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.

OmniSDF: Scene Reconstruction using Omnidirectional Signed Distance Functions and Adaptive Binoctrees

Hakyeong Kim KAIST
Andreas Meuleman INRIA
Hyeonjoong Jang KAIST
James Tompkin Brown University
Min H. Kim KAIST

Abstract

We present a method to reconstruct indoor and outdoor static scene geometry and appearance from an omnidirectional video moving in a small circular sweep. This setting is challenging because of the small baseline and large depth ranges, making it difficult to find ray crossings. To better constrain the optimization, we estimate geometry as a signed distance field within a spherical binoctree data structure and use a complementary efficient tree traversal strategy based on a breadth-first search for sampling. Unlike regular grids or trees, the shape of this structure well-matches the camera setting, creating a better memory-quality trade-off. From an initial depth estimate, the binoctree is adaptively subdivided throughout the optimization; previous methods use a fixed depth that leaves the scene undersampled. In comparison with three neural optimization methods and two non-neural methods, ours shows decreased geometry error on average, especially in a detailed scene, while significantly reducing the required number of voxels to represent such details.

1. Introduction

When reconstructing the geometry of a static unbound scene, say an outdoor space, most image-based multi-view reconstruction pipelines use perspective cameras. Given their limited field of view, they suffer from high data acquisition costs as many images must be captured and calibrated. But, given their physically-compact size, it is possible to move to different spatial sampling positions and capture many light ray crossings from which to accurately constrain the surface geometry location. To solve for the location, modern auto-differentiation systems let us flexibly implement algorithms like sphere tracing with neural signed-distance functions (SDFs).

Given enough ray crossings, these algorithms show promise in capturing geometry from real-world perspective images, both in geometry detail [15] and in large-scale structures [24], thanks to the robust fitting properties of neural networks. If memory is not a concern, we can exploit voxel grid structures for rapid position indexing [15, 24], but unbound scenes require careful memory consideration.

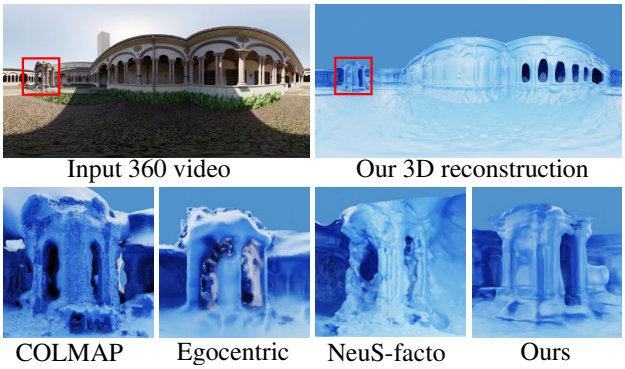

Figure 1. We introduce a memory-efficient neural 3D reconstruction method tailored to work with short egocentric omnidirectional video inputs. The geometry is estimated using a signed distance field and a novel adaptive spherical binoctree data structure subdivided through iterative optimization. We show that our method outperforms other state-of-the-art 3D reconstruction methods in balancing detail and memory cost [8, 21, 33].

To help overcome the acquisition cost, one alternative is to use an omnidirectional camera that captures a view of all angles of the surroundings. This is a critical capability when comprehensive spatial understanding is necessary, such as in virtual reality and robotic navigation. But, now a wider field of view is represented within a limited camera sensor resolution, such that the angular extent of each pixel is typically greater than in a perspective camera ($\approx 5\times$). To gain parallax for distance estimation, suppose we additionally consider a limited spatial sampling setting, e.g., an omnidirectional video of a circular camera motion—this would be convenient for data capture [7]. But, as the baseline between spatial positions is small with respect to the large range of scene distances, and as each pixel covers a greater angular extent, it is difficult to accurately constrain the location of the geometry or to reproduce its fine detail. These factors decrease the effectiveness of ray marching techniques and, within an optimization, often necessitate an impractical number of samples for successful training convergence. This issue is further compounded by rapid and

large changes in depth in outdoor scenes—a difficult function to fit—that typically leads to a loss of detail in near-field geometry. In terms of memory, naive grids do not scale well to unbound scenes, and approaches tailored to small-baseline omnidirectional cameras should be more efficient.

To address these challenges, we propose a neural 3D reconstruction method specifically designed for short egocentric omnidirectional video inputs. Central to our approach is an *adaptive* spherical binoctree formed of spherical voxels ('sphoxels'). This dynamically subdivides from coarse to fine granularity through an iterative optimization, such that both ray samples and memory can focus on areas with more detail. Within the spatially-varying subdivisions of the leaf nodes, each sphoxel is fine-tuned to optimize spatial resolution and prioritize sampling in areas closer to the observed surfaces. Beyond this, we also provide two practical solutions: an efficient tree traversal strategy that manages the inconsistent number of child nodes based on breadth-first search, and an intersection test method to accommodate uneven shapes of the sphoxels in the binoctree structure.

Binoctrees were originally designed to subdivide a spherical space while compensating for voxel elongation that occurs at far distances [8]. However, this previous approach does not adapt the structure: not by the achievable depth accuracy, which can be estimated given camera poses, nor by any potential initial depth value, nor by the depth during optimization itself. We show that our approach improves upon an existing binoctree-based reconstruction method, reducing error by 38% on average. We also show improvement over a recent neural-field-based SDF reconstruction method that does not efficiently partition space within large-scale scenes [28]. Overall, our approach allows for more efficient and accurate 3D surface reconstruction of large-scale unbounded scenes using omnidirectional video inputs. Our method more effectively reconstructs the 3D environment from egocentric video inputs, addressing challenges in neural geometry with omnidirectional cameras. Our code is freely available for research purposes[1].

