Establishing the Gold Standard for 360° Visual-Inertial Reconstruction *

Gilbert Tanner[1] Giorgos Evangelou[1] Julian Lechner[2] Zador Pataki[1]
Xudong Jiang[1] Paul-Edouard Sarlin[3] Shaohui Liu[1]

[1]ETH Zürich [2]AAU Klagenfurt & DLR [3]Google

Abstract

We describe our submission to the Hilti × Trimble SLAM Challenge 2026. Our main contribution is a 360° Structure-from-Motion (SfM) pipeline built on COLMAP [9] and GLOMAP [8] that turns each fisheye pair into twelve perspective virtual views, performs feature extraction, matching, and retrieval-based loop closure on the resulting rig, and integrates inertial information through an IMU-derived gravity prior in rotation averaging and a tightly-coupled visual–inertial (VI) bundle adjustment (BA) in the spirit of Mur-Artal/Forster [4, 6]. We eventually align the reconstructed point cloud to the building floorplan with 2D Iterative Closest Point (ICP) [2] to compensate for the residual scale and yaw error of the resulting trajectories.

1. Introduction

Visual–inertial odometry (VIO) systems, such as OpenVINS [5], delivers a strong real-time baseline on the Hilti × Trimble SLAM Challenge 2026 dataset, but accumulates orientation drift over long indoor traverses where global optimization becomes attractive. Starting from the VIO trajectory, we first created a COLMAP BA that jointly refines the baseline poses and triangulated 3D points on a panoramic rig (intrinsics and rig extrinsics frozen), to see whether we could push the OpenVINS scores higher at modest cost; we then built a full panorama SfM pipeline that performs global optimization from scratch on the same rig. The submitted trajectories are a mix of the two pipelines across the 25 sequences: time constraints kept some sequences on the lighter BA refinement, and on a subset of sequences that path is currently still scoring higher than the from-scratch SfM — a gap we are actively working to close; both are therefore described. A unified post-processing step then aligns the resulting trajectory and 3D points to the floorplan.

*Emails: {gtanne,gevangelou}@ethz.ch, jullechner@edu.aau.at, {zador.pataki,xudong.jiang,shaohui.liu}@inf.ethz.ch.

2. Baseline and BA refinement

The baseline VIO already achieves competitive scores on most sequences and provides a metric, gravity-aligned trajectory. Because re-running global SfM from scratch is expensive, we additionally provide a lighter alternative that only refines the baseline. The dual-fisheye images are remapped onto the same twelve virtual perspective views described in Section 3, masked, feature-extracted, and matched. Every rig pose is initialized from the baseline trajectory; 3D points are then triangulated from the known poses and matches and the whole reconstruction is jointly refined by BA with intrinsics, extrinsics, and the metric scale frozen. The result is a drift-corrected reconstruction that retains the gravity-aligned, metric character of the baseline at a small fraction of the cost of the full mapping pipeline.

3. Panorama SfM

Figure 1 summarises the panorama SfM pipeline described in the remainder of this section.

3.1. From dual fisheye to twelve virtual views

Standard learned features and matchers are trained on perspective images and degrade on the strong Kannala-Brandt distortions of the fisheye cameras. Following the panorama SfM idea distributed with COLMAP, we render four-yaw × three-pitch $(-35°, 0°, +35°)$ virtual perspective cameras per frame with $90°$ horizontal and vertical field of view (Fig. 2). For every virtual pixel we precompute a remap table that fuses the two physical fisheyes using a $1/\theta$ angle-of-incidence weighting, and a Voronoi mask assigns each pixel to the virtual camera whose optical axis is closest, so feature observations on overlap seams are not double-counted.The twelve virtual cameras are exposed to COLMAP as a rigid sensor rig sharing a common reference frame. A prompt-conditioned YOLOE [11] segmentation pass masks out the carrier (person, helmet, tablet) on the rendered images (Fig. 6) so the operator does not contaminate the reconstructed map.

