Refin3R: Geometry-Guided Refinement for Feed-Forward 3D Reconstruction

Jongmin Lee Jaeyeong Kim Seungryong Kim†

KAIST AI

https://cvlab-kaist.github.io/Refin3R

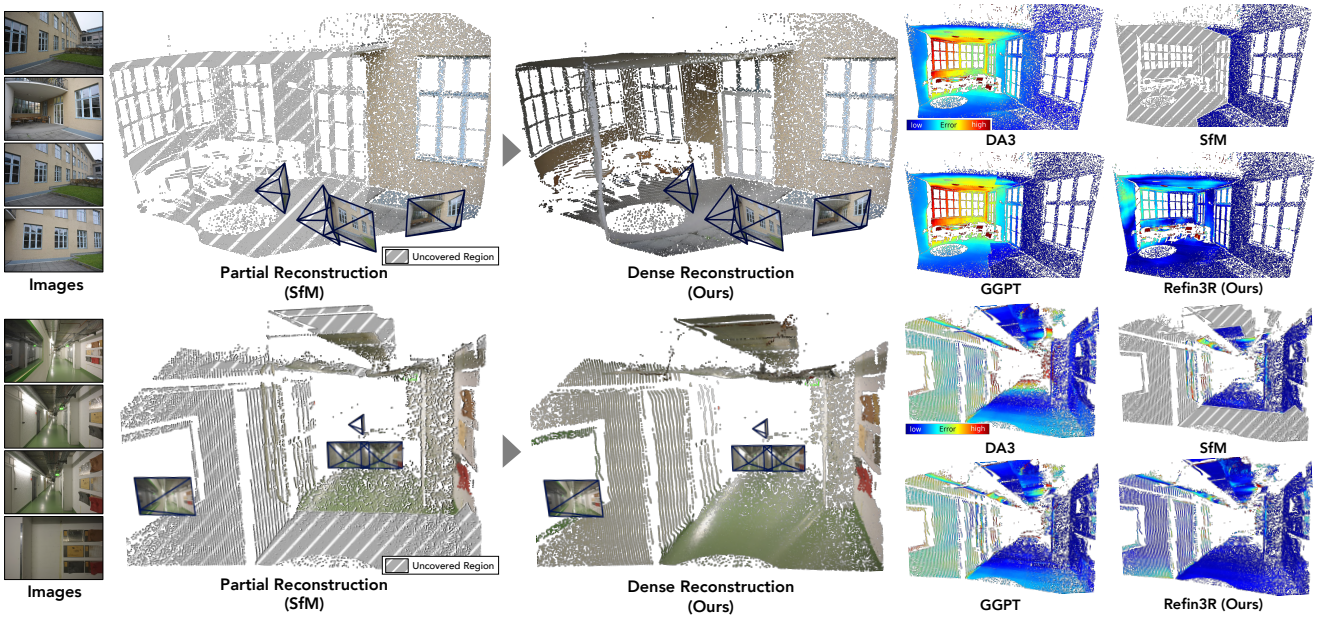


Figure 1. **Refin3R refines feed-forward 3D reconstruction with SfM points.** Given multi-view images, feed-forward models such as Depth Anything 3 (DA3) [27] predict dense but less accurate point maps, while SfM points are accurate but incomplete. Leveraging both, **Refin3R** produces dense and accurate point maps. **Right**: error maps of DA3, SfM (GGPT pipeline [6]), GGPT, and Refin3R.

Abstract

Feed-forward 3D reconstruction models predict dense point maps from multiple images in a single pass. Without explicit geometric constraints, however, their predictions are often inconsistent across views and lack fine-grained precision. Recent methods incorporate geometric priors such as camera pose or depth, but use them merely as auxiliary inputs, process each view independently, or refine the geometry only where the priors are available. We present Refin3R, which refines feed-forward point maps with Structure-from-Motion (SfM) points, propagating SfM geometry over the image grid. In its dual-branch architecture, a 2D branch extends SfM geometry beyond its coverage using geometry-aware backbone features, while a 3D branch reasons over the SfM points of all views, capturing multi-view structure that the image grid alone cannot represent. The two branches exchange information bidirectionally to jointly refine each view, yielding dense and accurate point maps. Across ScanNet++, ETH3D, Tanks&Temples, and out-of-domain human and surgical scenes, Refin3R consistently outperforms feed-forward and geometry-guided methods in point map accuracy while running faster than the strongest geometry-guided baseline.

1. Introduction

Multi-view 3D reconstruction from RGB images is a long-standing problem in computer vision. Recent feed-forward methods [27, 50, 52], trained on large-scale datasets, predict dense point maps for all views in a single forward pass. Their fast and scalable inference has established this paradigm as a promising direction, and follow-up works are rapidly emerging [20, 51, 55]. However, these mod-

†Corresponding author.

els regress point maps purely from learned priors, without enforcing explicit geometric constraints such as triangulation [16]. As a result, their predictions are often inconsistent across views and lack fine-grained precision.

Structure-from-Motion (SfM) [1, 32, 40, 58], in contrast, establishes correspondences across images and triangulates them under explicit multi-view constraints, yielding consistent and accurate points. Yet it often recovers only sparse geometry and becomes fragile under wide baselines or low overlap [35, 41]. Moreover, since it relies solely on correspondences without semantic understanding, mismatches can introduce outliers [5]. The two paradigms are thus complementary: feed-forward models are dense but imprecise, whereas SfM is accurate but incomplete.

Recent works incorporate explicit geometric cues, such as correspondences, camera poses, or depth, into feed-forward reconstruction. Optimization-based methods [9, 26] match pixels across views with learned matchers and jointly refine cameras and dense depth, initialized from feed-forward predictions. However, they adjust depth only via a coarse grid of anchor depths or a few per-image coefficients, limiting fine-grained refinement. Learning-based methods instead feed geometric priors into networks: geometry-conditioned feed-forward models [19, 20] take cameras or depth as extra inputs, and depth completion methods [14, 64, 65] densify sparse depth under image guidance. However, the former use the priors merely as auxiliary inputs, while the latter process each view independently, ignoring the 3D structure shared across views.

Closer to our work, GGPT [6] refines the feed-forward point maps of all views jointly in 3D space with accurate points from its SfM pipeline, enforcing multi-view consistency around these points. However, it refines local cubic patches of points from geometric inputs alone, so its refinement remains largely confined to the regions covered by SfM. Areas without such coverage, typically textureless surfaces and regions observed by a single view, thus benefit little from the SfM guidance (Fig. 1).

