

# AMB3R: Accurate Feed-forward Metric-scale 3D Reconstruction with Backend

Hengyi Wang   Lourdes Agapito  
Department of Computer Science, University College London

<https://hengyiwang.github.io/projects/amber>

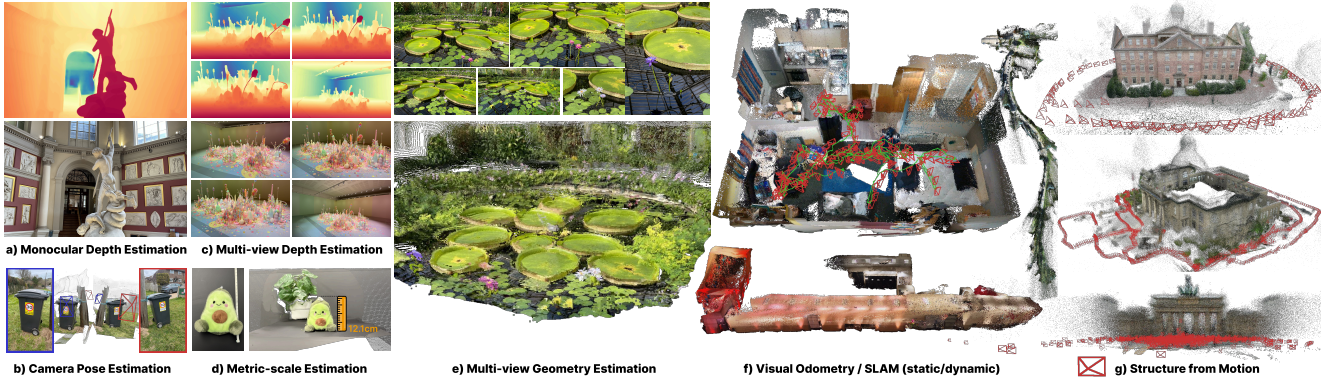


Figure 1. **Overview.** We present AMB3R, a **feed-forward model for metric-scale 3D reconstruction**. AMB3R supports camera pose estimation, monocular/multi-view metric depth/3D reconstruction, and can be seamlessly extended to **visual odometry (VO)/SLAM** and **Structure from Motion (SfM)** with **no task-specific fine-tuning or test-time optimization**. We use in-the-wild images for a)-e), scenes from Co-SLAM [123], TTT3R [19], and KITTI [36] for f), scenes from COLMAP [99], Tanks&Temples [55], and IMC PhotoTourism [49] (all images) for g). No confidence threshold is used. f) & g) results are randomly down-sampled to 3 million points for visualization.

## Abstract

We present AMB3R, a multi-view feed-forward model for dense 3D reconstruction on a metric-scale that addresses diverse 3D vision tasks. The key idea is to leverage a sparse, yet compact, volumetric scene representation as our backend, enabling geometric reasoning with spatial compactness. Although trained solely for multi-view reconstruction, we demonstrate that AMB3R can be seamlessly extended to uncalibrated visual odometry (online) or large-scale structure from motion without the need for task-specific fine-tuning or test-time optimization. Compared to prior pointmap-based models, our approach achieves state-of-the-art performance in camera pose, depth, and metric-scale estimation, 3D reconstruction, and even surpasses optimization-based SLAM and SfM methods with dense reconstruction priors on common benchmarks.

## 1. Introduction

Pointmaps have emerged as the cornerstone of modern 3D foundation models. By exploiting the one-to-one correspondence between 2D pixels and 3D point coordinates, we can train a neural network to tackle various 3D vision tasks.

Yet, this raises a fundamental question: *Is the mapping from 2D pixels to 3D scene points truly one-to-one?*

In practice, this is not the case. Due to visual overlap, multiple pixels often correspond to the same 3D point. This many-to-one mapping, known as **correspondence**, lies at the heart of decades of research in 3D vision. Early sparse reconstruction methods [3, 23] leverage correspondences between interest points to recover camera parameters and sparse geometry. Multi-view stereo (MVS) methods [34] construct cost volumes for depth estimation, which essentially match pixel correspondences within the volume. For dense 3D reconstruction, approaches ranging from traditional methods (e.g., KinectFusion [85]) to modern neural implicit models (e.g., DeepSDF [90], NeRF [80], NeuralRecon [109]) adopt compact 3D representations such as TSDF grids, feature grids, or coordinate-based networks. Despite their diverse forms, these representations share a commonality—**spatial compactness** (i.e., the same 3D coordinate can only have a unique property). This compactness enforces the fusion of multiple corresponding observations of the same scene point into a coherent geometry.

Recent feed-forward models such as Spann3R [122] or VGGT [124] follow the DUST3R [128] paradigm, formulating the task as per-pixel 2D-to-3D regression. While the pointmap output space implicitly encourages multiple cor-

responding pixels to have the same 3D location regardless of viewpoint, the network itself operates on 2D grids and lacks explicit geometric reasoning or spatial compactness.

In this paper, we propose AMB3R, a feed-forward reconstruction pipeline with a compact scene representation as backend. We use VGGT as our front-end to predict features and geometry, with an additional scale head trained to recover the metric scale. For the backend, we adopt sparse voxels as our spatial data structure. Thanks to decades of progress in 3D geometry processing, we organize these sparse voxels into a 1D sequence via space-filling curves and process them using a transformer-based architecture, enabling efficient reasoning in compact 3D space. The fused features are injected back into the front-end decoder via zero-convolution layers as in ControlNet [146], allowing the model to benefit from pre-trained weights and the learned confidence function of the front-end, substantially reducing the training cost ( $\sim 80$  H100 GPU hours).

Furthermore, we demonstrate that AMB3R can be seamlessly extended to perform uncalibrated visual odometry or structure from motion without the need for fine-tuning or test time optimization. To sum up, our contributions are:

1. **A Compact Backend Scene Representation** for a feed-forward pointmap-based foundation model that enables explicit 3D reasoning with spatial compactness.
2. **Metric-scale Reconstruction** with a light-weight scale head to recover metric depth information.
3. **Uncalibrated Visual Odometry** without the need for fine-tuning or an optimization-based backend. Notably, we show for the first time, feed-forward visual odometry can surpass its optimization-based counterparts.
4. **Feed-forward Structure from Motion** capable of large-scale reconstruction without optimization.
5. **State-of-the-Art Performance** on 7 tasks across 13 datasets, surpassing existing 3D foundation models and only requiring academic-level training resources ( $\sim 80$  H100 GPU hrs) to train the metric-scale head and new backend on top of frozen VGGT [124] weights.
6. **Open Source Release** of code, weights, and an evaluation toolkit to facilitate future research.

## 2. Related Work

### 2.1. 3D Scene Representations

A 3D scene can be represented through various scene properties – such as signed distance functions (SDFs), volume density, radiance, surface normals, or learned features – stored in spatial data structures like voxel grids, point clouds, meshes, or even MLPs. Among these, voxel grids are especially popular in geometric deep learning, as they naturally extend 2D pixels into 3D. Since grids grow cubically with scene size, many sparse alternatives have been

introduced, including lossless representations such as voxel hashing [86] and octrees [46, 78], as well as lossy ones such as hash feature grids [81], triplanes [16, 91], and vector-matrix [17], many of which rely on a low-rank approximation of the 3D scene. Point clouds are also a widely used spatial data-structure as they are readily available from posed RGB-D images. Unlike voxels, point clouds are irregular and typically sampled only around visible surface regions. To process such data, HilbertNet [18] proposes mapping points onto a Hilbert curve [1, 39] – a fractal space-filling curve that traverses the entire 3D space while preserving spatial locality – thereby serializing the point cloud into a 1D sequence that can be efficiently processed with a transformer-based architecture [64, 131, 132, 150].

### 2.2. Pointmap-based Reconstruction

Pointmaps, or scene coordinates, were originally introduced by SCoRF [105] as a scene-specific 3D representation for camera localization [7, 9, 10, 96]. DUST3R [128] and MAST3R [58] extend this concept by training neural networks to regress pointmaps from two-view inputs in a generalizable manner. Since pointmaps inherently encode both camera information and scene geometry, they provide a unified representation that enables a single network to address diverse geometric vision tasks. To scale feed-forward reconstruction beyond two views, Spann3R [122] and others [13, 19, 20, 54, 57, 63, 69, 73, 75, 126, 133, 156] adopt an incremental paradigm in a similar spirit to incremental SfM/SLAM [23, 99], whereas VGGT [124] and others [30, 113, 120, 129, 136] follow a global paradigm that jointly processes all images as in global SfM [3, 88]. Beyond multi-view dense reconstruction, extensions have been proposed for calibrated settings [47, 52], camera calibration [72], camera localisation [27, 42, 68, 70], monocular depth estimation [60, 71, 111], dynamic reconstruction [33, 48, 59, 76, 114, 147], novel view synthesis [44, 61, 65, 66, 79, 103, 135, 148], surface reconstruction [154], semantic segmentation [157], spatial reasoning [32, 41, 93], and scientific imaging [145]. Since feed-forward approaches are typically constrained to moderate scene sizes [122] or limited image counts [124], several works [24, 28, 83, 143, 153] also explore optimization-based formulations to scale up reconstruction.

## 3. Preliminaries: Pointmap-based Front-end

Given a set of input images  $\{I_t\}_{t=1}^T$ , pointmap-based models independently encode each image into visual features:

$$F_t = \text{Encoder}(I_t) \quad (1)$$

Since the pointmap is defined up to a rigid transformation, we fix an anchor image and assume its camera pose to be the identity. The task becomes to predict a pointmap per image



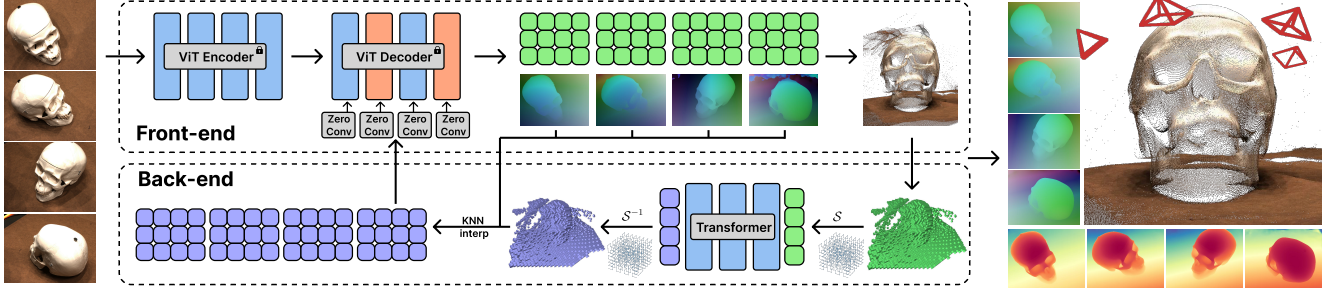


Figure 2. **Overview of AMB3R.** AMB3R consists of a front-end that predicts pointmaps and geometric features, and a back-end that fuses them into sparse voxels, which are serialized into a 1D sequence, processed by a transformer, and unserialized back to 3D. Per-pixel features are obtained via KNN interpolation and injected into the frozen front-end via zero-convolution for final prediction.

expressed in the anchor coordinate system. To achieve this, the model requires a temporal encoding that (i) disentangles visual features across images while being generalizable to many images, and (ii) identifies the anchor image to provide a consistent reference frame. Existing models can be categorized into three paradigms:

(a) **Network-based temporal encoding.** DUST3R [128] employs two intertwined decoders for two images, where the coordinate system is defined by the reference decoder:

$$P_1^{(1)} = \text{Decoder}^1(I_1), \quad P_2^{(1)} = \text{Decoder}^2(I_2). \quad (2)$$

This design choice satisfies condition (ii), but inherently limits its scalability to only two images.

(b) **Causal temporal encoding.** Spann3R [122] proposes an incremental reconstruction framework, where each incoming frame queries accumulated memory features:

$$P_t^{(1)} = \text{Decoder}(M_{1:t-1}, F_t). \quad (3)$$

This formulation naturally satisfies (i) and (ii) in a causal setting. By sparsifying memory features, Spann3R can process long sequences without exploding GPU memory.

(c) **Token-based temporal encoding.** VGGT [124] introduces two tokens for the anchor and other images and uses alternating attention as a generalizable temporal encoding:

$$\{P_t^{(1)}\}_{t=1}^T = \text{Decoder}(F_1 + e_1, \{F_t + e_2\}_{t=2}^T). \quad (4)$$

This enables offline reconstruction with densely connected attention, leading to superior reconstruction quality.

**Coordinate normalization.** Pointmap prediction from images is always up to an unknown scale. To address this during training, DUST3R and Spann3R normalize both the predicted and ground-truth geometry, whereas VGGT normalizes only the ground-truth geometry and trains the model to predict in this normalized space. This forms a canonical space and generally leads to more stable training.

**Confidence-aware supervision.** These models are trained with a confidence loss on the ground-truth geometry:

$$\mathcal{L}_{\text{conf}} = \sum_t \sum_{i \in \mathcal{I}} (C_t^i \mathcal{L}_{\text{reg}}(i) - \alpha \log C_t^i), \quad (5)$$

where  $\mathcal{I}$  denotes the set of valid pixels. Confidence  $C_t$  is learned in a self-supervised way that depends on both the data distribution and the training recipe. Consequently, fine-tuning a pre-trained model on different data distributions using a modified training setup without careful initialization may lead to catastrophic forgetting of the learned confidence, potentially degrading reconstruction performance.

## 4. Method

### 4.1. Metric-scale Reconstruction

VGGT [124] predicts pointmaps normalized by the median distance across all frames (Sec. 3), suggesting that it might implicitly encode metric cues. Thus, we employ a scale head to recover metric scale from its frozen features.

A simple way is to regress the global scale difference between GT and the prediction via a ROE solver [127]. Since this global information depends on ALL frames, we use all intermediate features after the encoder for regression. In practice, though, we find this approach is difficult to train and prone to overfitting, as the scale difference can vary with different frame combinations or even frame order.

Instead, for each frame we regress the **metric log depth** of the pixel with median predicted depth. This captures the intrinsic, per-frame property recoverable from individual encoder features, avoiding a strong dependence on the model’s (global) prediction, making the task substantially easier. In addition we include the decoder depth feature as guidance. At inference, we estimate per-frame scales and take their median to align the reconstruction to metric space.