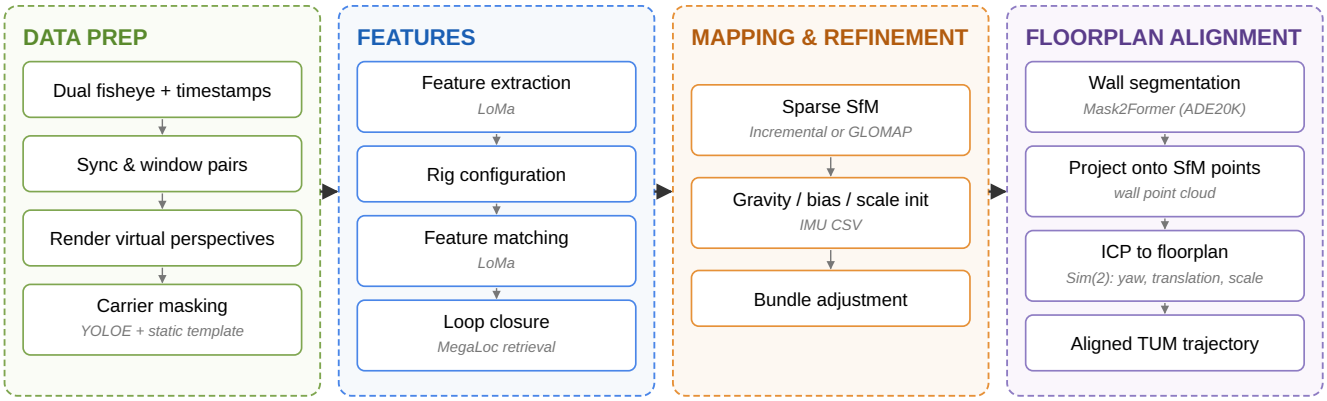

Figure 1. Overview of our panorama SfM pipeline. Dual-fisheye frames are remapped to twelve virtual perspective views and masked, LoMa features are extracted and matched on the rig with MegaLoc loop closures, mapping and VI refinement produce a metric reconstruction, and a final wall-segmentation + 2D ICP stage aligns it to the floorplan.

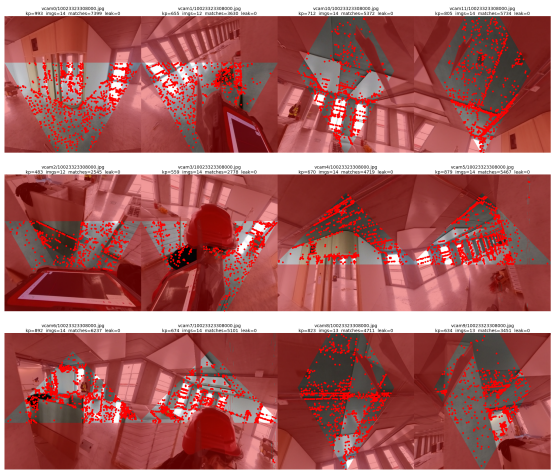

Figure 2. The twelve virtual perspective views rendered from one dual-fisheye frame, arranged as 4 yaws × 3 pitches. Translucent red overlays show the carrier mask; red dots are the extracted keypoints.

3.2. Feature extraction and matching

We use the end-to-end LoMa stack [7] with its transformer matcher. Its feature extractor and matcher are trained on substantially more matching data than prior work, which translates into more reliable matches on the dark indoor frames and the repetitive structures (tiled corridors, blank walls, near-identical doorways) that dominate construction-site sequences (Fig. 3). The rotation-invariant LoMa variant is preferred for the 360° rig. Image pairs are generated by sequential pairing through pycolmap with rig-aware verification, so two virtual cameras of the same instant are never matched against each other and per-pair geometric verification benefits from the known relative orientation inside the rig.

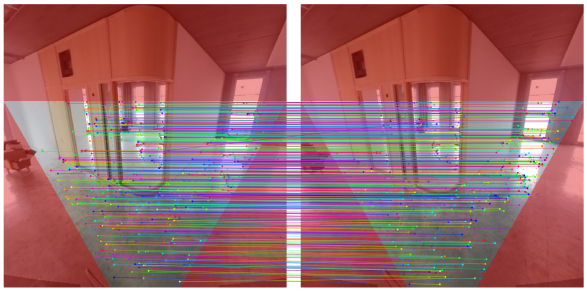

Figure 3. Dense LoMa correspondences between two virtual perspective views, recovered across repetitive flooring and blank walls that defeat classical matchers.

3.3. Loop closure

The vocabulary-tree loop detector built into the sequential matcher is brittle on construction-site sequences: many virtual cameras observing white walls or 90°-rotated corridors produce high-inlier matches that are not real loops. We replace it with a frame-level retrieval pipeline. For every rendered virtual camera image we compute a global descriptor with MegaLoc [1], then average the twelve per-virtual-camera descriptors of one timestamp into a single *frame descriptor*. Two 360° frames at the same physical location yield near-identical descriptors regardless of the carrier's heading, because each frame sees the full sphere (Fig. 4).

