Refinement of Monocular Depth Maps via Multi-View Differentiable Rendering

Laura Fink[1,2] Linus Franke[1] Bernhard Egger[1] Joachim Keinert[2] Marc Stamminger[1]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg
{first.last}@fau.de

[2]Fraunhofer IIS Erlangen
{first.last}@iis.fraunhofer.de

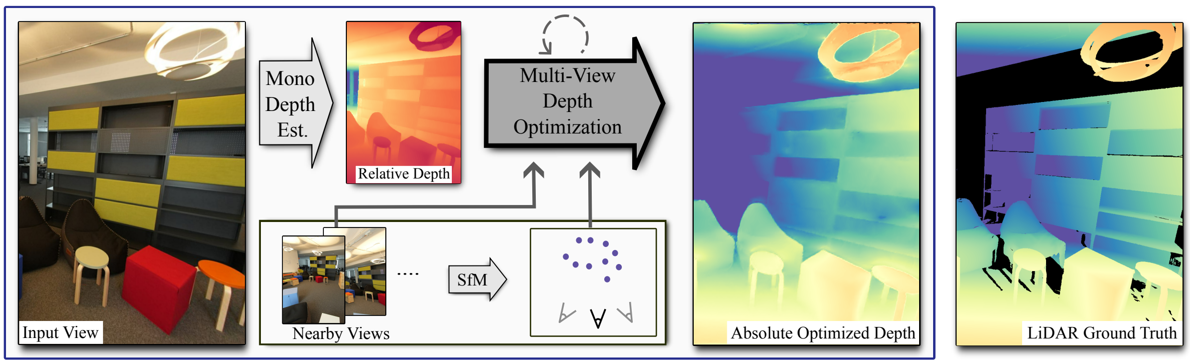


Figure 1. Our novel algorithm refines an initial monocular depth estimation with multi-view differentiable rendering based on a meshed representation of the depth map. This results in dense, accurate, and absolute depth maps, especially in challenging indoor scenarios. Note that the ground truth here is obtained by LiDAR, as such uncertain areas are left blank.

Accurate depth estimation is at the core of many applications in computer graphics, vision, and robotics. Current state-of-the-art monocular depth estimators, trained on extensive datasets, generalize well but lack 3D consistency needed for many applications. In this paper, we combine the strength of those generalizing monocular depth estimation techniques with multi-view data by framing this as an analysis-by-synthesis optimization problem to lift and refine such relative depth maps to accurate error-free depth maps. After an initial global scale estimation through structure-from-motion point clouds, we further refine the depth map through optimization enforcing multi-view consistency via photometric and geometric losses with differentiable rendering of the meshed depth map. In a two-stage optimization, scaling is further refined first, and afterwards artifacts and errors in the depth map are corrected via nearby-view photometric supervision. Our evaluation shows that our method is able to generate detailed, high-quality, view consistent, accurate depth maps, also in challenging indoor scenarios, and outperforms state-of-the-art multi-view depth reconstruction approaches on such datasets.
Project page and source code can be found at https://lorafib.github.io/ref_depth/.

1. Introduction

Depth estimation is crucial in computer vision, graphics, and robotics [9]. They support online use cases like autonomous driving [53], augmented reality [21], or robot navigation [9, 21], but also many offline scenarios like 3D reconstruction [25], supervision for 3D foundation models like DUSt3r or MASt3R [27], post-processing for computational photography and video editing (defocus blur, background removal, etc.), content creation for light field displays, or virtual scene exploration [10, 17]. In this paper, we focus on building a *stable* optimization pipeline that favors highest fidelity of *individual* depth maps for such non-realtime use cases that especially profit from dense and accurate depth information.

For the mentioned scenarios, most often depth maps are used that have been generated by patch-matching methods like COLMAP [42] or MVSNets [65], which often suffer from incomplete reconstructions. Alternatively, recent pretrained monocular depth estimation methods [12, 22, 63, 64] provide good and complete results, although mostly in the form of *relative* depth maps in a normalized scale. Transferring these depth maps to absolute values is possible via least squares fitting to LiDAR data [22, 40], or

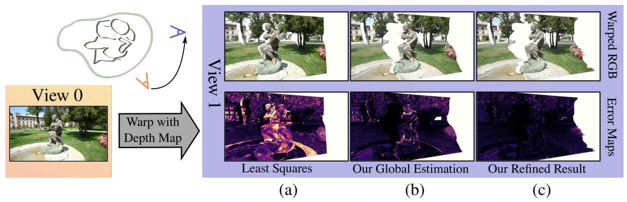


Figure 2. Warping with depth maps reveals inconsistencies and artifacts. Absolute depth maps [63] estimated with least squares display missalignments upon warping. In contrast, our approach, which combines estimation and optimization, effectively preserves and enforces multi-view consistency.

with sparse, absolute multi-view point cloud data [65] as obtained by Structure-from-Motion (SfM) [42]. Yet, this approach also poses problems due to unreliable min/max points and uneven SfM point cloud distributions (see Fig. 2 (a)).

The approach proposed in this paper follows this idea to rescale estimated depth maps to absolute values using SfM data. It is based on two observations: First, similar to the Multi-view Stereo (MVS) community [65], where bounding volumes are guided by SfM, we find that the scene extent can be determined by aligning the medians of the depth map and sparse point cloud leading to a robust initial alignment (see Fig. 2 (b)). Second, we find that explicitly utilizing SfM data and multi-view images and combining it with a differentiable renderer can significantly enhance *scale refinement* and enable *error correction* of depth map inaccuracies (see Fig. 2 (c)). These errors, typically manifesting as noise, oversmoothing or distance errors between objects, change the relative alignments in a depth map, as seen in the histogram in Fig. 4 (b). By *warping* the input image to adjacent views using the depth map and performing a comparison, we observe that depth errors become highly apparent (Fig. 2). Consequently, we propose using a differentiable nearby-view rendering framework, which enables an optimization strategy tailored to address this challenge.

