This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Neuralangelo: High-Fidelity Neural Surface Reconstruction

Zhaoshuo Li[1,2] Thomas Müller[1] Alex Evans[1] Russell H. Taylor[2] Mathias Unberath[2]
Ming-Yu Liu[1] Chen-Hsuan Lin[1]

[1]NVIDIA Research [2]Johns Hopkins University

https://research.nvidia.com/labs/dir/neuralangelo

Figure 1. We present **Neuralangelo**, a framework for high-fidelity 3D surface reconstruction from RGB images using neural volume rendering, even without auxiliary data such as segmentation or depth. Shown in the figure is an extracted 3D mesh of a courthouse.

Abstract

Neural surface reconstruction has been shown to be powerful for recovering dense 3D surfaces via image-based neural rendering. However, current methods struggle to recover detailed structures of real-world scenes. To address the issue, we present ***Neuralangelo****, which combines the representation power of multi-resolution 3D hash grids with neural surface rendering. Two key ingredients enable our approach: (1) numerical gradients for computing higher-order derivatives as a smoothing operation and (2) coarse-to-fine optimization on the hash grids controlling different levels of details. Even without auxiliary inputs such as depth, Neuralangelo can effectively recover dense 3D surface structures from multi-view images with fidelity significantly surpassing previous methods, enabling detailed large-scale scene reconstruction from RGB video captures.*

1. Introduction

3D surface reconstruction aims to recover dense geometric scene structures from multiple images observed at different viewpoints [9]. The recovered surfaces provide structural information useful for many downstream applications, such as 3D asset generation for augmented/virtual/mixed reality or environment mapping for autonomous navigation of robotics. Photogrammetric surface reconstruction using a monocular RGB camera is of particular interest, as it equips users with the capability of casually creating digital twins of the real world using ubiquitous mobile devices.

Classically, multi-view stereo algorithms [6, 16, 29, 34] had been the method of choice for sparse 3D reconstruction. An inherent drawback of these algorithms, however, is their inability to handle ambiguous observations, *e.g.* regions with large areas of homogeneous colors, repetitive texture patterns, or strong color variations. This would result in inaccurate reconstructions with noisy or missing surfaces. Recently, neural surface reconstruction methods [36, 41, 42] have shown great potential in addressing these limitations. This new class of methods uses coordinate-based multi-layer perceptrons (MLPs) to represent the scene as an implicit function, such as occupancy fields [25] or signed distance functions (SDF) [36, 41, 42]. Leveraging the inherent continuity of MLPs and neural volume rendering [22], these techniques allow the optimized surfaces to meaningfully interpolate between spatial locations, resulting in smooth and complete surface representations.

Despite the superiority of neural surface reconstruction methods over classical approaches, the recovered fidelity of current methods does not scale well with the capacity of MLPs. Recently, Müller *et al*. [23] proposed a new scalable representation, referred to as Instant NGP (Neural Graphics Primitives). Instant NGP introduces a hybrid 3D grid structure with a multi-resolution hash encoding and a lightweight MLP that is more expressive with a memory footprint log-linear to the resolution. The proposed hybrid representation greatly increases the representation power of neural fields and has achieved great success at representing very fine-grained details for a wide variety of tasks, such as object shape representation and novel view synthesis problems.

In this paper, we propose **Neuralangelo** for high-fidelity surface reconstruction (Fig. 1). Neuralangelo adopts Instant NGP as a neural SDF representation of the underlying 3D scene, optimized from multi-view image observations via neural surface rendering [36]. We present two findings central to fully unlocking the potentials of multi-resolution hash encodings. First, using *numerical* gradients to compute higher-order derivatives, such as surface normals for the eikonal regularization [8, 12, 20, 42], is critical to stabilizing the optimization. Second, a *progressive* optimization schedule plays an important role in recovering the structures at different levels of details. We combine these two key ingredients and, via extensive experiments on standard benchmarks and real-world scenes, demonstrate significant improvements over image-based neural surface reconstruction methods in *both* reconstruction accuracy and view synthesis quality.

In summary, we present the following contributions:

- We present the Neuralangelo framework to naturally incorporate the representation power of multi-resolution hash encoding [23] into neural SDF representations.
- We present two simple techniques to improve the quality of hash-encoded surface reconstruction: higher-order derivatives with numerical gradients and coarse-to-fine optimization with a progressive level of details.
- We empirically demonstrate the effectiveness of Neuralangelo on various datasets, showing significant improvements over previous methods.

2. Related work

Multi-view surface reconstruction. Early image-based photogrammetry techniques use a volumetric occupancy grid to represent the scene [4, 16, 17, 29, 32]. Each voxel is visited and marked occupied if strict color constancy between the corresponding projected image pixels is satisfied. The photometric consistency assumption typically fails due to auto-exposure or non-Lambertian materials, which are ubiquitous in the real world. Relaxing such color constancy constraints across views is important for realistic 3D reconstruction.

Follow-up methods typically start with 3D point clouds from multi-view stereo techniques [6, 7, 28, 34] and then perform dense surface reconstruction [13, 14]. Reliance on the quality of the generated point clouds often leads to missing or noisy surfaces. Recent learning-based approaches augment the point cloud generation process with learned image features and cost volume construction [2, 10, 40]. However, these approaches are inherently limited by the resolution of the cost volume and fail to recover geometric details.