Top-k retrieval over the frame descriptors — with a minimum temporal gap and a cosine-similarity floor — yields candidate frame pairs which are expanded to image pairs by selecting the most similar virtual-camera combinations per frame pair (Fig. 5). The selected pairs are matched explicitly and tagged as loop-closure edges so the global mapper can route them through dedicated robust losses during rotation averaging, global positioning, and a final track-establishment pass.

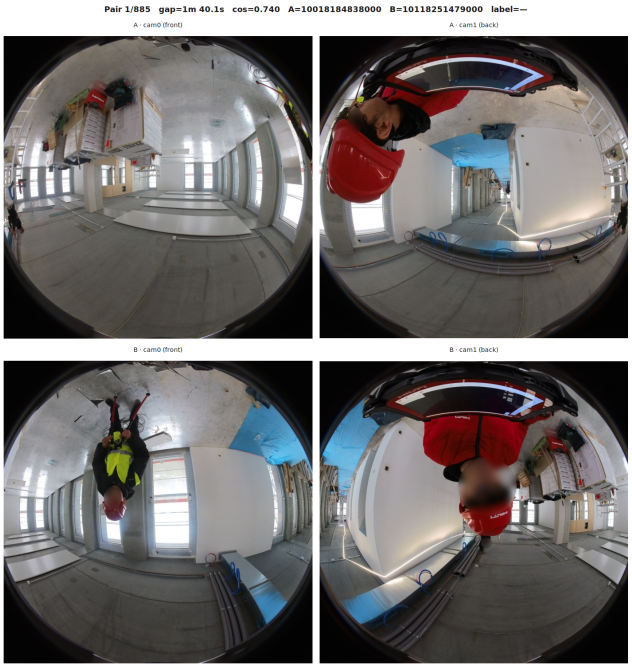

Figure 4. Example loop closure recovered by the MegaLoc retrieval pipeline: the same physical location revisited at the start (top row) and end (bottom row) of the sequence, with cam0 (front) and cam1 (back) shown side by side. Heading differs, but the per-frame descriptor matches.

3.4. Mapping

Our pipeline supports both incremental and global mapping approaches. The incremental pipeline uses COLMAP's classical image-by-image registration with fixed intrinsics and rig extrinsics and GPU BA. The global pipeline runs GLOMAP-style rotation averaging, global positioning, and joint BA [8], with rotation averaging consuming a per-image gravity prior recovered from the inertial measurement unit (IMU). The upstream global mapper only partially honours rig extrinsics — some BA passes silently re-optimise the per-camera sensor-from-rig transforms even when intrinsics are marked fixed — so we extended its rig support to enforce frozen extrinsics across every stage of the global pipeline, which is essential for our rendered virtual-camera rig where the relative orientations are known exactly by construction.

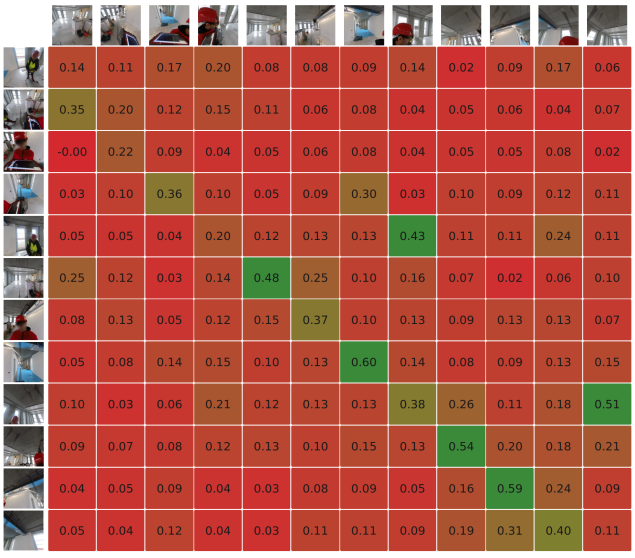

Figure 5. Per-virtual-camera cosine-similarity matrix between two frames flagged as a loop pair. Rows and columns index the twelve virtual cameras of each frame; green cells are the top-k image pairs retained for explicit matching.

3.5. IMU integration

Inertial information enters the pipeline at two levels. First, the global mapper consumes an IMU-derived per-image *gravity prior* during rotation averaging, which removes the gravity-direction ambiguity from the rotation graph. Second, we run a tightly-coupled VI refinement. A closed-form initialization following Mur-Artal and Tardós [6] first estimates the gyroscope bias from rotation pairs, then sets up a least-squares system over time-consecutive keyframes to recover the metric scale, gravity direction, and per-frame velocities, and finally refines gravity and adds the accelerometer bias under the constraint $\|g\| = 9.81\,\mathrm{m/s}^2$. The closed-form state warm-starts a Ceres-based VI BA that jointly optimizes camera poses, 3D points, per-keyframe velocities, and IMU biases under reprojection residuals and IMU preintegration factors [4] between consecutive keyframes; bias random-walk priors regularise the bias variables across the trajectory. This stage recovers metric scale, brings the reconstruction into a gravity-aligned frame, and significantly tightens the trajectory against the inertial measurements.

