INCREMENTAL RECONSTRUCTION OF LONG-RANGE PANORAMAS WITH GLOBAL CONSISTENCY

Anonymous authors
Paper under double-blind review



Figure 1: **Globally consistent reconstruction from a long panoramic sequence.** Our method reconstructs long-range panoramic sequence along a 1.2 km route by coupling sparse mapping with feed-forward geometry.

ABSTRACT

Long-sequence panoramic reconstruction requires detailed scene geometry and globally consistent camera poses across distant revisits. Panoramic SLAM and incremental SfM rely on feature correspondences, but repeated, similar structures and small partial overlap make reliable long-range connections difficult to establish, leaving accumulated drift unresolved. Feed-forward models offer detailed local geometry, but their high computational costs limit inference window size, necessitating additional constraints for global consistency. This motivates combining continuous feature-based tracking and sparse mapping with rapid, high-quality local reconstruction from feed-forward models to achieve accurate localization and globally consistent dense mapping over long sequences. We therefore propose a panoramic reconstruction framework that couples an ORB-based sparse panoramic mapping frontend with a fine-tuned feed-forward model. Our key insight is that predicted local geometry enables 3D verification of feature matches across distant revisits, yielding long-range camera constraints even between regions. We calibrate predicted depths with sparse landmarks and use verified connections to optimize the global pose graph, updating dense geometry with the corrected camera poses. We also introduce PanoRecon, a seven-scene benchmark with repetitive architecture, large and small loops, and reference camera poses and LiDAR geometry for evaluating global consistency. On the long-range panoramic reconstruction benchmarks YUTO MMS and our newly introduced PanoRecon, our method achieves the lowest mean absolute trajectory error and high F-1 Score among the evaluated methods.

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.

1 INTRODUCTION

Panoramic reconstruction recovers camera trajectories and dense geometry from equirectangular projection (ERP) images covering the full surroundings. Along extended routes through buildings and streets, reconstruction requires detailed surfaces and consistent placement of observations, particularly across distant revisits.

Feature-based panoramic SLAM and incremental structure-from-motion (SfM) register cameras and maintain sparse landmarks through image correspondences. SLAM systems use loop closure to correct accumulated drift, with acceptance thresholds controlling which revisits enter global optimization (Surmann et al., 2022; Campos et al., 2021). Relaxed thresholds admit false associations, whereas strict thresholds reject valid loops with limited matching support. Repeated facade textures and symmetric layouts make distinct places visually similar, while distant revisits often share only partial overlap. False loops distort the map, and missed loops leave drift uncorrected, producing misaligned structural layers. Reliable global correction of panoramic reconstruction therefore requires stronger evidence for distinguishing true revisits from repeated scene patterns.

Feed-forward reconstruction rapidly estimates camera poses and detailed local geometry from panoramic views, as demonstrated by PanoVGGT (Guo et al., 2026b) and panoramic adaptations of Pi3 (Wang et al., 2026). Memory and computational costs limit the number of jointly processed views, while separate inference windows still require global constraints to remain consistent. *This motivates combining continuous feature-based tracking and mapping with rapid local feed-forward reconstruction for accurate localization and globally consistent reconstruction over long sequences.*

Based on the above mentioned discussion, we therefore introduce an incremental panoramic reconstruction framework that couples ORB-based sparse mapping with a fine-tuned feed-forward panoramic model (Fig. 1). Our key insight is that **the predicted local geometry can enable 3D verification of feature matches across distant revisits**. We first calibrate predicted geometry with sparse landmarks and verify a common relative transform across views to establish long-range camera constraints. Accepted connections further drive pose-graph correction, which establishes shared observations for panoramic bundle adjustment with a sparse differentiation GPU backend. After refinement, we recalibrate cached depths and update each dense point using its source camera's corrected pose, without repeating inference.

To evaluate global consistency under these conditions, we further introduce the PanoRecon benchmark, comprising seven real-world scenes with repeated architectural textures and geometric structures. Every scene contains multiple revisits separated by long travel distances. We also collect LiDAR data, providing reference camera poses and dense geometry for evaluating trajectory accuracy and reconstruction consistency across distant revisits.

Our contributions are as follows:

- We propose **a long-range panoramic reconstruction framework** that couples feature-based sparse mapping with feed-forward geometry to recover globally consistent camera trajectories and detailed dense maps.
- We introduce **a long-range global connection mechanism** that integrates feed-forward panoramic geometry with feature matching to enable 3D verification and establish camera constraints without requiring shared sparse landmarks between revisited regions.
- We introduce the **PanoRecon benchmark** with repetitive architecture and long-range revisits to evaluate trajectory accuracy and dense reconstruction consistency.

2 RELATED WORK

2.1 STRUCTURE FROM MOTION

SfM recovers cameras and structure through incremental registration (Schonberger & Frahm, 2016; Heinly et al., 2015) or global motion estimation (Moulon et al., 2013; Wilson & Snavely, 2014; Pan et al., 2025). Resent study extends these operations to spherical imagery (da Silveira et al., 2022), including spherical pose estimation and bundle adjustment (Jiang et al., 2024), dense matching (Jung et al., 2025), and structural constraints from lines and architectural layouts (Ikehata et al., 2016; Huang et al., 2026). These advances improve registration and geometry estimation. Distant regions

with partial overlap still require reliable associations for global refinement, motivating our use of predicted depths to support feature correspondences across views.

2.2 Visual Odometry and Panoramic SLAM

Visual odometry estimates sequential motion, with direct methods optimizing photometric error (Engel et al., 2018), while feature-based, direct, and learned SLAM maintain maps with multi-view constraints (Mur-Artal & Tardos, 2017; Gao et al., 2018; Teed & Deng, 2021). OpenVSLAM (Sumikura et al., 2019) supports purely visual equirectangular mapping. Visual–inertial estimation (Qin et al., 2018) extends to panoramic and spherical multi-camera observations (Wang et al., 2022; 2024b; Wu et al., 2026; Yang et al., 2026), while HDPV-SLAM (Ahmadi et al., 2023) incorporates LiDAR depth. Their global accuracy depends on distinguishing true revisits from repeated appearance.

2.3 Feed-Forward Reconstruction

Feed-forward geometry and matching models (Wang et al., 2024a; Leroy et al., 2025; Wang et al., 2025a; 2026) supply dense reconstruction priors. CUT3R maintains persistent state (Wang et al., 2025b), SLAM3R learns local-to-world registration (Liu et al., 2025), and StreamVGGT and InfiniteVGGT use causal memory (Zhuo et al., 2026; Yuan et al., 2026). MASt3R-SLAM, VGGT-SLAM, VGGT-Long, and VGGT-SLAM 2.0 couple predictions with graph optimization and loop closure (Murai et al., 2025; Maggio et al., 2025; Deng et al., 2026; Maggio & Carlone, 2026). VGGT-Long processes overlapping chunks, while VGGT-SLAM 2.0 uses attention to verify revisits. For panoramas, PanoVGGT (Guo et al., 2026b) predicts poses and geometry, while HALO-SLAM (Xiong et al., 2026) exploits attention, and dense geometry for tracking and loop verification.

3 Method

Given an ERP image sequence, we reconstruct camera poses and dense geometry by combining a ORB-based sparse map with local feed-forward predictions (Fig. 2). We first calibrates and places the dense predictions in a common coordinate (Sec. 3.1). We then use these geometries to verify feature matches across views and construct long-range camera constraints (Sec. 3.2). Optimizing the sparse map with these constraints updates the cameras and landmarks, which we use to recalibrate and reposition the cached dense points (Sec. 3.3).

3.1 Anchoring Feed-Forward Geometry to the Sparse Map

Local predictions and the sparse map. An ORB-based ERP frontend Surmann et al. (2022) incrementally estimates camera-to-world poses $T_i = (R_i, \mathbf{c}_i)$ and sparse landmarks $\mathbf{L}_j$, where i indexes keyframes, j indexes landmarks, and $T_i\mathbf{x} = R_i\mathbf{x} + \mathbf{c}_i$. We group successive keyframes into overlapping windows and render their panoramas into calibrated perspective views. Pi3 (Wang et al., 2026), fine-tuned on Holo360D (Ou et al., 2026), jointly processes these views to predict local geometry. Each window defines a submap, whose predicted depths we anchor to the frontend's cameras and landmarks.

We additionally replace the per-view sky segmentation network with a lightweight sky head supervised by Holo360D geometry ground truth. It predicts a sky mask directly on each ERP image and projects to the perspective views, reducing inference time per submap from 40 s to 13 s.

Camera-anchored dense representation. Global optimization of ORB frontend can update the cameras within a submap differently. We therefore retain each prediction in its source camera coordinates and determine its world position from the current sparse-map pose. For pixel $\mathbf{p}$ in virtual view v of keyframe i and window w, let $\hat{z}_{wiv}(\mathbf{p})$ be the optical-axis depth of the Pi3 local point. Its corresponding world point is

$$\mathbf{X}_{wiv}(\mathbf{p}) = T_i\big(s_{wi}Q_v\hat{z}_{wiv}(\mathbf{p})K_v^{-1}\bar{\mathbf{p}}\big)\,, \qquad s_{wi} = \exp(a_w + b_{wi}), \tag{1}$$

where $\bar{\mathbf{p}}$ is the homogeneous pixel, K_v is the rendering intrinsic matrix, and Q_v rotates the virtual view into the panoramic camera. The virtual views share the panorama's optical center and fixed relative rotations. The shared window log scale a_w aligns predicted depths to the sparse map, while

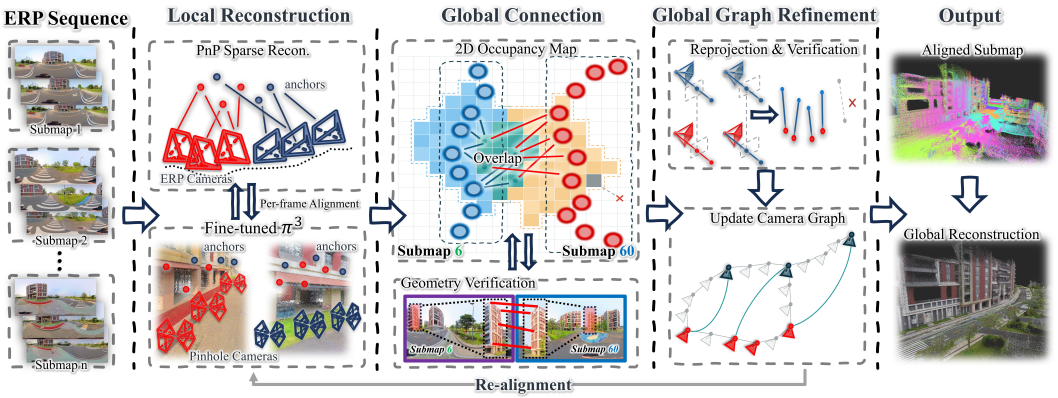

Figure 2: **Overview of the proposed method.** Sparse landmarks calibrate Pi3 depths, and registered panoramic cameras place them in the map. These predictions support cross-view ORB correspondences for verifying long-range camera constraints. Global correction and observation fusion refine the sparse map, which recalibrates and repositions the cached dense geometry.

the frame log-scale correction b_{wi} captures variation within the window. The calibrated rays and sparse-map poses determine where points are placed, while depth calibration estimates their scales.

Submap alignment from landmarks. We estimate these scales from reliable landmark observations pooled within each window. For landmark j, its observation maps to pixel $\mathbf{p}_{ijv}$ in virtual view v, and the current pose gives its optical-axis depth z_{ijv}^{geom}. Comparing this depth with the prediction gives the log ratio $y_{wijv} = \log[z_{ijv}^{\text{geom}} / \hat{z}_{wiv}(\mathbf{p}_{ijv})]$. We fit the shared scale and the vector of frame scale corrections $\mathbf{b}_w = (b_{wi})_i$ through

$$\min_{a_w, \mathbf{b}_w} \sum_{(i,j,v)\in\mathcal{A}_w} \rho(y_{wijv} - a_w - b_{wi}) + \frac{\lambda_p}{2} \sum_i b_{wi}^2 + \frac{\lambda_s}{2} \sum_{(i,k)\in\mathcal{N}_w} (b_{wi} - b_{wk})^2. \tag{2}$$

Here $\mathcal{A}_w$ contains reliable observations with positive geometric and predicted depths, ρ is the Huber penalty, and $\mathcal{N}_w$ links adjacent frames with enough valid landmark observations. The weights λ_p and λ_s respectively constrain frame scale corrections toward the shared scale and encourage temporal smoothness. Log ratios measure proportional depth errors across scene distances. All frames share the window scale, with frame-specific corrections estimated from sufficient landmark support. After robust fitting, we align depths at common source pixels in overlapping windows, as detailed in Appendix B.1. Substituting the calibrated scales into Eq. (1) expresses each submap in the ORB landmarks' coordinates for the connection estimation below.

3.2 Verifying Long-Range Connections with Predicted Geometry

We use the aligned submaps to identify and verify overlap between distant observations. For a new submap B, we propose historical submaps A using voxel overlap and line-of-sight visibility at their current map positions (Figure 3). Expanded spatial bounds and neighboring historical windows broaden this candidate set. Within each candidate pair, we compute mutual ORB matches across all pairs of keyframes with predicted depths. Combining matches across image pairs lets us use overlap distributed across views.