2. Related Work

360° photography. Scene understanding from 360° images has been actively explored together with the growth of the VR/AR industry. Using VR as a motivating application, several works focus on image-based rendering of 360° images from novel viewpoints [2, 4, 5]. These works adopt convolutional neural networks to construct multi-sphere images, proxy geometry from learned optical flow, and neural radiance fields. However, these works do not explicitly aim to understanding scene geometry and, therefore, suffer distortion when rendering trajectory deviates from training trajectories. Other work estimates dense depth and normal maps for corresponding 360° RGB images for inverse rendering [14]. However, this is limited to indoor scenes, and output depth maps require post-processing steps for explicit mesh extraction from obtained point clouds [9, 10]. On the other hand, our work focuses more on the geometrical understanding of spherical scenes by directly estimating the accurate signed distance function (SDF).

Jang et al. [8] share our goal in reconstructing a mesh of large-scale outdoor scenes from omnidirectional video. They use a binoctree structure to efficiently subdivide the unbounded space, then fuse truncated SDF (TSDF) values directly. This makes the output mesh resolution depend upon the voxel grid resolution, which is fixed from an initial depth estimate (e.g., from a pre-trained depth estimation network). Any errors in the initial depth also cause errors in fusion. Our work adopt a neural implicit function for surface representation, using an adaptive binoctree to guide sampling during reconstruction. This lets us refine any incorrect initial depth estimates, and does not require parameter tuning for fusion weights and truncation thresholds.

Scene-scale surface reconstruction. Structure-from-motion (SfM) and multi-view stereo (MVS) techniques have successfully reconstructed city-scale scenes from multi-view images [1, 11, 21]. However, these methods represent the 3D geometry with sparse point clouds that require extra steps to extract a surface mesh [9, 10]. With the emergence of radiance fields as a 3D scene representation [6], follow-up multi-view reconstruction studies have tackled outdoor scenes from object-centric unbounded scenes [3, 34] to larger scenes approaching urban scale [6, 17, 18, 20, 25, 29]. Radiance-field-based methods work well for novel view synthesis but often fail to estimate correct geometries from sparse views. Tang et al. [26] refine an extracted surface mesh to acquire delicate geometry and a texture map from radiance fields. However, they do not aim to reconstruct scene-scale geometry.

Neural implicit representations using SDFs show potential for image-based 3D reconstruction through several works [12, 19, 28, 30, 31]. These methods optimize an implicit function within the volumetric rendering pipeline by designing a density function derived from the estimated SDF. Recent works [7, 15, 24] extend the reconstruction scale to unbounded outdoor scenes. To handle optimization difficulties and preserve high-fidelity details, Sun et al. [24] use a sparse voxel grid built with SfM depth for the sampling, Li et al. [15] adopt numerical gradients and level-wise optimization of hash grid features [18], and Guo et al. [7] used cuboid-shaped hash grids and long-trajectory initialization schemes to handle unbounded street scenes.

Our work is built upon NeuS [28] to take advantage of a neural implicit representation; this embeds the geometry reconstruction optimization procedure inside a differentiable image-based rendering pipeline. However, our work dif-

[1] https://vclab.kaist.ac.kr/cvpr2024p2/

fers from previous research in how we interpret the reconstruction space. To handle narrow-baseline outward-facing scenes, we use an adaptive binoctree within spherical space for sampling. This is different from common approaches that use Cartesian coordinates for space sampling.

Grid-based training strategies. Voxel grids are widely used to accelerate rendering at the cost of memory. NSVF [16] uses a sparse voxel grid that adaptively prunes itself to narrow the sampling range. DVGO [23] and Plenoctrees [32] store density and appearance encodings to speed up training time and rendering. Instant-NGP [18] uses a hash table across multiple grid levels. These methods build a grid in NDC space based on Cartesian coordinates. Cartesian voxel grids assume that regions of interest are spread evenly within a cube and that sampling of the space with cameras follows similarly. However, for narrow-baseline outward-facing sampling (sometimes called egocentric), we need a novel design that can reduce sampling to plausible regions while encompassing unbounded scene points. EgoNeRF [5] uses a spherical feature grid as a neural scene representation to optimize a distant environment but focuses on novel view synthesis and does not reconstruct a surface. Our adaptive spherical binoctree method for omnidirectional inputs recovers 3D geometry with improved details and convergence speed over Cartesian grids.

3. Omnidirectional SDF Reconstruction

Given an omnidirectional video captured in a circular sweep, our goal is to optimize a neural SDF within a spherical space from which to reconstruct a 3D surface mesh. Our reconstruction algorithm starts from training NeuS [28] along with an initial binoctree constructed by the per-frame initial depth from a pre-trained spherical depth estimation network [8]. Then, we use step-wise sampling strategies with adaptive binoctree subdivision to optimize both the spherical grid and the neural networks with an image-based reconstruction loss. The core ideas of our approach are: 1) voxel-guided sampling with a sphere-shaped grid that subdivides the reconstruction space, which is specialized for memory efficiency, and 2) online and iterative refinement of grid structures based on the intermediate results.