### 4.2. Backend

Given pointmap predictions  $\{P_t^{(1)}\}_{t=1}^T$  and corresponding geometric features  $\{G_t\}_{t=1}^T$  extracted from visual features and decoder features, we align their resolution and construct a sparse voxel grid  $\mathcal{V}$ . Voxel features  $\{H_i\}_{i=1}^N$  are obtained by averaging features within the same voxel  $\mathcal{V}_i$ :

$$H_i = \frac{1}{|\mathcal{P}_i|} \sum_{(t, \mathbf{u}) \in \mathcal{P}_i} G_t[\mathbf{u}], \quad (6)$$

where  $\mathcal{P}_i = \{(t, \mathbf{u}) \mid P_t^{(1)}[\mathbf{u}] \in \mathcal{V}_i\}$ ,  $\mathbf{u}$  denotes the pixel coordinates. We set the voxel size to 0.01 in normalized space, naturally enabling adaptive voxel resolution depending on the scene scale. The sparse voxel grid is serialized into 1D feature sequences via space-filling curves, processed by a transformer, and un-serialized back to voxel space:

$$\{\hat{H}_i\}_{i=1}^N = (\mathcal{S}^{-1} \circ f_\theta \circ \mathcal{S}) (\{H_i\}_{i=1}^N), \quad (7)$$

where  $\mathcal{S}$  denotes the serialization and  $f_\theta$  is implemented via Point Transformer v3 [132], a U-Net-like architecture for efficient geometry processing. We then use K-nearest neighbor interpolation to obtain per-point features:

$$\tilde{G}_t[\mathbf{u}] = \text{KNN}(P_t^{(1)}[\mathbf{u}], \{\hat{H}_i\}_{i=1}^N). \quad (8)$$

These features are fused back to each decoder layer using zero convolution, allowing the model to reuse the attention and confidence functions learned by the front-end model while substantially reducing training cost. See Fig. 2.

### 4.3. Training

We freeze our front-end model (i.e., VGGT [124]) and only train our backend. Our loss is the same as VGGT for pointmap, depth, and camera pose (except tracking loss):

$$\mathcal{L} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pointmap}} + \mathcal{L}_{\text{camera}}. \quad (9)$$

Since our training loss, training data, and pre-processing scripts differ from those in VGGT, the expected canonical scale learned by minimizing  $\mathcal{L}$  may not align with that of VGGT. Directly training on normalized data would therefore force the model to waste capacity compensating for this scale mismatch. To mitigate this, we align the predicted geometry with the normalized ground truth using the ROE [127] solver before supervision. For pointmap prediction, we estimate a single scale factor per sequence, while for depth prediction we relax this constraint and estimate an independent scale per image. Because the front-end (VGGT) is trained to produce geometry in a consistent scale and remains frozen, this strategy allows our backend to focus on refining fine structural details without compromising global consistency. The backend is trained for 40 epochs on a mixture of 12 datasets, each epoch containing 2000 samples—a total of 80K samples, still less than a single epoch of VGGT training data—with 5–16 frames per sample, requiring roughly 50 H100 GPU hours to train.

### 4.4. Uncalibrated Visual Odometry

To extend type-(c) methods into a visual odometry (VO) or SLAM system, a straightforward approach is to sequentially build overlapping submaps and estimate both the relative transformations and scale differences between them using the Kabsch–Umeyama algorithm [50, 117]. Since such

alignment introduces non-negligible errors, existing methods usually rely on an optimisation-based backend [74]. However, we argue this ignores a strong prior in pointmap-based methods: predictions are always expressed in the reference (first) frame coordinate system up to an unknown (median) scale. That is to say, it is not necessary to estimate the transformation for coordinate system alignment.

**Keyframes as memory.** If we rewrite Eq. 4 as

$$\{P_t^{(1)}\}_{t=1}^{T-1}, P_T^{(1)} = \text{Decoder}(\mathcal{M}_{1:T-1}, F_T + e_2), \quad (10)$$

where  $\mathcal{M}_{1:T-1} = \{F_1 + e_1, \{F_t + e_2\}_{t=2}^{T-1}\}$ . Compared to Eq. 3 under the streaming setting, type-(c) methods can be considered as a special memory network that can also update the previous predictions with non-causal attention. Thus, we can use carefully selected keyframes as our memory, enabling the model to run in a visual odometry mode.

**Keyframe selection.** We can measure the pose distance as in VGGT [124] and our training frame sampling strategy:

$$D_{i,j} = \arccos\left(\frac{\text{Tr}(R_j R_i^T) - 1}{2}\right) + \lambda \|\tau'_i - \tau'_j\|_2, \quad (11)$$

where  $R$  and  $\tau'$  are rotation and normalized translation vector. We set a minimal pose distance with respect to the poses of existing keyframes  $\eta_d = 0.15$  and iteratively add the frame with the highest confidence within each mapping window as our new keyframe. These keyframes serve as our memory for the next mapping window of  $N_w = 8$  new frames. Since our prediction is up to the median scale of all input frames, we estimate the scale of the new window  $w$  prediction relative to the global map based on keyframe  $k \in KF_w$ :

$$s^w = \text{ROE}(P_k^{(1)}, P_k^{(1),w}) \quad (12)$$

After the scale estimation, we will fuse the new window prediction to the existing map in a running average manner:

$$P_k \leftarrow (C_k P_k + C_k^w s^w P_k^w) / (C_k + C_k^w), \quad (13)$$

$$m \leftarrow (C_k m + C_k^w m^w / s^w) / (C_k + C_k^w), \quad (14)$$

$$\tau_k \leftarrow (C_k \tau_k + C_k^w s^w \tau_k^w) / (C_k + C_k^w), \quad (15)$$

$$q_k \leftarrow \text{slerp}(q_k, q_k^w, C_k^w / (C_k + C_k^w)), \quad (16)$$

$$C_k \leftarrow C_k + C_k^w. \quad (17)$$

where  $q_k$  is the quaternion vector of the rotation matrix.  $C_k$  is the predicted confidence.  $m$  is the metric-scale factor.

**Active keyframe management.** Once the number of active keyframes reaches  $N_{max} = 10$ , we resample  $N_{min} = 7$  keyframes from all historical keyframes. The last keyframe will always be preserved, and we will sample  $N_{topk}$  keyframes that have the least pose distance to the last keyframe. To encourage the model to use earlier frames to form a loop, we set a backward search window  $\eta_b = 0.4$  and iteratively add the earliest keyframe that fits this search

window, resulting in  $N_{old}$  in total. A keyframe that has the minimal sum pose distance to existing keyframes is sampled to serve as the first frame. Note that once the max pose distance of the active keyframes are larger than  $\eta_{max} = 1.2$ , we resample keyframes to ensure the model only covers a reasonable scene region within the training distribution.

**Coordinate alignment.** After keyframe resampling, the first keyframe in memory might not be the first frame, resulting in a different coordinate system between our global map and local prediction. Instead of jointly estimating scale and relative transformation from the local map to the global map, we can first map our global map into the local map:

$$P_k^{(k_0)} = T_{k_0}^{-1} P_k^{(1)}, \quad (18)$$

where  $T_{k_0} = [R_{k_0}, \tau_{k_0}]$ . We can then estimate the scale between  $P_k^{(k_0)}$  and  $P_k^{(k_0),w}$ . After scaling the camera pose, we can obtain the relative pose of each corresponding keyframe and use their weighted average to map the local map back to the global frame of reference and fuse them as in Eq. 13-17, thus avoiding explicit Kabsch–Umeyama alignment.

**Robust estimation.** To reduce inference cost without sacrificing performance, we only run the backend when the confidence of the front-end prediction falls below a pre-defined threshold. Based on both the confidence and a self-consistency score from the raw pointmap and the unprojected pointmap from the front-end and backend, we either use backend prediction or blend them, ensuring robust performance in the sequential system.

#### 4.5. Structure from Motion

Feed-forward pointmap-based models, while efficient, are typically limited by the scale and motion range observed during training. Moreover, type-(c) approaches exhibit quadratic computational complexity with respect to the number of images, making them impractical for large-scale reconstruction. To extend AMB3R into a scalable feed-forward SfM system, we adopt a divide-and-conquer strategy: images are partitioned into small clusters and solved following incremental SfM without optimization.

**Image clustering.** For each given image  $I_t$ , we extract its feature descriptor  $\bar{F}_t \in \mathbb{R}^C$ , apply feature whitening, and construct the feature distance matrix  $D^F \in \mathbb{R}^{T \times T}$ . Furthest Point Sampling (FPS) with iterative splitting and merging is then applied to construct image clusters, each containing between  $N_{cmin}$  and  $N_{cmax}$  images.

**Coarse registration.** We initialize the map using the cluster with highest confidence and maintain a global keyframe list with  $\eta_d = 0.2$ . After initialization, the map is incrementally built by: (1) identifying the top-k=5 unmapped clusters with the smallest feature distance to existing keyframes, (2) mapping with existing keyframes, and 3) updating global map using the cluster with highest confidence. When the number of global keyframes exceeds  $N_{kmax} = 8$ , we partition

| Method            | NYUv2 [84] |                 | KITTI [36] |                 | ETH3D [100] |                 | ScanNet [22] |                 | DIODE [119] |                 |
|-------------------|------------|-----------------|------------|-----------------|-------------|-----------------|--------------|-----------------|-------------|-----------------|
|                   | Rel        | $\delta_{1.25}$ | Rel        | $\delta_{1.25}$ | Rel         | $\delta_{1.25}$ | Rel          | $\delta_{1.25}$ | Rel         | $\delta_{1.25}$ |
| Omnidata [29]     | 7.4        | 94.5            | 14.9       | 83.5            | 16.6        | 77.8            | 7.5          | 93.6            | 33.9        | 74.2            |
| DepthAny v2 [137] | 4.3        | 97.9            | 8.0        | 94.4            | 5.7         | 98.3            | (4.2)        | (97.9)          | 21.6        | 75.2            |
| Marigold [51]     | 5.5        | 96.4            | 9.9        | 91.6            | 6.5         | 96.0            | 6.4          | 95.1            | 30.8        | 77.3            |
| Diffusion-E2E[35] | 5.2        | 96.6            | 9.6        | 91.9            | 6.2         | 95.9            | 5.8          | 96.2            | 30.2        | 77.9            |
| MoGe [127]        | 3.6        | 98.0            | 5.5        | 97.6            | 3.3         | 98.8            | 3.5          | 98.3            | 22.4        | 82.3            |
| VGGT [124]        | 3.6        | 98.0            | 8.8        | 92.7            | 3.8         | 97.9            | (2.7)        | (98.8)          | 26.9        | 79.1            |
| AMB3R             | 3.0        | 98.9            | 7.3        | 95.4            | 3.2         | 98.8            | (2.7)        | (98.9)          | 24.7        | 80.7            |

Table 1. **Monocular depth estimation.** Our method achieves competitive zero-shot monocular depth performance.

|             | DUS3R [128] | MASt3R [58] | Fast3R [136] | VGGT [124] | VGGT [124] | AMB3R |
|-------------|-------------|-------------|--------------|------------|------------|-------|
| Re10K [152] | 67.7        | 76.4        | 72.7         | 85.3       | 81.8       | 86.3  |

Table 2. **Camera Pose Estimation.** We evaluate AUC@30 by randomly sampling 10 frames from the entire sequence.

the keyframe list into sub-clusters based on pose distance. Subsequent clusters are then mapped against the top-k=5 smaller sub-clusters to improve registration accuracy. For each frame, only the prediction from the highest-confidence mapping is kept; low-confidence frames are marked as unregistered and mapped again after registration finishes. Map update and coordinate alignment follow Sec. 4.4.

**Global mapping.** After coarse registration, we perform a two-stage mapping as refinement. First, we refine all keyframes: starting from the first, we find its top-k closest keyframes within  $\eta_r = 1.5$  and perform mapping. We then apply a confidence-prioritized breadth-first search (BFS), adding keyframes in the current mapping window to a priority queue sorted by confidence, thereby traversing the entire keyframe graph, and mapping each keyframe with its closest top-k keyframes. Finally, each non-keyframe is refined by mapping with the top-k closest frames.

## 5. Experiments

### 5.1. Setup

**Benchmarks.** We evaluate across 7 tasks on 13 datasets to show the effectiveness and versatility of our approach:

- **Camera Pose Estimation.** We evaluate against SOTA pointmap-based methods on RealEstate10K [152].
- **Monocular Depth Estimation.** We follow prior works [51] and show zero-shot generalization of our model on five benchmarks: NYUv2 [84], KITTI [36], ETH3D [100], ScanNet [22], and DIODE [119].
- **Multi-View (Metric) Depth Estimation.** We compare against various (metric) depth estimation methods on RMVDB [102], using KITTI [36], ETH3D [100], ScanNet [22], DTU [2], and Tanks & Temples [55] datasets.
- **3D Reconstruction.** We benchmark against pointmap-based models on ETH3D [100] and DTU [2] (RMVDB splits), as well as 7Scenes [105] (Spann3R [122] splits). Note that Replica [107] is within DUS3R [128] and VGGT [124] training data, and we remove NRGBD [5] due to the potential overlap with synthetic training data.



| Approach                                                    | KITTI                   | ScanNet                 | ETH3D                   | DTU                     | T&T                     | Average                 |
|-------------------------------------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|                                                             | rel ↓ $\delta_{1.03}$ ↑ | rel ↓ $\delta_{1.03}$ ↑ | rel ↓ $\delta_{1.03}$ ↑ | rel ↓ $\delta_{1.03}$ ↑ | rel ↓ $\delta_{1.03}$ ↑ | rel ↓ $\delta_{1.03}$ ↑ |
| <b>a) Classic approaches</b>                                |                         |                         |                         |                         |                         |                         |
| COLMAP [99]                                                 | 12.0 58.2               | 14.6 34.2               | 16.4 55.1               | 0.7 96.5                | 2.7 95.0                | 9.3 67.8                |
| <b>b) Single-view depth</b>                                 |                         |                         |                         |                         |                         |                         |
| UniDepthV2 [92]                                             | 4.0 55.3                | (2.1) (82.6)            | 3.7 66.2                | 3.2 72.3                | 3.6 68.4                | 3.3 68.9                |
| DepthAny V2 [137]                                           | 6.6 38.6                | 4.0 58.6                | 4.7 56.5                | 2.6 74.7                | 4.5 57.5                | 4.8 54.1                |
| <b>c) Depth from frames and poses (w/ per-image range)</b>  |                         |                         |                         |                         |                         |                         |
| Vis-MVSNet [144]                                            | 9.5 55.4                | 8.9 33.5                | 10.8 43.3               | (1.8) (87.4)            | 4.1 87.2                | 7.0 61.4                |
| MVSFormer++ [15]                                            | 4.4 65.7                | 7.9 39.4                | 7.8 50.4                | (0.9) (95.3)            | 3.2 88.1                | 4.8 67.8                |
| <b>d) Depth from frames and poses (w/o per-image range)</b> |                         |                         |                         |                         |                         |                         |
| Robust MVD [102]                                            | 7.1 41.9                | 7.4 38.4                | 9.0 42.6                | 2.7 82.0                | 5.0 75.1                | 6.3 56.0                |
| MVSA [45]                                                   | 3.2 68.8                | 3.7 62.9                | 3.2 68.0                | 1.3 95.0                | 2.1 90.5                | 2.7 77.0                |
| <b>e) Depth from frames (w/o poses)</b>                     |                         |                         |                         |                         |                         |                         |
| DeMoN [118]                                                 | 15.5 15.2               | 12.0 21.0               | 17.4 15.4               | 21.8 16.6               | 13.0 23.2               | 16.0 18.3               |
| DUST3R [128]                                                | 5.4 49.5                | (3.1) (71.8)            | 3.0 76.0                | 3.9 68.6                | 3.3 75.1                | 3.7 68.2                |
| Spann3R [122]                                               | 7.9 36.2                | (3.3) (67.1)            | 5.7 58.6                | 3.5 65.2                | 4.7 58.5                | 5.0 57.1                |
| Pow3R [47]                                                  | 5.7 45.7                | (3.2) (68.8)            | 3.0 74.7                | 3.0 74.3                | 3.3 76.6                | 3.6 68.0                |
| MUS3R [13]                                                  | 4.5 55.0                | (4.0) (59.8)            | 2.5 80.3                | 4.6 55.4                | (2.6) (80.4)            | 3.7 66.2                |
| VGGT [124]                                                  | 4.5 59.6                | (2.3) (80.8)            | 1.8 86.3                | 0.9 95.6                | 2.4 84.1                | 2.4 81.3                |
| $\pi 3^{\ddagger}$ [129]                                    | 2.8 72.9                | (2.0) (83.6)            | 1.3 92.4                | 1.3 91.8                | 1.8 87.3                | 1.8 85.6                |
| MapAny <sup>†</sup> [52]                                    | 4.0 59.4                | 4.0 60.5                | 2.8 73.2                | 3.9 63.7                | 3.3 73.0                | 3.6 66.0                |
| <b>AMB3R</b>                                                | 2.8 74.4                | (1.9) (85.8)            | 1.4 90.9                | 0.9 95.1                | 1.7 90.2                | 1.7 87.3                |

Table 3. **Multi-view depth estimation** on RMVDB [102].  $\ddagger$  means concurrent works, and () indicate training dataset. Please refer to supplement for full table.