4. Floorplan alignment

Even after VI refinement, the reconstruction lives in its own metric world frame whose planar position and yaw need not agree with the floorplan-defined map frame. Roll and pitch are already recovered by the VI refinement (gravity is observable from the IMU) and the floor is itself horizontal in the map frame, so the remaining alignment problem is genuinely 2D. The metric scale recovered by VI typically agrees with the true scale to within $\sim$0.5 %, so the dominant residual is an in-plane rigid offset — a small global rotation about gravity and a lateral shift — with scale acting only as a small refinement. We nonetheless estimate all four parameters jointly to absorb whatever residual stretch remains, and write the transform as a Sim(2) (s, θ, t_x, t_y) between the horizontal projections of the reconstructed cloud and the floorplan walls.

We solve this transform with ICP because we have no

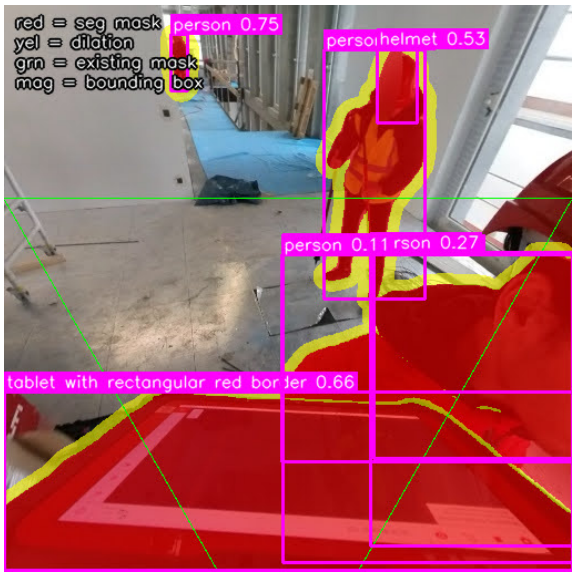

Figure 6. Carrier masking on a virtual perspective view. YOLOE detects the person, helmet, and tablet; the segmentation mask (red) is dilated (yellow) before being unioned with a static carrier template (green) to suppress feature observations on the operator.

point-to-point correspondences across the two representations: the floorplan is a binary wall mask of the building, while our reconstruction is an unstructured 3D point cloud whose individual points have no a priori identity in the floorplan. ICP alternates between (i) assigning each reconstructed point to its currently-nearest floorplan wall pixel and (ii) re-estimating the Sim(2) transform that minimises the distance between matched pairs, which lets the geometry of the cloud and the geometry of the floorplan walls drive the alignment without any explicit annotations.

A single ground-truth pose per sequence, provided by the challenge organizers, fixes the unknown initial rotation around gravity and the unknown lateral offset to within a few metres / degrees. Composing it with the corresponding reconstruction pose at the same timestamp yields a single-anchor rigid transform that brings the reconstruction into the floorplan map frame and serves as the initialisation for the iterative refinement below.

Wall extraction by semantic segmentation. Cluttered scenes, furniture, and floor/ceiling features make the raw point cloud a poor ICP source against a wall-only floorplan. We therefore reduce the cloud to its *wall* observations. For every frame we render an equirectangular projection of the dual-fisheye pair, pass it through a wall-segmentation network (Mask2Former [3] trained on the ADE20K [12] "wall" class), and back-project the resulting wall mask to the 3D points observed in that frame (Fig. 7). A 3D point is kept when a sufficient majority of its observing frames vote "wall". The resulting cloud is further restricted to a

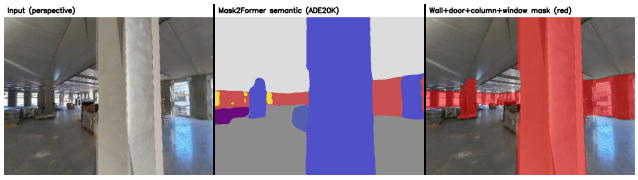

Figure 7. Wall extraction. Left: input perspective view. Middle: Mask2Former semantic prediction on ADE20K classes. Right: the wall/door/column/window classes are merged into the binary wall mask used to vote SfM points as wall or non-wall.

