Fast Monocular Scene Reconstruction with Global-Sparse Local-Dense Grids

Wei Dong* Chris Choy† Charles Loop† Or Litany† Yuke Zhu† Anima Anandkumar†

Abstract

Indoor scene reconstruction from monocular images has long been sought after by augmented reality and robotics developers. Recent advances in neural field representations and monocular priors have led to remarkable results in scene-level surface reconstructions. The reliance on Multilayer Perceptrons (MLP), however, significantly limits speed in training and rendering. In this work, we propose to directly use signed distance function (SDF) in sparse voxel block grids for fast and accurate scene reconstruction without MLPs. Our globally sparse and locally dense data structure exploits surfaces' spatial sparsity, enables cache-friendly queries, and allows direct extensions to multi-modal data such as color and semantic labels. To apply this representation to monocular scene reconstruction, we develop a scale calibration algorithm for fast geometric initialization from monocular depth priors. We apply differentiable volume rendering from this initialization to refine details with fast convergence. We also introduce efficient high-dimensional Continuous Random Fields (CRFs) to further exploit the semantic-geometry consistency between scene objects. Experiments show that our approach is 10× faster in training and 100× faster in rendering while achieving comparable accuracy to state-of-the-art neural implicit methods.

1. Introduction

Reconstructing indoor spaces into 3D representations is a key requirement for many real-world applications, including robot navigation, immersive virtual/augmented reality experiences, and architectural design. Particularly useful is reconstruction from monocular cameras which are the most prevalent and accessible to causal users. While much research has been devoted to this task, several challenges remain.

Conventional monocular reconstruction from multi-view RGB images uses patch matching [36], which takes hours to reconstruct even a relatively small scene. Several 3D reconstruction methods [40,48] have demonstrated fast reconstruction by applying 3D convolutional neural networks to feature volumes, but they have limited resolution and struggle to generalize to larger scenes.

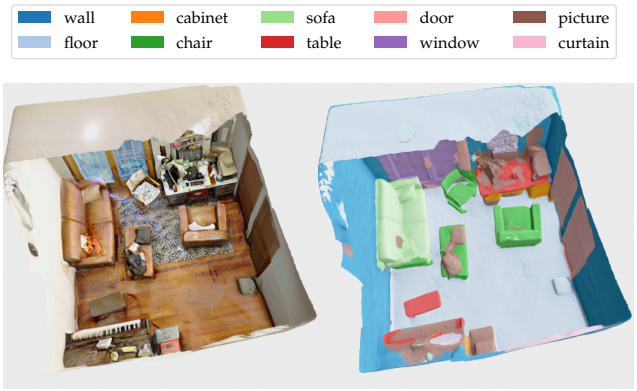
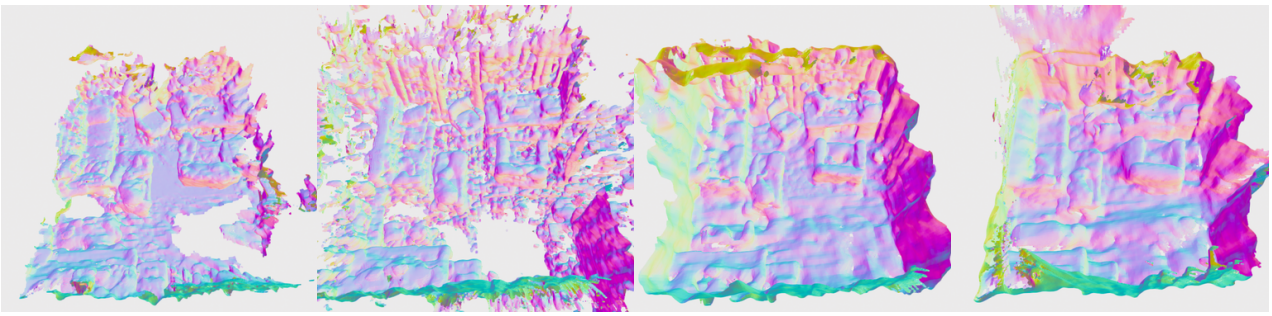


Figure 1. Color and semantic scene reconstruction from our system with monocular images and learned monocular priors.

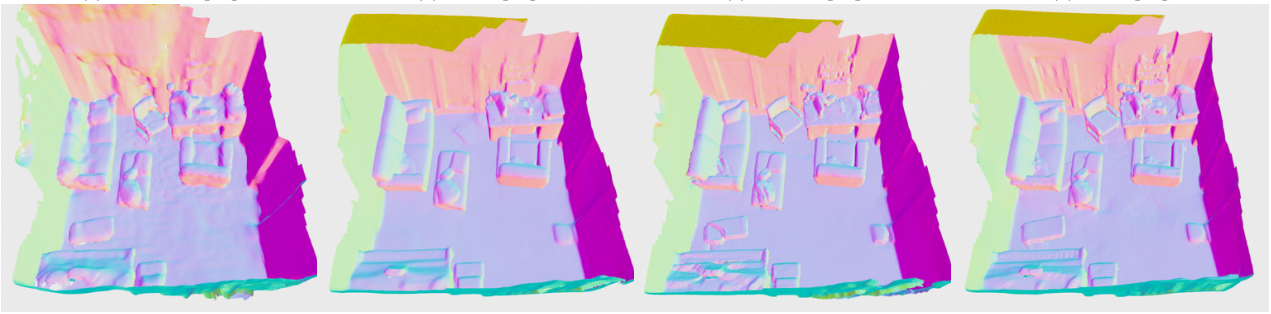
Recently, unified neural radiance fields [23] and neural implicit representations were developed for the purpose of accurate surface reconstruction from images [31, 45, 49]. While this was successfully demonstrated on single objects, the weak photometric constraint leads to poor reconstruction and slow convergence for large-scale scenes. Guo *et al*. [14] and Yu *et al*. [51] improved the quality and convergence speed of neural field reconstruction on large-scale scenes by incorporating learned geometrical cues like depth and normal estimation [11, 33], however, training and evaluation remain inefficient. This is primarily because these approaches rely on MLPs and feature grids [24] that encode the entire scene rather than concentrating around surfaces.

In contrast to MLPs, an explicit SDF voxel grid can be adaptively allocated around surfaces, and allows fast query and sampling. However, an efficient implementation of differentiable SDF voxel grids without MLPs is missing. Fridovich-Keil and Yu *et al*. [12] used an explicit density and color grid, but is limited to rendering small objects. Muller *et al*. [24] developed a feature grid with spatial hashing for fast neural rendering, but its backbone hash map is not collision-free, causing inevitable slow random access and inaccurate indexing at large scales. Dong *et al*. [10] pro-

*CMU RI. Work done during the internship at NVIDIA.
†NVIDIA.



(a) COLMAP [36] (b) NeRF [23] (c) VolSDF [49] (d) NeuS [45]



(e) ManhattanSDF [14] (f) MonoSDF-MLP [51] (g) MonoSDF-Grid [51] (h) Ours

Figure 2. Qualitative reconstruction comparison on ScanNet [7]. While being 10× faster in training, we achieve similar reconstruction results to state-of-the-art MonoSDF [51], with fine details (see Fig. 9).

posed a collision-free spatially hashed grid following Niessner *et al*. [30], but lacks support for differentiable rendering. Several practical challenges hinder the implementation of an efficient differentiable data structure: 1. a collision-free spatial hash map on GPU that supports one-to-one indexing from positions to voxels; 2. differentiable trilinear interpolations between spatially hashed voxels; 3. parallel ray marching and uniform sampling from a spatial hash map.

Our approach: we address such challenges using a differentiable *globally sparse* and *locally dense* voxel grid. We transform a collision-free GPU hash map [37] to a differentiable tensor indexer [32]. This generates a one-to-one map between positions and *globally sparse* voxel blocks around approximate surfaces, and enables skipping empty space for efficient ray marching and uniform sampling. We further manage *locally dense* voxel arrays within sparse voxel blocks for GPU cache-friendly contiguous data query via trilinear interpolation. As a result, using explicit SDF grids leads to fast SDF gradient computation in a single forward pass, which can further accelerate differentiable rendering.

This new data structure presents a new challenge — we can only optimize grid parameters if they are allocated around surfaces. To resolve this, we make use of off-the-shelf monocular depth priors [11, 33] and design a novel initialization scheme with global structure-from-motion (SfM) constraints to calibrate these unscaled predicted depths. It results in a consistent geometric initialization via volumetric fusion ready to be refined through differentiable volume rendering.