- **Video Depth Estimation.** We evaluate on Sintel [11], BONN [87], and KITTI [36] without finetuning.
- **VO/SLAM.** Our model is compared against traditional sparse/dense and uncalibrated VO/SLAM methods on static and dynamic scenes from TUM [108] and ETH3D SLAM [101] and pseudo groundtruth on 7scenes [104].
- **SfM.** We compare against SOTA optimization-based SfM on ETH3D [100] and Tanks&Temples [55] datasets.

**Metrics.** For depth estimation, we report absolute relative error (rel) and inlier ratios of 3% ( $\delta_{1.03}$ ) and 25% ( $\delta_{1.25}$ ). For 3D reconstruction, we evaluate rel ratio as well as accuracy, and completeness, which measures distances between prediction and ground-truth. We remove ICP alignment in Spann3R [122] evaluation as pointmap has exact correspondence with ground-truth in contrast to prior works [123, 155] that requires randomly sampling points from their mesh. We use median-scale alignment for all depth & 3D reconstruction tasks except metric and monocular depth estimation. For all VO/SLAM tasks, we evaluate the trajectory using ATE RMSE with evo package [37]. All distance metrics are reported with same unit [cm].

## 5.2. Monocular depth estimation

We compared our zero-shot monocular depth estimation performance in Tab. 1 with VGGT [124] and various representative works. Our model shows consistent improvement over VGGT and SOTA performance on NYUv2 [84] and ETH3D [100], outperforming models specifically trained for monocular depth estimation on those datasets.

## 5.3. Camera pose estimation.

Following [124, 128], we evaluate camera pose estimation (see Tab. 2) by randomly sampling 10 frames per scene and evaluating AUC@30. For VGGT, we report their official results (gray) (which include some non-accessible datasets),

| Method                   | KITTI [36]                          | ScanNet [22]                        | ETH3D [100]                         | DTU [2]                             | T&T [55]                            |
|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
|                          | rel $\delta_{1.03}$ $\delta_{1.25}$ | rel $\delta_{1.03}$ $\delta_{1.25}$ | rel $\delta_{1.03}$ $\delta_{1.25}$ | rel $\delta_{1.03}$ $\delta_{1.25}$ | rel $\delta_{1.03}$ $\delta_{1.25}$ |
| MAS3R [58]               | 56.8 0.1 8.4 (10.5) (27.0)          | (91.0) 40.4 10.3 26.7 129.0         | 1.7 8.1 63.7 0.0 1.5                |                                     |                                     |
| MUS3R [13]               | 82.4 0.0 0.0 (40.2) (0.7)           | (6.2) 63.3 1.0 3.9 46.8             | 2.6 25.3 77.0 0.0 0.0               |                                     |                                     |
| Spann3R [122]            | 16.4 18.9 76.5 (33.1) (3.3)         | (23.0) 25.1 18.6 65.3 67.8          | 8.7 42.7 23.3 17.6 57.4             |                                     |                                     |
| CUT3R [126]              | 28.6 4.1 29.5 (9.9) (19.2)          | (96.3) 29.4 6.4 40.7 138.5          | 0.7 2.3 46.7 0.1 1.9                |                                     |                                     |
| MapAny <sup>†</sup> [52] | 12.8 16.1 78.2 31.9 7.7             | 41.2 15.4 26.6 83.1 718.5           | 0.0 0.2 24.6 15.3 41.5              |                                     |                                     |
| <b>AMB3R</b> (dec. sc.)  | 4.5 56.5 97.8 (9.3) (41.4)          | (91.7) 12.2 42.1 83.9 96.2          | 25.6 59.3 9.4 42.6 83.8             |                                     |                                     |
| <b>AMB3R</b> (enc. z)    | 8.2 27.9 95.6 (9.3) (26.9)          | (95.2) 8.5 48.3 90.4 240.9          | 30.3 60.9 6.3 49.9 95.7             |                                     |                                     |

Table 4. **Multi-view metric-scale depth estimation** on RMVDB [102].  $\ddagger$ : concurrent work.

| Method | ETH3D                    |       |       | DTU   |       |      | 7-Scenes |       |      |       |
|--------|--------------------------|-------|-------|-------|-------|------|----------|-------|------|-------|
|        | rel↓                     | Acc↓  | Cpl↓  | rel↓  | Acc↓  | Cpl↓ | rel↓     | Acc↓  | Cpl↓ |       |
| Onl.   | Spann3R [122]            | 24.96 | 47.63 | 31.87 | 4.96  | 1.89 | 0.64     | 8.58  | 4.81 | 5.48  |
|        | MUS3R [13]               | 19.91 | 45.19 | 82.72 | 5.26  | 1.68 | 1.09     | 11.57 | 8.94 | 14.97 |
|        | CUT3R [126]              | 18.83 | 38.90 | 22.93 | 9.11  | 3.75 | 0.96     | 6.32  | 2.88 | 3.26  |
|        | VGGT [124]               | 6.02  | 12.81 | 11.89 | 0.83  | 0.22 | 0.08     | 5.51  | 2.32 | 3.51  |
| Offl.  | $\pi 3^{\ddagger}$ [129] | 5.82  | 10.54 | 10.42 | 1.57  | 0.50 | 0.29     | 5.92  | 2.60 | 3.42  |
|        | MapAny <sup>†</sup> [52] | 11.20 | 21.87 | 19.76 | 10.37 | 4.35 | 1.54     | 7.61  | 3.48 | 3.33  |
|        | AMB3R                    | 4.64  | 9.98  | 9.69  | 0.81  | 0.22 | 0.08     | 4.74  | 1.74 | 2.84  |

Table 5. **Multi-view 3D Reconstruction [cm]**. For ETH3D and DTU, we use image tuples from RMVDB [102]. For 7 scenes, we use frames as in Spann3R with improved GT poses (See Tab. 7).

and our run of VGGT on datasets available to us. AMB3R outperforms VGGT in both cases with our performance gain resulting purely from the improved geometric features.

## 5.4. Multi-view (metric) depth estimation

We conduct a comprehensive evaluation on RMVDB [102] in Tab. 3, where our model establishes a new SOTA on RMVDB, outperforming concurrent pointmap-based models ( $\pi 3$  [129] and MapAnything [52]) while requiring one order of magnitude less training resources. Our model also outperforms SOTA depth estimation model MVSA [45], which requires ground-truth poses and intrinsics.

For metric depth evaluation (Tab. 4), we report results under both 3% and 25% inlier thresholds, as 3% is overly strict for metric-depth methods. Our approach achieves the best overall performance under both metrics. Compared to directly regressing scale-differences from decoder features, our log-depth regression yields overall better results. On DTU, larger absolute errors occur in scenes with toy buildings placed on white tables with dark backgrounds, which some models mistakenly interpret as real structures.

## 5.5. 3D reconstruction

We evaluate 3D reconstruction in Tab. 5 and show some qualitative examples in Fig. 3. For object-level reconstruction (DTU), both our model and VGGT achieve millimeter accuracy with less than 1% error. For outdoor (ETH3D) and indoor (7-scenes) scenes, we achieve SOTA performance compared to concurrent work  $\pi 3$  [129].

## 5.6. Dynamic reconstruction

We evaluate video depth estimation in Tab. 6. To align with other depth estimations tasks for static scenes, we additionally report inlier ratio of 3% for a reality check. Despite



Figure 3. **Qualitative showcase** of generalization to in-the-wild images such as Longmen Grottoes [151].

| Method        | Sintel |                         |                         | Bonn |                         |                         | KITTI |                         |                         |
|---------------|--------|-------------------------|-------------------------|------|-------------------------|-------------------------|-------|-------------------------|-------------------------|
|               | rel↓   | $\delta_{1.03}\uparrow$ | $\delta_{1.25}\uparrow$ | rel↓ | $\delta_{1.03}\uparrow$ | $\delta_{1.25}\uparrow$ | rel↓  | $\delta_{1.03}\uparrow$ | $\delta_{1.25}\uparrow$ |
| Spann3R [122] | 55.2   | 7.3                     | 46.3                    | 5.5  | 40.9                    | 97.8                    | 24.7  | 10.4                    | 64.4                    |
| CUT3R [126]   | 58.4   | 9.6                     | 46.6                    | 5.1  | 47.4                    | 98.0                    | 13.6  | 18.0                    | 81.0                    |
| VGGT [124]    | 28.0   | 13.6                    | 61.8                    | 3.6  | 63.2                    | 98.4                    | 8.2   | 40.4                    | 90.9                    |
| $\pi^3$ [129] | 31.9   | 13.3                    | 66.8                    | 3.0  | 72.0                    | 98.8                    | 3.6   | 59.5                    | 98.8                    |
| <b>AMB3R</b>  | 23.6   | 14.0                    | 62.7                    | 3.3  | 66.0                    | 98.5                    | 4.1   | 53.8                    | 98.3                    |

Table 6. **Video depth estimation on Sintel, Bonn, and KITTI.** We evaluate all methods on all scenes from Sintel and Bonn, rather than selecting 14 and 4 scenes as in prior works. In contrast to Pi3 [129], we did not do specific finetuning on dynamic datasets.

|                     | chess | fire | head | office | pump | redkit | stair | avg  |
|---------------------|-------|------|------|--------|------|--------|-------|------|
| Pseudo GT [85]      | 3.5   | 2.5  | 1.4  | 9.4    | 15.1 | 4.9    | 2.9   | 5.7  |
| <b>AMB3R (ALL)</b>  | 1.8   | 1.8  | 2.1  | 4.7    | 2.4  | 1.6    | 3.7   | 2.1  |
| COLMAP (GT) [8]     | 25.3  | 25.7 | 22.6 | 25.8   | 25.0 | 21.9   | 24.7  | 24.4 |
| PSNR Pseudo GT [85] | 21.2  | 22.1 | 21.2 | 22.6   | 22.6 | 19.9   | 23.1  | 21.8 |
| <b>AMB3R</b>        | 20.6  | 22.3 | 20.2 | 22.0   | 23.6 | 21.8   | 22.2  | 21.8 |

Table 7. **VO: ATE RMSE [cm] on 7 scenes.** We show our online model surpasses the pseudo GT prior works used [83, 122] on ATE and achieve on-par novel view synthesis PSNR. COLMAP [8] GT is obtained via all sequences on each scene.

AMB3R not being specifically trained for dynamic scene reconstruction, it still shows consistent improvement over VGGT [124] and competitive reconstruction results compared to concurrent work  $\pi^3$ [129], which includes dynamic scenes at training. This lays a foundation for making a VO system possible in dynamic environments.

### 5.7. Uncalibrated Visual Odometry/SLAM

We present VO/SLAM results in Tab. 7- 11. Our model, Spann3R [122], and MUST3R [13], together establish a new class of *uncalibrated visual odometry* methods with unknown camera intrinsics. Both MUST3R and our approach outperform classic optimization-based VO, with our model further reducing the tracking error of previous SOTA from 7.1cm to 3.2cm on TUM and from 11.2cm to 2.6cm on ETH3D. Our method even surpasses pseudo GT used in prior work [83, 122] on 7-Scenes (Tab. 7), where we validate ground-truth reliability through novel-view synthesis.

