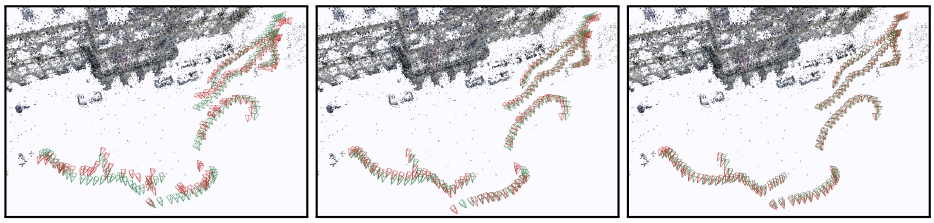
Boosting 3D Foundation Models with Edge-based Pose Optimization

Mattia D'Urso[1] Christian Sormann[2] Mattia Rossi[2]
Friedrich Fraundorfer[1]

[1]Graz University of Technology [2]Sony Europe



(a) Initial VGGT output **(b)** Intermediate step **(c)** Final convergence

Fig. 1: Visualization of three stages of our pose optimization method and the ground truth sparse point cloud for the Graz Town Hall scene (TerraSky3D). Starting from the initial optimization stage (a), provided by the VGGT output, we illustrate an intermediate state (b) and the final refined poses (c). The ground truth poses are shown in green and the optimized ones in red.

Abstract. We introduce **Edge-based Pose Optimization (EPO)**, a trackless geometric optimization framework specifically designed to boost the Structure-from-Motion reconstructions generated by 3D Foundation Models. These models achieve rapid inference by bypassing the time-consuming feature extraction and matching stages of traditional pipelines, where explicit correspondences between each 3D point and multiple images, referred to as tracks, are established. However, their geometric accuracy currently falls short of traditional pipelines. While this can be addressed in a post-processing step via Bundle Adjustment-like refinement, doing so requires extracting feature tracks, thus defeating the original speed advantage. Instead, our fully differentiable framework uses edge map alignment as a proxy for geometric optimization, avoiding feature extraction and track construction entirely. Through extensive evaluation across multiple datasets and tasks, we demonstrate that EPO matches or outperforms Bundle Adjustment-like methods while requiring significantly lower runtime and memory. Notably, its reduced memory footprint makes EPO suitable for consumer-grade hardware, where competing refinement methods cannot run. Code is available at `https://github.com/mattiadurso/EPO`.

Keywords: 3D Reconstruction · Learning-based Optimization · SfM

1 Introduction

The problem of recovering the 3D structure of a scene from a collection of 2D images, known as Structure-from-Motion (SfM), is a fundamental and longstanding problem in computer vision. While the geometric foundations were established early on, in recent years both incremental [31, 43, 44] and global [23, 33, 46, 62] pipelines have advanced in terms of accuracy, reliability, and scalability. These frameworks typically rely on feature extraction and matching to then estimate camera poses and 3D scene points jointly. These are then refined via Bundle Adjustment (BA) [48], a non-linear optimization technique, normally leveraging the Levenberg-Marquardt solver, that minimizes the error between the observations of each reconstructed 3D point at the different images and its actual projections, referred to as reprojection error. Despite its robustness, traditional SfM often struggles with textureless environments or sparse-view scenarios, where point-based features [29] are frequently unreliable.

Recent research has shifted toward replacing complex, handcrafted pipelines with streamlined neural networks [19, 52, 53, 56], referred to as 3D Foundation Models (3DFMs). By leveraging vast quantities of training data, these models generate reconstructions in a rapid, feed-forward fashion. However, while 3DFMs produce results in seconds, their geometric precision rarely matches that of traditional SfM pipelines. Although their estimates can be refined through post-processing, conventional methods such as BA rely on point features and tracks, which defeats the inherent speed advantage of these models. Furthermore, existing global alignment frameworks, primarily designed for DUSt3R-like models [7, 55] or long-sequence scaling [13, 57], typically require explicit 3D point clouds and server-grade hardware.

To address these limitations, we introduce **EPO**, a trackless refinement framework that decouples pose optimization from explicit feature matching, thereby preserving the speed advantages of feed-forward models. By avoiding the overhead of track generation, EPO runs in seconds while maintaining geometric precision comparable to BA-like refinement methods, as shown in Fig. 1. Our main contributions are summarized as follows:

- We propose a robust, differentiable, and trackless optimization framework that boosts the geometric accuracy of 3DFMs through neural pose refinement and first-order optimization.
- We introduce an edge-based reprojection loss and a pose-based early stopping criterion that together reduce runtime to few seconds compared to minutes needed by BA-based refinement methods, while requiring lower memory.
- We provide a comprehensive evaluation across diverse 3DFM, benchmarks and real-world scenarios, including TerraSky3D [9], ScanNet++ [24, 58], and Mip-NeRF 360 [1], demonstrating that EPO consistently performs on par with, or superior to, state-of-the-art refinement methods.

2 Related Work

Geometric Priors and Optimization While sparse keypoints [29] have remained the *de facto* standard for establishing geometric correspondences in SfM, they often prove insufficient in textureless or repetitive environments. Consequently, several works have explored alternative geometric primitives. Research into line-based SfM [27, 35, 36] demonstrates that lines and segments provide superior constraints in man-made environments, where point features are sparse or collinear. Works on edge-map-based SLAM [21, 41, 42, 60, 64] have shown that edges, similarly to lines [26, 27], serve as highly informative observations, preserving structural integrity in scenes where point-based descriptors show poor performance. However, SLAM methods can exploit temporal continuity, incremental spatial priors, and often metrically accurate depth measurements to guide edge-based alignment, whereas our setting involves unordered image collections with none of these advantages, initialized only from noisy model-predicted depth, making trackless geometric optimization inherently more challenging. Building upon this paradigm, our work leverages 2D edge maps [4] and distance transform fields [14] to formulate a differentiable objective function, enabling joint optimization of camera parameters and semi-dense depth by minimizing the distance between projected edges and their 2D observations, achieving robust alignment without explicit feature points or descriptors.