We additionally incorporate semantic monocular priors [17] to provide cues for geometric refinement in 3D. For instance, we use colors and semantics to guide the sharpening of normals around object boundaries, which in turn improves the quality of colors and semantics. We enforce these intuitive notions through our novel continuous Conditional Random Field (CRF). We use Monte Carlo samples on the SDF zero-crossings to create continuous CRF nodes and define pairwise energy functions to enforce local consistency of colors, normals, and semantics. Importantly, we define similarity in a *high dimensional space* that consists of coordinates, colors, normals, and semantics, to reject spatially close samples with contrasting properties. To make inference tractable, we follow Krahenbuhl *et al*. [16] and use variational inference, leading to a series of convolutions in a high-dimensional space. We implement an efficient permutohedral lattice convolution [1] using the collision-free GPU hashmap to power the continuous CRF inference.

The final output of our system is a scene reconstruction with geometry, colors, and semantic labels, as shown in Fig. 1. Experiments show that our method is 10× faster in training, 100× faster in inference, and has comparable accuracy measured by F-scores against state-of-the-art implicit reconstruction systems [14, 51]. In summary, we propose a fast scene reconstruction system for monocular images. Our contributions include:

- A globally sparse locally dense differentiable volumetric data structure that exploits surface spatial sparsity without an MLP;

- A scale calibration algorithm that produces consistent geometric initialization from unscaled monocular depths;
- A fast monocular scene reconstruction system equipped with volume rendering and high dimensional continuous CRFs optimization.

2. Related Work

Surface reconstruction from 3D data. Surface reconstruction has been well-studied from 3D scans. The general idea is to represent the space as an implicit signed distance function, and recover surfaces at zero-crossings with Marching Cubes [19]. Classical works [6,8,10,28,30] quantize the 3D space into voxels, and integrate frame-wise SDF observations into voxels. Instead of direct voxels, recent neural representations [38, 41, 54] use either a pure MLP or a feature grid to reconstruct smoother surfaces. These approaches are often fast and accurate, but heavily depends on high-quality depth input from sensors.

Surface reconstruction from RGB. A variety of classical and learning-based methods [29, 40, 48, 52] have been developed to achieve high quality multi-view depth reconstruction from monocular images. These techniques usually construct a cost volume between a reference frame and its neighbor frames, and maximize the appearance consistency. A global volume can be optionally grown from the local volumes [26, 40, 52] for scene reconstruction. While these approaches succeed on various benchmarks, they rely on fine view point selection, and the performance may be significantly reduced when the view points and surfaces are sparse in space. Training on a large datasets is also required.

Recent advances in neural rendering [3, 23, 44] and their predecessors have defined the surface geometry by a density function predicted by an MLP in 3D space. They seek to minimize ray consistency with the rendering loss using test-time optimization. While being able to achieve high rendering quality, due to the ambiguity in the underlying density representation, accurate surfaces are hard to recover. In view of this, implicit SDF representations [39,45,49,50] are used to replace density, where surfaces are better-defined at zero-crossings. To enable large-scale indoor scene reconstruction, ManhattanSDF [14] and MonoSDF [51] incorporate monocular geometric priors and achieve state-of-the-art results. These approaches initialize the scene with a sphere [2], and gradually refine the details. As a result, the training time can be long, varying from hours to half a day.

Monocular priors in surface reconstruction. Priors from monocular images have been used to enhance reconstruction and neural rendering from images, by providing reasonable initialization and better constrained sampling space. A light weight prior is the structure-from-motion (SfM) supervision [36], where poses and sparse point clouds are reconstructed to provide the geometry. Similarly, dense monocular priors including depths [18,33,34], normals [11], and semantic segmentations [17]. Existing approaches either use only SfM [9], or enhance dense priors by SfM via finetuning depth prediction networks [46] or depth completion networks [35] for density-based neural rendering. Similarly, monocular priors are used to enhance SDF-based neural reconstruction [14, 51] with remarkable performance. While emphasizing the guidance in sampling, these approaches usually stick to MLPs or dense voxel grids, without being able to fully exploit the sparsity of the surface distribution.

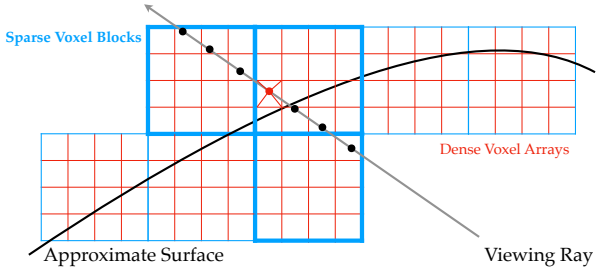


Figure 3. Illustration of the sparse-dense data structure. The large voxel blocks (in blue) are allocated only around approximate surfaces, and indexed by a collision-free hash map. The voxel arrays (in red) further divide the space to provide high-resolution details. Ray marching skips empty space and only activates hit blocks (in bold blue). Trilinear interpolation of neighbor voxel properties allows sampling at continuous locations.

Sparse spatial representations. Sparse spatial representations have been well-studied for 3D data, especially point clouds [4, 10, 30]. Often used data structures are hash maps, Octrees [22], or a combination [27]. These data structures have been adapted to neural 3D reconstructions and rendering to exploit spatial sparsity, but they either depend on high quality 3D input [41], or focus on object-centered reconstruction [5, 12, 24]. Their usage to scene reconstruction from monocular images is yet to be explored.

3. Method

3.1. Overview

The input to our method is a sequence of monocular images $\{\mathcal{I}_i\}$. Prior to reconstruction, similar to previous works [14, 51], we generate per-image monocular priors including unscaled depth $\{\mathcal{D}_i\}$ and normal $\{\mathcal{N}_i\}$ predicted by Omnidata [11], and semantic logits $\{\mathcal{S}_i\}$ from LSeg [17]. Afterwards, the system runs in three major stages.

- Sparse SfM reconstruction [36] and initial depth scale optimization;
- Direct volumetric fusion for sparse voxel allocation and geometric initialization;
- Differentiable volume rendering and dense CRF smoothing for detail refinement.

Fig. 4 shows the pipeline of our framework. We will describe these stages in detail after introducing our core data structure.

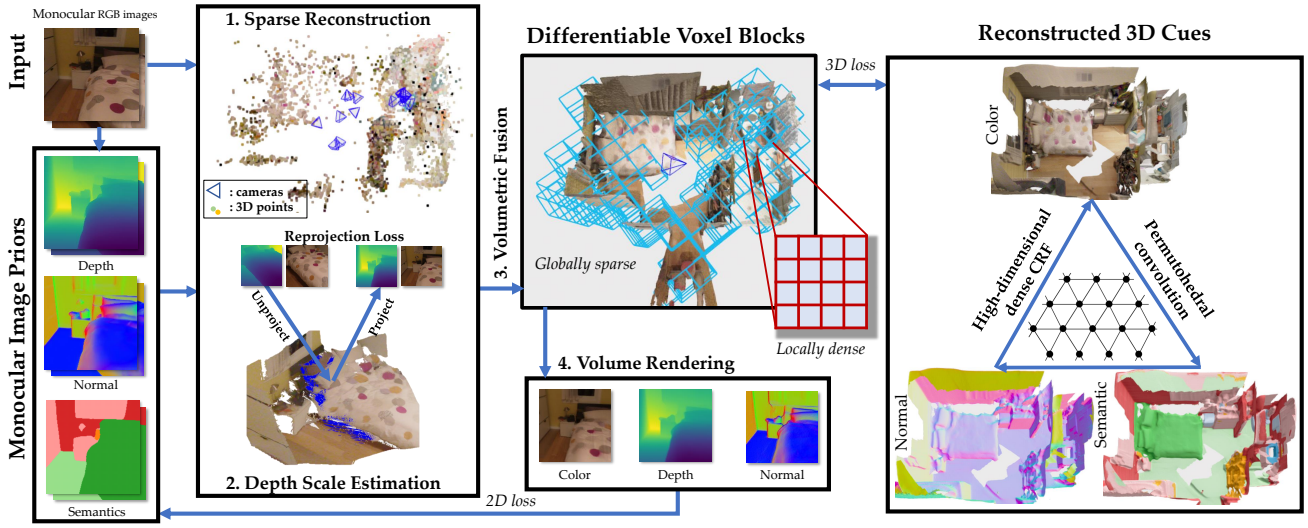


Figure 4. Illustration of our pipeline. We first use structure-from-motion (SfM) to obtain sparse feature-based reconstruction. With the sparse point cloud and covisibility information from SfM, we optimize the scale of predicted monocular depth images (§3.3), and perform volumetric fusion to construct a globally sparse locally dense voxel grid (§3.4). After initialization, we perform differentiable volume rendering to refine the details (§3.5.1), and apply high dimensional continuous CRFs to finetune normals, colors, and labels (§3.5.3).

