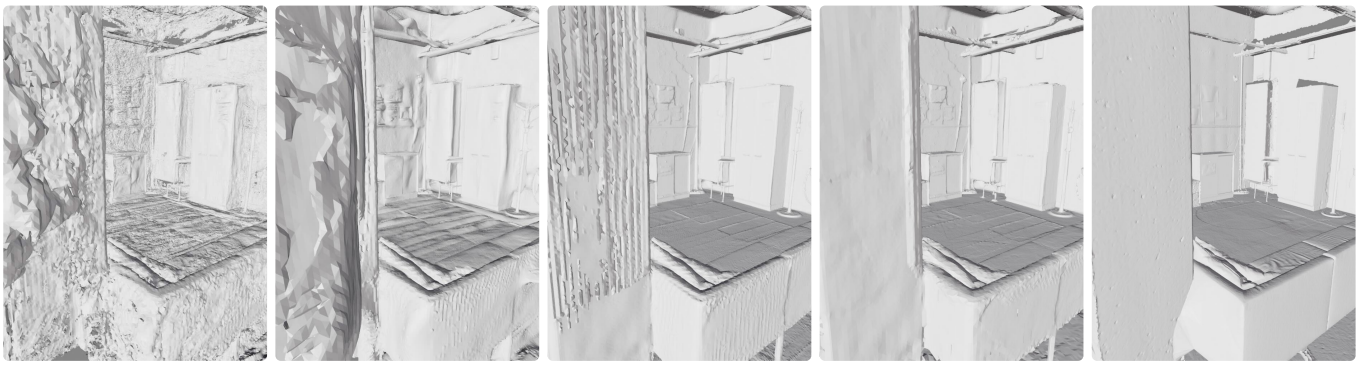
AGS-Mesh: Adaptive Gaussian Splatting and Meshing with Geometric Priors for Indoor Room Reconstruction Using Smartphones

Xuqian Ren[1‡] Matias Turkulainen[2] Jiepeng Wang[3†] Otto Seiskari[4]
Iaroslav Melekhov[2] Juho Kannala[2,5] Esa Rahtu[1]
[1] Tampere University, [2] Aalto University, [3] University of Hong Kong,
[4] Spectacular AI, [5] University of Oulu



(a) 3DGS [25] (b) 2DGS [19] (c) DN-Splatter [42] (d) 2DGS + Ours (e) Reference mesh

Figure 1. We present AGS-Mesh, a method that adaptively integrates geometric priors into Gaussian Splatting for indoor room reconstruction using a mobile device. We enhance existing Gaussian Splatting based methods (a), (b), (c) by filtering inconsistent prior estimates during optimization and utilize an IsoOctree based meshing strategy that recovers a smoother scene surface with higher detail (d).

Abstract

Geometric priors are often used to enhance 3D reconstruction. With many smartphones featuring low-resolution depth sensors and the prevalence of off-the-shelf monocular geometry estimators, incorporating geometric priors as regularization signals has become common in 3D vision tasks. However, the accuracy of depth estimates from mobile devices is typically poor for highly detailed geometry, and monocular estimators often suffer from poor multi-view consistency and precision. In this work, we propose an approach for joint surface depth and normal refinement of Gaussian Splatting methods for accurate 3D reconstruction of indoor scenes. We develop supervision strategies that adaptively filters low-quality depth and normal estimates by comparing the consistency of the priors during optimization. We mitigate regularization in regions where prior estimates have high uncertainty or ambiguities. Our filtering strategy and optimization design demonstrate significant improvements in both mesh estimation and novel-view synthesis for both 3D and 2D Gaussian Splatting-based methods on challenging indoor room datasets. Furthermore, we explore the use of alternative meshing strategies for finer geometry extraction. We develop a scale-aware meshing strategy inspired by TSDF and octree-based isosurface extraction, which recovers finer details from Gaussian models compared to other commonly used open-source meshing tools. Our code is released in https://xuqianren.github.io/ags_mesh_website/.

1. Introduction

Photorealistic and geometrically accurate reconstruction of real-world indoor scenes has various applications in virtual reality, augmented reality, and video games. Traditional approaches have addressed the problem by creating textured meshes that can be rendered using conventional graphics pipelines. Depth sensors, such as high-precision 3D LiDAR scanners or Kinect sensors, are often used to aid geometric reconstruction; however, these devices are generally expensive for consumer users and require considerable technical expertise. With the rapid development of consumer mobile devices, the latest iPhone smartphones now come equipped with multiple high-resolution RGB cameras and miniaturized LiDAR sensors, making them ideal tools for millions

‡Denotes Project Lead
†Denotes Corresponding Author

of users to collect 3D data. While the low-resolution depth maps obtained from such LiDAR sensors enable accurate planar reconstruction, they struggle with highly detailed geometry and edges, as shown in Fig. 2. These inaccuracies become evident when using the low-resolution depth maps for 3D mesh reconstruction with traditional methods like volumetric fusion [9, 20].

Recent advances in differentiable inverse rendering have provided alternative 3D representations to meshes, achieving photorealistic novel-view synthesis with relatively fast rendering and training times. 3D Gaussian Splatting (3DGS)-based methods [16, 25, 42, 51, 58] represent 3D scenes with millions of differentiable 3D Gaussians, enabling high-quality and real-time rendering. However, the geometry extracted from such scenes for larger scenes is poor. 2D Gaussian Splatting (2DGS) [19] further extends 3DGS towards mesh reconstruction, showing promising results for object-centric reconstruction. However, performance on room-scale reconstruction with data captured by a mobile device is still lacking. Low-texture surfaces and sparse, outward-facing captures, common in indoor room datasets [36, 54], pose challenges and ambiguities for purely photometric-based reconstruction. Given these observations, it is clear that achieving photorealistic and geometrically accurate reconstructions of common indoor scenes using consumer devices remains an open challenge.