Neural Radiance Fields (NeRF). NeRF [22] achieves remarkable photorealistic view synthesis with view-dependent effects. NeRF encodes 3D scenes with an MLP mapping 3D spatial locations to color and volume density. These predictions are composited into pixel colors using neural volume rendering. A problem of NeRF and its variants [1, 30, 43, 46], however, is the question of how an isosurface of the volume density could be defined to represent the underlying 3D geometry. Current practice often relies on heuristic thresholding on the density values; due to insufficient constraints on the level sets, however, such surfaces are often noisy and may not model the scene structures accurately [36, 41]. Therefore, more direct modeling of surfaces is preferred for photogrammetric surface reconstruction problems.

Neural surface reconstruction. For scene representations with better-defined 3D surfaces, implicit functions such as occupancy grids [24, 25] or SDFs [42] are preferred over simple volume density fields. To integrate with neural volume rendering [22], different techniques [36, 41] have been proposed to reparametrize the underlying representations back to volume density. These designs of neural implicit functions enable more accurate surface prediction with view synthesis capabilities of unsacrificed quality [42].

Follow-up works extend the above approaches to real-time at the cost of surface fidelity [18, 37], while others [3, 5, 44] use auxiliary information to enhance the reconstruction results. Notably, NeuralWarp [3] uses patch warping given co-visibility information from structure-from-motion (S*f*M) to guide surface optimization, but the patch-wise planar assumption fails to capture highly-varying surfaces [3]. Other methods [5, 45] utilize sparse point clouds from S*f*M to supervise the SDF, but their performances are upper-bounded by the quality of the point clouds, as with classical approaches [45]. The use of monocular depth and segmentation as auxiliary data has also been explored with unconstrained image collections [31] or using scene representations with hash encodings [44]. In contrast, our work Neuralangelo builds upon hash encodings [23] to recover surfaces but *without* the need for auxiliary inputs used in prior work [3, 5, 31, 44, 45]. Concurrent work [38] also proposes coarse-to-fine optimization for improved surface details, where a displacement network corrects the shape predicted by a coarse network. In contrast, we use hierarchical hash grids and control the level of details based on our analysis of higher-order derivatives.

3. Approach

Neuralangelo reconstructs dense structures of the scene from multi-view images. Neuralangelo samples 3D locations along camera view directions and uses a multi-resolution hash encoding to encode the positions. The encoded features are input to an SDF MLP and a color MLP to composite images using SDF-based volume rendering.

3.1. Preliminaries

Neural volume rendering. NeRF [22] represents a 3D scene as volume density and color fields. Given a posed camera and a ray direction, the volume rendering scheme integrates the color radiance of sampled points along the ray. The i-th sampled 3D position $\mathbf{x}_i$ is at a distance t_i from the camera center. The volume density σ_i and color $\mathbf{c}_i$ of each sampled point are predicted using a coordinate MLP. The rendered color of a given pixel is approximated as the Riemann sum:

$$\hat{\mathbf{c}}(\mathbf{o}, \mathbf{d}) = \sum_{i=1}^{N} w_i \mathbf{c}_i, \quad \text{where } w_i = T_i \alpha_i. \tag{1}$$

Here, $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ is the opacity of the i-th ray segment, $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples, and $T_i = \Pi_{j=1}^{i-1}(1-\alpha_j)$ is the accumulated transmittance, indicating the fraction of light that reaches the camera. To supervise the network, a color loss is used between input images $\mathbf{c}$ and rendered images $\hat{\mathbf{c}}$:

$$\mathcal{L}_{\text{RGB}} = \|\hat{\mathbf{c}} - \mathbf{c}\|_1 . \tag{2}$$

However, surfaces are not clearly defined using such density formulation. Extracting surfaces from density-based representation often leads to noisy and unrealistic results [36, 41].

Volume rendering of SDF. One of the most common surface representations is SDF. The surface $\mathcal{S}$ of an SDF can be implicitly represented by its zero-level set, *i.e.*, $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\}$, where $f(\mathbf{x})$ is the SDF value. In the context of neural SDFs, Wang *et al*. [36] proposed to convert volume density predictions in NeRF to SDF representations with a logistic function to allow optimization with neural volume rendering. Given a 3D point $\mathbf{x}_i$ and SDF value $f(\mathbf{x}_i)$, the corresponding opacity value α_i used in Eq. 1 is computed as

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{x}_i)) - \Phi_s(f(\mathbf{x}_{i+1}))}{\Phi_s(f(\mathbf{x}_i))}, 0\right), \tag{3}$$

where Φ_s is the sigmoid function. In this work, we use the same SDF-based volume rendering formulation [36].

Multi-resolution hash encoding. Recently, multi-resolution hash encoding proposed by Müller *et al.* [23] has shown great scalability for neural scene representations, generating fine-grained details for tasks such as novel view synthesis. In Neuralangelo, we adopt the representation power of hash encoding to recover high-fidelity surfaces.

The hash encoding uses multi-resolution grids, with each grid cell corner mapped to a hash entry. Each hash entry stores the encoding feature. Let $\{V_1, ..., V_L\}$ be the set of different spatial grid resolutions. Given an input position $\mathbf{x}_i$, we map it to the corresponding position at each grid resolution V_l as $\mathbf{x}_{i,l} = \mathbf{x}_i \cdot V_l$. The feature vector $\gamma_l(\mathbf{x}_{i,l}) \in \mathbb{R}^c$ given resolution V_l is obtained via trilinear interpolation of hash entries at the grid cell corners. The encoding features across all spatial resolutions are concatenated together, forming a $\gamma(\mathbf{x}_i) \in \mathbb{R}^{cL}$ feature vector:

$$\gamma(\mathbf{x}_i) = \big(\gamma_1(\mathbf{x}_{i,1}), ..., \gamma_L(\mathbf{x}_{i,L})\big). \tag{4}$$

The encoded features are then passed to a shallow MLP.