3.2. Sparse-Dense Data Structure

In order to facilitate multi-view sensor fusion, SDF are approximated by truncated SDF (TSDF) that maintain averaged signed distances to surface in a narrow band close to the surface [6, 28, 30]. We take advantage of this property and develop a globally sparse locally dense data structure. Global sparsity is attained through allocating voxels only around approximate surfaces, which we index using a collision-free hash map. Within these sparse voxels we allocate cache-friendly small dense arrays that allow fast indexing and neighbor search storing SDF, color, and optionally labels. The data structure is visualized in Fig. 3.

While similar structures have been used for RGB-D data that focus on forward fusion [10, 30], our implementation supports both forward fusion via hierarchical indexing, and auto-differentiable backward optimization through trilinear interpolation, allowing refinement through volume rendering. In addition, SDF gradients can be explicitly computed along with SDF queries in the same pass, allowing efficient regularization during training.

Our data structure is akin to any neural networks that maps a coordinate $\mathbf{x} \in \mathbb{R}^3$ to a property [47], thus in the following sections, we refer to it as a function f. We use $f_{\theta_d}, f_{\theta_c}, f_{\theta_s}$ to denote functions that query SDF, color, and semantic labels from the data structure, respectively, where θ_d, θ_c, and θ_s are properties directly stored at voxels.

3.3. Depth Scale Optimization

Our sparse-dense structure requires approximate surface initialization at the allocation stage, hence we resort to popular monocular geometry priors [11] also used in recent works [51]. Despite the considerable recent improvement of monocular depth prediction, there are still several known issues in applications: each image's depth prediction is scale-ambiguous, often with strong distortions. However, to construct an initial structure we require a consistent scale.

To resolve this, we define a scale function ϕ_i per monocular depth image $\mathcal{D}_i$ to optimize scales and correct distortions. ϕ_i is represented by a 2D grid, where each grid point stores a learnable scale. A pixel's scale $\phi_i(\mathbf{p})$ can be obtained through bilinear interpolating its neighbor grid point scales. We optimize $\{\phi_i\}$ to achieve consistent depth across frames

$$\min_{\{\phi_i\}} \sum_{i,j\in\Omega} h(\phi_i, \phi_j) + \lambda \sum_i g(\phi_i), \tag{1}$$

where h and g impose mutual and binary constraints, and Ω is the set of covisible image pairs.

Previous approaches [15] use fine-grained pixel-wise correspondences to construct h via pairwise dense optical flow, and introduce a regularizer g per frame. This setup is, however, computationally intensive and opposes our initial motivation of developing an efficient system. Instead, we resort to supervision from SfM's [36] sparse reconstruction. It estimates camera poses $\{R_i, t_i\}$, produces sparse reconstruction with 3D points $\{\mathbf{x}_k\}$ and their associated 2D projections $\mathbf{p}_{\mathbf{x}_k \to i}$ at frame $\{\mathcal{I}_i, \mathcal{D}_i\}$, and provides the covisible frame pair set Ω. With such, we can define the unary constraint g via a reprojection loss

$$g(\phi_i) = \sum_{\mathbf{x}_k} \| d_{\mathbf{x}_k \to i} - \mathcal{D}_i(\mathbf{p}_{\mathbf{x}_k \to i}) \phi_i(\mathbf{p}_{\mathbf{x}_k \to i}) \|^2, \tag{2}$$

$$d_{\mathbf{x}_k \to i} \cdot \begin{bmatrix} \mathbf{p}_{\mathbf{x}_k \to i} & 1 \end{bmatrix}^\top \triangleq \Pi \left(\mathbf{R}_i^\top (\mathbf{x}_k - \mathbf{t}_i) \right), \tag{3}$$

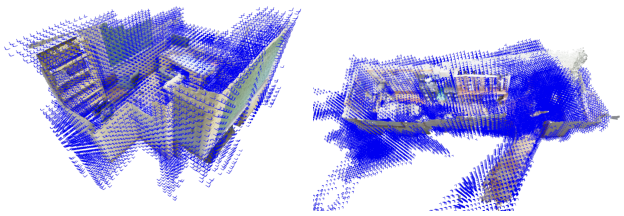


Figure 5. Sparse voxel grid (blue) allocation around 3D points from unprojection. The grids are adaptive to scenes with different overall surface shapes. Ground truth surface mesh are visualized for illustration.

where Π is the pinhole projection. Similarly, we define binary constraints by minimizing reprojection errors across covisible frames:

$$h(\phi_i, \phi_j) = \sum_{\mathbf{p} \in \mathcal{D}_i} \|d_{i\to j} - \mathcal{D}_j(\mathbf{p}_{i\to j})\phi_j(\mathbf{p}_{i\to j})\|^2 + \|\mathcal{I}_i(\mathbf{p}) - \mathcal{I}_j(\mathbf{p}_{i\to j})\|^2, \quad (4)$$

$$\mathbf{x} = \Pi^{-1}\Big(\mathbf{p}, \mathcal{D}_i(\mathbf{p})\phi_i(\mathbf{p})\Big), \quad (5)$$

$$d_{i\to j} \cdot \begin{bmatrix}\mathbf{p}_{i\to j} & 1\end{bmatrix}^\top \triangleq \Pi(\mathbf{R}_{i,j}\mathbf{x} + \mathbf{t}_{i,j}), \quad (6)$$

where Π^{-1} unprojects a pixel $\mathbf{p}$ in frame i from deformed depth to a 3D point $\mathbf{x}$, and $\{\mathbf{R}_{i,j}, \mathbf{t}_{i,j}\}$ are relative poses. This loss enforces local consistency between visually adjacent frames.

We use a 24×32 2D grid per image, $\lambda = 10^{-3}$, and optimize $\{\phi_i\}$ via RMSprop with a learning rate of 10^{-2} for 500 steps.

3.4. Direct Fusion on Sparse Grid

3.4.1 Allocation

Similar to aforementioned works for online reconstruction [10, 30], the sparse blocks are allocated by the union of voxels containing the unprojected points,

$$\mathbf{X} = \cup_i \mathbf{X}_i, \mathbf{X}_i = \cup_p \Big\{\text{Dilate}\left(\text{Voxel}(\mathbf{p})\right)\Big\}, \quad (7)$$

$$\text{Voxel}(\mathbf{p}) = \left\lfloor \frac{\mathbf{R}_i\Pi^{-1}(\mathbf{p}, \mathcal{D}_i(\mathbf{p})\phi_i(\mathbf{p})) + \mathbf{t}_i}{L} \right\rfloor, \quad (8)$$

where L is the sparse voxel block size, and the dilate operation grows a voxel block to include its neighbors to tolerate more uncertainty from depth prediction. A dynamic collision-free hash map [10] is used to efficiently aggregate the allocated voxel blocks. The dense voxel arrays are correspondingly allocated underlying the sparse voxel blocks.

Fig. 5 shows the surface-adaptive allocation. In contrast to popular sparse grids in a fixed bounding box used by neural rendering [12, 24], this allocation strategy is more flexible to various shapes of rooms.

Figure 6. With scale calibration and volumetric fusion, room-scale geometry initialization can be achieved from monocular depth without any optimization of the voxel grid parameters. The remaining task would be refining noisy regions and prune outliers.