3.1. Preliminaries

Implicit surface rendering. We define a surface $\mathcal{S}$ as the zero-level set of a signed distance function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$:

$$\mathcal{S} = \left\{ \mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0 \right\}. \quad (1)$$

We use two neural networks to estimate $\mathcal{S}$ [28, 30]: 1) Given a 3D point $\mathbf{x}$, $F_\theta^{\text{SDF}} : (\mathbf{x}) \rightarrow (d, \zeta)$ estimates the signed distance d to the surface at scene point $\mathbf{x}$, and additionally outputs a 256-dimensional feature vector ζ. 2) $F_\theta^{\text{c}} : (\mathbf{x}, \mathbf{n}, \mathbf{v}, \zeta) \rightarrow (c)$ estimates the view-dependent appearance c at scene point $\mathbf{x}$ using the normal $\mathbf{n}(t)$ to the SDF as its derivative. To render an image, we sample points along a ray $\{\mathbf{x}(t) = \mathbf{o} + t\mathbf{v} | t \geqslant 0\}$, where $\mathbf{o}$ is the ray origin and $\mathbf{v}$ is the ray direction. Then, we compute Wang et al.'s [28] unbiased and occlusion-aware weights $w(t)$ from the estimated SDFs to accumulate the final output color C by volume rendering:

$$C(\mathbf{o}, \mathbf{v}) = \int_{\text{near}}^{\text{far}} w(t)\, c(\mathbf{x}(t), \mathbf{n}(t), \mathbf{v}, \zeta(t))\, dt. \quad (2)$$

Binoctree. The spherical space is defined by a spherical binoctree [8]: a sphere-shaped octree structure that subdivides a sphere in radial (r), polar (θ) and azimuthal (ϕ) directions. We call each cell a sphoxel. This structure allows binary subdivisions in the radial direction to prevent sphoxel elongation as r increases, and is bound by near and far spheres. Each sphoxel stores its boundary in spherical coordinates $[r_{\text{min}}, r_{\text{max}}, \theta_{\text{min}}, \theta_{\text{max}}, \phi_{\text{min}}, \phi_{\text{max}}]$ and its index. We store sphoxel indices for each sampling stage and traverse the binoctree to efficiently fetch intersecting sphoxels.

Preprocessing. Following Jang et al. [8], this stage estimates camera poses and initializes the spherical binoctree from a depth estimate. We estimate depth for the initial tree construction using Jang et al.'s public trained model and code. We use structure from motion to solve for per-frame camera poses from the omnidirectional video (OpenVSLAM [22]). Given the poses, we extract a dense 3D point cloud for each frame using spherical rectification [13] and a fine-tuned disparity estimation network [27].

Using per-frame camera poses and calibrated depth maps as input, we form a global coordinate frame in metric units for the 3D point cloud. We place the origin at the center of the cameras and set the near sphere bound to contain all the camera positions. To represent the color of scene elements at infinity, such as the sky, we place a final sphere at the far plane with view-independent color decoded by a background MLP. We set the far sphere bound per scene to fit the recovered point cloud inside the sphere. Before binoctree construction and the neural network training, we scale both binoctree and point cloud coordinates into a unit sphere, and store the coefficient for later metric re-scaling.

For the initial binoctree structure, we use the scaled initial 3D point cloud to subdivide the tree to its finest level until all sphoxels reach a minimum solid angle α_{min}. We derive this size per scene from the baseline and image sizes, and set α_{min} between $5e^{-4}$ and $1e^{-3}$.

3.2. Sphoxel Intersection and Surface Existence

Unlike the previous binoctree method that directly integrates the SDF from depth maps [8], we optimize a hybrid binoctree/MLP SDF representation. We use the spherical

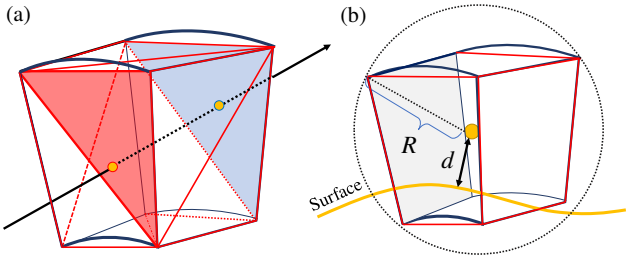

Figure 2. (a) Cuboid approximation of a sphoxel for intersection testing. Ray-triangle intersection tests occur for all triangles that comprise cuboid faces. (b) Illustration of rubrics for surface detection in sphoxels. A surface exists in a sphoxel if $\|d\| < R$.

binoctree only to guide sampling within the optimization, providing an efficient subdivision for large spherical spaces. Given the irregular shape of sphoxels, we cannot adopt conventional Cartesian grid-based algorithms for ray intersection and occupancy search [16, 32]. Thus, we suggest a novel intersection and surface existence test for sphoxels.

Each sphoxel is formed of two curved surfaces that each lie on a sphere defined by a near and far distance along radius r, and four surfaces by the intersection of those spheres with bounding planes formed at $\theta_{\text{min}}, \theta_{\text{max}}, \phi_{\text{min}}, \phi_{\text{max}}$ (Figure 2). Only two of these surfaces are simple in that they can be approximated by two triangles. Therefore, we cannot adopt common AABB cube intersection algorithms for sphoxel intersections. We re-define intersection for sphoxels by approximating them as trapezoidal prisms (square-based frusta) that share the same vertices (Figure 2(a)). Using the triangle-ray intersection algorithm, a ray intersects a sphoxel if we can detect intersections between a ray and a pair of triangles—one front-facing, one back-facing—that form the faces. We implement a CUDA kernel that fetches all intersecting sphoxels denoted by V. Since the number of marked voxels easily exceeds 10 k, we traverse the binoctree in a breadth-first order to relieve the bottleneck caused by unnecessary intersection computations.