Deep Learning for Tracking and Matching The evolution of SfM has seen handcrafted components systematically replaced by neural networks. Initial efforts focused on learned feature extractors [5, 8, 10, 38, 39, 49, 61] and learned matchers [11, 12, 25, 40, 45]. To address temporal consistency, sparse matching has evolved into persistent point tracking [6, 15]. Architectures such as CoTracker [17, 18] track points jointly across long sequences, effectively handling occlusions and maintaining trajectory coherence. However, these models remain computationally demanding, often requiring significant GPU memory for extended sequences or large keypoint budgets.

End-to-End Pipelines and Foundation Models Recently, the field has transitioned toward "all-in-one" learned frameworks that infer 3D attributes directly from image collections. VGGSfM [53] pioneered this shift with a fully differentiable pipeline that recovers camera poses and 3D structure by leveraging deep 2D point tracks and a differentiable BA layer. This paradigm was significantly advanced by 3DFMs such as DUSt3R [55], which predicts, given a pair of images, the corresponding depth maps that can be used to estimate camera intrinsics and extrinsics. This approach was further extended by MASt3R [22], which augments the DUSt3R architecture with a dedicated matching head. By regressing dense local features and employing a computationally efficient reciprocal matching scheme, MASt3R achieves high-precision correspondences while maintaining robustness to extreme viewpoint changes. MASt3R-SfM [7] then scales this local feature matching approach into a complete SfM solution, by first employing an

efficient image retrieval strategy to construct a viewgraph and then executing a global optimization scheme refined through successive 3D and 2D loss functions.

Building on these advancements, VGGT [52] employs a transformer-based architecture to infer camera poses, depth, and point tracks in a single feed-forward pass. Subsequent research has significantly broadened this paradigm. For instance, MapAnything [19] facilitates metric-scale reconstruction by integrating diverse geometric and semantic priors directly into the inference process. Complementing these efforts, Pi3 [56] introduces a permutation-equivariant framework designed to ensure robust global registration across heterogeneous viewing conditions. Fast3R [57] and Light3R [13] further extend this paradigm to long-sequence settings, enabling efficient reconstruction from hundreds to thousands of images, while Spann3R [51], CUT3R [54], and MUSt3R [3] pursue scalable and streaming reconstruction by predicting pointmaps in a shared coordinate frame without explicit global alignment.

Despite their impressive performance, modern 3DFMs often face a trade-off between inference speed and geometric fidelity. Higher precision typically requires BA-like refinement methods, which rely on feature-based tracks. Unfortunately, building tracks as a post-processing step significantly increases reconstruction runtime, defeating the speed advantage of feed-forward models. In this work, we propose EPO, a trackless geometric optimization framework that preserves this speed advantage while achieving geometric precision on par with, or superior to, BA-like refinement methods.

3 Preliminaries

3DFMs, such as VGGT [52], employ a unified transformer-based architecture that leverages Alternating-Attention layers to reconstruct 3D scene parameters from a collection of N input images $\{I_i\}_{i=1}^{N}$. The process begins by patchifying the input images into visual tokens using a DINO-based backbone [32]. These image tokens are augmented with learnable camera and register tokens before being processed by a transformer backbone to aggregate spatial and temporal context. The resulting tokens are then routed to specialized prediction heads:

- **Camera Head:** Uses self-attention followed by a linear layer to estimate the camera intrinsics $\mathbf{K}$ and extrinsic matrices $\mathbf{P} = [\mathbf{R} \mid \mathbf{t}]$, where $\mathbf{R}$ and $\mathbf{t}$ denote rotation and translation, respectively.
- **Dense Head:** Uses a Dense Prediction Transformer [37] to simultaneously predict per-pixel depth maps $\mathbf{Z}$, 3D point maps, and dense tracking features. The depth maps are subsequently unprojected to 3D to form a point cloud.
- **Tracking Head:** Uses a CoTracker-based architecture [17, 18] to establish consistent 2D correspondences across an unordered image set. While this module provides the geometric constraints required for BA-based scene refinement, achieving high 2D localization precision demands iterative refinement through dense feature correlation and self-attention, making this step computationally expensive and memory-intensive.

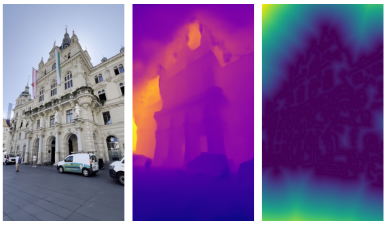


Fig. 2: Example of input RGB image, VGGT raw depth, and the corresponding DTF.

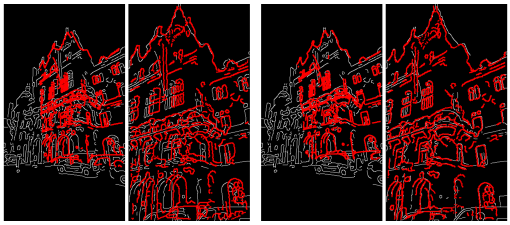


Fig. 3: Bidirectional edge alignment using the initial $\mathcal{G}$ (left) and after refining camera parameters and depth with EPO (right).

4 Method

Given an unordered set of RGB images $\mathcal{I}$, we employ a 3DFM to produce an initial geometric estimate

$$\mathcal{G} = \{(\mathbf{K}_i, \mathbf{P}_i, \mathbf{Z}_i) \mid \forall I_i \in \mathcal{I}\}, \tag{1}$$

as described in Sec. 3. We augment $\mathcal{G}$ by computing, for each image, the edge map $\mathbf{E}_i$ via the Canny edge detector [4] and the corresponding Distance Transform Field (DTF_i) [14], in which each pixel encodes the Euclidean distance to the nearest edge (Fig. 2). Rather than relying on feature-based matching, we use edges to define a geometric alignment measure. We exploit the dense depth maps $\mathbf{Z}_i$ provided by the 3DFM to assess pairwise pose quality from edge proximity in the image plane.

We formulate the objective as an iterative minimization problem. At each iteration s, we compute the pairwise edge alignment (shown in Fig. 3) and the corresponding loss $\mathcal{L}$ over the pairs in the viewgraph. In the first phase, this loss drives updates to the camera intrinsics $\mathbf{K}$ and poses $\mathbf{P}$. In the subsequent joint phase, optimization is extended to include the depth maps $\mathbf{Z}$. All updates are computed via backpropagation using a gradient-based optimizer.

