DMAP: EFFICIENT DENSE MATCHING AND BUNDLE ADJUSTMENT FOR ACCURATE 3D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

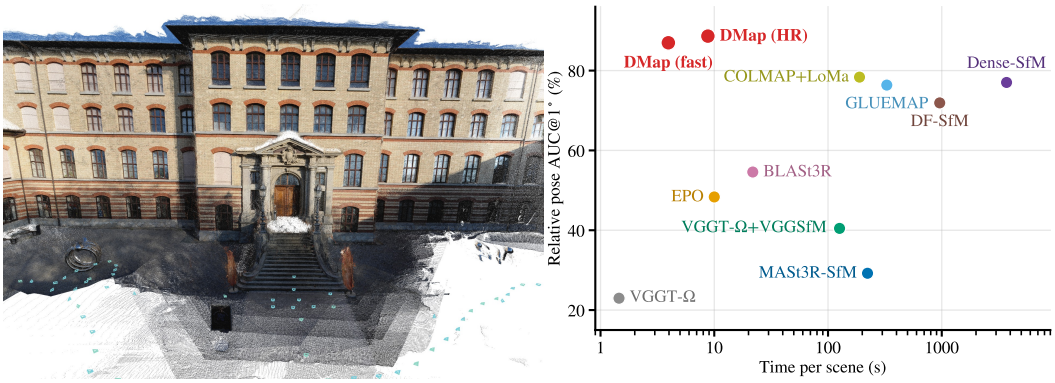


Figure 1: **Accurate and fast dense bundle adjustment.** (Left) Dense points and cameras (cyan) that DMap reconstructs on the ETH3D (Schöps et al., 2017) `facade` scene. (Right) Relative-pose AUC at 1° against time on the 25 ETH3D scenes average. At 1°, DMap reaches 88.6 % in 8.8 s per scene.

ABSTRACT

Deep 3D models provide remarkable global robustness, but direct regression acts as an information bottleneck that discards sub-pixel geometric constraints. Conversely, classical optimization achieves sub-pixel precision but fails to scale to dense correspondences and relies on heuristic, discrete track building. We present DMap, a framework that bridges frozen foundation representations with efficient dense optimization. First, DMap extracts dense multi-view correspondences from the latent features of a frozen multi-view backbone, amortizing per-view representation computation so that per-pair decoding reduces to lightweight coarse feature matching and local convolutional refinement. Second, we resolve the efficiency bottleneck of non-linear optimization by formulating dense bundle adjustment on a regular view graph over regular 2D pixel grids rather than irregular point graphs. By anchoring tracks to source rays, dense warps enter the optimization directly without keypoint quantization or track merging, and millions of depth variables are marginalized in parallel within milliseconds. DMap achieves state-of-the-art accuracy across unordered (ETH3D, Texture-Poor) and sequential (TUM RGB-D, ETH3D-SLAM) benchmarks, improving relative-pose AUC@1° to 88.6% on ETH3D while running 13–17× faster than existing dense optimization baselines.

1 INTRODUCTION

3D reconstruction from unposed images requires jointly estimating camera poses, intrinsic parameters, and scene geometry. Learning-based 3D foundation models (3DFMs) (Wang et al., 2025a; 2026a; Burzio et al., 2026; Leroy et al., 2026) have recently emerged as a powerful paradigm, regressing this geometry in a single forward pass with remarkable robustness. However, direct regression can only describe a unimodal parametric approximation centered on a single hypothesis. In contrast, dense correspondences and depth priors better model the true non-linear negative log-likelihood (energy landscape), preserving the multi-modal ambiguities, basin curvatures, and fine-grained visual evidence that a single forward regression cannot capture. Formulating an explicit probabilistic

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.

landscape decouples raw visual evidence from the inductive biases baked into training data, such as ideal global-shutter pinholes with centered principal points.

Finding most probable reconstruction within this landscape is the domain of Bundle Adjustment (BA), solving the non-linear least-squares problem via gradient-based / second-order descent. In recent years, the community has increasingly favored purely feed-forward models with belief that iterative optimization is inherently slow and unscalable for dense observations. Modern methods frequently bypass BA entirely or severely thin out visual evidence into sparse tracks (Duisterhof et al., 2025) or edges (D'Urso et al., 2026) just to keep optimization tractable. We show dense BA need not be the bottleneck. The classical performance penalty arose from managing irregular sparse point graphs, not from the iterative optimization itself. Parameterizing geometry constraints in regular view graph with regular pixel grids yields a predictable, coalesced memory layout. Fused GPU kernels evaluate reprojection residuals and marginalize millions of scene geometry variables via the Schur complement in parallel, forming the reduced camera system in few milliseconds.

As a result, dense BA is so computationally lightweight that the real bottleneck shifts back to the matching step that populates the objective. Pairwise dense matchers (Edstedt et al., 2024; 2025; Nordström et al., 2026b) run a heavy cross-view backbone for every image pair, coupling scene representation and match decoding. While pairwise correspondence output inevitably scales with the number of edges, extracting the scene representation need not. The feed-forward model that initialises the poses has already processed the entire view set, encoding rich multi-view context into its latent representations. We amortize this expensive representation step by adapting per-image features from the shared backbone only once, regardless of the number of pairs to decode. The per-edge decoding then reduces to a lightweight operation: a coarse descriptor matching followed by local convolutional refinement. By decoupling joint scene representation from per-edge match decoding, dense multi-view matching scales efficiently even across dense view graphs.

This division of labour mirrors the scale hierarchy of the problem, as we move from high-level understanding to low-level alignment, the required receptive field collapses from global context to locality. At the highest level, resolving view overlaps and coarse layouts requires a global receptive field, where attention excels. The coarse decoder then narrows the candidates to local patches. Next, the convolutional refiner further utilizes high-frequency local evidence from image to pinpoint sub-pixel location. Once this multi-scale visual evidence defines the energy landscape, Bundle Adjustment acts on the search side. This principle reflects a broader paradigm shift across modern foundation models: while a language model can be scaled to approximate multi-digit arithmetic, yet querying a simple calculator remains exponentially cheaper while mathematically exact. Instead of scaling 3DFMs to accurately solve the non-linear optimization, they can be used to provide coarse global initialization that places the system in the correct basin of convergence, while local second-order optimization descends the reprojection residuals to reach the true minimum. Simply providing the right computational tools allows the model to operate at the scale of abstraction it is naturally suited for.

The result is a complete pipeline with feed-forward initialisation, multiview match decoding, and dense bundle adjustment where geometric refinement is both fast and accurate. Using explicit reprojection residuals instead of implicit network weights allows the model to naturally handle lens distortion, rolling shutter, and uncertainty directly and without needing to retrain.

We make three contributions:

1. **A lightweight multi-view match decoder.** A hierarchical decoder with a looped transformer adapter that extracts dense correspondences from a frozen multi-view backbone, sharing per-view computation across all edges. The marginal cost per pair reduces to a lightweight coarse feature matching and convolutional refinement. The head leverages multi-view context to resolve low-overlap pairs and the architecture is applicable across backbones.
2. **Efficient dense bundle adjustment.** We perform dense bundle adjustment over regular pixel grids rather than irregular point tracks which enables fused GPU kernels that construct and reduce the normal equations via the Schur complement in parallel. The solver self-calibrates intrinsics, principal points, radial distortion, under a robust Cauchy loss.
3. **A validated paradigm for tool-augmented 3D foundation models.** We combine foundation multi-view model feature with exact non-linear optimization into a complete, efficient reconstruction pipeline for 3D reconstruction. Across extensive benchmarks, we demon-

strate that delegating local optimization to numerical solvers enables learned feedforward models to achieve accurate 3D reconstruction while remaining robust and efficient.

2 Related Work

Feed-forward reconstruction. Feed-forward models regress geometry directly from images. DUSt3R (Wang et al., 2024b) predicts pointmaps for an image pair and aligns the pairs globally, and MASt3R (Leroy et al., 2024) adds a matching head. Later models reconstruct many views at once, with memory (Wang & Agapito, 2025a; Wang et al., 2025b; Cabon et al., 2025) or with global attention (Yang et al., 2025; Wang et al., 2025a; 2026b; Keetha et al., 2026; Wang et al., 2026a). Déjà View (Burzio et al., 2026) loops a shared transformer block to trade compute for accuracy at test time. These models are robust to wide baselines and weak texture, but their predictions can lack geometric precision. VGGT (Wang et al., 2025a) achieves higher accuracy with bundle adjustment (BA), but the complete method takes substantially longer than feed-forward inference alone. DMap addresses this trade-off, improving geometric precision without compromising efficiency.

Reconstruction with learned geometric priors. BA can improve geometric precision, but limited multiview overlap or parallax can leave the reconstruction weakly constrained. Learned geometric priors provide additional constraints in these cases. MP-SfM (Pataki et al., 2025) adds monocular depth and normal priors to incremental SfM. VidMap (Pataki et al., 2026) incorporates metric depth priors into global positioning and BA. MASt3R-SfM (Duisterhof et al., 2025) globally aligns the pairwise geometry predictions of MASt3R with sparse correspondences. SLAM methods align feed-forward predictions through pose-graph optimization (Murai et al., 2025; Maggio et al., 2025; Deng et al., 2025). GlueMap (Pan et al., 2026) runs a feed-forward model on local subgraphs of a large image set and turns its depth maps into virtual tracks to complement classical matching in global SfM. EPO (D'Urso et al., 2026) refines camera parameters and depths predicted by feed-forward models through cross-view edge alignment. GGPT (Chen et al., 2026b) initializes sparse BA from feed-forward geometry, using sparse tracks sampled from pairwise dense matches. BLASt3R (Leroy et al., 2026) couples learned multiview matching with depth-regularized BA, jointly refining cameras, sparse 3D tracks, and coefficients of a learned log-depth basis. DMap instead reuses the features of the feed-forward model for matching and uses dense matches in BA. Its formulation supports depth priors from the feed-forward model to regularize geometry.

Dense bundle adjustment. Classical SfM chains sparse keypoint matches into tracks and refines cameras and points by BA (Triggs et al., 1999; Schonberger & Frahm, 2016; Pan et al., 2024). The Schur complement eliminates the points that couple only to the cameras that observe them (Triggs et al., 1999; Agarwal et al., 2010), while multicore and GPU solvers parallelize the reduced camera system (Wu et al., 2011; Ren et al., 2022). Sparse keypoints, however, can suffer from localisation noise and poor repeatability across views. Later works refine keypoints and tracks with dense features (Dusmanu et al., 2020; Lindenberger et al., 2021) or replace the detector by semi-dense and dense matchers (He et al., 2024; Lee & Yoo, 2025; Lee et al., 2026), but still merge the matches into sparse, irregular tracks. Photometric methods optimise dense structure directly (Newcombe et al., 2011; Delaunoy & Pollefeys, 2014), and LSPBA (Woodford & Rosten, 2020) anchors each landmark on the ray of a source pixel and refines millions of them with hundreds of cameras. For video, DROID-SLAM (Teed & Deng, 2021) builds on learned dense BA (Tang & Tan, 2019; Teed & Deng, 2020a). It solves for one inverse depth per pixel against dense flow and eliminates the diagonal depth block by the Schur complement. DroidCalib (Hagemann et al., 2023) adds the camera intrinsics to this BA, including models with lens distortion, and MegaSaM (Li et al., 2025) and ViPE (Huang et al., 2025) later add monocular depth priors. DMap keeps the source-anchored depth parameterisation but applies it to unordered image sets as well as video, at higher resolution. It self-calibrates a camera with radial distortion (Fitzgibbon, 2001), corrects rolling shutter (Hedborg et al., 2012), and exploits the regular pixel grid for fused GPU kernels.