We further compare against state-of-the-art SLAM sys-

|   |                  | fr1  |      |       |       |      |      |       |     | fr2  |       | fr3   | AVG  |
|---|------------------|------|------|-------|-------|------|------|-------|-----|------|-------|-------|------|
|   |                  | 360  | desk | desk2 | plant | room | rrpy | teddy | xyz | xyz  | desk  | long  |      |
| S | ORB-SLAM3 [82]   | X    | 2.0  | X     | 11.8  | X    | 5.6  | X     | 1.0 | 0.5  | 1.3   | 1.7   | X    |
|   | DSO [31]         | X    | 27.2 | 66.0  | 6.0   | 58.6 | X    | X     | 3.8 | 0.3  | 2.2   | 9.9   | X    |
|   | DPVO [67]        | 13.1 | 9.4  | 6.5   | 3.0   | 39.8 | 3.5  | 6.2   | 1.3 | 0.5  | 3.5   | 5.5   | 8.4  |
| D | TANDEM [56]      | X    | 4.3  | 33.7  | X     | X    | 4.9  | 43.1  | 2.4 | 0.3  | 2.0   | 8.3   | X    |
|   | MonoGS [77]      | 14.2 | 6.3  | 74.0  | 9.3   | 64.9 | 3.4  | 35.6  | 1.6 | 4.5  | 133.1 | 3.3   | 31.8 |
|   | DeepFactors [21] | 17.9 | 15.9 | 20.2  | 31.9  | 38.3 | 3.8  | 56.0  | 5.9 | 8.4  | 26.3  | 49.0  | 24.9 |
|   | DepthCov [25]    | 12.8 | 5.6  | 4.8   | 26.1  | 25.7 | 5.2  | 47.5  | 5.6 | 1.2  | 15.9  | 68.8  | 19.9 |
|   | DROID-VO [116]   | 15.7 | 5.2  | 11.1  | 6.0   | 33.4 | 3.2  | 19.1  | 5.6 | 10.7 | 7.9   | 7.3   | 11.4 |
|   | COMO [26]        | 12.9 | 4.9  | 9.5   | 13.8  | 27.0 | 4.8  | 24.5  | 4.0 | 0.7  | 6.3   | 10.5  | 10.8 |
|   | GORIE-VO[142]    | 13.1 | 4.0  | 8.6   | 4.1   | 32.7 | 2.9  | 14.5  | 1.2 | 0.2  | 16.1  | 4.8   | 9.3  |
| U | Spann3R [122]    | 20.7 | 16.1 | 28.3  | 57.4  | 84.8 | 6.1  | 92.4  | 2.1 | 4.4  | 20.7  | 193.9 | 47.9 |
|   | MUST3R [13]      | 8.9  | 5.1  | 7.1   | 5.4   | 13.4 | 5.2  | 6.9   | 2.7 | 1.7  | 15.6  | 4.3   | 7.1  |
|   | MUST3R* [13]     | 7.8  | 4.0  | 4.6   | 4.0   | 9.9  | 4.3  | 4.2   | 1.3 | 1.2  | 15.3  | 4.3   | 5.5  |
|   | AMB3R (ALL)      | 4.6  | 1.9  | 2.8   | 2.9   | 5.8  | 2.3  | 3.7   | 1.1 | 2.1  | 3.4   | 5.0   | 3.2  |
|   | AMB3R (KF)       | 3.9  | 1.7  | 2.3   | 2.7   | 5.5  | 2.2  | 2.8   | 0.8 | 0.6  | 3.0   | 4.0   | 2.7  |

Table 8. **VO: ATE RMSE [cm] on TUM RGB.** We compare with Sparse (S) versus dense (D) versus dense unconstrained (U) methods. The baseline methods are re-run without Loop Closure and global bundle adjustment. We compare with MUST3R [13] on all frames while keeping keyframe predictions as a reference.

|              |                    | 360  | desk | desk2 | floor | plant | room | rrpy | teddy | xyz | avg  |
|--------------|--------------------|------|------|-------|-------|-------|------|------|-------|-----|------|
|              |                    |      |      |       |       |       |      |      |       |     |      |
| Calibrated   | ORB-SLAM3 [82]     | X    | 1.7  | 21.0  | X     | 3.4   | X    | X    | X     | 0.9 | -    |
|              | DeepV2D [115]      | 24.3 | 16.6 | 37.9  | 165.3 | 20.3  | 24.6 | 10.5 | 31.6  | 6.4 | 37.5 |
|              | DeepFactors [21]   | 15.9 | 17.0 | 25.3  | 16.9  | 30.5  | 36.4 | 4.3  | 60.1  | 3.5 | 23.3 |
|              | DPV-SLAM [67]      | 11.2 | 1.8  | 2.9   | 5.7   | 2.1   | 33.0 | 3.0  | 8.4   | 1.0 | 7.6  |
|              | DPV-SLAM++ [67]    | 13.2 | 1.8  | 2.9   | 5.0   | 2.2   | 9.6  | 3.2  | 9.8   | 1.0 | 5.4  |
|              | GO-SLAM [149]      | 8.9  | 1.6  | 2.8   | 2.5   | 2.6   | 5.2  | 1.9  | 4.8   | 1.0 | 3.5  |
|              | DROID-SLAM [116]   | 11.1 | 1.8  | 4.2   | 2.1   | 1.6   | 4.9  | 2.6  | 4.8   | 1.2 | 3.8  |
|              | MAS3R-SLAM [83]    | 4.9  | 1.6  | 2.4   | 2.5   | 2.0   | 6.1  | 2.7  | 4.1   | 0.9 | 3.0  |
| Uncalibrated | DROID-SLAM [116]   | 20.2 | 3.2  | 9.1   | 6.4   | 4.5   | 91.8 | 5.6  | 4.5   | 1.2 | 15.8 |
|              | MAS3R-SLAM [83]    | 7.0  | 3.5  | 5.5   | 5.6   | 3.5   | 11.8 | 4.1  | 11.4  | 2.0 | 6.0  |
|              | VGGT-SLAM [74]     | 7.1  | 2.5  | 4.0   | 14.1  | 2.3   | 10.2 | 3.0  | 3.4   | 1.4 | 5.3  |
|              | <b>AMB3R (ALL)</b> | 4.6  | 1.9  | 2.8   | 3.2   | 2.9   | 5.8  | 2.3  | 3.7   | 1.1 | 3.1  |
|              | <b>AMB3R (KF)</b>  | 3.9  | 1.7  | 2.3   | 2.7   | 2.7   | 5.5  | 2.2  | 2.8   | 0.8 | 2.7  |

Table 9. **SLAM: ATE RMSE [cm] on TUM RGB.** We use image stride of 2 and keyframe poses evaluation as in MAS3R-SLAM [83] with all poses evaluation shown as a reference. Note that we did not do any post-processing on trajectories

tems in Tab. 9, where baseline methods are allowed to use loop closure and global bundle adjustment as post-processing after tracking the last frame. The impact of these post-processing steps can be seen by comparing Tab. 8 and Tab. 9. Following prior work [74, 83], we evaluate on raw keyframe poses for fair comparison. Our method shows for

|                    | cables1    | camshake1  | einstein1  | plant1     | plant2     | sofa1      | table3     | table7     | avg        |
|--------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Spann3R [122]      | 33.2       | 5.1        | 30.9       | 4.1        | 5.7        | 17.1       | 19.3       | 18.9       | 16.8       |
| MUS3R [13]         | 20.7       | 5.6        | 15.4       | 2.3        | 2.7        | 15.8       | 17.6       | 9.5        | 11.2       |
| MUS3R* [13]        | 20.7       | 5.3        | 11.2       | 1.8        | 2.7        | 15.5       | 17.3       | 5.5        | 10.0       |
| <b>AMB3R (ALL)</b> | <b>2.8</b> | <b>2.4</b> | <b>2.3</b> | <b>1.5</b> | <b>1.6</b> | <b>3.6</b> | <b>3.7</b> | <b>2.8</b> | <b>2.6</b> |
| AMB3R (KF)         | 2.3        | 2.1        | 1.9        | 0.5        | 0.9        | 3.1        | 2.6        | 2.4        | 2.0        |

Table 10. **VO: ATE RMSE [cm] on ETH3D SLAM [101]**. We use the same scene sets and image stride of 2 as in MUS3R [13]

|              | sit-xyz    | desk-p     | walk-ppy   | walk-s     | walk-h     | sit-s      | sit-h      | walk-x     | sit-r      | avg        |
|--------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| MUS3R [13]   | 3.5        | 3.3        | 6.7        | 2.2        | 5.5        | 1.5        | 5.3        | 7.8        | 4.9        | 4.5        |
| MUS3R* [13]  | 2.4        | 2.2        | 5.0        | 1.9        | 3.9        | 1.3        | 4.6        | 5.4        | 4.0        | 3.4        |
| MegaSaM [62] | 1.1        | OOM        | 3.3        | 0.6        | 1.8        | 0.6        | 2.5        | 1.4        | 2.5        | -          |
| <b>AMB3R</b> | <b>1.3</b> | <b>2.1</b> | <b>4.2</b> | <b>0.9</b> | <b>1.9</b> | <b>0.8</b> | <b>2.2</b> | <b>2.0</b> | <b>2.0</b> | <b>1.9</b> |

Table 11. **VO: ATE RMSE on TUM Dynamic**. MUS3R\* uses re-rendering with trajectory smoothness. MegaSaM uses keyframe & global BA at the end followed by a trajectory smoother.

| Scenes        | COLMAP [99] |       | ACE-Zero [10] |       | VGGSfM [124] |       | DF-SfM [38] |       | MAS3R-SfM |       | AMB3R (SfM) |       |
|---------------|-------------|-------|---------------|-------|--------------|-------|-------------|-------|-----------|-------|-------------|-------|
|               | RRA@5       | RTA@5 | RRA@5         | RTA@5 | RRA@5        | RTA@5 | RRA@5       | RTA@5 | RRA@5     | RTA@5 | RRA@5       | RTA@5 |
| courtyard     | 56.3        | 60.0  | 4.0           | 1.9   | 50.5         | 51.2  | 80.7        | 74.8  | 89.8      | 64.4  | 100.0       | 96.5  |
| delivery area | 34.0        | 28.1  | 27.4          | 1.9   | 22.0         | 19.6  | 82.5        | 82.0  | 83.1      | 81.8  | 91.0        | 76.6  |
| electro       | 53.3        | 48.5  | 16.9          | 7.9   | 79.9         | 58.6  | 82.8        | 81.2  | 100.0     | 95.5  | 95.6        | 81.2  |
| facade        | 92.2        | 90.0  | 74.5          | 64.1  | 57.5         | 48.7  | 80.9        | 82.6  | 74.3      | 75.3  | 100.0       | 95.4  |
| kicker        | 87.3        | 86.2  | 26.2          | 16.8  | 100.0        | 97.8  | 93.5        | 91.0  | 100.0     | 100.0 | 100.0       | 99.2  |
| meadow        | 0.9         | 0.9   | 3.8           | 0.9   | 100.0        | 96.2  | 56.2        | 58.1  | 58.1      | 58.1  | 100.0       | 95.2  |
| office        | 36.9        | 32.3  | 0.9           | 0.0   | 64.9         | 42.1  | 71.1        | 54.5  | 100.0     | 98.5  | 100.0       | 53.9  |
| pipes         | 30.8        | 28.6  | 9.9           | 1.1   | 100.0        | 97.8  | 72.5        | 61.5  | 100.0     | 100.0 | 100.0       | 87.9  |
| playground    | 17.2        | 18.1  | 3.8           | 2.6   | 37.3         | 40.8  | 70.5        | 70.1  | 100.0     | 93.6  | 98.7        | 62.2  |
| relief        | 16.8        | 16.8  | 16.8          | 17.0  | 59.6         | 57.9  | 32.9        | 32.9  | 34.2      | 40.2  | 100.0       | 90.1  |
| relief 2      | 11.8        | 11.8  | 7.3           | 5.6   | 69.9         | 70.3  | 40.9        | 39.1  | 57.4      | 76.1  | 100.0       | 75.7  |
| terrace       | 100.0       | 100.0 | 5.5           | 2.0   | 38.7         | 29.6  | 100.0       | 99.6  | 100.0     | 100.0 | 100.0       | 97.2  |
| terrains      | 100.0       | 99.5  | 15.8          | 4.5   | 70.4         | 54.9  | 100.0       | 91.9  | 58.2      | 52.5  | 91.6        | 53.8  |
| Average       | 49.0        | 47.8  | 16.4          | 9.7   | 65.4         | 58.9  | 74.2        | 70.7  | 81.2      | 79.7  | 98.2        | 81.9  |

Table 12. **SfM on ETH3D [100]**. Our method does not need optimization-based BA, unlike all baseline methods.

the first time an uncalibrated feed-forward approach outperforming optimization-based methods (with calibration and post-processing) on the TUM [108] dataset.

Compared to hybrid systems combining dense priors with optimization-based backends, such as MAS3R-SLAM [83] and VGGT-SLAM [74], our model achieves superior performance in the uncalibrated setting, and even surpasses MAS3R-SLAM with known calibration.

Finally, we evaluate on dynamic environments (Tab. 11), where our model again surpasses the previous SOTA uncalibrated VO, MUS3R [13]. Neither model is trained for dynamic reconstruction, yet our raw predictions remain comparable to MegaSaM [62], a method explicitly designed for dynamic scenes and equipped with global bundle adjustment and trajectory smoothing as post-processing steps. Please refer to suppl. material/video for qualitative results.

## 5.8. Structure from Motion

We report SfM results in Tab. 12-13. Unlike all baselines, our method operates in a feed-forward manner without optimization-based BA. Despite this, it achieves substantial improvements in rotation accuracy over MAS3R-SfM [28] on ETH3D [100], an unordered image collection. On Tanks&Temples [55, 97] with video input, our method further surpass ACE-Zero [10] by a large margin. Please refer to suppl. material/video for extensive qualitative results.

| Method             | Training (7 scenes) |             | Intermediate (8 scenes) |             | Advanced (6 scenes) |             |
|--------------------|---------------------|-------------|-------------------------|-------------|---------------------|-------------|
|                    | RRA@5               | RTA@5       | RRA@5                   | RTA@5       | RRA@5               | RTA@5       |
| ACE-Zero [10]      | 73.9                | 72.9        | 67.6                    | 74.0        | 22.9                | 19.1        |
| MAS3R-SfM [28]     | 56.2                | 64.9        | 50.8                    | 57.5        | 38.8                | 36.5        |
| <b>AMB3R (SfM)</b> | <b>95.0</b>         | <b>94.5</b> | <b>98.7</b>             | <b>96.9</b> | <b>68.0</b>         | <b>72.4</b> |

Table 13. **SfM on Tanks&Temples [55]**. Each scene contains 151 to 1106 images. We only retain methods succeed on every scene.

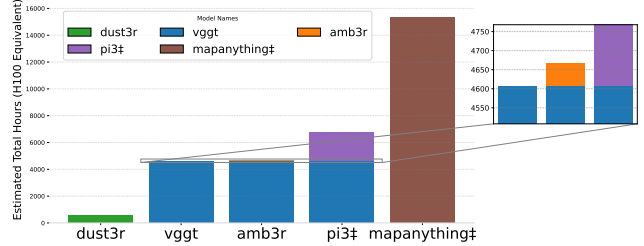


Figure 4. **Training cost comparison**. We roughly estimate the training cost of each model. ‡ indicates concurrent works.

| Method       | ETH3D       |             |             | DTU         |             |             | 7-Scenes    |             |             |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|              | rel.↓       | Acc.↓       | Cp.↓        | rel.↓       | Acc.↓       | Cp.↓        | rel.↓       | Acc.↓       | Cp.↓        |
| w/o backend  | 6.02        | 12.81       | 11.89       | 0.83        | <b>0.22</b> | 0.08        | 5.51        | 2.32        | 3.51        |
| w 2D backend | 5.32        | 11.78       | 12.78       | 0.92        | 0.23        | 0.19        | 5.15        | 1.92        | 3.10        |
| Full         | <b>4.64</b> | <b>9.98</b> | <b>9.69</b> | <b>0.81</b> | 0.22        | <b>0.08</b> | <b>4.74</b> | <b>1.74</b> | <b>2.84</b> |

Table 14. **Ablation study**. We ablate on our 3D backend versus 2D backend implemented by alternating attention.

## 6. Discussion

### 6.1. Analysis

**Training cost.** We show the estimated training cost comparison in Fig. 4, where 1 H100 hour counts as 3 A100 hours. Since  $\pi 3$  did not report their overall training time, we roughly estimate their training cost based the number of training epochs and GPUs used for training. The training of our model requires around 80 H100 hours. Compared to  $\pi 3$  [129] and MapAnything [129], our model requires significantly less add-on cost on top of VGGT [124].

**Ablation studies.** Tab. 14 compares our compact 3D backend with a 2D variant using alternating attention. Our 3D backend consistently outperforms the 2D one, highlighting the advantage of maintaining a sparse yet compact 3D scene representation for explicit geometric reasoning.