Motivated by these limitations, we propose **Refin3R**, which refines the feed-forward point maps as per-view depth on the image grid rather than as points in 3D space. Operating on the image grid allows Refin3R to leverage the dense image features of the pretrained feed-forward backbone [27, 55], which guide the refinement across the whole image. In addition, Refin3R exploits the SfM points of all views in 3D, so that each view is refined with the multi-view structure of the scene rather than only the points it observes. By combining both, Refin3R recovers accurate point maps even in regions without SfM coverage (Fig. 1).

We realize this with a dual-branch architecture: a 2D branch on the image grid and a 3D branch on sparse voxels. The 2D branch refines the feed-forward depth of each view with its own SfM depth, guided by the dense image features from the feed-forward backbone, whose semantic and geometric cues let the SfM depth propagate beyond its coverage. This per-view input, however, holds at most one depth per pixel, so SfM points occluded in the current view or triangulated only from other views are left out. The 3D branch complements it by gathering the SfM points of all views into sparse voxels within the space seen by each view, providing a richer 3D structure than the view's own SfM depth. As each voxel is indexed by a pixel and a depth bin, the two branches share the same pixel coordinates and exchange information: the 2D branch receives the multi-view evidence that the image grid alone cannot represent, and the 3D branch receives the image context around each point. Spatial propagation then fuses their predictions into the refined depth, which is back-projected to obtain the refined point maps.

Extensive experiments show that Refin3R achieves state-of-the-art point map accuracy on both standard and out-of-domain benchmarks, consistently outperforming feed-forward and geometry-guided methods by a large margin. Moreover, while sharing its SfM front-end with the strongest geometry-guided baseline, Refin3R runs substantially faster, making it an efficient refinement module for feed-forward 3D reconstruction.

2. Related works

2.1. 3D reconstruction

Multi-view 3D reconstruction has long been studied in computer vision, with its foundations in classical multi-view geometry [2, 13, 15, 32], and has matured into practical SfM systems [1, 18, 34, 58] such as COLMAP [40]. Learning-based components have since been integrated throughout the SfM pipeline, from feature detection and matching [8, 10–12, 24, 30, 39, 43, 63] to track refinement [17, 23, 29]. Beyond individual components, VGGSfM [49] makes the entire pipeline end-to-end differentiable, while MP-SfM [35] leverages monocular priors to improve robustness under low overlap. Recently, feed-forward reconstruction models infer scene geometry and camera parameters directly from images. Pairwise models [25, 52] regress point maps from image pairs but require a global alignment for more views, whereas subsequent multi-view models [20, 27, 50, 51, 55] process all views jointly in a single forward pass. Trained on large-scale 3D data, these models are fast and robust in sparse-view settings, yet without explicit geometric constraints their predictions often lack multi-view consistency and local precision, especially on out-of-domain scenes.

2.2. Geometry-guided reconstruction

Recent works leverage both feed-forward models and multi-view geometry, following two directions. One direction

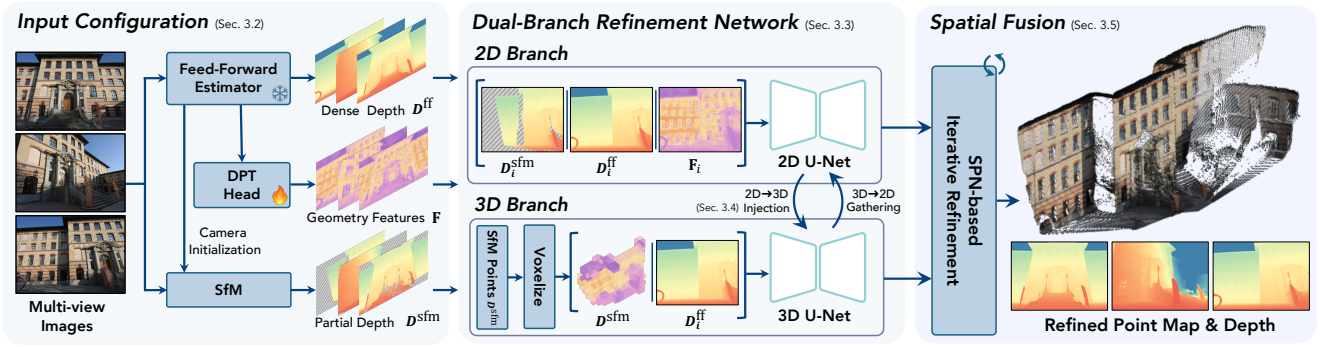


Figure 2. **Overview of Refin3R.** A feed-forward model provides dense depth D^{ff} and geometry-aware features $\mathbf{F}$, and SfM provides partial but accurate depth D^{sfm} (Sec. 3.2). For each view, a 2D U-Net refines the dense depth on the image grid, while a sparse 3D U-Net refines the SfM points of all views that project into its image, with both branches exchanging information (Secs. 3.3 and 3.4). A spatial propagation network (SPN) fuses their estimates into the refined depth (Sec. 3.5), which is back-projected into the final point map.

refines feed-forward predictions by correspondence-based optimization. VGGT+BA [50] refines the predicted cameras through bundle adjustment [45] to improve their accuracy, and MASt3R-SfM [9] and BLASt3R [26] further optimize dense depth jointly with the cameras. The other direction feeds geometric priors into feed-forward models, *e.g.*, multi-view transformers that take camera parameters or depth as optional inputs [19, 20, 55]. Most closely related, GGPT [6] combines both directions: it obtains accurate SfM points with dense matching, sparse bundle adjustment initialized by feed-forward predictions, and linear triangulation [15], and refines the feed-forward point maps with them using a 3D point transformer [59]. We adopt its SfM pipeline but refine depth in the image domain with geometry-aware image features, while aggregating the SfM points of all views in 3D.

2.3. Depth completion

Geometry-guided reconstruction is closely related to depth completion, which densifies sparse depth, typically from LiDAR [47, 48, 62, 65], under image guidance, with recent methods leveraging monocular depth foundation models [56, 64]. Following this formulation, Murre [14] performs multi-view reconstruction by guiding a monocular diffusion model with the SfM depth of each view. However, these methods process each view independently and only use the measurements observed in that view. Our image-domain refinement shares this spirit, but both the image features and the SfM evidence in our method are multi-view: the features come from a multi-view feed-forward model, and the 3D branch aggregates the SfM points of all views.

3. Method

3.1. Overview

Given N RGB images $\{I_i\}_{i=1}^{N}$, our goal is to reconstruct a dense point map $P_i \in \mathbb{R}^{H\times W\times 3}$ for each view in a shared coordinate system. As illustrated in Fig. 2, Refin3R first obtains two complementary reconstructions: a dense but less accurate one from a feed-forward model and a partial but accurate one from SfM. Representing both as pixel-aligned depth maps of each view (Sec. 3.2), it then refines the feed-forward depth on the image grid with per-pixel residuals guided by the SfM points.