Our pipeline works as follows: First, we use a extensively trained monocular depth estimator [22, 64] to predict individual image depths, generating smooth but non-metric depth maps. As mentioned before, we estimate our global scale and convert the depth map into a triangle mesh for rendering and optimization.

Then we optimize the resulting depth mesh towards a finely detailed multi-view consistent absolute depth map using a differentiable renderer [26]. Optimization takes place in two stages. First, we learn a neural field that remaps the depth map mesh coarsely to the correct values, fitting it directly against the SfM point cloud. This efficiently mitigates distance errors. Second, we apply local depth map refinement in an analysis-by-synthesis paradigm, supervising warped-colored renderings with the captured input images. This removes oversmoothing and restores fine details in the depth map.

As we show in our evaluation on synthetic and real-world data, our method provides highly detailed and accurate depth maps, surpassing competing methods.

Our contributions are as follows:

- A novel analysis-by-synthesis technique to refine monocular depth maps for accurate 3D information via view consistency optimization.
- A two-step refinement scheme that first performs a coarse alignment via shallow neural fields and second follows a local refinement strategy to optimize depth values on fine granularity for highly detailed depth maps.
- An edge aware and a Poisson blending [39] inspired regularizer to exploit the strong initial estimates from monocular estimators. Yielding robust results even in challenging scenarios.
- An extensive evaluation of the proposed method, showcasing its efficiency especially in difficult feature-scarce indoor scenes.

2. Related Work

Image-based Matching & Triangulation. Multi-view-stereo (MVS) [16, 43] methods integrate data from multiple images by matching and triangulating patches based on horizontal disparity [2]. This resembles stereo pair patch matching along the epipolar line to increase match probability through similarity computations. Scarce textures pose challenges, causing sparse results in such areas. Coarse-to-fine strategies [54, 58] or plane prior methods have been implemented to address these issues [54, 59, 69].

The plane sweep algorithm [7] is a method for estimating MVS depth by dividing the viewing area into frontoparallel planes at varying depths and warping them into other views. It projects colors from all views into the camera's frustum, choosing the depth hypothesis with optimal photometric consistency. The quality of depth estimates heavily depends on the number of planes, or essentially the resolution of this volumetric representation.

Yao et al. [65] combined the plane sweep approach with neural networks. Several follow-up works refined the initial architecture and training [3–5, 8, 14, 30, 31, 55, 60, 62, 66, 73]. To date, these networks dominate popular benchmarks such as the Tanks and Temples [24] point cloud reconstruction benchmark.

Novel View Synthesis & Inverse Rendering. In the realm of novel view synthesis (NVS), depth maps serve as a common way to represent an underlying 3D scene [10, 17]. Thus, some methods started to produce depth maps within their pipeline as a byproduct to enable high quality NVS [52, 56, 71]. However, these depth maps often have very limited depth precision [56] or resolution [52].

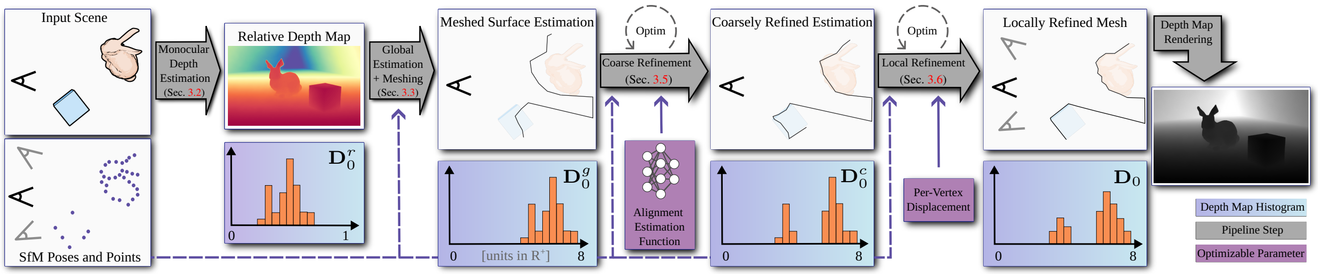


Figure 3. Overview of our method. We employ monocular depth estimation for a relative, but topologically complete depth map. Results from Structure-from-Motion are used to scale the depth map to absolute space. Following, we convert the depth map to a surface mesh for refinement via differentiable rendering. The refinement is done in two consecutive steps: first, we learn a mapping function that smoothly aligns the depth map to the sparse point cloud and second, we refine per-vertex positions, yielding accurate depth maps.

Others use advancements from the novel view synthesis and inverse rendering community to directly optimize for depth maps. Kim et al. [23] integrate image-based rendering methods [29] to infer view-consistent depth maps. Chugunov et al. [6] transfer multi resolution hash grids [34] to optimize near-field depth maps from RAW camera bursts. Mirdehghan et al. [33] and Shandilya et al. [44] cast structured-light scanning as an inverse rendering problem.

Refinement. To mitigate problems in depth maps, such as sparseness and noise, various methods aim to improve low-quality maps by using additional information or integrating data from other sensors, such as aligned photographs. Shape-from-shading techniques estimate the curvature of light-surface interactions, using registered color data to refine the quality of the depth map [15, 20, 35, 36, 50, 57]. Some also optimize the photometric consistency between views with NeRF-based schemes [19].

Other methods [13, 32, 38, 70] use neural networks to complete depth maps from sparse data such as LiDAR of SfM point clouds. Similarly, low-resolution depth maps are upsampled using shape-from-shading [15] or temporal information [48, 68]. Other approaches tackle noise [46, 61] in depth maps.

Monocular Depth Estimation. Monocular depth estimators [22, 37, 45, 63] take shading-based refinement and image-based depth densification further by inferring 3D structures from a single image. Many methods already deliver depth maps with pixel-perfect silhouettes and close approximations of curvatures. However, these depth estimators are geometrically inaccurate and ambiguous due to their lack of geometric triangulation and struggle with ambiguous scenes [12, 37]. Using their learned priors, these estimators can be integrated into multi-view systems [23] or for the regularization of related tasks [28].