Dense matching. Dense matchers estimate a correspondence for every pixel. DKM (Edstedt et al., 2023), RoMa (Edstedt et al., 2024) and RoMa v2 (Edstedt et al., 2025) match globally at a coarse stride and refine with convolutional stages. UFM (Zhang et al., 2025) unifies wide-baseline matching with optical flow (Teed & Deng, 2020b). Their pair-dependent processing scales with the number of image pairs. MV-RoMa (Lee et al., 2026) extends RoMa to one source and several targets at

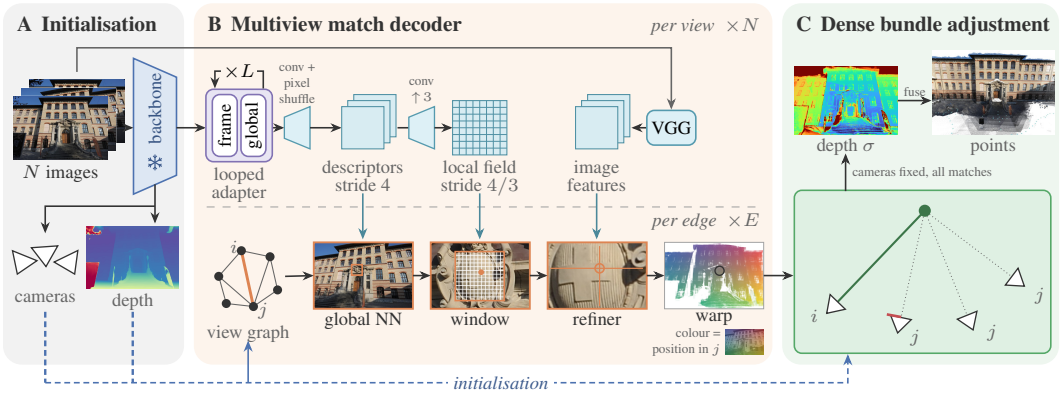


Figure 2: **Pipeline overview.** (A) A frozen feed-forward network predicts the initial cameras and depths. (B) The upper lane runs once per view and is shared by all edges $i \to j$ (lower lane). The insets follow the circled source pixel; the warp is coloured by the matched position in j, with the confidence as opacity. (C) Bundle adjustment refines cameras and depths, then the depths alone from all matches; their uncertainty σ guides fusion. Examples from ETH3D `facade`.

once to build tracks. Probing studies find correspondences inside the attention of feed-forward 3D models (Bratulić et al., 2025; Zhang et al., 2026). RoMa-Ω (Nordström et al., 2026b) replaces the frozen encoder of RoMa v2 with VGGT-Ω and confirms that these representations carry strong matching cues, but it still runs the trunk once per image pair. DMap decodes correspondences from the feed-forward model it already runs for initialisation. A looped adapter (Dehghani et al., 2019; Burzio et al., 2026) runs once over all views, and the descriptor fields are computed once per view, so each pair costs only a descriptor comparison and a convolutional refinement.

3 Method

Problem formulation. We consider N images, denoted by $\{\mathbf{I}_i \in \mathbb{R}^{H_i \times W_i} \mid i \in \mathcal{I} = \{1 \dots N\}\}$, each with size $H_i \times W_i$. We aim to estimate the pose ${}_w\mathbf{T}_i \in \mathbf{SE}(3)$ of each image, which transforms a 3D point from the camera to the world coordinate frame, the intrinsic camera parameters $\mathbf{C}_i$, and a dense depth map $\mathbf{D}_i \in \mathbb{R}^{H_i \times W_i}$, whose scale is consistent with the poses.

Overview. A feed-forward model predicts all three quantities in one forward pass, providing a fast and robust initialisation (Section 3.1). We extract dense correspondences from its frozen features with a lightweight multiview decoder (Section 3.2) and refine the geometry with BA over the reprojection residuals and a depth prior from the feed-forward model (Section 3.3). The pipeline is summarised in Figure 2.

3.1 Robust initialisation

We use VGGT-Ω (Wang et al., 2026a) by default, sharing its forward pass between initialisation and matching. The backbone is frozen; only the matching decoder is trained (Section A). Our approach generalises across feed-forward backbones, with consistent final accuracy (Table 5).

3.2 Multiview match decoder

For each pair (i, j), the decoder estimates a dense correspondence map $\mathbf{P}_{i \to j} \in \mathbb{R}^{H_i \times W_i \times 2}$ with confidence $\mathbf{W}_{i \to j} \in \mathbb{R}^{[0,1], H_i \times W_i}$, where $\mathbf{P}^k_{i \to j}$ is the 2D coordinate in image j of pixel k in image i. We write $\mathbf{P}_i$ the coordinates of pixel centres in i, so that $(\mathbf{P}^k_i, \mathbf{P}^k_{i \to j})$ forms a correspondence. The target coordinate is not restricted to an integer value and is thus sub-pixel accurate.

Loop transformer. We adapt and extract the matching-relevant representation with a recurrent, weight-shared transformer (Dehghani et al., 2019; Burzio et al., 2026). The input is first linearly

projected. Then K iterations alternate a within-view block and a cross-view attention block (Wang et al., 2025a), with the same weights at every iteration. Learned per-channel gates, conditioned on the iteration coordinate via sinusoidal embeddings (Vaswani et al., 2017), scale each block's residual contribution. At training time, K is sampled per update from a Beta distribution ($K \in [3, 12]$), as in Déjà View (Burzio et al., 2026); at inference it is selectable within this range. All views share the adapter. Within-view computation is reused across the pairs incident to each image; only the global attention scales quadratically with the number of views.

Hierarchical decoding. We separate finding a corresponding region from locating the match precisely within it. Per-view convolutional heads produce unit-normalised 64-dimensional descriptors on a stride-4 grid, using pixel-shuffle upsampling (Shi et al., 2016) from the token resolution. A second finer per-view head upsamples the features by $3\times$ to a finer grid, whose cell centres align with those of the coarser grid. The global nearest neighbour centres a 13×13 window on this finer grid; a soft-argmax over descriptor similarities within the window yields sub-cell target coordinates. Both descriptor fields are computed once per view and reused across all pairs from that view. Pair-dependent work consists only of descriptor comparisons and coordinate averaging. Finally, a convolutional refiner with the RoMa v2 architecture (Edstedt et al., 2025) refines the coarse warp. Its resolution and number of passes can be set at runtime to trade speed for accuracy (Table 1).

Efficiency. Everything computed per view (the frozen cross-view context, the adapter, the descriptor heads, the local field and the refiner features) is shared across all edges from that view. Per-edge cost consists only of descriptor comparisons, coordinate decoding, and convolutional refinement. The coarse decode alone typically costs under 1 ms per edge; the refiner dominates, and its cost depends on the chosen resolution and strides (Figure 3).

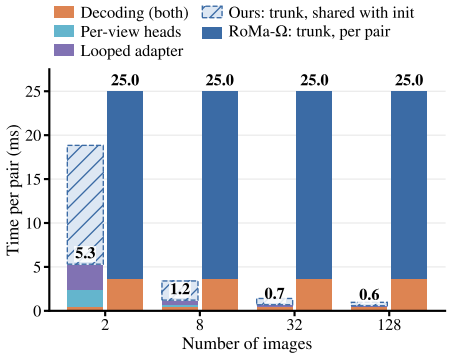


Figure 3: **Per pair average coarse matching cost.** Left ours, right RoMa-Ω.

3.3 Efficient dense bundle adjustment

From dense correspondences to tracks. For each pixel k in image i, the matcher produces continuous target coordinates $\mathbf{P}^k_{i\to j}$ in each neighbouring view $j \in \mathcal{N}_i$. These observations share one depth variable anchored to the source ray, forming a star-shaped track without keypoint detection or global track merging, as in LSPBA and DROID-SLAM (Woodford & Rosten, 2020; Teed & Deng, 2021). The source location is exact by construction, and the target coordinates keep their continuous, sub-pixel values; no snapping or quantisation is needed. Every pixel in a source image has the same set of target views, a regular computation pattern that parallelises efficiently on a GPU.

Cost function. Given an image pair (i, j), the correspondence of each pixel k in i yields a reprojection residual

$$\mathbf{r}_{ijk} = \Pi\Big({}_j\mathbf{T}_i\ \mathbf{D}_i^k\ \Pi^{-1}\Big(\mathbf{P}_i^k, \mathbf{C}_i\Big)\ ,\ \mathbf{C}_j\Big) - \mathbf{P}^k_{i\to j}, \tag{1}$$

where Π projects a 3D point onto the image plane given intrinsic parameters, with inverse Π^{-1}, and ${}_j\mathbf{T}_i = {}_w\mathbf{T}_j^{-1}\,{}_w\mathbf{T}_i$ is the relative pose between i and j. The total cost is

$$E = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{N}_i}\sum_{k\in H_i\times W_i} \mathbf{W}^k_{i\to j}\ \rho\Big(\|\mathbf{r}_{ijk}\|^2\Big) + \sum_{i\in\mathcal{I}}\sum_{k\in H_i\times W_i} \omega_i^k\ \rho_{\mathrm{p}}\left(\Big(\log\mathbf{D}_i^k - \log\hat{\mathbf{D}}_i^k - s_i\Big)^2\right), \tag{2}$$

where $\mathbf{W}^k_{i\to j} \in [0, 1]$ is the matching confidence, ρ is a robust loss (Cauchy (Triggs et al., 1999)) that down-weights gross matching errors, and $\hat{\mathbf{D}}_i$ is the feed-forward depth prediction. The second term is a log-depth prior that regularises weakly constrained rays, with weight $\omega_i^k = \lambda_{\mathrm{p}}\,\tilde{c}_i^k$ scaling a global strength λ_{p} by the feed-forward model's per-pixel confidence. Operating in log space with a per-image scale s_i supervises only relative depth and does not shrink the reconstruction.

Table 1: **ETH3D pose estimation** (Schöps et al., 2017). Relative-pose AUC (%) at 1/3/5/30° and mean seconds per scene. On these high-resolution scenes, upsampling and re-refining improves accuracy at the cost of speed.

Method	@1°	@3°	@5°	@30°	Time (s)
MASt3R-SfM (Duisterhof et al., 2025)	29.22	54.09	62.47	79.69	222
VGGT-Ω (Wang et al., 2026a) + VGGSfM BA (Wang et al., 2024a)	40.46	69.87	79.64	97.02	126
EPO (D'Urso et al., 2026) (VGGT-Ω init)	48.34	76.09	84.19	98.16	10.0
BLASt3R (Leroy et al., 2026)	54.59	74.92	80.53	91.09	21.8
DF-SfM (He et al., 2024)	71.91	77.39	78.59	81.17	959
GLUEMAP (Pan et al., 2026)	76.37	88.03	91.28	97.33	329
Dense-SfM (Lee & Yoo, 2025)	77.03	86.59	89.08	93.16	3712
COLMAP (Schonberger & Frahm, 2016) + LoMa-G (Nordström et al., 2026a)	78.36	87.20	89.31	93.08	189
VGGT-Ω (Wang et al., 2026a) (init)	22.97	57.27	70.93	95.75	1.5
DMap (fast)	86.97	95.26	97.13	99.90	**3.9**
DMap (HR)	**88.64**	**95.93**	**97.54**	**99.92**	8.8

Camera models. Beyond the pinhole model, we extend the BA to handle radial lens distortion (Fitzgibbon, 2001; Hagemann et al., 2023) and rolling-shutter effects (Hedborg et al., 2012). For uncalibrated video, we optimise a shared camera with focal length, principal point, and a radial coefficient k_1. For rolling-shutter cameras, we correct the correspondence coordinates with a rotation-only model estimated from neighbouring poses. The readout duration is selected by comparing BA merits across candidate values, including a global-shutter hypothesis.

Edge selection. Exhaustive matching over all image pairs is unnecessary. For unordered image sets, frustum overlap from the initial cameras scores candidate pairs; per-source target views are sampled without replacement using Laplacian edge-leverage scores (Spielman & Srivastava, 2011), which favour less redundant connections, at a budget of 8–16 edges per source image. For video sequences, uniformly spaced keyframes match against each other, while the remaining frames match against their nearest keyframes and temporal neighbours.

Optimisation. We solve this nonlinear least-squares problem with Levenberg–Marquardt (Levenberg, 1944; Marquardt, 1963). Each depth $\mathbf{D}_i^k$ is a scalar that couples only to the cameras observing pixel k, so the depth block of the Hessian is diagonal and is eliminated by one division per pixel via the Schur complement (Triggs et al., 1999; Teed & Deng, 2021). The reduced system has only about seven unknowns per image and is solved exactly by direct factorisation.