Depth-supported similarity estimation. To check their geometric consistency, we back-project matched keypoints using the calibrated depths from Sec. 3.1 (Fig. 3). A sample's distance from its source camera is $s_{wi}\hat{z}_{wiv}(\mathbf{p})\|K_v^{-1}\bar{\mathbf{p}}\|_2$. Within each submap, we project samples into their source camera's ERP image and take the median distance near each feature, denoted $d_i(\mathbf{u})$ for ERP pixel $\mathbf{u}$. With unit bearing $\mathbf{b}(\mathbf{u})$, the matched feature becomes $\mathbf{P}_i(\mathbf{u}) = T_i[d_i(\mathbf{u})\mathbf{b}(\mathbf{u})]$ in world coordinates. The resulting point correspondences from keyframes $a \in A$ and $b \in B$ are combined to estimate $S_{A\leftarrow B} \in \mathrm{Sim}(3)$ by RANSAC and least-squares refitting (Umeyama, 1991), and define submaps' rotation, translation, and positive scale. It aligns corresponding points through $\mathbf{P}_a \approx S_{A\leftarrow B}\mathbf{P}_b$, correcting the displacement caused by drift despite their shared map coordinates.

Cross-view geometric verification. Repeated structures can produce plausible local matches, so we test whether a common similarity explains correspondences across multiple views. We check bidirectional distance and bearing agreement, nondegenerate geometry, and distinct feature support from multiple keyframes on both sides. For large corrections, we divide the current submap's keyframes into groups with balanced numbers of matches before fitting. We estimate a transform from a group's matches and verify it using matches in the other, then repeat with the groups exchanged. Both transforms must agree with the estimate from all matches. Verified transforms from neighboring historical submaps must also agree before acceptance. Appendix B.2 specifies these checks and their thresholds.

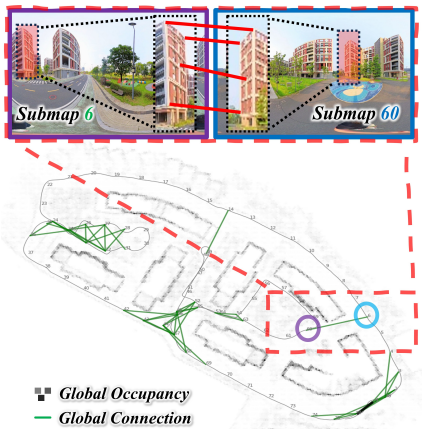

Figure 3: Cross-view evidence for a global connection.

Camera constraint construction. We convert each verified submap transform into relative camera constraints for the sparse map. We select image pairs with enough inliers and distinct camera endpoints, then form the relative constraint mapping camera a to camera b,

$$\widehat{G}_{b\leftarrow a} = T_b^{-1} S_{A\leftarrow B}^{-1} T_a, \tag{3}$$

using the camera poses at the time the constraint is formed. The measurement remains fixed during subsequent optimization. The inverse similarity maps points reconstructed from submap A onto those from submap B before conversion into camera b. We cap the total weight for each verified submap transform based on its inlier count and divide this weight among the selected camera constraints. These weighted constraints enter the pose graph below.

3.3 Propagating Global Corrections to Sparse and Dense Geometry

Graph correction and observation fusion. We insert the relative camera constraints from Eq. (3) into the similarity pose graph alongside existing sparse-map constraints. We optimize the graph, remove incompatible new edges, and re-solve to update camera poses and landmark positions. In the corrected geometry, standard projection-based landmark fusion (Mur-Artal & Tardos, 2017) runs bidirectionally between the cameras connected by accepted constraints and a limited number of their covisible neighbors. Merging duplicate landmarks establishes observations shared across revisited regions. We denote the updated camera–landmark observation set by $\mathcal{O}$, which supplies cross-region reprojection factors for global BA.

GPU-accelerated panoramic bundle adjustment. The shared observations allow BA to jointly refine the graph-corrected poses T_i and landmarks $\mathbf{L}_j$. We use the original ERP feature pixels (u_{ij}, v_{ij}), retaining one pose per panorama so that BA directly refines the sparse map used to align dense predictions. As long-range connections enlarge this observation graph, repeated global refinement becomes costly, motivating GPU acceleration. With predicted pixels $(\hat{u}_{ij}, \hat{v}_{ij}) = \pi_{\mathrm{ERP}}(T_i^{-1}\mathbf{L}_j)$, where π_{ERP} is the panoramic projection, we minimize

$$\min_{\{T_i, \mathbf{L}_j\}} E_{\mathrm{BA}} = \sum_{(i,j)\in\mathcal{O}} \psi_{\delta_b}\left(\sigma_{ij}^{-2} \|\mathbf{r}_{ij}\|_2^2\right), \qquad \mathbf{r}_{ij} = \left[\mathrm{wrap}_W(u_{ij} - \hat{u}_{ij}),\ v_{ij} - \hat{v}_{ij}\right]^\top. \tag{4}$$

Here ψ_{δ_b} is the Huber loss on the squared weighted residual norm, with transition at δ_b^2, and σ_{ij}^{-2} is the feature-pyramid inverse variance. For ERP width W, $\mathrm{wrap}_W(q) = q - W\lfloor q/W + 1/2 \rfloor$ measures horizontal error across the image seam. Starting from the corrected geometry with the root pose fixed, sparse automatic differentiation (Zhan et al., 2026) linearizes the residuals. After landmark elimination, GPU preconditioned conjugate gradients solve the reduced camera system without storing the explicit Schur complement. Appendix B.3.2 details the solver, and Table 3 reports its efficiency.

Updating cached dense geometry. The refined cameras and landmarks also change the depth references used in Sec. 3.1. We recompute the landmark depths, refit the scales in Eq. (2), and

Dataset property	Value
Number of scenes	7
Mean video duration	11.49 min
Frame sampling interval	0.1 s
Mean trajectory length (km)	0.87 (0.26–1.69)
Reprojection RMSE (pix)	0.1917
Loop structure	Large and small loops

(a) Dataset statistics



(b) Dataset example

Figure 4: Dataset statistics and GT examples of the PanoRecon benchmark.

place the cached predictions using the updated T_i and s_{wi} in Eq. (1). Each dense point is updated using its source camera's optimized pose and recalibrated depth, without repeating Pi3 inference. At each submap boundary, any triggered global refinement and dense updates finish before frame input resumes, so subsequent tracking and connection estimation use the corrected map.

4 The PanoRecon Benchmark

We independently capture panoramic videos and RTK-assisted LiDAR data along the same planned routes in seven scenes with low-texture surfaces and repeated architecture. Each route includes a large loop and distant revisits with limited overlap, testing reconstruction methods' ability to correct accumulated drift.

Reference geometry and camera poses. We optimize the LiDAR trajectory with RTK corrections and export the point cloud and calibrated poses of its three onboard cameras. In *RealityScan*, these define LiDAR reference images with fixed poses. ERP frames sampled every 0.1 s are split into calibrated perspective rigs and globally matched to these images. We optimize only the ERP rig poses, then recover one pose per panorama using the rig extrinsics. The reference poses share the LiDAR's metric coordinate system. The mean reprojection RMSE is 0.1917 pixels (Fig. 4). Appendix A.1 details acquisition and processing.

Reconstruction evaluation. For a reconstruction in frame E, trajectory alignment to the reference poses gives the similarity $S_{L\leftarrow E}$ into the LiDAR frame L. We apply it to the reconstructed geometry and refine registration with ICP. Following Tanks and Temples (Knapitsch et al., 2017), bidirectional nearest-neighbor distances yield precision, recall, and F-score. The complete transformation is

$$\mathbf{x}_L^{\text{eval}} = H_{\text{ICP}}\, S_{L\leftarrow E}\, \mathbf{x}_E, \tag{5}$$

where H_{ICP} refines cloud alignment for geometric evaluation.

5 Experiments

Implementation. We process submaps of 16 panoramic keyframes with four shared keyframes between consecutive submaps. Depth scale calibration uses a log-ratio Huber threshold of 0.05. Connection verification uses bidirectional distance and bearing thresholds of 0.3 map units and 1°, with support from at least three keyframes on each side. GPU bundle adjustment uses double precision and at most 250 PCG iterations with a tolerance of 10^{-5}. Further settings are provided in Appendix B. All experiments are conducted on a single NVIDIA RTX PRO 6000 Blackwell Workstation Edition GPU.

Datasets. We use the mapping sequences from all four scenes of 360LOC (Huang et al., 2024): Atrium, Concourse, Hall, and Piatrium. We also evaluate all four sequences (A–D) of YUTO MMS (Zhang et al., 2024), a panoramic dataset covering 20.1 km and the Sequence C alone spans 9.137 km with more than 9,000 ERP images. Finally, we evaluate the seven PanoRecon scenes introduced in Sec. 4. Both YUTO MMS and the PanoRecon benchmark contain long trajectories and complex loop structures, which challenge sustained tracking and global drift correction.

Table 1: **Quantitative comparison of localization.** Mean scene-wise RMSE on 360LOC (4 scenes), YUTO MMS (4 sequences), and PanoRecon (7 scenes). OOM denotes GPU out-of-memory.

Method	360LOC			YUTO MMS			PanoRecon		
	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓
Feed-Forward									
PanoVGGT (ERP)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
Scal3R (Pinhole)	5.366	0.124	1.354	187.407	2.911	1.025	9.794	0.282	0.340
SLAM									
360DVO (ERP)	0.954	0.026	0.039	68.313	0.183	**0.011**	2.799	0.104	0.039
Open-vSLAM (ERP)	0.441	0.711	22.184	281.793	41.690	37.574	7.399	6.580	25.242
Stella-vSLAM (ERP)	0.706	0.047	0.147	82.173	0.656	0.131	7.424	1.329	0.470
Structure-from-Motion									
COLMAP (ERP)	0.355	**0.020**	0.037	124.490	0.814	0.066	3.510 (6/7)	2.575 (6/7)	1.452 (6/7)
COLMAP (Pinhole)	0.399	0.021	**0.037**	113.317	25.949	1.046	1.736	**0.099**	**0.036**
Spherical-SfM (ERP)	0.491	0.024	0.038	21.186	3.312	3.989	11.779	6.915	1.418
Ours (ERP)	**0.270**	0.030	0.044	**3.527**	**0.052**	0.022	**1.697**	0.111	0.086

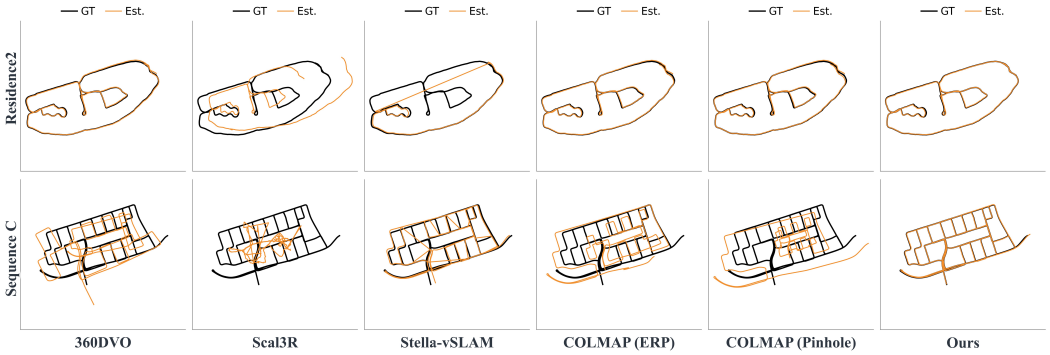

Figure 5: Qualitative comparison of localization.

Baselines. We compare against PanoVGGT (Guo et al., 2026b) and Scal3R (Xie et al., 2026) for feed-forward reconstruction, 360DVO (Guo et al., 2026a), Open-vSLAM (Sumikura et al., 2019), and Stella-vSLAM-Dense (Surmann et al., 2022) for panoramic visual odometry and SLAM, and COLMAP (Schonberger & Frahm, 2016) and Spherical-SfM for SfM (Table 1). Scal3R receives distortion-free perspective views rendered from ERP images using COLMAP's official panorama renderer, with a 90° field of view (front view). We evaluate COLMAP in two configurations: directly on ERP images with an equirectangular camera model, and on 12 such perspective overlapping views per panorama with known intrinsics and fixed per-ERP rig extrinsics.

Metrics. We evaluate camera poses after one global Umeyama $\mathrm{Sim}(3)$ alignment to the reference. ATE, translational RPE (RPE-t), and rotational RPE (RPE-r) are reported as RMSE. For dense reconstruction, we apply the trajectory similarity to the point cloud. We report mean scene-wise precision, recall, and F-score at 20 cm on 360LOC, YUTO MMS, and PanoRecon (Table 2). Appendices D and E provide per-scene results and evaluation details.

Table 2: **Quantitative comparison of dense reconstruction.** Mean scene-wise precision (P), recall (R), and F-score (F) in percent, at 20 cm on 360LOC (4 scenes), YUTO MMS (4 sequences), and PanoRecon (7 scenes). Higher is better.