3. Method

We aim to build a pipeline which provides a highly detailed and multi-view consistent depth map for a *set* of posed input images, see Fig. 3 for a visualization.

The pipeline begins with a relative depth map from monocular depth estimation. We then scale this non-metrical estimation to absolute values (refer to Secs. 3.1 and 3.2). This adjustment uses the SfM point cloud derived from nearby views. However, the map still exhibits local view inconsistencies (see Fig. 2). We transform the depth map into a fine-grained triangle depth mesh, which is then optimized using inverse rendering, incorporating other images as RGB supervision.

This optimization is performed in two steps: Firstly, we do a *coarse refinement* with only a few degrees of freedom via a very shallow coordinate MLP, which optimized the scaling and offset for the depth map (see Sec. 3.5). In the second step, we *locally* refine (see Sec. 3.6) the depth mesh on a per-vertex basis using differentiable rendering (see Sec. 3.4) aiming for photo-consistency with the other images. To enable robust optimization even in varying lighting conditions, we further integrate a differential tonemapping module (see the supplemental material).

3.1. Input

$\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}, i \in \{0,\dots,n\}$, denotes one of n input images, with W and H being the images' width and height. We first use SfM, specifically COLMAP [42], to estimate intrinsics $\mathbf{C}_i$ and pose $\{(\mathbf{R},\mathbf{t})_i\}$ of each image, as well as a sparse point cloud $\mathbf{X} = \{\mathbf{x}_j\}$ with all $\mathbf{x}_j$ being visible in $\mathbf{I}_0$. $\mathbf{C}$ is defined with the common convention of focal lengths (f_x, f_y) and principal point (c_x, c_y) and images are undistorted.

3.2. Monocular Depth and Global Estimation

We estimate a depth map $\mathbf{D}_0^r \in \mathbb{R}^{H\times W\times 1}$ for $\mathbf{I}_0$ using a strongly pretrained monocular depth estimator.

To approximately map the relative depth map $\mathbf{D}_0^r$ to an

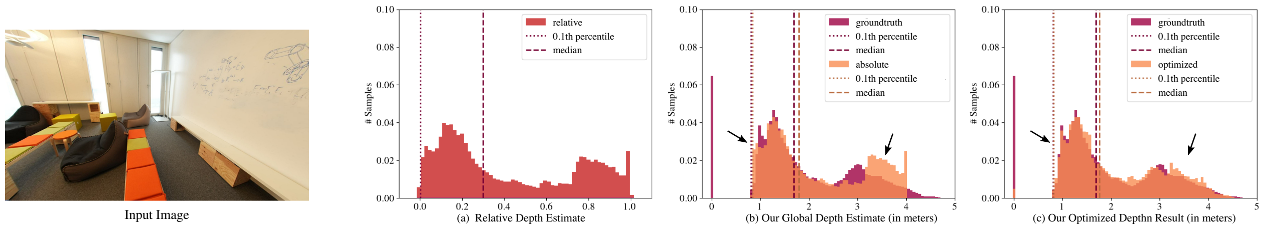


Figure 4. Histograms of different states of depth maps during our optimization. The initial estimate (a) is in relative space (red) and taken as input to be refined (orange). Our absolute estimation (b) brings this to absolute space by aligning the medians and 0.1th percentiles. Then, we optimize (c) to closely capture the depth distribution of the ground truth (red). Missing values, like windows, are counted as 0 m.

absolute scale, we use the sparse point cloud $\mathbf{X}$: we globally offset and scale $\mathbf{D}_0^r$ such that its median distance and the value of its 0.1th percentile are aligned with the corresponding values of the sparse point cloud $\mathbf{X}$, resulting in depth image $\mathbf{D}_0^g$.

Note that we also considered to refine multiple depth maps in parallel (see the supplemental for more details). However, we opted to not do so as the resulting differences are negligible while the VRAM drastically increases.

3.3. Meshing

Next, we convert $\mathbf{D}_0^g$ to a 3D depth mesh $\mathbf{M}$. We opted for a meshed representation as it can be trivially initialized from any dense depth estimation. Furthermore, meshes allow for a non-uniform representation of the underlying geometry. In particular, we can use reduction techniques that further regularize our results and reduce resource consumption.

Triangle mesh. We unproject depth samples at image position (u, v) with $u \in [0, W-1]$ and $v \in [0, H-1]$ to obtain a vertex $\mathbf{p} = (x, y, z)^T$ with

$$x = (u - c_x)\frac{z}{f_x}, \quad y = (v - c_y)\frac{z}{f_y}. \tag{1}$$

and connect the vertices via triangle-based quads.

We usually use a mesh resolution of $W/d \times H/d$ triangle quads, with $d = 4$ to avoid sub-pixel sized triangles, which do not receive gradients during differentiable rasterization and may lead to outliers (see Sec. 3.4). Although this seems as constraining resolution, note that the default lower resolution of estimators [22] equates to d between 2.3 and 2.6 and furthermore that (u, v) can shift within a quad, facilitating piecewise linear silhouette and edge approximation.

The mesh colors are taken from $\mathbf{I}_0$ using the interpolated image coordinates (u, v), separating geometric complexity from image resolution and allowing for vertex reduction.

Decimation. Scene geometry frequently features flat surfaces. Standard mesh simplification techniques identify these areas and replace smaller triangles with larger ones, simplifying the mesh. From the initial depth estimation,

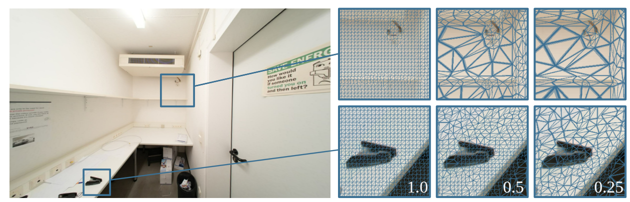


Figure 5. Visualization of our depth mesh $\mathbf{M}$, where triangle size varies with scene complexity. Numbers in lower left corner indicate decimation ratio r.

planar surfaces can be anticipated, allowing us to automatically allocate optimizable parameters strategically. The decimation process naturally regularizes refinement, enhancing robustness even in complex scenes with numerous sparse-feature walls, as demonstrated in Fig. 5. This can be likened to the plane prior support used in patch match-based approaches [59].