Viewgraph Estimation Similarly to BA, our framework requires establishing geometric correspondences across views. However, we relax the strict requirement for explicit 2D-2D feature correspondences, bypassing the need for feature detectors, matchers, or trackers. Given $\mathcal{G}$, we employ an exhaustive cycle-consistent reprojection strategy to filter out pairs with insufficient reprojected points. Specifically, we project each pixel $\mathbf{p}$ of image I_i onto I_j using the projective function π:

$$\mathbf{p}_{i\to j} = \pi(\mathbf{K}_i, \mathbf{K}_j, \mathbf{P}_i, \mathbf{P}_j, \mathbf{Z}_i; \mathbf{p}). \tag{2}$$

The resulting $\mathbf{p}_{i\to j}$ is then projected back onto I_i to obtain $\mathbf{p}_{i\to j\to i}$. An image pair is called *valid* if sufficiently many pixels satisfy a bidirectional reprojection error $\|\mathbf{p} - \mathbf{p}_{i\to j\to i}\|_2 < \tau$. The set of *valid* pairs forms the viewgraph $\mathcal{E}$.

Pose Refinement We employ a six-layer Multi-Layer Perceptron (MLP) with a skip connection after the third layer, following ACE0 [2]. Camera rotations are represented using the continuous 6D parameterization suggested by Zhou et al. [63]. We model the pose refinement as

$$\mathbf{P}_i^s = \phi(\mathbf{P}_i^0 + \mathrm{MLP}(\mathbf{P}_i^0)) + [\mathbf{0} \mid \delta_i], \tag{3}$$

where $\mathbf{P}_i^0$ and $\mathbf{P}_i^s$ denote the camera pose matrices at initialization and at iteration s, respectively. The operator ϕ applies Gram-Schmidt orthonormalization to recover a valid rotation matrix from the 6D representation. The variable δ_i is a set of learnable parameters acting as a translation offset. We find that the MLP alone struggles to predict accurate translation offsets, often failing to capture fine-grained adjustments; δ_i is therefore introduced to absorb this residual error.

Camera Refinement For images sharing a camera, we first average their focal length predictions to establish a robust initial estimate. We then refine only the focal length via a learnable scaling factor γ_i, such that

$$f_i^s = f_i^0 \cdot (1 + \gamma_i). \tag{4}$$

Depth Refinement We refine depth *pixel-wise* by learning two sets of parameters, α_i and β_i, such that

$$\mathbf{Z}_{i,(x,y)}^s = \mathbf{Z}_{i,(x,y)}^0 \cdot \alpha_{i,(x,y)} + \beta_{i,(x,y)} \tag{5}$$

for each pixel location (x, y). We empirically find that this *pixel-wise* affine parameterization yields better performance than both a single per-pixel offset and a per-image affine correction.

Edge Reprojection Loss We define the global optimization objective as the minimization of the average edge reprojection error across the viewgraph $\mathcal{E}$:

$$\mathcal{L} = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \mathcal{L}_{ij}, \tag{6}$$

where $\mathcal{L}_{ij}$ denotes the bidirectional alignment cost for image pair (i, j):

$$\mathcal{L}_{ij} = \frac{1}{|\mathbf{E}_i|} \sum_{\mathbf{p}\in\mathbf{E}_i} \mathcal{H}\Big(\hat{d}_{i\to j}(\mathbf{p})\Big) + \frac{1}{|\mathbf{E}_j|} \sum_{\mathbf{p}\in\mathbf{E}_j} \mathcal{H}\Big(\hat{d}_{j\to i}(\mathbf{p})\Big). \tag{7}$$

Here, $\mathcal{H}$ denotes the Huber loss, and $\hat{d}_{i\to j}(\mathbf{p}) = \min(d_{i\to j}(\mathbf{p}), \lambda)$ clamps per-pixel distances to mitigate the influence of outliers. The per-pixel distance $d_{i\to j}(\mathbf{p})$ is obtained by projecting each edge pixel $\mathbf{p} \in \mathbf{E}_i$ into the target frame I_j and sampling the precomputed distance transform:

$$d_{i\to j}(\mathbf{p}) = \mathrm{DTF}_j[\pi(\mathbf{K}_i, \mathbf{K}_j, \mathbf{P}_i, \mathbf{P}_j, \mathbf{Z}_i; \mathbf{p})], \quad \forall\, \mathbf{p} \in \mathbf{E}_i. \tag{8}$$

Sampling DTF_j at the projected locations yields a differentiable geometric signal that guides the optimization toward dense edge alignment.

Optimization Schedule After each step s, we evaluate the change in rotation and translation at each pose $\mathbf{P}_i^s$ with respect to the previous iteration $s-1$. The rotation change ΔR_i^s and translation change Δt_i^s, both expressed in degrees, are defined as:

$$\Delta R_i^s = \frac{180}{\pi} \arccos\left(\frac{\mathrm{tr}(\mathbf{R}_i^s(\mathbf{R}_i^{s-1})^\top) - 1}{2}\right), \tag{9}$$

$$\Delta t_i^s = \frac{180}{\pi} \arccos\left(\frac{\mathbf{t}_i^s}{\|\mathbf{t}_i^s\|_2} \cdot \frac{\mathbf{t}_i^{s-1}}{\|\mathbf{t}_i^{s-1}\|_2}\right). \tag{10}$$

To ensure robustness against outliers, we summarize the per-image rotation and translation changes by their 95^{th} percentile over all poses, ΔR_{95}^s and Δt_{95}^s, and define the global pose change as $\theta_{95}^s = \max(\Delta R_{95}^s, \Delta t_{95}^s)$.

The optimization follows a dual-phase convergence schedule designed to stabilize camera parameters before initiating depth refinement.

1. **First Convergence:** Reached when θ_{95}^s remains below a tolerance threshold ϵ_1 for w_1 consecutive steps, at which point depth optimization is initiated.
2. **Second Convergence:** θ_{95}^s is further monitored until it remains below a stricter threshold $\epsilon_2 < \epsilon_1$ for w_2 consecutive steps, at which point the optimization terminates.

Implementation Details We construct the viewgraph $\mathcal{E}$ by selecting image pairs for which at least 12.5% of their reprojected 2D points have a bidirectional reprojection error below $\tau = 3.0$ pixels, as described in Sec. 4.