In this work, we extend recent state-of-the-art Gaussian Splatting approaches [19, 42] by adaptively combining low-resolution depth estimates and off-the-shelf monocular geometry estimates for high-fidelity indoor room reconstruction using a mobile device. Specifically, we design two regularization strategies to address inconsistencies between low-resolution depth maps and off-the-shelf monocular normal estimates during training. We regularize Gaussian positions using low-resolution depth maps on planar and smooth surfaces. We introduce a new depth regularization strategy, coined Depth Normal Consistency (DNC), to filter inconsistencies in depth maps by considering the consistency between normals derived from noisy depth maps and those estimated via pretrained monocular networks. Additionally, we propose an Adaptive Normal Regularization strategy (ANR) to refine normals by mitigating regularization in regions where monocular normal estimators struggle to provide accurate prior estimates.

By carefully filtering inconsistencies between various geometric estimates during training, our approach improves both novel-view synthesis and geometry extraction compared to previous methods. Furthermore, we develop a meshing strategy inspired by TSDF [9, 20] and octree-based isosurface extraction [23] methods, which accounts for the hierarchical details present in larger indoor scenes. This post-processing strategy enhances the fine details recovered from the 3D Gaussian scene without requiring additional optimization, surpassing conventional Poisson and Marching Cubes-based [27] TSDF alternatives commonly used for 3D mesh reconstruction.

We summarize our contributions with the following statements:

- We propose a novel regularization strategy for indoor room reconstruction that adaptively filters geometric priors from mobile devices and off-the-shelf monocular estimators, enhancing photorealism and geometry reconstruction from Gaussian Splatting-based methods.
- We introduce a mesh post-processing method based on adaptive TSDF and IsoOctree meshing that recovers finer scene details compared to commonly used alternatives.
- We demonstrate, through extensive experiments, improvements in both novel-view synthesis and geometry extraction for high-fidelity indoor room reconstruction.

2. Related Work

Traditional Meshing Techniques. Classical 3D reconstruction methods utilize multi-view stereo (MVS) techniques. Prior methods use RGB images as input [13, 15, 15, 43], while some integrate depth maps [9, 17, 29, 33] for 3D reconstruction. Learning-based methods [28, 37, 47, 56] extract, match, and fuse image features by utilizing neural networks, often improving the quality of reconstruction compared to classical methods. However, they struggle to accurately reconstruct casually captured scenes with poor textures, which are common in everyday scenes.

Meshing With Neural Implicit Representations. Recent neural implicit representations, particularly NeRF-based methods [3, 30, 31], excel at novel-view synthesis by encoding 3D scenes into multi-layer perceptrons (MLPs) and using volume rendering [21] to synthesize views. However, these methods struggle with accurate geometry and physical surfaces and are primarily for photorealistic rendering. Some approaches [11, 38, 48] incorporate depth supervision for better geometry, and others [35, 41] attempted to extract watertight meshes from NeRFs. However, these methods are generally limited to small synthetic objects or carefully constructed inward-facing captures.

Alternatively, Signed Distance Function (SDF)-based methods [46, 53] define surfaces as level set crossings of an implicit function. These methods produce smoother surfaces with Marching Cubes [27] but often lose detail in thin objects, perform worse in novel-view synthesis, and are costly to train on consumer GPUs.

Utilizing Geometric Priors for Meshing. MonoSDF [57] uses depth and normal predictions from a pretrained model as regularization signals to improve an SDF-based implicit model [46]. However, monocular priors can yield inconsistent predictions, with poor multi-view consistency, leading to misguided regularization signals during optimization. Neural RGB-D [2] and GO-Surf [44] address this by using real

sensor depth readings for indoor room reconstruction. Other methods [45, 55] adaptively apply normal priors, while Deb-SDF [50] and H_2O-SDF [32] manage uncertainty in priors. Choi et al. [7] also observe that noisy depth readings can hinder training and attempt to filter them as a pre-processing step. In this work, we adaptively mitigate inconsistencies in priors during optimization of Gaussian scenes to achieve better geometric reconstructions.

Meshing Using Gaussian Splatting. 3D Gaussian Splatting (3DGS) [25] significantly improves novel-view synthesis speed by direct projection and blending of Gaussian points instead of the more costly ray tracing method in NeRF. To obtain a mesh from 3DGS, follow up work have employed different techniques: optimizing 3D Gaussians along with mesh optimization methods [16] or by replacing 3D Gaussians with 2D surfels [10, 19] to better align Gaussians with the surface. GOF [58] learns a Gaussian opacity field and directly extracts a mesh from it using tetrahedral grids. Although these methods [16, 49] show promising performance on object-level data, realistic indoor room reconstruction remains challenging. DN-Splatter [42] demonstrated that sensor depth and monocular normals can help in extracting meshes from larger indoor scenes. However, noisy sensor depth from mobile devices and inconsistencies in priors can harm optimization. VCR-GauS [6] addresses this by learning a confidence term to weigh normal regularization. In our work, we focus on achieving 3D room reconstruction by adaptively integrating noisy geometric priors into Gaussian Splatting frameworks. This approach facilitates real-time photorealistic image rendering, accelerates training for mesh generation, and more effectively utilizes geometric priors.

Mesh Extraction Algorithms. We briefly summarize post-processing techniques for mesh extraction from 3D representations in the closest prior work. To extract a mesh from a Gaussian scene, 2DGS [19] uses a well-established TSDF fusion algorithm from [59]. SuGaR [16] utilizes Poisson reconstruction [22] to reconstruct a watertight mesh from an oriented point cloud. GOF [58] establishes tetrahedral grids around each Gaussian and applies Marching Tetrahedra [39] to extract a triangle mesh. In this work, we modify the classical TSDF with a scale-aware aspect and replace the naive Marching Cubes with the unconstrained octree isosurface extraction method [24] which we refer to as *IsoOctree* in this work. This allows adapting the level of detail in the mesh to the varying precision in the scene, producing more optimized meshes with less computation.

3. Preliminaries

3.1. Geometric Priors from Handheld Devices

Depth from Kinect. Azure Kinect devices are popular in industry and research for indoor 3D reconstruction. They generate depth maps from a 1-megapixel Time-of-Flight (ToF) sensor with a 1024×1024 resolution with a per-pixel error range from 1.4mm to 12mm [26]. Depths are denoised through post-processing by considering systematic and random errors. Fig. 2 shows how the Kinect system adequately filters depth estimates for problematic regions that can occur, such as pixels being outside of the ToF illumination mask, saturated IR signal regions, and depth ambiguities due to poor sensor readings [1].