We employ quadric mesh decimation [11] on unprojected mesh vertices before local refinement (refer to Sec. 3.6). Decimating vertices in non-linear space (see the supplemental) favors a vertex density near the camera. Quadric mesh decimation efficiently maintains position accuracy, avoiding erroneous non-linear interpolation.

3.4. Differentiable Rendering

We use a differentiable render function Φ defined as

$$(\mathbf{I}_i^*, \mathbf{D}_i^*) = \Phi(\mathbf{C}_i, \mathbf{R}_i, \mathbf{t}_i, \mathbf{C}_0, \mathbf{R}_0, \mathbf{t}_0, \mathbf{M}). \tag{2}$$

Φ projects the current mesh $\mathbf{M}$ with camera parameters $\mathbf{R}_0, \mathbf{t}_0, \mathbf{C}_0$ to a neighboring view i with $\mathbf{R}_i, \mathbf{t}_i, \mathbf{C}_i$, creating a pair of synthesized RGB images and depth maps. Specifically, all vertices $\mathbf{p} \in \mathbf{M}$ (see Eq. 1) are projected with

$$\mathbf{p}_i^* = \mathbf{C}_i[\mathbf{R}_i|\mathbf{t}_i][\mathbf{R}_0|\mathbf{t}_0]^{-1}\mathbf{p} \tag{3}$$

and then rasterized. During optimization, the state of $\mathbf{M}$ continuously changes as described in Sec. 3.5 and 3.6. In the following, we denote the current state of the depth map as $\mathbf{D}_0^*$. Returning from the meshed representation $\mathbf{M}$ to $\mathbf{D}_0^*$ is straight forward by evaluating Φ with $i = 0$, thus

$\mathbf{C}_0, \mathbf{R}_0, \mathbf{t}_0$, which also yields $\mathbf{I}_0^*$, both of which are rasterized versions of $\mathbf{M}$.

3.5. Coarse Refinement

In the first optimization stage, the goal is to transfer the initial depth estimation $\mathbf{D}_0^g$ to more accurate values. Note that in this stage, we do not aim for highest precision, but mainly to coarsely align our depth buffer to the sparse SfM point cloud having only a few degrees of freedom.

To this end, we determine a neural field that yields an offset o and a scale s for any given image coordinate (u, v), thus mapping depth values $z^g \in \mathbf{D}_0^g$ to refined z^c:

$$\begin{aligned} z^c &= z^g(1+s) + o \qquad \forall z^g \in \mathbf{D}_0^g, \\ \text{with } (o, s) &= \Psi(u, v, z^g), \end{aligned} \tag{4}$$

thus arriving at depth map $\mathbf{D}_0^c$.

Specifically, Ψ is modeled as a shallow multi-layer perceptron (MLP):

$$(o, s) = \Psi(\gamma^m(u), \gamma^m(v), \gamma^k(z^g)), \tag{5}$$

where γ^m and γ^k indicate positional encoding [49] functions of degree m and k, respectively.

For optimization of the function, we evaluate Ψ directly at available image positions of the sparse point cloud $\mathbf{X}$ and also compare $\mathbf{D}_0^c$ against $\mathbf{X}$ and use the regularizer as described in Sec. 3.7 using gradient descent. Evaluating the losses of $\mathbf{D}_0^c$ will prevent overfitting of Ψ solely to the available points. The weights of Ψ are initialized very small (see the supplemental for details), thus o^c and s^c will be small, and thus z^c close to z^g.

Using a shallow MLP with low frequent positional encoding inherently exploits the spectral bias of MLPs, mapping similar inputs to similar outputs [49]. Hence, the remapped depth map retains smooth image regions as well as discontinuities.

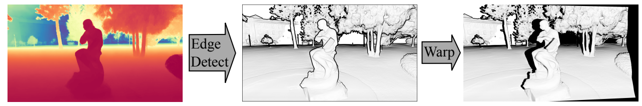
3.6. Local Refinement

The last refinement stage aims to finetune $\mathbf{D}_0^c$, arriving at $\mathbf{D}_0$. Therefore, we jointly optimize depth values z with vertex image coordinates u, v and interject mesh decimation steps in between. The local refinement step aligns the triangle edges with silhouettes in the image and adjusts depth values to accurately trace the surface.

We introduce the per-vertex inverse scaling factor f_z and image space offsets $\Delta u, \Delta v$, thus allowing for a piece-wise linear approximation of silhouettes, as optimizable parameters:

$$z = \frac{z^c}{f_z}, \quad (u, v) = (u^g + \Delta u, v^g + \Delta v). \tag{6}$$

$\Delta u, \Delta v$ are limited by $\frac{d}{2}$ to ensure that the mesh vertices won't intersect. (Note that in practice, we enable optimization of $\Delta u, \Delta v$ already during the coarse refinement.)



Monocular Depth Map $\mathbf{D}_0^r$ | Sobel Filtered Edge Map $\mathbf{E}_0$ | Projected Edge Map $\mathbf{E}_i^*$

Figure 6. To ensure proper visibility masking in nearby-view rendering, we project the edge map $\mathbf{E}_0$ onto the rendered view to weight color loss (white fully weighted, black not weighted).

As in the previous refinement, $\Delta u, \Delta v$ and f_z are optimized with gradient descent as Φ is evaluated repeatedly with $(\mathbf{C}_i, \mathbf{R}_i, \mathbf{t}_i)$ and optimized according to the losses described in Sec. 3.7.

For challenging datasets with varying lighting conditions, we also introduce a tonemapping module that optimizes per image exposure and response parameters. For details, we refer the reader to the supplemental material.