BA is often assumed to be the slow stage of reconstruction, yet ours takes less time than the matching that feeds it (Figure 4). The reason is regularity: all pixels of a source image are observed in the same target views on the same grid, so the observations form fixed-shape tensors rather than tracks of varying length that require scattered memory access. A fused kernel computes residuals and derivatives in one pass, and the Schur complement of all pixels of an image reduces to one batched matrix product. We cap the optimisation at 20 iterations, by which the BA has usually converged.

Depth uncertainty. The diagonal depth block of the Hessian supplies a per-pixel uncertainty estimate at no additional cost:

$$\Sigma_{\mathbf{D}_i}^{k\ -1} = \sum_{j \in \mathcal{N}_i} \mathbf{W}_{i \to j}^k \left(\frac{\partial \mathbf{r}_{ijk}}{\partial \mathbf{D}_i^k}^\top \frac{\partial \mathbf{r}_{ijk}}{\partial \mathbf{D}_i^k} \right). \tag{3}$$

This uncertainty is reduced with more views, a higher correspondence confidence, or better geometric constraints. It guides downstream filtering and fusion (Section C): the most certain fraction of the dense cloud is both more accurate and more complete than published sparse reconstructions (Figure 5).

4 Experiments

4.1 Pose estimation

We evaluate pose estimation on four benchmarks: ETH3D (Schöps et al., 2017) (25 high-resolution unordered scenes, relative-pose AUC), Texture-Poor from DFSfM (He et al., 2024) (840 test bags,

Table 2: **Texture-Poor** (He et al., 2024). Mean-of-per-bag relative-pose AUC (%) at 3/5/10° and mean seconds per bag. Weak texture challenges matching; uncropped images are out-of-distribution for most feed-forward models.

Method	@3°	@5°	@10°	Time (s)
MASt3R-SfM (Duisterhof et al., 2025)	7.10	18.08	42.44	—
LoFTR+DF-SfM (He et al., 2024)	26.07	35.77	45.43	—
RoMa (Edstedt et al., 2024)+Dense-SfM (Lee & Yoo, 2025)	49.94	66.23	81.41	—
MV-RoMa (Lee et al., 2026)	51.79	66.77	81.74	—
EPO (D'Urso et al., 2026) (VGGT-Ω init)	6.11	12.87	30.38	11.2
GLUEMAP (Pan et al., 2026) (VGGT-Ω)	17.74	26.76	45.23	49.9
VGGT-Ω (Wang et al., 2026a) + VGGSfM BA	44.45	61.31	77.88	27.1
BLASt3R (Leroy et al., 2026)	58.61	73.37	86.05	12.0
COLMAP (Schonberger & Frahm, 2016) + LoMa-G (Nordström et al., 2026a)	59.42	74.02	86.42	14.4
VGGT-Ω (init)	3.68	12.21	37.80	0.3
DMap (ours)	**66.33**	**78.93**	**89.33**	**1.6**

Table 3: **TUM RGB-D** (Sturm et al., 2012). ATE RMSE (cm) and average time per sequence. This dataset has strong rolling shutter; modelling it explicitly improves accuracy beyond the calibrated setting, where the provided undistorted frames make rolling shutter harder to compensate.

	Method	360	desk	desk2	floor	plant	room	rpy	teddy	xyz	**Avg.**	Time (s)
Calibrated	ORB-SLAM3 (Campos et al., 2021)	×	1.7	21.0	×	3.4	×	×	×	0.9	N/A	
	DPV-SLAM (Lipson et al., 2024)	11.2	1.8	2.9	5.7	2.1	33.0	3.0	8.4	1.0	7.6	25
	DROID-SLAM (Teed & Deng, 2021)	11.1	1.8	4.2	2.1	1.6	4.9	2.6	4.8	1.2	3.8	**23**
	BLASt3R (Leroy et al., 2026)	5.2	1.4	2.4	2.0	1.3	5.1	2.0	2.6	0.9	2.5	384
	VidMap (Pataki et al., 2026)	5.5	1.6	2.5	2.1	1.1	3.6	2.1	3.2	1.1	2.5	214
	DMap (global shutter)	5.6	1.3	2.0	1.7	0.9	3.6	1.9	2.6	0.8	2.3	29
	DMap (rolling shutter)	**1.7**	**1.0**	**1.0**	**1.0**	**0.8**	**1.9**	**0.6**	**1.6**	**0.5**	**1.1**	33
Uncalibrated	CUT3R (Wang et al., 2025b)	17.4	59.2	54.6	66.2	46.7	91.1	5.1	84.5	12.9	48.6	**7**
	MASt3R-SLAM (Murai et al., 2025)	7.0	3.2	5.5	5.6	3.5	11.8	4.1	11.6	2.0	6.0	29
	VGGT-SLAM 2.0 (Maggio & Carlone, 2026)	5.1	4.0	5.3	11.1	4.5	7.3	2.6	4.5	10.6	6.1	9
	LingBot-Map (Chen et al., 2026a)	8.7	2.8	3.9	5.5	2.9	5.4	2.4	3.4	1.5	4.1	20
	AMB3R (Wang & Agapito, 2025b)	4.5	1.8	2.8	3.7	3.4	5.2	2.6	5.2	1.1	3.4	54
	ViPE (Huang et al., 2025)	8.0	2.8	4.2	9.2	2.8	7.0	2.5	6.0	1.3	4.9	149
	BLASt3R (Leroy et al., 2026)	5.7	1.8	2.9	1.8	2.2	5.8	2.5	3.2	1.1	3.0	455
	VidMap (Pataki et al., 2026)	5.5	1.7	2.6	2.5	1.1	6.8	2.1	3.3	1.1	3.0	420
	VGGT-Ω-Chunk (Wang et al., 2026a) (init)	9.3	3.8	4.7	8.4	3.3	18.1	2.1	4.4	1.0	6.1	15
	DMap (global shutter)	5.2	1.3	2.0	1.4	0.9	3.8	2.0	2.4	1.0	2.2	30
	DMap (rolling shutter)	**1.4**	**0.9**	**1.0**	**0.9**	**0.8**	**1.3**	**0.6**	**1.5**	**0.5**	**1.0**	34

relative-pose AUC), TUM RGB-D *fr1* (Sturm et al., 2012) (9 indoor sequences, ATE RMSE, calibrated and uncalibrated), and ETH3D-SLAM (Schöps et al., 2019) (8 sequences, ATE RMSE). We follow the BLASt3R (Leroy et al., 2026) protocol for the video benchmarks and self-calibrate with camera sharing unless ground-truth intrinsics are provided. All timings are on one RTX 5090.

4.2 Triangulation

Following DFSfM (He et al., 2024), we fix the ground-truth poses and intrinsics on the 13 ETH3D training scenes (Schöps et al., 2017), and score the triangulated points against the laser scans at 1, 2 and 5 cm with the official ETH3D evaluator. Every point carries a predicted uncertainty (Equation (3)), so we report a curve that keeps the most certain fraction (Figure 5). The direct BA curve already dominates every published sparse triangulation method at every threshold. With fusion (Section C), the dense cloud also surpasses COLMAP MVS (Schönberger et al., 2016). The dense matches come almost for free, since the matcher already runs for pose estimation, and the additional refinement and fusion take about 3 s per scene. When DMap estimates its own cameras instead of using the ground truth, it still surpasses existing triangulation methods (Section D).

4.3 Ablations

Backbone ablation. Our decoder generalises across frozen feed-forward trunks. We train the same head on Pi3X (Wang et al., 2026b) and DVLT (Burzio et al., 2026) and run the same high-resolution refinement and dense BA. From a common VGGT-Ω initialisation, all three reach nearly the same

Table 4: **ETH3D-SLAM** (Schöps et al., 2019). ATE RMSE (cm) and average time per sequence. Fast feed-forward methods are inaccurate; the most accurate methods are an order of magnitude slower.

Method	cables1	camshake1	einstein1	plant1	plant2	sofa1	table3	table7	**Avg.**	Time (s)
VGGT-SLAM 2.0 (Maggio & Carlone, 2026)	41.3	2.7	4.5	3.8	1.9	7.6	4.7	6.2	9.1	**4**
LingBot-Map (Chen et al., 2026a)	19.6	3.0	4.1	2.0	1.3	5.7	4.5	4.2	5.6	12
AMB3R (Wang & Agapito, 2025b)	2.8	2.6	1.8	0.8	0.9	3.0	5.9	2.8	2.6	33
ViPE (Huang et al., 2025)	1.3	2.1	1.1	0.3	0.4	1.5	1.2	4.4	1.5	104
BLASt3R (Leroy et al., 2026)	1.5	1.5	0.9	0.4	0.3	1.7	1.9	1.8	1.2	401
VidMap (Pataki et al., 2026)	1.3	1.4	0.4	0.3	**0.2**	**0.3**	0.5	0.6	0.6	298
VGGT-Ω-Chunk (Wang et al., 2026a) (init)	6.5	1.7	1.4	0.5	0.5	4.0	2.2	2.2	2.4	12
DMap (ours)	**0.9**	**1.1**	**0.3**	**0.2**	**0.2**	0.5	**0.4**	**0.4**	**0.5**	23

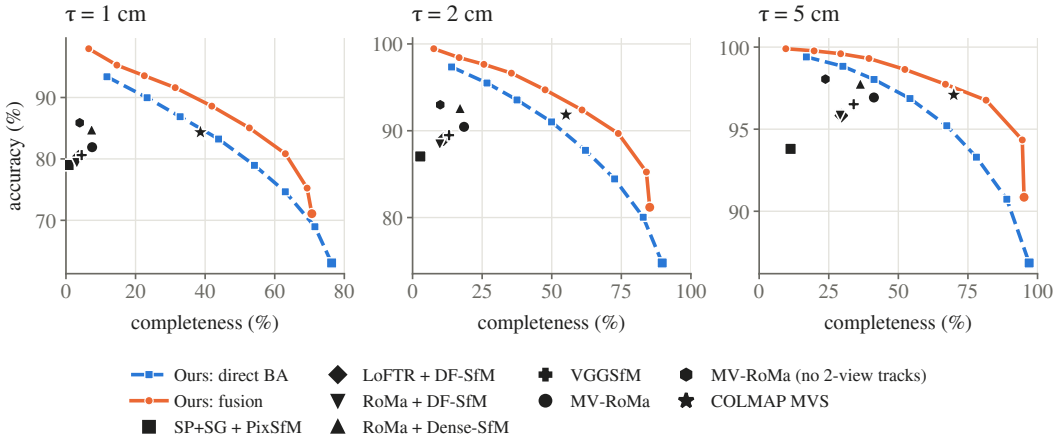


Figure 5: **Triangulation on ETH3D with known poses.** Accuracy against completeness at 1, 2 and 5 cm. Direct BA already dominates every published SfM triangulation method at every threshold. With fusion, the dense cloud suffers less from limited track length and surpasses COLMAP MVS (Schönberger et al., 2016).

accuracy, within 0.2 AUC@1° (Table 5). Each trunk can also supply its own initialisation: DVLT then matches VGGT-Ω (88.54 vs. 88.60), so the whole pipeline runs on a single backbone. Pi3X drops by 2.9; one scene (*exhibition_hall*) explains most of it, where its predicted poses are essentially wrong and BA converges to a wrong minimum. Dense BA makes the result accurate but relies on the feed-forward model for robustness: it does not recover a grossly wrong initial pose.

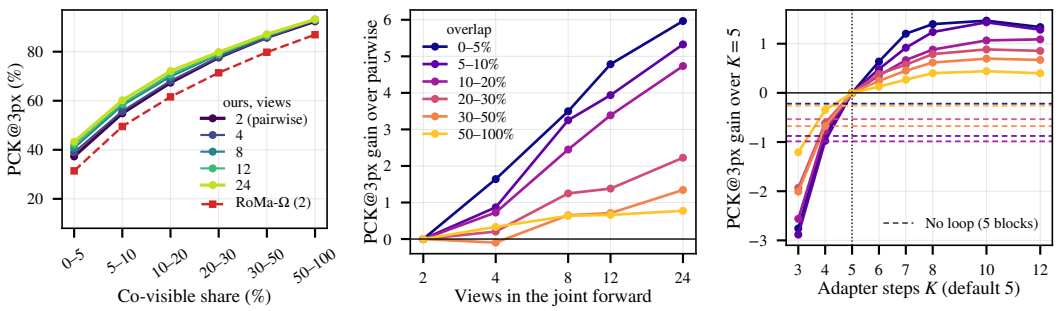
Multiview context on MegaDepth-X. MegaDepth-X (Li et al., 2026) samples difficult pairs from a long-tailed distribution of viewpoint changes; we evaluate on its 127 held-out test scenes (Figure 6). Context views improve accuracy in every overlap bin, most at low overlap. Increasing the adapter from $K{=}5$ to $K{=}8$ steps adds 0.9 PCK@3, and a non-looped variant with five separate blocks, trained identically, generalises worse, which supports weight sharing.