Method	360LOC			YUTO MMS			PanoRecon		
	P↑	R↑	F↑	P↑	R↑	F↑	P↑	R↑	F↑
Stella-vSLAM-Dense (ERP)	62.42	**58.23**	58.68	23.78	14.52	17.24	36.61	31.10	31.97
COLMAP (Pinhole)	**81.27**	44.79	57.25	29.15	13.23	17.28	**71.32**	51.83	59.75
Ours (ERP)	79.22	52.87	**62.04**	**60.56**	**40.77**	**48.71**	64.22	**62.36**	**63.16**

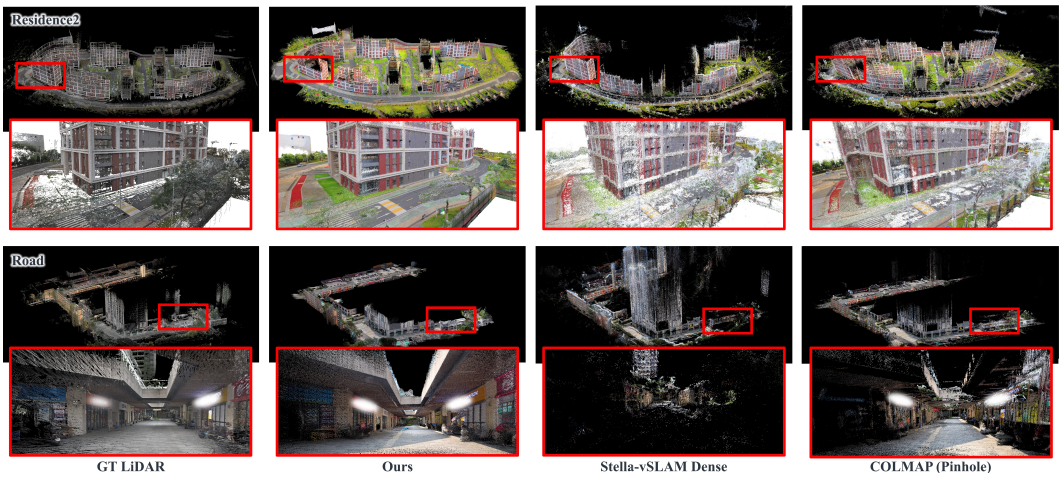

Figure 6: Point-cloud comparison on the Residence2 and Road scenes of the PanoRecon benchmark.

5.1 Trajectory Comparison

Our method achieves the lowest mean ATE on all three datasets, with the largest advantage on YUTO MMS (Table 1). False loop closures distort ORB-based Stella-vSLAM trajectories on Residence2 and Sequence C, where buildings share similar appearance and geometry (Fig. 5). Scal3R exhibits scale drift on large scenes with one perspective view per panorama, as does COLMAP during sequential reconstruction in both ERP and pinhole configurations. Our trajectories follow the reference street layouts and loop structures. On our single GPU, PanoVGGT (Guo et al., 2026b) ran out of memory (OOM) on every evaluated scene under the input settings in Appendix A.2. HALO-SLAM (Xiong et al., 2026) had no public implementation at evaluation time.

5.2 Dense Reconstruction Comparison

Accurate camera poses keep local predictions aligned across distant revisits, reducing drift-induced misalignment in the global point cloud. Our largest F-score gains occur on the long-range YUTO MMS sequences (Table 2), consistent with the trajectory results. Feed-forward geometry also improves local completeness on weakly textured walls and reflective surfaces, where photometric matching is unreliable. In these regions, the PatchMatch-based COLMAP and Stella-vSLAM-Dense baselines exhibit holes and noisy geometry, while our method recovers more continuous surfaces (Fig. 6 and supplementary Figs. 10–12). Appendix E reports per-scene results.

5.3 Runtime.

Appendix C.1 provides the runtime analysis and stage-wise comparison on YUTO MMS Sequence C. The GPU backend reduces the cost of global refinement while preserving comparable final reprojection error (Table 3). Matrix-free Schur optimization makes the growing observation graph more efficient to solve, supporting repeated correction as long-range connections accumulate.

Table 3: **CPU and GPU bundle adjustment.**

Backend	Solve (s)↓	Total BA (s)↓	Speedup↑	Reproj. RMSE↓
CPU (OpenMP, 32 threads)	19.430	32.649	1.00×	0.8466
GPU (CUDA Schur)	**2.244**	**7.581**	**4.31×**	**0.8420**

5.4 Ablation Studies

Effect of global constraints and bundle adjustment. In the Residence2 submap experiment, BA alone changes little, while a single camera constraint with BA provides most of the ATE reduction (Table 5). Additional constraints improve ATE further. Graph correction improves global alignment, and BA restores local motion consistency, showing how a few constraints can benefit reconstruction through joint refinement.

Table 4: **Effect of global constraints.** (Residence2 full trajectory)

Configuration	ATE (m)↓	RPE-t (m)↓	RPE-r (°)↓	Reproj. (px)↓	Long-range observations
No connection	1.358	0.061	**0.045**	1.623	160,186
PG	0.977	0.060	0.046	1.659	160,186
PG + BA	0.922	0.057	**0.045**	**1.620**	160,186
PG + fusion	0.877	0.060	0.046	1.760	**293,827**
PG + fusion + BA	**0.708**	**0.048**	**0.045**	1.662	**293,827**

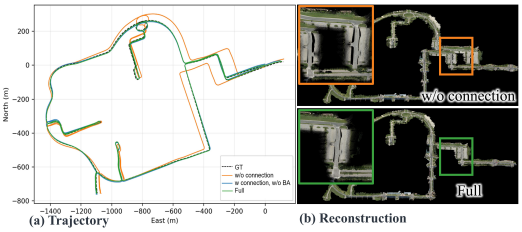

Figure 7: Effect of global constraints.

Table 5: **Global connections and joint refinement.**(Residence2 Submap $6 \rightarrow 60$; $16 \rightarrow 49$)

Edges	Weights	BA	ATE (m)↓	RPE-t (m)↓	RPE-r (°)↓	F-1 (%)↑
0	–	Off	0.370	0.0128	0.0177	68.36
0	–	On	0.371	0.0129	**0.0175**	68.36
5	Inlier	Off	0.296	0.0633	0.0939	71.76
1	Inlier	On	0.185	0.0095	0.0176	71.49
2	Inlier	On	0.180	**0.0093**	0.0176	71.59
5	Uniform	On	0.170	0.0094	0.0176	71.81
5	Inlier	On	**0.165**	0.0094	0.0176	**71.97**

Table 6: **Submap alignment.** (Residence2 Submap $6 \rightarrow 60$; $16 \rightarrow 49$)

Alignment	P↑	R↑	$F-1$ ↑
Pose-derived scale	85.14	61.22	71.23
Whole-submap Sim(3)	85.08	**61.93**	71.68
Sparse window scale	86.16	60.72	71.24
Sparse per-frame scale (ours)	**86.48**	61.62	**71.97**

Table 7: **Cross-view support and verification.** (Residence2 Submap $6 \rightarrow 60$; $16 \rightarrow 49$)

Submap pair	Matching / validation	Image pairs	Support Inliers	Support A/B	Support Test
$6 \rightarrow 60$	Sampled	16	58	1/1	Fail
$6 \rightarrow 60$	All frames	256	1,843	10/14	Pass
$16 \rightarrow 49$	Sampled	32	259	3/4	Fail
$16 \rightarrow 49$	All frames / alternating groups	256	3,990	6/14	Fail
$16 \rightarrow 49$	All frames / balanced groups	256	3,990	6/14	Pass

Observation fusion and global refinement. Low reprojection error after local BA can coexist with global drift. PG reduces ATE while slightly increasing reprojection error, consistent with optimizing camera constraints before pixel-level refinement (Table 4, Fig. 7). Fusion merges landmark tracks across revisits, increasing long-range support and changing the reprojection constraints. Subsequent BA reduces residuals and further improves trajectory accuracy.

Effect of depth scale calibration. Frame-specific calibration improves the precision–recall balance after registration, showing the value of accommodating scale variation within a prediction window (Table 6). Whole-submap alignment retains higher recall. The advantage of per-frame calibration lies in the fidelity of the registered geometry and F-score.

Multi-view support and geometric verification. On the selected Residence2 pairs (small overlap long-range connections), all-frame matching increases support, while balanced groups allow verification to pass at unchanged thresholds (Table 7). Predicted depth expands 3D support, and pooling views improves correct recall (Table 10). A retrieval study on six long-range revisits also shows that expanded bounds improve candidate recall under added drift (Appendix C.2.1).

6 Conclusion

In this paper, we study how local feed-forward geometry can support global consistency in long-sequence panoramic reconstruction. The local geometry of feed-forward model can provide 3D evidence for verifying distant feature matches and establishing camera constraints. The key insight is this mutual dependence: sparse mapping anchors local geometry, and local geometry strengthens the constraints used to correct the sparse map. Our seven-scene PanoRecon benchmark evaluates trajectory accuracy and reconstruction consistency under repetitive architecture and long-range revisits using reference poses and LiDAR geometry. Experiments across 360LOC, YUTO MMS, and PanoRecon show the improved mean ATE and high fidelity dense reconstruction results.

Limitations and Future Work. Severe motion blur or prolonged low-texture segments can reduce the reliable feature observations needed for tracking and depth calibration. Our system also does not explicitly handle dynamic objects, such as moving vehicles and pedestrians, and does not yet achieve real-time reconstruction. Future work will focus on more robust correspondences and efficient feed-forward models for streaming reconstruction from 360° video in dynamic scenes.

AI USE STATEMENT

We used generative AI tools for language editing and translation, grammar and cross-section consistency checks, drafting and revising text from author-provided material, citation verification, and conversion of experimental results from text files into LaTeX tables. AI tools also assisted with parts of the implementation and with discussing interpretations of experimental results. The research ideas and algorithmic designs were developed by the authors. Generative AI was not used to synthesize experimental data, fabricate or alter measurements, or independently establish mathematical or empirical claims. The authors reviewed and verified all mathematical formulations, algorithmic designs, implementation changes, experimental protocols, quantitative results, references, and conclusions. Where applicable, AI-assisted code was manually inspected and tested against controlled cases or reference implementations. Suggested references and factual statements were checked against original sources. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies camera localization and scene geometry from panoramic imagery and LiDAR, without identifying individuals or analyzing personal attributes. We do not identify ethical concerns specific to the proposed reconstruction method. Panoramic recordings may incidentally contain identifiable people or vehicles. Before public release of PanoRecon, we will review the imagery for privacy-sensitive content and address applicable data-sharing permissions.

REPRODUCIBILITY STATEMENT

Section 3 presents the method, and Appendix B provides implementation details for depth scale calibration, long-range connection verification, and global refinement. Appendices A.1 and A.2 describe dataset acquisition and processing, reference construction, and baseline input and sampling settings. We will publicly release our source code and the PanoRecon benchmark to support independent reproduction and further research by the community.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Mostafa Ahmadi, Amin Alizadeh Naeini, Mohammad Moein Sheikholeslami, Zahra Arjmandi, Yujia Zhang, and Gunho Sohn. Hdpv-slam: Hybrid depth-augmented panoramic visual slam for mobile mapping system with tilted lidar and panoramic visual camera. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pp. 1–8. IEEE, August 2023. doi: 10.1109/case56687.2023.10260361. URL http://dx.doi.org/10.1109/CASE56687.2023.10260361.

Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, December 2021. ISSN 1941-0468. doi: 10.1109/tro.2021.3075644. URL http://dx.doi.org/10.1109/TRO.2021.3075644.

Thiago L. T. da Silveira, Paulo G. L. Pinto, Jeffri Murrugarra-Llerena, and Cláudio R. Jung. 3d scene geometry estimation from 360° imagery: A survey. *ACM Computing Surveys*, 55(4):1–39, November 2022. ISSN 1557-7341. doi: 10.1145/3519021. URL http://dx.doi.org/10.1145/3519021.

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, 2026. URL https://arxiv.org/abs/2507.16443.

Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, March 2018. ISSN 2160-9292. doi: 10.1109/tpami.2017.2658577. URL http://dx.doi.org/10.1109/TPAMI.2017.2658577.

Xiang Gao, Rui Wang, Nikolaus Demmel, and Daniel Cremers. Ldso: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2198–2204. IEEE, October 2018. doi: 10.1109/iros.2018.8593376. URL http://dx.doi.org/10.1109/IROS.2018.8593376.

Xiaopeng Guo, Yinzhe Xu, Huajian Huang, and Sai-Kit Yeung. 360dvo: Deep visual odometry for monocular 360-degree camera. *IEEE Robotics and Automation Letters*, 11(3):3079–3086, 2026a. doi: 10.1109/LRA.2026.3655280.

Yijing Guo, Mengjun Chao, Luo Wang, Tianyang Zhao, Haizhao Dai, Yingliang Zhang, Jingyi Yu, and Yujiao Shi. Panovggt: Feed-forward 3d reconstruction from panoramic imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36444–36453, 2026b.

Jared Heinly, Johannes L. Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days *(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Howard Huang, Bharath Surianarayanan, Keifer Lee, Chenyu Wang, and Chen Feng. Savmap: Structure-aided visual mapping of large-scale 2.5d manhattan wireframes from panoramic video, 2026. URL https://arxiv.org/abs/2606.01939.

Huajian Huang, Changkun Liu, Yipeng Zhu, Hui Cheng, Tristan Braud, and Sai-Kit Yeung. 360loc: A dataset and benchmark for omnidirectional visual localization with cross-device queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22314–22324, June 2024.

Satoshi Ikehata, Ivaylo Boyadzhiev, Qi Shan, and Yasutaka Furukawa. Panoramic structure from motion via geometric relationship detection, 2016. URL https://arxiv.org/abs/1612.01256.