The refinement is performed by a dual-branch network that takes both reconstructions along with geometry-aware features from the feed-forward backbone (Sec. 3.3). Its 2D branch combines the feed-forward and SfM depth of each view with these features and refines the dense depth on the image grid. In parallel, its 3D branch refines the depth at the SfM-observed pixels of the current view, leveraging the SfM points aggregated from all views in sparse voxels. The two branches exchange information (Sec. 3.4), and a spatial fusion stage merges their estimates across the image through iterative spatial propagation (Sec. 3.5). Finally, the refined depth D_i^{pred} is back-projected along the pixel rays of the SfM cameras to obtain the point map P_i of each view.

3.2. Input representation

Initial reconstructions. Following GGPT [6], we obtain two initial reconstructions. A pretrained multi-view feed-forward model, such as Depth Anything 3 (DA3) [27] or Pi3X [55], first predicts initial camera parameters and a dense depth map $D_i^{\text{ff}} \in \mathbb{R}^{H\times W}$ for every view. Building on this prediction, an SfM pipeline establishes correspondences with RoMa v2 [12], refines the cameras $(\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i)$ by sparse bundle adjustment, and triangulates valid tracks by multi-view DLT [16]; see GGPT [6] for details. This yields a partial SfM depth map D_i^{sfm} with a validity mask $M_i \in \{0,1\}^{H\times W}$ marking triangulated pixels.

Geometric inputs. We express both reconstructions as depth along the pixel rays, so that they are aligned on the image grid of each view, and parameterize depth in log space, $\zeta = \log D$, which compresses its wide dynamic

range [22]. Each pixel $p = (u, v)$ of view i is then described by its ray $\mathbf{r}_i(p) = \mathbf{K}_i^{-1}[u, v, 1]^\top$, the feed-forward and SfM log-depths $\zeta_i^{\mathrm{ff}}(p)$ and $\zeta_i^{\mathrm{sfm}}(p)$, and the mask $M_i(p)$, where we set $\zeta_i^{\mathrm{sfm}}(p) = 0$ if $M_i(p) = 0$.

Geometry-aware image features. As in depth completion, propagating sparse geometry across an image requires dense geometric context, indicating how depth relates between neighboring pixels. We therefore reuse intermediate features of the frozen feed-forward backbone, whose multi-view transformer is trained to predict geometry and thus carries both semantic and geometric cues. We take the image tokens $\mathbf{T} \in \mathbb{R}^{4\times(H_pW_p)\times C_t}$ from four intermediate layers of the backbone, where H_pW_p is the number of patches and C_t the token dimension. A DPT head [36], trained end-to-end as part of the full model, fuses and upsamples them into an image-resolution feature map $\mathbf{F}$:

$$\mathbf{F} = \mathrm{DPT}_{\mathrm{feat}}(\mathbf{T}) \in \mathbb{R}^{H\times W\times C_F}. \tag{1}$$

3.3. Dual-branch refinement network

Our setting resembles depth completion, which densifies sparse depth measurements under RGB guidance [14, 64, 65], but differs in two aspects: the feed-forward depth already provides a dense initial estimate, and the partial measurements come from all views rather than the current one alone. We therefore design one branch for each: a dense 2D branch that refines the feed-forward depth on the image grid, and a sparse 3D branch that refines the SfM depth using the points of all views. Both predict residuals in log-depth space, the 2D branch on ζ_i^{ff} and the 3D branch on ζ_i^{sfm} at the SfM-observed pixels.

2D branch. Guided by the image features $\mathbf{F}$, the 2D branch $U_{2\mathrm{D}}$ propagates the partial SfM depth to pixels without any triangulated point, *e.g.*, textureless regions or regions observed by a single view. It is a U-Net [38] built from ConvNeXt blocks [57] with six scales (stride 1 to 32). Per pixel, it takes the geometric channels:

$$\mathbf{g}_i(p) = \big[\, \mathbf{r}_i \,\|\, \zeta_i^{\mathrm{ff}} \,\|\, \zeta_i^{\mathrm{sfm}} \,\|\, (\zeta_i^{\mathrm{sfm}} - \zeta_i^{\mathrm{ff}}) \,\|\, M_i \,\big](p), \tag{2}$$

where $\|$ denotes channel-wise concatenation. The difference $\zeta_i^{\mathrm{sfm}} - \zeta_i^{\mathrm{ff}}$ (set to zero where $M_i = 0$) makes the discrepancy between the two reconstructions explicit, indicating how much the feed-forward depth deviates from the SfM depth.

Given $\mathbf{g}_i$ and the image features $\mathbf{F}_i$, $U_{2\mathrm{D}}$ predicts a log-depth residual $\Delta\zeta_i^{2\mathrm{D}}$ and a confidence logit $\hat{c}_i$:

$$(\Delta\zeta_i^{2\mathrm{D}}, \hat{c}_i) = U_{2\mathrm{D}}\big([\, \mathbf{g}_i \,\|\, \mathbf{F}_i \,]\big), \quad \zeta_i^{2\mathrm{D}} = \zeta_i^{\mathrm{ff}} + \Delta\zeta_i^{2\mathrm{D}}, \tag{3}$$

where $\hat{c}_i$ is used for training supervision (Sec. 3.6).

3D branch. The 2D branch takes only the SfM depth of the current view, which holds at most one value per pixel; points occluded in this view or triangulated only from other views are thus left out, leaving the full scene geometry unexploited. Inspired by multi-view reconstruction methods that aggregate evidence in 3D space [33, 37], we introduce a 3D branch that reasons over the SfM points of all views with sparse 3D convolution to refine each view's depth.

Following MoGe-3 [22], we construct for each view i a voxel grid in its image coordinates and log depth at the original image resolution, so that a voxel (u, v, z) corresponds to pixel (u, v) and z indexes its discretized log depth. We then voxelize the SfM points of *all* views into this grid, keeping only those that lie in front of camera i and project inside its image. Further details, including how points are assigned to voxels, are in the supplementary material.

The 3D branch $U_{3\mathrm{D}}$ is a sparse U-Net with five levels of residual sparse 3D convolution blocks on this grid, through which each voxel aggregates features from points close to it in 3D space. For each occupied voxel m, let p_m denote its pixel and ζ_m the log depth of its SfM point in view i. As in Eq. (2), $U_{3\mathrm{D}}$ takes per voxel an input feature that combines ζ_m with the offsets of ζ_m from the feed-forward and SfM depths at p_m:

$$\begin{aligned}\mathbf{v}_m^{(0)} = \big[\, &\mathbf{r}_i(p_m) \,\|\, \zeta_m \,\|\, \zeta_m - \zeta_i^{\mathrm{ff}}(p_m) \,\| \\ &\zeta_m - \zeta_i^{\mathrm{sfm}}(p_m) \,\|\, M_i(p_m) \,\|\, \mathbb{1}_m^{\mathrm{own}} \,\big],\end{aligned} \tag{4}$$

where $\mathbb{1}_m^{\mathrm{own}}$ indicates whether m holds a point triangulated by view i. The two differences encode the depth of each SfM point relative to the feed-forward depth and to the pixel's own SfM depth, respectively (the latter difference is set to zero where $M_i(p_m) = 0$), so that the scene geometry aggregated from all views is represented in terms of the current view.