### 6.2. Conclusion

We presented AMB3R, a feed-forward model that integrates a sparse yet compact volumetric scene representation as its backend. Our paper shows this spatial compactness leads to state-of-the-art performance across diverse 3D vision tasks—including monocular/multi-view metric-scale depth/3D reconstruction, uncalibrated visual odometry and structure-from-motion. Notably, all these results are achieved without test time optimization or task-specific fine-tuning. This shows a step towards a scalable, unified, and generalizable feed-forward 3D perception system.



## Supplementary Material

### 7. Limitations

#### 7.1. Scene Representations

**Local scene representation.** Our backend representation is a local scene representation that cannot perform joint reasoning across multiple chunks predicted by the front-end. Extending it to a global representation can be a promising direction as it would enable long-term consistency across chunks. However, training such a global backend would also require substantially more computational resources.

**Computation complexity.** Our model shares the inherent limitation of type-(c) methods regarding the quadratic computational complexity with respect to the number of input images. However, a key advantage of our sparse yet compact 3D representation is that its complexity depends on the amount of 3D content rather than the number of views. This is, in fact, one of our motivations to introduce a compact 3D scene representation as a backend. Due to the computational constraint, we are not able to investigate on how to re-design the entire pointmap-based foundation models. However, our work serves as a proof-of-concept demonstrating the effectiveness of the spatial compactness. We believe this opens a promising direction towards making pointmap-based foundation models themselves more scalable to longer sequences. One possible direction is to reduce the usage of global attention and leverage the compact 3D scene representation as an alternative to reduce the computational complexity for long sequences.

**Dynamic scenes.** Since our model has not been trained for dynamic scenarios, it heavily relies on static cues for dynamic environment. Thus, it might fail when the target scene is dominated by dynamic objects. This can be solved by including dynamic scenes as training datasets.

#### 7.2. Visual Odometry

**Reliance on dense reconstruction prior.** Our visual odometry relies on the prior that the model predicts geometry in reference (first) frame’s coordinate system, up to an unknown scale factor. When this assumption breaks, typically in scenes with diverse depth ranges and complex thin structures, the scale alignment becomes unreliable, often leading to tracking failure. Similarly, scenes dominated by distant content can increase prediction variance, resulting in poor scale alignment and trajectory accuracy. In this case, type-(b) method might benefit from its persistent memory and a generally longer perception window.

**Drift/Kidnapping issue.** Our system is a visual odometry without explicit loop closure or relocalization. Consequently, tracking might drift in large-scale environments

or fail under long-term kidnapping scenarios. In this case, an optimization-based module could help for loop closure or relocalization. Notably, our backward search strategy in keyframe selection could also mitigate this issue. Once loop is closed, as long as the accumulative pose error is less than the backward search pose distance, our model can still leverage the earlier frame to bring the system back on track. This is useful for an online system, as it is not necessary to fix the earlier trajectory in that case.

#### 7.3. Structure from Motion

**Reliance on dense reconstruction prior.** Our SfM shares the same limitations as in VO as they rely on the same prior.

**Initialization.** Our initialization is purely based on the feature-based image clustering and the confidence of the local chunk. We notice that our confidence usually prefers indoor regions. In some photo-tourism scenarios, our model might initialize with indoor images that do not guarantee overlap with the outdoor regions that are of primary interest. One possible solution is to use explicit feature matching to construct the view graph at cost of extra complexity.

**Mapping window selection.** Our mapping window selection for global mapping is based on pose distance. This is usually sufficient for visual odometry due to the local smoothness of the trajectory. However, for a large-scale structure from motion system. The same geometry is likely to be observed from diverse views. Selecting mapping frames using pose distance can therefore omit overlapping views with wide baselines. In that case, a geometry-based view selection might help at the cost of extra complexity.

**Non-overlap reconstruction.** The ability to reconstruct non-overlapping views is useful for small-scale problems where overlap genuinely does not exist. However, this could become problematic at larger scales: image clustering may group visually similar images from entirely different locations into the same cluster despite having no geometric overlap. In such cases, the model may hallucinate geometry for these non-overlapping images and give them moderate confidence, preventing them from being pruned and leading to irreversible reconstruction errors. One possible remedy is to filter out false clusters (e.g., Doppelgangers [14]).

**Pose consistency.** The final camera poses are weighted average feed-forward poses. Compared to COLMAP [99] poses obtained via BA, our poses may not strictly follow the geometry constraint. For downstream tasks like novel view synthesis, which requires strict pose coherence, our poses, even metrically better in some cases, might still result in lower PSNR. This is the inherent limitation of feed-forward poses and can be addressed via BA as post-processing.

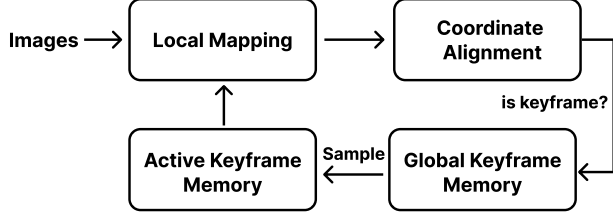


Figure 5. **Overview of AMB3R (VO).** Input frames are mapped with the keyframes in the active keyframe memory to predict geometry and camera poses. After coordinate alignment, we select new keyframes and update the global keyframe memory; poses and geometry for non-keyframes are also stored. If the active keyframe memory is not full, the new keyframe is appended; otherwise, we refresh the active keyframe memory by resampling a new set of keyframes from the global keyframe memory.

## 8. Additional Details

### 8.1. Metric-scale Head

To predict metric scale factor, we first map encoder feature into one feature vector via a three-layer MLP. We then add this feature with depth feature from depth DPT [94] branch of VGGT [124], and map this new feature into a metric scale factor via a two-layer convolution. We supervise this prediction using an  $\mathcal{L}_1$  loss. In cases where the selected pixels contain missing depth values, which frequently occurs in the Waymo [110] dataset, we use a ROE solver to estimate the relative scale difference. This allows us to recover a consistent ground-truth log-depth value for supervision even when raw depth is unavailable.

### 8.2. Training Datasets

We use ScanNet [22], ScannetPP [139], WildRGBD [134], Mapfree [4], Aria [89], Waymo [110], Virtual Kitti2 [12], GTASfM [125], MVS-Synth [43], OminiObject3d [130], and Hypersim [98] to train our model. Due to the constraint of the data storage, we only store a subset of each dataset. Although we only sample 2k samples in total for each epoch, we find data diversity is significantly more important than the data amount for training pointmap-based foundation model. We exclude Co3D [95] as we observe that VGGT might overfit on certain patterns on Co3D.

### 8.3. Visual Odometry Pipeline

Fig. 5 illustrates our visual odometry pipeline. Input frames are mapped with keyframes stored in the active keyframe memory to predict camera poses and geometry. The coordinate alignment is done via 1) transforming the active keyframe map from global to local coordinates 2) estimating the relative scale of the corresponding keyframe geometry 3) transforming the local map to global coordinates via the weighted average of relative poses of each corre-

sponding keyframe. We then select new keyframes from the newly mapped frames, and update the global keyframe memory. If the number of keyframes in active keyframe memory has not reached its capacity, we append the new keyframe; otherwise, we refresh the entire active keyframe memory by resampling from the global keyframe memory.

### 8.4. Structure from Motion Pipeline

We show the pipeline of our SfM as in Fig. 6. Our SfM mainly consists of 3 stages: 1) Image clustering that groups images into small clusters, 2) Coarse registration that registers each cluster incrementally, and 3) Global mapping that refines keyframe and non-keyframes via mapping.

### 8.5. Visual Odometry Runtime Analysis

We evaluate the runtime of our visual odometry on the TUM dataset [108]. Note that we exclude the data-loading overhead. Our method runs at an average of 4.2 FPS, with a best case of 6.0 FPS and a worst case of 3.4 FPS on an NVIDIA RTX 4090 GPU with input resolution of (392, 518). The variation in speed is primarily caused by the fluctuating number of active keyframes and the scene difficulty (The backend can be skipped if the front-end confidence is sufficiently high). Importantly, since we cap the maximum number of active keyframes at 10, **the computational complexity does not grow with respect to number of frames**, unlike type-(b) methods that do not have memory pruning.

Depending on the computation budget, one might reduce the input resolution or number of active keyframes for better runtime at the cost of accuracy (See Fig. 7).

### 8.6. Imperfect Ground-truth on 7 Scenes

We find that the pseudo ground truth in the original 7-Scenes [105] dataset often contains noticeable drift in certain sequences. This is because those poses are obtained via ICP-based KinectFusion [85]. ICP is prone to have rotation drift in scenes with near-spherical geometry and translation drift in scenes with many flat surfaces. 7 scenes dataset is the latter case. For instance, in Pumpkin/seq01, we find a substantial drift toward the end of the sequence. As a result, it is fundamentally impossible to achieve accurate tracking when evaluated against this pseudo GT unless the model happens to drift in exactly the same way. This makes the original pseudo GT unsuitable for fair comparison.

To address this, we adopt COLMAP GT [8] that is obtained via global optimization across all sequences of each scene. To ensure its reliability, we follow ACE-Zero [10] and evaluate novel view synthesis using NeRFstudio [112]. As shown in Tab. 7, the PSNR obtained from rendering with the new GT is consistently higher, confirming its accuracy. We therefore use it as the new ground-truth for evaluating 3D reconstruction and visual odometry.

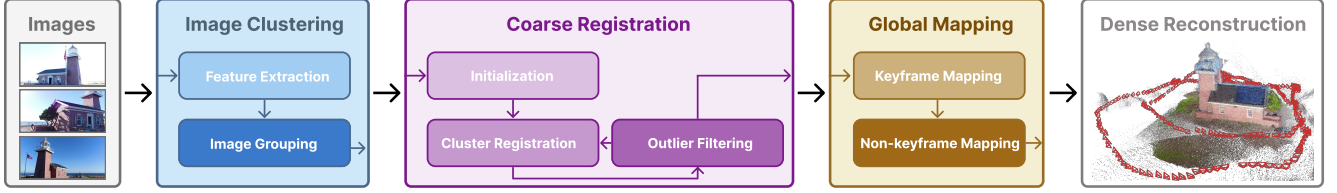


Figure 6. **Overview of AMB3R (SfM).** Our SfM pipeline contains 1) Image clustering that groups images into small clusters, 2) Coarse registration that constructs an initial coarse reconstruction, and 3) Global mapping that performs keyframe and non-keyframe refinement.

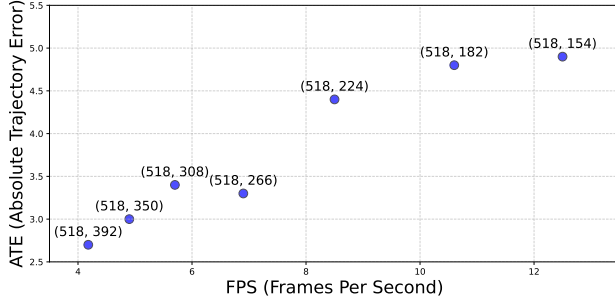


Figure 7. **VO runtime with respect to resolution.** By reducing input resolution, our method can achieve over 10FPS inference while being more accurate compared to other uncalibrated VO.

## 8.7. Baselines and Evaluation Datasets

**Monocular depth estimation.** We compare against Ominidata [29], Depth Anything v2 [137], Marigold [51], Diffusion-E2E [35], and MoGe [127], which are explicitly trained for monocular depth estimation. In contrast, VGGT [124] and our model are trained with a multi-view objective, and evaluated on monocular depth in a zero-shot manner. We use NYUv2 [84], KITTI [36], ETH3D [100], ScanNet [22], and DIODE [119] for evaluation following Marigold [51]. Note that for DIODE, there is a known issue about floaters in ground-truth geometry. Existing works like depth anything v2 [137] and MoGe [127] evaluate on DIODE with a pre-processing script that excludes those floaters. Since these scripts are not publicly available, we evaluate on the original noisy ground-truth following Marigold and Diffusion-E2E.

**Multi-view depth estimation.** We compare with 5 different categories of methods here: a) classic approach: COLMAP [99], which is an integrated solution of SfM and MVS based on optimization, b) monocular depth estimation methods that estimate depth from a single view: Depth Pro [6], Metric3D [140], UniDepthV2 [92], and Depth Anything v2 [137], c) Multi-view depth estimation methods that requires known calibration, camera poses, and per-image range: MVSNet [138], Vis-MVSNet [144], PatchmatchNet [121], and MVSFormer++ [15]. d) Multi-view depth estimation methods without the need for per-image range: Fast-MVSNet [141], Robust MVD baseline [102], and MVSA [45]. and e) Depth estimation from



Figure 8. **Examples of buildings in DTU [2] dataset.**

raw images without any prior information required: DeMoN [118], DUST3R [128], Spann3R [122], Pow3R [47], MUST3R [13], VGGT [124], as well as concurrent works  $\pi 3$  [129], and MapAnything [52]. We evaluate those methods on RMVDB [102], using KITTI [36], ETH3D [100], ScanNet [22], DTU [2], and Tanks & Temples [55] datasets. **Metric-scale estimation.** We compare with existing pointmap-based foundation models which can recover metric-scale factors: MAST3R [58], Spann3R [122], MUST3R [13], CUT3R [126], and concurrent work MapAnything [52] on RMVDB [102]. Note that DTU datasets contains many buildings placed on a pure white table as in Fig. 8. Due to the lack of background environment for scale reasoning, existing models would usually predict those buildings as real buildings, resulting in high absolute relative errors. In that case, inlier ratio could be a good measure of the metri-scale prediction.

**3D reconstruction.** We compare with existing representative pointmap-based foundation models: Spann3R [122], MUST3R [13], CUT3R [126], VGGT [124], as well as concurrent works  $\pi 3$  [129], and MapAnything [52] on ETH3D [100], DTU [2], and 7 scenes [105]. We use image tuples from RMVDB [102] for ETH3D and DTU. For 7 scenes, we use Spann3R [122] split with improved GT.

**Video depth estimation.** We compare with pointmap-based models: Spann3R [122], CUT3R [126], VGGT [124], and  $\pi 3$  [129] on dynamic video depth estimation task on Sintel [11], Bonn [87], and Kitti [36] datasets. Among those methods, CUT3R [126] and  $\pi 3$  [129] are trained on dynamic datasets while our model is not specifically trained on dynamic data.