3.4.2 Depth, Color, and Semantic Fusion

Following classical volumetric fusion scheme [28, 30], we project voxels $\mathbf{v}$ back to the images to setup voxel-pixel associations, and directly optimize voxel-wise properties, namely SDF (θ_d), color (θ_c), and semantic label logits (θ_s).

$$\theta_d(\mathbf{v}) = \arg\min_d \sum_i \Big(- \big(d_{\mathbf{v}\to i} - \mathcal{D}_i(\mathbf{p}_{\mathbf{v}\to i})\phi_i(\mathbf{p}_{\mathbf{v}\to i})\big)\Big)^2, \quad (9)$$

$$\theta_c(\mathbf{v}) = \arg\min_{\mathbf{c}} \sum_i \|\mathbf{c} - \mathcal{I}_i(\mathbf{p}_{\mathbf{v}\to i})\|^2, \quad (10)$$

$$\theta_s(\mathbf{v}) = \frac{\mathbf{s}^*}{\|\mathbf{s}^*\|}, \mathbf{s}^* = \arg\min_{\mathbf{s}} \sum_i \|\mathbf{s} - \mathcal{S}_i(\mathbf{p}_{\mathbf{v}\to i})\|^2, \quad (11)$$

where the projection $\mathbf{v} \to i$ is given by Eq. 3. Note by definition, only associations with SDF smaller than a truncation bound will be considered, minimizing the effect of occlusion. It is worth mentioning that we use a simple L2 loss for semantic logit instead of entropy losses, as it is considered one of the best practices in label fusion [21]. The closed-form solutions of aforementioned voxel-pixel association losses are simply averages. Therefore, with minimal processing time, we can already achieve reasonable initial surface reconstruction by classical volumetric SDF and color fusion, see Fig. 6.

3.4.3 De-noising

Direct fused properties, although being smoothed average of observations across frames *per voxel*, are spatially noisy and can result in ill-posed SDF distributions along rays. Therefore, we perform a Gaussian blurring for the voxels along all the properties. Thanks to the direct representation, with a customized forward sparse convolution followed by a property assignment, we could accomplish the filtering without backward optimizations. The effect of the de-noising operation can be observed in Fig. 7(a)-(b).

(a) Init fusion (b) De-noised (c) Refined

Figure 7. Comparison between 3 stages of reconstruction: initialization, de-noising, and volume rendering refinement.

3.5. Differentiable Geometry Refinement

3.5.1 Volume Rendering

We follow MonoSDF [51] to refine geometry using monocular priors. For a pixel $\mathbf{p}$ from frame i, we march a ray $\mathbf{x}(t) = \mathbf{r}_o + t \cdot \mathbf{r}_d$ to the sparse voxel grid, sample a sequence of points $\{\mathbf{x}_k = \mathbf{x}(t_k)\}$, apply volume rendering, and minimize the color, depth, and normal losses respectively:

$$\mathcal{L}_c(\theta_c, \theta_d) = \left\| \sum_k w(\mathbf{x}_k) f_{\theta_c}(\mathbf{x}_k) - \mathcal{I}_i(\mathbf{p}) \right\|, \tag{12}$$

$$\mathcal{L}_d(\theta_d) = \left\| \sum_k w(\mathbf{x}_k) t_k - (a\mathcal{D}_i(\mathbf{p}) + b) \right\|^2, \tag{13}$$

$$\mathcal{L}_n(\theta_d) = \left\| \sum_k w(\mathbf{x}_k) \nabla f_{\theta_d}(\mathbf{x}_k) - \mathcal{N}_i(\mathbf{p}) \right\|, \tag{14}$$

$$w(\mathbf{x}_k) = \exp\Big(-\sum_{j<k} \alpha(\mathbf{x}_j)\delta_j \Big)\big(1 - \exp(-\alpha(\mathbf{x}_k)\delta_k)\big), \tag{15}$$

where $\delta_i = t_{i+1} - t_i$, depth scale a and shift b are estimated per minibatch in depth loss with least squares [33], and the density $\alpha(\mathbf{x}_k) = l(f_{\theta_d}(\mathbf{x}_k))$ is converted from SDF with a Laplacian density transform from VolSDF [49]. To accelerate, points are sampled in the sparse grid where valid voxel blocks have been allocated, and the empty space is directly skipped.

3.5.2 Regularization

Eikonal regularization [2] forces SDF gradients to be close to 1,

$$\mathcal{L}_{\text{Eik}} = \left(\|\nabla f_{\theta_d}(\mathbf{x})\| - 1\right)^2. \tag{16}$$

Similar to related works [49, 51], $\{\mathbf{x}\}$s are samples combined with ray-based samples around surfaces, and uniform samples in the sparse grids. It is worth noting that in an explicit SDF voxel grid, f_{θ_d} and ∇f_{θ_d} can be jointly computed in the same pass:

$$f_{\theta_d}(\mathbf{x}) = \sum_{\mathbf{x}_i \in \text{Nb}(\mathbf{x})} r(\mathbf{x}, \mathbf{x}_i)\theta_d(\mathbf{x}_i), \tag{17}$$

$$\nabla_{\mathbf{x}} f_{\theta_d}(\mathbf{x}) = \sum_{\mathbf{x}_i \in \text{Nb}(\mathbf{x})} \nabla_{\mathbf{x}} r(\mathbf{x}, \mathbf{x}_i)\theta_d(\mathbf{x}_i), \tag{18}$$

where $\theta_d(\mathbf{x}_i)$ are directly stored SDF values at voxel grid points $\mathbf{x}_i$, and r is the trilinear interpolation ratio function that is a polynomial with closed-form derivatives. This circumvents costly double backward pass for autodiff gradient estimation [49, 51], therefore speeds up training both by reducing computation burden and allowing larger batch size.

3.5.3 Differentiable Continuous Semantic CRF

Through differentiable volume rendering, we have achieved fine geometry reconstructions. We want to further sharpen the details at the boundaries of objects (*e.g.*, at the intersection of a cabinet and the floor). We resort to CRFs for finetuning all the properties, including colors, normals, and labels. Unlike conventional CRFs where energy functions are defined on discrete nodes, we propose to leverage our data structure and devise a continuous CRF to integrate energy over the surface

$$E(\mathbb{S}) = \int_{\mathbb{S}} \psi_u(\mathbf{x}) d\mathbf{x} + \int_{\mathbb{S}} \int_{\mathbb{S}} \psi_p(\mathbf{x}_i, \mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j, \tag{19}$$

where $\mathbf{x} \in \mathbb{S}$ denotes a point on the surface. ψ_u and ψ_p denote unary and pairwise Gibbs energy potentials. Following Krahenbuhl *et al.* [16], we adopt the Gaussian edge potential

$$\psi_p(\mathbf{x}_i, \mathbf{x}_j) = \mu_{\text{prop}}(\mathbf{x}_i, \mathbf{x}_j) \exp\left(-(\mathbf{f}_i - \mathbf{f}_j)^T \Lambda (\mathbf{f}_i - \mathbf{f}_j)\right), \tag{20}$$

where μ_{prop} denotes a learnable compatibility function of a node property (*e.g.* normal), exp computes the consistency strength between nodes from feature distances with the precision matrix Λ. A feature $\mathbf{f}_i$ concatenates 3D positions, colors, normals, and label logits queried at $\mathbf{x}_i$. We approximate the integration over the surface with Monte Carlo sampling by finding zero-crossings from random camera viewpoints.

The variational inference of the Gibbs energy potential with the mean-field approximation results in a simple update equation

$$Q(\mathbf{x}_i)^+ \propto \exp\left(-\psi_u(\mathbf{x}_i) - \sum_j \psi_p(\mathbf{f}_i, \mathbf{f}_j) Q(\mathbf{x}_j) \right). \tag{21}$$

Note that the summation in Eq. 21 is over all the sample points and computationally prohibitive. Thus, we use a high-dimensional permutohedral lattice convolution [1] to accelerate the message passing, driven by our collision-free hash map at high dimensions.

For each of the target properties $\text{prop} \in \{\text{color}, \text{normal}, \text{label}\}$, we define a loss $\mathcal{L}_{\text{prop}} = D_f\left(\mathbf{x}_{\text{prop}} \| Q(\mathbf{x}_{\text{prop}})\right)$ with f-diveregence, conditioned on the remaining properties plus the 3D positions. A joint loss is defined to optimize all the properties:

$$\mathcal{L}_{\text{CRF}} = \lambda_{\text{color}}\mathcal{L}_{\text{CRF}}^{\text{color}} + \lambda_{\text{normal}}\mathcal{L}_{\text{CRF}}^{\text{normal}} + \lambda_{\text{label}}\mathcal{L}_{\text{CRF}}^{\text{label}}. \tag{22}$$

3.5.4 Optimization

The overalls loss function at refinement stage is

$$\mathcal{L} = \mathcal{L}_c + \lambda_d\mathcal{L}_d + \lambda_n\mathcal{L}_n + \lambda_{\text{Eik}}\mathcal{L}_{\text{Eik}} + \mathcal{L}_{\text{CRF}}. \tag{23}$$

We optimize the grid parameters $\{\theta_d, \theta_c, \theta_s\}$ with RMSProp starting with a learning rate 10^{-3}, and an exponential learning rate scheduler with $\gamma = 0.1$.