From these features, $U_{3\mathrm{D}}$ predicts a log-depth residual and a confidence logit for every occupied voxel at the input resolution, *i.e.*, that of $\mathbf{v}_m^{(0)}$. Reading them out at the voxels with $\mathbb{1}_m^{\mathrm{own}} = 1$ yields per-pixel maps on $M_i = 1$:

$$(\Delta\zeta_i^{3\mathrm{D}}, \hat{\beta}_i) = U_{3\mathrm{D}}\big(\{\mathbf{v}_m^{(0)}\}_m\big), \quad \zeta_i^{3\mathrm{D}} = \zeta_i^{\mathrm{sfm}} + \Delta\zeta_i^{3\mathrm{D}}. \tag{5}$$

The confidence $\beta_i = \sigma(\hat{\beta}_i)$, where σ is the sigmoid function, is used in Sec. 3.5.

3.4. Bidirectional 2D–3D interaction

The two branches are complementary: the 2D branch uses image features to propagate depth to pixels without SfM points, whereas the 3D branch holds multi-view evidence that the image grid alone cannot represent, such as SfM points triangulated without correspondences in view i. We let them exchange information in their encoders at two coarse scales, $l \in \{3, 4\}$ (stride 8 and 16), where both branches have the same feature width. Denoting by $\mathbf{X}_i^{(l)}$ the 2D feature map of view i and by $\mathbf{V}_i^{(l)}$ the voxel features of its grid at scale l, each exchange proceeds as:

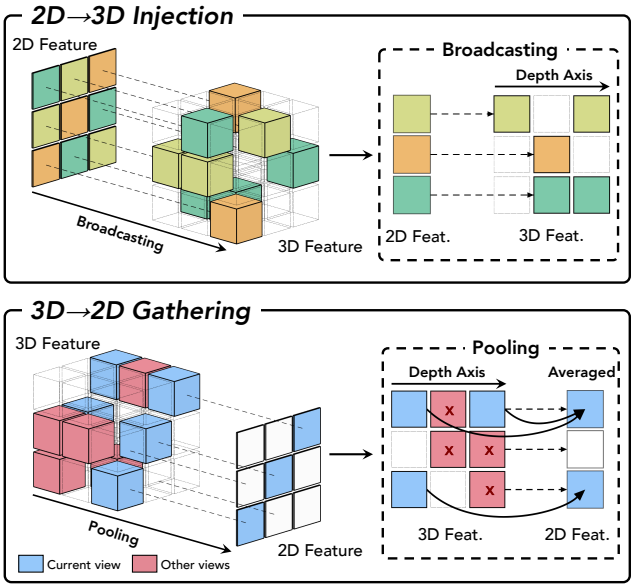


Figure 3. **Bidirectional 2D–3D interaction.** The 2D feature map of each view and the voxel features of its grid exchange information along pixel columns. *Top*: In 2D→3D injection, the image feature of each pixel is broadcast to all occupied voxels in its column, providing the 3D branch with image context. *Bottom*: In 3D→2D gathering, the voxel features in each column are average-pooled along the depth axis and fused into the image feature map, providing multi-view evidence back to the 2D branch. Only voxels containing points of the current view (blue) are pooled, while those formed solely by points of other views (red) are excluded.

$$\begin{aligned}\mathbf{X}^{(l)} &\leftarrow \mathrm{Enc}^{(l)}_{\mathrm{2D}}\big(\mathbf{X}^{(l-1)}\big),\\ \mathbf{V}^{(l)} &\leftarrow \mathrm{Blocks}^{(l)}_{\mathrm{3D}}\big(\mathrm{Inject}_l(\mathbf{V}^{(l)}, \mathbf{X}^{(l)})\big),\\ \mathbf{X}^{(l)} &\leftarrow \mathbf{X}^{(l)} + \mathrm{Gather}_l(\mathbf{X}^{(l)}, \mathbf{V}^{(l)}),\end{aligned} \tag{6}$$

where $\mathrm{Enc}^{(l)}_{\mathrm{2D}}$ is the scale-l encoder stage of U_{2D}, $\mathrm{Blocks}^{(l)}_{\mathrm{3D}}$ are the sparse convolution blocks of U_{3D} at the same scale, and Inject_l and Gather_l are the two exchange operations detailed below.

Since both branches halve the image axes at each scale, the 2D feature map and the voxel grid at scale l share the same image grid, downsampled by 2^l. Every voxel m thus lies at a pixel p_m of this grid, and we call the set of voxels corresponding to a pixel p_m its *column*. In the 2D→3D direction (Inject), voxels in a column receive the image feature at its pixel, $\mathbf{X}^{(l)}_i(p_m)$, by a direct lookup. In the 3D→2D direction (Gather), $\mathbf{X}^{(l)}_i(p)$ receives a single feature reduced from the voxel features in its column.

2D→3D injection. Each voxel feature $\mathbf{v}^{(l)}_m$ is concatenated with the image feature at the corresponding pixel, $\mathbf{X}^{(l)}_i(p_m)$, providing image context absent from the 3D branch's input:

$$\mathbf{v}^{(l)}_m \leftarrow \mathrm{MLP}\left([\,\mathbf{v}^{(l)}_m \,\|\, \mathbf{X}^{(l)}_i(p_m)\,]\right). \tag{7}$$

All voxel features in a pixel column receive the same image feature (Fig. 3 top).

3D→2D gathering. The voxel features contain multi-view 3D information, aggregated from the SfM points of all views. We provide this information to the 2D branch by reducing each column to a single feature at a pixel (Fig. 3 bottom). Since the 2D branch operates on the image of view i, we use only the voxels relevant to this view. Specifically, we extend $\mathbb{1}^{\mathrm{own}}$ to scale l by marking a voxel as own if it contains at least one own voxel at the input resolution (blue in Fig. 3 bottom), and exclude voxels that contain only other views' voxels (red in Fig. 3 bottom).