**VO and SLAM.** For visual odometry, we compare with 1) Sparse VO: ORB-SLAM3 [82], DSO [31], and DPVO [67], 2) Dense VO: TANDEM [56], MonoGS [77], DeepFactors [21], DepthCov [25], DROID-VO [116], COMO [26], and GLORIE-VO [142], and 3) Uncalibrated VO: Spann3R [122] and MUST3R [13]. In addition to visual



| Method          | KITTI      |             | ScanNet      |               | ETH3D      |             | DTU        |             | T&T        |             | Avg        |             |
|-----------------|------------|-------------|--------------|---------------|------------|-------------|------------|-------------|------------|-------------|------------|-------------|
|                 | rel ↓      | δ ↑         | rel ↓        | δ ↑           | rel ↓      | δ ↑         | rel ↓      | δ ↑         | rel ↓      | δ ↑         | rel ↓      | δ ↑         |
| w/o backend     | 4.5        | 60.4        | (2.3)        | (80.8)        | 1.8        | 85.3        | 1.0        | 94.8        | 2.0        | 83.9        | 2.3        | 80.6        |
| w/ 2d backend   | 2.9        | 73.9        | (2.0)        | (84.2)        | 1.4        | 90.2        | 1.1        | 94.4        | 1.8        | 88.4        | 1.8        | 86.2        |
| w/o scale align | 3.0        | 72.0        | (2.0)        | (84.8)        | 1.5        | 89.7        | <b>0.9</b> | <b>95.3</b> | 1.9        | 89.0        | 1.9        | 86.2        |
| w/o zero conv   | 26.9       | 7.8         | (17.5)       | (15.5)        | 19.8       | 17.8        | 7.3        | 34.6        | 16.2       | 23.9        | 17.5       | 19.9        |
| Full            | <b>2.8</b> | <b>74.4</b> | <b>(1.9)</b> | <b>(85.8)</b> | <b>1.4</b> | <b>90.9</b> | 0.9        | 95.1        | <b>1.7</b> | <b>90.2</b> | <b>1.7</b> | <b>87.3</b> |

Table 15. **Ablation studies.** We ablate design choices of backend, scale alignment for supervision, and zero convolution on RMVDB [102].

odometry baselines, we also consider SLAM baselines with global bundle adjustment and loop closure: 1) SLAM with calibration: ORB-SLAM3 [82], DeepV2D [115], DeepFactors [21], DPV-SLAM [67], GO-SLAM [149], DROID-SLAM [116], MAST3R-SLAM [83] and 2) SLAM without calibration: DROID-SLAM [116], MAST3R-SLAM [83], VGGT-SLAM [124]. We compare with those methods using common SLAM benchmarks, including TUM [108], ETH-SLAM [101], and 7scenes [105]. We also compare with MegaSaM [62], a structure-and-motion methods specifically designed for dynamic environment, on TUM Dynamic [108] dataset to test the generalization on dynamic scenes.

**Structure from Motion.** We compare with optimization-based SfM methods: COLMAP [99], ACE-Zero [10], FlowMAP [106], VGGsFm [124], DF-SfM [38], and MAST3R-SfM [28] on ETH3D [100] and Tanks&Temples [55] dataset. ETH3D [100] contains unordered image collection while Tanks&Temples [55] contains images from video.

## 9. Additional Quantitative Results

**Ablation study.** We present additional ablation studies in Tab. 15 to analyze the impact of key design choices, including the backend (2D vs. 3D), scale alignment for supervision, and zero convolution. Notably, training without zero convolution fails to converge under our current computational budget and dataset size. As discussed in Sec. 3, this is caused by catastrophic forgetting of the learned confidence: without zero convolution, the confidence function with randomly initialized weights of the backend will shift drastically, and lead to inconsistent learning objectives. In this case, convergence would likely require training resources and data comparable to those used for VGGT [124].

**Multi-view depth estimation.** We show the full table of multi-view depth estimation results in Tab. 16 with detailed input modality and the alignment.

**Structure from motion.** We show per-scene decomposition of SfM on Tanks&Temples [55] in Tab. 17. The COLMAP results [97] are used as the ground-truth.

## 10. Qualitative examples

**Visual Odometry.** As in Fig. 9 and Fig. 10, we present qualitative results of our visual odometry on all static scenes from the 7Scenes [105], TUM [108], and ETH SLAM [101] datasets used in our evaluation.

**Structure from Motion.** Fig. 11 and Fig. 12 show qualitative results of SfM on all scenes from the ETH3D [100] and Tanks&Temples [55] datasets. In Fig. 13, we further include qualitative results on the Cambridge Landmarks dataset, in-the-wild image collections, and the IMC Phototourism [49] dataset, which contains large-scale unordered images captured by tourists.

| Approach                                             | GT    | GT    | GT        | Align   | KITTI      |                   | ScanNet      |                   | ETH3D      |                   | DTU        |                   | T&T        |                   | Average    |                   |
|------------------------------------------------------|-------|-------|-----------|---------|------------|-------------------|--------------|-------------------|------------|-------------------|------------|-------------------|------------|-------------------|------------|-------------------|
|                                                      | Poses | Range | Intrinsic |         | rel ↓      | $\delta_{1.03}$ ↑ | rel ↓        | $\delta_{1.03}$ ↑ | rel ↓      | $\delta_{1.03}$ ↑ | rel ↓      | $\delta_{1.03}$ ↑ | rel ↓      | $\delta_{1.03}$ ↑ | rel ↓      | $\delta_{1.03}$ ↑ |
| a) Classic approaches                                |       |       |           |         |            |                   |              |                   |            |                   |            |                   |            |                   |            |                   |
| COLMAP [99]                                          | ✗     | ✗     | ✗         | ✗       | 12.0       | 58.2              | 14.6         | 34.2              | 16.4       | 55.1              | <b>0.7</b> | <b>96.5</b>       | 2.7        | <b>95.0</b>       | 9.3        | 67.8              |
| COLMAP Dense [99]                                    | ✗     | ✗     | ✗         | ✗       | 26.9       | 52.7              | 38.0         | 22.5              | 89.8       | 23.2              | 20.8       | 69.3              | 25.7       | 76.4              | 40.2       | 48.8              |
| b) Single-view depth                                 |       |       |           |         |            |                   |              |                   |            |                   |            |                   |            |                   |            |                   |
| Depth Pro [6]                                        | ✗     | ✗     | ✓         | med     | 6.1        | 39.6              | (4.3)        | (58.4)            | 6.1        | 53.5              | 5.6        | 49.6              | 5.6        | 57.5              | 5.6        | 51.7              |
| Metric3D [40]                                        | ✗     | ✗     | ✓         | med     | 5.1        | 44.1              | 2.4          | 78.3              | 4.4        | 54.5              | 10.1       | 39.5              | 6.2        | 48.0              | 5.6        | 52.9              |
| UniDepthV2 [92]                                      | ✗     | ✗     | ✓         | med     | 4.0        | 55.3              | (2.1)        | (82.6)            | 3.7        | 66.2              | 3.2        | 72.3              | 3.6        | 68.4              | 3.3        | 68.9              |
| DepthAnything V2 [137]                               | ✗     | ✗     | ✗         | lstsq † | 6.6        | 38.6              | 4.0          | 58.6              | 4.7        | 56.5              | 2.6        | 74.7              | 4.5        | 57.5              | 4.8        | 54.1              |
| c) Depth from frames and poses (w/ per-image range)  |       |       |           |         |            |                   |              |                   |            |                   |            |                   |            |                   |            |                   |
| MVSNet [138]                                         | ✓     | ✓     | ✓         | ✗       | 22.7       | 36.1              | 24.6         | 20.4              | 35.4       | 31.4              | (1.8)      | (86.0)            | 8.3        | 73.0              | 18.6       | 49.4              |
| Vis-MVSNet [144]                                     | ✓     | ✓     | ✓         | ✗       | 9.5        | 55.4              | 8.9          | 33.5              | 10.8       | 43.3              | (1.8)      | (87.4)            | 4.1        | 87.2              | 7.0        | 61.4              |
| PatchmatchNet [121]                                  | ✓     | ✓     | ✓         | ✗       | 10.8       | 45.8              | 8.5          | 35.3              | 19.1       | 34.8              | (2.1)      | (82.8)            | 4.8        | 82.9              | 9.1        | 56.3              |
| MVSFormer++ [15]                                     | ✓     | ✓     | ✓         | ✗       | 4.4        | 65.7              | 7.9          | 39.4              | 7.8        | 50.4              | (0.9)      | (95.3)            | 3.2        | 88.1              | 4.8        | 67.8              |
| d) Depth from frames and poses (w/o per-image range) |       |       |           |         |            |                   |              |                   |            |                   |            |                   |            |                   |            |                   |
| Fast-MVSNet [141]                                    | ✓     | ✗     | ✓         | ✗       | 12.1       | 37.4              | 287.1        | 9.4               | 131.2      | 9.6               | (540.4)    | (1.9)             | 33.9       | 47.2              | 200.9      | 21.1              |
| Robust MVD Baseline [102]                            | ✓     | ✗     | ✓         | ✗       | 7.1        | 41.9              | 7.4          | 38.4              | 9.0        | 42.6              | 2.7        | 82.0              | 5.0        | 75.1              | 6.3        | 56.0              |
| MVSA [45]                                            | ✓     | ✗     | ✓         | ✗       | 3.2        | 68.8              | 3.7          | 62.9              | 3.2        | 68.0              | 1.3        | 95.0              | 2.1        | <u>90.5</u>       | 2.7        | 77.0              |
| e) Depth from frames (w/o poses)                     |       |       |           |         |            |                   |              |                   |            |                   |            |                   |            |                   |            |                   |
| DeMoN [118]                                          | ✗     | ✗     | ✓         | $\ t\ $ | 15.5       | 15.2              | 12.0         | 21.0              | 17.4       | 15.4              | 21.8       | 16.6              | 13.0       | 23.2              | 16.0       | 18.3              |
| DUST3R [128]                                         | ✗     | ✗     | ✗         | med     | 5.4        | 49.5              | (3.1)        | (71.8)            | 3.0        | 76.0              | 3.9        | 68.6              | 3.3        | 75.1              | 3.7        | 68.2              |
| Spann3R [122]                                        | ✗     | ✗     | ✗         | med     | 7.9        | 36.2              | (3.3)        | (67.1)            | 5.7        | 58.6              | 3.5        | 65.2              | 4.7        | 58.5              | 5.0        | 57.1              |
| Pow3R [47]                                           | ✗     | ✗     | ✗         | med     | 5.7        | 45.7              | (3.2)        | (68.8)            | 3.0        | 74.7              | 3.0        | 74.3              | 3.3        | 76.6              | 3.6        | 68.0              |
| MUST3R [13]                                          | ✗     | ✗     | ✗         | med     | 4.5        | 55.0              | (4.0)        | (59.8)            | 2.5        | 80.3              | 4.6        | 55.4              | (2.6)      | (80.4)            | 3.7        | 66.2              |
| VGGT [124]                                           | ✗     | ✗     | ✗         | med     | 4.5        | 59.6              | (2.3)        | (80.8)            | 1.8        | 86.3              | <u>0.9</u> | <u>95.6</u>       | 2.4        | 84.1              | 2.4        | 81.3              |
| $\pi 3^\ddagger$ [129]                               | ✗     | ✗     | ✗         | med     | 2.8        | 72.9              | (2.0)        | (83.6)            | <b>1.3</b> | <b>92.4</b>       | 1.3        | 91.8              | 1.8        | 87.3              | 1.8        | 85.6              |
| MapAnything <sup>‡</sup> [124]                       | ✗     | ✗     | ✗         | med     | 4.0        | 59.4              | 4.0          | 60.5              | 2.8        | 73.2              | 3.9        | 63.7              | 3.3        | 73.0              | 3.6        | 66.0              |
| <b>AMB3R</b>                                         | ✗     | ✗     | ✗         | med     | <b>2.8</b> | <b>74.4</b>       | <b>(1.9)</b> | <b>(85.8)</b>     | 1.4        | 90.9              | 0.9        | 95.1              | <b>1.7</b> | 90.2              | <b>1.7</b> | <b>87.3</b>       |

Table 16. **Multi-view depth estimation.** Our method achieves state-of-the-art performance on RMVDB [102].  $\ddagger$  means concurrent works, and (parentheses) indicate. We only report a subset of a)-e) methods, and please refer to supplementary material for the full table.

| Scenes       |             | COLMAP [99] |       | ACE-Zero [10] |       | FlowMap [106] |       | VGGSfM [124] |       | DF-SfM [38] |       | MASt3R-SfM [28] |       | AMB3R (SfM) |       |
|--------------|-------------|-------------|-------|---------------|-------|---------------|-------|--------------|-------|-------------|-------|-----------------|-------|-------------|-------|
|              |             | RRA@5       | RTA@5 | RRA@5         | RTA@5 | RRA@5         | RTA@5 | RRA@5        | RTA@5 | RRA@5       | RTA@5 | RRA@5           | RTA@5 | RRA@5       | RTA@5 |
| Training     | Barn        | GT          | GT    | 56.1          | 55.6  | -             | -     | -            | -     | 100.        | 99.8  | 52.6            | 85.6  | 91.8        | 96.4  |
|              | Caterpillar | GT          | GT    | 87.3          | 95.6  | -             | -     | -            | -     | -           | -     | 84.2            | 92.3  | 100.0       | 98.6  |
|              | Church      | GT          | GT    | 90.5          | 76.3  | -             | -     | -            | -     | -           | -     | 11.6            | 16.8  | 93.0        | 92.9  |
|              | Courthouse  | GT          | GT    | 44.1          | 45.0  | -             | -     | -            | -     | -           | -     | 8.8             | 9.9   | 79.9        | 77.7  |
|              | Ignatius    | GT          | GT    | 100.          | 99.9  | 62.5          | 70.0  | -            | -     | 100.        | 99.9  | 43.6            | 60.1  | 100.0       | 99.5  |
|              | Meetingroom | GT          | GT    | 39.3          | 38.5  | 26.3          | 39.8  | -            | -     | 84.1        | 89.0  | 92.6            | 89.9  | 100.0       | 96.3  |
|              | Truck       | GT          | GT    | 100.          | 99.7  | 53.4          | 69.6  | -            | -     | 100.        | 99.8  | 100.            | 99.7  | 100.0       | 99.7  |
| Intermediate | Family      | GT          | GT    | 38.9          | 44.6  | -             | -     | -            | -     | -           | -     | 22.3            | 25.9  | 100.0       | 99.6  |
|              | Francis     | GT          | GT    | 57.4          | 79.0  | 57.6          | 67.7  | -            | -     | 100.        | 99.7  | 17.0            | 41.0  | 100.0       | 92.1  |
|              | Horse       | GT          | GT    | 68.2          | 81.8  | -             | -     | -            | -     | -           | -     | 6.4             | 6.3   | 100.0       | 98.8  |
|              | Lighthouse  | GT          | GT    | 30.6          | 38.8  | 4.8           | 9.5   | -            | -     | 66.3        | 66.0  | 50.8            | 72.1  | 100.0       | 98.7  |
|              | M60         | GT          | GT    | 100.          | 99.9  | 50.4          | 48.3  | -            | -     | 100.        | 99.8  | 100.            | 100.  | 100.0       | 99.0  |
|              | Panther     | GT          | GT    | 100.          | 99.5  | 100.          | 77.6  | -            | -     | 100.        | 99.1  | 100.            | 99.5  | 100.0       | 99.6  |
|              | Playground  | GT          | GT    | 82.7          | 85.5  | 49.1          | 63.8  | -            | -     | 100.        | 99.9  | 99.3            | 99.3  | 100.0       | 98.3  |
|              | Train       | GT          | GT    | 62.6          | 62.5  | 18.4          | 29.2  | -            | -     | 42.8        | 41.8  | 10.6            | 15.8  | 89.4        | 88.8  |
| Advanced     | Auditorium  | GT          | GT    | 1.6           | 1.1   | 1.3           | 1.4   | -            | -     | 1.7         | 1.7   | 1.7             | 1.5   | 96.0        | 88.7  |
|              | Ballroom    | GT          | GT    | 56.4          | 43.2  | 14.1          | 16.7  | -            | -     | 56.0        | 44.4  | 43.8            | 29.6  | 35.2        | 37.5  |
|              | Courtroom   | GT          | GT    | 62.5          | 54.1  | 5.3           | 3.6   | -            | -     | 66.8        | 66.3  | 67.2            | 69.1  | 38.7        | 51.1  |
|              | Museum      | GT          | GT    | 13.5          | 11.1  | 0.8           | 1.2   | -            | -     | 14.8        | 13.5  | 12.3            | 11.0  | 100.0       | 98.4  |
|              | Palace      | GT          | GT    | 3.1           | 3.9   | -             | -     | -            | -     | 25.6        | 27.7  | 27.0            | 35.7  | 38.6        | 61.4  |
|              | Temple      | GT          | GT    | 0.4           | 0.9   | 0.5           | 1.2   | -            | -     | 55.5        | 60.7  | 80.7            | 72.2  | 99.3        | 96.9  |

Table 17. **Detailed per-scene SfM results on Tanks and Temples [55].** We use COLMAP GT [97] for evaluation as in MASt3R-SfM [28]. The baseline results are obtained from MASt3R-SfM. (-) indicates failure.