3.7. Losses & Regularizers

We outline losses and regularizers used for the optimization. Exact formulas and setups are in the supplemental.

Geometric Loss. We define $\mathcal{L}_{geo}$ as the relative Huber loss [18] between the current depth buffer and the depth values of the projected sparse point cloud. To get the correspondence between the $\mathbf{D}_0^*$ and $\mathbf{X}$, we calculate $[\mathbf{R}_0|\mathbf{t}_0]X$ to get the linear z^x, and $\mathbf{C}_0[\mathbf{R}_0|\mathbf{t}_0]X$ to get the pixel coordinates per point. Note that we use the point visibility information that COLMAP provides, so we only consider points that are actually visible in I_0. We use $\mathcal{L}_{geo}$ during coarse and local refinement directly on $\mathbf{D}_0^*$. During coarse refinement, we employ $\mathcal{L}_{geo}$ on z^c (see Eq. 4) with $\mathbf{X}$.

Photometric Loss. We use a standard mean squared error formulation for our photometric consistency loss $\mathcal{L}_{photo}$ between rendered $\mathbf{I}_i^*$ and captured images $\mathbf{I}_i$. However, $\mathbf{D}_0$ cannot represent what is behind visible objects in the scene, so as we render $\mathbf{I}_i^*$, regions that were occluded in $\mathbf{I}_0$ are revealed. To prevent false comparisons, such regions are masked in the loss. To this end, we rely on $\mathbf{D}_0^r$ and calculate its edge image $\mathbf{E}_0$ by utilizing a Sobel filter-based edge detection. We use the the projected edge map $\mathbf{E}_i^*$ to mask out the loss in occluded regions, which can be seen in Fig. 6.

Optimizing for photometric consistency presents a non-convex challenge with a chance of becoming trapped in local minima if the optimum is too distant. Thus, we opted to only apply this loss during local refinement when our current depth estimate $\mathbf{D}_0^*$ is closer to the optimum.

Regularizers. To stabilize the optimization, we apply different regularizers, which carefully balance the contributions of dense photometric consistency, sparse geometric cues and the initial estimation. Their general idea is to incorporate and reuse the monocular depth image $\mathbf{D}_0^r$ as prior knowledge about structural information and scene layout and apply that to the meshed and *rendered* version $\mathbf{D}_0^*$.

The edge targeting regularizer $\mathcal{R}_e(\mathbf{E}_0^*, \mathbf{E}_0)$, with $\mathbf{E}_0^*$ being the edge image of $\mathbf{D}_0^*$, encourages that silhouettes of the rendered depth map $\mathbf{D}_0^*$ and estimated depth map $\mathbf{D}_0^r$ should align by calculating their mean squared error and an additional term that moves (u, v) on to the edges by minimizing the values sampled from $\mathbf{E}_0$.

A Poisson blending-inspired regularizer $\mathcal{R}_p$ is proposed for preserving the local structure from $\mathbf{D}_0^r$ to $\mathbf{D}_0^*$ by assessing their image space gradients. Again we mask out silhouettes using $\mathbf{E}_0^r$ to neglect discontinuities, which are prone to be inaccurate. An alternative to reusing the depth map gradients is to regularize on available normals. Notably, $\mathcal{R}_p$ performed on par with using a SotA normal estimator [1] (see the supplemental material) and demonstrating that our regularizer should be well applicable also outside of the context of this work. Our total evaluation loss combines the introduced objective functions using a weighted sum of each of the elements, see the supplemental material.

4. Results

4.1. Setup & Datasets

We evaluate our method on the synthetic Replica dataset [41, 47] and the real-world dataset ScanNet++ [67]. Both offer dense metric scale ground truth, thus allowing for an isolated evaluation of individual depth maps.

For Replica, we use color and ground truth depth maps from rendered meshes, from 2 room and 2 office scenes randomly selected from the subset prepared by Rosinol et al. [41] and optimizing over 64 views per scene. In ScanNet++, ground truth depth maps come from meshes, and the sparse point clouds and poses are computed by COLMAP [42]. For results covered in Sec. 4.2, we used 10 ScanNet++ validation scenes, each with 10 depth maps, totaling 100. Scenes where chosen to have little blurry images, and only consecutive frames. More details regarding the dataset preparation are available in the supplement.

We evaluate standard error metrics: root mean squared error (RMSE), mean absolute error (MAE), L1-rel [51], L1-inv [51], and δ_1 [63]. We also assess depth sample accuracy ($|z - z_{GT}| < \tau$) at various thresholds τ. Additionally, we measure the density of depth maps by counting the number of valid pixels. Values in the top 20 percent are indicated in green, the greener the better.

Notably, we used the exact same hyperparameters for all scenes and datasets, highlighting the stability of our pipeline. This set of parameters was determined by an extensive set of experiments on withheld ScanNet++ scenes. Ablation studies regarding the size of the involved mini MLP during coarse refinement, the mesh decimation ratios d and r, the tonemapper, different monocular depth estimators (generally, we used Marigold [22]), and multi depthmap refinement can be found in the supplemental material. Additionally, we evaluate various loss configurations, even using additional normal estimators, which underlines the flexibility of our method as information from different sources can be trivially fused by adding a loss or adapting the initialization of $\mathbf{D}_0^*$.

	↓ RMSE	MAE	L1-inv	↑ Acc. τ=0.01 m	Acc. τ=0.05 m	Acc. τ=0.10 m	Cumul. time [s]
Global estimation (least squares)	0.29	0.21	0.17	0.06	0.27	0.47	
After step:							
Init. & our global est. ($\mathbf{D}_0^g$)	0.20	0.14	0.11	0.08	0.32	0.55	11.2
Meshing ($\mathbf{M}$)	0.20	0.14	0.11	0.08	0.32	0.55	13.3
Coarse alignment ($\mathbf{D}_0^c$)	0.22	0.12	0.12	0.12	0.44	0.66	128
Local refinement ($\mathbf{D}_0$)	0.17	0.09	0.07	0.22	0.58	0.76	437

Table 1. Intermediate results. On the ScanNet++ test data, each step of our method significantly improves intermediate outcomes, outperforming the least squares baseline [22]. Notably, meshing the depth map preserves quality from $\mathbf{D}_0^g$. Time measured on a Nvidia RTX3090 with OpenGL backend.