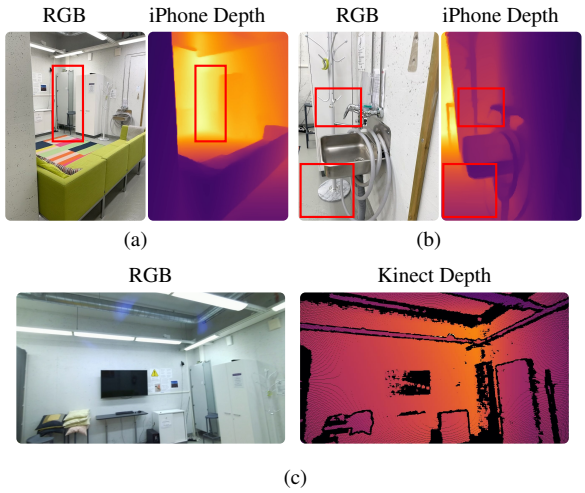


Figure 2. **Demonstration of iPhone and Kinect sensor depths.** The iPhone struggles to capture accurate depth values for (a) objects at a far distance, and (b) small objects, and edges. Instead, the Kinect sensor [1] (c) is able to filter unconfident depth values

Depth From a Mobile Phone. Some of the latest smartphones, such as the iPhone Pro versions, are integrated with small LiDAR sensors that measure distances to objects. In the case of the iPhone, the resolution of the physical sensors is usually very small (*e.g.*, only 16×16); however, sophisticated post-processing and enhancement with RGB frames are used to return depth maps with 256×192 resolution, synchronized with high-quality RGB images through developer APIs. Compared to Kinect, the depth estimates are low-resolution with an error range that can reach ± 3 cm horizontally and ± 7 mm vertically [5]. We illustrate the quality of the depth estimates in Fig. 2. We note that the low-resolution depths perform generally well for closeby objects but lose accuracy for faraway and very thin structures.

Depth and Normal Priors from Monocular Networks. Recent monocular models [4, 18, 52] can produce high-resolution metric depths and normal estimates from a single RGB image. Although these models learn from large-scale image-geometry pairs, they struggle to achieve the same accuracy in metric depth compared to physical sensors found in devices like the iPhone or Kinect. Therefore, in this work, we utilize sensor depth measurements from mobile devices for indoor room reconstruction to obtain metrically accurate reconstruction.

3.2. Gaussian Splatting

3D Gaussian Splatting, introduced in [25], explicitly represents a 3D scene with three-dimensional Gaussian primitives characterized by a center $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{1}$$

where the covariance $\boldsymbol{\Sigma}$ is composed of scaling $\boldsymbol{S}$ and rotation $\boldsymbol{R}$ components. Each primitive also encodes a color $\boldsymbol{c}$ value via spherical harmonics and an opacity o value utilized in alpha-compositing. During rendering, 3D Gaussians are projected into 2D Gaussians based on the camera pose, sorted by their z-depths, and the final pixel color is accumulated using volumetric alpha blending:

$$\hat{\mathbf{C}} = \sum_{i \in N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{2}$$

where i is the index of a Gaussian, α_i is calculated by multiplying 2D Gaussian's contribution with its opacity o.

Depth maps can be estimated from the Gaussian scene using the same discrete volume rendering equation [42]:

$$\hat{D} = \frac{\sum_i d_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j)}{\sum_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j))} \tag{3}$$

where d_i is a Gaussian's z-depth in view space, and the denominator normalizes alpha estimates. However, as noted in prior research [34], this is just an approximation for per-pixel depth estimates.

To extend 3DGS's geometric accuracy, 2DGS [19] represents the scene with flat 2D Gaussians that better capture surfaces. Each primitive is composed of center points, two principal tangential vectors t_u and t_v, and a scaling vector $\mathbf{S} = (s_u, s_v)$ to control the variance of the 2D Gaussian. With this formulation, 2DGS ensures that per-pixel depth estimates are exact and calculated explicitly through a ray-plane intersection.

In this work, we demonstrate a regularization strategy applicable to both 3D and 2D Gaussian variants by carefully supervising Gaussian positions in 3D space with sensor depth measurements and normal estimates from pretrained networks, while mitigating uncertainties in the priors. Our method serves as a plug-in module for all Gaussian-based representations to enhance geometry performance in real-world indoor reconstruction.

4. Method

Our method consists of two adaptive supervision strategies for Gaussian Splatting-based methods that effectively combine supervision signals from geometric priors obtained from mobile devices and monocular networks. An overview of the proposed approach is illustrated in Fig. 3. We first predict normal estimates from a pretrained monocular estimation model [14] for input RGB images captured with a mobile device. Next, in Section 4.1, we develop a depth regularization strategy that filters noisy sensor depth readings based on a normal consistency criterion. In Section 4.2, we carefully utilize the pretrained monocular normal estimates for normal supervision, mitigating regularization in cases where the pretrained estimates – due to multi-view inconsistencies or other inaccuracies – deviate significantly from the normal estimates derived from the geometry of the optimized Gaussian scene. We describe the overall optimization process in Section 4.3. Lastly, in Section 4.4, we propose a novel octree-based mesh extraction method that enhances surface quality and detail preservation compared to previous approaches.

4.1. Regularization with Depth Normal Consistency

We observe that depth sensor readings from mobile phones are relatively accurate for flat surfaces but tend to be bad for edges and far away objects. Instead, normal maps predicted from pretrained monocular models have clear object boundaries, which can serve as guidance for depth filtering. We propose an adaptive depth regularization method based on the consistency of normals derived from noisy depth images and those from pretrained networks. We coin this method as Depth Normal Consistency (DNC).