4. Experiments

4.1. Setup

We follow Manhattan SDF [14] and evaluate on 4 scenes from ScanNet [7] and 4 scenes from 7-scenes [13] in evaluation. We use reconstruction's F-score as the major metric, along with distance metrics (accuracy, completeness), precision, and recall. We compare against COLMAP [36], NeRF [23], UNISURF [31], NeuS [45], VolSDF [49], Manhattan SDF [14], and MonoSDF [51]. We train MonoSDF to obtain output mesh. For the rest of the compared approaches, we reuse reconstructions provided by the authors from Manhattan SDF [14], and evaluate them against high-resolution ground truth via TSDF fusion. The evaluation metric and implementation details are in supplementary.

For geometric priors, unlike MonoSDF [51] that generates monocular cues from 384×384 center crops, we follow DPT [33]'s resizing protocol and adapt Omnidata [11] to obtain 480×640 full resolution cues.

In all the experiments, we use a 8^3 voxel block grid with a voxel size 1.5cm. At each step, we randomly sample 1024 rays per image with a batch size of 64. Due to the reasonable geometric initialization, the loss usually drops drastically within 2×10^3 iterations, and converges at 10^4 iterations, therefore we terminate training at 10^4 steps for all scenes. Thanks to the efficient data structure, accelerated ray marching, and closed-form SDF gradient computation, it takes less than 30 mins to reconstruct a scene on a mid-end computer with an NVIDIA RTX 3060 GPU and an Intel i7-11700 CPU.

Table 1. Train and inference time (per image) analysis on the ScanNet scene 0084. Our approach both trains and evaluates faster.

Method	Train (h)	Inference (s)
NeuS [45]	6.64	28.32
VolSDF [49]	8.33	29.64
ManhattanSDF [14]	16.68	28.49
MonoSDF (MLP) [51]	9.89	33.80
MonoSDF (Grid) [51]	4.36	19.13
Ours	**0.47**	**0.25**

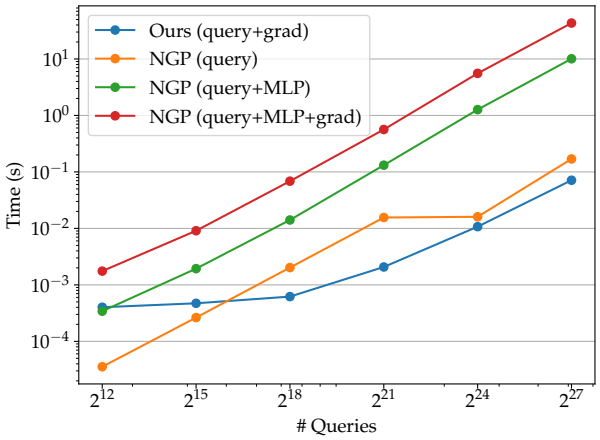


Figure 8. Query time comparison between ours and NGP-grid, lower is better. For end-to-end query, ours is two magnitudes faster, and maintains a high efficiency with a large number of point query. For the grid query operation itself, ours also have a better performance than multiresolution feature grids.

4.2. Runtime Comparison

We first profile the SDF query time given a collection of points on the aforementioned machine. Specifically, we sample $k^3, k \in \{2^4, \cdots, 2^9\}$ grid points of 3D resolution k, query the SDF and their gradients, and compare the run time. This is frequently used for Marching Cubes [19] (requiring SDFs) and global Eikonal regularization [2] (requiring SDF gradients). We compare against MonoSDF's NGP-grid backbone that uses the multi-resolution grid from Instant-NGP [24]. In this implementation, three steps are conducted to obtain required values: query from the feature grid; SDF inference from an MLP; SDF grad computation via autograd. In contrast, ours allows its explicit computation in one forward pass, see Eq. 18. Fig. 8 shows the breakdown time comparison.

We also show the training and inference time comparison in Table 2. Due to the fine initialization and sparse data structure with accelerated ray sampling, our approach can complete training in less than half an hour, and allows fast rendering of color and depth at inference time.

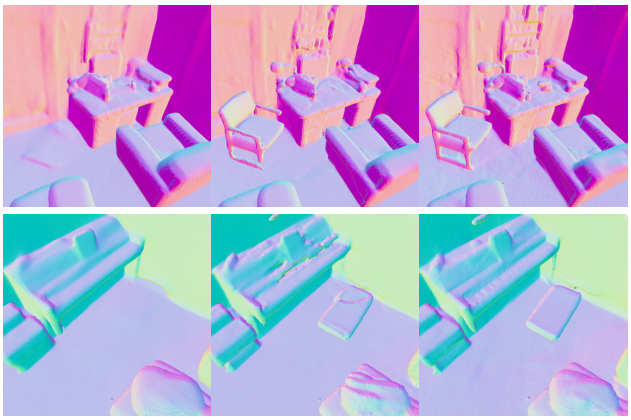
Table 2. Quantitative comparison of reconstruction quality. While being much faster, our approach is comparable to the state-of-the-art MonoSDF [51] on ScanNet [7] and better on 7-scenes [13].

Method	ScanNet					7-Scenes				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
COLMAP [36]	0.074	0.239	0.602	0.363	0.442	**0.069**	0.417	**0.536**	0.202	0.289
NeRF [23]	0.605	0.178	0.186	0.302	0.225	0.573	0.321	0.159	0.085	0.083
UNISURF [31]	0.497	0.167	0.224	0.327	0.265	0.407	0.136	0.195	0.301	0.231
NeuS [45]	0.166	0.221	0.296	0.237	0.262	0.151	0.247	0.313	0.229	0.262
VolSDF [49]	0.378	0.139	0.284	0.330	0.301	0.285	0.140	0.220	0.285	0.246
ManhattanSDF [14]	0.081	0.099	0.626	0.544	0.581	0.112	0.133	0.351	0.326	0.336
MonoSDF (MLP) [51]	**0.031**	0.057	0.783	0.652	0.710	0.097	0.192	0.441	0.311	0.361
MonoSDF (Grid) [51]	0.034	**0.046**	**0.796**	**0.711**	**0.750**	0.113	0.100	0.433	0.392	0.411
Ours (Scale Optim.)	0.058	0.064	0.655	0.605	0.627	0.151	0.080	0.367	0.462	0.409
Ours (+ Volume Rendering)	0.045	0.060	0.774	0.667	0.714	0.140	0.081	0.417	0.450	0.433
Ours (+ CRF)	0.042	0.056	0.751	0.678	0.710	0.136	**0.079**	0.436	**0.475**	**0.454**

4.3. Reconstruction Comparison

The comparison of reconstruction accuracy can be seen in Table 2. We can see that our approach achieves high accuracy at initialization that surpasses various baselines. With volume rendering and CRF refinement, it reaches comparable accuracy to the state-of-the-art MonoSDF [51] on ScanNet scenes, and achieves better results on 7-scenes. The last three rows serve as the ablation study, showing a major gain from volume rendering followed by a minor refinement gain from CRF.

We also demonstrate qualitative scene-wise geometric reconstruction in Fig. 2, and zoomed-in details in Fig. 9. It is observable that while achieving similar global completeness, our method enhances details thanks to the adaptive voxel grid and direct SDF mapping from coordinates to voxels.



(a) MonoSDF-MLP (b) MonoSDF-Grid (c) Ours

Figure 9. Detail comparisons between our method and current state-of-the-art neural implicit method MonoSDF [51]. We preserve better geometry details while being faster.

The control experiments of CRF's incorporated properties are visualized in Fig. 10, where we see that semantic labels and normals have the highest impact on reconstruction quality. Colors, on the other hand, have a lower impact mostly due to the prevalent appearance of motion blurs and exposure changes in the benchmark dataset. The same reason also affects feature-based SfM and monocular depth estimate and leads to reduced performance of our approach on certain sequences, see supplementary. We plan to incorporate more advanced semi-dense reconstruction [42, 43] for robust depth prior estimate.

