ODDEYE: BRIDGING PAIRWISE DENSE MATCHING AND INCREMENTAL STRUCTURE-FROM-MOTION

Anonymous authors
Paper under double-blind review



Figure 1: **OddEye** is an incremental SfM pipeline that efficiently integrates dense matchers via attention-guided scheduling. When registering a new candidate image (green frustum), it dynamically matches only against selected registered views (blue frustums) on demand.

ABSTRACT

While pairwise dense matchers offer dense coverage and robustness in low-texture scenes, their integration into incremental Structure-from-Motion (SfM) is hindered by two fundamental issues: fragmented multi-view tracks caused by independent pairwise sampling, and a high per-pair matching cost that makes exhaustive matching prohibitive. To overcome these challenges, we present **OddEye**, an incremental SfM framework that pairs off-the-shelf dense matchers with dynamic, reconstruction-aware scheduling. First, OddEye enforces consistent observation identities across views by anchoring queries to a fixed set of source coordinates per image, forming well-connected star-shaped multi-view tracks. Second, OddEye extracts a compact geometric prior from the cross-view attention of a feed-forward 3D model, and uses it to dynamically schedule dense matching only between candidate unregistered images and already registered partners. OddEye attains state-of-the-art results on ETH3D, IMC 2021, and Texture-Poor SfM, achieving 94.98% AUC@3° on ETH3D with RoMa v2. It also delivers a $9.53\times$ speedup on the largest ETH3D scene with negligible pose loss and scales to SMERF scenes of up to 1,874 images, registering 99.8% of images with under 10 GiB of peak GPU memory.

1 INTRODUCTION

Incremental Structure-from-Motion (SfM) reconstructs camera poses and 3D scene structure by registering images one at a time. Classical pipelines such as COLMAP (Schonberger & Frahm, 2016) detect sparse keypoints once per image and aggregate pairwise matches into a correspondence graph. Reusing keypoints across image pairs provides shared observations through which matches can form multi-view tracks. These tracks support camera registration and joint refinement of camera

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

poses and 3D points through bundle adjustment. The connectivity of the correspondence graph also guides initial-pair selection and next-view ordering.

Pairwise dense matchers such as RoMa (Edstedt et al., 2024) predict continuous correspondence fields, providing dense coverage and robustness in low-texture scenes. These fields are predicted independently for each image pair and do not define a shared set of source observations. When source coordinates are sampled independently for each pair, matches involving the same source image generally originate from different locations (Fig. 2, top). Without shared source observations, matches to different target images can remain disconnected, producing short tracks that provide limited constraints for triangulation and bundle adjustment. Existing dense SfM pipelines address this with additional stages: DetectorFreeSfM (He et al., 2024) builds tracks from quantized matches and DenseSfM (Lee & Yoo, 2025) extends them with Gaussian Splatting, and both then rely on trained multi-view refinement networks to correct the 2D observations within each track.

Dense pairwise matching is also far more expensive per image pair than sparse keypoint matching, and exhaustive matching over N images requires evaluating $\mathcal{O}(N^2)$ such pairs. To reduce this cost, global image retrieval methods such as NetVLAD (Arandjelovic et al., 2016) preselect a limited set of matching neighbors for each image before reconstruction. However, a neighbor suitable for pairwise matching is not necessarily useful for registration at the current reconstruction step. Incremental SfM registers each new image through 2D–3D correspondences obtained from already registered views, so matches to preselected neighbors that are still unregistered may not yet help. This motivates selecting matching partners from the currently registered views as reconstruction progresses. Without precomputed matches, however, there is no correspondence graph to guide initial-pair selection, next-view ordering, or partner choice, so these decisions require a co-visibility estimate available before matching.

We introduce **OddEye** (On-demand Dense Matching for Incremental SfM), an incremental SfM framework that addresses these challenges through *fixed source anchors* and *attention-guided on-demand matching*. OddEye samples a fixed set of 2D source anchors once per image and reuses them whenever that image serves as the source of a pairwise correspondence field. Verified matches from the same source anchor extend a shared, star-shaped track while preserving subpixel target coordinates. This construction integrates off-the-shelf dense matchers into incremental SfM without an additional learned track-refinement module.

OddEye reduces matching cost by selecting image pairs as reconstruction progresses. Before reconstruction begins, we extract a compact geometric prior from the cross-view attention of a feed-forward 3D model. This prior provides that co-visibility estimate: it selects the initial pair, ranks unregistered images by their estimated number of visible 3D tracks, and chooses registered matching partners for each candidate (Fig. 1). We run dense matching only on the selected pairs and use the verified correspondences to register cameras, extend tracks, and triangulate new points.

Experiments across benchmarks show that OddEye improves reconstruction across multiple dense matchers and achieves the highest pose AUC among the compared methods with RoMa v2. On ETH3D, OddEye outperforms DenseSfM by 17.55 percentage points in AUC@1° when both methods use RoMa. Our ablations show that fixed anchors produce longer multi-view tracks and improve camera registration. In fixed-camera triangulation, these tracks also support more complete point clouds with competitive geometric accuracy. On-demand matching improves registration over precomputed pairing, while attention-based ranking provides additional gains over appearance-based retrieval. Scheduling also reduces runtime: on the 110-image *bridge* scene, $k = 3$ gives a $9.53\times$ end-to-end speedup over exhaustive matching while losing only 0.37 points of AUC@3°. On SMERF scenes of up to 1,874 images, OddEye registers 99.8% of images within 10 GiB of GPU memory.

Our contributions are threefold. First, we introduce fixed-anchor track construction that connects continuous pairwise correspondence fields through shared source observations. Second, we introduce attention-guided on-demand scheduling that uses a compact feed-forward prior to select the initial pair, order candidate views, and choose registered matching partners during reconstruction. Third, we evaluate OddEye across multiple dense matchers and benchmarks, demonstrating improvements in pose accuracy, track connectivity, and point-cloud completeness, along with reduced runtime relative to exhaustive matching.

2 Related Work

Dense Correspondences for SfM. Pairwise dense matchers provide broad correspondence coverage (Edstedt et al., 2023; 2024; Leroy et al., 2024; Zhang et al., 2025; Edstedt et al., 2025). DetectorFreeSfM (He et al., 2024) initializes tracks from quantized matches and alternates learned multi-view matching with geometric refinement. DenseSfM (Lee & Yoo, 2025) uses Gaussian Splatting-guided track extension followed by learned multi-view refinement. MV-RoMa (Lee et al., 2026b) jointly estimates correspondences from a source image to multiple target views. OddEye connects pairwise predictions through fixed per-image source anchors, preserving subpixel target coordinates without an additional learned track-refinement module.

Learned Multi-View Reconstruction. Learned reconstruction systems recover scene geometry through pointmap alignment, learned tracking, and camera estimation (Wang et al., 2024b;a; Elflein et al., 2025; Liu et al., 2025). Feed-forward multi-view models predict geometry directly from image collections (Yang et al., 2025; Wang et al., 2025a;b; Lin et al., 2025; Burzio et al., 2026; Keetha et al., 2026). Hybrid methods such as MASt3R-SfM (Duisterhof et al., 2025), GLUEMAP (Pan et al., 2026), and Glob3R (Deng et al., 2026) combine learned geometric predictions with explicit alignment or geometric optimization. OddEye uses the model's cross-view attention to schedule dense matching during incremental reconstruction. Camera poses and 3D points are estimated from the verified dense correspondences.

Image-Pair Selection and Match Timing. SfM pipelines reduce matching cost through image retrieval, overlap estimation, geometric disambiguation, and view-graph selection (Arandjelovic et al., 2016; Yan et al., 2021; Shi et al., 2026; Xiangli et al., 2025). SARA (Lee et al., 2026a) uses lightweight pre-matching to estimate overlap and parallax and construct a sparse matching graph. EC-SfM (Ye et al., 2023) iteratively expands feature matches using a correspondence-derived covisibility graph and simulated registration. Foundation-model representations have also been used to assess relationships between views. Co-VGGT (Ziliotto et al., 2026) learns a covisibility predictor on frozen VGGT features, and Han et al. (2026) use internal representations for outlier-view rejection. OddEye extracts patch-level and image-pair scores from a calibrated attention head. The scheduler combines these scores with the current registered views and active tracks to decide which images to match as reconstruction progresses.

3 OddEye: On-Demand Dense Matching for Incremental SfM

Overview. OddEye integrates off-the-shelf pairwise dense matchers into incremental SfM (Fig. 3). We build multi-view tracks by querying dense correspondence fields at fixed per-image source anchors (Sec. 3.1). A compact cross-view attention prior from a feed-forward 3D model (Sec. 3.2) guides initial-pair selection, next-view ordering, and registered-partner selection during reconstruction (Sec. 3.3). We match the selected pairs on demand and estimate camera poses and 3D points from the verified correspondences.

3.1 Anchor-Based Track Construction

We form multi-view tracks by reusing source observations across image pairs (Fig. 2). For each image I_i, we sample N_a anchor coordinates uniformly at random to form $\mathcal{A}_i = \{\mathbf{x}_i^a\}_{a=1}^{N_a}$. We sample these coordinates once and reuse them whenever I_i serves as the source image.

For a requested image pair (I_i, I_j), we evaluate the predicted correspondence field $\Phi_{i\to j}$ and confidence map $\mathcal{C}_{i\to j}$ at each source anchor,

$$\left(\mathbf{x}_{i\to j}^a, c_{i\to j}^a\right) = \left(\Phi_{i\to j}(\mathbf{x}_i^a),\, \mathcal{C}_{i\to j}(\mathbf{x}_i^a)\right). \qquad (1)$$

Here, $\mathbf{x}_{i\to j}^a \in \mathbb{R}^2$ is the predicted subpixel target location in I_j, and $c_{i\to j}^a$ is its confidence. Target locations are continuous and need not coincide with anchors in $\mathcal{A}_j$.

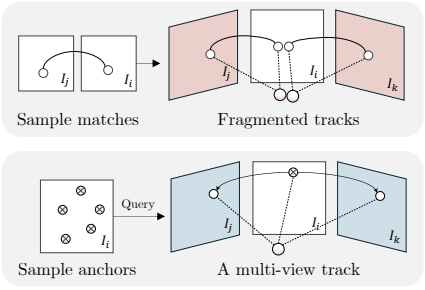


Figure 2: **Fixed anchors create multi-view tracks. Top**: Independent pairwise sampling produces fragmented two-view tracks. **Bottom**: Querying dense matches from the same source anchor across multiple target views forms a shared multi-view track.