When an angular subdivision spans a pole, then the binoctree around the pole forms a spherical sector (a cone and spherical cap) that contains the polar axis. We divide the spherical sector around the polar axis and approximate these sphoxel segments as triangle-based frusta.

Computing sphoxel intersections lets us detect where the current surface is, but we need another way to mark sphoxels with likely surface-crossings to be sampled in future iterations. For this, we define sphoxel center and radii. Since we compute intersections from the square-frusta approximation, we also define the center and radii from approximate geometry. A sphoxel's center is the mean position of its constituent eight vertices, and its radius is the distance from its center to the farthest vertex. Figure 2(b) compares the inferred absolute SDF value $\|d\|$ at the sphoxel center with the corresponding sphoxel radius R. We assume the surface passes through the sphoxel if $\|d\|$ is smaller than R.

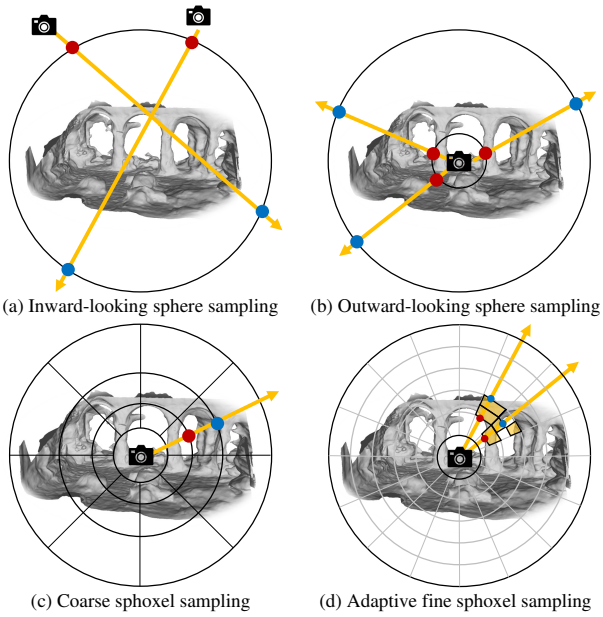

Figure 3. Sampling strategies on a unit sphere. (a) Sphere sampling range for the exocentric inward-looking model. (b)–(d) Sampling for egocentric data using a coarse-to-fine binoctree.

3.3. Spatial Sampling Types

We use three different levels of space sampling.

Whole-space spherical sampling. This occurs along each input camera ray between its intersections with the near and far bounding spheres. Since we assume that all cameras are located inside the near-bound sphere, rays uniquely intersect with both spheres. Figure 3(a) illustrates how this sampling differs from common ray-sphere intersection due to differing observation directions. We distribute sample points along each ray; we retain these samples throughout training and subsequent sphoxel-based samplings to prevent the optimization from failing to escape local minima.

Coarse sphoxel sampling. We select a set of coarse sphoxels V_{coarse} for sampling. Given the initial binoctree subdivision, we mark a sphoxel as $\in V_{\text{coarse}}$ if it is a parent or grandparent of a leaf sphoxel. Given the small baseline and large scene depth, this dilation step helps to resolve the sparse distribution of leaf sphoxels to better optimize plausible surface regions. We compute ray entry and exit points for all marked sphoxels and uniformly sample in between [16].

One issue is the sky: if sampled, this background appearance will be optimized into the binoctree at an incorrect depth. Sky regions typically produce few points. Another is outlier points in the initial 3D point cloud. To handle both, we additionally prune sphoxels from V_{coarse} if the number of contained initial points is lower than a threshold, which varies per scene based on the point density from 3 to 20.

Fine sphoxel sampling via adaptive subdivision. Coarse sphoxel sampling provides a geometrical guide in the initial

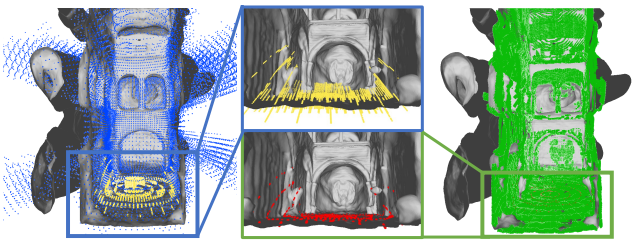

Figure 4. Adaptive sampling sphoxel bounds. *Left:* Coarse (blue) sphoxel vertices for sampling. *Center:* A sample of points from coarse (top) and fine (bottom) sphoxel intersections. *Right:* Fine (green) sphoxel vertices, which are denser near the surface.

optimization iterations. However, sampling between all intersecting coarse sphoxels samples a larger region than the true surface location. Therefore, to create V_{fine}, we periodically reassess V_{fine} membership and then adaptively subdivide member sphoxels during optimization. First, we initialize V_{fine} with V_{coarse}. Then, as optimized SDF values may push a surface outside of V_{fine}, we periodically fetch all leaf sphoxels in V_{coarse}, infer the SDF d, and mark a sphoxel as being in V_{fine} if $\|d\|$ is smaller than each sphoxel's radius. Then, we subdivided all marked sphoxels as V_{fine} such that surface optimization occurs at finer subdivisions. Figure 4 shows sphoxel vertices in coarse and fine levels. We can see that adaptive subdivision detects the surface successfully and reduces the sampling range appropriately.