One alternative to hash encoding is sparse voxel structures [30, 33, 39, 43], where each grid corner is uniquely defined without collision. However, volumetric feature grids require hierarchical spatial decomposition (*e.g.* octrees) to make the parameter count tractable; otherwise, the memory would grow cubically with spatial resolution. Given such hierarchy, finer voxel resolutions by design cannot recover surfaces that are misrepresented by the coarser resolutions [33]. Hash encoding instead assumes no spatial hierarchy and resolves collision automatically based on gradient averaging [23].

3.2. Numerical Gradient Computation

We show in this section that the analytical gradient w.r.t. position of hash encoding suffers from localities. Therefore, optimization updates only propagate to local hash grids, lacking non-local smoothness. We propose a simple fix to such a locality problem by using numerical gradients. An overview is shown in Fig. 2.

A special property of SDF is its differentiability with a gradient of the unit norm. The gradient of SDF satisfies the eikonal equation $\|\nabla f(\mathbf{x})\|_2 = 1$ (almost everywhere). To enforce the optimized neural representation to be a valid SDF, the eikonal loss [8] is typically imposed on the SDF predictions:

$$\mathcal{L}_{\text{eik}} = \frac{1}{N} \sum_{i=1}^{N} (\|\nabla f(\mathbf{x}_i)\|_2 - 1)^2, \tag{5}$$

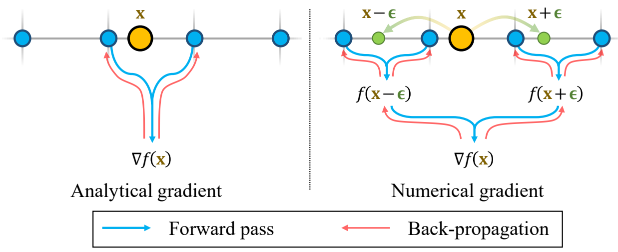

Figure 2. Using **numerical gradients** for higher-order derivatives distributes the back-propagation updates beyond the local hash grid cell, thus becoming a smoothed version of **analytical gradients**.

where N is the total number of sampled points. To allow for end-to-end optimization, a double backward operation on the SDF prediction $f(\mathbf{x})$ is required.

The *de facto* method for computing surface normals of SDFs $\nabla f(\mathbf{x})$ is to use analytical gradients [36, 41, 42]. Analytical gradients of hash encoding w.r.t. position, however, are *not* continuous across space under trilinear interpolation. To find the sampling location in a voxel grid, each 3D point $\mathbf{x}_i$ would first be scaled by the grid resolution V_l, written as $\mathbf{x}_{i,l} = \mathbf{x}_i \cdot V_l$. Let the coefficient for (tri-)linear interpolation be $\beta = \mathbf{x}_{i,l} - \lfloor \mathbf{x}_{i,l} \rfloor$. The resulting feature vectors are

$$\gamma_l(\mathbf{x}_{i,l}) = \gamma_l(\lfloor \mathbf{x}_{i,l} \rfloor) \cdot (1-\beta) + \gamma_l(\lceil \mathbf{x}_{i,l} \rceil) \cdot \beta, \quad (6)$$

where the rounded position $\lfloor \mathbf{x}_{i,l} \rfloor, \lceil \mathbf{x}_{i,l} \rceil$ correspond to the local grid cell corners. We note that rounding operations $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are non-differentiable. As a result, the derivative of hash encoding w.r.t. the position can be obtained as

$$\begin{aligned} \frac{\partial \gamma_l(\mathbf{x}_{i,l})}{\partial \mathbf{x}_i} &= \gamma_l(\lfloor \mathbf{x}_{i,l} \rfloor) \cdot (-\frac{\partial \beta}{\partial \mathbf{x}_{i,l}}) + \gamma_l(\lceil \mathbf{x}_{i,l} \rceil) \cdot \frac{\partial \beta}{\partial \mathbf{x}_{i,l}} \\ &= \gamma_l(\lfloor \mathbf{x}_{i,l} \rfloor) \cdot (-V_l) + \gamma_l(\lceil \mathbf{x}_{i,l} \rceil) \cdot V_l \,. \end{aligned} \quad (7)$$

The derivative of hash encoding is local, *i.e.*, when $\mathbf{x}_i$ moves across grid cell borders, the corresponding hash entries will be different. Therefore, the eikonal loss defined in Eq. 5 only back-propagates to the locally sampled hash entries, *i.e.* $\gamma_l(\lfloor \mathbf{x}_{i,l} \rfloor)$ and $\gamma_l(\lceil \mathbf{x}_{i,l} \rceil)$. When continuous surfaces (*e.g.* a flat wall) span multiple grid cells, these grid cells should produce coherent surface normals without sudden transitions. To ensure consistency in surface representation, joint optimization of these grid cells is desirable. However, the analytical gradient is limited to local grid cells, unless all corresponding grid cells happen to be sampled and optimized simultaneously. Such sampling is not always guaranteed.