To derive robust normal estimates from noisy depth maps, we backproject depths $D(x, y)$ into world coordinates, and determine the K-Nearest Neighbors (KNN) [8] in world coordinates to each depth value (we set k=200). Then, a robust normal estimate $\mathbf{N}_d$ is generated per coordinate as the maximum eigenvector corresponding to the maximum eigenvalue of the eigenvalue decomposition of the covariance matrix determined from these K-Nearest points and their center.

To filter inaccurate depth estimates, we check the orientation consistency between $\mathbf{N}_d$ and $\mathbf{N}_p$ generated from pre-train model with an angle threshold τ_d for filtering:

$$D_f = \begin{cases} 0 & \text{if } \theta_d > \tau_d \\ D & \text{otherwise} \end{cases} \tag{4}$$

where

$$\theta_d = \arccos\left(\frac{\mathbf{N}_d \cdot \mathbf{N}_p}{\|\mathbf{N}_d\| \, \|\mathbf{N}_p\|}\right) \tag{5}$$

During training, we first regularize rendered depth estimates $\hat{D}$ with noisy sensor depth readings and then further refine depth estimates using the DNC filtered depths D_f:

$$\mathcal{L}_D = \begin{cases} \|\hat{D} - D\|_1 & \text{when step} < T_d \\ \|\hat{D} - D_f\|_1 & \text{otherwise} \end{cases} \tag{6}$$

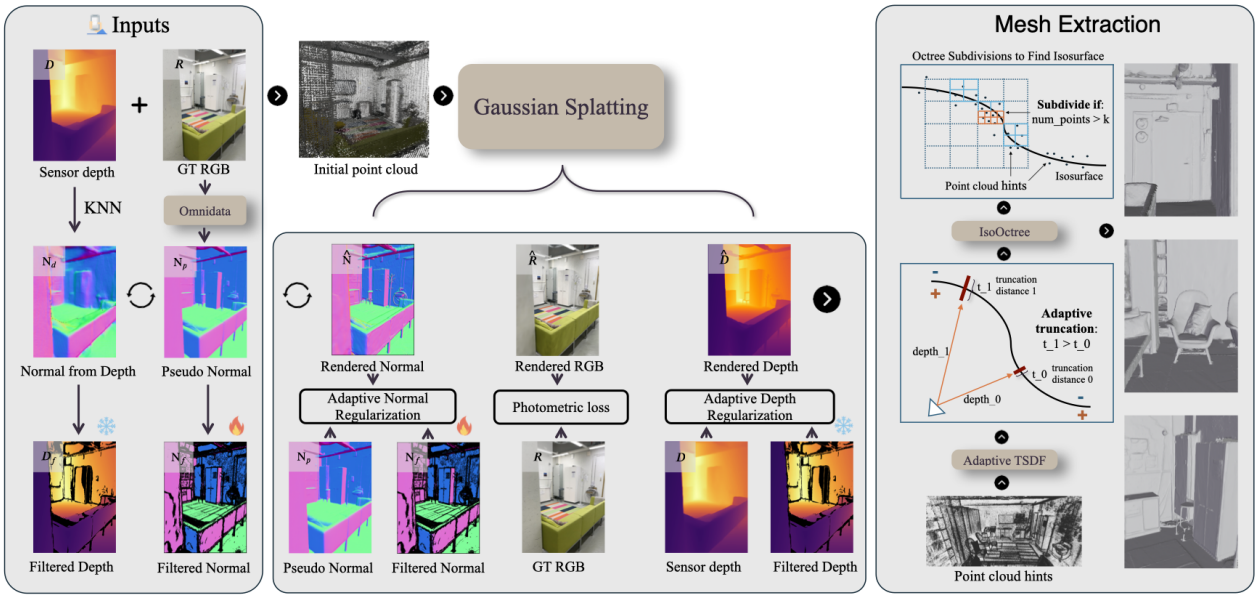


Figure 3. **Pipeline Overview**. Our approach leverages geometric consistency between normals derived from raw sensor depths and those predicted by a pretrained model to filter out noisy sensor depth data. Likewise, we compare rendered normals from a Gaussian scene with pseudo-normal estimates to dynamically filter uncertainties in normal supervision during optimization. Our adaptive depth and normal regularization terms assist various Gaussian-based frameworks in accurately reconstructing the underlying scene. Furthermore, we propose an adaptive TSDF and octree-based Marching Cubes meshing strategy enabling the extraction of smoother and more geometrically detailed meshes.

4.2. Adaptive Normal Regularization

Similar to sensor depth readings, normals predicted from pretrained models $\mathbf{N}_p$ also suffer from poor multi-view consistency and precision. To alleviate the influence raised by wrong predictions, we designed an Adaptive Normal Regularization (ANR) strategy to mitigate normal supervision in uncertain regions that hard to optimize. Rendered normals $\hat{\mathbf{N}}$ derived from the Gaussian scene are supervised with the following adaptive normal loss:

$$\mathcal{L}_{\mathbf{N}} = \begin{cases} \|\hat{\mathbf{N}} - \mathbf{N}_p\|_1 & \text{when step} < T_n \\ \|\hat{\mathbf{N}} - \mathbf{N}_f\|_1 & \text{otherwise} \end{cases} \tag{7}$$

where $\mathbf{N}_f$ are filtered normals calculated by comparing the angle difference between $\hat{\mathbf{N}}$ and $\mathbf{N}_p$ with a threshold τ_N.

$$\mathbf{N}_f = \begin{cases} 0 & \text{if } \theta_n > \tau_N \\ \mathbf{N}_p & \text{otherwise} \end{cases} \tag{8}$$

where θ_i is the angle difference between $\hat{\mathbf{N}}$ and $\mathbf{N}_p$ like Eq. (5). The ANR strategy is designed to first regularize Gaussian normals using the fully pre-trained normals $\mathbf{N}_p$, and subsequently relax the training by relying only on the multi-view consistent and more reliable filtered normal $\mathbf{N}_f$.