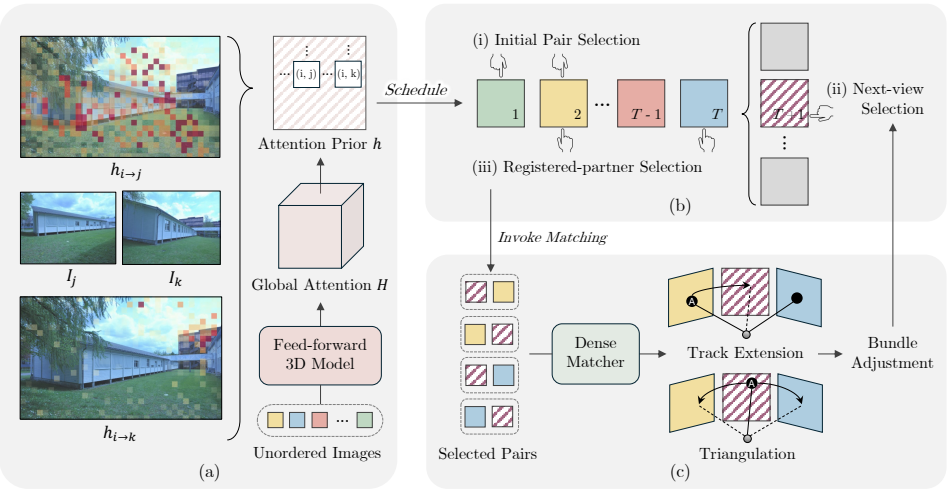


Figure 3: **Overview of OddEye.** (a) Cross-view attention from a single feed-forward 3D model pass provides a compact geometric prior h. (b) This prior guides initial-pair selection, next-view selection, and registered-partner selection inside the incremental loop. (c) Pairwise dense matching is invoked on demand for the selected pairs, querying fixed source anchors (marked as A) to support track extension and point triangulation.

We retain predictions that satisfy the confidence threshold $c^a_{i\to j} \geq \tau_c$ and forward–backward cycle consistency,

$$\left\|\Phi_{j\to i}(\mathbf{x}^a_{i\to j}) - \mathbf{x}^a_i\right\|_2 \leq \tau_{\text{cycle}}.$$

The remaining matches undergo two-view geometric verification using epipolar constraints.

Each verified target observation $\mathbf{x}^a_{i\to j}$ is linked to its source anchor $\mathbf{x}^a_i$. The first verified match initializes a two-view correspondence track. Verified matches from the same anchor to additional target images extend this track, forming a star-shaped structure rooted at a single source anchor.

3.2 Compact Geometric Prior from Cross-View Attention

Before reconstruction, we process all N input images jointly in a single forward pass through a feed-forward 3D model. We compress its cross-view attention into patch-level and image-pair scores for the scheduling policies in Sec. 3.3.

Selecting the Attention Head. Let $\mathcal{P}_i$ denote the set of patch tokens for image I_i. At layer ℓ and head h, we extract the attention block $\mathbf{H}^{(\ell,h)}_{i\to j} \in \mathbb{R}^{|\mathcal{P}_i|\times|\mathcal{P}_j|}$ from the model's global post-softmax attention. This block contains the attention responses from patches in I_i to patches in I_j. We use it directly, without per-target-image renormalization.

We calibrate head selection on held-out Tanks & Temples (Knapitsch et al., 2017) scenes using a pseudo-ground-truth correspondence set $\mathcal{Q} = \{(i, j, p, \bar{q})\}$. Each tuple associates a source patch $p \in \mathcal{P}_i$ with its corresponding target patch $\bar{q} \in \mathcal{P}_j$. These pairs are constructed from geometrically verified dense matches, as described in Appendix D.1. For each layer–head pair, we measure how often the target patch receiving the highest attention coincides with $\bar{q}$,

$$S^{(\ell,h)} = \frac{1}{|\mathcal{Q}|} \sum_{(i,j,p,\bar{q})\in\mathcal{Q}} \mathbb{I}\left[\arg\max_{q\in\mathcal{P}_j} \mathbf{H}^{(\ell,h)}_{i\to j}[p, q] = \bar{q}\right]. \tag{2}$$

We select $(\ell^\star, h^\star) = \arg\max_{\ell,h} S^{(\ell,h)}$ once and keep this layer–head pair fixed across all evaluation scenes.

Building the Geometric Prior. To estimate whether a source patch has a valid correspondence in a target view, we summarize its cross-view attention with a single score. For each source patch p,

we retain the maximum response over the target patches,

$$h_{i\to j}(p) = \max_{q\in\mathcal{P}_j} \mathbf{H}_{i\to j}^{(\ell^\star,h^\star)}[p,q], \qquad p \in \mathcal{P}_i. \tag{3}$$

Averaging these scores over the source patches gives the directed image-pair score,

$$\bar{h}_{i\to j} = \frac{1}{|\mathcal{P}_i|} \sum_{p\in\mathcal{P}_i} h_{i\to j}(p). \tag{4}$$

The scheduler uses the patch scores to estimate active-track visibility and the image-pair scores to guide initial-pair and registered-partner selection (Sec. 3.3).

This compression represents each ordered image pair with a vector of source-patch scores. For N images with P patch tokens each, it reduces prior storage from $\mathcal{O}(N^2P^2)$ to $\mathcal{O}(N^2P)$. With $N = 100$ and $P = 768$, FP32 storage over ordered cross-image pairs decreases from approximately 23.4 GB to 30.4 MB.

3.3 On-Demand Matching During Reconstruction

During reconstruction, we maintain the registered views $\mathcal{R}$, unregistered views $\mathcal{U}$, and active 3D tracks $\mathcal{T}$. The scheduler uses these sets together with the attention prior from Sec. 3.2 to choose which images to match. We update the reconstruction state after each successful registration.

Initial-Pair Selection. Before initialization, $\mathcal{R} = \emptyset$ and $\mathcal{T} = \emptyset$. We compute symmetric image-pair scores w_{ij} and their per-image sums d_i,

$$w_{ij} = \sqrt{\bar{h}_{i\to j}\bar{h}_{j\to i}}, \qquad d_i = \sum_{j\neq i} w_{ij}. \tag{5}$$

We consider starting images in decreasing order of d_i. For each starting image I_i, we consider partners I_j in decreasing order of w_{ij}. We run the dense matcher for each candidate pair, query correspondences at the fixed anchors, and apply the verification procedure in Sec. 3.1. The first pair passing verification initializes the reconstruction. If verification fails, we proceed to the next pair.

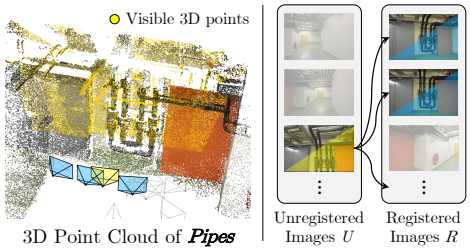


Figure 4: **On-demand scheduling.** Given the candidate view (yellow), top-k registered partners (blue) are selected for on-demand dense matching using the attention prior.

Next-View Selection. To choose the next view, we estimate how many active 3D tracks are visible in each unregistered image. Each track $X_t \in \mathcal{T}$ is rooted at a source anchor $\mathbf{x}_{r_t}^{a_t}$ in a registered image $I_{r_t} \in \mathcal{R}$. Let $p_t \in \mathcal{P}_{r_t}$ denote the patch containing this anchor. For a candidate $I_c \in \mathcal{U}$, we use $h_{r_t\to c}(p_t)$ to estimate the track's visibility. We count tracks whose scores meet the visibility threshold τ_v,

$$v(I_c) = \sum_{X_t\in\mathcal{T}} \mathbb{I}\left[h_{r_t\to c}(p_t) \geq \tau_v\right]. \tag{6}$$

We rank unregistered views in decreasing order of $v(I_c)$ and attempt registration in this order, starting with $I_{c^\star} = \arg\max_{I_c\in\mathcal{U}} v(I_c)$.

Registered-Partner Selection. For candidate $I_{c^\star}$, we rank registered views by the directed image-pair score $\bar{h}_{c^\star\to r}$ and select the top k partners,

$$\mathcal{R}_{c^\star} = \mathrm{TopK}_{I_r\in\mathcal{R}}\left(\bar{h}_{c^\star\to r}\right). \tag{7}$$

We run dense matching only between the candidate and these selected partners, obtaining correspondence fields in both directions (Fig. 4).

Reconstruction Update. We use the two matching directions to register the candidate, extend existing tracks, and triangulate new points.

- *Registered-to-candidate matching ($\Phi_{r\to c^\star}$).* For each partner $I_r \in \mathcal{R}_{c^\star}$, we query $\Phi_{r\to c^\star}$ at the fixed anchors $\mathcal{A}_r$. When an anchor belongs to an active 3D track, its verified match in $I_{c^\star}$ provides a 2D–3D correspondence. We aggregate these correspondences across the selected partners to estimate the candidate pose with PnP and extend the corresponding tracks.
- *Candidate-to-registered matching ($\Phi_{c^\star\to r}$).* We query $\Phi_{c^\star\to r}$ at the candidate anchors $\mathcal{A}_{c^\star}$ for each selected partner. Verified matches from the same candidate anchor form a new track rooted at that anchor, with target observations in $\mathcal{R}_{c^\star}$. Once the candidate pose has been estimated, we triangulate new 3D points from these tracks.

If PnP fails, we proceed to the next candidate in the current ranking. After a successful reconstruction update and local bundle adjustment, we move $I_{c^\star}$ from $\mathcal{U}$ to $\mathcal{R}$. We then recompute the ranking over all remaining unregistered views, including previously unsuccessful candidates. Reconstruction stops when $\mathcal{U}$ is empty or all candidates in the current ranking fail to register.

4 Experiments

Overview. We first evaluate multi-view camera-pose estimation and fixed-camera triangulation in Secs. 4.1 and 4.2. Next, we analyze anchor-based track construction, attention calibration, and on-demand scheduling in Secs. 4.3–4.5. Finally, Sec. 4.6 assesses runtime and scalability.

Experimental Setup. We evaluate camera-pose estimation on 22 ETH3D (Schops et al., 2017) scenes, IMC 2021 (Jin et al., 2020) with 5-, 10-, and 25-image subsets, and Texture-Poor SfM (He et al., 2024) with 5-, 10-, and 20-image subsets. We report pose AUC based on the maximum of rotation and translation angular errors, with thresholds listed in Table 1. For ETH3D and IMC 2021, focal lengths are initialized to COLMAP defaults and refined during bundle adjustment, whereas intrinsics are fixed to ground-truth values on Texture-Poor SfM following prior work (Lee & Yoo, 2025). We calibrate the attention prior on held-out Tanks & Temples (Knapitsch et al., 2017) scenes (Appendix D.1). Unless otherwise stated, we use DVLT (Burzio et al., 2026) as the attention backbone and select $k = 5$ registered partners per step. Additional evaluation settings and implementation details are provided in Appendix A.