3.4. Implementation Details

Neural networks. We optimize three multi-linear perceptrons (MLPs): one each to decode the SDF and color features within the binoctree and one for the background. The SDF network has eight layers of 256 neurons, and the color network has four layers of 256 neurons. We use standard positional encoding with six frequency bands for the SDF network and ten for the background NeRF.

Sample counts. We use 32 samples for each of the whole-space sphere, coarse voxel, and adaptive fine voxel stages. We used 32 samples for the background. Within the sphoxels, we also conduct importance sampling along the ray based on the PDF along it, using another 32 samples.

Subdivision schedule. The binoctree is subdivided every 10 k iterations until the leaf node size reaches a threshold angular size ranging 0.001–0.0001.

Supervision. The rendered color is supervised with an L1 photometric loss, and the SDFs are supervised with an eikonal loss. We add a loss to manually mask out the tripod that supports the camera and to mask initial depth values near the epipoles where depth cannot be estimated. Therefore, the total loss is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{RGB}} + \omega_{\text{eik}}\mathcal{L}_{\text{eik}} + \omega_{\text{mask}}\mathcal{L}_{\text{mask}}. \quad (3)$$

Iterations and computation. We optimize each scene on a computer equipped with a single NVIDIA A6000 GPU and an Intel Xeon Silver 4214R 2.40 GHz CPU with 256 GB RAM. It takes ∼20 hours for 300 k iterations. For the synthetic scenes, we evaluate the model after 100–200 k iterations depending on their complexity: 200 k for *Sponza*, 200 k for *San Miguel*, and 150 k for *Lone-monk*.

4. Results

Evaluation Method. We compare to five methods for reconstructing omnidirectional images. First, we use the classical structure-from-motion method COLMAP [21] after converting omnidirectional images to a six-face perspective cube map. Second, we compare to EgocentricRecon [8], an omnidirectional reconstruction method designed for the same kind of input as ours, and also using a binoctree. Third, fourth, and fifth, we compare our reconstruction results with three neural surface reconstruction methods: original NeuS [28], NeuS-facto [28, 33], and Neuralangelo [15]. NeuS-facto is a neural SDF reconstruction method that adapts NeuS by using the proposal network from mip-NeRF360 for sampling points along the ray. This comparison is suitable for unbounded scenes, whereas the original NeuS does not support these. Neuralangelo trains coarse to fine hash table grids with numerical gradients for higer-order derivatives.We use each method's implementation from SDF studio [33]. Approximate runtime took 10 hours for COLMAP, 30 minutes for EgocentricRecon, 5.5 hours for NeuS, 25 minutes for NeuS-Facto, and 31 hours for Neuralangelo to process the *Sponza* scene (Fig. 5) with 200 images. Our method took 12.5 hours.

Dataset. We use synthetic scenes for ground truth quantitative evaluation, using 3D scenes from the Blender Online Community and the McGuire Computer Graphics Archive. We render equirectangular RGB-D video with Blender at 2048 × 1024 pixels. The camera has a circular trajectory within a central region inside the scenes, spanning 200 frames. These three scenes—*Sponza*, *Lone-monk*, and *San Miguel*—have fine detail, especially *San Miguel*, which has many detailed trees that are challenging to reconstruct. We also show real-world reconstruction from the Omniphotos dataset captured with a swinging selfie stick [4].

Space subdivision efficiency. Our binoctree subdivision is an efficient method to reduce the number of voxels and tree depths when compared to the naive Cartesian subdivision. The number of uniform Cartesian voxels required to fill a unit sphere with the same size as the smallest sphoxel in our binoctree is great (Table 1): our spherical subdivision requires about 10 k × fewer voxels.

Surface reconstruction accuracy. Since only a portion of the complete ground truth mesh is reconstructed within the omnidirectional video sweep setting, a direct mesh accu-

Table 1. The number of voxels in each scene for a Cartesian subdivision grid is more than in our adaptive sphoxel grid. Our method is significantly more efficient while still achieving high-frequency details in reconstructed results.

Scene	Method	Number of voxels	Minimum sphoxel size
Sponza	Dense regular grid	33,335,054,331	
	Ours	4,346,041	1.25e-10
Lone-monk	Dense regular grid	2,234,638,740	
	Ours	231,237	1.87e-9
San Miguel	Dense grid	1,953,273,076	
	Ours	951,703	2.14e-9

Table 2. Quantitative evaluation of surface mesh accuracy for both classical and neural methods. We measure MAE and RMSE by rendering the inverse depth of each mesh and comparing to the ground truth mesh. Methods in bold achieve the best accuracy.

	Sponza		Lone-monk		San Miguel		Average	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
COLMAP	0.88	0.28	**0.40**	**0.15**	1.24	0.41	0.84	**0.28**
EgocentricRecon	**0.75**	**0.27**	1.05	0.43	**0.82**	**0.29**	0.88	0.33
NeuS	2.88	1.43	1.39	1.24	–	–	2.13	1.33
Neus-facto	2.44	1.12	2.25	1.68	4.02	2.84	2.91	1.88
Neuralangleo	0.99	0.44	1.91	1.71	0.58	0.29	1.16	0.81
Ours	**0.82**	**0.35**	**0.69**	**0.36**	**0.56**	**0.27**	**0.69**	**0.32**

racy metric such as Hausdorff distance is inappropriate for evaluation. Instead, to evaluate the accuracy of surface reconstruction, we render the depth of the reconstructed mesh from the center of the camera trajectory for all methods. Then, we compare each to the rendered depth of the ground truth mesh from the same position.