The MLP pose refiner consists of 12 input units, corresponding to the flattened pose matrix, and 12 output units representing the pose offset. The network has a constant hidden size of 128 channels and ReLU activations. All layers use Kaiming normal initialization [16], except the final layer, which is initialized to zero. All other learnable parameters are initialized to their identity values, ensuring a neutral contribution at the first iteration that grows smoothly as optimization progresses.

To ensure stable convergence, we apply a linear learning rate warmup from 0 to 3×10^{-3} over the first 25 steps, followed by cosine annealing decay over a maximum of $N = 2000$ iterations. Note that the output of the optimization is the refined $\mathcal{G}$, not the MLP pose refiner itself. The framework uses `BF16` for the MLP and the Edge extraction and `FP32` precision for everything else. It is implemented in PyTorch [34] and Triton [47]. All parameters are optimized with the AdamW optimizer [28] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 0.01, except for the translation offset δ, whose weight decay is increased to 0.1.

The optimization terminates based on the convergence criterion of Sec. 4, with sliding window sizes $w_1 = 25$ and $w_2 = 50$ and tolerance levels $\epsilon_1 = 0.5$ and $\epsilon_2 = 0.1$.

5 Experiments

Testing Details We use VGGT [52] as a representative state-of-the-art 3DFM to generate the initial pose and geometry. Nevertheless, our framework remains agnostic to the specific 3DFM, as we show in the Supplementary Material. We compare EPO against two refinement baselines introduced in VGGT: VGGT+BA and VGGT+Ref+BA.

VGGT+BA applies BA directly to the raw VGGT output using 10 query frames and 2048 query points. As in EPO, shared camera intrinsics are averaged to provide a consistent geometric constraint during optimization.

VGGT+Ref+BA incorporates an additional track refinement step prior to BA, consisting of an iterative update of 2D point tracks to improve correspondence accuracy. However, this step relies on dense feature correlation and iterative self-attention, making it computationally intensive and requiring high-end hardware with at least 40 GB of VRAM. On the other hand, EPO runs on consumer-grade GPUs requiring only few GB of VRAM.

Datasets We evaluate across datasets representing diverse capturing scenarios:

- **Indoor:** *ScanNet++* [58], following the data split of Depth Anything 3 [24].
- **Outdoor Forward-Facing:** The test set of *TerraSky3D* [9], comprising large-scale scenes with cameras oriented toward outdoor buildings.
- **Object-Centric:** The scenes provided by the *Mip-NeRF 360* dataset [1].

Due to the memory constraints of VGGT+BA, all scenes are capped at 150 images, equivalent to 2.5 minutes of videos at 1 fps, uniformly sampled from their respective sorted sequences. For *Mip-NeRF 360*, we retain only scenes containing at least 150 images to maximize multi-view coverage while preserving a sufficient number of views for Novel View Synthesis evaluation.

Methods marked with † are run on an NVIDIA H200 due to their higher memory requirements. Transformer-based models benefit from the H200's increased memory bandwidth and architectural optimizations, yielding lower inference times than on the RTX 4090 used for all other methods. To reduce VGGT's memory footprint, we discard intermediate layer outputs that are saved but never consumed; this increases the maximum number of processable images from approximately 50 to 150 on the RTX 4090.

5.1 Pose Refinement

For all datasets, we evaluate pose accuracy and runtime. Pose accuracy is measured against ground-truth poses as the Area Under the recall Curve (AUC) of the maximum angular error between the relative rotation and translation, $\max(\Delta R, \Delta T)$, in degrees, at a threshold of 5°. Runtime is reported as the total *inference* wall-clock time excluding I/O.

Table 1: Comparison of AUC at 5° and the total *inference* wall-clock time on ScanNet++ [24, 58]. +BA, +Ref+BA, and +EPO are optimization methods running on top of VGGT. † indicates run on an NVIDIA H200. We highlight the best and second best AUC score.

	VGGT		+BA		+Ref+BA†		+EPO	
Scene	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓
09C1414F1B	40.1	40.3	66.4	121.4	66.8	335.6	72.3	54.3
1Ada7A0617	48.5	36.8	64.0	129.3	64.8	304.4	66.5	49.4
21D970D8De	66.1	37.1	76.8	118.6	73.5	251.7	83.5	53.2
286B55A2Bf	48.0	37.2	72.4	394.8	72.9	584.3	76.7	54.3
38D58A7A31	61.2	36.7	75.1	96.7	76.8	282.0	81.4	50.2
3E8Bba0176	73.0	36.9	76.7	117.7	75.5	267.0	83.7	50.3
40Aec5Fffa	45.8	36.7	50.7	101.0	51.4	229.8	72.4	50.1
578511C8A9	50.9	36.8	64.6	298.4	65.3	497.3	68.8	55.5
5F99900F09	44.9	36.7	71.8	107.6	71.1	230.5	76.4	49.4
7831862F02	70.1	37.1	80.0	193.5	80.8	218.6	87.6	48.3
7Bc286C1B6	50.3	36.7	62.2	228.9	62.9	258.0	71.0	48.7
9071E139D9	64.9	36.9	73.3	108.4	75.9	179.8	74.1	50.8
Acd95847C5	69.3	37.3	70.8	175.7	74.5	352.4	79.8	50.4
Bcd2436Daf	53.8	38.2	73.2	209.0	78.4	267.6	82.7	61.7
Bde1E479Ad	56.8	37.5	70.0	248.1	73.5	340.8	80.8	46.5
C4C04E6D6C	50.2	36.4	68.2	181.0	69.9	405.2	78.9	59.1
C5439F4607	48.5	36.9	68.2	218.6	69.1	250.2	73.8	53.3
Cc5237Fd77	46.9	36.9	67.1	157.1	67.8	348.2	72.9	50.7
F3D64C30F8	57.2	36.1	74.0	118.5	69.0	248.4	77.6	54.1
Fb5A96B1A2	66.0	37.1	75.2	108.1	77.1	217.6	81.4	51.9
Mean	55.6	37.1	70.0	171.6	70.9	303.5	77.1	52.1

Indoor ScanNet++ [58] is a challenging indoor benchmark characterized by complex geometries and varying lighting conditions. As shown in Tab. 1, the raw VGGT output is rather inaccurate on these scenes. Both BA variants improve upon the raw results, though track refinement yields little additional benefit. We hypothesize that this is due to the prevalence of textureless regions and repetitive structures, which hinder reliable keypoint detection and accurate refinement. EPO is less sensitive to these conditions, as edges provide more informative geometric observations in feature-poor environments. In particular, EPO improves AUC by 6 points over the best BA-based method, while reducing execution time by ∼6×.