Table 1: **Multi-view camera reconstruction.** Highlighting indicates **first-**, second-, and third-best results. †marks values reported by prior work; ‡marks values obtained with our implementation. For MV-RoMa, ETH3D results are obtained with our implementation, while others are reported.

		ETH3D			IMC 2021			Texture-Poor SfM		
Matcher	**Method**	@1°	@3°	@5°	@3°	@5°	@10°	@3°	@5°	@10°
Detector-based										
SIFT+NN	COLMAP†	26.71	38.86	42.14	23.58	32.66	44.79	2.87	3.85	4.95
SP+SG	PixSfM†	50.82	68.52	72.86	45.19	57.22	70.47	14.00	19.23	24.55
End-to-end										
	VGGSfM†	28.09	44.41	49.91	45.23	58.89	73.92	1.86	6.26	21.10
	MASt3R-SfM†	27.05	45.84	52.61	31.77	46.36	64.37	7.10	18.08	42.44
	MV-RoMa‡/†	37.43	43.29	44.85	51.31	62.92	75.92	51.79	66.77	81.74
(Semi-)dense matching										
DKM	DetectorFreeSfM‡	54.55	72.66	77.55	48.31	60.80	74.38	37.12	51.80	66.55
	DenseSfM†	59.04	77.73	82.20	48.65	61.09	74.37	41.19	57.26	72.95
	OddEye (Ours)‡	74.73	83.44	85.43	51.29	63.07	75.66	45.71	60.32	74.07
RoMa	DetectorFreeSfM‡	57.35	75.45	80.07	48.60	61.06	74.57	48.71	64.55	79.21
	DenseSfM†	60.92	78.41	82.63	48.48	60.79	73.90	49.94	66.23	81.41
	OddEye (Ours)‡	78.47	87.27	89.43	52.70	64.52	77.01	54.25	67.98	80.27
RoMa v2	DetectorFreeSfM‡	62.07	80.11	84.59	49.22	61.81	75.32	58.16	73.31	86.08
	OddEye (Ours)‡	**86.16**	**94.98**	**96.93**	**53.42**	**65.18**	**77.53**	**63.21**	**76.47**	**87.53**

4.1 Multi-View Reconstruction

We compare OddEye with detector-based pipelines COLMAP (Schonberger & Frahm, 2016) and PixSfM (Lindenberger et al., 2021), end-to-end models VGGSfM (Wang et al., 2024a), MASt3R-SfM (Duisterhof et al., 2025), and MV-RoMa (Lee et al., 2026b), and (semi-)dense matching baselines DetectorFreeSfM (He et al., 2024) and DenseSfM (Lee & Yoo, 2025). We evaluate OddEye with DKM (Edstedt et al., 2023), RoMa (Edstedt et al., 2024), and RoMa v2 (Edstedt et al., 2025) as the pairwise dense matcher.

Table 1 shows that OddEye consistently improves pose accuracy over DetectorFreeSfM across all benchmarks and dense matchers. On ETH3D, OddEye outperforms DenseSfM by 15.69 and 17.55 percentage points in AUC@1° with DKM and RoMa, respectively, while substantially surpassing detector-based and end-to-end baselines. Switching to RoMa v2 achieves the best results across all datasets and thresholds, including the challenging Texture-Poor SfM benchmark. While MV-RoMa trains a multi-view track prediction model, OddEye directly benefits from stronger off-the-shelf pairwise matchers without architectural changes or retraining.

4.2 3D Triangulation

Following prior work (He et al., 2024), we evaluate triangulation on the 13 ETH3D training scenes with fixed ground-truth camera poses and intrinsics. We use exhaustive image pairing and sample $N_a = 32{,}768$ source anchors per image. Table 2 reports point-cloud accuracy and completeness at distance thresholds of 1, 2, and 5 cm.

Table 2: **Fixed-camera 3D triangulation on ETH3D.** All results exclude two-view tracks.

Matcher	Method	Accuracy (%) ↑			Completeness (%) ↑		
		1 cm	2 cm	5 cm	1 cm	2 cm	5 cm
–	MV-RoMa†	85.88	92.99	98.05	3.95	9.94	23.81
DKM	**OddEye** (Ours)‡	84.51	91.84	97.11	9.46	21.41	43.23
RoMa	DetectorFreeSfM‡	79.32	88.42	95.82	3.13	9.79	29.10
	DenseSfM†	84.79	92.62	97.77	7.38	17.06	36.35
	OddEye (Ours)‡	84.34	91.76	97.28	9.55	22.04	43.86
RoMa v2	DetectorFreeSfM‡	77.02	86.92	95.01	4.30	14.04	40.73
	OddEye (Ours)‡	**88.49**	**94.51**	**98.37**	**21.77**	**43.66**	**72.25**

With RoMa, OddEye outperforms DetectorFreeSfM in accuracy and completeness, and generates denser point clouds than DenseSfM with comparable accuracy. While MV-RoMa achieves strong accuracy, it suffers from low completeness (3.95% at 1 cm). In contrast, OddEye consistently delivers high completeness across all matchers (DKM, RoMa, and RoMa v2). Paired with RoMa v2, it achieves the best overall performance, leading across all thresholds and more than doubling completeness at 1 cm (21.77%).

4.3 Fixed Anchors Improve Multi-View Track Construction

We evaluate source sampling with RoMa v2 on ETH3D using a precomputed pair graph from our attention-based ranking to isolate track construction from on-demand scheduling. We compare three source sampling strategies: (i) *Independent*, which samples source locations independently for each image pair; (ii) *Quantized* (16 px), which rounds sampled coordinates to a 16-pixel grid before querying correspondence fields; and (iii) *Anchor-based* (Ours), which defines a fixed set of anchor coordinates per image and reuses them whenever that image acts as the source view.

Table 3: **Effect of anchor-based sampling**

Sampling method	Track length ↑	Tracks ≥ 3 views (%) ↑	AUC@3° (%) ↑	Reg. (%) ↑
Independent	2.00	0.0	0.35	5.6
+ Quantized (16 px)	2.05	5.0	43.16	55.1
Anchor-based (Ours)	**4.98**	**91.1**	**94.89**	**100.0**

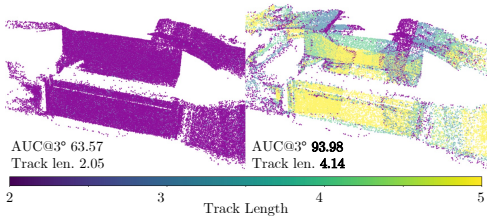


Figure 5: **Track length visualization.** *Lecture room* in ETH3D. **Left:** Quantized grid (16 px). **Right:** Fixed anchors.

As shown in Table 3, independent sampling yields almost exclusively two-view tracks (0.0% with ≥ 3 views) and causes catastrophic registration failure (5.6% registered). While coarse spatial quantization restores basic registration (55.1%), it still leaves 95% of tracks fragmented. In contrast, our anchor-based sampling directly enforces multi-view track consistency, raising multi-view tracks to 91.1% and average track length to 4.98 (Fig. 5). This connectivity translates directly to superior pose accuracy (94.89% AUC@3°) and registration rate (100.0%).

4.4 Calibrating Cross-View Attention

We compare VGGT (Wang et al., 2025a), π^3 (Wang et al., 2025b), Depth Anything 3 (Lin et al., 2025), and DVLT (Burzio et al., 2026) using predicted camera-pose accuracy on held-out Tanks & Temples scenes. DVLT achieves the highest scene-averaged pose AUC on the calibration set, so we use it as the default attention backbone.

To select specific attention heads without running full reconstructions for every candidate, we evaluate DVLT's layers and heads using top-1 patch-correspondence accuracy (Eq. 2) against pseudo-ground-truth correspondences. We select $(\ell^\star, h^\star) = (11, 0)$, which achieves the highest correspondence accuracy (31.7%), and fix it across all evaluation scenes. Figure 6 illustrates the differences between heads. While some produce diffuse attention or confuse visually similar whiteboards, the selected head accurately isolates the geometrically consistent region in both target views. Appendix D.1 details the calibration protocol, and Appendix D.2 compares all backbones, including downstream reconstruction.

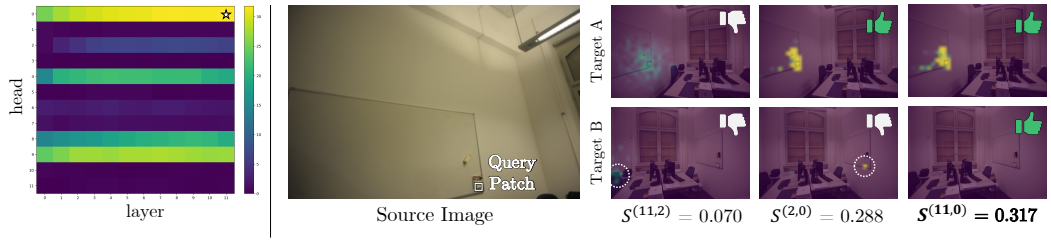


Figure 6: **DVLT attention calibration. Left:** Top-1 patch-correspondence accuracy across layers ℓ and heads h. We select $(\ell^\star, h^\star) = (11, 0)$, yielding the peak accuracy of 31.7%. **Right:** Attention maps for query patches. While $(11, 2)$ produces spatially diffuse attention and $(2, 0)$ confuses visually similar whiteboards, $(11, 0)$ accurately isolates the true geometric match across both views.

4.5 Ablating On-Demand Scheduling

We evaluate next-view selection and registered-partner selection on ETH3D using RoMa as the pairwise dense matcher.

Next-View Selection. We evaluate next-view ranking policies using $k = 5$ registered partners per step (Table 4). As baselines, we rank candidate views by maximum cosine similarity to registered views using global features from DINOv2 (Oquab et al., 2023) or NetVLAD (Arandjelovic et al., 2016). In contrast, our policy ranks views by the estimated visibility of active 3D tracks (Eq. 6). This track-visibility policy achieves the highest pose AUC (87.27%) and PnP success rate (87.86%), while matching NetVLAD's registration rate (97.14%).

Table 4: **Next-view selection.**

View ranking	AUC@3° (%) ↑	PnP succ. (%) ↑	Reg. (%) ↑
DINOv2	82.06	78.13	96.79
NetVLAD	81.78	86.55	**97.14**
Attention	**87.27**	**87.86**	**97.14**

Registered-Partner Selection. We compare pair-ranking criteria under precomputed and on-demand matching using a tighter budget of $k = 3$ (Table 5). Precomputed matching fixes partners beforehand, whereas on-demand matching selects partners dynamically from registered views. Retrieval baselines rank pairs by cosine similarity using global features, while our policy scores pairs using cross-view attention (Eq. 4).