San Jiang, Kan You, Wu Chen, Duojie Weng, and Yaxin Li. 3d reconstruction of spherical images based on incremental structure from motion. *International Journal of Remote Sensing*, 45(8):2596–2621, April 2024. ISSN 1366-5901. doi: 10.1080/01431161.2024.2334810. URL http://dx.doi.org/10.1080/01431161.2024.2334810.

Dongki Jung, Jaehoon Choi, Yonghan Lee, and Dinesh Manocha. Im360: Large-scale indoor mapping with 360 cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 29040–29050, October 2025.

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

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 71–91, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73220-1.

Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16651–16662, June 2025.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl(4) manifold. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 129839–129867. Curran Associates, Inc., 2025. doi: 10.52202/085713-4324. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/bc65ab11abfbad890171109686233f4e-Paper-Conference.pdf.

Dominic Rosario Maggio and Luca Carlone. VGGT-SLAM 2.0: Real-time Dense Feed-forward Scene Reconstruction. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.051.

Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

Raul Mur-Artal and Juan D. Tardos. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, October 2017. ISSN 1941-0468. doi: 10.1109/tro.2017.2705103. URL http://dx.doi.org/10.1109/TRO.2017.2705103.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16695–16705, June 2025.

Jing Ou, Zidong Cao, Yinrui Ren, Zhuoxiao Li, Jinjing Zhu, Tongyan Hua, Shuai Zhang, Hui Xiong, and Wufan Zhao. Holo360d: A large-scale real-world dataset with continuous trajectories for advancing panoramic 3d reconstruction and beyond. *arXiv preprint arXiv:2604.22482*, 2026.

Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L. Schönberger. Global structure-from-motion revisited. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 58–77, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73661-2.

Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, August 2018. ISSN 1941-0468. doi: 10.1109/tro.2018.2853729. URL http://dx.doi.org/10.1109/TRO.2018.2853729.

Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. Openvslam: A versatile visual slam framework. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pp. 2292–2295. ACM, October 2019. doi: 10.1145/3343031.3350539. URL http://dx.doi.org/10.1145/3343031.3350539.

Hartmut Surmann, Marc Thurow, and Dominik Slomma. Patchmatch-stereo-panorama, a fast dense reconstruction from 360° video images. In *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 366–372. IEEE, November 2022. doi: 10.1109/ssrr56537.2022.10018698. URL http://dx.doi.org/10.1109/SSRR56537.2022.10018698.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16558–16569. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/89fcd07f20b6785b92134bd6c1d0fa42-Paper.pdf.

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, April 1991. ISSN 0162-8828. doi: 10.1109/34.88573. URL http://dx.doi.org/10.1109/34.88573.

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

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522, June 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 (CVPR)*, pp. 20697–20709, June 2024a.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/11a09e0aaa74867c6b0719c639fc09f8-Paper-Conference.pdf.

Ze Wang, Kailun Yang, Hao Shi, Peng Li, Fei Gao, and Kaiwei Wang. Lf-vio: A visual-inertial-odometry framework for large field-of-view cameras with negative plane. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4423–4430. IEEE, October 2022. doi: 10.1109/iros47612.2022.9981217. URL http://dx.doi.org/10.1109/IROS47612.2022.9981217.

Ze Wang, Kailun Yang, Hao Shi, Peng Li, Fei Gao, Jian Bai, and Kaiwei Wang. Lf-vislam: A slam framework for large field-of-view cameras with negative imaging plane on mobile agents. *IEEE Transactions on Automation Science and Engineering*, 21(4):6321–6335, October 2024b. ISSN 1558-3783. doi: 10.1109/tase.2023.3324495. URL http://dx.doi.org/10.1109/TASE.2023.3324495.

Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 61–75, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10578-9.

Yiyang Wu, Xiaohu Zhang, Yanjin Du, Tongsu Zhang, Chujun Li, Siyang Chen, Guoyi Zhang, and Xiangpeng Xu. Panoair: A panoramic visual-inertial slam with cross-time real-world uav dataset, 2026. URL https://arxiv.org/abs/2604.00852.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3r: Scalable test-time training for large-scale 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21760–21771, June 2026.

Zhuang Xiong, Guohao Zhang, Chen Zhang, Zheyu Jiang, Yuchao Mei, Qingshan Xu, and Wenbing Tao. Look up and look back: Hidden attention and latent orientation in a frozen foundation model for panoramic slam, 2026. URL https://arxiv.org/abs/2608.00925.

Yueteng Yang, Yusen Xie, Hao Wei, Qianhao Wang, Boyu Zhou, Fei Gao, Jun Ma, and Jinni Zhou. Sphere-vio: Fast and robust visual-inertial odometry via unified spherical representation for heterogeneous multi-camera systems, 2026. URL https://arxiv.org/abs/2606.29910.

Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. Infinitevggt: Visual geometry grounded transformer for endless streams, 2026. URL https://arxiv.org/abs/2601.02281.

Zitong Zhan, Huan Xu, Zihang Fang, Xinpeng Wei, Yaoyu Hu, and Chen Wang. Bundle adjustment in the eager mode. *IEEE Transactions on Robotics*, 42:2727–2744, 2026. ISSN 1941-0468. doi: 10.1109/tro.2026.3706557. URL http://dx.doi.org/10.1109/TRO.2026.3706557.

Yiujia Zhang, SeyedMostafa Ahmadi, Jungwon Kang, Zahra Arjmandi, and Gunho Sohn. Yuto mms: A comprehensive slam dataset for urban mobile mapping with tilted lidar and panoramic camera integration. *The International Journal of Robotics Research*, 44(1):3–21, June 2024. ISSN 1741-3176. doi: 10.1177/02783649241261079. URL http://dx.doi.org/10.1177/02783649241261079.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8da04a60948be713dc766f0c7e3a5b1f-Paper-Conference.pdf.

A Additional Experiment Details

A.1 Datasets.

360LOC. We use the four daytime mapping sequences of 360LOC (Huang et al., 2024): Atrium, Concourse, Hall, and Piatrium. The released reference split contains 2,244 panoramas (491–632 per scene) at 6144×3072 resolution, with reference poses and LiDAR maps. Their indoor and outdoor settings, symmetric layouts, repeated structures, and moving pedestrians let us assess reconstruction consistency in visually ambiguous scenes.

For dense evaluation, we back-project the official per-ERP depth maps along spherical viewing rays and transform the resulting points into the world frame using the official camera poses. Valid points across mapping frames are merged by voxel averaging to form the reference point cloud.

YUTO MMS. We evaluate all four sequences (A–D) of YUTO MMS (Zhang et al., 2024), captured by a vehicle-mounted panoramic camera. The released sequences cover 20.1 km and contain 24,360 panoramas. Sequence C alone spans 9.137 km over 25 minutes with 10,778 images. The routes traverse campus and residential areas with large and small loops, frequent turns, and dynamic traffic. This scale and revisit structure let us evaluate accumulated drift and global consistency over extended trajectories, using reference poses obtained through offline GPS and LiDAR adjustment.

For dense evaluation, we construct the reference point cloud from the released LiDAR scans, ground-truth camera poses, and sensor calibration. We apply the calibrated LiDAR-to-camera transform, and accumulate the transformed scans in a common world frame.

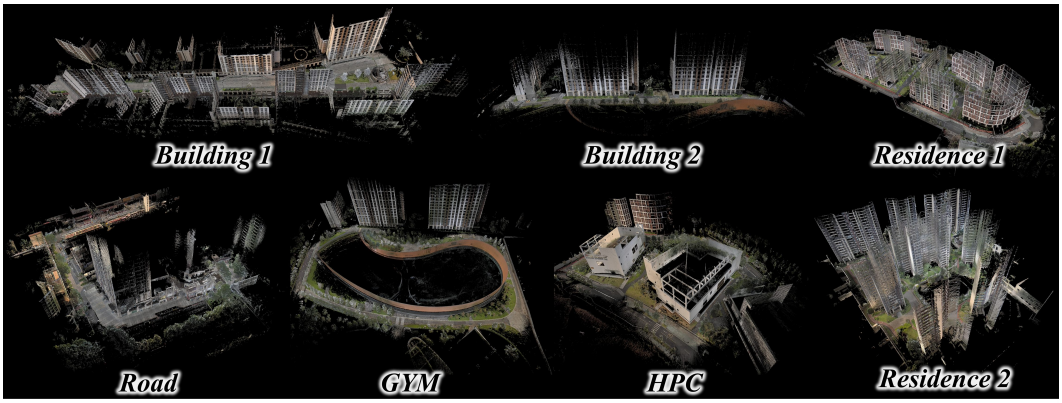

Figure 8: Visualization of ours PanoRecon benchmark ground-truth LiDAR geometry.

Our PanoRecon. The PanoRecon benchmark evaluates global consistency under repeated architecture and limited overlap between distant revisits. We select seven scenes with low-texture surfaces and buildings that share similar geometry and appearance (Fig. 8). Each planned route contains a large loop, smaller loops, and repeated visits to the same buildings after long travel distances and time intervals. At revisits, the buildings are viewed from farther away or at oblique angles and occupy only a small region of the ERP image. The resulting limited overlap tests whether reconstruction methods can establish long-range connections and correct accumulated drift.

Data acquisition. We independently acquire panoramic video and LiDAR data along the same planned route in each scene. The panoramic recordings use an Insta360 X4 at 5K resolution and 30 fps. We mount the camera on a 3 m extension pole to reduce the operator's presence in the ERP images. We export the recordings as ERP videos and sample 2K frames every 0.1 s. The separate LiDAR survey uses RTK positioning and three onboard cameras for point-cloud colorization.

Data processing. The two acquisition sessions are registered through reference images tied to the LiDAR coordinate system. We first post-process the LiDAR SLAM trajectory with RTK differential corrections and export the resulting point cloud, the three onboard cameras' poses, and their intrinsic

and extrinsic calibration. We import these data into *RealityScan* to render LiDAR reference images, whose camera poses remain fixed throughout registration.

Each ERP frame is split into perspective views with known intrinsics, a shared optical center, and fixed relative rotations. We import these views as a calibrated rig and globally match them to the fixed LiDAR reference images. Pose optimization updates only the panoramic rigs, with the reference cameras and rig calibration held fixed. We then convert the estimated perspective-view poses back to one ERP pose per frame using the rig extrinsics. This yields reference panoramic poses and LiDAR geometry in the same metric coordinate system.

A.2 Baselines Settings.

All methods use 2K input images except PanoVGGT and Scal3R, which use their official default resizing settings.

Input sequences and evaluation frames. On 360LOC, SLAM methods process all mapping images and retain frame-wise poses for evaluation. On PanoRecon, SLAM methods use all images for localization and retain evaluation keyframes at 1 s intervals. On YUTO MMS, SLAM methods process the full image sequences, and evaluation poses for both SLAM and feed-forward methods are sampled every three images. For these vehicle sequences with varying speeds, sampling follows frame indices. SfM methods use all 360LOC mapping images, sample PanoRecon inputs every 1 s, and use every 3 images, following the corresponding evaluation sampling intervals.

Feed-forward inference. Feed-forward methods receive the full YUTO MMS sequences, except that Scal3R uses every third image on Sequences C and D, where full-sequence inputs cause GPU out-of-memory (OOM) errors. PanoVGGT encounters OOM on every evaluated scene with its official default resizing and our single-GPU setup.

B Additional Method Implementation Details

We provide implementation details for depth scale calibration, long-range connection verification, and global refinement, following the order and notation of Sec. 3. Figure 9 shows intermediate reconstructions and how global correction resolves duplicated building surfaces.

B.1 Local Reconstruction Details

B.1.1 Sparse mapping and prediction windows

The panoramic frontend tracks ERP images through projection matching and camera pose optimization (Fig. 9(a)). New keyframes add observations and triangulated landmarks, which local BA refines before depth scale calibration.

We use the overlapping windows specified in Implementation. The first and last two inferred panoramas supply context for Pi3 and cached depths for connection matching. Scale fitting and point-cloud output use the interior frames.

We use the camera-to-world poses T_i defined in Sec. 3.1, with the camera's y axis pointing downward. For an ERP image of width W and height H, pixel $\mathbf{u} = (u, v)$ has longitude $\theta = 2\pi(u/W - 1/2)$ and upward-positive latitude $\phi = \pi(1/2 - v/H)$. Its unit bearing and the inverse projection are

$$\begin{aligned} \mathbf{b}(u,v) &= (\cos\phi\sin\theta,\ -\sin\phi,\ \cos\phi\cos\theta)^\top, \\ \pi_{\mathrm{ERP}}(\mathbf{x}) &= \begin{bmatrix} W\big(\frac{1}{2} + \mathrm{atan2}(x,z)/(2\pi)\big) \\ H\big(\frac{1}{2} + \arcsin(y/\|\mathbf{x}\|_2)/\pi\big) \end{bmatrix}, \quad \mathbf{x} = (x,y,z)^\top. \end{aligned} \tag{6}$$

The horizontal coordinate is periodic. Reconstruction distances use the sparse map's scale, with metric scale recovered through alignment for evaluation.