4.4 Runtime Analysis

Figure 4 breaks down the per-video cost on ETH3D-SLAM: dense BA is the cheapest stage, and matching costs about as much as the feed-forward pass.

5 Conclusion

We have shown that dense geometric refinement of feed-forward 3D predictions can be both fast and accurate. The correspondence information is already in the feed-forward features; a lightweight decoder extracts it by sharing per-view computation across all edges. The regular pixel grid and fixed view graph, without track merging, give aligned memory access that makes problem construction memory-bound, so dense BA with millions of observations typically costs less than the matching



(a) PCK@3px by overlap (b) Gain vs. context views (c) Gain vs. adapter steps K

Figure 6: **Multiview context on MegaDepth-X (Li et al., 2026).** Refined PCK@3px at 518^2. (a, b) Context views improve accuracy across all overlap bins, with the largest gains at low overlap. (c) More adapter steps improve accuracy; a non-looped variant with five separate blocks (dashed) generalises worse.

Table 5: **Backbone ablation** on ETH3D. The same decoder, refiner and dense BA on three frozen trunks, started either from a common VGGT-Ω initialisation or from each trunk's own prediction. Pose AUC (%). With a common init the final accuracy is nearly identical; with own init, Pi3X drops because one scene's poses are grossly wrong and BA cannot recover.

	VGGT-Ω init			Own init		
Trunk	@1°	@3°	@5°	@1°	@3°	@5°
VGGT-Ω (Wang et al., 2026a)	**88.60**	**95.91**	**97.53**	**88.60**	**95.91**	**97.53**
Pi3X (Wang et al., 2026b)	88.40	95.68	97.37	85.47	93.15	94.88
DVLT (Burzio et al., 2026)	88.45	95.80	97.44	88.54	95.82	97.46

that produces them. Representing the geometry as explicit reprojection residuals, rather than implicit network weights, naturally accommodates lens distortion, rolling shutter, and calibrated depth uncertainty without retraining.

DMap consistently improves accuracy across four pose benchmarks, three backbones, and a reconstruction evaluation, setting the state of the art on ETH3D, TUM, and ETH3D-SLAM while being 13–17× faster than BLASt3R on video. A component-swap ablation confirms that both the matcher and the dense BA contribute.

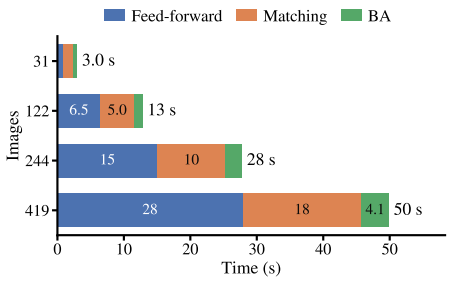


Figure 4: **Time by stage**, measured on one RTX 5090 for 4 ETH3D-SLAM videos (Table 4). With our accelerated dense BA, it is the cheapest stage (8–16 % of the total) and never a bottleneck. The lightweight matching decoder makes matching comparable to, and even cheaper than, the feed-forward backbone pass itself.

AI USE STATEMENT

In this work, we used generative AI tools for polishing, retrieval and discovery, research ideation and execution, and to draft sections of the paper. We have not used generative AI tools for generating synthetic datasets or proving mathematical claims. and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

No ethics concerns with image matching.

REPRODUCIBILITY STATEMENT

The full code and training pipeline will be made publicly available.

REFERENCES

Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European Conference on Computer Vision*, 2010.

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, 2022.

Jelena Bratulić, Sudhanshu Mittal, Thomas Brox, and Christian Rupprecht. On geometric understanding and learned priors in feed-forward 3D reconstruction models. *arXiv preprint arXiv:2512.11508*, 2025. URL https://arxiv.org/abs/2512.11508.

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. URL https://arxiv.org/abs/2605.30215.

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021.

Lin-Zhuo Chen, Jian Gao, Shangzhan Zhang, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. LingBot-Map: Geometric context transformer for streaming 3D reconstruction. *arXiv preprint arXiv:2604.14141*, 2026a. URL https://arxiv.org/abs/2604.14141.

Yutong Chen, Yiming Wang, Xucong Zhang, Sergey Prokudin, and Siyu Tang. GGPT: Geometry-grounded point transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b. URL https://arxiv.org/abs/2603.11174.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3D modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443*, 2025.

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.

Mattia D'Urso, Christian Sormann, Mattia Rossi, and Friedrich Fraundorfer. EPO: Boosting 3D foundation models with edge-based pose optimization. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2607.00579.

Mihai Dusmanu, Johannes L. Schönberger, and Marc Pollefeys. Multi-view optimization of local feature geometry. In *European Conference on Computer Vision*, 2020.

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.

Andrew W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2001.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, 2024.

Annika Hagemann, Moritz Knorr, and Christoph Stiller. Deep geometry-aware camera self-calibration from video. In *IEEE/CVF International Conference on Computer Vision*, pp. 3438–3448, 2023.

Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2306.15669.

Johan Hedborg, Per-Erik Forssén, Michael Felsberg, and Erik Ringaby. Rolling shutter bundle adjustment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1434–1441, 2012.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.

Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2):517–547, 2021.

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*, 2026.

JongMin Lee and Sungjoo Yoo. Dense-SfM: Structure from motion with dense consistent matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2501.14277.

Jongmin Lee, Seungyeop Kang, and Sungjoo Yoo. MV-RoMa: From pairwise matching into multi-view track reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7446–7456, 2026. URL https://arxiv.org/abs/2603.27542.

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.

Vincent Leroy, Philippe Weinzaepfel, Lojze Zust, Yohann Cabon, and Jérôme Revaud. BLASt3R: Bundle adjustment of any image set with multi-view matching and monocular priors. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2609.05210.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

Yuan Li, Yuanbo Xiangli, Hadar Averbuch-Elor, Noah Snavely, and Ruojin Cai. Long-tail internet photo reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 417–426, 2026.

Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10486–10496, 2025.

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

Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual SLAM. In *European Conference on Computer Vision*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Dominic Maggio and Luca Carlone. VGGT-SLAM 2.0: Real-time dense feed-forward scene reconstruction. In *Robotics: Science and Systems (RSS)*, 2026. URL https://arxiv.org/abs/2601.19887.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: dense RGB SLAM optimized on the SL(4) manifold. In *Advances in Neural Information Processing Systems*, 2025.

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, pp. 2320–2327, 2011.

David Nordström, Johan Edstedt, Georg Bökman, Jonathan Astermark, Anders Heyden, Viktor Larsson, Mårten Wadenbäck, Michael Felsberg, and Fredrik Kahl. LoMa: Local feature matching revisited. *arXiv preprint arXiv:2604.04931*, 2026a. URL https://arxiv.org/abs/2604.04931.

David Nordström, Xinyue Zhang, Thibaut Loiseau, Vincent Lepetit, and Fredrik Kahl. RoMa-Ω: What feed-forward 3D models know about image matching. *arXiv preprint arXiv:2609.09507*, 2026b. URL https://arxiv.org/abs/2609.09507.

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

Linfei Pan, Johannes Schönberger, and Marc Pollefeys. Global structure-from-motion meets feedforward reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2605.26103.

Zador Pataki, Paul-Edouard Sarlin, Johannes L Schönberger, and Marc Pollefeys. Mp-sfm: Monocular surface priors for robust structure-from-motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21891–21901, 2025.

Zador Pataki, Paul-Edouard Sarlin, and Marc Pollefeys. VidMap: Exploiting temporal structure for video-based structure-from-motion. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2607.27194.

Jie Ren, Wenteng Liang, Ran Yan, Luo Mai, Shiwen Liu, and Xiao Liu. MegBA: A GPU-based distributed library for large-scale bundle adjustment. In *European Conference on Computer Vision*, 2022.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *IEEE/CVF International Conference on Computer Vision*, 2021.

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.

Johannes L Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518, 2016.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 134–144, 2019.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016.

Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.

Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations*, 2019.

Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. In *International Conference on Learning Representations*, 2020a.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020b.

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2108.10869.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *International Workshop on Vision Algorithms*, 1999.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Khiem Vuong, Anurag Ghosh, Deva Ramanan, Srinivasa Narasimhan, and Shubham Tulsiani. AerialMegaDepth: Learning aerial-ground reconstruction and view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *International Conference on 3D Vision*, 2025a.

Hengyi Wang and Lourdes Agapito. AMB3R: Accurate feed-forward metric-scale 3D reconstruction with backend. *arXiv preprint arXiv:2511.20343*, 2025b. URL https://arxiv.org/abs/2511.20343.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21686–21697, 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.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21486–21499, 2026a. URL https://arxiv.org/abs/2605.15195.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024b.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2507.13347.

Oliver J. Woodford and Edward Rosten. Large scale photometric bundle adjustment. In *British Machine Vision Conference*, 2020. URL https://arxiv.org/abs/2008.11762.

Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore bundle adjustment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3057–3064, 2011.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Yiwen Zhang, Joseph Tung, Ruojin Cai, David Fouhey, and Hadar Averbuch-Elor. Emergent extreme-view geometry in 3D foundation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36411–36421, 2026.

Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, Sebastian Scherer, and Wenshan Wang. UFM: A simple path towards unified dense correspondence with flow. In *Advances in Neural Information Processing Systems*, 2025.

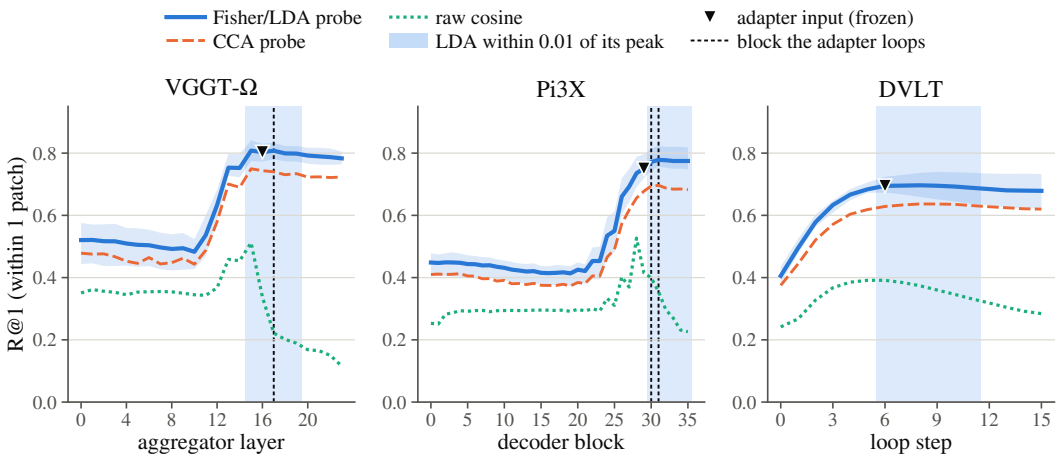


Figure 7: **Which trunk block to read.** R@1 of three linear read-outs of frozen patch tokens for every block of each trunk, the share of covisible query patches whose nearest neighbour lies within one patch of the ground-truth match. Curves average MegaDepth, ScanNet++ and BlendedMVS (50 scene-disjoint pairs each); the envelope around the Fisher/LDA curve spans the three datasets. Blue band: blocks within 0.01 of the Fisher/LDA peak. Triangle: frozen input of the matching adapter; dashed lines: the block (VGGT-Ω) or block pair (Pi3X) the looped adapter is initialised from; DVLT's adapter continues DVLT's own loop. Raw cosine peaks at or below the start of the plateau and declines after it, while the discriminative probe stays close to its peak through the last block, so a cosine sweep would pick Pi3X block 28, outside the plateau.

