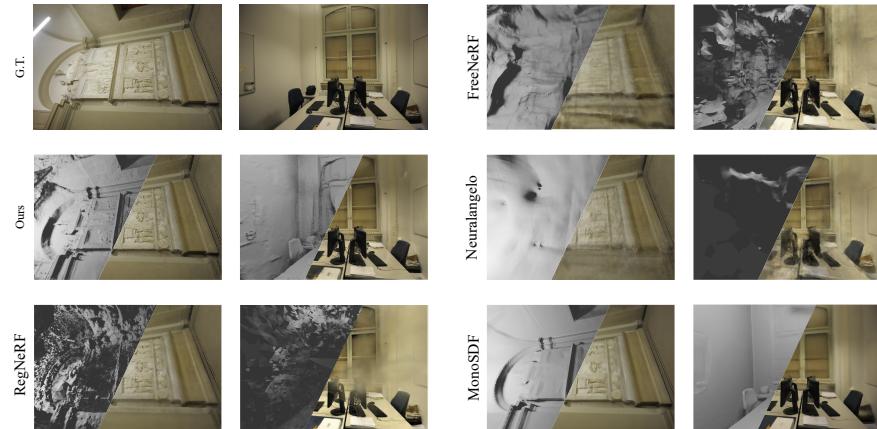


Fig. 11: Additioanl comparison of novel view images and meshes on ETH3D.
We provide additional comparisons with baselines [16, 24, 34, 38].



Fig. 12: Point clouds visualization for TnT advanced scenes [11]. We visualize interior and far-away views for point clouds to have a better visualization of the reconstructed geometry. Our method reconstructs complete and accurate point clouds.

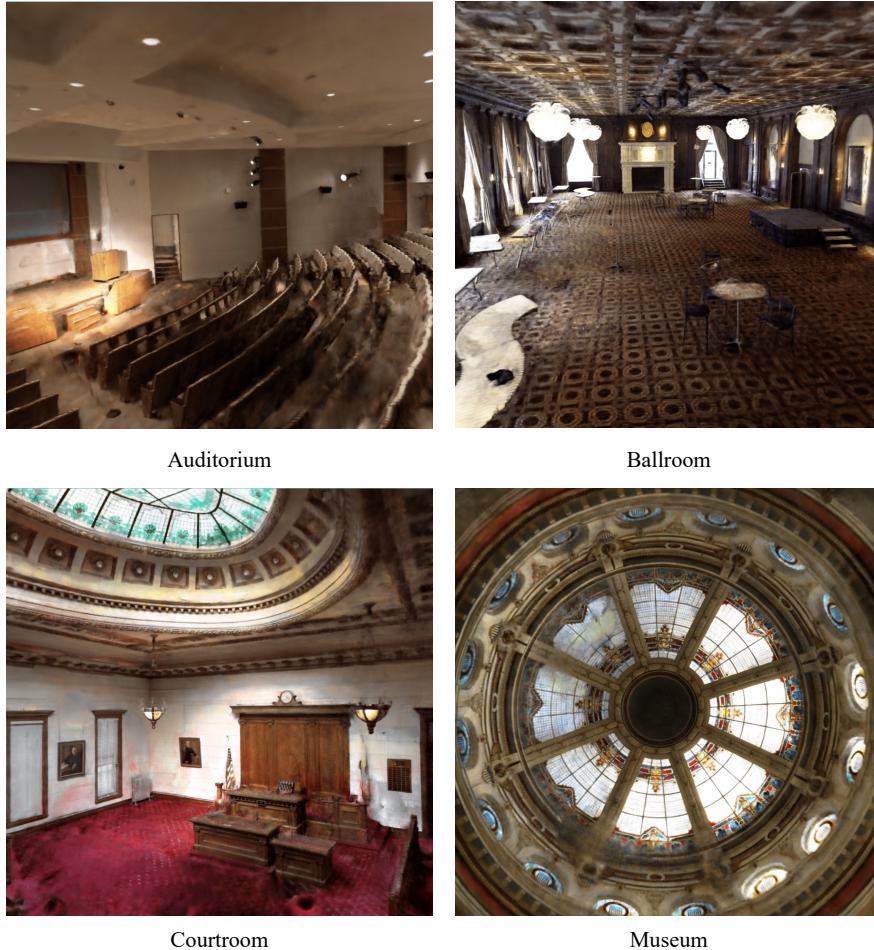


Fig. 13: Free-form views rendering for TnT advanced scenes [11].