See Tab. 1 for a detailed summary of the efficiency each involved step. Each step of our method notably enhances results, surpassing the least squares baseline [22].

While this section mainly focuses on per depth map quality, we further evaluated scene global reconstruction quality on a subset of the Tanks and Temples [24] dataset, showing similar results compared to COLMAP.

4.2. Comparison against Prior Arts

We compare our approach with three highly ranked (and previously best) methods of the Tanks and Temples benchmark [24]. All of them are MVSNet-based methods [65]: MVSFormer [3], MVSFormer++ [4], and GeoMVSNet [72]. For GeoMVSNet however, only weights trained on the DTU dataset were available, which impacted generalization capabilities. These methods produce a certainty value that can be thresholded to control outlier removal and depth map completeness. For Replica and ScanNet++ depth map assessments, we focused on valid ground truth depth values within 7 meters, aligning with MVSFormers' bounding volumes, and applied certainty masks at thresholds of 0, 0.25, and 0.5. The varied completeness metrics for these depth maps are detailed in the tables. Additionally, we compare against the recent MASt3R [27], which works on a quite low resolution (512×384) leading to high errors for small details and around edges. As an established representative of patch match-based methods, we also compared with photometric depth maps generated by COLMAP [43]. Finally, we compared with the recent indoor-specialized *metric* configuration of the monocular depth estimator DepthAnything2 [64]. We used the publicly available code (and weights) for comparisons.

Method	↑ Valid Samples	↓ RMSE	MAE	L1-rel	↑ δ_1	Acc. 0.01 m	Acc. 0.05 m	Acc. 0.10 m
MVSFormer (0)	1.00	0.80	0.31	0.13	0.88	0.21	0.58	0.73
MVSFormer++ (0)	1.00	0.56	0.27	0.10	0.89	0.23	0.61	0.76
MVSFormer (0.25)	0.97	0.75	0.27	0.11	0.90	0.21	0.58	0.73
MVSFormer++ (0.25)	0.95	0.42	0.17	0.06	0.93	0.23	0.61	0.76
MVSFormer (0.5)	0.87	0.46	0.14	0.06	0.96	0.21	0.57	0.71
MVSFormer++ (0.5)	0.85	0.19	0.07	0.01	0.98	0.22	0.59	0.73
Ours (+)	0.99	0.08	0.04	0.02	1.00	0.37	0.81	0.92
Ours (-)	0.99	0.09	0.05	0.02	0.99	0.26	0.69	0.87

Table 2. Results on the Replica dataset. In brackets are confidence thresholds for MVSNet-based methods and sparse point cloud accuracy with good (+) and poor (-) quality for our method.

Method	↑ Valid Samples	↓ RMSE	MAE	L1-rel	↑ δ_1	Acc. 0.01 m	Acc. 0.05 m	Acc. 0.10 m
DepthAnything2 (metric)	100%	1.91	1.79	1.51	0.00	0.00	0.00	0.00
COLMAP	69%	0.60	0.30	0.24	0.73	0.17	0.35	0.41
MASt3R	98%	0.19	0.16	0.16	0.89	0.07	0.30	0.51
MVSFormer$_0$	100%	1.08	0.47	0.41	0.68	0.16	0.40	0.52
MVSFormer++$_0$	100%	0.67	0.28	0.24	0.79	0.22	0.53	0.65
GeoMVSNet$_0$	100%	1.19	0.61	0.77	0.53	0.05	0.18	0.28
MVSFormer$_{0.25}$	90%	1.09	0.46	0.38	0.72	0.16	0.40	0.51
MVSFormer++$_{0.25}$	88%	0.61	0.22	0.19	0.86	0.22	0.52	0.63
GeoMVSNet$_{0.25}$	100%	1.19	0.61	0.77	0.53	0.05	0.18	0.28
MVSFormer$_{0.5}$	47%	0.52	0.16	0.10	0.93	0.14	0.30	0.36
MVSFormer++$_{0.5}$	62%	0.26	0.10	0.07	0.96	0.19	0.43	0.50
GeoMVSNet$_{0.5}$	23%	1.45	0.90	1.11	0.38	0.00	0.02	0.04
Ours	99%	0.17	0.090	0.07	0.93	0.22	0.58	0.76

Table 3. Results on the ScanNet++ dataset [67]. Numbers in subscript indicate the confidence threshold for MVSNet methods.

For the synthetic Replica dataset, sparse point clouds are generated from ground truth data with added realistic levels of noise and outliers. Our method excelled the dataset, as noted in Tab. 2, regardless of point cloud quality. Here, 'Ours (+)' denotes high-quality point clouds (2% noise + 2% outliers). In contrast, 'Ours (-)'indicates use of very low-quality point clouds (5-10% noise + 5-10% outliers) leading to artifacts in extreme cases (see supplemental material for visual results and details on data preparation).

Quantitative results for ScanNet++ are presented in Tab. 3. We achieved top or on-par results in RMSE, completeness (valid samples), MAE, L1-rel, and accuracy at $\tau = 0.05$ and $\tau = 0.1$. MVSFormer++ with a certainty threshold of 0.5 slightly outperformed us in L1-inv, δ_1, and accuracy at $\tau = 0.01$. However, their results only included 62% of values, compared to our over 99.5% of samples.

The qualitative results and ScanNet++ are shown in Fig. 7. Our outputs are clean, dense, and plausible. However, some artifacts appear on the glossy table in `Scene 3`. Sometimes, our results surpass the ground truth, as in the crop of `Scene 4` or the table leg in `Scene 5`, right crop. MVSNet-like results are sparse with noisy walls. DepthAnything2's metric version offers promising details, though its global scale is inaccurate.