The own voxels in each column are then average-pooled along the depth axis:

$$\mathbf{q}_p = \tfrac{1}{|\mathcal{Z}_p|}\sum_{m\in\mathcal{Z}_p} \bar{\mathbf{v}}^{(l)}_m, \quad \mathcal{Z}_p = \{m : p_m = p,\ \mathbb{1}^{\mathrm{own}}_m = 1\}, \tag{8}$$

where $\bar{\mathbf{v}}$ denotes layer-normalized voxel features, and $\mathbf{q}_p = \mathbf{0}$ if $\mathcal{Z}_p$ is empty. The pooled feature $\mathbf{q}_p$ is then concatenated with the image feature $\mathbf{X}^{(l)}_i(p)$, fed into an MLP, and added to it as a residual:

$$\mathbf{X}^{(l)}_i(p) \leftarrow \mathbf{X}^{(l)}_i(p) + \mathrm{MLP}\left([\,\mathbf{X}^{(l)}_i(p) \,\|\, \mathbf{q}_p\,]\right). \tag{9}$$

3.5. Spatial fusion

The two branches provide two refined depth estimates: a dense one from the 2D branch and, on SfM-observed pixels, one from the 3D branch. Fusing them per pixel would mix the two only where both exist, so the 3D-branch depth would refine only the SfM-observed pixels without affecting their neighbors. We therefore employ spatial propagation [7], which iteratively updates a dense depth map with learned, content-dependent neighbor weights to mix the two estimates adaptively across the entire image.

Specifically, we adapt the dynamic variant [28] and propagate in linear depth, $D = \exp\zeta$. On SfM-observed pixels, the confidence β_i of Eq. (5) blends the SfM and 3D-branch depths into an anchor, $a_i = (1-\beta_i)\,D^{\mathrm{sfm}}_i + \beta_i\,D^{\mathrm{3D}}_i$. The 2D branch is equipped with a second decoder, which shares the encoder after the 2D–3D interaction and predicts the propagation parameters: affinities $w_{t,k}$ and offsets $\mathbf{o}_{t,k}$ over $K = 5$ neighbors for each round t, and an anchoring weight η_i set to zero where no SfM observation exists. Starting from $D^{(0)}_i = D^{\mathrm{2D}}_i$, each of the T rounds updates:

$$D^{(t)}_i(p) = (1-\eta_i)\sum_{k=1}^{K} w_{t,k}\,D^{(t-1)}_i(p+\mathbf{o}_{t,k}) + \eta_i\,a_i, \tag{10}$$

where η_i, a_i, $w_{t,k}$, and $\mathbf{o}_{t,k}$ are taken at p, and $w_{t,\cdot}$ is normalized by a softmax over the K neighbors. After $T = 6$ rounds, $D^{pred}_i = D^{(T)}_i$ gives the refined depth, which is back-projected with the SfM cameras to obtain the point map P_i.

3.6. Training objectives

We supervise the outputs of the 2D branch, the 3D branch, and the propagation. For the dense outputs, we use the confidence-weighted loss of [6, 52] with the depth error,

$$\mathcal{L}_{\text{conf}}(D, c) = \frac{1}{|\Omega_i|} \sum_{p \in \Omega_i} c(p) \left| D(p) - D^{\text{gt}}(p) \right| - \alpha \log c(p), \tag{11}$$

where Ω_i is the set of pixels with valid ground truth, $c_i = 1 + \exp(\hat{c}_i) \geq 1$ is the confidence obtained from the logit of Eq. (3), and $\alpha = 0.2$ balances the regularizer. The 3D branch is supervised only at the view's own SfM pixels with a plain L1 loss, $\mathcal{L}_{\text{3D}} = \frac{1}{|\Omega_i \cap M_i|} \sum_{p \in \Omega_i \cap M_i} |D_i^{\text{3D}}(p) - D_i^{\text{gt}}(p)|$, without confidence. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{conf}}(D_i^{pred}, c_i) + \lambda_{\text{2D}}\, \mathcal{L}_{\text{conf}}(D_i^{\text{2D}}, c_i) + \lambda_{\text{3D}}\, \mathcal{L}_{\text{3D}}, \tag{12}$$

where the first term supervises the final output, and the second keeps a gradient on the 2D-branch depth from which the propagation starts.

4. Experiments

4.1. Implementation details

Inputs and architecture. For both training and evaluation, we obtain the initial reconstructions from a feed-forward backbone, either DA3 [27] or Pi3X [55], and the SfM pipeline of GGPT [6] (Sec. 3.2); we report results for both versions, each trained separately. The 2D branch follows the U-Net of PacGDC [47], the DPT head follows the architecture of the DA3 depth head, and the 3D branch follows the sparse U-Net of MoGe-3 [22].

Training. We train on the training splits of ScanNet++ [61], TartanAir v2 [53, 63], UnrealStereo4K [44], and BlendedMVS [60], all of which provide dense ground-truth depth and camera poses. Following MapAnything [20], we sample multi-view sets via a covisibility-constrained random walk, with 4–16 views for ScanNet++, 4–8 views for TartanAir v2 and UnrealStereo4K, and 3–6 for BlendedMVS. We train for 200K iterations on 4 NVIDIA RTX A6000 GPUs using AdamW [31] with a learning rate of 10^{-4} and weight decay 0.05. The learning rate is linearly warmed up over the first 2K iterations and then cosine-decayed to 10^{-5}. Further architectural and training details are in the supplementary material.

4.2. Evaluation protocol

Datasets. Following GGPT [6], we evaluate on ScanNet++ [61], ETH3D [42], and Tanks&Temples [21], each with 4-, 8-, and 16-view splits. Since ScanNet++ is part of our training data, we refer to it as within-domain, and to ETH3D and Tanks&Temples as cross-domain. We further evaluate on two out-of-domain datasets whose appearance and geometry differ substantially from the training data: 4D-DRESS [54], rendered from high-resolution scans of clothed humans, and MV-dVRK [4], photorealistic renderings of robotic abdominal surgery.

Metrics. We evaluate point map accuracy with the area under the accuracy curve (AUC) at several error thresholds. We align each predicted point map to the metric ground truth with robust Umeyama alignment [46] implemented in pycolmap [40], and measure the Euclidean error between each predicted point and its ground-truth counterpart.

Baselines. We compare against four groups of methods. For fairness, all methods that take geometric priors receive the same SfM output as ours. (1) *Feed-forward models*: DA3 [27], Pi3X (RGB only) [55], and VGGT-Ω [51], which take only images as input. (2) *Geometry-conditioned feed-forward models*: MapAnything [20] and Pi3X, conditioned on our SfM cameras and the per-view SfM depth D_i^{sfm} (Sec. 3.2). (3) *Depth completion*: Murre [14] and Any2Full [64], which complete D_i^{sfm} of each view independently; the results are back-projected with the SfM cameras into a global point map. (4) *Feed-forward reconstruction with multi-view geometry*: MASt3R-SfM [9] and BLASt3R [26], which run their own pipelines to jointly optimize cameras and dense depth using correspondences and depth priors, and GGPT [6], which refines the feed-forward point maps with the SfM points in 3D. We apply GGPT to the same DA3 and Pi3X initial reconstructions as ours.