Table 5: **Registered-partner selection.**

Match timing	Pair ranking	AUC@3° (%) ↑	Reg. (%) ↑
Precomputed	DINOv2	43.41	69.15
	NetVLAD	56.99	73.02
	Attention	76.91	89.03
On-demand	DINOv2	68.87	89.26
	NetVLAD	81.47	96.03
	Attention	**89.07**	**97.14**

As shown in Table 5, attention-based ranking outperforms appearance baselines under both schedules, lifting precomputed AUC by 19.92 percentage points over NetVLAD. On-demand matching further boosts performance across all criteria, reaching 89.07% AUC and 97.14% registration with our attention policy. Figure 7 shows how cross-view attention correctly downweights visually similar, non-overlapping pairs that deceive appearance retrieval. Appendix D.4 reports the corresponding ablation for initial-pair selection.



Figure 7: **Ranking visually similar but non-overlapping pairs.** Each query (left) is paired with a candidate (right) that has no geometric overlap with it. Labels give the candidate's rank among all other images (#1 = most similar), so larger is correct here. DINOv2 ranks every candidate in its top three, while the attention prior (w_{ij}) ranks them last or near last.

4.6 Runtime and Scalability

Runtime Comparison. We evaluate on-demand OddEye ($k \in \{2, 3, 5\}$) against exhaustive matching on the 110-image ETH3D *bridge* scene using RoMa v2 on an NVIDIA RTX 4090 GPU (24 GB) and an AMD Ryzen 9 7950X CPU. As shown in Fig. 8(a), setting $k = 2$ achieves a 12.78× end-to-end speedup, reducing total runtime from 88.7 to 6.9 minutes while incurring only a 0.82 percentage-point decrease in AUC@3° (96.40% vs. 97.22%). Figure 8(b) shows that dense matching accounts for the vast majority of time savings, while computing the attention prior requires 0.58 minutes.

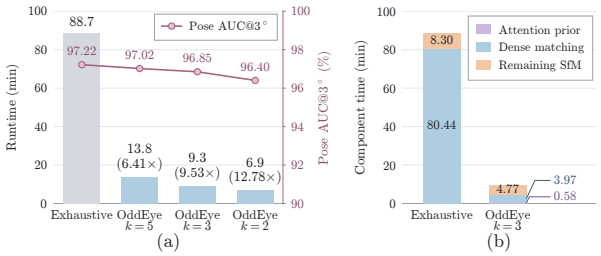


Figure 8: **Runtime analysis on ETH3D** *bridge*. (a) Total runtime and Pose AUC@3° across different values of k, compared with exhaustive matching. (b) Runtime breakdown for exhaustive matching and OddEye ($k = 3$).

Large-Scale Reconstruction on SMERF. To demonstrate scalability to thousands of images, we evaluate OddEye on four SMERF (Duckworth et al., 2024) scenes with 990 to 1,874 images each, using the published intrinsics and $N_a = 1{,}024$. Across all scenes, OddEye registers 99.8% of 6,109 images with 92.85–95.33% AUC@3° and under 10 GiB of peak GPU memory. On the largest scene, *London* (Fig. 9), it registers 1,865 of 1,874 images with 93.92% AUC@3° using 9.33 GiB. Per-scene results and the techniques for keeping peak memory low are detailed in Appendix E.

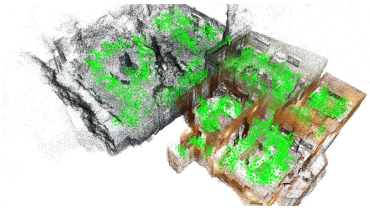


Figure 9: **Reconstruction of SMERF** *London* (1,874 images).

5 Conclusion

We presented **OddEye**, which integrates off-the-shelf pairwise dense matchers into incremental SfM without retraining or learned track refinement. Fixed per-image source anchors connect independent pairwise predictions into multi-view tracks. Because dense matching is expensive per image pair, a compact prior from the cross-view attention of a feed-forward 3D model decides which pairs to match as reconstruction progresses, while all camera poses and 3D points come from verified correspondences. OddEye achieves the highest pose accuracy among compared methods on ETH3D, IMC 2021, and Texture-Poor SfM, reduces runtime by an order of magnitude over exhaustive matching on the largest ETH3D scene with negligible pose loss, and scales to SMERF scenes of up to 1,874 images. Future work includes cheaper prior extraction and partner selection for densely captured sequences.

AI USE STATEMENT

Large Language Models (LLMs) were used solely as general assistance tools during the preparation of this manuscript. Specifically, LLMs assisted with text polishing, code implementation, refactoring, and drafting visualization scripts. All core ideas, theoretical formulations, experimental designs, and analyses were entirely developed, executed, and verified by the authors, who take full responsibility for the paper's contents and factual accuracy.

REFERENCES

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

Alessandro Burzio, Tobias Fischer, Sven Elflein, Qunjie Zhou, Riccardo de Lutio, Jiawei Ren, Jiahui Huang, Shengyu Huang, Marc Pollefeys, Laura Leal-Taixé, Zan Gojcic, and Haithem Turki. Déjà View: Looping transformers for multi-view 3D reconstruction. *arXiv preprint arXiv:2605.30215*, 2026.

Junyuan Deng, Heng Li, Kejie Qiu, Lingteng Qiu, Rui Peng, Weichao Shen, Weihao Yuan, Siyu Zhu, Zilong Dong, and Ping Tan. Glob3r: Global structure-from-motion with 3d foundation models. *arXiv preprint arXiv:2607.09225*, 2026.

Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lučić, Richard Szeliski, and Jonathan T Barron. Smerf: Streamable memory efficient radiance fields for real-time large-scene exploration. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.

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

Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. 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.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19790–19800, 2024.

Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. Roma v2: Harder better faster denser feature matching. *arXiv preprint arXiv:2511.15706*, 2025.

Sven Elflein, Qunjie Zhou, and Laura Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16774–16784. IEEE, 2025.

Jisang Han, Sunghwan Hong, Jaewoo Jung, Wooseok Jang, Honggyu An, Qianqian Wang, Seungryong Kim, and Chen Feng. Emergent outlier view rejection in visual geometry grounded transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 427–437, 2026.

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/CVF conference on computer vision and pattern recognition*, pp. 21594–21603, 2024.

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*, 2020.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

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

Jee Won Lee, Hansol Lim, Minhyeok Im, Dohyeon Lee, and Jongseong Brad Choi. Sara: Scene-aware reconstruction accelerator. *arXiv preprint arXiv:2601.06831*, 2026a.

JongMin Lee and Sungjoo Yoo. Dense-sfm: Structure from motion with dense consistent matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6404–6414, 2025.

Jongmin Lee, Seungyeop Kang, and Sungjoo Yoo. Mv-roma: From pairwise matching into multi-view track reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7446–7456, 2026b.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

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

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

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

Sidun Liu, Wenyu Li, Peng Qiao, and Yong Dou. Regist3r: Incremental registration with stereo foundation model. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 4484–4493, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Linfei Pan, Johannes Schönberger, and Marc Pollefeys. Global structure-from-motion meets feed-forward reconstruction. *arXiv preprint arXiv:2605.26103*, 2026.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

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*, pp. 3260–3269, 2017.

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

Xulei Shi, Maoyu Wang, Yuning Peng, Guanbo Wang, Xin Wang, Yifan Liao, Qi Chen, and Pengjie Tao. Supscene: Scene-structured overlap supervision for image retrieval in unconstrained sfm. *arXiv preprint arXiv:2601.11930*, 2026.

Xianbing Sun, Zhikai Zhu, Zhengyu Lou, Bo Yang, Jinyang Tang, Liqing Zhang, He Wang, and Jianfu Zhang. Avggt: Rethinking global attention for accelerating vggt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 251–260, 2026.

Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Block-sparse global attention for efficient multi-view geometry transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14546–14555, 2026a.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024b.

Weitian Wang, Lukas Meiner, Rai Shubham, Cecilia De La Parra, and Akash Kumar. Httm: Head-wise temporal token merging for faster vggt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26379–26388, 2026b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

Yuanbo Xiangli, Ruojin Cai, Hanyu Chen, Jeffrey Byrne, and Noah Snavely. Doppelgangers++: Improved visual disambiguation with geometric 3d features. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27166–27175. IEEE, 2025.

Shen Yan, Maojun Zhang, Shiming Lai, Yu Liu, and Yang Peng. Image retrieval for structure-from-motion via graph convolutional network. *Information Sciences*, 573:20–36, 2021.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025.

Zhichao Ye, Chong Bao, Xin Zhou, Haomin Liu, Hujun Bao, and Guofeng Zhang. Ec-sfm: Efficient covisibility-based structure-from-motion for both sequential and unordered images. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(1):110–123, 2023.

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

Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023.

Filippo Ziliotto, Luciano Serafini, Lamberto Ballan, and Tommaso Campari. What vggt knows about overlap: Probing geometric foundation models for co-visibility. *arXiv preprint arXiv:2607.09503*, 2026.

A Reproducibility and Experimental Details

A.1 Evaluation Protocols and Hyperparameters

Benchmarks and Validation Splits. We evaluate OddEye on ETH3D (Schops et al., 2017), IMC 2021 (Jin et al., 2020), and Texture-Poor SfM (He et al., 2024). Following the evaluation protocol

of DetectorFreeSfM (He et al., 2024), we reserve *observatory*, *relief*, and *relief_2* from ETH3D, as well as scans 1012, 1026, and 1029 from Texture-Poor SfM, as validation scenes. Attention-head calibration is conducted on seven held-out scenes from the Tanks & Temples (Knapitsch et al., 2017) training split (Appendix D.1).

Image Resolution and Baseline Settings. Across all benchmarks, OddEye passes the input images directly to the dense matcher, which runs its internal coarse and refinement stages at its default resolutions. On Texture-Poor SfM, DetectorFreeSfM (He et al., 2024) and DenseSfM (Lee & Yoo, 2025) instead downsample the 840×840 input images to 420×420 in their initial matching stage and return to full resolution in their refinement stage. For a consistent comparison, we re-evaluate DetectorFreeSfM on the original 840×840 images, so that its matcher receives the same inputs as OddEye's. For DenseSfM, we report the published results, since only its pretrained track refiner, not the full pipeline, is publicly available.

Hyperparameter Configuration. Table 6 summarizes the dataset splits, evaluation thresholds, and default hyperparameters. Across all experiments, we adopt DVLT (Burzio et al., 2026) as our default feed-forward backbone, where "layer ℓ" denotes the ℓ-th recurrent iteration of the looped transformer. We fix the number of registered partners to $k = 5$ and the visibility threshold to $\tau_v = 0.05$ across all datasets. For Texture-Poor SfM, we scale the anchor budget to $N_a = 16{,}384$ and adjust the confidence threshold to $\tau_c = 0.3$ to better accommodate weakly textured regions.