We measure the depth MSE and RMSE of valid (e.g., non-sky) ground truth mesh surfaces in Table 2. Our approach can optimize an SDF with comparable results to classical reconstruction methods while greatly exceeding the reconstruction quality of existing neural SDF methods. NeuS-facto shows improved accuracy in the Sponza scene and reconstructs the surface of San Miguel, which NeuS completely fails to reconstruct. As such, we leave out the San Miguel reconstruction result of NeuS since the reconstruction fails to create a mesh and so we cannot align and measure its accuracy. Neuralangelo shows the best reconstruction results among previous neural methods. However, our method achieves a higher accuracy still. We provide the qualitative comparison with neural methods in Figure 5, and with classical methods in the supplemental material. This validates that our method can successfully reconstruct detailed surfaces of large and complex outdoor scenes that current neural methods could not—simple methods based only on MLPs cannot easily scale to large scenes.

Figure 6 shows our qualitative reconstruction accuracy on a real scene. Our method can reconstruct both the scene details and smooth surfaces within the geometry. COLMAP reconstructs details without filling in holes; EgocentricRecon is overly smooth, and NeuS-facto is often in error.

Comparison with Depth Supervision. We compare our approach, which uses a binoctree pre-built from depth input as a sampling guide, to NeuS-facto-D, a method that uses depth supervision directly to guide surface reconstruction. Our findings suggest that using the binoctree as a guide is superior to direct depth supervision. To incorporate depth supervision, NeuS-facto-D adds a depth loss component to the loss with a weight of 0.1. The depth loss is calculated using L1 loss between the rendered depth and input depth. Depth supervision can improve surface reconstruction on average but our approach yields more significant improvements (Table 3). One reason why is because depth supervision can often be inaccurate. Our approach conservatively sets bounds upon the depth and then relies upon other losses to optimize the surface location.

Table 3. Quantitative comparison of binoctree-based geometry guidance and depth supervision for geometry reconstruction quality evaluation using RMSE depth accuracy.

Method	Sponza	Lone-monk	San Miguel	Average
NeuS-facto [33]	2.44	2.25	4.03	2.91
NeuS-facto-D [33]	1.46	4.75	1.80	2.67
Ours	**0.82**	**0.69**	**0.56**	**0.69**

Table 4. Number of samplings for sampling strategies ablations

Sampling techniques	N_{sphere}.	N_{coarse}	N_{fine}	$N_{\text{importance}}$ / steps
Sphere sampling	64	0	0	64 / 4
Sphere + coarse voxel	32	32	0	64 / 4
Sphere + coarse + fine voxel	32	32	32	32 / 2

Ablation. To demonstrate the effectiveness of our sampling method, we display the training progress using normal maps for three different approaches: sphere sampling only, sphere and coarse sphoxel sampling, and hybrid (sphere, coarse, and fine sphoxel) sampling at 5 k, 10 k, and 100 k training iterations. For fairness, we keep the total number of samples per ray constant at 128. Our baseline setting (sphere sampling only) follows the default sampling strategies from NeuS except that it uses outward-looking sphere sampling instead(Fig 3 (b)). Then, we sample more between coarse sphoxels instead of sphere bounds. Lastly, we sample fine sphoxels instead of importance sampling. Table 4 shows the number of samples used for this ablation.

Figure 7 (bottom right) shows that scene details in concave regions are estimated with accuracy only if ray sampling is strongly guided by both coarse and fine sphoxels. Figure 8 qualitatively demonstrates that our adaptive fine voxel correctly converges to the surface location.

5. Discussion and Conclusions

We have presented an adaptive subdivision of spherical space via a binoctree with novel sampling techniques. This subdivision and sampling are useful to optimize a neural implicit representation of geometry from an omnidirectional video. When such a representation is optimized for large unbounded scenes, the continuous property of the represen-