To overcome the locality of the analytical gradient of hash encoding, we propose to compute the surface normals using *numerical* gradients. If the step size of the numerical gradient is smaller than the grid size of hash encoding, the numerical gradient would be equivalent to the analytical gradient; otherwise, hash entries of multiple grid cells would participate in the surface normal computation. Back-propagating through the surface normals thus allows hash entries of multiple grids to receive optimization updates simultaneously. Intuitively, numerical gradients with carefully chosen step sizes can be interpreted as a smoothing operation on the analytical gradient expression. An alternative of normal supervision is a teacher-student curriculum [35, 47], where the predicted noisy normals are driven towards MLP outputs to exploit the smoothness of MLPs. However, analytical gradients from such teacher-student losses still only back-propagate to local grid cells for hash encoding. In contrast, numerical gradients solve the locality issue without the need of additional networks.

To compute the surface normals using the numerical gradient, additional SDF samples are needed. Given a sampled point $\mathbf{x}_i = (x_i, y_i, z_i)$, we additionally sample two points along each axis of the canonical coordinate around x_i within a vicinity of a step size of ϵ. For example, the x-component of the surface normal can be found as

$$\nabla_x f(\mathbf{x}_i) = \frac{f\left(\gamma(\mathbf{x}_i + \boldsymbol{\epsilon}_x)\right) - f\left(\gamma(\mathbf{x}_i - \boldsymbol{\epsilon}_x)\right)}{2\epsilon} \,, \quad (8)$$

where $\boldsymbol{\epsilon}_x = [\epsilon, 0, 0]$. In total, six additional SDF samples are required for numerical surface normal computation.

3.3. Progressive Levels of Details

Coarse-to-fine optimization can better shape the loss landscape to avoid falling into false local minima. Such a strategy has found many applications in computer vision, such as image-based registration [19, 21, 26]. Neuralangelo also adopts a coarse-to-fine optimization scheme to reconstruct the surfaces with progressive levels of details. Using numerical gradients for the higher-order derivatives naturally enables Neuralangelo to perform coarse-to-fine optimization from two perspectives.

Step size ϵ. As previously discussed, numerical gradients can be interpreted as a smoothing operation where the step size ϵ controls the resolution and the amount of recovered details. Imposing $\mathcal{L}_{\text{eik}}$ with a larger ϵ for numerical surface normal computation ensures the surface normal is consistent at a larger scale, thus producing consistent and continuous surfaces. On the other hand, imposing $\mathcal{L}_{\text{eik}}$ with a smaller ϵ affects a smaller region and avoids smoothing details. In practice, we initialize the step size ϵ to the coarsest hash grid size and exponentially decrease it matching different hash grid sizes throughout the optimization process.

Hash grid resolution V. If all hash grids are activated from the start of the optimization, to capture geometric details, fine hash grids must first "unlearn" from the coarse optimization with large step size ϵ and "relearn" with a smaller ϵ. If such a process is unsuccessful due to converged optimization, geometric details would be lost. Therefore, we only enable an initial set of coarse hash grids and progressively activate

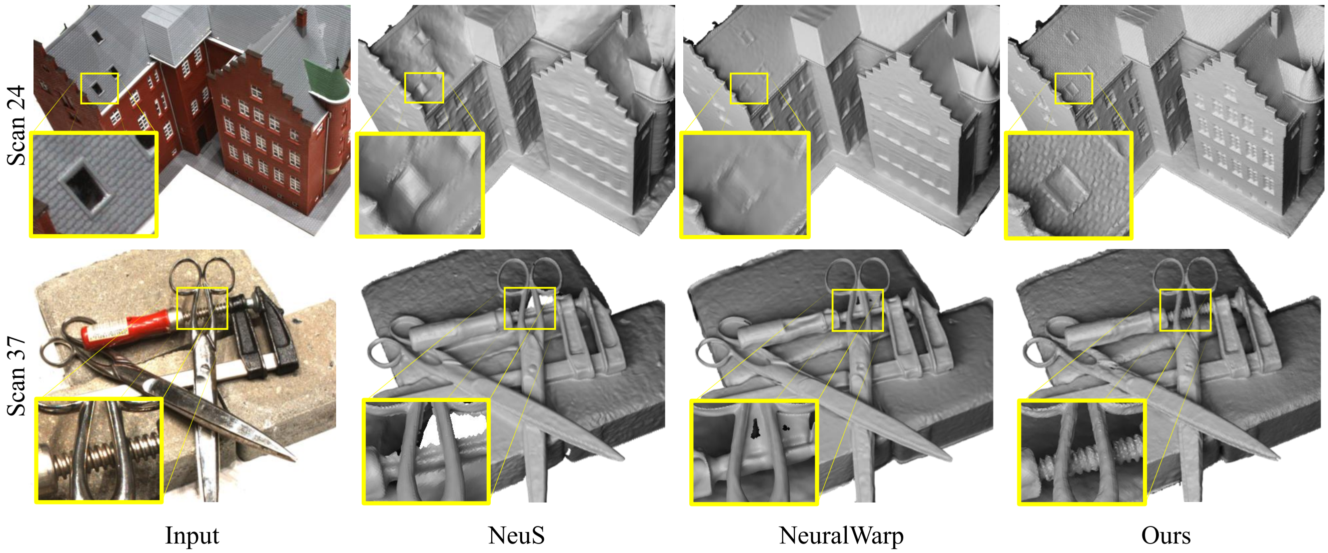

Figure 3. **Qualitative comparison on the DTU benchmark [11].** Neuralangelo produces more accurate and higher-fidelity surfaces.

finer hash grids throughout optimization when ϵ decreases to their spatial size. The relearning process can thus be avoided to better capture the details. In practice, we also apply weight decay over all parameters to avoid single-resolution features dominating the final results.