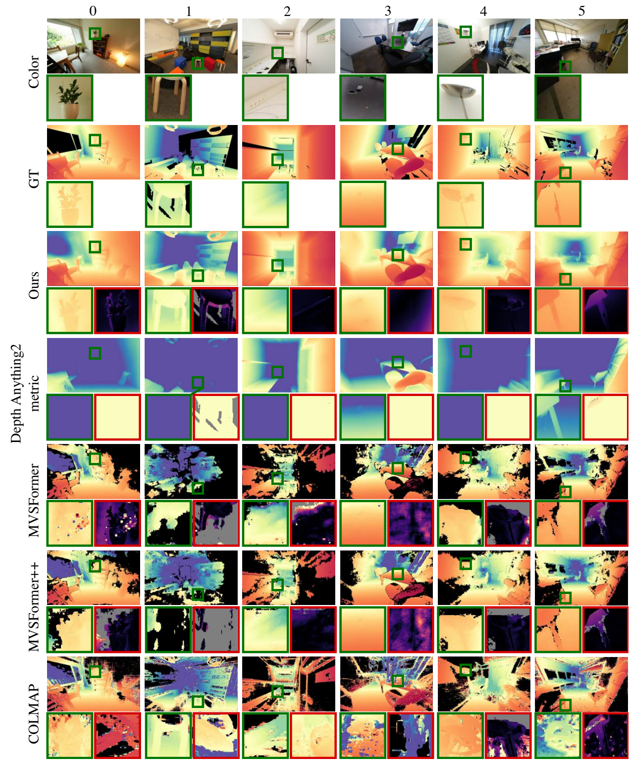


Figure 7. Qualitative results on the ScanNet++ dataset, with MVSFormers' certainty threshold at their default of 0.5. Green borders indicate image crops, while red highlights error maps; brighter areas imply greater errors, and grey indicates missing samples.

5. Limitations

While our method shows strong performance, it still has limitations. Photometric consistency proves useful, but fails in cases with specular materials (see e.g. Fig. 7) which our method does not account for. This could be addressed with uncertainty estimation, anchoring optimization with high-certainty depth estimates.

Ultimately, our approach relies on the initial input. Outliers in the sparse point cloud can result in spurious artifacts and the scene topology, in form of an edge image, has to be derived from the monocular depth map. The edge image is crucial for accurately masking occluded areas, influencing effectiveness of the losses.

6. Conclusion

In conclusion, our approach demonstrates that monocular depth maps can be significantly improved by incorporating photometric consistency across multiple views, leading to enhanced depth estimation. By optimizing a meshed version of the monocular depth map with a differentiable renderer, we directly leverage the strong initial estimates and achieve stable optimization, which refines scaling and corrects errors. In this context, our Poisson and edge regularizers have proven to be particularly effective, ensuring robust and reliable depth map refinement. Our simple yet effective pipeline consistently produces accurate and dense depth maps, which is highly desirable for downstream tasks.

References

[1] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024. 6

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, pages 1–11, 2011. 2

[3] Chenjie Cao, Xinlin Ren, and Yanwei Fu. MVSFormer: Multi-view stereo by learning robust image features and temperature-based depth. *Trans. Mach. Learn. Res.*, 2022, 2022. 2, 6

[4] Chenjie Cao, Xinlin Ren, and Fu Yanwei. MVSFormer++: Revealing the devil in transformer's details for multi-view stereo. In *International Conference on Learning Representations (ICLR)*, 2024. 6

[5] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2524–2534, 2020. 2

[6] Ilya Chugunov, Yuxuan Zhang, and Felix Heide. Shakes on a plane: Unsupervised depth estimation from unstabilized photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13240–13251, 2023. 3

[7] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363. IEEE, 1996. 2

[8] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: 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

[9] Xingshuai Dong, Matthew A Garratt, Sreenatha G Anavatti, and Hussein A Abbass. Towards real-time monocular depth estimation for robotics: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16940–16961, 2022. 1

[10] Laura Fink, Darius Rückert, Linus Franke, Joachim Keinert, and Marc Stamminger. Livenvs: Neural view synthesis on live rgb-d streams. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 1, 2

[11] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 209–216, 1997. 4

[12] Yongtao Ge, Guangkai Xu, Zhiyue Zhao, Libo Sun, Zheng Huang, Yanlong Sun, Hao Chen, and Chunhua Shen. GeoBench: Benchmarking and analyzing monocular geometry estimation models. *arXiv preprint arXiv:2406.12671*, 2024. 1, 3

[13] Tobias Gruber, Frank Julca-Aguilar, Mario Bijelic, and Felix Heide. Gated2depth: Real-time dense lidar from gated images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1506–1516, 2019. 3

[14] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2020. 2

[15] Bjoern Haefner, Yvain Quéau, Thomas Möllenhoff, and Daniel Cremers. Fight ill-posedness with ill-posedness: Single-shot variational depth super-resolution from shading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 164–174, 2018. 3

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003. 2

[17] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018. 1, 2

[18] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 492–518. Springer, 1992. 5

[19] Shintaro Ito, Kanta Miura, Koichi Ito, and Takafumi Aoki. Neural radiance field-inspired depth map refinement for accurate multi-view stereo. *Journal of Imaging*, 10(3):68, 2024. 3

[20] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Depth sensing using geometrically constrained polarization normals. *International Journal of Computer Vision*, 125:34–51, 2017. 3

[21] Megha Kalia, Nassir Navab, and Tim Salcudean. A real-time interactive augmented reality depth estimation technique for surgical robotics. In *2019 international conference on robotics and automation (icra)*, pages 8291–8297. IEEE, 2019. 1

[22] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 1, 2, 3, 4, 6

[23] Subin Kim, Seong Hyeon Park, Sihyun Yu, Kihyuk Sohn, and Jinwoo Shin. MC2: Multi-view consistent depth estimation via coordinated image-based neural rendering. 3