A Initialising and training the match decoder

The match decoder reads one block of a frozen trunk and starts from a copy of that block (Section A.1). Section A.2 gives the data, losses and optimisation of the VGGT-Ω decoder used throughout the paper.

A.1 Choosing the trunk block

The matching adapter (Section 3.2) reads one block of a frozen trunk, and a block whose tokens do not separate corresponding patches from the rest leaves the decoder little to recover. Ranking blocks by the raw cosine similarity of their tokens, the usual criterion, conflates the information a block holds with the basis it is expressed in. We instead rank blocks with a closed-form discriminative probe. For every block we fit a Fisher linear discriminant on frozen patch tokens: it whitens by the covariance of token differences between corresponding patches and keeps the 128 directions along which a query patch differs most from non-matching patches of the other image. The probe is fitted on 1600 image pairs and scored on 50 scene-disjoint pairs from each of MegaDepth, ScanNet++ and BlendedMVS, two views per forward pass, with every trunk at the resolution its matcher trains at. The score is the share of covisible query patches whose nearest neighbour under the probe lies within one patch of the ground-truth match (R@1), and the share more than eight patches away, the gross errors that bundle adjustment cannot reject. For reference we also score canonical correlation analysis, a linear map fitted without negatives, and raw cosine.

Figure 7 shows a late plateau on every trunk: layers 15 to 19 of VGGT-Ω, blocks 30 to 35 of Pi3X and loop steps 6 to 11 of DVLT lie within 0.01 R@1 of the probe's peak. Raw cosine ranks the same blocks differently. On VGGT-Ω it peaks at the first layer of the plateau with R@1 0.51, against 0.81 for the probe on that layer; on Pi3X it peaks below the plateau, at block 28. The best block differs between the three datasets but stays inside the plateau, so R@1 alone does not single out a block. On VGGT-Ω we break the tie by the gross-outlier rate: layer 17 lowers it from 0.81% at layer 15 to 0.46% and has the fewest gross outliers on all three datasets. On Pi3X we take block 31, the R@1 peak of the three-dataset mean (0.78); it is a global-attention block, so the representation has just been mixed across views.

Table 6: **Training settings** of the VGGT-Ω match decoder. With the trunk frozen, 51.1 M parameters train in two days on four GPUs.

Setting	Value
Trainable parameters	51.1 M, of which 26.7 M in the looped adapter; trunk frozen
Views per sample	N with $p(N) \propto N^{-1/2}$ on $[2, 32]$
Samples per GPU and step	$\lfloor 32/N \rfloor$, about 32 images; up to 64 images for $N \leq 5$
GPUs, accumulation	4 RTX 5090, none; about 128 images per step
Input size	aspect ratio kept, about 1024 tokens of 16^2 pixels (e.g. 416×624)
Optimiser	AdamW, weight decay 0.01, gradient norm clipped at 1
Learning rate	4×10^{-4}
Schedule	linear warm-up for 1k steps, constant to 72k, $1 - \sqrt{t}$ decay to 0 at 90k
Steps	90k
Adapter steps	$K = 3 + \mathrm{round}(9b)$, $b \sim \mathrm{Beta}(2, 5)$, $\mathbb{E}[K] = 5.6$, one draw per step
Precision	bfloat16 autocast in trunk and adapter, float32 heads and losses
Weight averaging	exponential moving average, decay 0.999; final average evaluated
Wall-clock	49 h, 1.7–1.8 s per step, including validation

The looped adapter is a copy of the selected block (on Pi3X, of the local and global pair that ends at it), applied to the frozen output just before that block and initialised as an identity. Its input is therefore layer 16 of VGGT-Ω, inside the plateau, and block 29 of Pi3X, just below it (R@1 0.75, with 1.34% gross outliers against 0.72% at block 31). DVLT is itself a looped transformer, so its adapter continues DVLT's own loop from step 6, inside its plateau (R@1 0.695 against 0.697 at the peak, step 8).

A.2 Training recipe

Parameters. The VGGT-Ω trunk stays frozen through layer 16, and the looped adapter starts from layer 17 as described above. The adapter, the stride-4 and local descriptor heads and the token-level matchability head hold 51.1 M trainable parameters, 26.7 M of them in the adapter. Refinement uses the released RoMa-Ω refiner (Nordström et al., 2026b) without further training, and training never differentiates through bundle adjustment.

Data. The decoder trains on seven datasets with depth maps and camera poses: MegaDepth (Li & Snavely, 2018), AerialMegaDepth (Vuong et al., 2025), BlendedMVS (Yao et al., 2020), Hypersim (Roberts et al., 2021), ScanNet++ (Yeshwanth et al., 2023), TartanAir (Wang et al., 2020) and Map-free (Arnold et al., 2022). Each dataset reserves up to eight evenly spaced scenes for validation. MegaDepth also drops the two scenes of the MegaDepth-1500 test set (Sun et al., 2021), and MegaDepth and AerialMegaDepth drop the eight scenes that show landmarks of the IMC benchmark (Jin et al., 2021). A training sample is a window of N consecutive views from a 128-view random walk over the covisibility graph of one scene. The index holds 676k walks, at most 4000 per scene: 68k from MegaDepth, 115k from AerialMegaDepth, 98k from BlendedMVS, 42k from Hypersim, 125k from ScanNet++, 158k from TartanAir and 70k from Map-free. Datasets carry equal weight, and a walk from a scene with n_s walks is drawn with probability proportional to $n_s^{-0.75}$, so the share of a scene grows only as $n_s^{1/4}$ and large scenes do not dominate. N is drawn per micro-batch with probability proportional to $N^{-1/2}$ on $[2, 32]$, which covers the pairwise case and the many-view context of deployment. Images keep their aspect ratio and are resized to about 1024 tokens of 16×16 pixels, for example 416×624 for a 2:3 image. The only augmentation is a horizontal flip applied to all views of a sample together.

Batch. Each of four GPUs processes $\lfloor 32/N \rfloor$ samples per step, about 32 images. A sample with few views contains few pairs, so for $N \leq 5$ the image budget grows with the inverse square root of the number of ordered pairs, up to 64 images at $N{=}2$. One step therefore covers about 128 images, without gradient accumulation. The losses supervise every ordered pair of views within a sample.

Losses. The global loss is the cross-entropy between the softmax of stride-4 descriptor similarities over all target cells and a bilinear target on the four cells around the ground-truth match. The local

loss applies the same bilinear target on the $3\times$ finer lattice inside the 13×13 window, and adds the generalised Charbonnier penalty of RoMa (Edstedt et al., 2024) (α=0.5, c=10^{-3} in normalised image coordinates) on the soft-argmax coordinate. Both terms use 1024 covisible query cells per pair; a query whose ground truth falls outside its window keeps only the global loss. Auxiliary correspondence losses on the token grid train the coarse branch of the head, and a covisibility loss with weight 0.01 trains the matchability head; every other term has weight 1. Ground-truth correspondences and covisibility come from the dataset depths and cameras.

Optimisation. AdamW (Loshchilov & Hutter, 2019) runs 90k steps with the warmup-stable-decay schedule of Hägele et al. (2024) (Table 6). Unlike a cosine schedule, its constant phase fixes no horizon in advance: training continues at a constant rate until the decay starts, so the run can be extended without a restart. The adapter step count K is drawn once per step and shared by all GPUs, which keeps K selectable at inference (Section 3.2). We evaluate the exponential moving average of the weights after the final step rather than an intermediate checkpoint. Training takes 49 hours on four RTX 5090 GPUs, including periodic validation.

Other trunks. The Pi3X and DVLT decoders of Table 5 use the same data, batch, losses and optimiser for 30k steps with a cosine schedule. Their 14-pixel patches take 518-pixel images at 1369 tokens, so they see about as many pixels as VGGT-Ω at 1024 tokens.

B Depth uncertainty

Calibration without ground truth. The uncertainty σ of Equation (3), expressed in log depth, is the inverse curvature of the cost in each depth. Its scale assumes one squared pixel of reprojection noise per unit of matching confidence, while the actual noise differs between scenes and between pixels. We calibrate it from quantities that every reconstruction has, without ground truth. Let κ be the confidence-weighted variance of all reprojection residuals of a scene, and B the confidence-weighted sum of squared residuals of one pixel over its n observations. The residual variance of the pixel, shrunk towards that of the scene, is $s^2 = (2\kappa + B)/(2n + 1)$. The calibrated uncertainty is

$$\tilde{\sigma} = \sigma \, \frac{s}{\sqrt{\kappa}}, \qquad \sigma_{\mathrm{cal}} = \sqrt{\kappa} \, \tilde{\sigma}_{\mathrm{med}} \left(\frac{\tilde{\sigma}}{\tilde{\sigma}_{\mathrm{med}}} \right)^{\beta}, \tag{4}$$

where $\tilde{\sigma}_{\mathrm{med}}$ is the median over the scene. For $\beta = 1$, $\sigma_{\mathrm{cal}} = \sigma s$, the uncertainty of the solver rescaled by the measured noise of the pixel. The exponent β corrects the shape. A pixel of view i, lifted with its depth and projected into a neighbouring view j, lands at a depth that differs from the depth map of j by Δ in log depth. If the errors of the two views were independent, Δ would have variance $a^2\sigma_i^2 + \sigma_j^2$, where a is the derivative of the depth in view j with respect to the depth in view i. We fit $\sigma' = e^{\alpha}\tilde{\sigma}^{\beta}$ per scene so that Δ, normalised by this prediction, has unit robust standard deviation in every decile of σ. Only β is kept. The errors of neighbouring views are correlated, so the differences underestimate the level α, and $\sqrt{\kappa}$ sets the level instead. The consolidation of Section C weights the depths with σ_{cal}.

Evaluation. We compare σ_{cal} of the direct BA depth with the laser depth on the 13 ETH3D training scenes, with ground-truth poses as in Section 4.2 and 100k pixels per scene. For a calibrated uncertainty, the normalised log-depth error e/σ has unit standard deviation. Its robust standard deviation is 1.04 with σ_{cal} and 0.70 with the uncalibrated σ, which is thus too large for typical pixels. Calibration also improves the ranking, raising the rank correlation between the uncertainty and $|e|$ from 0.45 to 0.55. Figure 8 shows the densities. The median error follows the median of a calibrated Gaussian, $0.674\,\sigma_{\mathrm{cal}}$, over two orders of magnitude of σ_{cal}. The tail is heavier than Gaussian, with 82% of the errors within $1.96\,\sigma_{\mathrm{cal}}$ instead of 95%. Figure 9 shows the depth and σ_{cal} for ten scenes.

C Depth fusion

The direct BA gives each view its own depth map, and weakly constrained rays keep errors that the other views could correct. Fusion consolidates the depth maps with the cameras fixed, without new matching or a volumetric representation. It takes the calibrated uncertainty σ_{cal} of Section B, relative

to depth, and returns fused depth maps with an uncertainty that also encodes visibility. Every step operates on log depth, so each tolerance below is a fraction of depth.

Prior blend. The BA regularises depth with the feed-forward prior at a fixed weight (Section 3.3), although the accuracy of the prior varies between views. We measure it per view as the robust spread $\sigma_{\mathrm{p},i}$ (1.4826 times the median absolute deviation) of the difference between the BA depth and the scale-aligned prior over the pixels with $\sigma_{\mathrm{cal}} < 1\%$. Each pixel then minimises, independently of its neighbours, a quadratic data term around the BA log depth x_0 with curvature $\sigma_{\mathrm{cal}}^{-2}$ plus a Cauchy prior term (scale 0.3) with weight $\tilde{c}/\sigma_{\mathrm{p},i}^2$, where $\tilde{c}$ is the confidence of the prior. A linear term cancels the gradient of the original BA prior at x_0, so the blend replaces that prior rather than adding to it. Five reweighting iterations suffice, since each is a closed-form step per pixel.

