# $\sqrt{\text{VINS}}$-Based 360 Visual-Inertial SLAM Systems: Hilti–Trimble SLAM Challenge 2026 Entry

Jungwoo Lee[1], Hogyun Kim[1], Jiwon Choi[1], Jinyong Jeong[2], and Younggun Cho[1†]
[1]Dept. of Electrical and Computer Engineering, Inha University, Incheon, South Korea
[2]Riibotics, Siheung, South Korea
[pihsdneirf, hg.kim, jiwon2]@inha.edu, jinyong.jeong@riibotics.io, yg.cho@inha.ac.kr

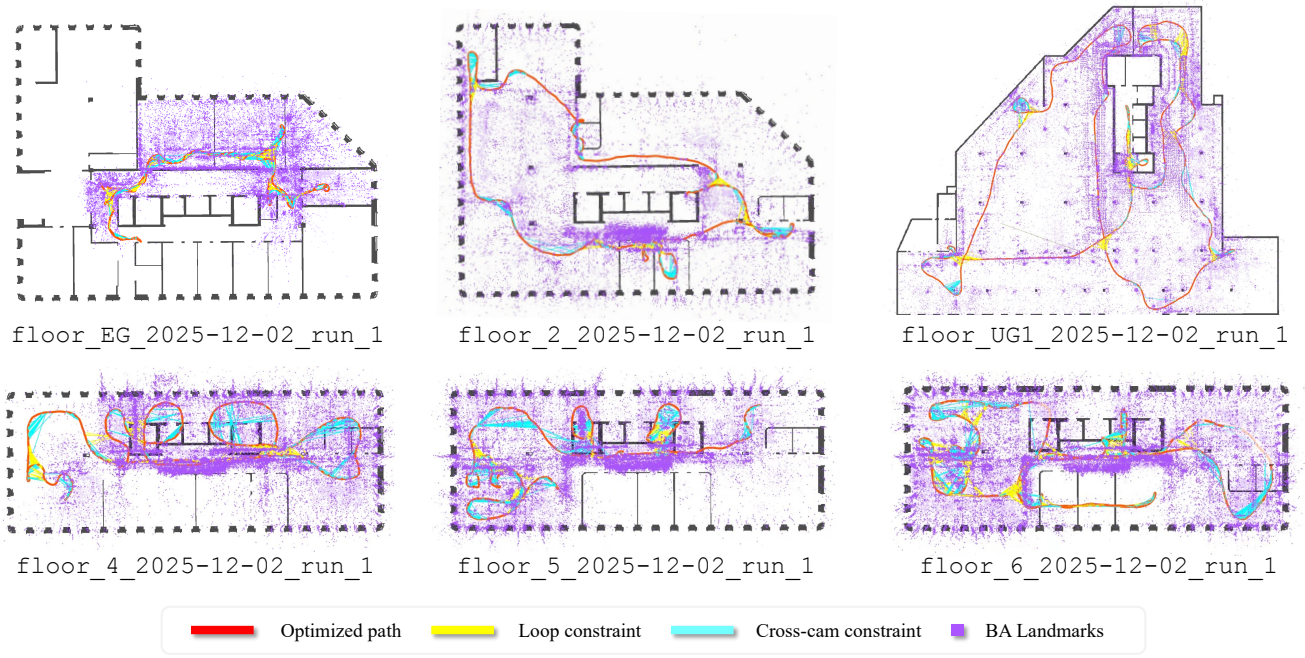


Fig. 1: Qualitative results on several Hilti-Trimble indoor sequences [1]. Our simultaneous localization and mapping (SLAM) algorithm produces bundle adjustment (BA)-optimized landmarks that align well with the floorplan map. Note that the alignment between the estimated map and the floorplan was performed manually for visualization.

***Abstract*— This report describes our submission to the 2026 Hilti-Trimble SLAM Challenge. The system is a 360-camera visual-inertial pipeline that uses only the provided dual-fisheye image streams and IMU measurements. First, raw images are preprocessed with low-light enhancement [2] and static/dynamic masking [3], and then fed into a $\sqrt{\text{VINS}}$ [4] frontend as the local odometry, with deep learning-based SuperPoint features [5], ELSED line segments [6], and LightGlue matching [7]. The resulting trajectory is refined using a GTSAM-based backend that performs tightly-coupled factor graph optimization with SALAD global descriptor [8] and visual bundle adjustment (BA) [9]. The backend performs local-to-global structureless BA, tightly regularized intrinsic and extrinsic refinement, and global scale adjustment supported by the known inter-camera baseline and cross-view constraints. The design targets long indoor sequences with dark regions, repetitive corridors, dynamic objects, and limited geometric parallax, while keeping the pipeline reproducible in the provided ROS2 workspace. Fig. 1 demonstrates that our proposed method aligns well with the floorplan across several indoor sequences.**