Outdoor TerraSky3D [9] is a recent dataset of popular European landmarks, whose test set comprises nine large-scale outdoor scenes captured from a ground perspective with various iPhone cameras.

As reported in Tab. 2, VGGT accuracy is similar to the indoor case, yet both BA-based approaches are considerably more effective here. We attribute this to the greater texture prevalence in outdoor scenes, which enables more reliable track refinement and improves the initial camera trajectories. Nevertheless, EPO improves AUC by ∼4 points over the best BA-based method, adding only an average of ∼14 seconds on top of VGGT runtime.

Table 2: Comparison of AUC at 5° and the total *inference* wall-clock time on TerraSky3D [9]. +BA, +Ref+BA, and +EPO are optimization methods running on top of VGGT. † indicates run on an NVIDIA H200. We highlight the best and second best AUC score.

	VGGT		**+BA**		**+Ref+BA†**		**+EPO**	
Scene	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓
Graz Church	72.1	37.9	81.6	929.7	87.4	1051.4	90.5	51.0
Graz Townhall	18.8	42.4	65.6	107.1	65.9	229.0	66.7	63.9
Graz University	35.5	37.9	41.5	256.1	41.8	374.6	50.5	56.6
Munich Frauenkirche	68.2	30.0	83.5	107.0	87.7	206.1	89.7	46.0
Munich Marienplatz	55.3	40.1	70.5	109.2	78.6	257.5	68.1	46.2
Munich Theatinerkirche	67.0	28.0	71.6	151.6	77.7	341.7	89.4	42.6
Salzburg Andräkirche	70.6	26.1	75.9	112.0	76.1	203.1	85.3	29.6
Salzburg Rechte Altstadt	68.8	9.5	86.4	42.1	93.5	91.1	90.7	20.8
Vienna State Opera	54.9	26.7	63.2	219.7	70.9	284.5	82.0	44.9
Mean	56.8	31.0	71.1	226.1	75.5	337.7	79.2	44.6

Table 3: Comparison of AUC at 5° and the total *inference* wall-clock time on Mip-NeRF 360 [1]. +BA, +Ref+BA, and +EPO are optimization methods running on top of VGGT. † indicates run on an NVIDIA H200. We highlight the best and second best AUC score.

	VGGT		**+BA**		**+Ref+BA†**		**+EPO**	
Scene	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓
Bicycle	77.8	44.0	83.4	88.6	86.5	179.8	90.0	54.6
Bonsai	69.7	39.8	87.7	103.1	87.8	215.6	90.9	46.2
Counter	83.5	40.8	91.5	128.2	94.7	249.3	94.7	45.3
Flowers	34.3	43.0	70.2	91.6	70.7	177.2	78.7	59.7
Garden	84.8	43.6	92.2	85.7	95.4	143.5	92.7	49.3
Kitchen	87.8	41.0	87.2	117.1	88.7	224.0	94.8	45.1
Room	68.6	41.1	88.5	113.0	91.8	281.9	91.8	47.3
Mean	72.2	41.9	85.5	103.9	87.8	210.2	90.5	49.6

Object-Centric Mip-NeRF 360 [1] is an object-centric dataset primarily used for NVS evaluation. As with the previous datasets, geometric optimization significantly improves upon the raw VGGT output. Also in this setting, VGGT+EPO achieves higher geometric accuracy than BA-based approaches, while delivering a 4× speedup on a consumer-grade GPU at a fraction of the cost, as shown in Tab. 3.

5.2 Reprojection Error

Tab. 4 reports the average median reprojection error (RE) for the sparse point clouds and the average 2D observation count (Obs.) across all datasets. We define an observation as any 2D image feature contributing to the optimization, either a detected keypoint (for BA-based methods) or an edge pixel (for EPO). For BA-based variants, RE is the L_2 distance between the projected 3D point and its associated 2D keypoint.

EPO does not rely on explicit 3D-2D correspondences; instead, we assess geometric consistency via bidirectional 2D-2D reprojection. Following Eq. (8), we

reproject edges across views and sample the DTF at the target locations, using these distances as a proxy for RE. Given the high density of such observations, we report the median RE averaged first per scene and then across each dataset.

EPO achieves a comparable mean RE of 2.0 px to Ref+BA, while producing 1.7× more 2D observations than the BA baselines.

Table 4: Average median RE (in pixels) and 2D observations (in thousands) per scene. RE is the raw reprojection error (no robustification). Best results are in green.

	TerraSky3D		**ScanNet++**		**Mip-NeRF 360**		**Mean**	
VGGT	RE↓	Obs.↑	RE↓	Obs.↑	RE↓	Obs.↑	RE↓	Obs.↑
↳ +BA	3.7	102	3.4	24	1.6	107	2.9	78
↳ +Ref+BA	2.6	102	2.2	25	1.3	109	2.0	79
↳ +EPO	1.2	150	1.9	60	3.0	199	2.0	136

5.3 Novel View Synthesis

Recently, due to a scarcity of real-world datasets with ground-truth poses, several works [2, 52, 62] have proposed evaluating SfM reconstruction quality via downstream tasks, the most common being Novel View Synthesis (NVS) [50]. This involves rendering frameworks such as Neural Radiance Fields (NeRF) [30] or 3D Gaussian Splatting (3DGS) [20] to reconstruct a scene from the camera poses estimated by the SfM pipeline. The quality of the synthesized views then serves as a proxy for the accuracy and consistency of the underlying geometric reconstruction: high-fidelity rendering indicates that the estimated camera trajectories and scene geometry satisfy the strict multi-view constraints required for photorealistic synthesis.

In our experiments, we train a 3DGS [20] model for 30,000 steps on each SfM reconstruction, following the data split of [20]. Note that COLMAP pseudo-GT values are lower than those reported in [20] because the scenes are downsampled as described in Sec. 5, slightly reducing visual coverage.