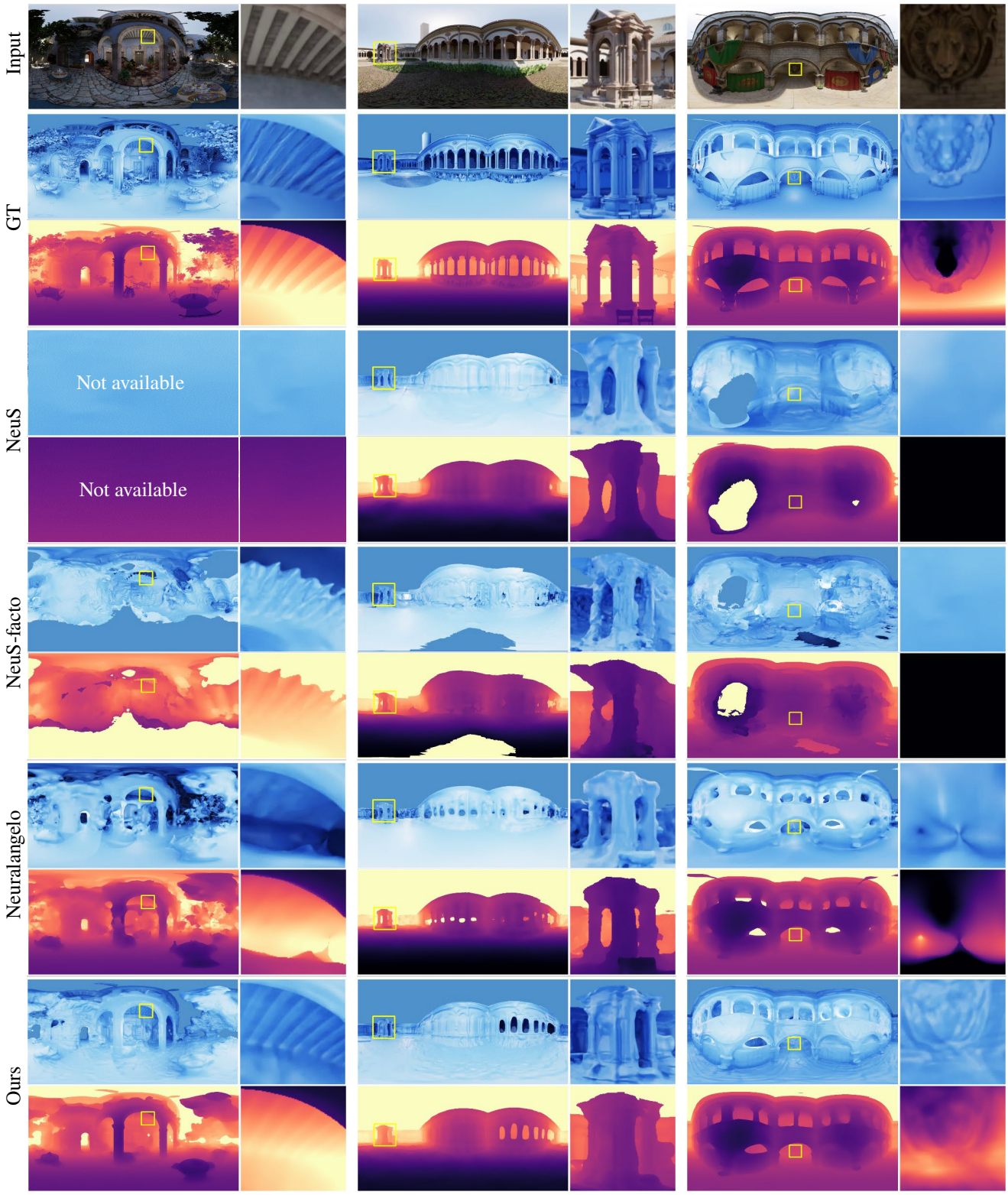

Figure 5. We compare our method with traditional and neural methods using ground-truth geometry. We present qualitative results of NeuS [28], NeuS-facto [33] and Neuralangelo [15] here; our method produce higher-quality 3D geometry. Please refer to Table 2 for complementary quantitative evaluation and to supplemental material for further qualitative comparisons of omitted traditional methods, including COLMAP [21], and EgocentricRecon [8].

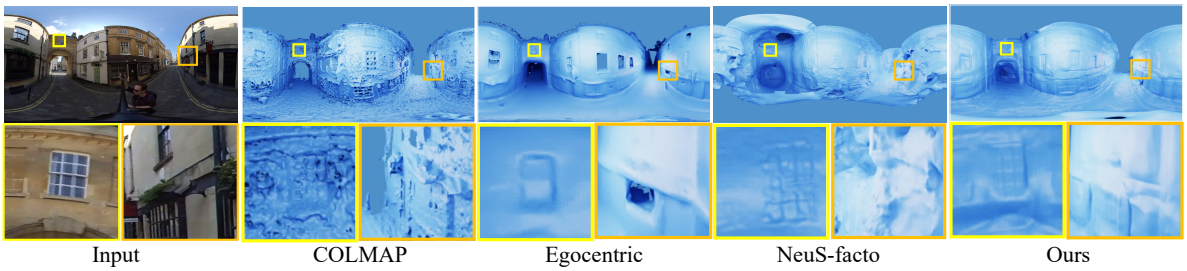

Figure 6. Qualitative comparisons of our reconstruction on a real-world scene from the Omniphotos dataset. Please refer to the supplemental video for more results.

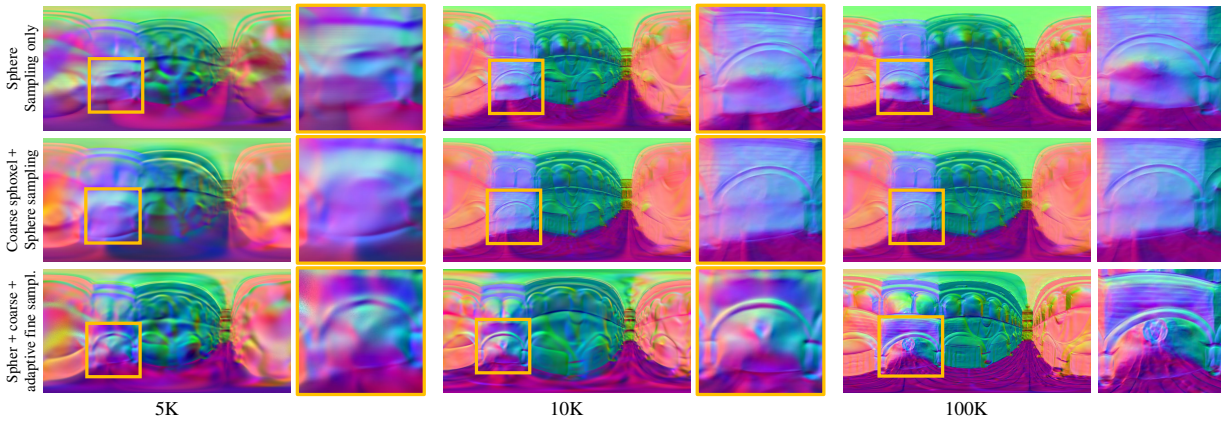

Figure 7. As an ablation, we show the progress of geometry optimization with the omnidirectional normal map of the *sponza* scene. Normal maps are rendered per training iterations of 5K, 10K, and 100K for each sampling strategy. The geometry in the hollow region was recovered only when both coarse and fine sampling was adopted.