To allow the gradients from the normal loss during optimization to directly influence the Gaussian geometry, $\hat{\mathbf{N}}$ is estimated from rendered depth maps as in [19]:

$$\hat{\mathbf{N}}(x,y) = \frac{\nabla_x D(x,y) \times \nabla_y D(x,y)}{|\nabla_x D(x,y) \times \nabla_y D(x,y)|} \tag{9}$$

4.3. Optimization

Our DNC and ANR regularization terms can be adapted to several Gaussian Splatting based frameworks, including 3DGS and 2DGS methods. Take 3DGS [25] as an example, the final optimization loss is expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda_d \mathcal{L}_D + \lambda_n \mathcal{L}_{\mathbf{N}} \tag{10}$$

where $\mathcal{L}_{\text{color}}$ is the original RGB supervision loss containing a $\mathcal{L}_1$ and D-SSIM terms [25].

4.4. Mesh Extraction

Most prior works [16, 19, 20, 57] utilize some form of Marching Cubes [27] as a final processing step to extract a mesh. This also includes TSDF-based approaches offered in open-source tools like Open3D [59]. However, due to the uniform grid discretization in naive Marching Cubes, recovering fine details in large scenes becomes computationally infeasible, as the number of evaluated points is proportional to V/h^3 where V is the volume of the scene's bounding box and h is the voxel size, which determines the finest level of detail. Alternative methods like IsoOctree [24] utilize a hierarchical coarse-to-fine meshing strategy, where finer details can be preserved with adaptive grid divisions. In this work, we investigate adapting a hierarchical meshing algorithm in the context of Gaussian Splatting.

We build upon previous work by integrating a TSDF-like volume from multiple backprojected rendered depths. However, our approach introduces a key difference: we adjust

the truncation distance based on the current depth value. For closer objects with small depth values, the truncation distance is reduced, whereas distant points, which tend to be less reliable, have larger truncation distances. The truncation distance follows a simple linear relationship with depth. Additionally, unlike traditional TSDF methods, we incorporate rendered normal maps alongside depth maps to filter and weight the contribution of each backprojected frame based on the consistency of their normal orientations. Our depth-aware truncated TSDF method results in a TSDF volume with a dynamically varying truncation factor.

Next, we apply the IsoOctree meshing algorithm [23], which starts with a uniform grid and progressively subdivides the volume into finer regions based on a user-defined function that determines which octree nodes to expand. To achieve this, we employ a *point cloud hint*: we back-project our output depth maps from all training images into a point cloud and expand a voxel of width h if it contains at least $N_{\mathrm{e}} = 50$ points within a radius h from its center. This choice is based on the assumption that the appropriate level of detail in the reconstruction is proportional to the density of the point cloud, which correlates with the distance to the cameras and the number of camera views covering a particular region. To produce a smooth and artifact-free mesh, the curvature of the underlying isosurface should also follow a similar pattern: it should be low in regions of low point cloud density. To satisfy this property, our TSDF, which defines the isosurface, also employs a camera-distance-based heuristic. In Table 3 and Fig. 6, we show that the underlying geometry from an optimized Gaussian scene can be further refined with this IsoOctree-based method. This results in a smoother surface compared to the baseline TSDF method, which lacks both subdivision and truncation distance modulation.

5. Experiments

In this section, we demonstrate the effectiveness of our adaptive depth and normal regularization strategy and our proposed IsoOctree meshing method. We evaluate mesh reconstruction performance and novel-view synthesis quality.

Datasets. We focus on real-world indoor scenes captured using a mobile device. We select two datasets containing iPhone captures with depth data: a) MuSHRoom [36] dataset: a real-world indoor dataset with different trajectories for training and evaluation; b) ScanNet++ [54] dataset: a large scale real-world indoor dataset with high fidelity 3D geometry and RGB data.

Baselines. We compared our method to the following baselines: a) Traditional 3D reconstruction method Volumetric Fusion [9]. b) state-of-the-art NeRF-based method Nerfacto [40]; c) its depth regularized version Depth-Nerfacto with a depth supervision loss similar to DS-NeRF [12]; d) MonoSDF [57] for SDF-based implicit surface reconstruction; e) original 3DGS [25] method and its advanced open-source reimplementation Splatfacto [40]; f) the following Gaussian based models focusing on mesh reconstruction: SuGaR [16], 2DGS [19], and GOF [58]; g) and lastly, DN-Splatter [42], a 3DGS-based method that also utilizes depth and normal supervision, similar to our work.

Evaluation metrics. For mesh reconstruction evaluation, we follow the evaluation protocol from [36, 44] and report Accuracy (Acc.), Completion (Comp.), Chamfer-L_1 distance (C-L_1), Normal Consistency (NC), and F-scores (F1) with a threshold of 5cm. For novel-view synthesis, we report PSNR, SSIM, and LPIPS metrics.

Implementation details. We implement our method using two recent open-source Gaussian splatting frameworks 2DGS [19] and Splatfacto [40] (a 3DGS re-implementation). For all TSDF based meshing baselines, we use the open-source implementation from Open3D [59] similar to prior work [19] with truncation distance of 0.03, depth truncation 10, and voxel size of 0.01. More settings and implementation details can be seen in the supplementary materials.

5.1. 3D Reconstruction Evaluation

We evaluate mesh reconstruction performance against baselines for the MuSHRoom in Table 1 and Fig. 4, and ScanNet++ datasets [†]. We demonstrate that our method can outperform the traditional volumetric fusion. Additionally, we showcase the performance of our proposed IsoOctree extraction method within the 2DGS-based framework, highlighting its superior surface creation quality. It is important to note that real-world indoor room reconstruction remains a significant challenge for existing Gaussian-based frameworks that rely solely on photometric supervision. Adding sensor depth regularization greatly enhances reconstruction quality in ambiguous, textureless regions. We also compare our method to the recent DN-Splatter [42] method, which utilizes sensor depth and normal priors for regularization. Our results demonstrate that the novel adaptive depth and normal regularization terms we propose (also showcased in the ablation study Table 3) improve mesh quality by effectively filtering out uncertain priors. Additionally, our strategy decreases the overall Gaussian count in a scene while providing equal or better novel-view synthesis results (*cf*. Table 2).