## I. Algorithm Overview

Our visual-inertial SLAM system is summarized in Fig. 2. We model the 360 camera as two calibrated fisheye views under the extended and unified camera model (EUCM) projection model and represent all poses in the `cam0` frame (Sec. I-A). First, raw fisheye frames are conditioned by static masking, low-light enhancement, and dynamic-object suppression before any feature extraction (Sec. I-B). Next, we pass the conditioned images to a $\sqrt{\text{VINS}}$-based frontend [4] that estimates locally metric visual-inertial odometry using SuperPoint point features [5], LightGlue matching [7], and ELSED line segments [6] (Sec. I-C). We then detect and geometrically verify loop closure and short-term cross-camera constraints using SALAD global descriptor [8] before inserting them into the factor graph (Sec. I-D). Finally, a GTSAM-based backend [10] then jointly optimizes `cam0` poses, extrinsic and intrinsic parameters, and a global scale variable via EUCM-compatible structureless smart projection factors [9] (Sec. I-E). Since loop closure and BA can revise earlier poses using later observations, the exported trajectory is non-causal.

### A. Sensor Model and Input Conditioning

First, we model the 360 camera as two calibrated fisheye views using the EUCM [11]. The optimized trajectory is represented in the `cam0` frame, where each graph node $\mathbf{T}_k \in \mathrm{SE}(3)$ denotes the `cam0` pose at frame $k$. The `cam0` observations are projected directly from $\mathbf{T}_k$, while `cam1`

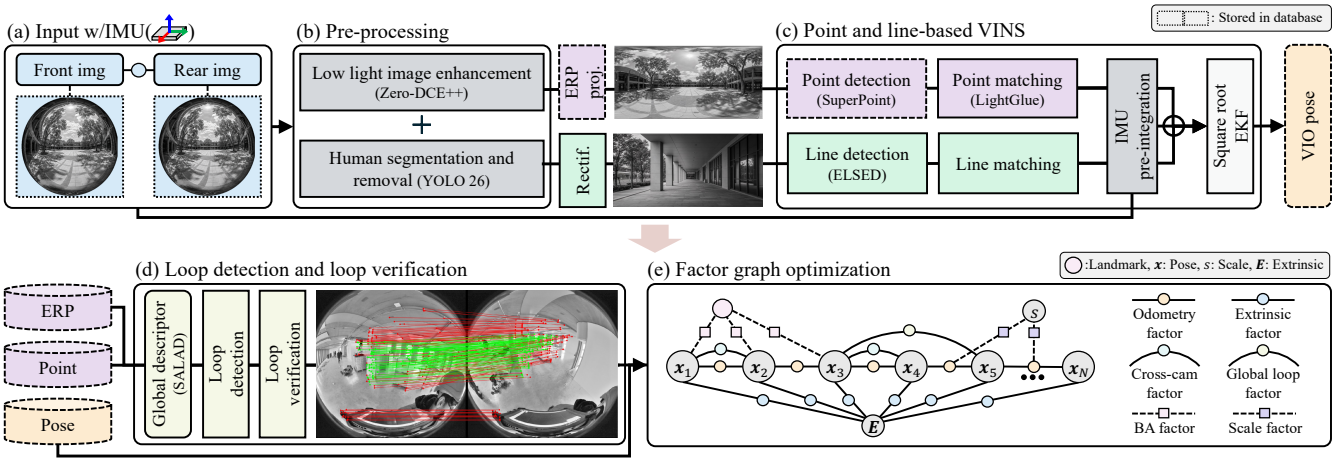


Fig. 2: Overview of the proposed 360-camera visual-inertial SLAM pipeline. Raw dual-fisheye frames are first conditioned by low-light enhancement [2] and dynamic masking [3], then tracked by a $\sqrt{\text{VINS}}$-based frontend [4] with SuperPoint-LightGlue point features [5, 7] and ELSED line segments [6]. Detected loop closures [8] and short-term cross-camera matches are geometrically verified and inserted as smart projection factors [9] into a GTSAM backend [10] that jointly optimizes poses, calibration, and a global scale variable.

observations are projected through a shared rigid transform $\mathbf{E}_{01} = \mathbf{T}_{\text{cam0}\to\text{cam1}}$, allowing both fisheye views to constrain one `cam0` trajectory without duplicate per-frame camera nodes.