3.4. Optimization

To further encourage the smoothness of the reconstructed surfaces, we impose a prior by regularizing the mean curvature of SDF. The mean curvature is computed from discrete Laplacian similar to the surface normal computation, otherwise, the second-order analytical gradients of hash encoding are zero everywhere when using trilinear interpolation. The curvature loss $\mathcal{L}_{\text{curv}}$ is defined as:

$$\mathcal{L}_{\text{curv}} = \frac{1}{N} \sum_{i=1}^{N} \left| \nabla^2 f(\mathbf{x}_i) \right| . \tag{9}$$

We note that the samples used for the surface normal computation in Eq. 8 are sufficient for curvature computation.

The total loss is defined as the weighted sum of losses:

$$\mathcal{L} = \mathcal{L}_{\text{RGB}} + w_{\text{eik}} \mathcal{L}_{\text{eik}} + w_{\text{curv}} \mathcal{L}_{\text{curv}} . \tag{10}$$

All network parameters, including MLPs and hash encoding, are trained jointly end-to-end.

4. Experiments

Datasets. Following prior work, we conduct experiments on 15 object-centric scenes of the DTU dataset [11]. Each scene has 49 or 64 images captured by a robot-held monocular RGB camera. The ground truth is obtained from a structured-light scanner. We further conduct experiments on 6 scenes of the Tanks and Temples dataset [15], including large-scale indoor/outdoor scenes. Each scene contains 263 to 1107 images captured using a hand-held monocular RGB camera. The ground truth is obtained using a LiDAR sensor.

Implementation details. Our hash encoding resolution spans 2^5 to 2^{11} with 16 levels. Each hash entry has a channel size of 8. The maximum number of hash entries of each resolution is 2^{22}. We activate 4 and 8 hash resolutions at the beginning of optimization for DTU dataset and Tanks and Temples respectively, due to differences in scene scales. We enable a new hash resolution every 5000 iterations when the step size ϵ equals its grid cell size. For all experiments, we do *not* utilize auxiliary data such as segmentation or depth during the optimization process.

Evaluation criteria. We report Chamfer distance and F1 score for surface evaluation [11, 15]. We use peak signal-to-noise ratio (PSNR) to report image synthesis qualities.

4.1. DTU Benchmark

We show qualitative results in Fig. 3 and quantitative results in Table 1. On average, Neuralangelo achieves the lowest Chamfer distance and the highest PSNR, even without using auxiliary inputs. The result suggests that Neuralangelo is more generally applicable than prior work when recovering surfaces and synthesizing images, despite not performing best in every individual scene.

We further ablate Neuralangelo against the following conditions: 1) AG: analytical gradients, 2) AG+P: analytical gradients and progressive activating hash resolutions, 3) NG: numerical gradients with varying ϵ. Fig. 4 shows the results qualitatively. AG produces noisy surfaces, even with hash resolutions progressively activated (AG+P). NG improves

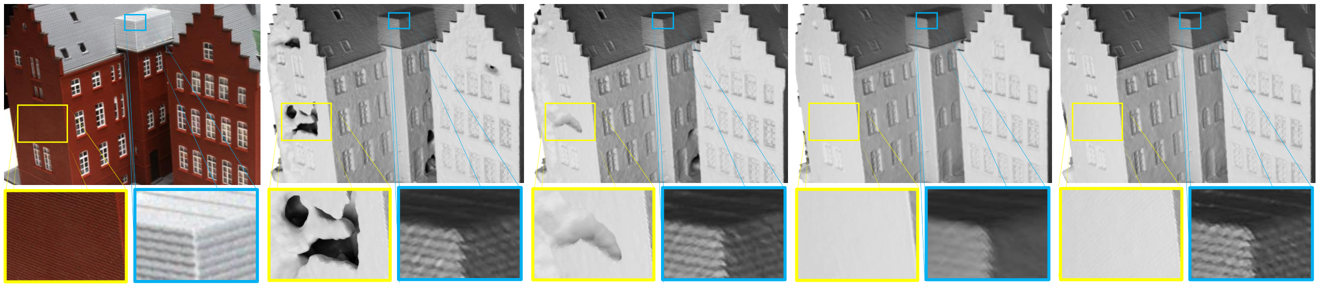

Figure 4. **Qualitative comparison of different coarse-to-fine optimization scheme.** When using the analytical gradient (AG and AG+P), coarse surfaces often contain artifacts. While using numerical gradients (NG) leads to a better coarse shape, details are also smoothed. Our solution (NG+P) produces both smooth surfaces and fine details.