5.2. Novel View Synthesis

We evaluate our regularization strategy on novel view synthesis metrics in Table 2. Rendering quality benefits from prior regularization, particularly when viewing from camera positions with less overlap with the training sequence (PSNR$+0.54/0.25$dB for the test set within the training sequence and $+0.93/0.35$dB for the test set from a wholly different camera trajectory for 2DGS/DN-Splatter). We visualize novel view synthesis examples in Fig. 5.

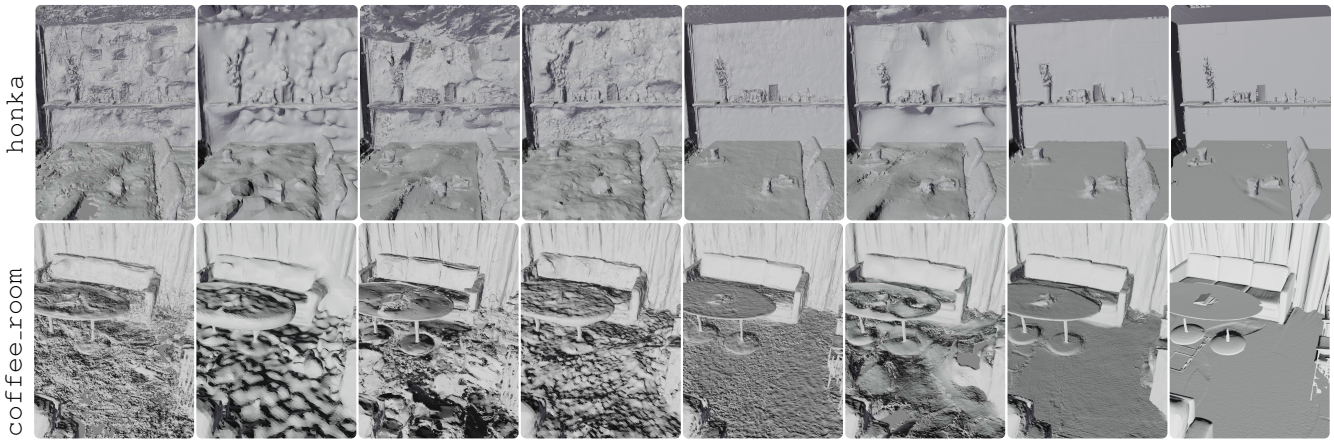
[†]Please refer to the table in the supplementary materials

Table 1. **Mesh reconstruction evaluation on MuSHRoom**. The mesh metrics are averaged over 6 scenes: "coffee_room", "honka", "kokko", "sauna", "computer", and "vr_room". The best results from each category are marked with **bold**. We also report Gaussian numbers after training on the "vr_room" scene in the last column.

	Methods		Sensor Depth	Meshing Algorithm	Accuracy ↓	Completion ↓	Chamfer-L_1 ↓	Normal Consistency ↑	F-score ↑	Num (M)
	Volumetric Fusion [9]		✓	TSDF	.0478	.0473	.0476	.7816	.8064	–
Implicit	Nerfacto [40]	NeRF	–	Poisson	.0430	.0578	.0504	.7822	.7212	–
	Depth-Nerfacto [40]		✓	Poisson	.0447	.0557	.0502	.7614	.6966	–
	MonoSDF [57]	SDF	✓	Marching-Cubes	**.0310**	**.0190**	**.0250**	**.8846**	**.9211**	–
Explicit	3DGS [25]	3DGS	–	TSDF	.0929	.0830	.0880	.6908	.4228	5.0
	SuGaR [16]		–	Poisson+IBR	**.0656**	**.0583**	**.0620**	**.8031**	**.6378**	0.7
	GOF [58]		–	Tetrahedral	.1452	.1102	.1277	.6839	.4515	3.3
	Splatfacto [40]	Splatfacto	–	Poisson	.0749	.0555	.0652	.7727	.5835	1.18
	DN-Splatter [42]		✓	Poisson	**.0239**	.0194	.0216	**.8822**	.9243	1.18
	DN-Splatter [42]		✓	TSDF	.0256	.0174	.0215	.8390	.9381	1.18
	Splatfacto [40] + Ours		✓	TSDF	.0253	**.0165**	**.0209**	.8328	**.9433**	1.11
	2DGS [19]	2DGS	–	TSDF	.0731	.0642	.0687	.8008	.6039	2.6
	2DGS [19] + Ours		✓	TSDF	.0286	.0228	.0257	**.8804**	.9053	2.5
	2DGS [19] + Ours		✓	SDF + IsoOctree (Ours)	**.0249**	**.0210**	**.0229**	.8754	**.9146**	2.5

Table 2. **Novel view synthesis evaluation** on the MuSHRoom dataset. The reported results are based on two distinct evaluation datasets: a test set obtained by uniformly sampling every 10 frames within the same training sequence, and a test set obtained from a completely different camera trajectory with no overlap with the training sequence. Results are averaged over 6 scenes.

	Method		Sensor Depth	Test within a sequence			Test with a different sequence		
				PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Implicit	Nerfacto [40]	NeRF	–	20.86	.7859	.2321	20.52	.7705	.2560
	Depth-Nerfacto [40]		✓	21.24	.7832	.2414	20.87	.7682	.2643
	MonoSDF [57]	SDF	✓	20.68	.7357	.3590	19.08	.7132	.3820
Explicit	3DGS [25]	3DGS	–	22.65	.8286	.1366	20.19	.7574	.1984
	SuGaR [16]		–	20.52	.7740	.2427	18.18	.7125	.2959
	GOF [58]		–	19.03	.7845	.2189	17.76	.7216	.2816
	Splatfacto [40]	Splatfacto	–	24.47	.8465	.1358	21.66	.7887	.1922
	DN-Splatter [42]		✓	24.58	.8558	.1293	21.89	.7984	.1797
	Splatfacto [40] + Ours		✓	**24.83**	**.8589**	**.1129**	**22.24**	**.8054**	**.1589**
	2DGS [19]	2DGS	–	22.52	.8185	.1773	20.04	.7587	.2292
	2DGS [19] + Ours		✓	**23.06**	**.8263**	**.1650**	**20.97**	**.7727**	**.2060**