4.3. Results

We summarize the main results below. Additional qualitative examples and detailed quantitative results are provided in the supplementary material.

Within- and cross-domain evaluation. Tab. 1 compares Refin3R with the baselines on ScanNet++, ETH3D, and Tanks&Temples. Refin3R achieves the best accuracy across all benchmarks and, with either backbone, consistently outperforms GGPT applied to the same initial reconstructions. Geometry-conditioned models such as Pi3X also benefit from the same SfM priors but remain far behind Refin3R.

Out-of-domain evaluation. Tab. 2 reports results on 4D-DRESS and MV-dVRK, whose appearance and geometry differ substantially from the training data. While the feed-forward backbones largely fail on these scenes, Refin3R achieves the best results on both datasets, with the largest margins at the strictest threshold. VGGT-Ω is competitive at the looser threshold on 4D-DRESS, likely owing to human-centric training data [3], yet falls far behind Refin3R at the 1 cm threshold.

Qualitative comparison. Fig. 4 compares point error maps of Refin3R and the baselines on all datasets. Refin3R keeps the accuracy of the SfM points while refining regions they do not cover, such as textureless surfaces and single-view regions, where the baselines retain large errors.

Table 1. **Multi-view 3D reconstruction on standard test sets.** We report AUC@1/5 cm (% ↑). ScanNet++ is within-domain, and ETH3D and Tanks&Temples are cross-domain. The best and second-best results for each metric within each dataset are shown in **bold** and underline, respectively.

	ScanNet++ [61]			ETH3D [42]			T&T [21]		
	4 imgs	8 imgs	16 imgs	4 imgs	8 imgs	16 imgs	4 imgs	8 imgs	16 imgs
Feed-forward									
Depth Anything 3 [27]	1.6/8.8	0.9/7.1	0.6/4.7	8.9/31.9	8.0/29.6	7.0/27.1	1.1/8.7	0.7/7.1	0.5/6.3
Pi3X (RGB only) [55]	14.7/50.4	15.3/53.9	14.3/54.8	9.8/34.8	6.1/28.8	4.8/26.4	5.1/25.2	3.9/23.5	3.4/22.7
VGGT-Ω [51]	2.1/16.3	1.3/13.5	1.0/11.9	4.1/25.6	3.1/22.2	2.2/17.2	2.0/16.0	1.5/14.6	0.9/9.9
Geometry-conditioned Feed-forward									
MapAnything [20]	15.6/43.9	13.3/43.8	12.5/42.0	6.8/27.3	6.3/25.0	5.3/25.2	3.4/18.7	2.5/15.1	2.2/14.7
Pi3X [55]	17.3/52.1	18.2/55.5	18.3/58.3	15.6/43.7	12.9/40.0	10.2/37.9	8.9/33.6	8.1/34.9	6.4/30.7
Depth completion									
Any2Full [64]	22.9/50.0	28.6/59.6	32.0/65.1	24.8/49.3	28.5/55.4	30.4/58.9	12.1/34.4	13.1/42.1	13.9/42.7
Murre [14]	16.6/44.1	21.3/53.1	23.9/57.4	13.0/34.4	13.6/35.4	15.2/38.7	5.5/22.2	6.6/25.7	7.7/28.2
Feed-forward w/ multi-view geometry									
MASt3R-SfM [9]	11.5/30.8	15.6/41.0	18.9/49.2	15.0/37.4	17.5/37.7	16.7/39.0	14.9/35.1	18.2/41.6	17.7/41.4
BLASt3R [26]	27.8/57.1	34.5/64.7	39.2/67.5	11.7/31.2	13.6/34.9	14.2/38.4	4.1/19.1	5.2/22.7	5.0/22.8
GGPT (w/ DA3) [6]	21.6/45.0	27.4/56.9	31.8/64.5	27.7/56.2	28.4/58.7	32.6/64.4	14.9/36.9	16.0/46.1	16.0/47.1
GGPT (w/ Pi3X) [6]	25.6/56.7	29.5/62.8	34.8/67.8	31.7/60.0	32.7/62.0	39.4/69.1	16.1/38.5	19.1/49.7	19.7/49.4
Refin3R (w/ DA3) (Ours)	31.5/60.6	37.5/68.8	43.4/74.0	42.7/67.3	**46.4**/69.9	**53.6**/76.1	15.6/**40.3**	20.5/**52.1**	21.8/**53.1**
Refin3R (w/ Pi3X) (Ours)	**33.7/65.6**	**39.6/71.4**	**45.2/75.7**	**43.1/67.6**	46.3/**70.2**	53.3/**76.4**	**17.1**/39.7	**21.5**/52.0	**22.4**/52.3