As shown in Tab. 5, VGGT+EPO outperforms VGGT+Ref+BA on all metrics, consistent with the geometric accuracy reported in Tab. 3. Fig. 4 further confirms this trend qualitatively: images rendered from VGGT+EPO reconstructions exhibit superior visual fidelity, with sharp structural details and high-frequency background textures well preserved, whereas renders from both BA-based methods show visible degradation.

5.4 Ablation Study

Design and Runtime To motivate our design choices, we conduct a two-stage ablation study evaluating pose accuracy (AUC at 5°), perceptual quality of rendered images (LPIPS [59]), and inference time.

Table 5: Novel View Synthesis evaluation on the Mip-NeRF 360 [1] dataset. All models were rendered using 3DGS [20] trained for 30,000 steps. Performance is quantified in terms of PSNR, SSIM, and LPIPS using standard test-set sampling. In green the best mean scores for each metric.

	GT			VGGT			VGGT+BA			VGGT+Ref+BA			VGGT+EPO		
Scene	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**
Bicycle	21.77	0.685	0.271	16.77	0.357	0.524	18.74	0.453	0.444	19.43	0.488	0.423	19.25	0.504	0.399
Bonsai	30.14	0.931	0.207	21.96	0.682	0.414	25.36	0.804	0.310	25.50	0.816	0.303	27.60	0.874	0.253
Counter	27.98	0.896	0.195	22.98	0.719	0.334	25.49	0.814	0.278	26.33	0.847	0.249	25.87	0.834	0.240
Flowers	20.69	0.624	0.319	14.86	0.269	0.612	17.41	0.375	0.467	17.32	0.380	0.467	19.09	0.505	0.386
Garden	27.03	0.848	0.128	20.80	0.467	0.387	22.86	0.669	0.270	23.73	0.733	0.222	23.27	0.685	0.234
Kitchen	28.83	0.927	0.116	20.69	0.595	0.333	20.61	0.652	0.333	20.73	0.677	0.320	23.37	0.776	0.201
Room	30.39	0.903	0.229	25.86	0.797	0.349	28.40	0.859	0.275	28.40	0.873	0.258	29.06	0.867	0.274
Mean	26.69	0.831	0.209	20.56	0.555	0.422	22.70	0.661	0.339	23.06	0.688	0.320	23.93	0.721	0.284

Table 6: Ablation study on Mip-NeRF 360 evaluating LPIPS and AUC at 5° threshold. Time (in seconds) refers to the optimization only duration.

	Steps	LPIPS↓	AUC↑	Time↓
VGGT (baseline)	0	0.422	72.2	0
Free Variables	100	0.343	77.0	4
↳ w/ Pose MLP	100	0.339	79.0	4
↳ w/ Parametric K and Z	100	0.323	80.2	4
↳ w/ Translation Offset δ	100	0.308	82.0	4
↳ w/ 10× Iterations	1000	0.269	91.0	14
↳ w/ 20× Iterations	2000	0.266	92.1	25
↳ w/ Stopping Criterion	*avg.* 387	0.284	90.5	7

Table 7: Ablation study on stopping criteria. Max refers to the maximum AUC at 5° threshold reached over 2,000 optimization steps. Time (in seconds) includes optimization runtime only.

	Max	Fixed		Loss-based		Pose-based	
	AUC↑	AUC↑	Time↓	AUC↑	Time↓	AUC↑	Time↓
TerraSky3D	80.1	80.0	24	76.0	7	79.2	14
ScanNet++	78.3	77.1	25	76.9	16	77.1	15
Mip-NeRF 360	92.3	91.1	25	89.5	9	90.5	7
Mean	83.6	82.7	25	80.8	11	82.2	12

The initial formulation, which optimizes the camera parameters in $\mathcal{G}$ as free variables, frequently becomes unstable beyond 100 iterations. We therefore first isolate the contribution of individual components within this low-iteration regime, then evaluate the full system at higher step counts using the more stable configurations. All configurations follow a two-stage schedule: the first stage stabilizes camera parameters, after which optimization continues for a fixed number of steps including depth refinement. The only exception is the last row, which uses our proposed adaptive stopping criterion.

As detailed in Tab. 6, our baseline optimizes the elements of $\mathcal{G}$ (encoded as q, t, $\log \frac{f}{W}$, $\frac{1}{Z}$) as free variables. We then incrementally add the MLP pose refiner, the reparameterizations of $\mathbf{K}$ and $\mathbf{Z}$ described in Sec. 4, and finally the translation offset δ. Each addition yields significant gains in pose accuracy and rendering quality at negligible computational cost.

The lower portion of Tab. 6 reports the effect of extended optimization. While increasing the iteration count brings incremental gains, it introduces substantial computational overhead. Our adaptive stopping criterion effectively addresses this trade-off, reducing the optimization runtime by approximately 50% and 70% relative to the 1000- and 2000-iteration baselines, respectively, while maintaining comparable geometric accuracy and perceptual fidelity. Since complex scenes often require more iterations to converge, this criterion is essential for ensuring efficiency across diverse environments without sacrificing reconstruction quality.

(a) VGGT **(b)** VGGT+Ref+BA **(c)** VGGT+EPO (Ours)

Fig. 4: Qualitative Novel View Synthesis Results. Example renderings from the *Garden* scene of the Mip-NeRF 360 dataset [1]. All 3DGS models were initialized using SfM reconstructions produced by their respective methods. (a) Base VGGT output. (b) Refined reconstruction utilizing track refinement and BA (VGGT+Ref+BA). (c) Our proposed edge-based optimization (VGGT+EPO). Notably, our approach preserves sharper structural details in the background compared to the BA-based refinement.

Stopping Criteria Tab. 7 reports AUC at 5° and runtime of EPO across datasets under various stopping criteria, with a maximum budget of 2000 iterations. The leftmost column presents the maximum AUC achieved by running EPO for the full 2000 iterations and sampling the output every 50 steps; it therefore represents an empirical upper bound on achievable accuracy.

The remaining three columns correspond to EPO configurations using a *fixed* iteration count, a *loss-based* stopping criterion, and our proposed *pose-based* stopping criterion. As described in Sec. 4, the optimization follows two stages: the first optimizes camera intrinsics and extrinsics only, while the second adds depth. All criteria share the same first stage; the distinction lies in how the second stage is terminated. The *fixed* approach exhausts the full iteration budget after the first stage. The *loss-based* criterion monitors relative changes in the loss, stopping when it remains below a threshold $\epsilon_2 = 5 \times 10^{-4}$ for $w_2 = 100$ consecutive steps. Our *pose-based* criterion instead monitors pose convergence as described in Sec. 4.