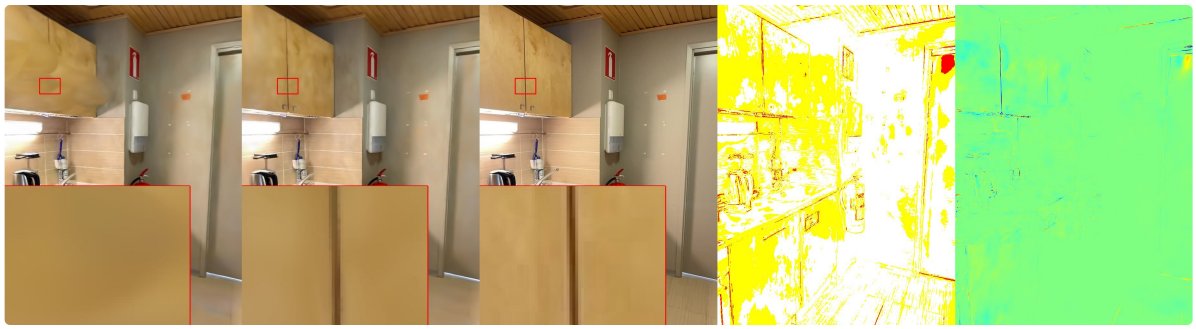


3DGS [25] SuGaR [16] GOF [58] Splatfacto Splatfacto + Ours 2DGS [19] 2DGS + Ours **GT**

Figure 4. We demonstrate our method with two Gaussian-based methods DN-Splatter [42] and 2DGS [19] with qualitative visuals of the reconstructed meshes for the "honka" (top) and "coffee_room" (bottom) scenes from the MuSHRoom dataset. Geometric priors significantly aid in surface optimization for the Gaussian models.

5.3. Ablation Studies

Regularization Strategy. We individually test the performance of our proposed optimization terms in Table 3. We observe that utilizing noisy depths significantly improves the baseline. Our more effective filtering strategy, using adaptive depth and normal supervision, further enhances meshing quality, resulting in a 2.12% increase in the F-score. Additionally, the extracted mesh can be further refined using the IsoOctree meshing method. Qualitative comparisons of each optimization term are visualized in Fig. 6. The



2DGS[19] **2DGS + Ours** **Reference** l_2 **contrib.** Δl_2

Figure 5. Novel view synthesis comparisons on the MuSHRoom dataset. From left to right: 2DGS [19] baseline, 2DGS with our proposed DNC and ANR optimization strategies, reference evaluation image, l_2 error contributions, 2DGS + Ours (red: 30%, yellow: 60%, white: 10%); l_2 error differences 2DGS + Ours vs 2DGS (red: higher error, blue: lower error) by comparing the two rendered images.

Table 3. **Ablation on supervision strategy and mesh performance (MuSHRoom)**. Results are averaged over three scenes.

Input	Acc. ↓	Comp. ↓	C-L_1 ↓	NC ↑	F1 ↑
2DGS [19]	.0676	.0605	.0641	.8156	.6345
+ Depth	.0316	.0246	.0281	.8800	.8861
+ Normal	.0659	.0652	.0656	.8678	.6342
+ Both	.0283	.0247	.0265	.8937	.8880
+ Both + DNC	.0265	.0227	.0246	**.8962**	.9061
+ Both + DNC + ANR	.0262	.0219	.0241	.8927	.9092
+ Both + DNC + ANR + IsoOctree	**.0235**	**.0217**	**.0226**	.8888	**.9157**

Table 4. **Ablation on monocular and sensor depth supervision** on the "vr_room" scene from MuSHRoom. We note that monocular depth supervision greatly under-performs compared to directly using noisy sensor depth readings for indoor room reconstruction.

Input	Acc. ↓	Comp. ↓	C-L_1 ↓	NC ↑	F1 ↑
2DGS [19]	.0652	.0673	.0662	.8275	.6476
+ Zoe depth [4]	.0448	.0480	.0464	.8754	.7224
+ raw sensor depth (no filtering)	**.0216**	**.0217**	**.0216**	**.9039**	**.9051**

IsoOctree meshing method removes some of the grid-like artifacts on smooth walls that are commonly seen in Marching Cubes-based methods. Lastly, the DNC and ANR terms help preserve details for objects and reduce overall noise.

Sensor Depth *vs*. Monocular Depth. In Table 4, we demonstrate that sensor depths remain crucial for indoor room meshing. We compare our approach with the ZoeDepth [4] monocular network estimates, utilizing the recently proposed Patch-based Depth Correlation Loss [51] for monocular depth supervision. For raw sensor depth input, we simply apply the L_1 loss. The quality of monocular depth reconstruction is significantly lower than that of sensor depth reconstruction because of a persistent domain gap in real-world applications.

6. Conclusion

In this work, we presented two regularization strategies to adaptively integrate unreliable geometric priors into Gaussian Splatting based frameworks allowing for better mesh extraction and novel-view synthesis using a mobile device.

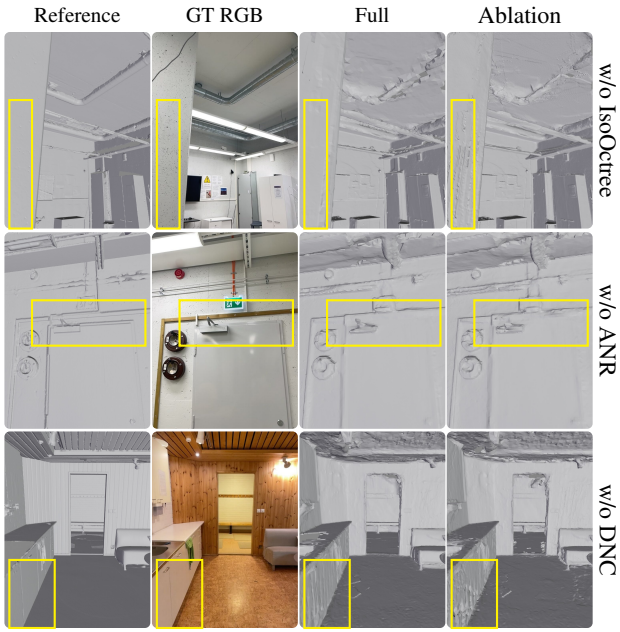


Figure 6. **Qualitative comparison with our optimization strategy.** IsoOctree mesh extraction can efficiently generate a smoother surface compared to the TSDF [59] baseline. Furthermore, our adaptive regularization terms help mitigate misleading reconstruction caused by inaccurate geometric priors.