[24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics*, 36 (4), 2017. 2, 6

[25] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1738–1764, 2020. 1

[26] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 39(6), 2020. 2

[27] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 1, 6

[28] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024. 3

[29] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023. 3

[30] Jinli Liao, Yikang Ding, Yoli Shavit, Dihe Huang, Shihao Ren, Jia Guo, Wensen Feng, and Kai Zhang. Wt-mvsnet: window-based transformers for multi-view stereo. *Advances in Neural Information Processing Systems*, 35:8564–8576, 2022. 2

[31] Tianqi Liu, Xinyi Ye, Weiyue Zhao, Zhiyu Pan, Min Shi, and Zhiguo Cao. When epipolar constraint meets non-local operators in multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18088–18097, 2023. 2

[32] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019. 3

[33] Parsa Mirdehghan, Maxx Wu, Wenzheng Chen, David B Lindell, and Kiriakos N Kutulakos. TurboSL: Dense Accurate and Fast 3D by Neural Inverse Structured Light. *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 3

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

[35] Roy Or-El, Guy Rosman, Aaron Wetzler, Ron Kimmel, and Alfred M Bruckstein. Rgbd-fusion: Real-time high precision depth recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416, 2015. 3

[36] Roy Or-El, Rom Hershkovitz, Aaron Wetzler, Guy Rosman, Alfred M Bruckstein, and Ron Kimmel. Real-time depth refinement for specular objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4378–4386, 2016. 3

[37] Evin Pınar Örnek, Shristi Mudgal, Johanna Wald, Yida Wang, Nassir Navab, and Federico Tombari. From 2d to 3d: Re-thinking benchmarking of monocular depth prediction. *arXiv preprint arXiv:2203.08122*, 2022. 3

[38] Jinhyung Park, Yu-Jhe Li, and Kris Kitani. Flexible depth completion for sparse and varying point densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21540–21550, 2024. 3

[39] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 2

[40] 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 Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2020. 1

[41] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444. IEEE, 2023. 6

[42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1, 2, 3, 6

[43] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 6

[44] Aarrushi Shandilya, Benjamin Attal, Christian Richardt, James Tompkin, and Matthew O'toole. Neural fields for structured lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3512–3522, 2023. 3

[45] Jaime Spencer, Fabio Tosi, Matteo Poggi, Ripudaman Singh Arora, Chris Russell, Simon Hadfield, Richard Bowden, GuangYuan Zhou, ZhengXin Li, Qiang Rao, et al. The third monocular depth estimation challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–14, 2024. 3

[46] Vladimiros Sterzentsenko, Leonidas Saroglou, Anargyros Chatzitofis, Spyridon Thermos, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, and Petros Daras. Self-supervised deep depth denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1242–1251, 2019. 3

[47] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6

[48] Zhanghao Sun, Wei Ye, Jinhui Xiong, Gyeongmin Choe, Jialiang Wang, Shuochen Su, and Rakesh Ranjan. Consistent direct time-of-flight video depth super-resolution. In *Proceedings of the Ieee/Cvf Conference on Computer Vision and Pattern Recognition*, pages 5075–5085, 2023. 3

[49] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 5

[50] Michael W Tao, Pratul P Srinivasan, Jitendra Malik, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1940–1948, 2015. 3

[51] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017. 6

[52] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2

[53] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8445–8453, 2019. 1

[54] Yuesong Wang, Zhaojie Zeng, Tao Guan, Wei Yang, Zhuo Chen, Wenkai Liu, Luoyuan Xu, and Yawei Luo. Adaptive patch deformation for textureless-resilient multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1621–1630, 2023. 2

[55] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6187–6196, 2021. 2

[56] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 2

[57] Chenglei Wu, Michael Zollhöfer, Matthias Nießner, Marc Stamminger, Shahram Izadi, and Christian Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics (ToG)*, 33(6):1–10, 2014. 3

[58] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5483–5492, 2019. 2

[59] Qingshan Xu, Weihang Kong, Wenbing Tao, and Marc Pollefeys. Multi-scale geometric consistency guided and planar prior assisted multi-view stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4945–4963, 2022. 2, 4

[60] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 674–689. Springer, 2020. 2

[61] Shi Yan, Chenglei Wu, Lizhen Wang, Feng Xu, Liang An, Kaiwen Guo, and Yebin Liu. Ddrnet: Depth map denoising and refinement for consumer depth cameras using cascaded cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–167, 2018. 3

[62] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020. 2

[63] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 1, 2, 3, 6

[64] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. *arXiv preprint arXiv:2406.09414*, 2024. 1, 2, 6

[65] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 1, 2, 6

[66] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5525–5534, 2019. 2

[67] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 6, 7

[68] Ming-Ze Yuan, Lin Gao, Hongbo Fu, and Shihong Xia. Temporal upsampling of depth maps using a hybrid camera. *IEEE Transactions on Visualization and Computer Graphics*, 25(3):1591–1602, 2018. 3

[69] Zhenlong Yuan, Jiakai Cao, Zhaoqi Wang, and Zhaoxin Li. Tsar-mvs: Textureless-aware segmentation and correlative refinement guided multi-view stereo. *Pattern Recognition*, 154:110565, 2024. 2

[70] Junming Zhang, Manikandasriram Srinivasan Ramanagopal, Ram Vasudevan, and Matthew Johnson-Roberson. Listereo: Generate dense depth maps from lidar and stereo imagery. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7829–7836. IEEE, 2020. 3

[71] Mingfang Zhang, Jinglu Wang, Xiao Li, Yifei Huang, Yoichi Sato, and Yan Lu. Structural multiplane image: Bridging neural view synthesis and 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16707–16716, 2023. 2

[72] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. Geomvsnet: Learning multi-view stereo with geometry perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21508–21518, 2023. 6

[73] Qingtian Zhu, Chen Min, Zizhuang Wei, Yisong Chen, and Guoping Wang. Deep learning for multi-view stereo via plane sweep: A survey. *arXiv preprint arXiv:2106.15328*, 2021. 2