As shown in Tab. 7, our pose-based criterion maintains geometric accuracy comparable to the full budget (82.2 vs. 83.6 AUC). It matches the accuracy of the *fixed* configuration (82.2 vs. 82.7 AUC) while reducing runtime by 2×, and improves over the *loss-based* one (82.2 vs. 80.8 AUC) at comparable runtime.

Time Breakdown Fig. 5 breaks down the per-stage runtime of VGGT+BA, VGGT+Ref+BA, and VGGT+EPO on Mip-NeRF 360. For BA-based methods, track computation constitutes the most time-consuming stage. By contrast, EPO's trackless design, entirely eliminates this stage, allowing it to outpace both BA-based baselines despite relying on a first-order optimizer rather than the second-order Levenberg–Marquardt solver used by BA.

The efficiency advantage of our method is further highlighted by a cross-hardware comparison. Running VGGT on an NVIDIA H200 yields a nearly 3× speedup, yet VGGT+Ref+BA† remains approximately 4× slower than EPO running on an NVIDIA RTX 4090 due to the track establisment process.

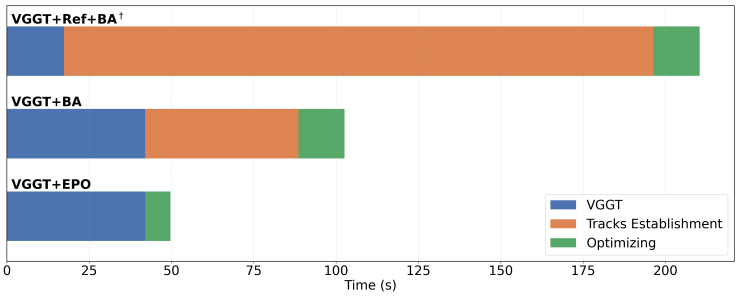


Fig. 5: Absolute and relative inference time breakdown for the optimization methods evaluated in Tab. 3. † indicates runs on an NVIDIA H200.

6 Discussion and Limitations

6.1 Boosting *Different* 3D Foundation Models

We conducted the same experiments described in Sec. 5.1 using various 3DFMs reconstructions as a starting point to demonstrate our method's generalization across different models. Aside from the RGB images used for edge extraction, EPO requires only camera parameters (focal length and pose) and depth maps, which are all provided by the considered models.

Furthermore, we disabled any masking of output depth maps to align with VGGT raw output format. This also ensures that sufficient depth information is available on detected edges.

In Tab. 8, we report the average AUC at 3° and 5° for the TerraSky3D, ScanNet++, and Mip-NeRF 360 datasets. We supplement the VGGT [52] results with evaluations for MapAnything [19] and Pi3 [56] leveraging [19]'s modular software framework. As shown by these results, EPO consistently improves geometric accuracy across all tested 3DFMs by a significant margin.

Table 8: Pose evaluation in terms of AUC at 3° and 5°. For each model, the first row measures model raw output performance, while the second row the output optimized with EPO. The best results are highlighted in green.

	TerraSky3D		**ScanNet++**		**Mip-NeRF 360**	
	AUC@3	AUC@5	AUC@3	AUC@5	AUC@3	AUC@5
VGGT [52]	39.7	56.8	38.9	55.6	59.5	72.2
↳ w/ EPO	69.4	79.2	64.6	77.1	84.3	90.5
MapAnything [19]	35.0	53.0	17.9	37.3	23.9	41.4
↳ w/ EPO	68.1	78.3	41.3	60.4	64.9	74.5
Pi3 [56]	48.3	65.1	55.0	70.0	60.4	71.8
↳ w/ EPO	72.0	81.2	68.9	80.3	76.8	85.9

Implicit Track Regularization and Limitations Our objective couples the edge reprojection loss across the entire viewgraph $\mathcal{E}$, so that each focal length, camera pose, and depth value is constrained by all pairs in which it participates. This implicitly regularizes the optimization toward multi-view consistency and prevents localized drift without explicit, memory-intensive tracking modules.

However, the effectiveness of this implicit regularization depends on the underlying scene geometry and viewgraph topology. We identify two primary limitations:

Texture Sensitivity: Reliance on edge maps makes the optimization vulnerable in scenes dominated by high-frequency, non-structural textures (e.g., dense foliage or reflections), where edge detectors tend to extract inconsistent noise rather than stable structural primitives, degrading the gradient signal and hindering convergence.

Viewgraph Topology: Viewgraphs with low node connectivity may provide insufficient geometric constraints to resolve ambiguities during optimization. This limitation depends on connection density rather than the absolute number of images in the scene.

7 Conclusion

We presented EPO, a trackless optimization framework and a lightweight alternative to traditional Bundle Adjustment. By leveraging edge map alignment, EPO guides the geometric refinement of 3D reconstructions without requiring explicit feature tracks, bypassing the computationally expensive and memory-intensive tracking modules of state-of-the-art 3DFMs that often demand server-grade hardware and dominate total inference time.

Our extensive evaluation across TerraSky3D, ScanNet++, and Mip-NeRF 360 demonstrates that EPO achieves competitive or superior geometric accuracy compared to BA-based pipelines while significantly reducing their computational overhead. We further validated the quality of the refined poses through Novel View Synthesis, where EPO consistently preserved sharper structural details. Ultimately, EPO demonstrates that first-order gradient-based optimization, coupled with dense edge priors, offers a fast, accessible, and trackless path toward high-fidelity 3D geometric refinement on consumer-grade hardware.