We showed how filtering noisy sensor depth readings with a normal consistency check and how mitigating uncertainties in monocular normal estimates can be used to better guide optimization. Moreover, we present a promising alternative to traditional meshing techniques using a depth adaptive TSDF and IsoOctree meshing method that can extract finer details from a Gaussian scene. Our method serves as an easy plug-in module to existing Gaussian-based frameworks and we demonstrate the effectiveness of our strategy against baseline methods with thorough experiments.

References

[1] Azure kinect dk depth camera documentation. https://learn.microsoft.com/en-us/azure/kinect-dk/depth-camera. 3

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields, 2022. 2

[4] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023. 3, 8

[5] Payton Chase, Kianna Clarke, Audrey Hawkes, Shabnam Jabari, and Jakov Jakus. Apple iphone 13 pro lidar accuracy assessment for engineering applications. *Transforming Construction with Reality Capture Technologies*, 2022. 3

[6] Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction. *arXiv preprint arXiv:2406.05774*, 2024. 3

[7] Jaehoon Choi, Dongki Jung, Taejae Lee, Sangwook Kim, Youngdong Jung, Dinesh Manocha, and Donghwan Lee. Tmo: Textured mesh acquisition of objects with a mobile device by using differentiable rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16674–16684, 2023. 3

[8] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13 (1):21–27, 1967. 4

[9] 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. 2, 6, 7

[10] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[12] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. 6

[13] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvs-net: Global context-aware multi-view stereo network with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8585–8594, 2022. 2

[14] 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. 4

[15] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[16] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering, 2023. 2, 3, 5, 6, 7

[17] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The international journal of Robotics Research*, 31(5):647–663, 2012. 2

[18] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 3

[19] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[20] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard A. Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew J. Davison, and Andrew William Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011. 2, 5

[21] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, page 143–150, New York, NY, USA, 1986. Association for Computing Machinery. 2

[22] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 3

[23] Michael Kazhdan, Allison Klein, Ketan Dalal, and Hugues Hoppe. Unconstrained isosurface extraction on arbitrary octrees. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, page 125–133, Goslar, DEU, 2007. Eurographics Association. 2, 6

[24] Michael Kazhdan, Allison Klein, Ketan Dalal, and Hugues Hoppe. Unconstrained isosurface extraction on arbitrary octrees. In *Symposium on Geometry Processing*, 2007. 3, 5

[25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4), 2023. 1, 2, 3, 4, 5, 6, 7

[26] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *sensors*, 12(2):1437–1454, 2012. 3

[27] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 2, 5

[28] I. Melekhov, J. Kannala, and E. Rahtu. Image patch matching using convolutional descriptors with euclidean distance. In *Proc. ACCVW*, 2016. 2

[29] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. Ieee, 2007. 2

[30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

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

[32] Minyoung Park, Mirae Do, YeonJae Shin, Jaeseok Yoo, Jongkwang Hong, Joongrock Kim, and Chul Lee. H2O-SDF: Two-phase learning for 3d indoor reconstruction using object surface fields, 2024. 3

[33] Victor Adrian Prisacariu, Olaf Kähler, Stuart Golodetz, Michael Sapienza, Tommaso Cavallari, Philip HS Torr, and David W Murray. Infinitam v3: A framework for large-scale 3d reconstruction with loop closure. *arXiv preprint arXiv:1708.00783*, 2017. 2

[34] Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering, 2024. 4

[35] Marie-Julie Rakotosaona, Fabian Manhardt, Diego Martin Arroyo, Michael Niemeyer, Abhijit Kundu, and Federico Tombari. Nerfmeshing: Distilling neural radiance fields into geometrically-accurate 3d meshes. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2024. 2

[36] Xuqian Ren, Wenjia Wang, Dingding Cai, Tuuli Tuominen, Juho Kannala, and Esa Rahtu. Mushroom: Multi-sensor hybrid room dataset for joint 3d reconstruction and novel view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4508–4517, 2024. 2, 6

[37] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *2017 International Conference on 3D Vision (3DV)*, pages 57–66. IEEE, 2017. 2

[38] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 2

[39] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 3

[40] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 6, 7

[41] 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*, 2022. 2

[42] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024. 1, 2, 3, 4, 6, 7

[43] Hoang-Hiep Vu, Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):889–901, 2011. 2

[44] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. In *2022 International Conference on 3D Vision (3DV)*, pages 433–442. IEEE, 2022. 2, 6

[45] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *European Conference on Computer Vision*, pages 139–155. Springer, 2022. 3

[46] 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

[47] Silvan Weder, Johannes L Schonberger, Marc Pollefeys, and Martin R Oswald. Neuralfusion: Online depth fusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3162–3172, 2021. 2

[48] 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. 2

[49] Yaniv Wolf, Amit Bracha, and Ron Kimmel. Gs2mesh: Surface reconstruction from gaussian splatting via novel stereo views. In *European Conference on Computer Vision*, pages 207–224, 2024. 3

[50] Yuting Xiao, Jingwei Xu, Zehao Yu, and Shenghua Gao. Debsdf: Delving into the details and bias of neural indoor scene reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3

[51] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting. *Arxiv*, 2023. 2, 8

[52] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv:2401.10891*, 2024. 3

[53] 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

[54] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 2, 6

[55] Xinyi Yu, Liqin Lu, Jintao Rong, Guangkai Xu, and Linlin Ou. Improving neural indoor surface reconstruction with mask-guided adaptive consistency constraints. *arXiv preprint arXiv:2309.09739*, 2023. 3

[56] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1949–1958, 2020. 2

[57] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *NeurIPS*, 2022. 2, 5, 6, 7

[58] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. 2, 3, 6, 7

[59] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 3, 5, 6, 8