Each virtual perspective view has known intrinsics K_v, rotation Q_v into the panoramic camera, and the same optical center as that camera. We take the optical-axis component $\hat{z}$ of each Pi3 local

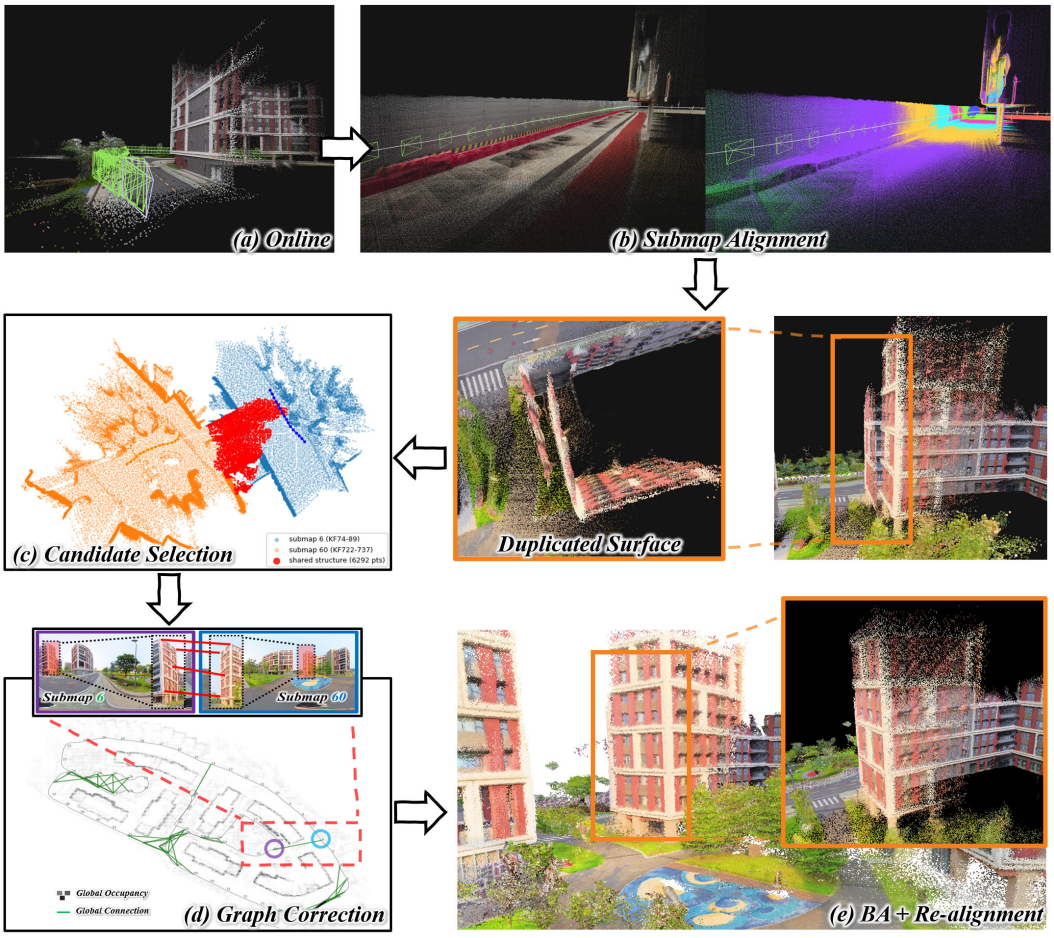

Figure 9: **From online submaps to globally consistent geometry.** (a) Online pose estimation and local reconstruction. (b) Submap alignment, with distinct colors indicating different submaps. (c) Candidate selection from geometric overlap. (d) Graph correction using verified long-range connections. (e) BA and realignment remove the duplicated building surfaces left by pose drift after online alignment.

point and back-project it along the calibrated ray $K_v^{-1}\bar{\mathbf{p}}$. After scale calibration, its distance from the camera is $d = s_{wi}\hat{z}\|K_v^{-1}\bar{\mathbf{p}}\|_2$. We store the source keyframe, virtual view, pixel, and unscaled depth so that Eq. (1) can update its world position after optimization.

B.1.2 Landmark observations and depth scale calibration

For landmark $\mathbf{L}_j$ observed by keyframe i, the geometric depth in virtual view v is

$$z_{ijv}^{\text{geom}} = \left[Q_v^\top R_i^\top (\mathbf{L}_j - \mathbf{c}_i)\right]_z, \tag{7}$$

where $[\cdot]_z$ selects the optical-axis component. Comparing this depth with the prediction at the corresponding pixel gives the log ratio in Eq. (2). The fit estimates multiplicative depth scales, which are applied to the original predicted depths in Eq. (1), producing the aligned submaps shown in Fig. 9(b).

We filter landmark observations by positive depth, observation support, parallax, and reprojection error, and exclude masked pixels, view boundaries, and depth discontinuities. We repeat these checks against the updated sparse map before recalibration.

Before fitting, we filter log ratios by median absolute deviation. Frames with sufficient landmark support define $\mathcal{A}_w$, and windows with enough pooled observations are calibrated using Eq. (2).



Figure 10: Point-cloud comparison of building facades and local structures in the PanoRecon benchmark's Building1 scene.

Temporal regularization links eligible frames that are adjacent in the sequence. The shared window scale initializes all frames, with $b_{wi} = 0$, and frame corrections are fitted for eligible frames.

We solve Eq. (2) by iteratively reweighted least squares with $(\lambda_p, \lambda_s) = (32, 64)$ and the Huber threshold specified in Implementation. The shared log scale a_w starts from the median log ratio and is fitted without a bound. Frame corrections start at zero and are clipped after fitting to $\pm\log(1.05)$ relative to the shared scale.

Overlapping windows provide an additional scale adjustment from depths at common source keyframes, views, and pixels. Each shared keyframe contributes a median log-depth ratio, and their sample-count-weighted median gives the correction toward the preceding window.

B.2 Global Connection Details

B.2.1 Candidate submap selection

We use calibrated submap points to restrict feature matching to spatial candidates (Fig. 9(c)). For occupied voxel sets V_A, V_B and one-voxel dilation $\mathcal{D}$, the overlap score is

$$\eta_{AB} = \max\left\{ \frac{|V_A \cap \mathcal{D}(V_B)|}{|V_A|}, \frac{|V_B \cap \mathcal{D}(V_A)|}{|V_B|} \right\}. \tag{8}$$

Taking the larger fraction accommodates partial overlap between submaps of different sizes. We use voxel size $0.25m$ in map units and require $\eta_{AB} \geq 0.15$. A line-of-sight check tests for occlusion along rays from camera centers toward the overlapping region, requiring a mean visibility fraction of at least 0.6.

To include nearby candidates displaced by drift, we expand robust axis-aligned submap bounds by 20% on each side. Intersecting bounds supply up to three additional candidates. We also consider historical windows within two indices of selected candidates, adding at most 12 pairs per new window while retaining all original visibility candidates. Within this spatial neighborhood, candidate submaps separated by at least four window indices undergo the matching and verification below.

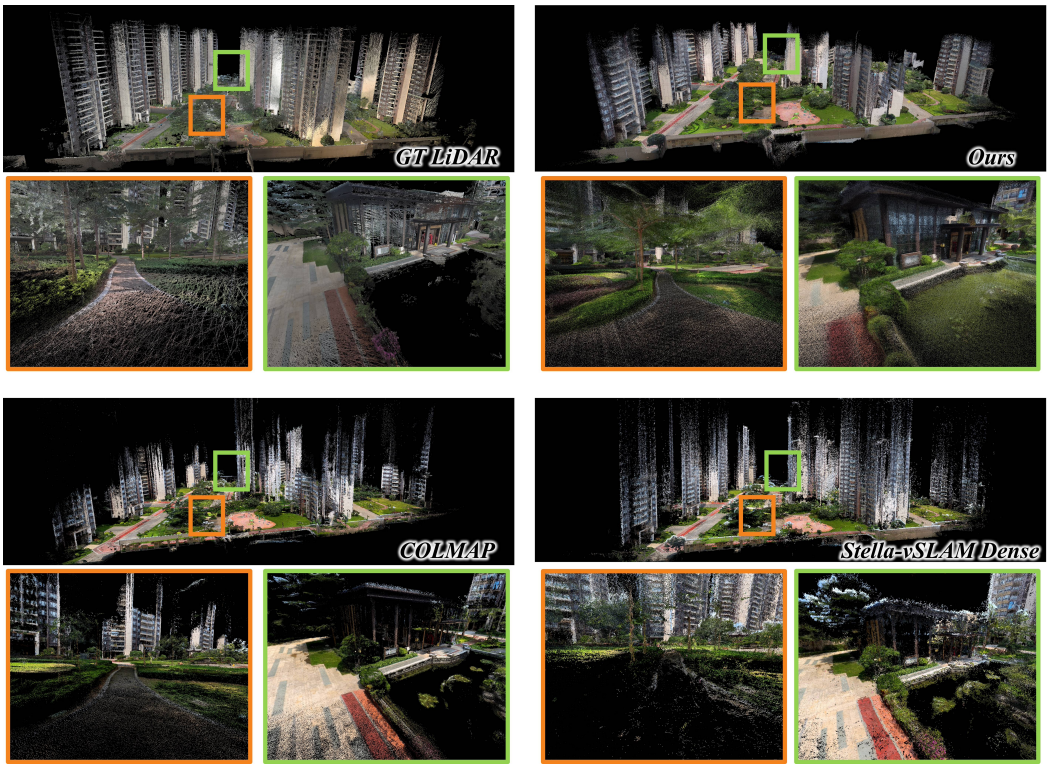

Figure 11: Point-cloud comparison in the Residence scene.

B.2.2 Feature matching and geometric verification

We match all inferred keyframes with available poses and depths, including frames excluded from point-cloud output. The median predicted camera distance in a local ERP neighborhood lifts each matched feature into 3D. Matches with valid depths on both sides form the correspondences in Sec. 3.2.

Each match retains its source keyframes and feature indices to distinguish repeated use of a feature from distinct feature support. We pool the 3D correspondences to estimate $S = S_{A \leftarrow B}$, with $S\mathbf{x} = sR\mathbf{x} + \mathbf{t}$, using RANSAC followed by least-squares refitting (Umeyama, 1991).

For corresponding points $\mathbf{P}_a, \mathbf{P}_b$ and source camera centers $\mathbf{c}_a, \mathbf{c}_b$, geometric verification requires

$$\begin{aligned} \|S\mathbf{P}_b - \mathbf{P}_a\|_2 &\le \tau_d, \\ \|S^{-1}\mathbf{P}_a - \mathbf{P}_b\|_2 &\le \tau_d, \\ \angle(S\mathbf{P}_b - \mathbf{c}_a, \mathbf{P}_a - \mathbf{c}_a) &\le \tau_\theta, \\ \angle(S^{-1}\mathbf{P}_a - \mathbf{c}_b, \mathbf{P}_b - \mathbf{c}_b) &\le \tau_\theta. \end{aligned} \tag{9}$$

Here $\angle(\mathbf{x}, \mathbf{y})$ is the angle between nonzero vectors, and τ_d and τ_θ are the distance and bearing thresholds specified in Implementation. The bidirectional distance test checks each submap at its own scale. We refit the similarity from the surviving correspondences and recompute inliers before the following checks.

B.2.3 Validation across views and camera constraints

We require inliers from multiple views because repeated matches can refer to the same scene point. Table 8 specifies the required support, including distinct features and keyframes on each side. We reject nearly collinear inlier points on either side while allowing planar configurations.

Large corrections require additional validation using separate frame groups. Let h be the median spacing between consecutive available keyframe centers, and let $\mathcal{P}_B$ contain the current submap's



Figure 12: Point-cloud comparison in the HPC scene.

frames with predicted depths. A correction is large if its rotation exceeds 3°, its mean displacement of these camera centers exceeds h, or its scale lies outside $[0.95, 1.05]$.

Before fitting, we divide the current frames into two similarly sized groups with balanced match counts. We estimate a transform from each group and verify it on the other group's matches. Each estimate $S_f = (s_f, R_f, \mathbf{t}_f)$ must also agree with the estimate from all matches, $S = (s, R, \mathbf{t})$:

$$\begin{aligned} |\log(s_f/s)| &\le 0.05, \\ \operatorname{angle}(R_f R^\top) &\le 3^\circ, \\ \max_{i \in \mathcal{P}_B} \|S_f \mathbf{c}_i - S\mathbf{c}_i\|_2 &\le \max(2h, 2\tau_d). \end{aligned} \tag{10}$$

Here angle denotes the rotation angle. The groups use disjoint current frames and can share historical frames and scene content. The same transform-agreement criteria apply to verified historical submaps within two window indices that connect to the same current submap.

From each accepted submap pair, we select image pairs with at least ten inliers in descending count order. Their camera endpoints must be distinct on each side, and duplicate camera pairs are excluded across selected submap pairs. At most three camera constraints are formed using Eq. (3). For n_{AB} inliers and $m_{AB} > 0$ selected constraints, each constraint receives weight

$$w_e = \frac{\operatorname{clip}(n_{AB}/3, 10, 100)}{m_{AB}}. \tag{11}$$

This normalization keeps each submap pair's total weight fixed across different numbers of selected camera edges.