Acknowledgements This work has been supported by the FFG under Contract No. 881844 within the project "Pro²Future". We also thank Felix Windisch for all the insightful discussions on Neural Rendering.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
2. Brachmann, E., Wynn, J., Chen, S., Cavallari, T., Monszpart, A., Turmukhambetov, D., Prisacariu, V.A.: Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In: European Conference on Computer Vision. pp. 421–440. Springer (2024)
3. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: Must3r: Multi-view network for stereo 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1050–1060 (2025)
4. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**(6), 679–698 (1986)
5. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
6. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10061–10072 (2023)
7. Duisterhof, B.P., Zust, L., Weinzaepfel, P., Leroy, V., Cabon, Y., Revaud, J.: Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In: 2025 International Conference on 3D Vision (3DV). pp. 1–10. IEEE (2025)
8. D'Urso, M., Santellani, E., Sormann, C., Rossi, M., Kuhn, A., Fraundorfer, F.: A streamlined attention-based network for descriptor extraction. In: 2026 International Conference on 3D Vision (3DV). pp. 247–256. IEEE Computer Society (2026)
9. D'Urso, M., Sormann, C., Rossi, M., Fraundorfer, F.: Multi-view reconstructions of european landmarks in 4k. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2026)
10. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint detection and description of local features. arXiv preprint arXiv:1905.03561 (2019)
11. Edstedt, J., Athanasiadis, I., Wadenbäck, M., Felsberg, M.: Dkm: Dense kernelized feature matching for geometry estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17765–17775 (2023)
12. Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M.: Roma: Robust dense feature matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19790–19800 (2024)
13. Elflein, S., Zhou, Q., Leal-Taixé, L.: Light3r-sfm: Towards feed-forward structure-from-motion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16774–16784 (2025)
14. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Theory of computing **8**(1), 415–428 (2012)
15. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: Tracking through occlusions using point trajectories. In: European Conference on Computer Vision. pp. 59–75. Springer (2022)

16. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
17. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6013–6022 (2025)
18. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. In: European conference on computer vision. pp. 18–35. Springer (2024)
19. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Rota Bulò, S., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction. In: 2026 International Conference on 3D Vision (3DV). pp. 499–509. IEEE Computer Society (2026)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
21. Kuse, M., Shen, S.: Robust camera motion estimation using direct edge alignment and sub-gradient method. In: 2016 IEEE international conference on robotics and automation (ICRA). pp. 573–579. IEEE (2016)
22. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024)
23. Li, J., Wang, H., Irshad, M.Z., Vasiljevic, I., Walter, M.R., Guizilini, V.C., Shakhnarovich, G.: FastMap: Revisiting structure from motion through first-order optimization. In: 2026 International Conference on 3D Vision (3DV). pp. 29–39. IEEE Computer Society (2026)
24. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Zhao, Y., Peng, S., Guo, H., Zhou, X., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=yirunib8l8, oral
25. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: Lightglue: Local feature matching at light speed. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 17627–17638 (2023)
26. Liu, S., Gao, Y., Zhang, T., Pautrat, R., Schönberger, J.L., Larsson, V., Pollefeys, M.: Robust incremental structure-from-motion with hybrid features. In: European Conference on Computer Vision. pp. 249–269. Springer (2024)
27. Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., Larsson, V.: 3d line mapping revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21445–21455 (2023)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations (2019)
29. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
30. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
31. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: Openmvg: Open multiple view geometry. In: International Workshop on Reproducible Research in Pattern Recognition. pp. 60–74. Springer (2016)

32. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
33. Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited. In: European Conference on Computer Vision. pp. 58–77. Springer (2024)
34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
35. Pautrat, R., Barath, D., Larsson, V., Oswald, M.R., Pollefeys, M.: Deeplsd: Line segment detection and refinement with deep image gradients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17327–17336 (2023)
36. Pautrat, R., Suárez, I., Yu, Y., Pollefeys, M., Larsson, V.: Gluestick: Robust image matching by sticking points and lines together. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9706–9716 (2023)
37. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021)
38. Revaud, J., De Souza, C., Humenberger, M., Weinzaepfel, P.: R2d2: Reliable and repeatable detector and descriptor. Advances in neural information processing systems **32** (2019)
39. Santellani, E., Sormann, C., Rossi, M., Kuhn, A., Fraundorfer, F.: Md-net: Multi-detector for local feature extraction. In: 2022 26th International conference on pattern recognition (ICPR). pp. 3944–3951. IEEE (2022)
40. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
41. Schenk, F., Fraundorfer, F.: Robust edge-based visual odometry using machine-learned edges. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1297–1304. IEEE (2017)
42. Schenk, F., Fraundorfer, F.: Reslam: A real-time robust edge-based slam system. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 154–160. IEEE (2019)
43. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
44. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. International journal of computer vision **80**(2), 189–210 (2008)
45. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
46. Sweeney, C., Hollerer, T., Turk, M.: Theia: A fast and scalable structure-from-motion library. In: Proceedings of the 23rd ACM international conference on Multimedia. pp. 693–696 (2015)
47. Tillet, P., Kung, H.T., Cox, D.: Triton: an intermediate language and compiler for tiled neural network computations. In: Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages. pp. 10–19 (2019)

48. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms. pp. 298–372. Springer (1999)
49. Tyszkiewicz, M., Fua, P., Trulls, E.: Disk: Learning local features with policy gradient. Advances in neural information processing systems **33**, 14254–14265 (2020)
50. Waechter, M., Beljan, M., Fuhrmann, S., Moehrle, N., Kopf, J., Goesele, M.: Virtual rephotography: Novel view prediction error for 3d reconstruction. ACM Transactions on Graphics (TOG) **36**(1), 1–11 (2017)
51. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 2025 International Conference on 3D Vision (3DV). pp. 78–89. IEEE (2025)
52. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
53. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Vggsfm: Visual geometry grounded deep structure from motion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21686–21697 (2024)
54. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025)
55. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024)
56. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=DTQIjngDta
57. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21924–21935 (2025)
58. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12–22 (2023)
59. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
60. Zhao, H., Shang, J., Liu, K., Chen, C., Gu, F.: Edgevo: An efficient and accurate edge-based visual odometry. arXiv preprint arXiv:2302.09493 (2023)
61. Zhao, X., Wu, X., Chen, W., Chen, P.C., Xu, Q., Li, Z.: Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. IEEE Transactions on Instrumentation and Measurement **72**, 1–16 (2023)
62. Zhong, J., Zhan, Z., Gao, Q., Chen, Z., Lou, H., Mao, J., Neumann, U., Wang, C., Wang, Y.: InstantSfM: Towards GPU-native SfM for the deep learning era. arXiv preprint arXiv:2510.13310 (2025)
63. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5745–5753 (2019)
64. Zhou, Y., Li, H., Kneip, L.: Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. IEEE Transactions on Robotics **35**(1), 184–199 (2018)