Projective fusion. View j is a candidate for view i when more than 1% of a stride-16 subsample of pixels of i, projected at their depths, land on the surface of j within 5%. A pixel of i at depth d looks up j by projecting its 3D point, which has depth z_j in j, and reading the depth map $\tilde{d}_j$ bilinearly. The look-up is refused if the point is behind j or outside the image, if the ray is nearly parallel to the image plane of j, or if the bilinear and nearest reads differ by more than 2%, which marks a depth edge. Four Newton steps along the ray move d to where it meets the surface of j; the intersection gives an estimate d^* whose uncertainty is that of j transferred onto the ray of i. The residual $\ell = \log(\tilde{d}_j/z_j)$ at the original depth is divided by the gain $g = \partial \log z_j / \partial \log d$, so that it measures a fraction of the depth of the pixel from any viewpoint. With the uncertainties σ of the pixel and s of the hit, the gate is

$$\tau = \mathrm{clamp}\Big(3\sqrt{\sigma^2 + s^2 + 0.002^2},\ 0.05,\ 0.25\Big)\,. \tag{5}$$

The hit supports the pixel if $|\ell/g| < \tau$, violates it if $\ell/g \geq \tau$, since j then sees through the point, and is occluded if $\ell/g \leq -\tau$. The pixel's own estimate and every converged supporting hit are candidates. The four with the smallest uncertainty are kept, and a candidate votes only if its uncertainty is at most twice the smallest $\sigma_{\min}$, so a coarse estimate from a distant view never overrides a close one. The fused depth is the weighted median of the votes, with weights $w_v = (\sigma_v^2 + \sigma_{\mathrm{f}}^2)^{-1}$ and $\sigma_{\mathrm{f}} = 5 \cdot 10^{-4}$, and its uncertainty is

$$\sigma_{\mathrm{fused}} = \max\Big(\Big(\textstyle\sum_v w_v\Big)^{-1/2},\ \sigma_{\min}/\sqrt{2}\Big)\,. \tag{6}$$

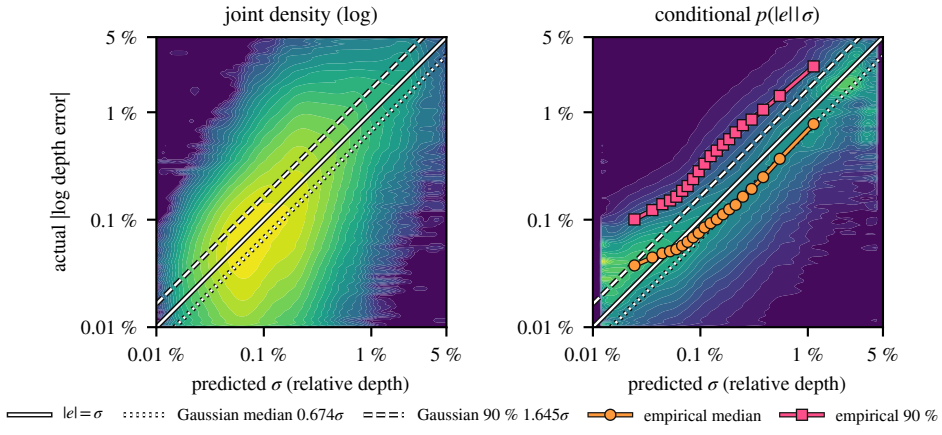


Figure 8: **Calibration of the depth uncertainty on ETH3D.** Calibrated uncertainty σ_{cal} of the direct BA depth against its log-depth error $|e|$ with respect to the laser scan, 100k pixels per scene over the 13 training scenes. The window $\sigma_{\mathrm{cal}}, |e| \leq 5\%$ holds 97.0% of the pixels; 2.4% have $\sigma_{\mathrm{cal}} > 5\%$ and 0.7% have $|e| > 5\%$. Left: joint density. Right: density of $|e|$ given σ_{cal}, with the empirical median and 90% quantile per bin of σ_{cal}. White lines: $|e| = \sigma$, and the median (0.674σ) and 90% quantile (1.645σ) of a calibrated Gaussian. The median error follows the calibrated prediction across two orders of magnitude, while the 90% quantile lies above it because the error tail is heavier than Gaussian.

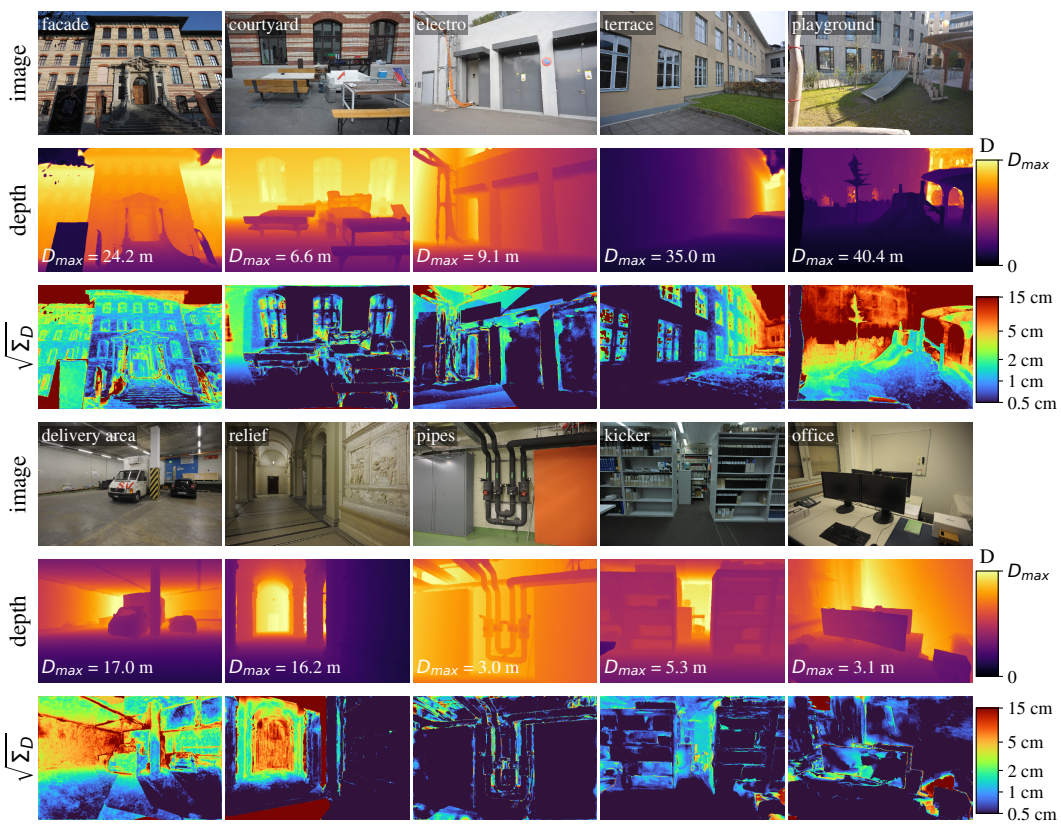


Figure 9: **Dense depth and calibrated uncertainty on ETH3D.** For ten training scenes: an image, the direct BA depth with ground-truth poses, scaled to the 99.5% quantile D_{max} of each view, and the calibrated standard deviation $\sqrt{\Sigma_D} = \sigma_{\text{cal}}\, D$ in metres. The uncertainty is low on surfaces that many views observe at a wide baseline, and high on windows, at sky borders and on distant surfaces. Filtering by uncertainty removes these regions first.

The floor limits the gain from agreement between views whose errors are correlated to a halving of the smallest variance.

Rescue. A pixel whose depth is wrong finds no support at that depth, so the vote cannot move it. For every pixel without a supporting hit, a line search places 96 depth hypotheses log-uniformly within a factor of 1.35 around its depth. Each hypothesis looks up every candidate view once, counting only constrained pixels of that view, and scores the views that support it within 2% minus those that see through it. The best hypothesis is accepted if at least two views support it and the score is positive; the pixel is then fused again from that depth with its own estimate excluded. A pixel that moves by more than 0.3 in log depth from its blended depth keeps the blended depth. The rescue leaves the uncertainty unchanged.

Visibility. A final pass casts every pixel into the fused depth maps of its candidate views with a fixed 1% gate. If any view sees a surface at least 1% behind the point, that view has looked through it, and the point receives $\sigma = \infty$. Such points usually lie close to the surface but are far more often grossly wrong than the others, so visibility enters as an outlier indicator rather than as a wider variance. Filtering by uncertainty then removes them without a separate threshold.

Point cloud and curve. Every valid pixel is back-projected with the fixed cameras. The points of a scene are ranked by their metric uncertainty σ times the distance to the camera centre, and the curve of Figure 5 keeps the most certain fraction, then all points with finite uncertainty, then all points. The

fused σ orders the points well but is not a calibrated interval, so we use it only for ranking. The gates and the visibility rule were set on the same 13 scenes that Figure 5 evaluates.

Cost. A fused Triton kernel casts one block of source pixels through all its candidate views in registers. On one RTX 5090, the fusion of all 13 scenes takes 3.9 s of GPU compute, and the frozen-pose BA, the uncertainty calibration and the fusion together take 82 s from the stored matches to the fused depth maps, including file reading.

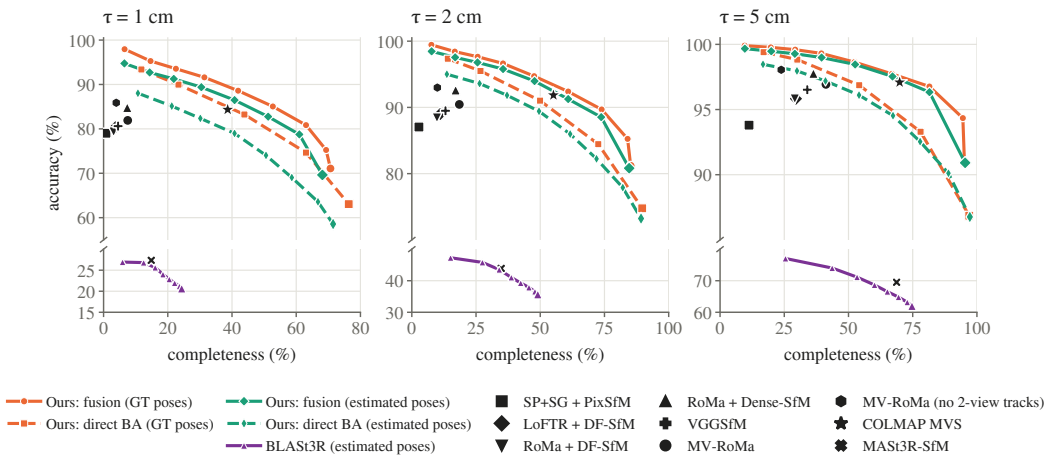


Figure 10: **Triangulation on ETH3D, with and without known poses.** Figure 5 with the curves from estimated poses and those of BLASt3R. Accuracy against completeness at 1, 2 and 5 cm (official ETH3D evaluator, mean over the 13 training scenes); each curve keeps the most certain fraction of its points, from 10% to all points. Orange curves and black marks use the ground-truth poses and intrinsics. Green curves start from the poses and focal length that DMap estimates (Section 4.1), and purple shows BLASt3R (Leroy et al., 2026) ranked by its confidence; both are mapped onto the scan by a similarity transform on the camera centres and one rigid ICP. The lower strip below the axis break holds BLASt3R and MASt3R-SfM (Duisterhof et al., 2025). With estimated poses, fusion is still more accurate and more complete than every published triangulation method at every threshold.

D Dense reconstruction without known poses

Protocol. DMap estimates the poses and one focal length shared by all images as in Section 4.1, solves one depth per refined pixel (832×1248 per view) with these poses frozen, and fuses the depths as in Section C. The calibration of Section B is fitted on the cross-view differences of this run, so no step before the evaluation reads the ground truth. BLASt3R (Leroy et al., 2026) runs its released offline setting for ETH3D (512 pixels, one shared camera). We score its depth maps (336×512) after its own outlier removal and rank its points by its confidence. Both clouds are mapped into the frame of the scan by the similarity transform that aligns the estimated camera centres with the ground-truth ones, followed by one rigid ICP onto the scan (10 cm correspondence distance, 2 cm subsampling). Accuracy and completeness come from the official evaluator. We also measure the distance from each point to the nearest scan point, capped at 1 m, over the points where the scanner has evidence, that is, the points the evaluator does not mark as unobserved. For fusion, this covers the points with finite uncertainty. The median and mean are taken per scene and averaged over the 13 scenes.