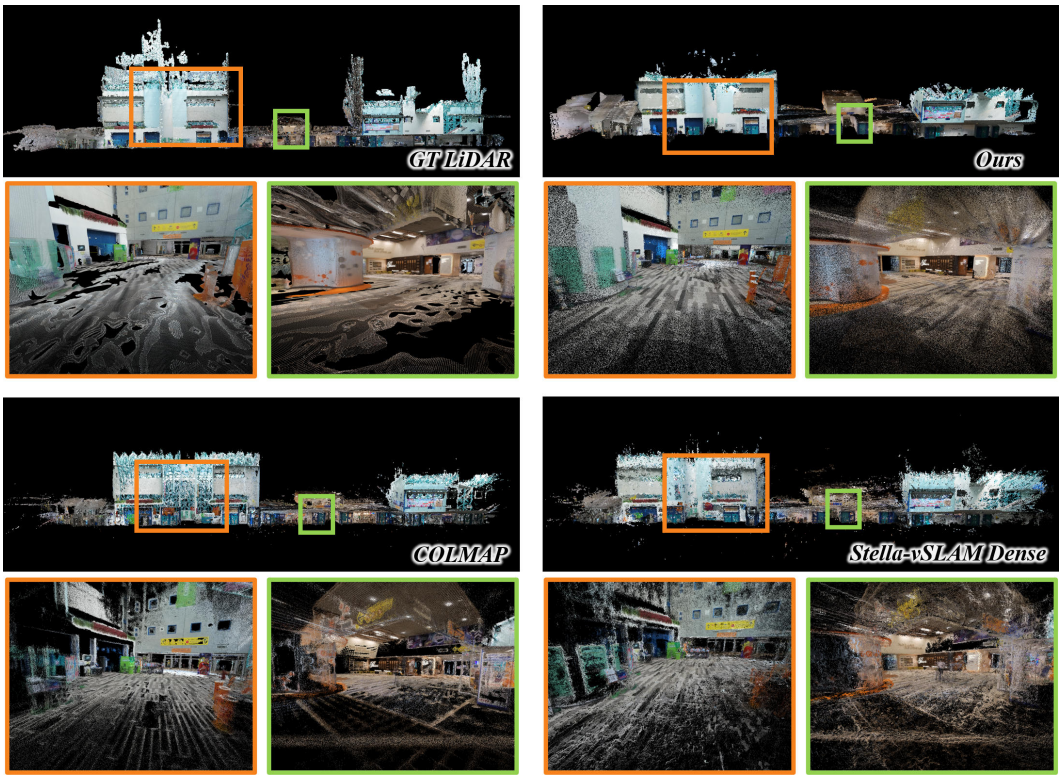

Figure 13: Point-cloud comparison in the 360LOC Concourse scene.

Table 8: Connection-verification thresholds. Distances use map units, and h is the median keyframe spacing.

Quantity	Requirement
Total inliers, small / large correction	≥ 60 / ≥ 100
Inlier fraction, small / large correction	≥ 0.35 / ≥ 0.50
Unique features per side	≥ 40
Supporting keyframes per side	≥ 3, each with ≥ 10 inliers
Scale / rotation / mean camera displacement	$[2/3, 1.5]$ / $\leq 15^\circ$ / $\leq 50h$
Bidirectional distance / viewing angle	≤ 0.3 / $\leq 1^\circ$
Estimation-group inliers	≥ 30
Validation-group inliers / fraction	≥ 30 / ≥ 0.40
Group and neighboring-transform agreement	Eq. (10)

B.2.4 Descriptor matching and cache reuse

The CPU and CUDA implementations use the same mutual ORB matching rules and cache descriptors and feature matches. After global correction, we recompute 3D coordinates and geometric verification from the updated poses and depths.

B.3 Global Refinement Details

B.3.1 Pose graph optimization and landmark fusion

The pose graph combines spanning-tree edges, strong covisibility edges, and the new camera constraints from Sec. 3.2 to correct the trajectory (Fig. 9(d)). Let $G_i \in \mathrm{Sim}(3)$ be a world-to-camera vertex, initialized from T_i^{-1} with unit scale. For edge set $\mathcal{E}$, relative measurements $\widehat{G}_{b\leftarrow a}$, and

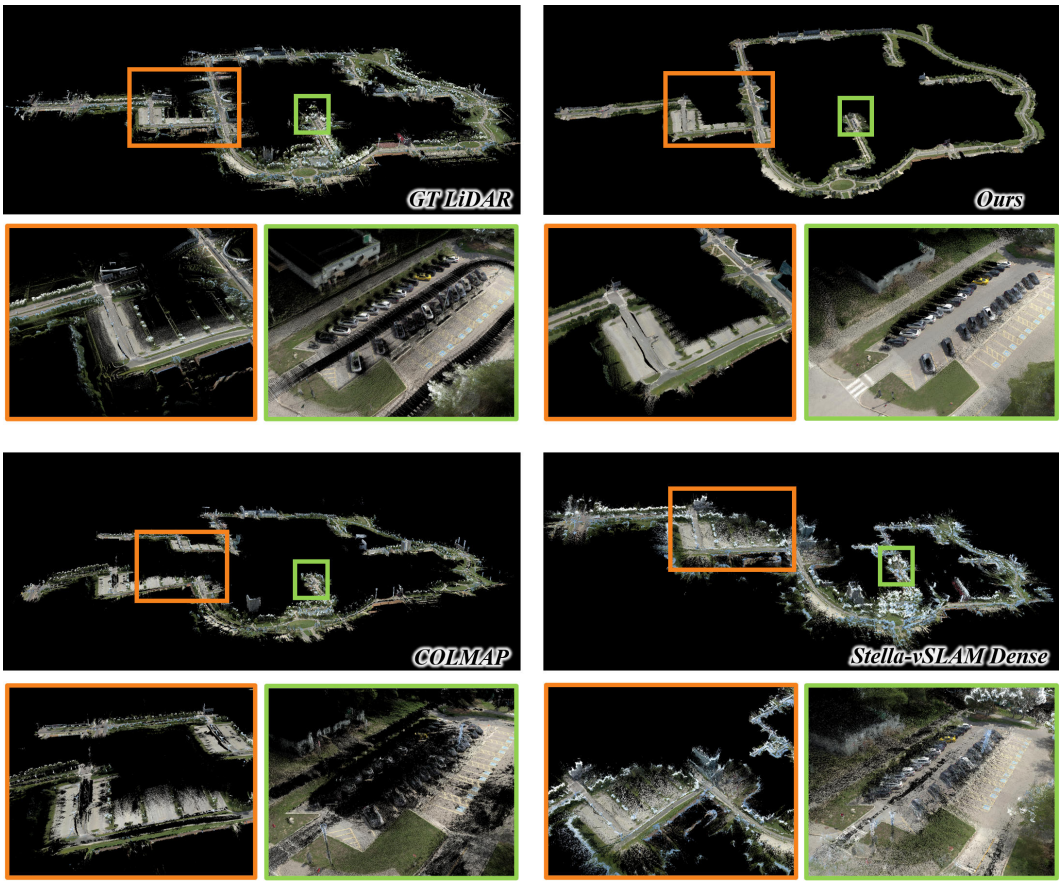

Figure 14: Point-cloud comparison in the YUTO MMS Sequence B scene.

weights w_{ab}, we fix the root vertex and minimize

$$\min_{\{G_i\}} \sum_{(a,b)\in\mathcal{E}} w_{ab} \left\| \mathrm{Log}\left(\widehat{G}_{b\leftarrow a} G_a G_b^{-1} \right) \right\|_2^2 . \tag{12}$$

Here Log maps a similarity transform to its seven-dimensional tangent vector. Existing edges have unit weight, and new edges use Eq. (11). After optimization, we check each new edge's relative error $H_{ab} = \widehat{G}_{b\leftarrow a}^{-1} G_b G_a^{-1} = (s_H, R_H, \mathbf{t}_H)$ against

$$\mathrm{angle}(R_H) \le 5^\circ, \qquad \|\mathbf{t}_H / s_H\|_2 \le 2, \qquad |\log s_H| \le 0.1. \tag{13}$$

The translation threshold uses map units. Pruning inconsistent new edges is followed by graph re-optimization and residual checks on the retained new constraints.

For each optimized world-to-camera similarity, dividing its translation by its scale gives the corresponding rigid transform. Inverting this transform yields the updated camera-to-world pose T_i^+. A landmark associated with reference keyframe $r(j)$ is updated by

$$\mathbf{L}_j^+ = (G_{r(j)}^+)^{-1} G_{r(j)}^0 \mathbf{L}_j^0, \tag{14}$$

where superscripts 0 and $+$ denote the states before and after graph optimization.

We then apply the frontend's bidirectional projection matching between accepted camera endpoints and a limited set of their strongest covisible neighbors. Duplicate-landmark fusion and covisibility updates produce the shared observation set $\mathcal{O}$ for global BA.

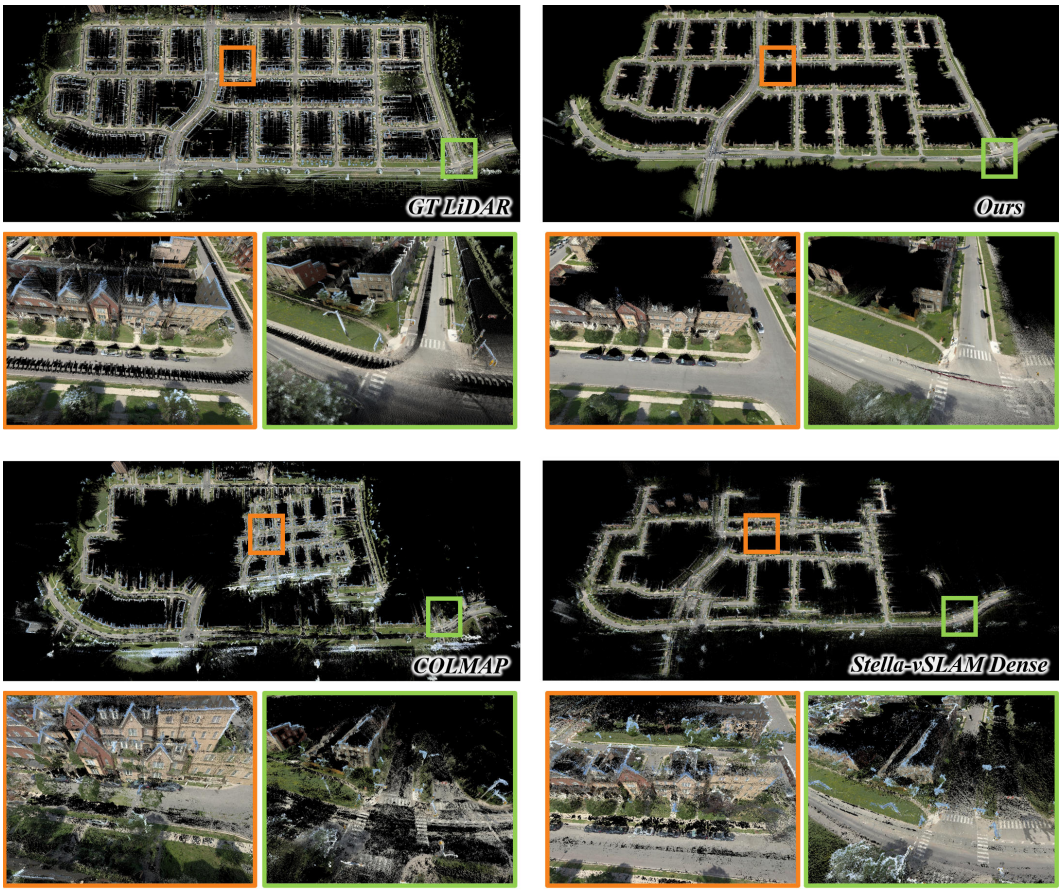

Figure 15: Point-cloud comparison in the YUTO MMS Sequence C scene.

B.3.2 Panoramic bundle adjustment

We refine one rigid pose per panorama and the shared landmarks using the ERP reprojection objective in Eq. (4). The root pose is fixed, while monocular scale remains unconstrained. The horizontal residual wraps to $[-W/2, W/2)$, and the Huber threshold is $\delta_b = \sqrt{5.99146}$.

Residuals for sparse automatic differentiation. We implement this objective in the BAE/PyPose framework (Zhan et al., 2026). To express the robust loss as a sum of squared residuals, define $\mathbf{e}_{ij} = \sigma_{ij}^{-1}\mathbf{r}_{ij}$ and $q_{ij} = \|\mathbf{e}_{ij}\|^2$, and supply

$$\mathbf{f}_{ij} = \begin{cases} \sqrt{\psi_{\delta_b}(q_{ij})/q_{ij}}\,\mathbf{e}_{ij}, & q_{ij} > 0, \\ \mathbf{0}, & q_{ij} = 0. \end{cases} \qquad \sum_{(i,j)\in\mathcal{O}} \|\mathbf{f}_{ij}\|^2 = E_{\mathrm{BA}}. \tag{15}$$

The CPU and GPU backends optimize the same scalar objective. The GPU implementation differentiates the residual scaling, while g2o uses a robust Gauss–Newton approximation.