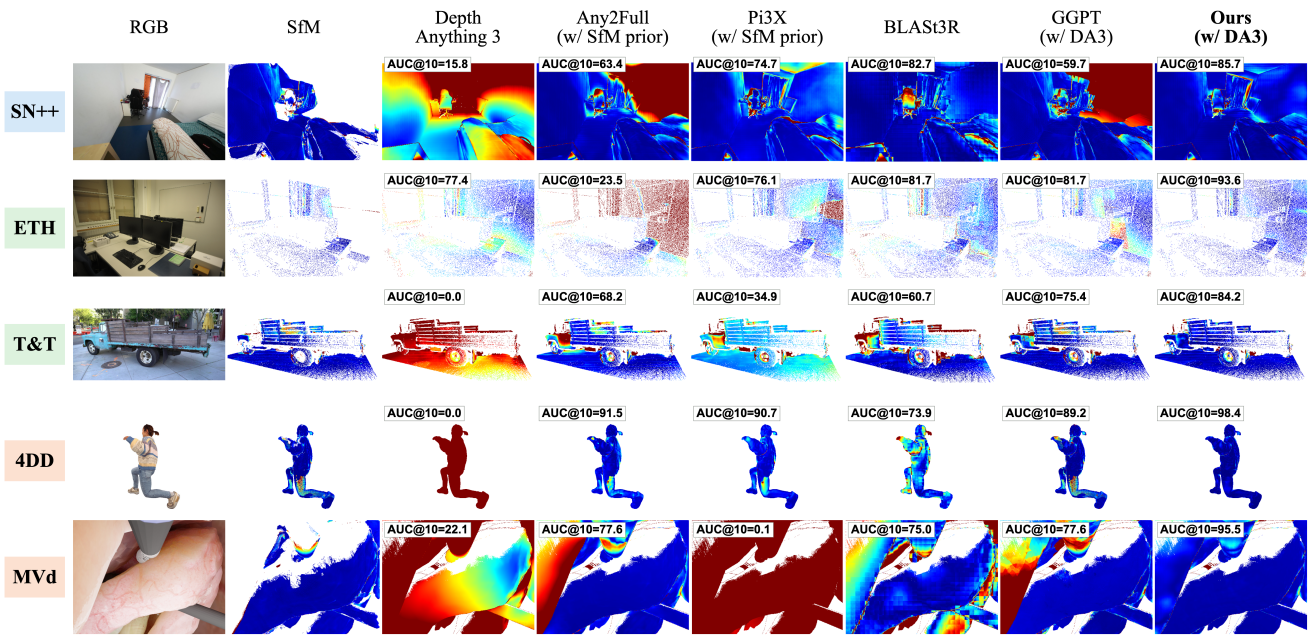


Figure 4. **Qualitative comparison.** Each row shows one example from ScanNet++, ETH3D, Tanks&Temples, 4D-DRESS, and MV-dVRK, with only one input view visualized. We compare a feed-forward model using only images (DA3), monocular depth completion (Any2Full), a geometry-conditioned model (Pi3X), and feed-forward methods with multi-view geometry (BLASt3R, GGPT), where Any2Full, Pi3X, and GGPT use the same SfM prior as ours. AUC@10 (cm; mm for MV-dVRK) is shown in each reconstruction.

Table 2. **Out-of-domain evaluation.** AUC (% ↑) on 4D-DRESS and MV-dVRK, with thresholds in centimeters and millimeters, respectively. The best and second-best results for each metric within each dataset are shown in **bold** and underline, respectively.

	4D-DRESS [54]		MV-dVRK [4]	
	@1 cm	@5 cm	@1 mm	@5 mm
Feed-forward				
Depth Anything 3	2.1	6.3	17.2	50.7
Pi3X (RGB only)	3.6	11.8	3.9	21.1
VGGT-Ω	42.0	81.6	17.7	54.0
Geometry-conditioned Feed-forward				
MapAnything	7.2	29.8	32.2	69.1
Pi3X	45.7	76.9	33.8	65.2
Depth completion				
Any2Full	63.2	79.7	65.3	84.1
Murre	48.5	74.1	59.5	81.7
Feed-forward w/ multi-view geometry				
MASt3R-SfM	51.1	70.1	43.9	64.6
BLASt3R	13.9	39.6	24.6	50.8
GGPT (w/ DA3)	50.7	71.2	67.0	83.1
GGPT (w/ Pi3X)	64.6	82.1	62.6	76.6
Refin3R (w/ DA3) (Ours)	75.1	83.6	**72.3**	**87.4**
Refin3R (w/ Pi3X) (Ours)	**80.2**	**89.6**	68.9	83.7

Table 3. **Ablation study.** AUC (%) at 1, 5, and 10 cm on ScanNet++ and ETH3D, averaged over the 4-, 8-, and 16-view splits. Rows (a)–(c) add one component at a time, and the last two rows compare how the depth of the 3D branch reaches the output on top of (c). †Applied at test time only, without retraining.

		ScanNet++ @1 / @5 / @10↑	ETH3D @1 / @5 / @10↑
	Feed-forward (DA3)	1.0 / 6.9 / 15.1	8.0 / 29.5 / 43.6
	GGPT (w/ DA3)	27.0 / 55.5 / 67.5	29.5 / 59.8 / 71.0
(a)	2D branch, geometry only	34.3 / 64.7 / 76.6	42.8 / 67.4 / 76.3
(b)	+ image features	35.4 / 66.3 / 77.9	44.7 / 68.8 / 77.2
(c)	+ 3D branch & interaction	36.7 / 67.2 / 78.4	46.2 / 69.7 / 77.9
	+ hard substitution†	37.1 / 67.5 / 78.6	46.7 / 70.3 / 78.5
(d)	+ Spatial fusion (Refin3R)	**37.5 / 67.8 / 78.9**	**47.6 / 71.1 / 79.1**

4.4. Ablation study

Tab. 3 ablates Refin3R on ScanNet++ and ETH3D. Rows (a)–(c) add one component at a time, and the last two rows compare how the 3D-branch depth reaches the output. All variants use DA3 and share the same training setting.

Image-grid refinement. Refining the feed-forward depth on the image grid from the geometric channels of Eq. (2) alone (a) already outperforms GGPT by a large margin, showing the benefit of refining on the image grid rather than as points in 3D. Adding the backbone features (b) further improves accuracy, as they provide geometric cues where the SfM depth alone is insufficient.

Table 4. **Efficiency.** Inference time and peak GPU memory on ETH3D, measured on an NVIDIA RTX A6000. The last row includes the DA3 encoder.

Method	Time per scene (s) ↓			Mem. (GB) ↓
	4 views	8 views	16 views	16 views
Any2Full	0.78	1.58	3.17	**5.0**
MapAnything	0.70	1.37	2.76	12.3
Pi3X	0.46	0.88	1.88	10.9
GGPT	4.36	7.20	12.92	6.2
Ours	**0.17**	**0.39**	**0.86**	5.1
Ours + DA3 Backbone	0.35	0.73	1.58	9.8

Bidirectional interaction. Adding the 3D branch with bidirectional interaction (c) brings consistent gains on both datasets. Since the 3D-branch depth is used only for supervision in this variant, these gains come solely from the feature exchange, which provides the 2D branch with multi-view evidence beyond the view's own SfM depth.

Spatial fusion. Substituting the 3D-branch depth on SfM-observed pixels at test time already helps, confirming that the 3D branch refines these pixels more accurately than the 2D branch. However, hard substitution trusts every triangulated pixel equally and leaves its neighbors untouched. Spatial fusion (d) instead anchors this depth with learned confidence and propagates it through learned affinities, yielding the best results.

4.5. Efficiency

Tab. 4 reports the inference time and peak GPU memory on ETH3D. Since all methods in the table receive the same SfM front-end output, we report only the time of each method itself; ours and Any2Full process the views in batches of four. Among them, Refin3R is the fastest at every view count, 2.2× faster than Pi3X and 15× faster than GGPT with 16 views. Our main numbers exclude the DA3 encoder, which already runs to obtain the SfM priors; even including it, Refin3R remains the fastest.