Results. Figure 10 adds the curves from estimated poses, and those of BLASt3R, to Figure 5. From estimated poses, fusion is more accurate and more complete than every published triangulation method at every threshold, although those use the ground-truth poses. Table 7 keeps all points. From estimated poses, fusion lies 0.42 cm from the scan at the median and 0.70 cm on average, against 0.36 and 0.64 cm from ground-truth poses and 2.99 and 6.55 cm for BLASt3R. Our measurement of BLASt3R agrees with its own evaluation, which compares to per-image ground-truth point maps instead of the scan and reports 2.6 cm median and 6.9 cm mean accuracy on the same 13 scenes. Figures 11 and 12 show three scenes, following the side views and horizontal slices of the BLASt3R appendix. In the slices, the walls of BLASt3R form bands several centimetres thick, while ours follow the scan.

E Component swap ablation

Table 8 crosses our matches and dense BA with BLASt3R's on all 25 ETH3D scenes, both BAs starting from the same VGGT-Ω cameras. Replacing either component improves over BLASt3R

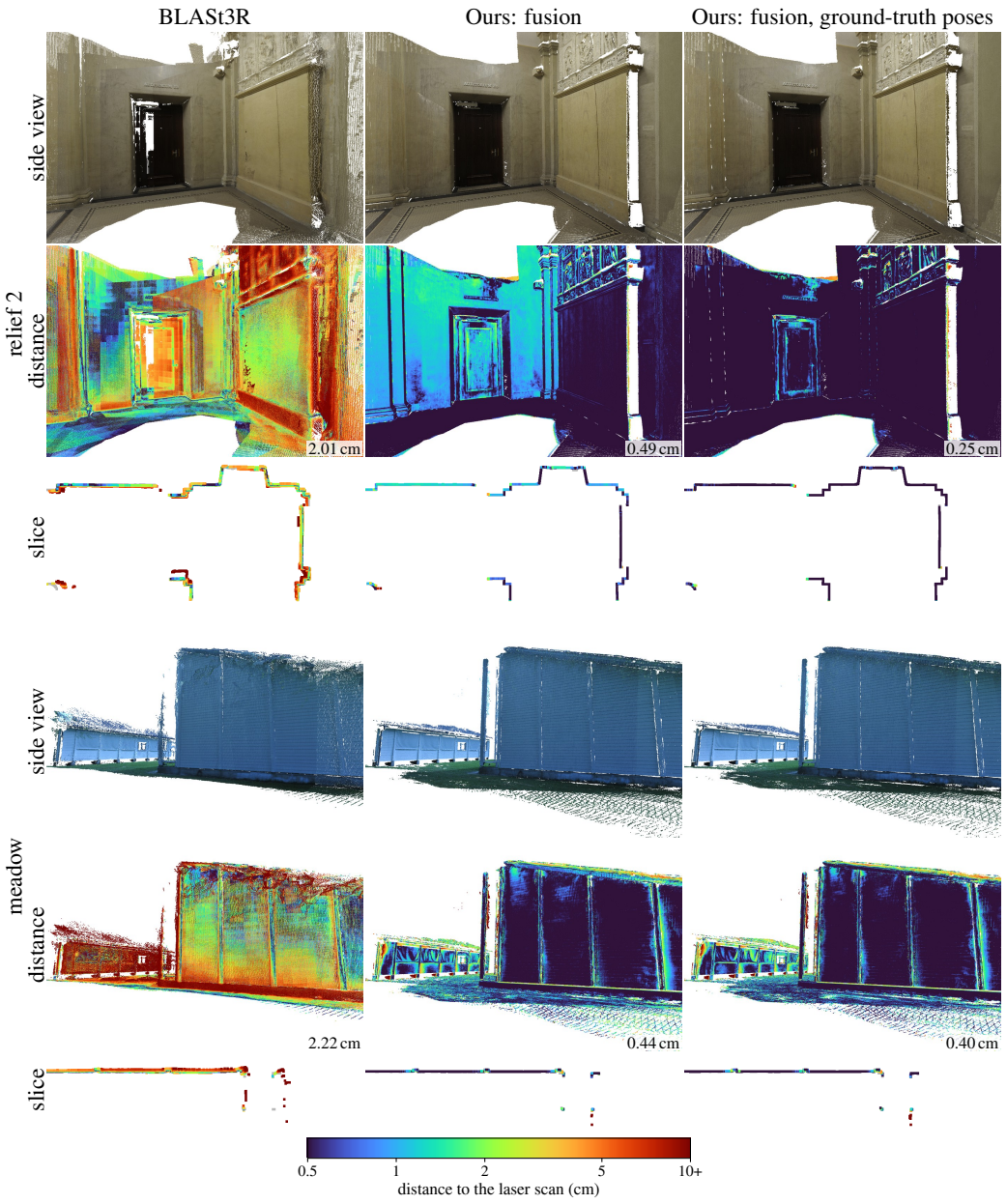


Figure 11: **Dense reconstruction without known poses: relief 2 and meadow.** Columns: BLASt3R and DMap with fusion, both from estimated poses and aligned by the camera centres and one ICP, and DMap with fusion from ground-truth poses. Per scene, the rows show a side view in colour, the same view coloured by the distance of each point to the scan with the median over the whole scene in the corner, and a zoomed horizontal slice seen from above (4 cm thick, 1 m above the ground; the scan in grey beneath). Both scenes are seen from behind one input camera. Only points where the scanner has evidence are drawn, so floaters in space the scanner saw as empty remain. From estimated poses, our fused points lie four to five times closer to the scan than those of BLASt3R at the median.

Table 7: **Dense reconstruction on ETH3D without known poses, all points kept.** Mean over the 13 training scenes. Accuracy and completeness from the official evaluator; distance from each point to the laser scan over the points where the scanner has evidence (for fusion, those with finite uncertainty). Estimated poses: similarity transform on the camera centres and one rigid ICP onto the scan. Ground-truth poses: the protocol of Figure 5. Bold: best with estimated poses. From estimated poses, fusion is seven times closer to the scan than BLASt3R at the median and within 0.1 cm of the ground-truth-pose result.

		Accuracy (%)			Completeness (%)			Distance (cm)	
Method	Poses	1 cm	2 cm	5 cm	1 cm	2 cm	5 cm	median	mean
BLASt3R	estimated	20.6	35.6	62.0	24.3	49.0	74.7	2.99	6.55
Ours: direct BA	estimated	58.6	73.2	86.8	**71.5**	**89.2**	**97.2**	0.51	1.32
Ours: fusion	estimated	**69.6**	**80.8**	**90.9**	68.2	84.6	95.4	**0.42**	**0.70**
Ours: direct BA	ground truth	63.1	74.8	86.8	76.3	89.7	97.0	0.39	1.22
Ours: fusion	ground truth	71.1	81.2	90.8	70.7	85.2	95.2	0.36	0.64

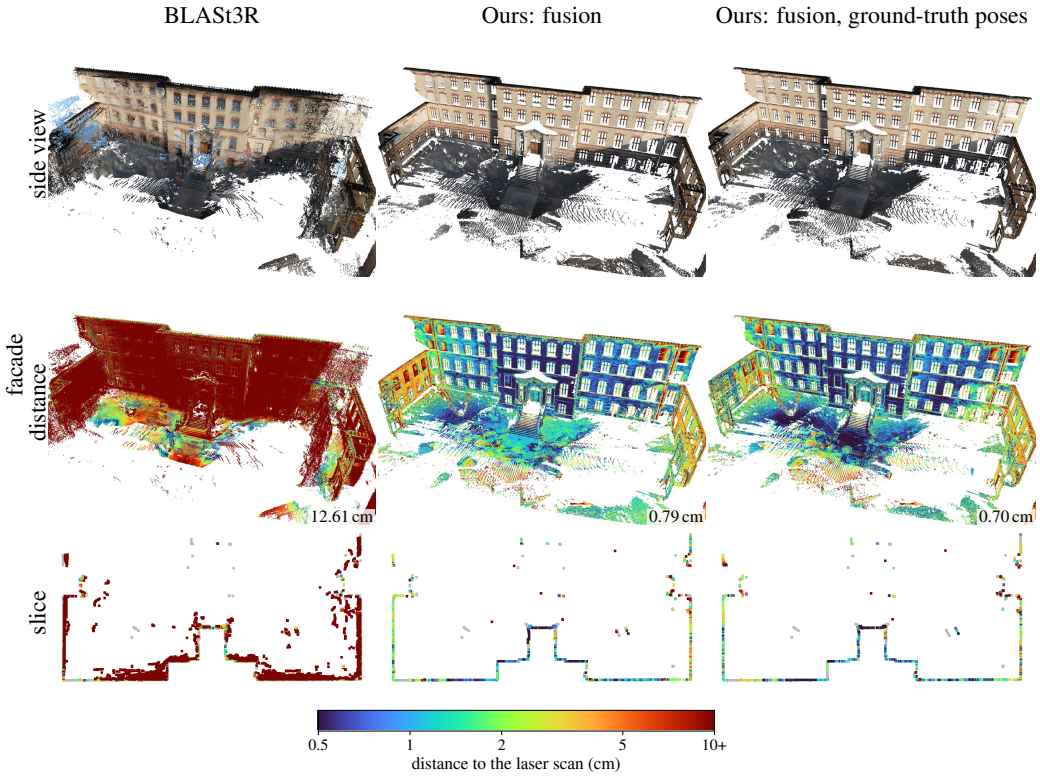


Figure 12: **Dense reconstruction without known poses: facade**, seen from above, drawn as in Figure 11. From estimated poses, our fused points lie sixteen times closer to the scan than those of BLASt3R at the median.

alone; both together are fastest and most accurate. Our dense BA does not rely on heuristic filtering of matches: the matching confidence and the robust loss suffice.

F Multiview context: examples and other methods

Example pairs. Figure 13 shows a pair that pairwise matching misses and one context image recovers. Figure 14 shows how the other methods match this pair and a second low-overlap pair; both pairs are hand-picked illustrations. No pairwise method recovers the first two pairs. With the same context image, BLASt3R also recovers them (98.8 and 98.6 PCK@3), but its matches are less

Table 8: **Component swap: matches × BA backend** on ETH3D (25 scenes). Both BAs start from the same VGGT-Ω cameras. AUC@1° and mean time per scene (s). Replacing either our matcher or our BA improves over BLASt3R alone; both together are the fastest and the most accurate.

	BLASt3R BA		Dense BA (ours)	
Matches	AUC@1°	Time	AUC@1°	Time
BLASt3R	55.31	24.1	74.89	18.7
Ours	**82.87**	**13.9**	**88.64**	**8.6**

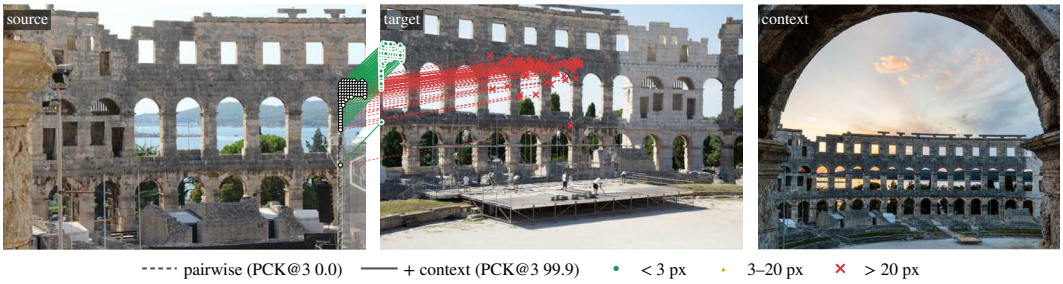


Figure 13: **One context image rescues a pair with low overlap and symmetry.** A MegaDepth-X pair with 1.7% overlap: matched alone, ours scores 0.0 PCK@3; with one bridging image that sees both, 99.9.

precise: PCK@1 is 36.5 and 60.3, against 95.9 and 85.3 for ours. MV-RoMa does not recover them. With all 24 views, the effect of context is not monotone for ours (54.3 and 0.4), while BLASt3R keeps 98.1 and 90.7. On the Câmara Municipal de Sintra pair of Figure 16, MV-RoMa succeeds pairwise (74.1 PCK@3) where ours fails (1.9); ours needs the context image (93.5).