Matrix-free Schur optimization. Let J_c and J_p be the Jacobians of the stacked residual vector $\mathbf{f}$ with respect to free camera poses and landmarks. Diagonal damping gives camera and landmark blocks U_λ and V_λ, with coupling $C = J_c^\top J_p$. For right-hand sides $\mathbf{b}_c = -J_c^\top \mathbf{f}$ and $\mathbf{b}_p = -J_p^\top \mathbf{f}$, eliminating landmark increments gives

$$\begin{aligned} &S_\lambda\,\Delta\mathbf{c} = \mathbf{b}_c - CV_\lambda^{-1}\mathbf{b}_p, \quad S_\lambda = U_\lambda - CV_\lambda^{-1}C^\top, \\ &V_\lambda\,\Delta\mathbf{p} = \mathbf{b}_p - C^\top\Delta\mathbf{c}. \end{aligned} \tag{16}$$

Here λ denotes damping, and $\Delta\mathbf{c}$ and $\Delta\mathbf{p}$ stack camera pose and landmark increments. The reduced camera operator is evaluated through sparse products:

$$S_\lambda \mathbf{x} = U_\lambda \mathbf{x} - J_c^\top \left[J_p \left(V_\lambda^{-1} \left[J_p^\top (J_c \mathbf{x}) \right] \right) \right]. \tag{17}$$

This avoids storing C or S_λ. The block-diagonal V_λ has 3×3 blocks that can be inverted separately. GPU preconditioned conjugate gradients solve the camera system using a preconditioner derived from U_λ. The implementation also uses preconditioned conjugate gradients to recover landmark increments from the second equation in Eq. (16).

Solver settings and map updates. The GPU solver uses the precision and PCG settings in Implementation, with trust-region damping and at most 100 outer iterations. After at least two iterations, relative objective improvement below 10^{-3} stops refinement. Accepted steps have non-increasing objective values, and the final solution updates the camera poses and landmarks. The CPU backend uses analytic ERP Jacobians, g2o robust linearization, and a CSparse direct solver with OpenMP support. Initialization and local BA use the CPU backend, while the selected backend handles global BA.

B.3.3 Dense reconstruction after global refinement

Updated poses and landmarks change both dense point positions and their depth references. For each affected window, we recheck its landmark observations, recompute Eq. (7), refit Eq. (2), and adjust overlapping predictions through their common samples. Equation (1) then places the cached predictions using the updated poses and scales, yielding the realigned geometry in Fig. 9(e).

At each submap boundary, accepted connections trigger graph correction, landmark fusion, and global BA before dense updates. Scheduled global BA also updates dense geometry. Frame input resumes once these operations finish.

C Additional Experiment Results

C.1 Runtime Analysis

Table 9 summarizes the instrumented reconstruction stages on YUTO MMS Sequence C. Our 257 Pi3 inference windows take 38.96 minutes, while 91 GPU BA solves take 0.87 minutes; connection matching and verification account for 56.60 minutes. In COLMAP, incremental reconstruction and BA take 618.52 minutes, and photometric and geometric MVS together take 609.85 minutes. We exclude separate image preparation, format conversion, and export. These historical runs use 10,778 frontend frames and 3,594 keyframes for our method, versus 3,593 sampled ERP images expanded into 43,116 perspective views for COLMAP.

Table 9: **Key-stage runtime on YUTO MMS Sequence C.** Times are in minutes, excluding separate preprocessing and export.

Ours		COLMAP (Pinhole)	
Stage	Time	Stage	Time
Frontend processing	42.71	Feature extraction	7.21
Pi3 inference	38.96	Matching and verification	26.17
Connection matching	44.66	Rig verification	7.39
Connection verification	11.94	Incremental reconstruction + BA	618.52
Pose-graph optimization	10.08	Photometric MVS	227.33
GPU BA solve	0.87	Geometric MVS	382.52
Post-BA realignment	32.16	Fusion	261.72
Total	181.38	Total	1530.86

C.2 Additional Analysis

Figure 16 shows global connections across revisited regions in PanoRecon and YUTO MMS.

Geometry source and cross-view aggregation. On the evaluated Residence2 submap pairs, predicted depth provides roughly four times the 3D coverage of sparse landmarks, measured as the fraction of selected matches with valid 3D support on both sides (Table 10). Correct recall is the fraction of evaluated reference-positive pairs accepted with an accurate transform. Pooling views improves this recall for both geometry sources despite slightly lower coverage. Their combination yields the highest correct recall, indicating that geometric support and its distribution across views jointly affect the recovery of valid connections.

Table 10: **Geometry source and cross-view verification on Residence2.**

Geometry and aggregation	3D coverage (%) ↑	Correct recall (%) ↑
Sparse, single	6.37	6.25
Sparse, multi	5.82	13.19
Predicted depth, single	25.77	12.50
Predicted depth, multi	23.52	15.42

Camera-level versus submap-level correction. Within the same 100-evaluation budget and without BA, camera-level correction yields a modest ATE reduction with nearly unchanged RPE (Table 11). A shared submap similarity moves its cameras together, whereas per-camera variables allow corrections to vary within a window. This improves global trajectory alignment in this comparison while retaining comparable local motion consistency.

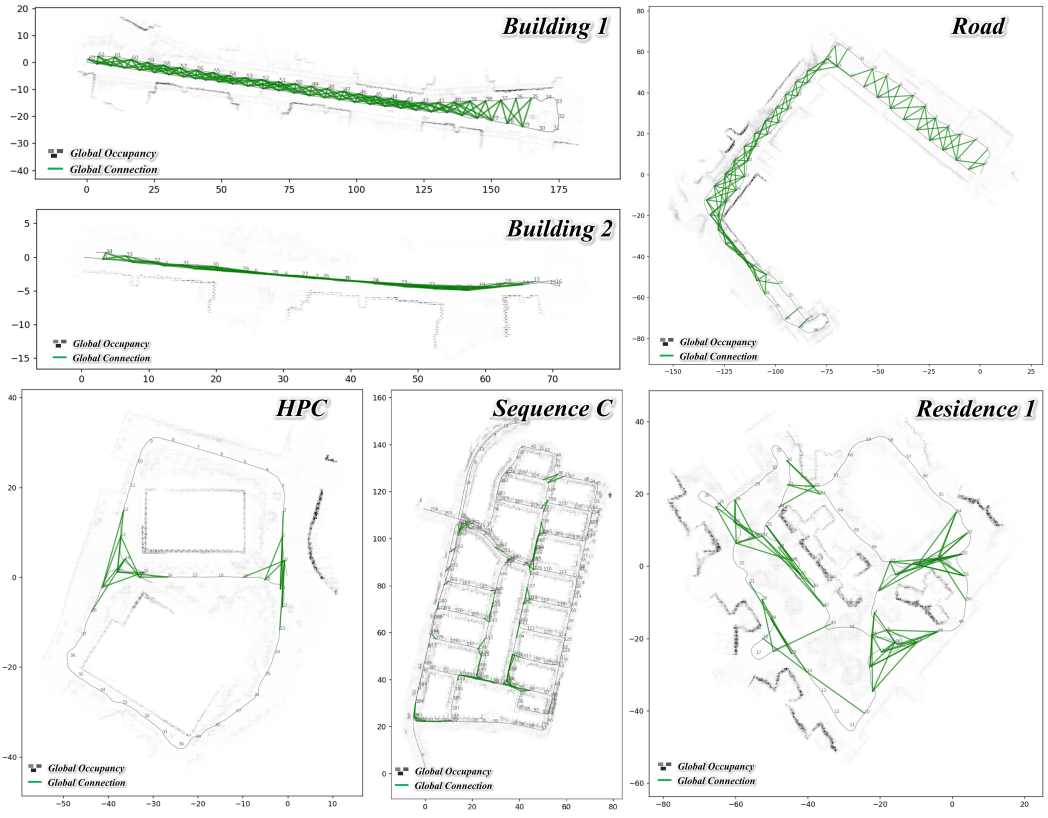

Figure 16: **Global connections across scenes.** Green edges connect revisited submaps, with trajectories and occupancy maps shown in gray.

Table 11: **Graph parameterization.** Both graphs use the same accepted measurements, pair-wise total weights, local edges, objective, and 100-evaluation limit, without BA.

Correction	ATE	RPE-t	RPE-r
Per-camera	0.749	0.060	0.046
Per-submap (16 ERP views)	0.778	0.060	0.047

Table 12: **Candidate recall under added drift.** Percentages on six Residence2 revisits. Joint perturbation combines 3 m translation, 10° yaw, and $1.1\times$ scale.

Added perturbation	$K = 6$		20% bounds expansion		
	Bounds off	10% expansion	$K = 3$	$K = 6$	$K = 12$
None	17.91	47.76	25.37	49.25	65.67
Translation 1 m	16.92	47.76	25.37	49.25	65.67
Translation 3 m	16.42	48.26	24.88	48.76	65.17
Translation 5 m	14.93	48.76	24.88	48.76	64.68
Yaw 5°	17.91	47.76	25.37	49.25	65.67
Yaw 10°	16.42	47.76	25.37	49.25	65.67
Yaw 20°	16.42	49.25	25.37	49.25	65.67
Scale $0.8\times$	8.96	47.76	25.37	49.25	56.72
Scale $1.2\times$	7.46	50.75	25.37	49.25	61.19
Joint	12.94	49.75	24.88	49.75	63.68

C.2.1 Candidate Retrieval under Controlled Drift

We evaluate candidate selection on six reference-positive revisit events sampled across Residence2 reconstruction. Reference poses and LiDAR visibility estimates identify 67 positive submap pairs separated by at least 60 s. A pair is positive when its shared visible LiDAR points cover at least 10% of the smaller visible set. We perturb the current submap's points and camera poses while keeping historical submaps fixed, then apply the candidate selection in Appendix B.2.1. Candidate recall is the fraction of these positive pairs retrieved among the first K candidates. Table 12 aggregates counts across the six events and three perturbation seeds. A 0% expansion disables proposals from expanded bounds.

Expanded bounds recover candidates missed by voxel overlap and visibility tests, while increasing expansion from 10% to 20% produces smaller changes. At 20% expansion, increasing K from 6 to 12 raises unperturbed pair recall from 49.25% to 65.67%, showing that the candidate budget limits coverage even when spatial search finds a revisit. With $K = 6$ and 20% expansion, every evaluated event retrieves at least one positive candidate under all tested perturbations. Thus, finding a revisit and retrieving all overlapping historical submaps are distinct outcomes.

Increasing K improves candidate recall in each revisit interval, measured between submap mean timestamps (Table 13). The longest interval contains seven positive pairs from one current submap.

C.3 Applying Submap Alignment to PanoVGGT

PanoVGGT's joint inference exceeds GPU memory on the complete sequences in our evaluation, motivating a windowed reconstruction pipeline for long panoramic inputs. We therefore combine its panoramic predictions with the submap construction and retrieval workflow of VGGT-SLAM (Guo et al., 2026b; Maggio et al., 2025). Since this integration is our own implementation, we present it separately in the supplement as a Residence2 case study, rather than as an official baseline. The experiment examines whether ours insight of using predicted geometry to align submaps and propagate global corrections can also benefit this reconstruction pipeline.

Re-implementation. We process Residence2 images at 518×1036 resolution in windows advancing by 16 images as submaps, with one image shared between consecutive windows. We then retrieval supplies at most one historical image for joint inference with each current window. PanoVGGT predicts camera poses and dense geometry directly from the ERP images. We express

Table 13: **Candidate recall by revisit interval.** No added perturbation and 20% bounds expansion. Recall is reported in percent.

Revisit interval (min)	Positive pairs	$K = 3$	$K = 6$	$K = 12$
$[1, 3)$	28	25.00	50.00	60.71
$[3, 10)$	32	21.88	43.75	62.50
≥ 10	7	42.86	71.43	100.00
All intervals	67	25.37	49.25	65.67

Table 14: **Adapt PanoVGGT with VGGT-SLAM on Residence2.**

Configuration	ATE ↓ (m)	RPE-t ↓ (m)	RPE-r ↓ (°)	Precision ↑ (%)	Recall ↑ (%)	F-score ↑ (%)
Default SL(4)	Failed at submap 57/59					
Adapted SE(3)	26.831	0.430	3.616	2.31	45.83	4.40
Without loop factors SE(3)	29.970	0.294	**0.112**	4.55	51.09	8.35
Ours submap alignment Sim(3)	**4.203**	**0.107**	0.149	**10.60**	**78.14**	**18.67**

these predictions in the first camera's coordinate system and pass them to the VGGT-SLAM backend, retaining its sequential scale estimation and graph optimization. Valid points have finite coordinates and positive radial depth, since the panoramic checkpoint does not provide learned depth confidence. This produces the adapted-backend reference in Table 14.

Geometry-based submap alignment. We transfer our submap alignment to this adapter using images shared across adjacent or retrieved windows. Corresponding pixels provide radial-depth ratios for robust relative scale estimation, while the predicted poses of the shared camera determine relative rotation and translation. These measurements form Sim(3) constraints for jointly optimizing submap poses and scales with a Huber loss, fixing the first submap. The optimized transformations update cached camera poses and dense points without rerunning PanoVGGT, propagating global corrections throughout the reconstruction.

Results. On Residence2, geometry-based submap alignment reduces ATE from 26.831 to 4.203 m and raises F-score from 4.40% to 18.67%. Removing loop factors alone yields the lowest rotational RPE but substantially worse trajectory and reconstruction accuracy.