Fig. 14: Point clouds visualization for ETH3D. We visualize point clouds for all scenes on ETH3D [29] except *Facade*. Our method reconstructs complete and accurate point clouds.

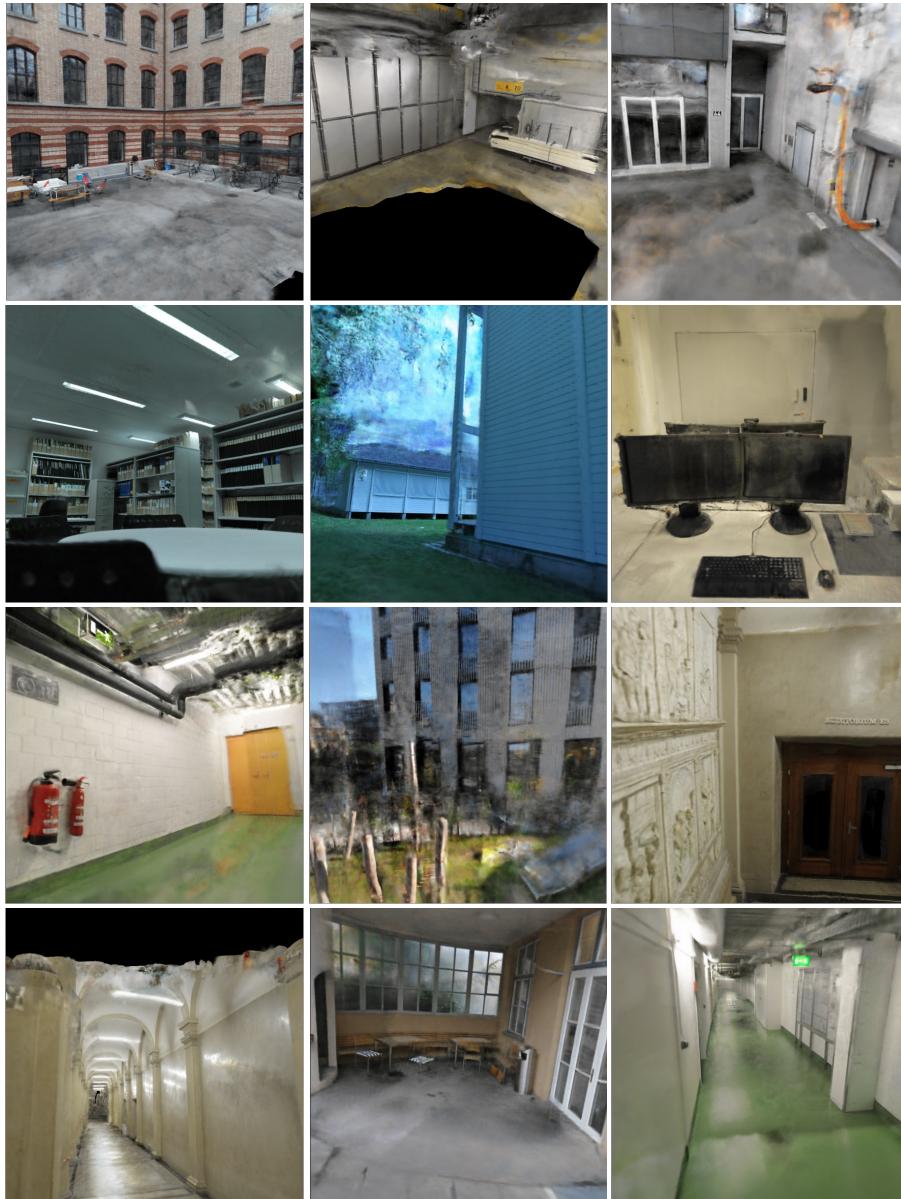


Fig. 15: Free-form views rendering for ETH3D [29]. We render novel views using our free-form viewer for each scenes in ETH3D except *Facade*. We show that our novel view rendering retains high-quality detailed textures in novel views.