All test pairs. Table 9 compares the methods on all 21,230 test pairs that every context size scored. Only ours improves consistently with context, from 77.9 to 80.3 with 24 views and from 37.3 to 43.3 on pairs with less than 5% overlap. BLASt3R stays between 71.8 and 72.5 PCK@3 for every context size, and MV-RoMa drops slightly, from 71.1 to 70.4. Figure 15 breaks this down by overlap, with the bins and pairs of Figure 6. With 24 views, ours gains 6.0 PCK@3 below 5% overlap and 0.8 above 50%; BLASt3R gains 1.9 and 1.8 in the two bins below 10% and changes by less than 0.6 elsewhere, and MV-RoMa loses in every bin, by 1.0 to 1.5 below 30%.

Representation. Figure 16 shows the representation that the loop transformer builds over its steps. We fit the first three principal components to the adapter stream after the last step, over all tokens of the 24 views, and project the stream of every step onto them, after centring each step on its own mean. This basis captures 2.7% of the variance at the input and 30–32% at the last step, so the early columns show how much of the final representation is already present. From step 2 on, periodic stripes appear and sharpen until step 5. They do not simply follow the image rows: on the Barringer Crater target they tilt with the crater floor.

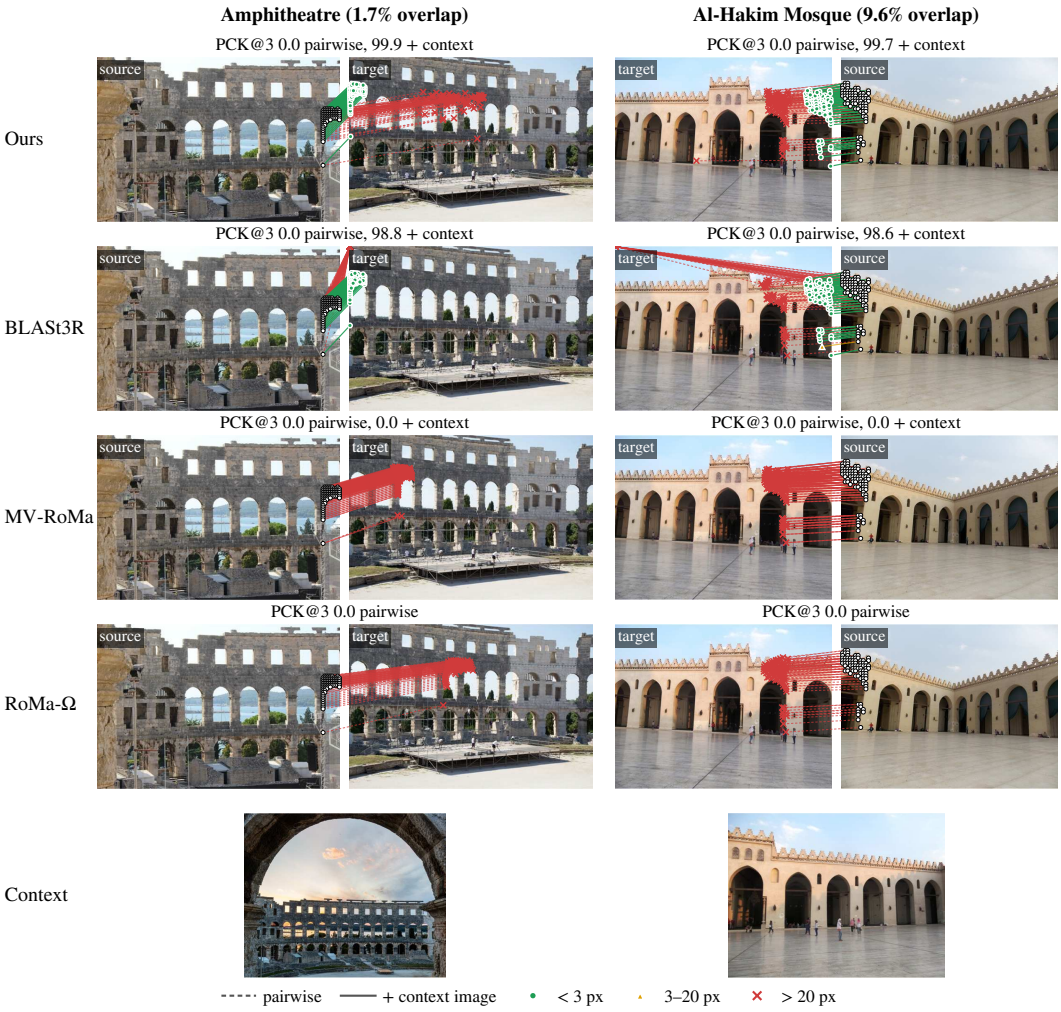
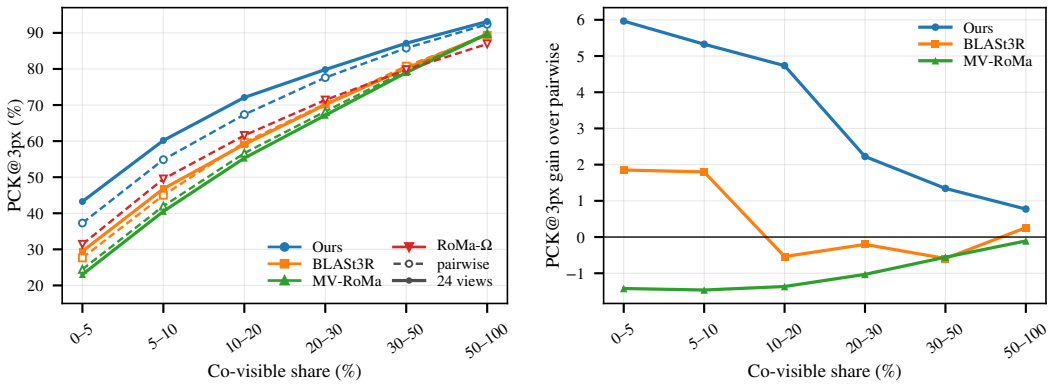


Figure 14: **Two low-overlap pairs, matched by every method.** Left: the pair of Figure 13; right: a source and a target image of the Al-Hakim Mosque (MegaDepth-X), drawn with the target on the left so that the lines stay between the images. Lines join a regular grid of source pixels to each method's final matches in the target, coloured by the error against the ground truth; only pixels with ground truth are drawn. Dashed: the pair alone; solid: with the context image (bottom row). Numbers give refined PCK@3; MV-RoMa runs at its high-resolution setting ($672 \rightarrow 1344$), and BLASt3R queries without a valid flow count as wrong. Only ours and BLASt3R recover the pairs, and only with the context image; ours places the matches more precisely (PCK@1 95.9 and 85.3 against 36.5 and 60.3).

Table 9: **Matching with multiview context on MegaDepth-X.** PCK in % at 1, 3 and 5 px (518^2), over the 21,230 ordered test pairs that every context size of every method scored (the pairs of Figure 6). Each method's final output: RoMa-Ω with its refiner, MV-RoMa at 560^2 without its high-resolution stage (which runs out of memory with 24 views), BLASt3R with its own flow (queries without a valid flow count as wrong), and ours with the RoMa-Ω refiner. Context: the pair alone, random groups of 4, 8 or 12 views that contain the pair, or all 24 views of the scene. The last column is PCK@3 on the pairs with less than 5% overlap.

Method	Context	PCK@1	PCK@3	PCK@5	PCK@3, <5%
RoMa-Ω	2 views	45.9	72.2	78.9	31.4
MV-RoMa	2 views	46.5	71.1	76.3	24.4
	4 views	46.2	70.4	75.4	22.9
	8 views	46.1	70.1	75.1	22.4
	12 views	46.2	70.2	75.3	22.7
	24 views	46.3	70.4	75.5	23.0
BLASt3R	2 views	45.8	72.3	77.2	27.6
	4 views	45.5	71.8	76.6	27.2
	8 views	45.5	71.8	76.4	26.6
	12 views	45.6	72.0	76.7	28.2
	24 views	45.8	72.5	77.2	29.5
DMap (ours)	2 views	51.0	77.9	84.1	37.3
	4 views	51.0	78.3	84.8	38.9
	8 views	51.4	79.3	85.9	40.8
	12 views	51.5	79.6	86.3	42.1
	24 views	51.8	80.3	87.1	43.3



(a) PCK@3px by overlap

(b) Gain of 24 views over pairwise

Figure 15: **Every method on MegaDepth-X by overlap.** Refined (final) PCK@3px at 518^2 on the 21,230 test pairs and overlap bins of Figure 6; settings as in Table 9. (a) Dashed, hollow markers: pairwise; solid, filled markers: all 24 views. (b) Difference of the per-bin means. Only ours turns context into a consistent gain, largest where the pair alone has the least overlap.

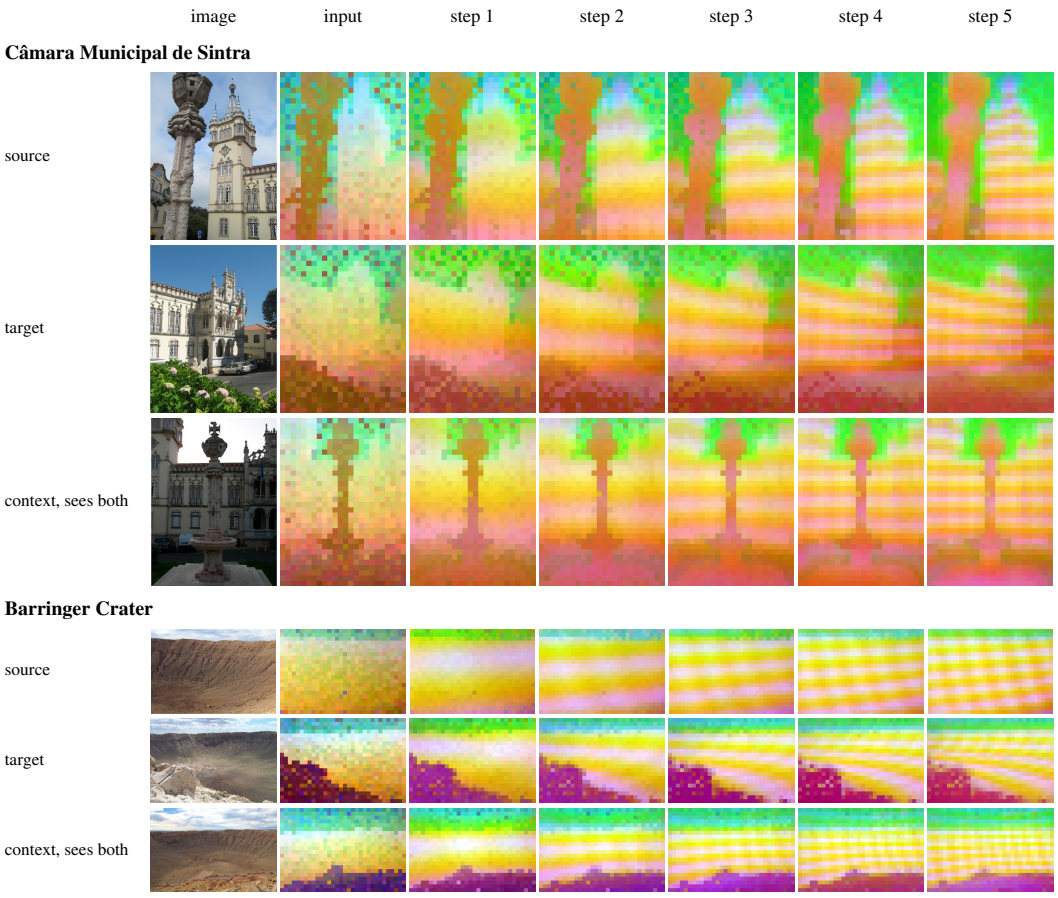


Figure 16: **The adapter stream across the loop steps.** Two MegaDepth-X scenes, forward pass over all 24 views: a source, a target and a context image that sees both. Each token is coloured by its projection on the first three principal components of the stream after the last step (all 24 views), with every step centred on its own mean and scaled to its own 1–99% range. The final representation builds up over the steps, and periodic stripes emerge from step 2 on.