		24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean
Chamfer distance (mm) ↓	NeRF [22]	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49
	VolSDF [41]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86
	NeuS [36]	1.00	1.37	0.93	0.43	1.10	0.65	**0.57**	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84
	HF-NeuS [38]	0.76	1.32	0.70	0.39	1.06	0.63	0.63	**1.15**	1.12	0.80	0.52	1.22	0.33	0.49	0.50	0.77
	RegSDF [45] †	0.60	1.41	0.64	0.43	1.34	**0.62**	0.60	**0.90**	**0.92**	1.02	0.60	**0.59**	**0.30**	**0.41**	**0.39**	0.72
	NeuralWarp [3] †	0.49	**0.71**	**0.38**	0.38	**0.79**	0.81	0.82	1.20	1.06	**0.68**	0.66	**0.74**	0.41	0.63	0.51	0.68
	AG	0.67	1.04	0.84	0.39	1.43	1.23	1.11	1.24	1.54	0.85	0.50	1.01	0.37	0.51	0.44	0.88
	AG+P	0.59	0.95	0.46	**0.34**	1.19	0.70	0.79	1.19	1.37	**0.69**	**0.49**	0.93	0.33	**0.44**	0.44	0.73
	NG	**0.48**	0.81	0.43	**0.35**	0.89	0.71	0.61	1.26	1.06	0.74	**0.47**	0.79	0.33	0.45	**0.43**	**0.65**
	NG+P (Ours)	**0.37**	**0.72**	**0.35**	**0.35**	**0.87**	**0.54**	**0.53**	1.29	**0.97**	0.73	**0.47**	**0.74**	**0.32**	**0.41**	**0.43**	**0.61**
PSNR ↑	RegSDF [45] †	24.78	23.06	23.47	22.21	28.57	25.53	21.81	28.89	26.81	27.91	24.71	25.13	26.84	21.67	28.25	25.31
	NeuS [36]	26.62	23.64	26.43	25.59	30.61	**32.83**	29.24	33.71	26.85	31.97	32.18	28.92	28.41	35.00	34.81	29.79
	VolSDF [41]	26.28	25.61	26.55	26.76	31.57	31.50	29.38	33.23	28.03	32.13	33.16	31.49	30.33	34.90	34.75	30.38
	NeRF [22]	26.24	**25.74**	26.79	27.57	**31.96**	31.50	29.58	32.78	**28.35**	32.08	33.49	**31.54**	**31.00**	35.59	35.51	30.65
	AG	29.97	24.98	23.11	30.27	30.60	31.27	29.27	34.22	27.47	33.09	33.85	29.98	29.41	35.69	35.11	30.55
	AG+P	30.12	24.63	29.59	30.29	31.60	32.04	**29.85**	34.19	27.82	33.23	33.95	29.15	29.44	35.99	35.67	31.17
	NG	**30.34**	25.14	**30.20**	**30.79**	31.72	31.86	29.81	**34.36**	28.01	**33.45**	**34.38**	30.39	29.88	**36.02**	**35.74**	**31.47**
	NG+P (Ours)	**30.64**	**27.78**	**32.70**	**34.18**	**35.15**	**35.89**	**31.47**	**36.82**	**30.13**	**35.92**	**36.61**	**32.60**	**31.20**	**38.41**	**38.05**	**33.84**

Table 1. **Quantitative results on DTU dataset [11].** Neuralangelo achieves the best reconstruction accuracy and image synthesis quality. **Best result**. **Second best result**. † Requires 3D points from S*f*M. Best viewed in color.

the smoothness of the surface, sacrificing details. Our setup (NG+P) produces both smooth surfaces and fine details.

4.2. Tanks and Temples

As no public result is available for Tanks and Temples, we train NeuS [36] and NeuralWarp [3] following our setup. We also report classical multi-view stereo results using COLMAP [27]. As COLMAP and NeuralWarp do not support view synthesis, we only report PSNR from NeuS. Results are summarized in Fig. 5 and Table 2.

Neuralangelo achieves the highest average PSNR and performs best in terms of F1 score. Comparing against NeuS [36], we can recover high-fidelity surfaces with intricate details. We find that the dense surfaces generated from COLMAP are sensitive to outliers in the sparse point cloud. We also find that NeuralWarp often predicts surfaces for the sky and backgrounds potentially due to their color rendering scheme following VolSDF [41]. The additional surfaces predicted for backgrounds are counted as outliers and worsen F1 scores significantly. We instead follow NeuS [36] and use an additional network [46] to model the background.

Similar to the DTU results, using the analytical gradient produces noisy surfaces and thus leads to a low F1 score. We further note that the reconstruction of Courthouse shown in Figs. 1 and 5 are the same building of different sides, demonstrating the capability of Neuralangelo for large-scale granular reconstruction.

4.3. Level of Details

As Neuralangelo progressively optimizes the hash features of increasing resolution, we inspect the progressive level of details similar to NGLOD [33]. We show a qualitative visualization in Fig. 6. While some surfaces are entirely missed by coarse levels, for example, the tree, table, and