Figure 9. **Qualitative showcase** of SLAM on 7scenes [105]. We visualize keyframe poses as red cameras and non-keyframe poses as green dots.



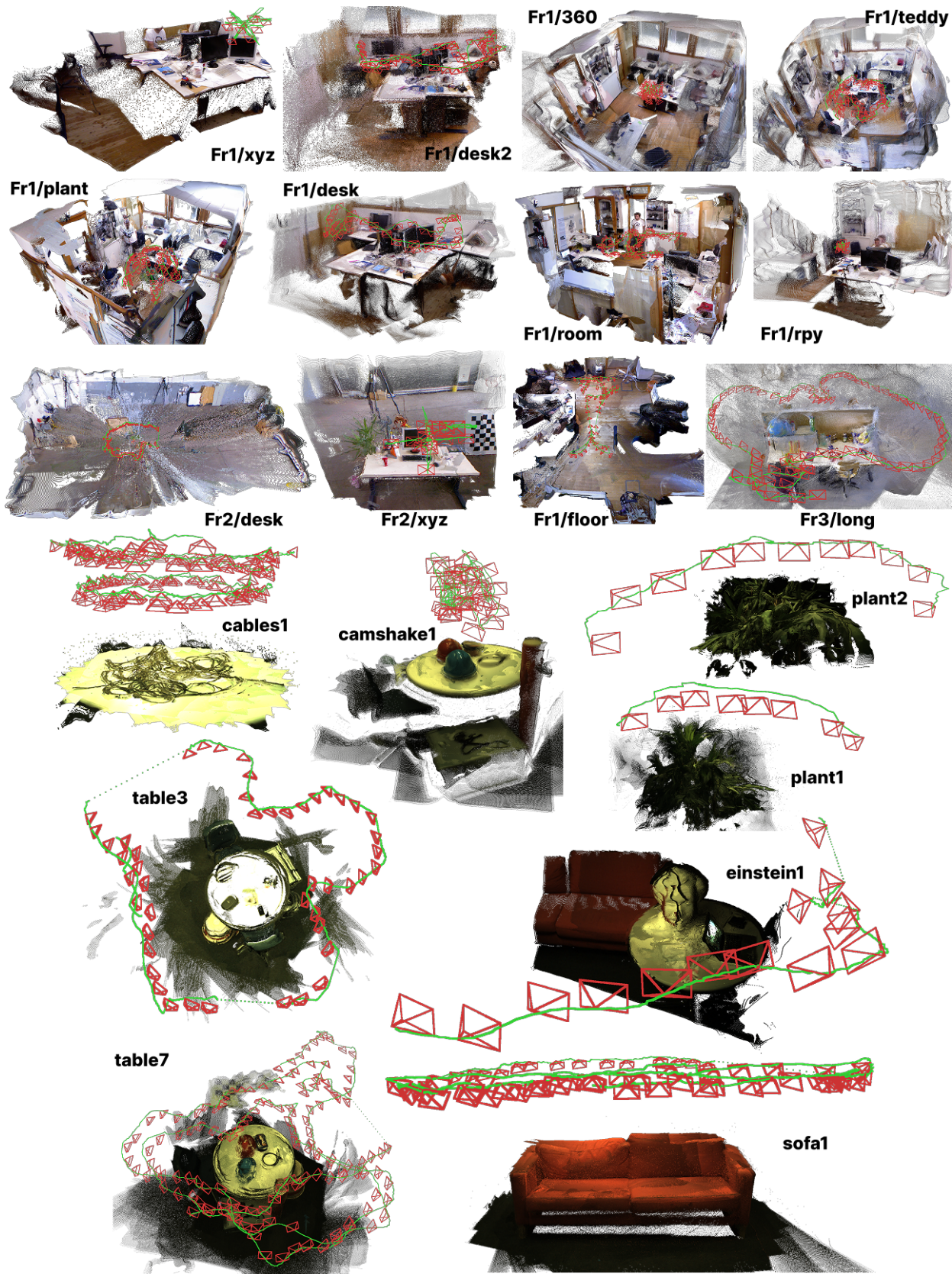


Figure 10. Qualitative showcase of SLAM.





Figure 11. **Qualitative showcase** of structure from motion on ETH3D [100]



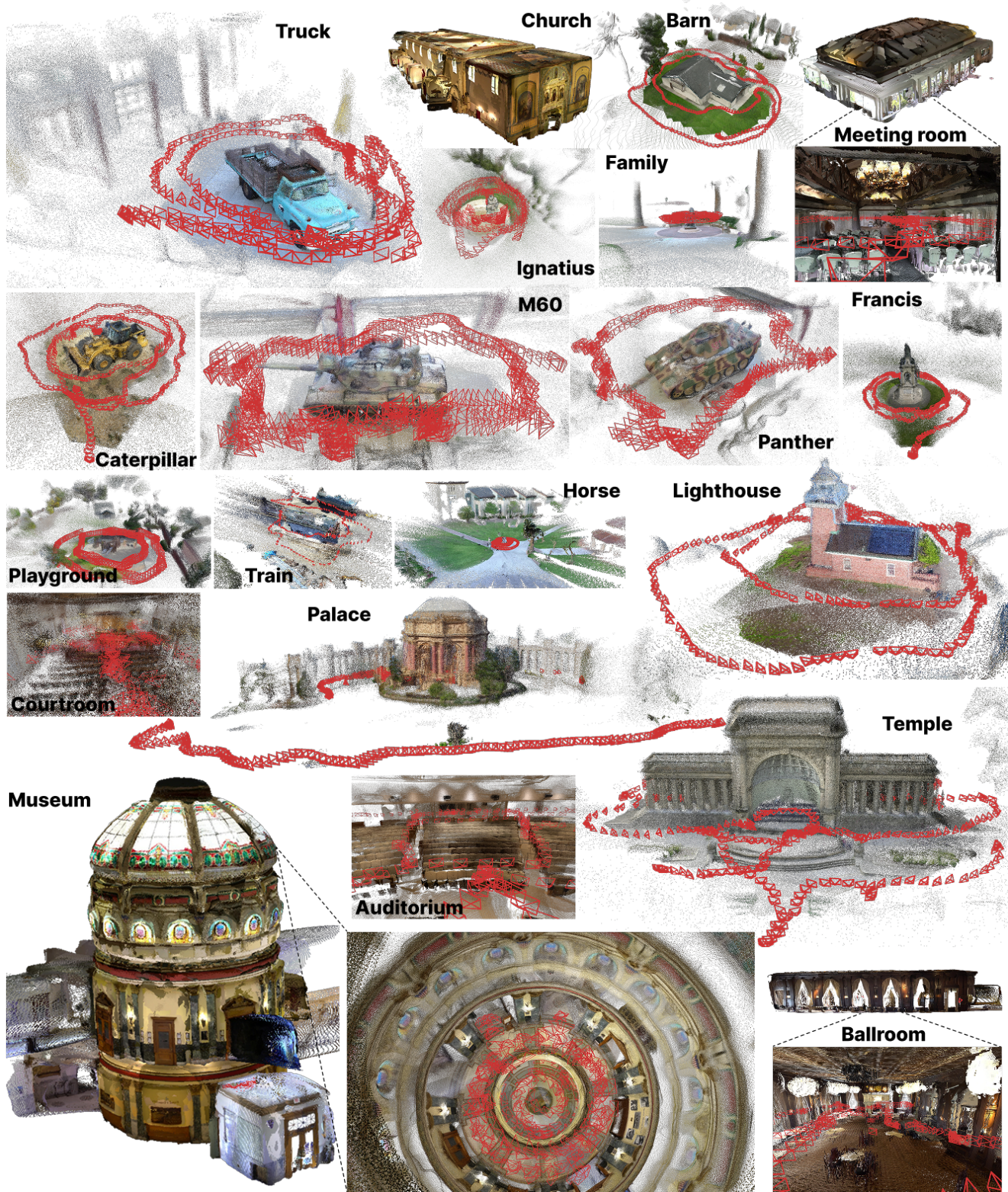


Figure 12. Qualitative showcase of structure from motion on Tanks and Temples [55] dataset.



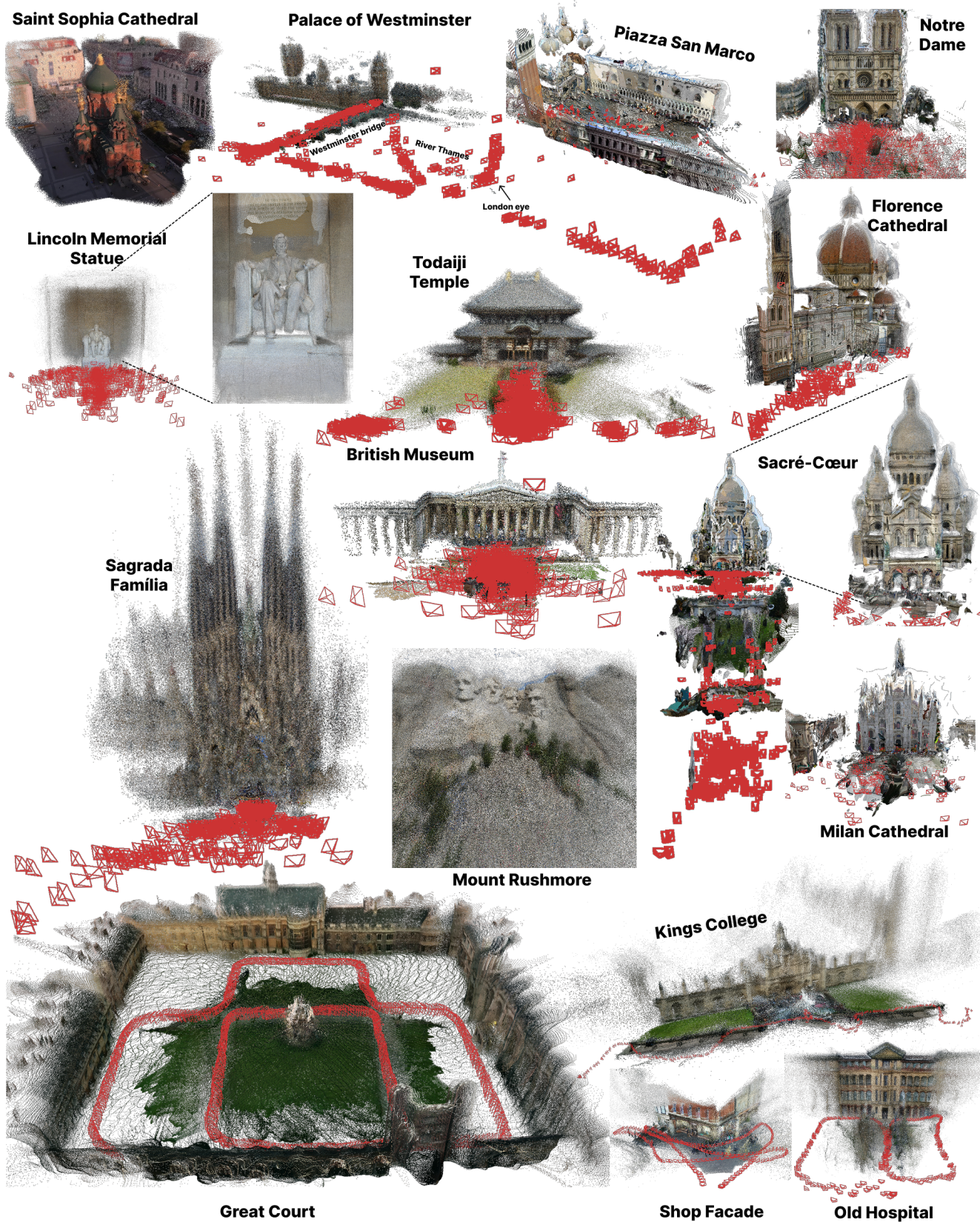


Figure 13. **Qualitative showcase** of structure from motion on IMC phototourism [49], Cambridge landmarks [53]. The results are randomly downsampled to 3 million points for visualization purpose.