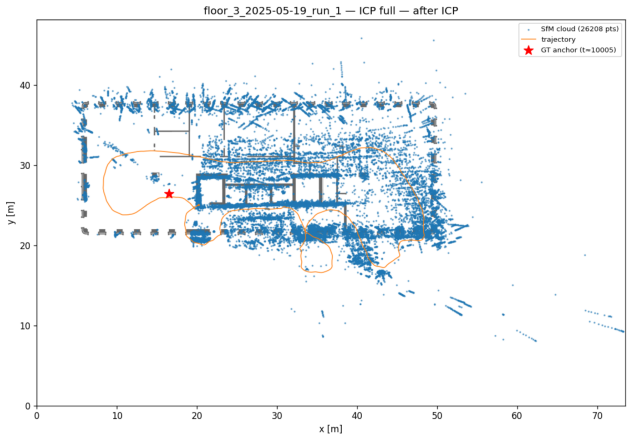

Figure 8. SfM point cloud (blue) and recovered trajectory (orange) overlaid on the floorplan walls (dark) after 2D ICP. The cloud tracks the floorplan walls across the full traverse, including the interior partitions.

wall-height band around the ground-truth anchor's height to remove the remaining floor and ceiling clutter.

2D ICP against the floorplan wall mask. The floorplan is a binary wall-mask image at a known metric scale. Wall pixels are loaded into a k-d tree and serve as the ICP target. Each iteration finds the nearest wall pixel for every transformed source point in the horizontal plane, drops correspondences beyond a fixed gating distance, downweights the rest with a Huber kernel, and solves a closed-form weighted similarity transform via Umeyama [10]. The aligned point cloud tracks the floorplan walls across the full traverse (Fig. 8), and the same transform brings the full reconstruction into the map frame (Fig. 9). The same refinement can also be performed for OpenVINS by extending OpenVINS to save its tracked 3D points.

5. Future work

Our VI refinement currently runs as a post-processing stage on top of an already-converged visual reconstruction. We prototyped a tighter integration that drives the incremental mapper in monotonic timestamp order, performs the closed-form initialization as soon as motion and heading diversity

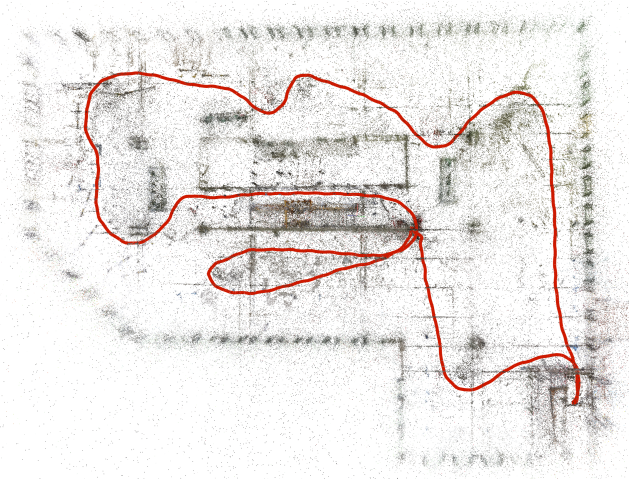

Figure 9. Final aligned reconstruction. The full SfM point cloud (gray) reveals the building structure, with the recovered trajectory (red) tracing the operator's path.

gates pass, and inserts VI BA inline during mapping; we did not have time to fully integrate this path into the submission. We expect it to be particularly valuable in the dark rooms that gave us the most trouble under the visual-only mapping plus post-hoc refinement scheme. Beyond that, exposing the floorplan as a plane-prior cost inside BA would let visual residuals participate in the alignment as well.

Acknowledgements

This project was completed as part of the ETH 3D Vision 2026 course (https://cvg.ethz.ch/lectures/3D-vision/).

References

[1] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. MegaLoc: One retrieval to place them all, 2025. arXiv:2502.17237. 2

[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 1

[3] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[4] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33 (1):1–21, 2017. 1, 3

[5] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A research platform for visual-inertial estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 1

[6] Raúl Mur-Artal and Juan D. Tardós. Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017. 1, 3

[7] David Nordström, Johan Edstedt, Georg Bökman, Jonathan Astermark, Anders Heyden, Viktor Larsson, Mårten Wadenbäck, Michael Felsberg, and Fredrik Kahl. Loma: Local feature matching revisited, 2026. 2

[8] Linfei Pan, Daniel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 1, 3

[9] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[10] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 4

[11] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOE: Real-time seeing anything. In *arXiv:2503.07465*, 2025. 1

[12] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 4