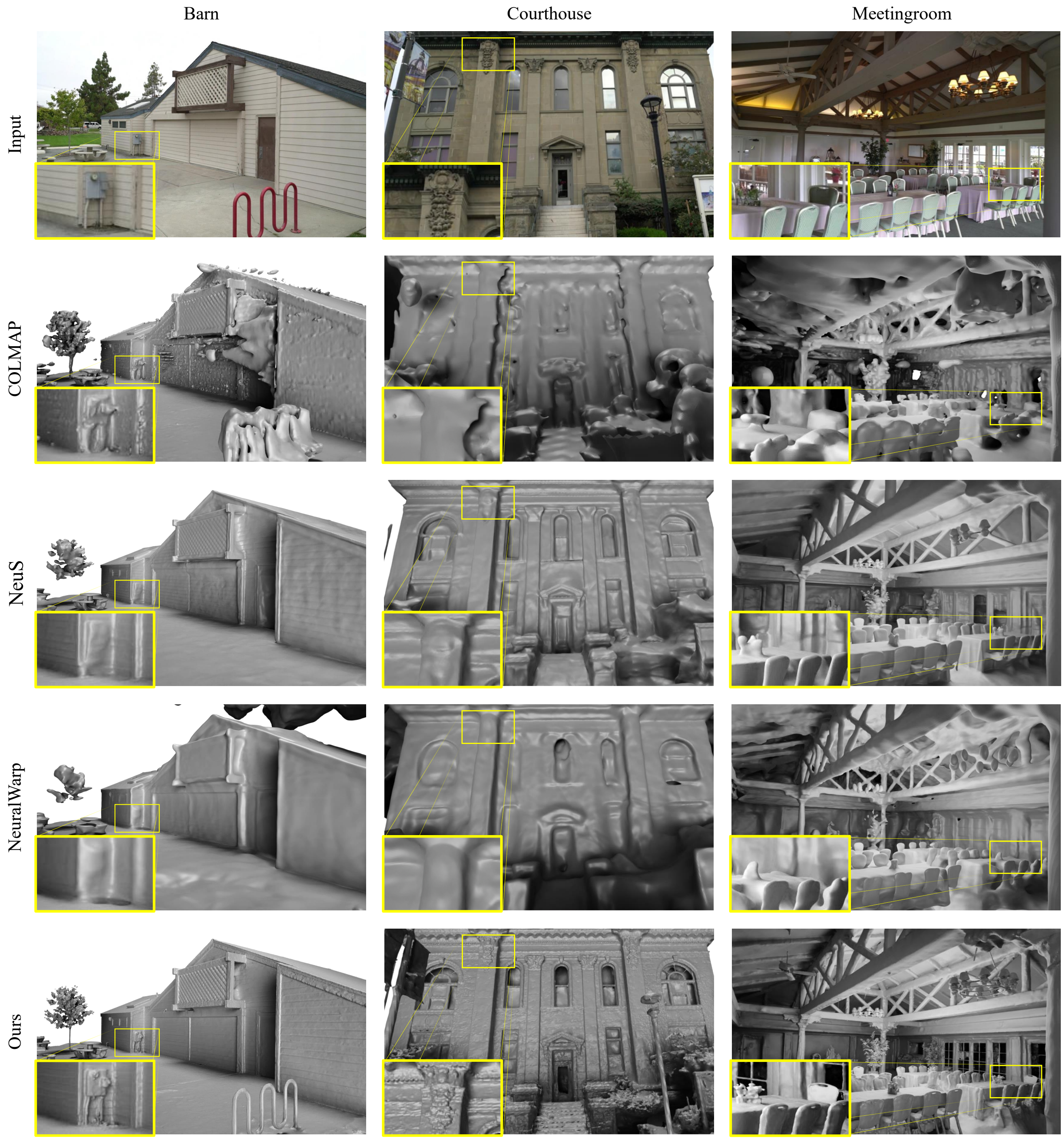

Figure 5. **Qualitative comparison on Tanks and Temples dataset [15].** Neuralangelo captures the scene details better compared to other baseline approaches, while baseline approaches have missing or noisy surfaces.

bike rack, these structures are recovered by finer resolutions successfully. The ability to recover missing surfaces demonstrates the advantages of our spatial hierarchy-free design.

Moreover, we note that flat surfaces are predicted at sufficiently high resolutions (around Level 8 in this example). Thus, only relying on the continuity of local cells of coarse resolutions is not sufficient to reconstruct large continuous surfaces. The result motivates the use of the numerical gradients for the higher-order derivatives, such that back-propagation is beyond local grid cells.

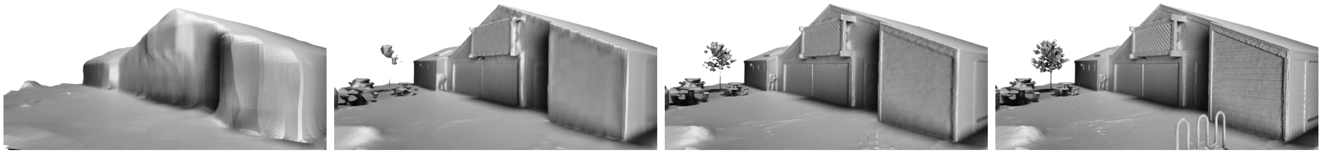

Figure 6. **Results at different hash resolutions.** While some structures, such as the tree, table, and bike rack, are missed at coarse resolutions (Level 4). Finer resolutions can progressively recover these missing surfaces. Flat continuous surfaces also require sufficiently fine resolutions to predict (Level 8). The result motivates the non-local updates when using numerical gradients for higher-order derivatives.

	F1 Score ↑							PSNR ↑				
	NeuralWarp [3]	COLMAP [27]	NeuS [36]	AG	AG+P	NG	NG+P (Ours)	NeuS [36]	AG	AG+P	NG	NG+P (Ours)
Barn	0.22	0.55	0.29	0.22	0.31	**0.63**	**0.70**	26.36	**26.91**	26.69	26.14	**28.57**
Caterpillar	0.18	0.01	0.29	0.23	0.24	**0.30**	**0.36**	25.21	**26.04**	25.12	26.16	**27.81**
Courthouse	0.08	0.11	0.17	0.08	0.09	**0.24**	**0.28**	23.55	25.43	**25.63**	25.06	**27.23**
Ignatius	0.02	0.22	0.83	0.72	0.73	**0.85**	**0.89**	**23.27**	22.69	22.73	**23.78**	**23.67**
Meetingroom	0.08	0.19	0.24	0.04	0.05	**0.27**	**0.32**	25.38	**28.13**	28.05	27.44	**30.70**
Truck	0.35	0.19	**0.45**	0.33	0.37	0.44	**0.48**	23.71	23.89	**23.95**	22.99	**25.43**
Mean	0.15	0.21	0.38	0.27	0.30	**0.45**	**0.50**	24.58	**25.51**	25.36	25.26	**27.24**

Table 2. **Quantitative results on Tanks and Temples dataset [15].** Neuralangelo achieves the best surface reconstruction quality and performs best on average in terms of image synthesis. **Best result**. **Second best result.** Best viewed in color.