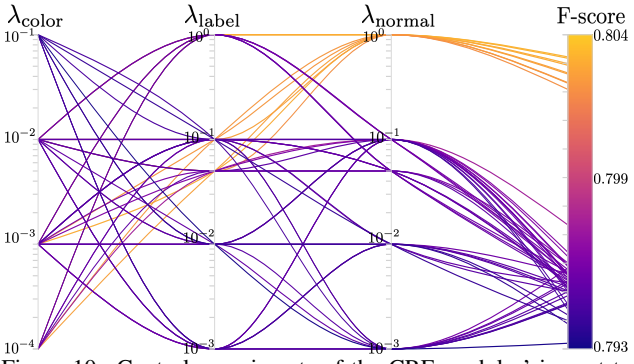


Figure 10. Control experiments of the CRF modules' impact to final reconstruction quality on scene 0084, see Eq. 22.

5. Conclusion

We propose an efficient monocular scene reconstruction system. Without an MLP, our model is built upon a differentiable globally sparse and locally dense data structure allocated around approximate surfaces. We develop a scale calibration algorithm to align monocular depth priors for fast geometric initialization, and apply direct refinement of voxel-level SDF and colors using differentiable rendering. We further regularize the voxel-wise properties with a high-dimensional continuous CRF that jointly refines color, geometry, and semantics in 3D. Our method is $10\times$ faster in training and $100\times$ faster in inference, while achieving similar reconstruction accuracy to state-of-the-art.

References

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer graphics forum*, volume 29, pages 753–762. Wiley Online Library, 2010. 2, 7

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *CVPR*, pages 2565–2574, 2020. 3, 6, 7

[3] 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 *ICCV*, pages 5855–5864, 2021. 3

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 3

[5] Ronald Clark. Volumetric bundle adjustment for online photorealistic scene capture. In *CVPR*, pages 6124–6132, 2022. 3

[6] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 3, 4

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 2, 7, 8, 13

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM TOG*, 36(4):1, 2017. 3

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, June 2022. 3

[10] Wei Dong, Yixing Lao, Michael Kaess, and Vladlen Koltun. Ash: A modern framework for parallel spatial hashing in 3d perception. *IEEE TPAMI*, 2022. 1, 3, 4, 5

[11] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, pages 10786–10796, 2021. 1, 2, 3, 4, 7, 11

[12] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, pages 5501–5510, 2022. 1, 3, 5

[13] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, October 2013. 7, 8, 14

[14] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *CVPR*, pages 5511–5520, 2022. 1, 2, 3, 7, 8, 11, 12, 13

[15] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 4

[16] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*, 24, 2011. 2, 6

[17] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2, 3

[18] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, pages 4521–4530, 2019. 3

[19] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 3, 7, 11

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 11

[21] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 international conference on 3D vision (3DV)*, pages 32–41. IEEE, 2018. 5

[22] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982. 3

[23] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 7, 8, 13

[24] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 1, 3, 5, 7

[25] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015. 11

[26] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, pages 414–431. Springer, 2020. 3

[27] Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. Openvdb: an open-source data structure and toolkit for high-resolution volumes. In *Acm siggraph 2013 courses*, pages 1–1. 2013. 3

[28] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 3, 4, 5, 11

[29] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, page ICCV. IEEE, 2011. 3

[30] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale us-

ing voxel hashing. *ACM TOG*, 32(6):1–11, 2013. 2, 3, 4, 5

[31] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, pages 5589–5599, 2021. 1, 7, 8, 13

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 2

[33] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021. 1, 2, 3, 6, 7, 11

[34] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 2020. 3

[35] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, pages 12892–12901, 2022. 3

[36] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 1, 2, 3, 4, 7, 8, 11, 13

[37] P. Stotko, S. Krumpen, M. B. Hullin, M. Weinmann, and R. Klein. SLAMCast: Large-Scale, Real-Time 3D Reconstruction and Streaming for Immersive Multi-Client Live Telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2102–2112, May 2019. 2

[38] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, 2021. 3

[39] 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

[40] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *CVPR*, pages 15598–15607, 2021. 1, 3

[41] 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. 2021. 3

[42] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 34:16558–16569, 2021. 8, 12

[43] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *arXiv preprint arXiv:2208.04726*, 2022. 8, 12

[44] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, pages 2626–2634, 2017. 3

[45] 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. 1, 2, 3, 7, 8, 13

[46] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, pages 5610–5619, 2021. 3

[47] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022. 4

[48] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783, 2018. 1, 3

[49] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 34:4805–4815, 2021. 1, 2, 3, 6, 7, 8, 13

[50] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 33:2492–2502, 2020. 3

[51] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. 2022. 1, 2, 3, 4, 6, 7, 8, 11, 12, 13, 14

[52] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *CVPR*, pages 5449–5458, 2022. 3

[53] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 11

[54] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, pages 12786–12796, 2022. 3

Supplementary

Depth Scaler Optimization

Our system adopts monocular depth map predictions from off-the-shelf networks [11] using the DPT backbone [33]. However, these depth priors are not metric and the scale of each depth prediction is independent of others. Thus, we define the unary and binary (pairwise) constraints to estimate consistent metric scales.

Unary Constraints

Our pipeline relies on COLMAP's [36] sparse reconstruction for unary constraints. COLMAP supports sparse reconstruction with or without poses. Both modes start with SIFT [20] feature extraction and matching. The *with pose* mode then runs triangulation, while the *without pose* mode runs bundle adjustment to also estimate poses. *With pose* mode usually runs within 1 min, while the *without pose* mode often finishes around 5 mins for a sequence with several hundred frames. While our system integrates both modes, for fair comparison on the benchmark datasets, we adopt the *with pose* mode in quantitative experiments where ground truth poses from RGB-D SLAM are given. Fig. 11 shows the sparse reconstructions from the *with pose* mode.

Binary Constraints

Once we have camera poses and the sparse reconstruction, we can define which triangulated feature points are visible to which cameras (covisible). Thus, we can create pairwise reprojection constraints between frames, similar to loop closures in the monocular SLAM context [25]. We directly retrieve the feature matches obtained by COLMAP, and setup such frame-to-frame covsibility constraints. Fig. 11 shows the covisibility matrices, where entry (i, j) indicates the number of covisible features between frame i and j. They are used to establish binary constraints between frames for refining monocular depth scales.

Volumetric Fusion

Eq. 9 in the main paper shows the least squares to initialize voxel-wise SDF. The more detailed implementation follows KinectFusion [28], where a truncation function ψ is used to reject associations.

$$\theta_d(\mathbf{v}) = \arg\min_d \sum_i \|d - \psi(d^o, \mu)\|^2, \tag{24}$$

$$d^o = d_{\mathbf{v}_{\to i}} - \mathcal{D}_i(\mathbf{p}_{\mathbf{v}_{\to i}})\phi_i(\mathbf{p}_{\mathbf{v}_{\to i}}), \tag{25}$$

$$\psi(x, \mu) = \min(x, \mu), \tag{26}$$

where μ is the truncation distance. μ is associated with the *Dilate* operation and voxel block resolution in Eq. 7-8 in the main paper. Formally, we define

$$\text{Dilate}_R(\mathbf{x}) = \left\{ \mathbf{x}_i \mid \left\| \mathbf{x}_i - \left\lfloor \frac{\mathbf{x}}{L} \right\rfloor \right\|_0 \leq R \right\}, \tag{27}$$

where L is the voxel block size, $\mathbf{x}_i$ are quantized grid points around, and R is the dilation radius. We use $R = 2$ (corresponding to two 8^3 voxel blocks) to account for the uncertainty around surfaces from the monocular depth prediction. Correspondingly, we use $\mu = L \cdot R$ to truncate the SDF.

The volumetric fusion runs at 50 Hz with RGB and SDF fusion, and at 30 Hz when additional semantic labels are also fused, hence serves as a fast initializer.

Hyper Parameters

We followed [51]'s hyperparameter choices and used $\lambda_d = 0.1, \lambda_n = 0.05$ for the rendering loss.

For regularizors, we obtained from hyper param sweeps from the 0084 scene of ScanNet that $\lambda_{\text{eik}} = 0.1$ for the Eikonal loss, and $\lambda_{\text{color}} = 10^{-3}, \lambda_{\text{label}} = 0.1, \lambda_{\text{normal}} = 1$ for the CRF loss.

In Gaussian kernels, we fix $\sigma_{\text{sdf}} = 1.0$ and $\sigma_{\text{color}} = 0.1$.