## References

- [1] 3Blue1Brown. Hilbert’s curve: Is infinite math useful?, 2017. Accessed: 2025-09-26. [2](#)
- [2] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjarholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. [5](#), [6](#), [11](#)
- [3] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 72–79, 2009. [1](#), [2](#)
- [4] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. [10](#)
- [5] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022. [5](#)
- [6] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. [11](#), [13](#)
- [7] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6684–6692, 2017. [2](#)
- [8] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6218–6228, 2021. [7](#), [10](#)
- [9] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5044–5053, 2023. [2](#)
- [10] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. *arXiv preprint arXiv:2404.14351*, 2024. [2](#), [8](#), [10](#), [12](#), [13](#)
- [11] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 611–625. Springer, 2012. [6](#), [11](#)
- [12] Johann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. [10](#)
- [13] Johann Cabon, Lucas Stofl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1050–1060, 2025. [2](#), [6](#), [7](#), [8](#), [11](#), [13](#)
- [14] Ruojin Cai, Joseph Tung, Qianqian Wang, Hadar Averbuch-Elor, Bharath Hariharan, and Noah Snavely. Doppel-gangers: Learning to disambiguate images of similar structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 34–44, 2023. [9](#)
- [15] Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer++: Revealing the devil in transformer’s details for multi-view stereo. *arXiv preprint arXiv:2401.11673*, 2024. [6](#), [11](#), [13](#)
- [16] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16133, 2022. [2](#)
- [17] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 333–350. Springer, 2022. [2](#)
- [18] Wanli Chen, Xinge Zhu, Guojin Chen, and Bei Yu. Efficient point cloud analysis using hilbert curve. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 730–747. Springer, 2022. [2](#)
- [19] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025. [1](#), [2](#)
- [20] Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. Long3r: Long sequence streaming 3d reconstruction. *arXiv preprint arXiv:2507.18255*, 2025. [2](#)
- [21] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020. [7](#), [11](#), [12](#)
- [22] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. [5](#), [6](#), [10](#), [11](#)
- [23] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007. [1](#), [2](#)
- [24] Junyuan Deng, Heng Li, Tao Xie, Weiqiang Ren, Qian Zhang, Ping Tan, and Xiaoyang Guo. Sail-recon: Large sfm by augmenting scene regression with localization. *arXiv preprint arXiv:2508.17972*, 2025. [2](#)
- [25] Eric Dexheimer and Andrew J Davison. Learning a depth covariance function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13122–13131, 2023. [7](#), [11](#)



- [26] Eric Dexheimer and Andrew J Davison. Como: Compact mapping and odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–365. Springer, 2024. 7, 11
- [27] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16739–16752, 2025. 2
- [28] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Johann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2025. 2, 8, 12, 13
- [29] Ainaz Eftekhari, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10786–10796, 2021. 5, 11
- [30] Sven Elfle, Qunjie Zhou, and Laura Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16774–16784, 2025. 2
- [31] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2017. 7, 11
- [32] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*, 2025. 2
- [33] Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Driv3r: Learning dense 4d reconstruction for autonomous driving. *arXiv preprint arXiv:2412.06777*, 2024. 2
- [34] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 1
- [35] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan De Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 753–762. IEEE, 2025. 5, 11
- [36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 1, 5, 6, 11
- [37] Michael Grupp. evo: Python package for the evaluation of odometry and slam., 2017. 6
- [38] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21594–21603, 2024. 8, 12, 13
- [39] David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. In *Dritter Band: Analysis- Grundlagen der Mathematik- Physik Verschiedenes: Nebst Einer Lebensgeschichte*, pages 1–2. Springer, 1935. 2
- [40] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. 2024. 13
- [41] Wenbo Hu, Yining Hong, Yanjun Wang, Leison Gao, Zibu Wei, Xingcheng Yao, Nanyun Peng, Yonatan Bitton, Idan Szepes, and Kai-Wei Chang. 3dllm-mem: Long-term spatial-temporal memory for embodied 3d large language model. *arXiv preprint arXiv:2505.22657*, 2025. 2
- [42] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. 2
- [43] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2821–2830, 2018. 10
- [44] Rui Huang, Guangyao Zhai, Zuria Bauer, Marc Pollefeys, Federico Tombari, Leonidas Guibas, Gao Huang, and Francis Engelmann. Video perception models for 3d scene synthesis. *arXiv preprint arXiv:2506.20601*, 2025. 2
- [45] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisín Mac Aodha, Gabriel Brostow, and Jamie Watson. Mvsanywhere: Zero-shot multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11493–11504, 2025. 6, 11, 13
- [46] Chris L Jackins and Steven L Tanimoto. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3):249–270, 1980. 2
- [47] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1071–1081, 2025. 2, 6, 11, 13
- [48] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. *arXiv preprint arXiv:2504.07961*, 2025. 2
- [49] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2): 517–547, 2021. 1, 12, 18
- [50] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5): 922–923, 1976. 4

- [51] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9492–9502, 2024. 5, 11
- [52] 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. *arXiv preprint arXiv:2509.13414*, 2025. 2, 6, 11
- [53] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015. 18
- [54] Ramil Khafizov, Artem Komarichev, Ruslan Rakhimov, Peter Wonka, and Evgeny Burnaev. G-cut3r: Guided 3d reconstruction with camera and depth prior integration. *arXiv preprint arXiv:2508.11379*, 2025. 2
- [55] 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. 1, 5, 6, 8, 11, 12, 13, 17
- [56] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Conference on Robot Learning*, pages 34–45. PMLR, 2022. 7, 11
- [57] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. *arXiv preprint arXiv:2508.10893*, 2025. 2
- [58] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–91. Springer, 2024. 2, 5, 6, 11
- [59] Haodong Li, Chen Wang, Jiahui Lei, Kostas Daniilidis, and Lingjie Liu. Stereodiff: Stereo-diffusion synergy for video depth estimation. *arXiv preprint arXiv:2506.20756*, 2025. 2
- [60] Rui Li, Biao Zhang, Zhenyu Li, Federico Tombari, and Peter Wonka. Lari: Layered ray intersections for single-view 3d geometric reasoning. *arXiv preprint arXiv:2504.18424*, 2025. 2
- [61] Zhiqi Li, Chengrui Dong, Yiming Chen, Zhangchi Huang, and Peidong Liu. Vicasplat: A single run is all you need for 3d gaussian splatting and camera estimation from unposed video frames. *arXiv preprint arXiv:2503.10286*, 2025. 2
- [62] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10486–10496, 2025. 8, 12
- [63] Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunchua Shen, and Tong He. Wint3r: Window-based streaming reconstruction with camera token pool. *arXiv preprint arXiv:2509.05296*, 2025. 2
- [64] Zhen Colin Li, Weiwei Sun, Shrisudhan Govindarajan, Shaobo Xia, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Noksr: Kernel-free neural surface reconstruction via point cloud serialization. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 78–89. IEEE, 2025. 2
- [65] Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. Movies: Motion-aware 4d dynamic view synthesis in one second. *arXiv preprint arXiv:2507.10065*, 2025. 2
- [66] Chin-Yang Lin, Cheng Sun, Fu-En Yang, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. Longsplat: Robust unposed 3d gaussian splatting for casual long videos. *arXiv preprint arXiv:2508.14041*, 2025. 2
- [67] Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 424–440. Springer, 2024. 7, 11, 12
- [68] Sidun Liu, Wenyu Li, Peng Qiao, and Yong Dou. Regist3r: Incremental registration with stereo foundation model. *arXiv preprint arXiv:2504.12356*, 2025. 2
- [69] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16651–16662, 2025. 2
- [70] Thibaut Loiseau, Guillaume Bourmaud, and Vincent Lepetit. Alligat0r: Pre-training through co-visibility segmentation for relative camera pose regression. *arXiv preprint arXiv:2503.07561*, 2025. 2
- [71] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22820–22830, 2025. 2
- [72] Ziqi Lu, Heng Yang, Danfei Xu, Boyi Li, Boris Ivanovic, Marco Pavone, and Yue Wang. Lora3d: Low-rank self-calibration of 3d geometric foundation models. *arXiv preprint arXiv:2412.07746*, 2024. 2
- [73] Jiahao Ma, Lei Wang, David Ahmedt-Aristizabal, Chuong Nguyen, et al. Puzzles: Unbounded video-depth augmentation for scalable end-to-end 3d reconstruction. *arXiv preprint arXiv:2506.23863*, 2025. 2
- [74] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl(4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 4, 7, 8
- [75] Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3r: Training-free token eviction for memory-bounded streaming visual geometry transformers. *arXiv preprint arXiv:2509.17650*, 2025. 2
- [76] Jinjie Mai, Wenxuan Zhu, Haozhe Liu, Bing Li, Cheng Zheng, Jürgen Schmidhuber, and Bernard Ghanem. Can



- video diffusion model reconstruct 4d geometry? *arXiv preprint arXiv:2503.21082*, 2025. 2
- [77] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18039–18048, 2024. 7, 11
- [78] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982. 2
- [79] Andreas Meuleman, Ishaan Shah, Alexandre Lanvin, Bernhard Kerbl, and George Drettakis. On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics (TOG)*, 44(4):1–14, 2025. 2
- [80] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421, 2020. 1
- [81] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022. 2
- [82] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 7, 11, 12
- [83] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 2, 7, 8, 12
- [84] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 5, 6, 11
- [85] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011. 1, 7, 10
- [86] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013. 2
- [87] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7855–7862. IEEE, 2019. 6, 11
- [88] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2
- [89] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 20133–20143, 2023. 10
- [90] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 1
- [91] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 523–540. Springer, 2020. 2
- [92] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. 6, 11, 13
- [93] Quanhao Qian, Guoyang Zhao, Gongjie Zhang, Jiuniu Wang, Ran Xu, Junlong Gao, and Deli Zhao. Gp3: A 3d geometry-aware policy with multi-view images for robotic manipulation. *arXiv preprint arXiv:2509.15733*, 2025. 2
- [94] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 10
- [95] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 10901–10911, 2021. 10
- [96] Jerome Revaud, Yohann Cabon, Romain Brégier, JongMin Lee, and Philippe Weinzaepfel. Sacreg: Scene-agnostic coordinate regression for visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 688–698, 2024. 2
- [97] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 8, 12, 13
- [98] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 10
- [99] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1, 2, 6, 8, 9, 11, 12, 13
- [100] Thomas Schops, Johannes L Schonberger, 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 *Proceedings*

- of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3260–3269, 2017. 5, 6, 8, 11, 12, 16
- [101] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019. 6, 8, 12
- [102] Philipp Schröppel, Jan Bechtold, Artemij Amiranashvili, and Thomas Brox. A benchmark and a baseline for robust multi-view depth estimation. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 637–645. IEEE, 2022. 5, 6, 11, 12, 13
- [103] Duochao Shi, Weijie Wang, Donny Y Chen, Zeyu Zhang, Jia-Wang Bian, Bohan Zhuang, and Chunhua Shen. Re-visiting depth representations for feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2506.05327*, 2025. 2
- [104] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 6
- [105] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 2, 5, 10, 11, 12, 14
- [106] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. *arXiv preprint arXiv:2404.15259*, 2024. 12, 13
- [107] 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. 5
- [108] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012. 6, 8, 10, 12
- [109] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15598–15607, 2021. 1
- [110] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 10
- [111] Yang-Tian Sun, Xin Yu, Zehuan Huang, Yi-Hua Huang, Yuan-Chen Guo, Ziyi Yang, Yan-Pei Cao, and Xiaojuan Qi. Unigeo: Taming video diffusion for unified consistent geometry estimation. *arXiv preprint arXiv:2505.24521*, 2025. 2
- [112] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 10
- [113] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5283–5293, 2025. 2
- [114] Aether Team, Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, et al. Aether: Geometric-aware unified world modeling. *arXiv preprint arXiv:2503.18945*, 2025. 2
- [115] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 7, 12
- [116] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:16558–16569, 2021. 7, 11, 12
- [117] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. 13(4):376–380, 2002. 4
- [118] 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 (CVPR)*, pages 5038–5047, 2017. 6, 11, 13
- [119] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 5, 11
- [120] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster vggnet with block-sparse global attention. *arXiv preprint arXiv:2509.07120*, 2025. 2
- [121] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14194–14203, 2021. 11, 13
- [122] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 78–89. IEEE, 2025. 1, 2, 3, 5, 6, 7, 8, 11, 13
- [123] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Coslam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13293–13302, 2023. 1, 6



- [124] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5294–5306, 2025. 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13
- [125] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020. 10
- [126] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10510–10522, 2025. 2, 6, 7, 11
- [127] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5261–5271, 2025. 3, 4, 5, 11
- [128] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024. 1, 2, 3, 5, 6, 11, 13
- [129] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 2, 6, 7, 8, 11, 13
- [130] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 803–814, 2023. 10
- [131] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:33330–33342, 2022. 2
- [132] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4840–4851, 2024. 2, 4
- [133] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025. 2
- [134] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgb-d objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22378–22389, 2024. 10
- [135] Jiale Xu, Shenghua Gao, and Ying Shan. Freesplatter: Pose-free gaussian splatting for sparse-view 3d reconstruction. *arXiv preprint arXiv:2412.09573*, 2024. 2
- [136] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21924–21935, 2025. 2, 5
- [137] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:21875–21911, 2024. 5, 6, 11, 13
- [138] 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. 11, 13
- [139] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 10
- [140] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9043–9053, 2023. 11
- [141] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1949–1958, 2020. 11, 13
- [142] Ganlin Zhang, Erik Sandström, Youmin Zhang, Manthan Patel, Luc Van Gool, and Martin R Oswald. Glorie-slam: Globally optimized rgb-only implicit encoding point cloud slam. *arXiv preprint arXiv:2403.19549*, 2024. 7, 11
- [143] Ganlin Zhang, Shenhan Qian, Xi Wang, and Daniel Cremers. Vista-slam: Visual slam with symmetric two-view association. *arXiv preprint arXiv:2509.01584*, 2025. 2
- [144] Jingyang Zhang, Shiwei Li, Zixin Luo, Tian Fang, and Yao Yao. Vis-mvsnet: Visibility-aware multi-view stereo network. *International Journal of Computer Vision (IJCV)*, 131(1):199–214, 2023. 6, 11, 13
- [145] Jiakai Zhang, Shouchen Zhou, Haizhao Dai, Xinhang Liu, Peihao Wang, Zhiwen Fan, Yuan Pei, and Jingyi Yu. Cryofastar: Fast cryo-em ab initio reconstruction made easy. *arXiv preprint arXiv:2506.05864*, 2025. 2
- [146] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023. 2
- [147] Songyan Zhang, Yongtao Ge, Jinyuan Tian, Guangkai Xu, Hao Chen, Chen Lv, and Chunhua Shen. Pomato: Marrying pointmap matching with temporal motion for dynamic 3d reconstruction. *arXiv preprint arXiv:2504.05692*, 2025. 2
- [148] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21936–21947, 2025. 2

- [149] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023. [7](#), [12](#)
- [150] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. [2](#)
- [151] Xinyi Zheng, Steve Zhang, Weizhe Lin, Aaron Zhang, Walterio W Mayol-Cuevas, Yunze Liu, and Junxiao Shen. Culture3d: A large-scale and diverse dataset of cultural landmarks and terrains for gaussian-based scene rendering. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 29064–29074, 2025. [7](#)
- [152] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. [5](#)
- [153] Yuxuan Zhou, Xingxing Li, Shengyu Li, Zhuohao Yan, Chunxi Xia, and Shaoquan Feng. Mast3r-fusion: Integrating feed-forward visual model with imu, gnss for high-functionality slam. *arXiv preprint arXiv:2509.20757*, 2025. [2](#)
- [154] Haodong Zhu, Changbai Li, Yangyang Ren, Zichao Feng, Xuhui Liu, Hanlin Chen, Xiantong Zhen, and Baochang Zhang. Surf3r: Rapid surface reconstruction from sparse rgb views in seconds. *arXiv preprint arXiv:2508.04508*, 2025. [2](#)
- [155] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, 2022. [6](#)
- [156] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. [2](#)
- [157] Lojze Zust, Yohann Cabon, Juliette Marrie, Leonid Antsfeld, Boris Chidlovskii, Jerome Revaud, and Gabriela Csurka. Panst3r: Multi-view consistent panoptic segmentation. *arXiv preprint arXiv:2506.21348*, 2025. [2](#)