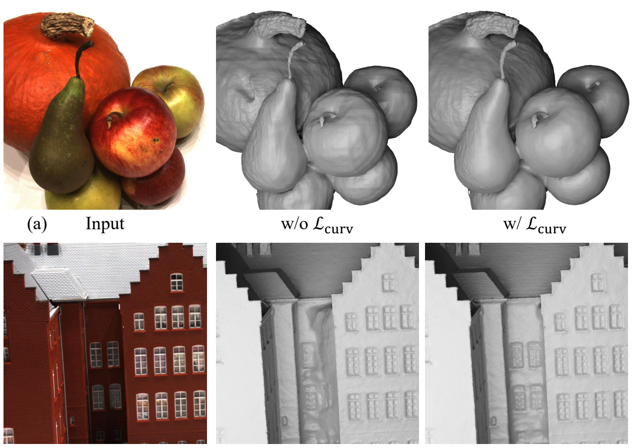

Figure 7. **Ablation results.** (a) Surface smoothness improves with curvature regularization $\mathcal{L}_{\text{curv}}$. (c) Concave shapes are better formed with topology warmup.

4.4. Ablations

Curvature regularization. We ablate the necessity of curvature regularization in Neuralangelo and compare the results in Fig. 7(a). Intuitively, $\mathcal{L}_{\text{curv}}$ acts as a smoothness prior by minimizing surface curvatures. Without $\mathcal{L}_{\text{curv}}$, we find that the surfaces tend to have undesirable sharp transitions. By using $\mathcal{L}_{\text{curv}}$, the surface noises are removed.

Topology warmup. We follow prior work and initialize the SDF approximately as a sphere [42]. With an initial spherical shape, using $\mathcal{L}_{\text{curv}}$ also makes concave shapes difficult to form because $\mathcal{L}_{\text{curv}}$ preserves topology by preventing singularities in curvature. Thus, instead of applying $\mathcal{L}_{\text{curv}}$ from the beginning of the optimization process, we use a short warmup period that linearly increases the curvature loss strength. We find this strategy particularly helpful for concave regions, as shown in Fig. 7(b).

5. Conclusion

We introduce Neuralangelo, an approach for photogrammetric neural surface reconstruction. The findings of Neuralangelo are simple yet effective: using numerical gradients for higher-order derivatives and a coarse-to-fine optimization strategy. Neuralangelo unlocks the representation power of multi-resolution hash encoding for neural surface reconstruction modeled as SDF. We show that Neuralangelo effectively recovers dense scene structures of both object-centric captures and large-scale indoor/outdoor scenes with extremely high fidelity, enabling detailed large-scale scene reconstruction from RGB videos. Our method currently samples pixels from images randomly without tracking their statistics and errors. Therefore, we use long training iterations to reduce the stochastics and ensure sufficient sampling of details. It is our future work to explore a more efficient sampling strategy to accelerate the training process.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[2] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1538–1547, 2019. 2

[3] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022. 2, 3, 6, 8

[4] Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 2, 1999. 2

[5] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *arXiv preprint arXiv:2205.15848*, 2022. 2, 3

[6] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 1, 2

[7] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015. 2

[8] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 2, 3

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[10] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[11] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 5, 6

[12] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1251–1261, 2020. 2

[13] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[14] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 2

[15] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 5, 7, 8

[16] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000. 1, 2

[17] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2):150–162, 1994. 2

[18] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2

[19] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 4

[20] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. Sdf-srn: Learning signed distance 3d object reconstruction from static images. *Advances in Neural Information Processing Systems*, 33:11453–11464, 2020. 2

[21] Bruce D Lucas, Takeo Kanade, et al. *An iterative image registration technique with an application to stereo vision*, volume 81. Vancouver, 1981. 4

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 2, 3, 6

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 2, 3

[24] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[25] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[26] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 4

[27] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6, 8

[28] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision*, pages 501–518. Springer, 2016. 2

[29] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999. 1, 2

[30] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2, 3

[31] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3d reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 3

[32] Richard Szeliski. Rapid octree construction from image sequences. *CVGIP: Image understanding*, 58(1):23–32, 1993. 2

[33] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 3, 6

[34] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012. 1, 2

[35] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 4

[36] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2, 3, 4, 6, 8

[37] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. *arXiv preprint arXiv:2212.05231*, 2022. 2

[38] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Hf-neus: Improved surface reconstruction using high-frequency details. In *Advances in Neural Information Processing Systems*. 3, 6

[39] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. *arXiv preprint arXiv:2208.12697*, 2022. 3

[40] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2

[41] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2, 3, 4, 6

[42] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 2, 4, 8

[43] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021. 2, 3

[44] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022. 2, 3

[45] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Critical regularizations for neural surface reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6270–6279, 2022. 2, 3, 6

[46] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2, 6

[47] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 4