5. Conclusion

We presented Refin3R, which refines feed-forward 3D reconstructions on the image grid of each view, guided by the SfM points of all views aggregated in 3D. Its dual-branch architecture couples dense image features with multi-view SfM evidence through bidirectional interaction, and spatial fusion propagates the refined geometry across the whole image. Refin3R achieves state-of-the-art accuracy on standard and out-of-domain benchmarks, consistently outperforming feed-forward and geometry-guided methods while adding little overhead to the SfM front-end.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009. 2

[2] Paul A. Beardsley, Philip H. S. Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *ECCV*, 1996. 2

[3] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, 2023. 6

[4] Guido Caccianiga, Sergey Prokudin, Yutong Chen, Bernard Javot, Rachael L'Orsa, Omer Burak Aladağ, Yarden Sharon, Jens Rolinger, Ivan Capobianco, Anton Deguet, et al. Mv-dvrk: A multi-viewpoint benchmark for spatial surgical perception. *arXiv preprint arXiv:2609.02717*, 2026. 6, 8

[5] Ruojin Cai, Joseph Tung, Qianqian Wang, Hadar Averbuch-Elor, Bharath Hariharan, and Noah Snavely. Doppelgangers: Learning to disambiguate images of similar structures. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 34–44. IEEE, 2023. 2

[6] Yutong Chen, Yiming Wang, Xucong Zhang, Sergey Prokudin, and Siyu Tang. Ggpt: Geometry grounded point transformer. *arXiv preprint arXiv:2603.11174*, 2026. 1, 2, 3, 6, 7

[7] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 5

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, pages 337–33712. IEEE, 2018. 2

[9] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2025. 2, 3, 6, 7

[10] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17765–17775. IEEE, 2023. 2

[11] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19790–19800. IEEE, 2024.

[12] Johan Edstedt, David Nördstrom, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. Roma v2: Harder better faster denser feature matching. In *European Conference on Computer Vision*, pages 664–682. Springer, 2026. 2, 3

[13] Andrew William Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV*, 1998. 2

[14] Haoyu Guo, He Zhu, Sida Peng, Haotong Lin, Yunzhi Yan, Tao Xie, Wenguan Wang, Xiaowei Zhou, and Hujun Bao. Multi-view reconstruction via sfm-guided monocular depth estimation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5272–5282. IEEE, 2025. 2, 3, 4, 6, 7

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 2, 3

[16] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997. 2, 3

[17] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21594–21603. IEEE, 2024. 2

[18] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3287–3295, 2015. 2

[19] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1071–1081. IEEE, 2025. 2, 3

[20] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction; map-anything. github. io. In *2026 International Conference on 3D Vision (3DV)*, pages 499–509. IEEE, 2026. 1, 2, 3, 6, 7

[21] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 6, 7

[22] Lingyu Kong, Ruicheng Li, Ruicheng Wang, Sicheng Xu, Chengtang Yao, Jianfeng Xiang, and Jiaolong Yang. Moge-3: Fine-detail monocular geometry estimation with self-guided sparse volumetric refinement. *arXiv e-prints*, pages arXiv–2607, 2026. 4, 6

[23] JongMin Lee and Sungjoo Yoo. Dense-sfm: Structure from motion with dense consistent matching. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6404–6414. IEEE, 2025. 2

[24] Jongmin Lee, Seungyeop Kang, and Sungjoo Yoo. Mv-roma: From pairwise matching into multi-view track reconstruction. *arXiv preprint arXiv:2603.27542*, 2026. 2

[25] 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. 2

[26] Vincent Leroy, Philippe Weinzaepfel, Lojze Zust, Yohann Cabon, and Jérôme Revaud. Blast3r: Bundle adjustment

of any image set with multi-view matching and monocular priors. In *European Conference on Computer Vision*, pages 641–662. Springer, 2026. 2, 3, 6, 7

[27] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 1, 2, 3, 6, 7

[28] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the aaai conference on artificial intelligence*, pages 1638–1646, 2022. 5

[29] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5967–5977. IEEE, 2021. 2

[30] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023. 2

[31] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5 (5):5, 2017. 6

[32] Roger Mohr, Long Quan, and Francoise Veillon. Relative 3d reconstruction using multiple uncalibrated images. *The International Journal of Robotics Research*, 1993. 2

[33] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European conference on computer vision*, pages 414–431. Springer, 2020. 4

[34] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024. 2

[35] Zador Pataki, Paul-Edouard Sarlin, Johannes L Schönberger, and Marc Pollefeys. Mp-sfm: Monocular surface priors for robust structure-from-motion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21891–21901. IEEE, 2025. 2

[36] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168. IEEE, 2021. 4

[37] Alexander Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. 3dvnet: Multi-view depth prediction and volumetric refinement. In *2021 International Conference on 3D Vision (3DV)*, pages 700–709. IEEE, 2021. 4

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[39] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946. IEEE, 2020. 2

[40] 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. 2, 6

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

[42] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2538–2547, 2017. 6, 7

[43] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *arXiv preprint arXiv:2104.00680*, 2021. 2

[44] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[45] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. 3

[46] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4): 376–380, 1991. 6

[47] Haotian Wang, Aoran Xiao, Xiaoqin Zhang, Meng Yang, and Shijian Lu. Pacgdc: Label-efficient generalizable depth completion with projection ambiguity and consistency. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7709–7720. IEEE, 2025. 3, 6

[48] Haotian Wang, Meng Yang, Xinhu Zheng, and Gang Hua. Scale propagation network for generalizable depth completion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1908–1922, 2025. 3

[49] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21686–21697. IEEE, 2024. 2

[50] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5294–5306. IEEE, 2025. 1, 2, 3

[51] Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026. 1, 2, 6, 7

[52] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vi-

sion made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709. IEEE, 2024. 1, 2, 6

[53] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 6

[54] Wenbo Wang, Hsuan-I Ho, Chen Guo, Boxiang Rong, Artur Grigorev, Jie Song, Juan Jose Zarate, and Otmar Hilliges. 4d-dress: A 4d dataset of real-world human clothing with semantic annotations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 550–560. IEEE, 2024. 6, 8

[55] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, pages 10481–10497, 2026. 1, 2, 3, 6, 7

[56] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. 3

[57] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16133–16142. Ieee, 2023. 4

[58] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision - 3DV 2013*, pages 127–134, 2013. 2

[59] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4840–4851. IEEE, 2024. 3

[60] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

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

[62] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18527–18536. IEEE, 2023. 3

[63] Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, et al. Ufm: A simple path towards unified dense correspondence with flow. *arXiv preprint arXiv:2506.09278*, 2025. 2, 6

[64] Zhiyuan Zhou, Ruofeng Liu, Taichi Liu, Weijian Zuo, Shanshan Wang, Zhiqing Hong, and Desheng Zhang. Any to full: Prompting depth anything for depth completion in one stage. In *European Conference on Computer Vision*, pages 442–462. Springer, 2026. 2, 3, 4, 6, 7

[65] Yiming Zuo, Willow Yang, Zeyu Ma, and Jia Deng. Omnidc: Highly robust depth completion with multiresolution depth integration. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9287–9297. IEEE, 2025. 2, 3, 4