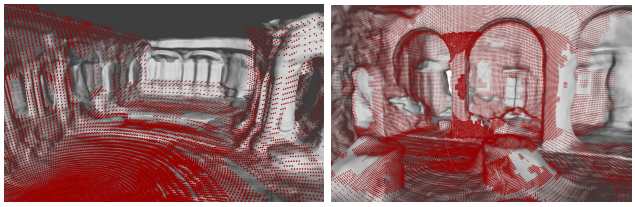

Figure 8. Adaptive fine voxel bounds (red point samples) have correctly converged to the surface location.

tations often causes smoothed surfaces that lack detail. To overcome this issue, we use an adaptive subdivision strategy that densifies samples near the surface. This strategy subdivides the sphoxel towards the surface and moves samples similarly so that density is placed where it is needed.

To address the challenges posed by using a neural function to estimate abrupt changes in disparity and sparse observations from omnidirectional input, we propose constructing coarse space bounds from depth input to provide a geometrical guide during optimization. The spherical binoctree efficiently subdivides the omnidirectional space, taking into account the rendering resolution and memory space. We subdivide the space based on solid angle measurement and distance from the camera, using a spatially variant tree depth, using orders of magnitude fewer voxels than a Cartesian grid with an equivalent minimum voxel size. This leads to improved surface detail without requiring significant memory. Our model performance is comparable to traditional surface reconstruction methods; concurrent neural methods using MLPs alone cannot achieve this quality for our inputs, according to our comparisons. Overall, quantitative, qualitative, and ablation experiments confirm the effectiveness of our approach as a promising method for small-baseline omnidirectional data.

Limitations. Our approach uses an initial depth map to initialize the coarse voxel before subdividing them into finer levels. Should the estimation of the coarse voxels be wrong, it can lead to inaccurate guidance for the coarse geometry. This also may cause inaccuracies when samples are estimated at scene regions that should belong to the background but have nearer initial depths. We attempt to lessen these effects by dilating and pruning the voxel before optimization, but it may be necessary to perform postprocessing to ensure better final meshes without spurious artifacts.

Further, some scene geometry details still remain challenging to reconstruct, e.g., concave region reconstruction may be sensitive to particular minima in the optimization landscape. Further, effects that should be explained by fine-scale texture variation may be baked into the geometry.

Acknowledgements

Min H. Kim acknowledges the MSIT/IITP of Korea (RS-2022-00155620, 2022-0-00058, and 2017-0-00072), LIG, and Samsung Electronics. James Tompkin acknowledges US NSF CAREER 2144956.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 2

[2] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. In *European Conference on Computer Vision (ECCV)*, pages 441–459. Springer, 2020. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[4] Tobias Bertel, Mingze Yuan, Reuben Lindroos, and Christian Richardt. Omniphotos: casual 360 vr photography. *ACM Transactions on Graphics (TOG)*, 39(6):1–12, 2020. 2, 5

[5] Changwoon Choi, Sang Min Kim, and Young Min Kim. Balanced spherical grid for egocentric view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16590–16599, 2023. 2, 3

[6] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022. 2

[7] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023. 1, 2

[8] Hyeonjoong Jang, Andreas Meuleman, Dahyun Kang, Donggun Kim, Christian Richardt, and Min H Kim. Egocentric scene reconstruction from an omnidirectional video. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 1, 2, 3, 5, 7

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

[10] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, page 0, 2006. 2

[11] 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. 2

[12] 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

[13] Shigang Li. Binocular spherical stereo. *IEEE Transactions on intelligent transportation systems*, 9(4):589–600, 2008. 3

[14] Zhen Li, Lingli Wang, Xiang Huang, Cihui Pan, and Jiaqi Yang. Phyir: Physics-based inverse rendering for panoramic indoor images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12713–12723, 2022. 2

[15] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 1, 2, 5, 7

[16] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3, 4

[17] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progressively optimized local radiance fields for robust view synthesis. In *CVPR*, 2023. 2

[18] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 3

[19] 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

[20] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 2

[21] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 2, 5, 7

[22] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. Openvslam: A versatile visual slam framework. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2292–2295, 2019. 3

[23] 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. 3

[24] 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. 1, 2

[25] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 2

[26] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. Delicate textured mesh recovery from nerf via adaptive surface refinement. *arXiv preprint arXiv:2303.02091*, 2023. 2

[27] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3

[28] 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, 5, 7

[29] Xiuchao Wu, Jiamin Xu, Xin Zhang, Hujun Bao, Qixing Huang, Yujun Shen, James Tompkin, and Weiwei Xu. Scanerf: Scalable bundle-adjusting neural radiance fields for large-scale scene rendering. *ACM Transactions on Graphics (TOG)*, 2023. 2

[30] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems*, pages 2492–2502. Curran Associates, Inc., 2020. 2, 3

[31] 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

[32] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 3, 4

[33] Zehao Yu, Anpei Chen, Bozidar Antic, Songyou Peng Peng, Apratim Bhattacharyya, Michael Niemeyer, Siyu Tang, Torsten Sattler, and Andreas Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022. 1, 5, 6, 7

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