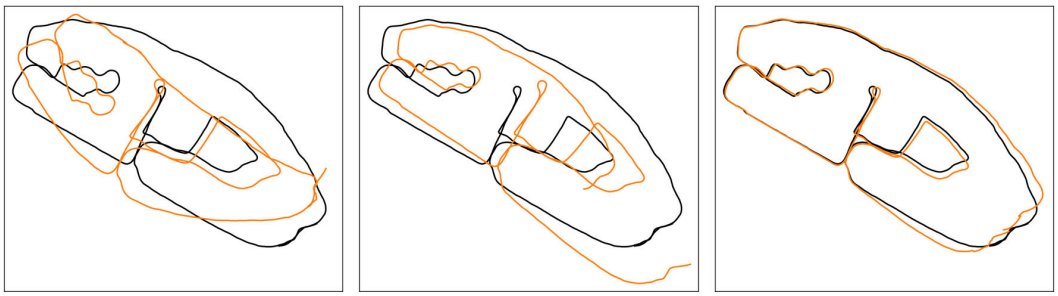

Figure 17: Applying Submap Alignment to PanoVGGT

D Per-Scene Localization Results

Tables 15, 16, and 17 provide per-scene trajectory results for 360LOC, YUTO MMS, and PanoRecon, respectively.

Table 15: **360LOC localization by scene.** ATE and RPE-t are in meters; RPE-r is in degrees. n is the number of associated poses (scene panels) or available/expected scenes (Mean). All errors are RMSE; lower is better.

Atrium					**Concourse**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	581	3.322	0.095	0.841	Scal3R	491	5.717	0.130	0.611
360DVO	581	0.654	0.044	0.034	360DVO	491	0.779	0.011	0.044
Open-vSLAM	580	0.255	0.691	22.408	Open-vSLAM	471	**0.081**	0.670	20.111
Stella-vSLAM	285	1.209	0.083	0.059	Stella-vSLAM	262	0.172	0.037	0.298
COLMAP (ERP)	581	**0.201**	**0.043**	**0.033**	COLMAP (ERP)	491	0.131	0.007	0.038
COLMAP (Pinhole)	581	0.232	0.043	0.033	COLMAP (Pinhole)	491	0.159	**0.007**	**0.037**
Spherical-SfM (ERP)	581	0.354	0.043	0.033	Spherical-SfM (ERP)	491	0.211	0.010	0.042
Ours	581	0.208	0.062	0.040	**Ours**	491	0.169	0.016	0.040
Hall					**Piatrium**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	540	11.008	0.172	3.758	Scal3R	632	1.417	0.099	0.205
360DVO	540	0.921	0.028	**0.039**	360DVO	632	1.460	0.021	0.038
Open-vSLAM	540	0.714	0.721	23.225	Open-vSLAM	612	0.713	0.762	22.993
Stella-vSLAM	363	0.793	0.035	0.064	Stella-vSLAM	350	0.652	0.033	0.167
COLMAP (ERP)	540	0.919	0.018	0.041	COLMAP (ERP)	632	**0.170**	**0.013**	**0.035**
COLMAP (Pinhole)	540	1.027	0.021	0.041	COLMAP (Pinhole)	632	0.177	0.013	0.035
Spherical-SfM (ERP)	–	1.093	0.027	0.042	Spherical-SfM (ERP)	540	0.308	0.015	0.036
Ours	540	**0.218**	**0.015**	0.054	**Ours**	632	0.484	0.027	0.041
Mean									
Method	n	ATE↓	RPE-t↓	RPE-r↓					
Scal3R	4/4	5.366	0.124	1.354					
360DVO	4/4	0.954	0.026	0.039					
Open-vSLAM	4/4	0.441	0.711	22.184					
Stella-vSLAM	4/4	0.706	0.047	0.147					
COLMAP (ERP)	4/4	0.355	**0.020**	0.037					
COLMAP (Pinhole)	4/4	0.399	0.021	**0.037**					
Spherical-SfM (ERP)	4/4	0.491	0.024	0.038					
Ours	4/4	**0.270**	0.030	0.044					

Table 16: **YUTO MMS localization by scene.** ATE and RPE-t are in meters; RPE-r is in degrees. n is the number of KF poses or available/expected scenes (Mean).

Sequence A					**Sequence B**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	234	24.591	0.301	0.409	Scal3R	2794	307.710	0.948	0.381
360DVO	234	0.244	**0.006**	**0.010**	360DVO	2794	135.769	0.277	**0.013**
Open-vSLAM	235	35.215	15.562	56.725	Open-vSLAM	2794	479.963	37.245	1.025
Stella-vSLAM	235	0.370	0.026	0.060	Stella-vSLAM	2794	252.571	1.074	0.223
COLMAP (ERP)	234	0.161	0.008	0.014	COLMAP (ERP)	2794	412.900	2.060	0.197
COLMAP (Pinhole)	234	0.474	0.017	0.017	COLMAP (Pinhole)	2794	75.064	0.954	0.036
Spherical-SfM (ERP)	234	0.409	2.036	4.139	Spherical-SfM (ERP)	2794	26.071	4.008	3.960
Ours	235	**0.082**	0.009	0.018	**Ours**	2795	**3.367**	**0.041**	0.024
Sequence C					**Sequence D**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	3593	236.298	9.695	3.007	Scal3R	1500	181.028	0.700	0.302
360DVO	3593	115.432	0.366	**0.011**	360DVO	1500	21.804	**0.083**	**0.011**
Open-vSLAM	3593	280.763	12.037	6.752	Open-vSLAM	1500	331.233	101.917	85.795
Stella-vSLAM	3593	12.848	0.714	0.166	Stella-vSLAM	1500	62.904	0.812	0.076
COLMAP (ERP)	3593	73.706	1.017	0.028	COLMAP (ERP)	1500	11.193	0.170	0.024
COLMAP (Pinhole)	3593	124.179	1.097	0.039	COLMAP (Pinhole)	1500	253.552	101.730	4.090
Spherical-SfM (ERP)	3593	54.137	3.858	4.076	Spherical-SfM (ERP)	1500	**4.128**	3.346	3.781
Ours	3594	**2.474**	**0.068**	0.022	**Ours**	1501	8.184	0.090	0.022
Mean									
Method	n	ATE↓	RPE-t↓	RPE-r↓					
Scal3R	4/4	187.407	2.911	1.025					
360DVO	4/4	68.313	0.183	**0.011**					
Open-vSLAM	4/4	281.793	41.690	37.574					
Stella-vSLAM	4/4	82.173	0.656	0.131					
COLMAP (ERP)	4/4	124.490	0.814	0.066					
COLMAP (Pinhole)	4/4	113.317	25.949	1.046					
Spherical-SfM (ERP)	4/4	21.186	3.312	3.989					
Ours	4/4	**3.527**	**0.052**	0.022					

Table 17: **PanoRecon localization by scene.** ATE and RPE-t are in meters; RPE-r is in degrees. n is the number of associated poses (scene panels) or available/expected scenes (Mean). All errors are RMSE; lower is better.

Building 1					**Building 2**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	765	14.643	0.281	0.182	Scal3R	419	**3.537**	0.223	0.128
360DVO	765	2.726	0.240	**0.023**	360DVO	419	4.548	**0.175**	0.047
Open-vSLAM	765	3.387	2.836	34.833	Open-vSLAM	418	5.764	3.636	0.135
Stella-vSLAM	765	2.506	0.411	0.646	Stella-vSLAM	418	5.018	1.024	0.121
COLMAP (ERP)	765	10.831	15.002	8.499	COLMAP (ERP)	419	5.369	0.202	0.048
COLMAP (Pinhole)	765	**1.618**	0.240	0.026	COLMAP (Pinhole)	419	5.431	0.205	**0.045**
Spherical-SfM (ERP)	765	11.907	15.817	8.852	Spherical-SfM (ERP)	419	30.653	25.932	0.059
Ours	764	1.620	**0.240**	0.024	**Ours**	418	5.302	0.201	0.052
Gym					**HPC**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	480	8.943	0.533	0.501	Scal3R	507	1.381	0.239	0.533
360DVO	480	1.876	0.037	0.023	360DVO	507	1.288	0.123	0.080
Open-vSLAM	480	17.622	13.582	56.856	Open-vSLAM	507	**0.673**	10.790	0.224
Stella-vSLAM	480	26.683	5.666	0.481	Stella-vSLAM	507	2.996	0.345	0.226
COLMAP (ERP)	480	0.629	0.013	0.020	COLMAP (ERP)	507	0.833	0.122	0.079
COLMAP (Pinhole)	480	**0.459**	**0.009**	**0.018**	COLMAP (Pinhole)	507	0.697	0.122	0.079
Spherical-SfM (ERP)	480	34.575	6.330	0.786	Spherical-SfM (ERP)	507	1.224	0.148	0.102
Ours	479	0.571	0.047	0.178	**Ours**	506	0.976	**0.106**	**0.064**
Residence					**Road**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	844	4.364	0.197	0.414	Scal3R	866	15.602	0.167	0.285
360DVO	844	1.583	0.022	0.030	360DVO	866	6.252	0.099	0.024
Open-vSLAM	719	1.384	4.831	37.360	Open-vSLAM	314	21.874	8.076	47.028
Stella-vSLAM	495	1.113	0.170	0.194	Stella-vSLAM	426	11.772	1.320	1.546
COLMAP (ERP)	–	–	–	–	COLMAP (ERP)	866	2.116	**0.081**	0.022
COLMAP (Pinhole)	844	0.193	**0.009**	**0.023**	COLMAP (Pinhole)	866	2.224	0.081	**0.020**
Spherical-SfM (ERP)	844	0.395	0.044	0.043	Spherical-SfM (ERP)	866	**2.006**	0.105	0.039
Ours	843	**0.115**	0.046	0.039	**Ours**	865	2.585	0.088	0.200
Residence2					**Mean**				
Method	n	ATE↓	RPE-t↓	RPE-r↓	Method	n	ATE↓	RPE-t↓	RPE-r↓
Scal3R	944	20.089	0.336	0.338	Scal3R	7/7	9.794	0.282	0.340
360DVO	944	1.323	0.033	0.043	360DVO	7/7	2.799	0.104	0.039
Open-vSLAM	944	1.086	2.308	0.260	Open-vSLAM	7/7	7.399	6.580	25.242
Stella-vSLAM	283	1.883	0.364	0.074	Stella-vSLAM	7/7	7.424	1.329	0.470
COLMAP (ERP)	944	1.284	0.031	0.042	COLMAP (ERP)	6/7	3.510 (6/7)	2.575 (6/7)	1.452 (6/7)
COLMAP (Pinhole)	944	1.528	**0.030**	**0.042**	COLMAP (Pinhole)	7/7	1.736	**0.099**	**0.036**
Spherical-SfM (ERP)	944	1.691	0.031	0.042	Spherical-SfM (ERP)	7/7	11.779	6.915	1.418
Ours	943	**0.708**	0.048	0.045	**Ours**	7/7	**1.697**	0.111	0.086

Table 18: **360LOC dense reconstruction by scene.** P/R/F in percent at 20 cm. Colors rank methods within each scene and metric.

Scene	Stella-vSLAM-Dense			COLMAP (Pinhole)			Ours		
	P↑	R↑	F↑	P↑	R↑	F↑	P↑	R↑	F↑
Atrium	47.83	**76.16**	58.76	**67.84**	46.63	55.27	65.51	66.92	**66.20**
Concourse	77.73	**59.92**	**67.67**	**85.15**	54.85	66.72	83.88	52.27	64.40
Hall	71.55	**49.30**	**58.38**	77.85	34.27	47.59	**85.95**	36.60	51.34
Piatrium	52.58	47.53	49.93	**94.22**	43.40	59.43	81.57	**55.71**	**66.20**

Table 19: **YUTO MMS dense reconstruction by scene.** P/R/F in percent at 20 cm. Colors rank methods within each scene and metric.

Scene	Stella-vSLAM-Dense			COLMAP (Pinhole)			Ours		
	P↑	R↑	F↑	P↑	R↑	F↑	P↑	R↑	F↑
Sequence A	46.13	39.25	42.41	88.09	49.94	63.74	**88.77**	**62.81**	**73.56**
Sequence B	18.74	4.19	6.85	10.13	0.81	1.50	**48.63**	**32.51**	**38.97**
Sequence C	13.42	7.10	9.29	9.85	1.36	2.39	**66.52**	**43.20**	**52.38**
Sequence D	16.84	7.51	10.39	8.51	0.81	1.48	**38.31**	**24.56**	**29.93**

Table 20: **PanoRecon dense reconstruction by scene.** P/R/F in percent at 20 cm. Colors rank methods within each scene and metric.

Scene	Stella-vSLAM-Dense			COLMAP (Pinhole)			Ours		
	P↑	R↑	F↑	P↑	R↑	F↑	P↑	R↑	F↑
Building 1	22.72	37.68	28.34	**86.74**	**71.51**	**78.39**	70.69	68.87	69.77
Building 2	49.98	55.64	52.66	**67.20**	**62.37**	**64.69**	64.06	58.56	61.19
Gym	34.40	10.87	16.52	58.60	41.62	48.67	**66.03**	**53.86**	**59.33**
HPC	51.50	30.84	38.58	**84.42**	55.84	**67.21**	65.74	**66.36**	66.05
Residence	50.17	52.52	51.32	**89.47**	56.71	69.42	76.00	**74.44**	**75.22**
Road	10.89	10.20	10.54	**48.96**	34.74	**40.65**	36.14	**40.14**	38.04
Residence2	36.58	19.95	25.82	63.86	40.04	49.22	**70.89**	**74.30**	**72.56**

E DENSE RECONSTRUCTION EVALUATION

Tables 18, 19, and 20 report per-scene dense reconstruction results on 360LOC, YUTO MMS, and PanoRecon, respectively. Additional point-cloud comparisons cover PanoRecon (Figs. 10, 11, and 12), 360LOC Concourse (Fig. 13), and YUTO MMS Sequences B and C (Figs. 14 and 15).