Table 6: **Benchmark evaluation settings and default hyperparameters.**

Configuration / Metric	ETH3D	IMC 2021	Texture-Poor SfM
Evaluation split	22 scenes	5-, 10-, 25-image bags	5-, 10-, 20-image bags
AUC thresholds	$1^\circ, 3^\circ, 5^\circ$	$3^\circ, 5^\circ, 10^\circ$	$3^\circ, 5^\circ, 10^\circ$
Focal-length initialization	$1.2\max(H, W)$	$1.2\max(H, W)$	Ground truth
Anchors (N_a)	4,096	4,096	16,384
Registered partners (k)	5	5	5
Confidence (τ_c)	0.5	0.5	0.3
Cycle consistency (τ_{cycle})	3.0 px	3.0 px	3.0 px
Visibility (τ_v)	0.05	0.05	0.05

A.2 Code Implementation

OddEye is implemented in Python and interfaces directly with COLMAP (Schonberger & Frahm, 2016) via PyCOLMAP. The Python module handles high-level scheduling, including extracting the feed-forward attention prior, selecting candidate views, and querying dense matches at fixed anchors on demand. By exposing COLMAP's incremental loop to Python, OddEye dynamically introduces verified matches as reconstruction progresses, while delegating two-view verification, PnP localization, triangulation, and Ceres-based bundle adjustment entirely to COLMAP's compiled C++ backend.

A.3 Camera-Intrinsics Initialization

As noted in Section 4, our main benchmark evaluations use COLMAP's heuristic initialization $f_0 = 1.2\max(H, W)$ for uncalibrated cameras rather than focal lengths predicted by the feed-forward model. We intentionally adhere to this protocol to ensure a fair comparison against baselines that lack learned intrinsics priors, even though our framework could readily leverage the feed-forward model's focal length predictions.

Table 7: **Effect of focal-length initialization.**

	ETH3D			IMC 2021		
Initialization	@1°	@3°	@5°	@3°	@5°	@10°
Default ($1.2\max(H, W)$)	**86.16**	**94.98**	**96.93**	53.42	65.18	77.53
DVLT prediction	85.76	94.82	96.83	**56.89**	**68.45**	**80.20**

Table 7 analyzes the effect of initializing uncalibrated views with focal lengths predicted by DVLT (Burzio et al., 2026) using RoMa v2. On IMC 2021, it improves AUC@3° by 3.47 pp. IMC 2021 bags consist of internet photos taken with different cameras, so focal lengths vary across images, and all images in a bag view the same side of a landmark. We attribute the gain to this

setting: with few images and a limited range of viewing directions, bundle adjustment has fewer constraints for recovering each image's focal length, so a good initialization matters more. On ETH3D, DVLT initialization performs on par with the default, suggesting that bundle adjustment already recovers accurate intrinsics there.

B Extended Reconstruction Results

This section complements the camera-pose and 3D triangulation evaluations in Sections 4.1 and 4.2. We first report scene-wise camera-pose results and compare OddEye with direct feed-forward reconstruction systems. We then provide qualitative reconstructions and detailed scene-wise triangulation results.

B.1 Per-Scene Camera-Pose Results on ETH3D

Table 8 details per-scene camera-pose accuracy (Pose AUC@1°) across all 22 ETH3D evaluation scenes. In addition to the incremental baselines from Table 1, we include GLOMAP (Pan et al., 2024) (ALIKED (Zhao et al., 2023) + LightGlue (Lindenberger et al., 2023)) and the hybrid global pipeline GLUEMAP (Pan et al., 2026), with baseline numbers taken from Pan et al. (2026). While GLUEMAP directly incorporates predicted feed-forward 3D geometry into global motion averaging, OddEye uses the feed-forward model only as an attention prior to schedule pairwise dense matching, optimizing poses entirely from verified correspondences. In the table, † marks values reported by prior work and ‡ denotes results reproduced using our implementation. OddEye achieves the highest pose accuracy on 20 of the 22 evaluation scenes, demonstrating superior robustness across diverse indoor and outdoor environments compared to both hybrid global and dense incremental baselines.

Table 8: **Per-scene camera-pose AUC@1° on ETH3D.** DetectorFreeSfM and OddEye use RoMa v2.

Scene	door	lounge	statue	terrace 2	pipes	meadow	lecture room	terrace	boulders	office	botanical garden
MV-RoMa‡	84.43	27.13	95.90	25.37	58.32	0.00	66.50	86.48	82.93	18.90	24.49
DetectorFreeSfM‡	**86.18**	27.89	66.25	**84.82**	63.64	61.70	67.35	63.15	69.55	31.27	61.27
GLOMAP†	70.70	28.10	64.40	82.40	52.60	33.10	58.30	47.70	4.00	21.90	59.40
GLUEMAP†	80.20	37.10	66.90	83.70	52.90	43.20	55.70	60.50	47.60	24.70	59.30
OddEye (Ours)‡	84.15	**87.56**	**96.40**	84.76	**91.66**	**81.11**	**85.02**	**91.49**	**88.29**	**59.70**	**87.13**

Scene	kicker	courtyard	playground	terrains	delivery area	electro	old computer	living room	exhibition hall	facade	bridge
MV-RoMa‡	66.25	13.88	12.53	23.16	4.99	30.59	4.49	25.62	7.71	62.39	1.47
DetectorFreeSfM‡	68.86	59.79	74.44	67.46	66.94	57.41	33.29	75.32	27.29	74.10	77.52
GLOMAP†	55.20	45.90	55.40	26.30	47.80	44.40	31.00	50.80	8.70	68.80	73.20
GLUEMAP†	56.60	52.80	68.00	33.60	43.90	50.80	29.60	66.20	26.90	70.40	74.40
OddEye (Ours)‡	**90.75**	**90.87**	**86.87**	**80.30**	**88.27**	**89.79**	**80.97**	**88.36**	**78.90**	**91.94**	**91.17**

Table 9: **Camera-pose reconstruction compared with direct feed-forward 3D models.**

	ETH3D			IMC 2021			Texture-Poor SfM		
Method	@1°	@3°	@5°	@3°	@5°	@10°	@3°	@5°	@10°
VGGT-1B (Wang et al., 2025a)	8.03	28.63	41.57	41.90	56.18	72.32	3.34	11.11	35.23
π^3 (Wang et al., 2025b)	6.07	28.84	43.49	40.72	55.95	72.54	2.69	9.85	32.02
DVLT (Burzio et al., 2026)	17.29	50.84	65.02	31.18	46.14	64.72	2.79	10.04	33.05
DA3-Large (Lin et al., 2025)	9.27	35.07	49.82	28.55	43.51	62.56	1.88	7.30	27.14
OddEye (Ours)	**86.16**	**94.98**	**96.93**	**53.42**	**65.18**	**77.53**	**63.21**	**76.47**	**87.53**

B.2 Comparison with Direct Feed-Forward 3D Reconstruction

Table 9 compares OddEye with direct pose predictions from feed-forward 3D foundation models across ETH3D, IMC 2021, and Texture-Poor SfM. While these feed-forward baselines estimate multi-view geometry directly, OddEye uses cross-view attention only to schedule pairwise matching, resolving camera poses through incremental geometry and bundle adjustment. Across all evaluated benchmarks and thresholds, the direct feed-forward predictions remain less accurate than OddEye.

These results support using the feed-forward representation as a scheduling prior rather than adopting its predicted camera poses as the final reconstruction.

B.3 Qualitative Reconstruction Results on Texture-Poor SfM and IMC 2021

Figures 10 and 11 provide qualitative comparisons with DetectorFreeSfM using the same RoMa v2 matcher.



Figure 10: **Qualitative reconstruction comparison on Texture-Poor SfM.** Six 10-image subsets are shown. Blue frustums represent predicted cameras, and red frustums denote ground truth. **Left**: OddEye + RoMa v2. **Right**: DetectorFreeSfM + RoMa v2.

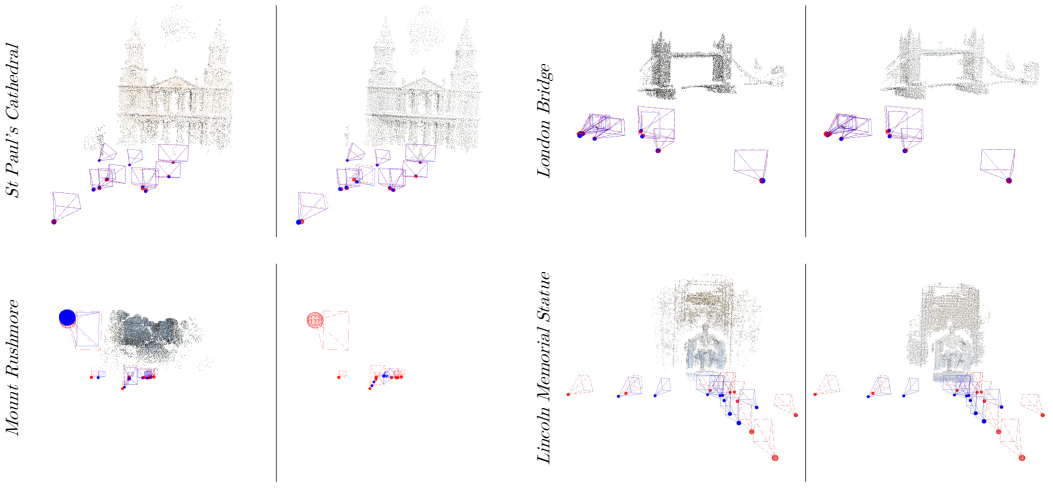


Figure 11: **Qualitative reconstruction comparison on IMC 2021.** Four 10-image landmark bags are shown. **Left**: OddEye + RoMa v2. **Right**: DetectorFreeSfM + RoMa v2.

B.4 Extended 3D Triangulation Analysis on ETH3D

Following the fixed-camera evaluation protocol in Section 4.2, we evaluate 3D point triangulation on the 13 ETH3D training scenes using ground-truth camera poses, intrinsics, and exhaustive image matching.

Figure 12 illustrates the relationship between triangulation accuracy and completeness. Dense matching pipelines typically exhibit an inverse correlation between these metrics: for instance, DetectorFreeSfM increases match density by reducing its quantization stride from $s = 16$ px to $s = 4$ px, improving completeness but degrading geometric accuracy. In contrast, OddEye modulates match density through the anchor budget N_a, shifting the Pareto frontier upward while preserving high accuracy across DKM, RoMa, and RoMa v2.

Table 10 reports the per-scene metrics at the 1 cm threshold. Scaling N_a consistently improves point-cloud completeness across all scenes with negligible impact on accuracy.