Evaluation

Metrics

We follow the evaluation protocols defined by ManhattanSDF [14], where the metrics between predicted point set P and ground truth point set P^* are

$$D(p, p^*) = \|p - p^*\|, \tag{28}$$

$$\text{D}_{\text{Acc}}(P, P^*) = \underset{p \in P}{\text{mean}} \min_{p^* \in P^*} D(p, p^*), \tag{29}$$

$$\text{D}_{\text{Comp}}(P, P^*) = \underset{p^* \in P^*}{\text{mean}} \min_{p \in P} D(p, p^*), \tag{30}$$

$$\text{Prec}(P, P^*) = \underset{p \in P}{\text{mean}} \Big(\big(\min_{p^* \in P^*} D(p, p^*) \big) < T \Big), \tag{31}$$

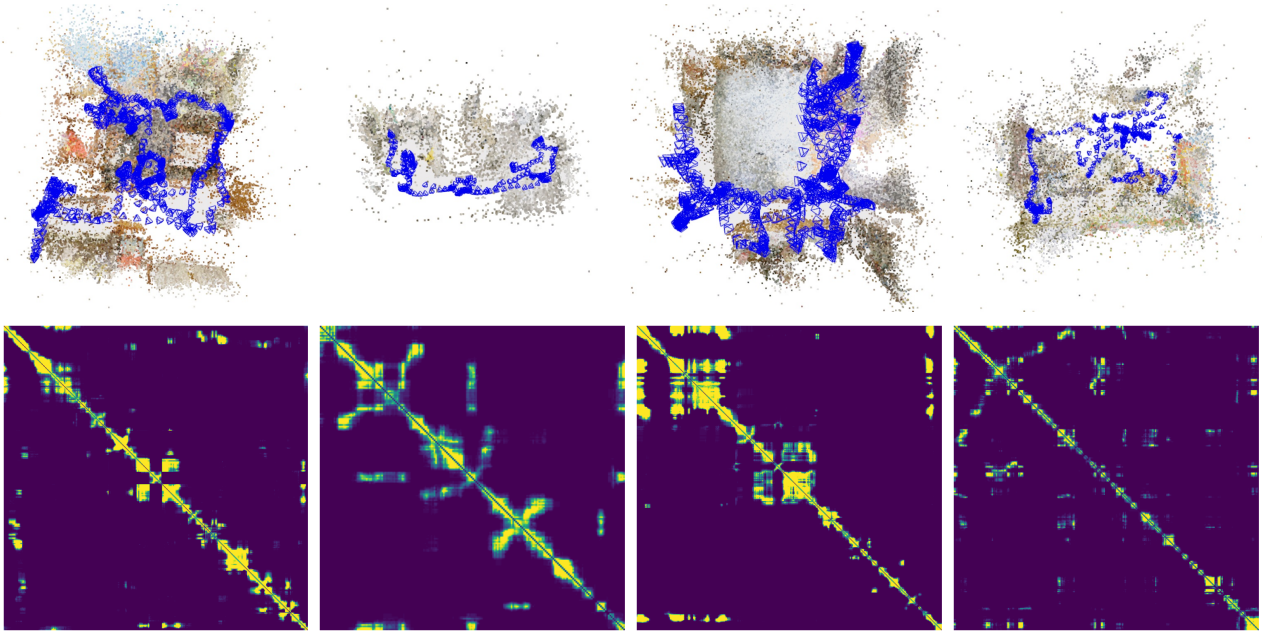
$$\text{Recall}(P, P^*) = \underset{p^* \in P^*}{\text{mean}} \Big(\big(\min_{p \in P} D(p, p^*) \big) < T \Big), \tag{32}$$

$$\text{F-score}(P, P^*) = \frac{2 \cdot \text{Prec} \cdot \text{Recall}}{\text{Prec} + \text{Recall}}, \tag{33}$$

where $T = 5$cm.

Generation of P and P^*

We follow previous works [14, 51] that applied TSDF refusion to generate P for evaluation: use Marching Cubes [19] to generate a global mesh; render depth map from mesh at selected viewpoints to crop points out of viewports; apply TSDF fusion [53] to obtain the final mesh and point cloud P. For fairness, we render depth at the resolution 480×640 for all approaches to be consistent with input (in contrast to MonoSDF that uses 968×1296 in their released evaluation code), and conduct refusion to a voxel grid at the resolution of 1cm.



0050 0084 0580 0616

Figure 11. Sparse reconstruction and covisibility matrix of ScanNet scenes selected by ManhattanSDF [14].

To ensure the same surface coverage, we generate ground truth P^* at the same viewpoints with the same image and voxel resolution, only replacing rendered depth with ground truth depth obtained by an RGB-D sensor.

Additional Experimental Results

Ablation of scale optimization

To further illustrate the necessity of per-frame scale optimization, we show quantitative reconstruction results without scale optimization in Table 3. Here, volumetric fusion is conducted on an estimated single scale factor across all frames between monocular depth and SfM, resulting in poor initial reconstruction.

Table 3. Initial reconstruction results without per-frame scale optimization (*c. f.* Ours (Init) in Table 4-5.)

	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
ScanNet	0.42	0.19	0.13	0.28	0.17
7-Scenes	0.36	0.12	0.19	0.43	0.26

Fusion and Refinement

Please see video supplementary for the incremental fusion from scaled depth, and the refinement stage that converges to general shapes within several hundred steps.

Scene-wise statistics on ScanNet

We use reconstructed mesh provided by ManhattanSDF [14], and report scene-wise statistics in Table 4. Reconstructions and corresponding ground truths are shown in Fig. 12.

It is observable that our reconstructions have low error at fine details with rich textures (*e.g.* 0050, furniture in 0580), but problems exist at texture-less regions (*e.g.* walls in 0580 and 0616, floor in 0084) due to the inaccurate scale estimate from sparse reconstructions. We plan to improve these by learning-based sparse or semi-dense reconstruction, *e.g.* [42, 43].

Scene-wise statistics on 7-scenes

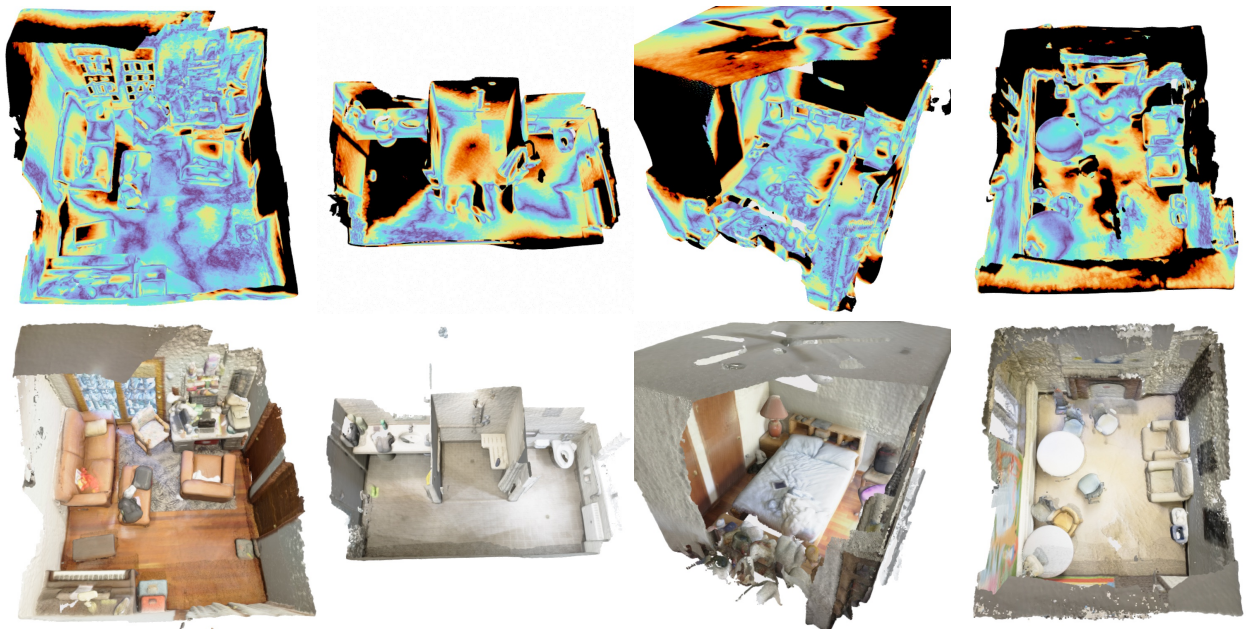
The reconstructed mesh and scene-wise statistics are not provided by ManhattanSDF [14] for COLMAP, NeRF, UNISURF, NeuS, VolSDF, and ManhattanSDF. Therefore, we reuse their reported averages as a reference in the main paper. Here we report scene-wise numbers in Table 5 for the state-of-the-art MonoSDF [51] and our method. Reconstructions and ground truths are in Fig. 13.