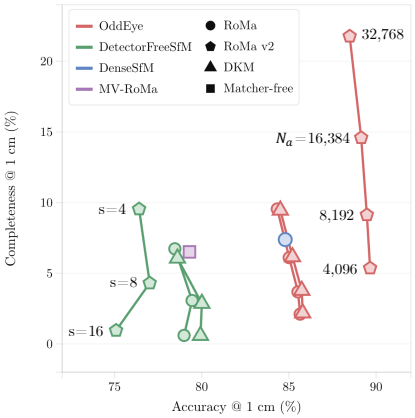


Figure 12: **Accuracy vs. Completeness tradeoff on ETH3D**

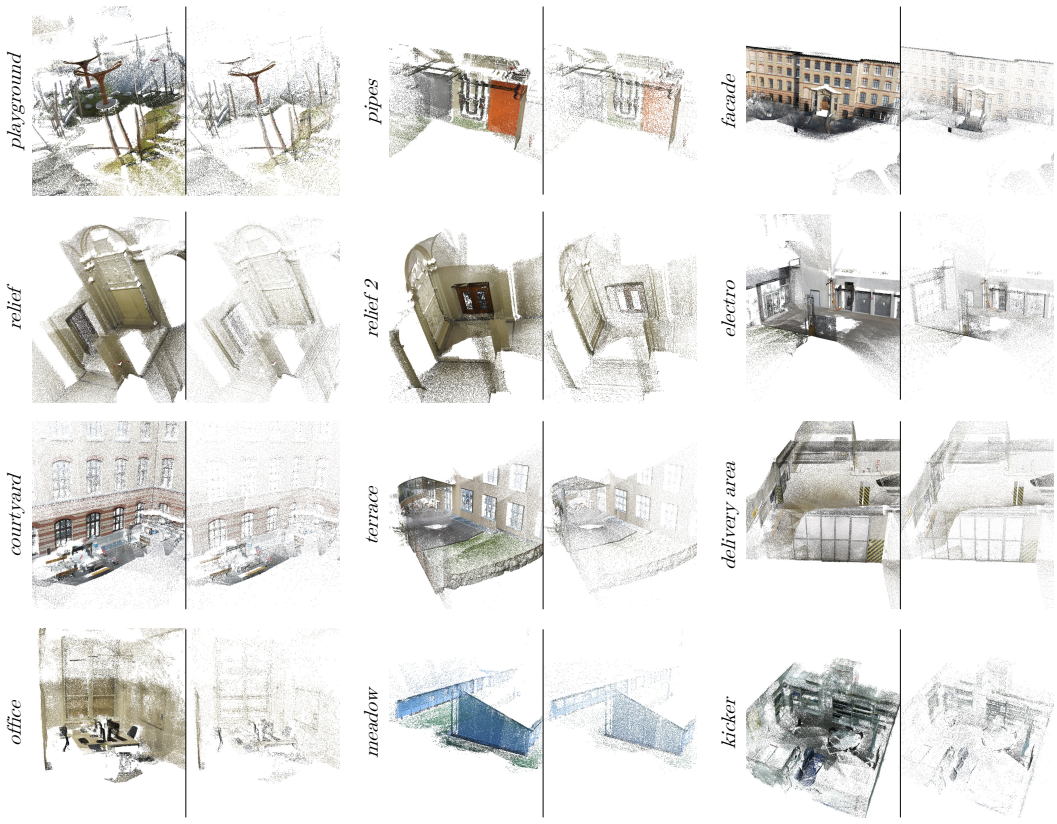


Figure 13: **Qualitative triangulation comparison on ETH3D.** Twelve of the 13 ETH3D training scenes are shown. **Left**: OddEye + RoMa v2. **Right**: DetectorFreeSfM + RoMa v2.

C Additional Analysis of Anchor-Based Track Construction

This section provides further analysis of the anchor-based track-construction mechanism introduced in Section 3.1. We analyze multi-view track length distributions, verify the generalization of fixed

Table 10: **ETH3D fixed-camera triangulation at 1 cm.** Scene-wise accuracy and completeness (%) are reported with exhaustive image pairing and random per-image anchors.

Matcher	DKM				RoMa				RoMa v2			
$\mathbf{N_a}$	16,384		32,768		16,384		32,768		16,384		32,768	
Scene	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.
Kicker	88.78	8.75	87.71	12.65	88.71	8.55	87.49	12.36	**92.29**	22.10	91.60	**33.28**
Courtyard	81.31	2.43	80.76	4.38	**83.38**	2.75	82.79	4.89	81.96	3.95	81.39	**7.02**
Delivery area	80.58	4.14	80.04	7.42	77.33	3.61	76.92	6.49	**85.63**	8.90	85.15	**15.41**
Electro	93.76	5.73	93.43	9.77	93.65	6.87	93.33	11.81	**95.45**	11.51	95.24	**19.22**
Facade	68.92	3.59	68.66	6.48	70.22	3.92	69.69	6.98	**70.39**	5.62	69.78	**9.93**
Meadow	67.26	0.53	66.34	1.00	64.12	0.75	62.95	1.39	**83.16**	4.39	82.87	**8.14**
Office	79.28	4.60	77.76	6.24	80.54	4.73	79.40	6.87	**87.75**	24.95	86.38	**34.15**
Pipes	94.28	3.97	94.33	5.97	93.73	3.14	94.31	4.77	**98.22**	13.73	97.88	**19.11**
Playground	80.46	4.04	79.47	6.69	80.96	5.92	80.00	9.55	**83.98**	11.70	82.52	**18.03**
Relief	95.12	12.19	94.50	17.21	94.91	11.52	94.35	16.47	**95.13**	20.03	94.33	**27.75**
Relief 2	94.68	10.69	94.00	15.66	94.52	10.34	93.76	15.49	**95.36**	17.90	94.63	**25.58**
Terrace	91.07	5.46	90.89	9.92	91.35	6.66	91.15	11.91	**94.13**	10.68	94.04	**18.77**
Terrains	92.19	13.91	90.74	19.54	91.82	10.58	90.21	15.13	**95.45**	34.18	94.51	**46.62**
Average	85.21	6.16	84.51	9.46	85.02	6.10	84.34	9.55	**89.15**	14.59	88.49	**21.77**

anchors to an existing dense multi-view matcher, evaluate alternative anchor-sampling strategies, and assess the impact of post-hoc learned track refinement.

C.1 Track Length Analysis

To measure multi-view track connectivity, we report the mean track length, defined as the number of 2D observations divided by the number of triangulated 3D points. Table 11 compares DenseSfM with OddEye under exhaustive matching and Top-5 on-demand scheduling, all using RoMa in incremental SfM, on the 13 ETH3D training scenes evaluated by Lee & Yoo (2025).

Averaged over scenes, Top-5 scheduling yields a mean track length of 5.14, compared with 5.71 for exhaustive matching and 4.97 for DenseSfM, and exceeds DenseSfM on 11 of the 13 scenes. Most of the gap to exhaustive matching comes from *facade* (5.74 vs. 12.53), the largest scene with 76 images. There, exhaustive matching can extend each anchor's track to any of the other 75 views, whereas on-demand scheduling extends a track only when its root image is selected as a partner. Excluding *facade*, Top-5 retains 99% of the exhaustive average (5.09 vs. 5.15). Attention-guided scheduling thus forms longer tracks than DenseSfM and preserves most of the connectivity of exhaustive matching while evaluating far fewer image pairs.

Table 11: **Mean track length comparison across 13 ETH3D scenes.** † marks values reported by Lee & Yoo (2025).

Method	Courtyard	Delivery	Electro	Facade	Kicker	Meadow	Office	Pipes	Playground	Relief	Relief 2	Terrace	Terrains	**Avg.**
DenseSfM†	5.12	4.59	5.11	9.48	4.18	3.58	4.23	3.97	4.25	5.75	5.15	4.61	4.64	4.97
OddEye (Exhaustive)	5.04	4.81	**5.56**	**12.53**	**5.03**	**4.29**	**4.68**	4.34	**4.52**	**6.69**	**6.27**	5.15	5.38	**5.71**
OddEye (Top-5)	**5.34**	**5.21**	5.37	5.74	3.96	4.22	4.28	**4.52**	4.45	6.55	6.15	**5.44**	**5.54**	5.14

C.2 Effect of Fixed Per-Image Anchors in MV-RoMa

MV-RoMa (Lee et al., 2026b) is a one-to-many dense matching framework that achieves competitive results on IMC 2021 (Jin et al., 2020) and Texture-Poor SfM (He et al., 2024), but exhibits substantially lower reconstruction accuracy on ETH3D (Table 1). We hypothesize that this performance gap stems from its source-coordinate sampling strategy: MV-RoMa resamples source coordinates independently for each local image group, preventing predictions from the same source image across different groups from forming connected multi-view tracks.

Table 12: **Effect of source-coordinate sampling in MV-RoMa on ETH3D.**

Source-sampling strategy	@1°	@3°	@5°	Reg. (%)
Per-group (original)	37.43	43.29	44.85	66.5
Fixed per-image (ours)	**59.53**	**69.21**	**71.60**	**88.5**

To test this hypothesis, we modify MV-RoMa by sampling $N_a = 4096$ anchors once per source image and reusing them across all groups containing that image, allowing predictions from common anchors to extend shared tracks.

As shown in Table 12, adopting fixed per-image anchors improves AUC across all three thresholds and increases the registration rate from 66.5% to 88.5% on ETH3D. This gain supports our hypothesis that persistent source identity is essential for building cohesive multi-view tracks on ETH3D, whose scenes contain up to 110 images. MV-RoMa remains competitive on IMC 2021 and Texture-Poor SfM, which use bags of only 5 to 25 images; in IMC 2021, all images in a bag view the same side of a landmark with substantial overlap. We attribute this to registration in such bags relying on a few directly overlapping views, for which short, fragmented tracks are often sufficient.

C.3 Comparison of Anchor-Location Sampling Strategies

OddEye samples anchor coordinates uniformly at random by default, without relying on keypoint detectors or saliency heuristics. To determine whether incremental SfM benefits from feature saliency at the source, we evaluate three anchor selection strategies under a fixed budget of $N_a = 4096$ points per image: (i) *Random*, sampled uniformly across the image plane; (ii) *Regular grid*, spaced evenly on a 2D lattice; and (iii) *ALIKED detector*, selecting the top-N_a keypoints ranked by ALIKED (Zhao et al., 2023) confidence.

As reported in Table 13, all three sampling strategies yield identical registration rates across both Texture-Poor SfM (5-image bags) and ETH3D, with Pose AUC values differing by at most 1.24 pp. These results demonstrate that tracking performance depends on maintaining consistent observation identities across views rather than the spatial or semantic saliency of the source locations.