7-scenes have challenging camera motion patterns and complex scenes, thus the overlaps between viewpoints are small, leading to reduced accuracy for all the approaches. Although our approach produces less accurate floor and walls with fewer features, it achieves fine reconstruction of desktop objects in general.

Table 4. Scene-wise quantitative results on ScanNet.

Method	0050					0084				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
COLMAP [36]	0.049	0.129	0.707	0.531	0.607	0.032	0.121	0.807	0.577	0.673
NeRF [23]	0.704	0.081	0.215	0.517	0.304	0.733	0.248	0.157	0.213	0.181
UNISURF [31]	0.432	0.087	0.309	0.482	0.376	0.594	0.242	0.218	0.339	0.266
NeuS [45]	0.091	0.103	0.528	0.455	0.489	0.231	0.365	0.159	0.090	0.115
VolSDF [49]	0.071	0.071	0.600	0.599	0.599	0.507	0.165	0.163	0.247	0.196
ManhattanSDF [14]	0.032	0.050	0.849	0.755	0.800	**0.029**	**0.041**	**0.822**	**0.784**	**0.802**
MonoSDF (MLP) [51]	**0.025**	0.054	0.865	0.713	0.781	0.036	0.048	0.700	0.646	0.672
MonoSDF (Grid) [51]	0.027	0.045	0.854	0.764	0.807	0.035	0.043	0.796	0.774	0.785
Ours (Init)	0.034	0.051	0.775	0.684	0.727	0.047	0.048	0.705	0.725	0.715
Ours (+Rendering)	0.026	**0.044**	0.875	0.780	0.825	0.038	0.046	0.762	0.748	0.755
Ours (+CRF)	0.026	**0.044**	**0.880**	**0.788**	**0.832**	0.043	0.043	0.750	0.780	0.765

Method	0580					0616				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
COLMAP [36]	0.169	0.300	0.204	0.112	0.145	0.045	0.406	0.689	0.230	0.344
NeRF [23]	0.402	0.186	0.125	0.216	0.159	0.582	0.196	0.249	0.263	0.256
UNISURF [31]	0.392	0.192	0.131	0.188	0.155	0.571	0.148	0.237	0.300	0.265
NeuS [45]	0.206	0.275	0.167	0.114	0.135	0.137	0.140	0.330	0.289	0.308
VolSDF [49]	0.197	0.183	0.197	0.189	0.193	0.736	0.129	0.176	0.284	0.217
ManhattanSDF [14]	0.205	0.240	0.149	0.124	0.135	0.058	0.066	0.684	0.513	0.586
MonoSDF (MLP) [51]	**0.025**	**0.040**	**0.867**	**0.759**	**0.809**	0.039	0.087	0.702	0.488	0.576
MonoSDF (Grid) [51]	0.039	0.048	0.718	0.661	0.688	**0.033**	**0.048**	**0.815**	**0.646**	**0.721**
Ours (Init)	0.076	0.059	0.574	0.582	0.578	0.076	0.097	0.566	0.427	0.487
Ours (+Rendering)	0.070	0.080	0.760	0.636	0.692	0.046	0.070	0.699	0.504	0.586
Ours (+CRF)	0.046	0.050	0.707	0.682	0.694	0.057	0.080	0.659	0.504	0.571



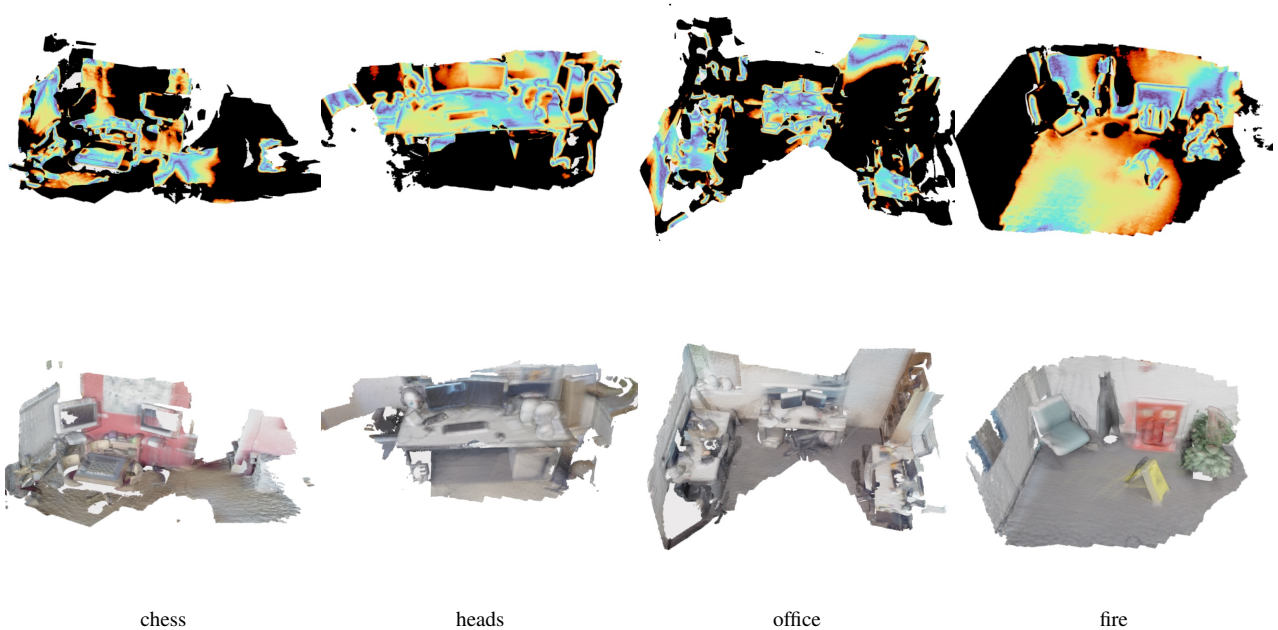
0050 0084 0580 0616

Figure 12. Error heatmap from our reconstruction (first row) to groundtruth (second row) for each scene in ScanNet [7]. Points are colorized by distance error ranging from 0 (blue) to 5cm (red) to its nearest neighbor in ground truth. Points with error larger than 5cm are regarded as outliers and colored in black.

Table 5. Scene-wise quantitative results on 7-Scenes.

Method	chess					heads				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
MonoSDF (MLP) [51]	0.160	0.390	0.250	0.132	0.173	**0.068**	0.188	**0.586**	0.353	0.440
MonoSDF (Grid) [51]	**0.113**	0.143	0.324	0.267	0.293	0.133	0.099	0.305	0.327	0.315
Ours (Init)	0.164	0.108	0.278	0.350	0.310	0.186	0.083	0.288	0.401	0.335
Ours (+Rendering)	0.147	0.111	0.367	0.389	0.378	0.074	0.062	0.543	0.568	0.555
Ours (+CRF)	0.147	**0.107**	**0.368**	**0.391**	**0.379**	0.071	**0.057**	0.559	**0.626**	**0.591**

Method	office					fire				
	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Prec ↑	Recall ↑	F-score ↑
MonoSDF (MLP) [51]	**0.087**	0.128	0.338	0.236	0.278	0.075	0.064	**0.592**	0.522	**0.555**
MonoSDF (Grid) [51]	0.147	0.077	**0.539**	0.471	**0.503**	**0.061**	0.081	0.564	0.504	0.533
Ours (Init)	0.168	**0.068**	0.398	**0.483**	0.436	0.087	**0.058**	0.503	**0.616**	0.554
Ours (+Rendering)	0.180	0.081	0.330	0.400	0.362	0.160	0.072	0.426	0.445	0.435
Ours (+CRF)	0.164	0.080	0.340	0.400	0.367	0.162	0.068	0.474	0.490	0.482



chess heads office fire

Figure 13. Error heatmap from our reconstruction (first row) to groundtruth (second row) for each scene in 7-Scenes [13]. The colorization is the same as Fig. 12.