Table 13: **Comparison of anchor-location sampling strategies.**

	Texture-Poor SfM (5-image bags)			ETH3D		
Sampling strategy	AUC@3°	AUC@5°	Reg. (%)	AUC@1°	AUC@3°	Reg. (%)
Random (default)	58.82	73.57	**99.8**	**86.16**	**94.98**	**100.0**
Regular grid	**59.13**	**73.91**	**99.8**	85.40	94.68	**100.0**
ALIKED detector	57.89	72.77	**99.8**	84.92	94.54	**100.0**

C.4 Effect of Post-Hoc Multi-View Track Refinement

DetectorFreeSfM (He et al., 2024) and DenseSfM (Lee & Yoo, 2025) refine the 2D observations of each track with a learned multi-view network; in DetectorFreeSfM, this compensates for quantization during track initialization. OddEye's observations are never quantized, since they are queried from continuous correspondence fields at fixed anchors (Section 3.1). To test whether refinement still helps, we apply the publicly released pretrained refiners of both methods to OddEye's tracks and then run global bundle adjustment.

Table 14: **Effect of post-hoc multi-view track refinement.** Pose AUC (%) of OddEye with RoMa v2 before and after applying the pretrained DetectorFreeSfM and DenseSfM refiners. Texture-Poor SfM results use 5-image bags.

	ETH3D			Texture-Poor SfM		
Refinement model	@1°	@3°	@5°	@3°	@5°	@10°
OddEye (no refinement)	**86.16**	**94.98**	**96.93**	**58.82**	**73.57**	**86.24**
+ DetectorFreeSfM refiner	84.23	94.12	96.47	55.18	71.25	85.09
+ DenseSfM refiner	77.97	91.53	94.69	55.02	70.98	84.88

Neither refiner improves OddEye (Table 14). Both lower average pose AUC at every threshold on both benchmarks. On ETH3D, AUC@1° drops from 86.16% to 84.23% with the DetectorFreeSfM refiner and to 77.97% with the DenseSfM refiner, which causes large drops on several scenes (Figure 14). We attribute this to a mismatch with the refiners' training data: each was trained on observations from its own pipeline, and applied to OddEye's subpixel observations, its corrections can shift them away from their true locations. OddEye's anchor-based tracks therefore do not require learned refinement. Omitting it also avoids substantial runtime overhead, since both refiners process each track individually. On ETH3D *facade*, measured on an RTX 4090 GPU, the DenseSfM refiner adds about 77 minutes to OddEye's 14-minute reconstruction, increasing total runtime by $6.5\times$.

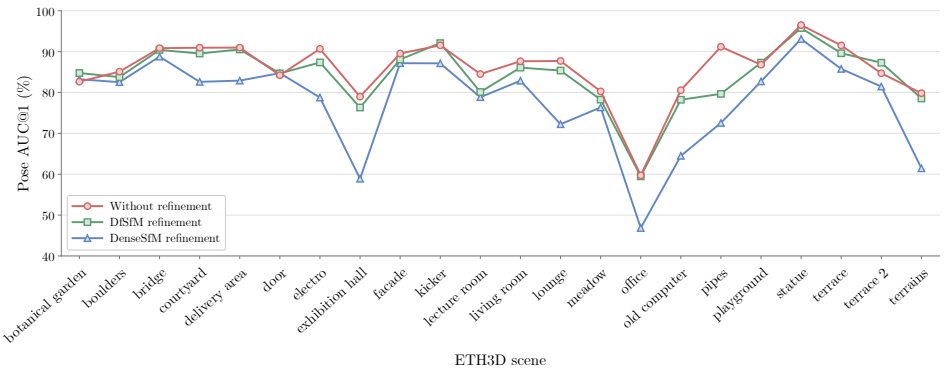


Figure 14: **Per-scene effect of multi-view track refinement on ETH3D.** Pose AUC@1° before refinement and after applying the DetectorFreeSfM (DfSfM) or DenseSfM refiner. The effect is scene-dependent, and the DenseSfM refiner produces the largest drops.

D Additional Analysis of the Attention Prior and On-Demand Scheduling

This section analyzes the feed-forward attention prior and the on-demand scheduler built on it. We describe attention-head calibration (Appendix D.1), select the backbone (Appendix D.2), and test robustness to inaccurate feed-forward reconstructions (Appendix D.3). We then ablate initial-pair selection (Appendix D.4) and examine failure cases of appearance-based retrieval (Appendix D.5).

D.1 Attention-Head Calibration

For each feed-forward backbone, we select one attention head on seven scenes from the Tanks & Temples (Knapitsch et al., 2017) training split (*Barn*, *Caterpillar*, *Church*, *Courthouse*, *Ignatius*, *Meeting Room*, and *Truck*), none of which appear in our evaluation benchmarks. From each scene, we sample 100 images at uniform intervals over the full image set; the same subsets are used for backbone selection in Appendix D.2. These scenes provide COLMAP reference poses but no per-image depth maps or 2D–2D correspondence annotations, so we derive pseudo-ground-truth correspondences from RoMa v2 (Edstedt et al., 2025) matches verified against the reference geometry.

For each image pair, we keep matches with confidence at least $\tau_c = 0.5$ and cycle error at most $\tau_{\text{cycle}} = 3$ px (Table 6) whose Sampson error under the reference cameras is below 2 px. To guard against errors in the reference reconstruction or the matcher, we discard pairs whose relative pose, estimated from the surviving matches, deviates from the reference pose by more than 5° in rotation or translation direction. We then quantize the remaining matches to the backbone's patch grid: each source patch p is assigned the target patch $\bar{q}$ that receives the most of its verified matches, and source patches with fewer than three verified matches are discarded. This yields patch-level correspondences $\mathcal{Q} = \{(i, j, p, \bar{q})\}$. Evaluating at patch resolution makes the supervision tolerant to small errors in both the reference poses and the matches. Requiring agreement with the reference geometry also limits the influence of RoMa v2's own errors, so the calibration targets scene geometry rather than matcher behavior.

We score every layer–head pair of a backbone with the top-1 patch-correspondence accuracy of Eq. 2 and keep the highest-scoring pair fixed for all evaluation scenes. For DVLT, the layer index ℓ denotes the recurrent iteration (Appendix A.1), and the selected head is $(\ell^\star, h^\star) = (11, 0)$ (Figure 6). Table 15 reports the selected head and its accuracy for every backbone.

D.2 Backbone Selection

We compare VGGT-1B (Wang et al., 2025a), π^3 (Wang et al., 2025b), DA3-Large (Lin et al., 2025), and DVLT (Burzio et al., 2026) using three signals. Two are measured on the calibration scenes of Appendix D.1: direct pose accuracy, which compares each model's predicted poses with the

Table 15: **Backbone selection.** Selected head, its top-1 patch-correspondence accuracy (Eq. 2), and direct pose AUC (%) on the Tanks & Temples calibration scenes, together with downstream OddEye pose AUC (%) and registration rate on the 22 ETH3D evaluation scenes with RoMa v2 and top-5 on-demand matching. Each backbone uses its own calibrated head.

	Head (T&T)		**Direct pose (T&T)**			**OddEye (ETH3D)**			
Backbone	$(\ell^\star, h^\star)$	Acc. (%)	@3°	@5°	@10°	@1°	@3°	@5°	Reg. (%)
VGGT-1B	(15, 8)	27.07	57.05	69.97	80.97	82.98	91.65	93.57	99.96
π^3	(31, 8)	26.90	59.14	71.98	83.26	86.02	94.82	96.76	**100.00**
DA3-Large	(11, 9)	23.35	39.22	55.72	71.86	70.87	78.71	80.61	94.15
DVLT	(11, 0)	**31.70**	**60.93**	**73.12**	**83.45**	**86.16**	**94.98**	**96.93**	**100.00**

COLMAP reference using the pose AUC metric of Section 4, and head accuracy, the Eq. 2 score of the selected head. The third is downstream accuracy: we extract the prior from each backbone with its own calibrated head and run OddEye on the 22 ETH3D evaluation scenes with RoMa v2 and top-5 on-demand matching, with all other settings following Table 6.

Table 15 reports all three signals. DVLT achieves the highest direct pose accuracy, and we adopt it as the default backbone. The downstream ranking matches the direct pose ranking at every threshold: DVLT $>$ π^3 $>$ VGGT-1B $>$ DA3-Large. Head accuracy agrees at both ends, ranking DVLT first and DA3-Large last, but does not resolve the middle: VGGT-1B and π^3 are nearly tied in head accuracy (27.07% vs. 26.90%), whereas π^3 leads VGGT-1B by about 3 pp downstream. DVLT and π^3 are nearly tied downstream, differing by at most 0.17 pp in AUC, and both register every image, so OddEye is not tied to a single backbone. Appendix D.3 shows that direct pose accuracy on other data can diverge from downstream quality, which is why we verify the choice downstream.

D.3 Robustness to Imperfect Feed-Forward Reconstructions

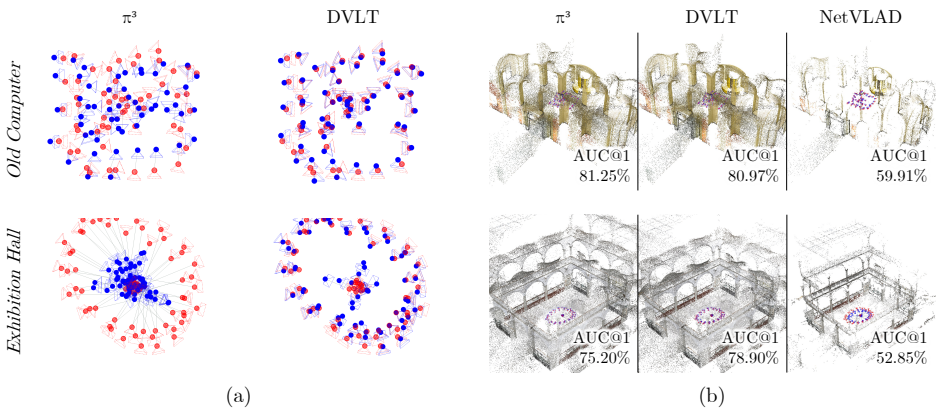


Figure 15: **Robustness to imperfect feed-forward 3D priors.** (a) π^3 and DVLT camera predictions aligned to the reference reconstruction on *Old Computer* and *Exhibition Hall* (blue: predicted; red: reference). (b) OddEye reconstructions scheduled by π^3, DVLT, and NetVLAD, with AUC@1°.

Because OddEye uses the feed-forward model only for scheduling and estimates all geometry from verified correspondences, the prior does not require an accurate global reconstruction; it only requires attention that identifies co-visible regions. Figure 15 illustrates this on *Exhibition Hall* and *Old Computer*. The π^3 camera predictions deviate markedly from the reference reconstruction, while DVLT's remain closer (Figure 15(a)). Nevertheless, π^3-guided OddEye stays close to DVLT-guided OddEye on both scenes, while NetVLAD-guided scheduling falls far behind (Figure 15(b)).

The same decoupling holds across all 22 ETH3D scenes. π^3's direct predictions reach 6.07% AUC@1°, about a third of DVLT's 17.29% (Table 9), yet π^3-guided OddEye is within 0.14 pp of DVLT-guided OddEye (Table 15). Conversely, DA3-Large produces more accurate direct poses

than π^3 at every threshold but yields the weakest schedule. Direct pose accuracy is therefore only a coarse indicator of how useful a model's attention is for scheduling.

D.4 Ablation of Initial-Pair Selection

We ablate the ranking used for initial-pair selection on the 22 ETH3D scenes with RoMa under the budget of Table 5 ($k = 3$), keeping next-view and registered-partner selection fixed to the attention prior. For the DINOv2 (Oquab et al., 2023) and NetVLAD (Arandjelovic et al., 2016) baselines, we rank candidate initial pairs by the cosine similarity of global descriptors, as in Section 4.5. As in Section 3.3, a pair that fails geometric verification is skipped, and the next-ranked pair is tried until initialization succeeds.

All three criteria eventually initialize every scene and reach similar registration rates (about 97.1%), but they differ in how quickly they find a valid pair (Table 16). The top-ranked attention pair passes verification in 10 of 22 scenes, compared with 6 for both baselines. Across all scenes, attention needs 72 matcher calls to initialize, 54% fewer than NetVLAD (156) and 73% fewer than DINOv2 (262). The initial pair also affects final accuracy: attention achieves the highest AUC@3° (89.07%), ahead of DINOv2 by 0.49 pp and NetVLAD by 2.08 pp.

Table 16: **Ablation of initial-pair selection on 22 ETH3D scenes.** Next-view and partner selection use the attention prior in all rows (RoMa, $k = 3$). *Top-ranked verified*: scenes whose highest-ranked pair passes geometric verification on the first attempt. *Matcher calls*: total dense-matcher pair evaluations during initialization, summed over all scenes.

Ranking Criterion	Top-Ranked Verified ↑	Matcher Calls ↓	Pose AUC@3° ↑	Reg. Rate (%) ↑
DINOv2	6 / 22 (27.3%)	262	88.58	97.08
NetVLAD	6 / 22 (27.3%)	156	86.99	97.14
Attention (Ours)	**10 / 22 (45.5%)**	**72**	**89.07**	**97.14**

D.5 Failure Cases of Appearance-Based Pair Retrieval



Figure 16: **Appearance-retrieval false positives on ETH3D.** Each query (left) is paired with a visually similar candidate (right) that has no geometric overlap with it. Labels give the candidate's rank among all other images (#1 = most similar), so larger is correct here. In every example, at least one global descriptor ranks the candidate in its top three, while the attention prior ranks it within its bottom four.

Figure 16 extends Figure 7 with four more ETH3D examples. Each pairs a query image with a candidate that looks similar but has no geometric overlap. In every case, at least one global descriptor ranks the candidate among its top three: DINOv2 (Oquab et al., 2023) in *Lecture Room* and *Delivery Area*, and NetVLAD (Arandjelovic et al., 2016) in *Facade* and *Old Computer*. The similarity comes from repeated or generic structure, such as repeated windows in *Facade* and warning-striped pillars in *Delivery Area*. The attention prior places all four candidates within the bottom four of its ranking, consistent with it encoding cross-view geometric compatibility rather than global appearance alone. Attention ranks are computed with the symmetric pair score w_{ij} (Eq. 5).

E SCALABILITY

Three costs grow with the number of images N: extracting the attention prior, which runs global attention over all NP patch tokens; storing it, which requires $\mathcal{O}(N^2P)$ memory (Section 3.2); and dense matching, which on-demand scheduling already reduces from $\mathcal{O}(N^2)$ pairs to about kN. Because OddEye uses only the cross-view attention of the feed-forward model, not its decoded cameras, depth, or point maps, we can specialize the first two for this purpose. Appendix E.1 bounds peak memory during extraction and reduces the size of the stored prior, and Appendix E.2 evaluates complete reconstructions on four SMERF scenes with up to 1,874 images.

E.1 MEMORY-EFFICIENT PRIOR EXTRACTION AND STORAGE

We drop the decoders, which the prior does not use, run the attention layers in BF16, and split the attention queries into chunks. Each chunk still attends to the full key–value set, and softmax normalizes each query row independently, so chunking is exact: it changes only the execution order. Within each chunk, we reduce the attention of the selected head $(\ell^\star, h^\star)$ to the per-image maxima of Eq. 3 and move the result to host memory before processing the next chunk. The full attention matrix is therefore never materialized, and the prior itself never occupies GPU memory. Figure 17 compares peak VRAM for FP32, BF16, and chunked BF16 on an NVIDIA RTX PRO 6000. Unchunked BF16 exhausts the GPU's 94.97 GiB at $N = 8{,}192$, whereas chunked BF16 completes with 39.45 GiB, without offloading model weights or activations to the CPU or approximating global attention.

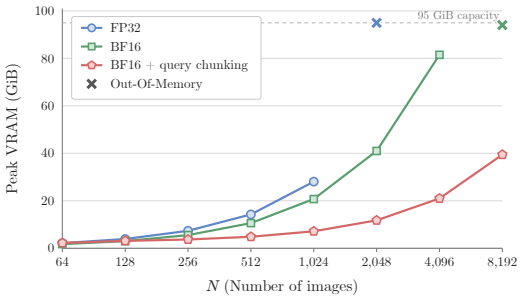


Figure 17: **Peak VRAM for attention-prior extraction**; × marks out-of-memory.

Chunking bounds transient memory, but the prior of Section 3.2 still stores an $N \times N \times P$ tensor of patch scores $h_{i\to j}(p)$ (Eq. 3), even though most image pairs barely overlap. We therefore drop every pairwise map whose pair score $\bar{h}_{i\to j}$ (Eq. 4) falls below a threshold. On the 990-image NYC scene, a threshold of 5×10^{-3} retains 43.42% of the maps and reduces the prior from 2701.3 MiB to 1170.5 MiB.

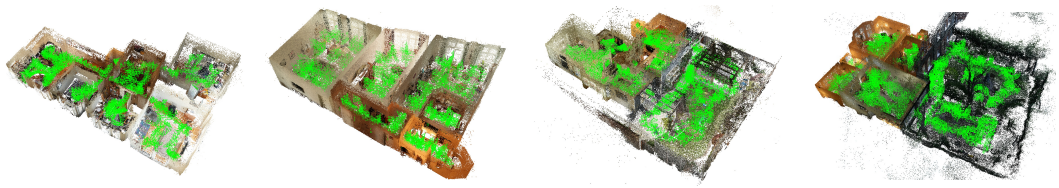
E.2 LARGE-SCALE RECONSTRUCTION ON SMERF

We run the complete pipeline on the full NYC, Berlin, Alameda, and London scenes of SMERF (Duckworth et al., 2024), which contain 990 to 1,874 images. All runs use RoMa v2, $k = 5$ registered partners, the published intrinsics, chunked prior extraction (Appendix E.1), and the uncompressed prior. We lower the anchor budget from the default 4,096 to $N_a = 1{,}024$ because the number of triangulated points, and with it bundle-adjustment time, grows with N_a. Pose AUC is computed against the published camera poses.

Table 17: **Large-scale reconstruction on SMERF.** Registered images, pose AUC@3°, peak GPU memory, and end-to-end runtime per scene on an NVIDIA RTX PRO 6000. All runs use RoMa v2, $N_a = 1{,}024$, $k = 5$, and published intrinsics.

Scene	Images	Registered	AUC@3° (%)	Peak mem. (GiB)	Runtime
NYC	990	990	95.33	5.35	3.26 h
Berlin	1,511	1,508	92.85	7.70	8.71 h
Alameda	1,734	1,734	94.75	8.70	9.30 h
London	1,874	1,865	93.92	9.33	12.56 h

Table 17 summarizes the results, and Figure 18 shows the reconstructions of all four scenes. OddEye registers 6,097 of 6,109 images (99.8%), including every image in NYC and Alameda, with AUC@3° between 92.85% and 95.33%. Peak GPU memory stays below 10 GiB in every scene and grows roughly linearly with the number of images, from 5.35 GiB on NYC to 9.33 GiB on London.



(a) NYC (b) Berlin (c) Alameda (d) London

Figure 18: **OddEye reconstructions of the four full SMERF scenes.** Reconstructed point clouds and estimated cameras, rendered as in Figure 9: (a) *NYC* (990 images), (b) *Berlin* (1,511 images), (c) *Alameda* (1,734 images), and (d) *London* (1,874 images). Per-scene registration, pose accuracy, memory, and runtime are reported in Table 17.

End-to-end runtime ranges from 3.26 h to 12.56 h. For reference, exhaustive matching of London would require about 1.76 M image pairs, whereas on-demand scheduling matches on the order of $kN \approx 9.4$ k pairs.

F Limitations and Future Directions

Scaling to larger collections. OddEye reconstructs SMERF scenes of up to 1,874 images with under 10 GiB of peak GPU memory (Appendix E.2), and chunked extraction computes the prior for up to 8,192 images (Appendix E.1). Internet-scale collections with tens of thousands of images remain challenging, because the cost of global cross-view attention grows quadratically with N and the prior must be extracted from all images in a single joint pass before reconstruction begins. Efficiency techniques for feed-forward 3D models, such as token merging (Shen et al., 2025; Wang et al., 2026b), block-sparse global attention (Wang et al., 2026a), and key–value subsampling (Sun et al., 2026), could reduce this cost.

Dependence on attention quality. OddEye tolerates inaccurate global predictions from the feed-forward model (Appendix D.3), but it relies on the model's attention to identify co-visible regions. When the prior is less reliable, reconstruction degrades: with DA3-Large as the backbone, ETH3D AUC@3° falls from 94.98% to 78.71% (Table 15). Because partners are chosen solely by attention, a candidate whose partners are misranked may fail to register even when a suitable registered view exists. In addition, the attention head is calibrated once on Tanks & Temples, and its transfer to substantially different domains, such as aerial or underwater imagery, remains untested. Falling back to appearance-based retrieval or expanding the partner set when registration fails could make the scheduler more robust to such errors.

Densely captured sequences. OddEye targets unordered image collections. In densely captured sequences such as video, adjacent frames overlap the most and tend to receive the highest attention scores, so registered-partner selection can favor near-duplicate views. These pairs have small baselines and triangulation angles, which leave newly triangulated points poorly constrained in depth and can destabilize bundle adjustment. Extending OddEye to sequential data could exploit the relative poses predicted by the feed-forward model, for example by excluding partners whose predicted baseline or viewing-angle difference falls below a threshold.