### B. Pre-processing

Next, before feature extraction, we apply three conditioning steps to each raw fisheye frame. Static masks remove invalid fisheye regions and fixed image artifacts. Low-light enhancement [2] then improves contrast in dark indoor regions. Finally, YOLO26 detector/segmentation outputs [3] conservatively suppress features on potentially moving objects.

### C. Point- and line-based Visual-Inertial Odometry

We then build the frontend on $\sqrt{\text{VINS}}$ [4]. Point tracking uses SuperPoint [5] and LightGlue [7] on the conditioned fisheye images, with state-predicted matching to improve robustness during fast motion and repetitive indoor appearance. To complement point features in low-texture indoor regions, we add a line-feature branch that detects line segments with ELSED [6] and maintains their temporal associations via line matching. These line observations are incorporated through an MSCKF-style update inspired by PL-VIWO [12]. Rather than enforcing a full Manhattan-world model, we adopt a vertical-world, or weak Manhattan-world, assumption: gravity provides a reliable vertical reference, while horizontal directions remain unconstrained.

### D. Loop Closure and Cross-View Constraints

Following odometry estimation, loop closure uses global image descriptors [8] to propose visually similar keyframes while rejecting immediate temporal neighbors, followed by SuperPoint-LightGlue geometric verification over valid `cam0`/`cam1` view pairings. We add accepted loop candidates as EUCM-compatible smart projection factors, keeping loop closure consistent with the keyframe BA formulation. Beyond long-range loop closure, we search a bounded temporal window for short-term cross-camera matches that connect nearby non-consecutive observations across the two fisheye views. These constraints are particularly useful during turns, corridor motion, and near-revisits, where structure observed in one fisheye view quickly reappears in the other. Because the verified matches span the calibrated inter-camera baseline, they also provide local metric evidence that supports the graph-wide scale correction.

### E. Factor Graph Optimization

Finally, the backend optimizes a factor graph over `cam0` poses, camera intrinsic and extrinsic parameters, and a global scale variable. Consecutive frontend estimates provide relative pose factors, while all visual constraints are represented as EUCM-compatible structureless smart projection factors [9] instantiated for keyframe BA, geometrically verified loop closure, and short-term cross-camera tracks. The backend objective is summarized as follows:

$$\min_{\{\mathbf{T}_k\},\mathbf{E}_{01},\boldsymbol{\theta},s} \sum_{(i,j)\in\mathcal{O}} \left\| r^{\text{odom}}_{i,j}(\mathbf{T}_i,\mathbf{T}_j,s) \right\|^2_{\Sigma_o} \tag{1a}$$

$$+ \sum_{m\in\mathcal{B}} \left\| r^{\text{ba}}_{m}(\{\mathbf{T}\},\mathbf{E}_{01},\boldsymbol{\theta}) \right\|^2_{\Sigma_b} \tag{1b}$$

$$+ \sum_{m\in\mathcal{L}} \left\| r^{\text{loop}}_{m}(\{\mathbf{T}\},\mathbf{E}_{01},\boldsymbol{\theta}) \right\|^2_{\Sigma_l} \tag{1c}$$

$$+ \sum_{m\in\mathcal{C}} \left\| r^{\text{cross}}_{m}(\{\mathbf{T}\},\mathbf{E}_{01},\boldsymbol{\theta}) \right\|^2_{\Sigma_c} \tag{1d}$$

$$+ \left\| r^{\text{ext}}(\mathbf{E}_{01}) \right\|^2_{\Sigma_e} + \left\| r^{\text{int}}(\boldsymbol{\theta}) \right\|^2_{\Sigma_\theta} + \left\| r^{\text{scale}}(s) \right\|^2_{\Sigma_s}, \tag{1e}$$

where $\mathcal{O}$ is the set of consecutive odometry edges, $\mathcal{B}$ is the set of keyframe BA tracks, $\mathcal{L}$ is the set of accepted loop tracks, and $\mathcal{C}$ is the set of short-term cross-camera tracks. The residual $r^{\text{odom}}$ is a relative $\text{SE}(3)$ constraint whose translation is modulated by the shared scale variable $s$, while $r^{\text{ba}}$, $r^{\text{loop}}$, and $r^{\text{cross}}$ are all EUCM-compatible smart projection residuals. We tightly regularize the shared extrinsic, intrinsic, and scale variables so that they absorb small residual calibration or metric-scale errors without accommodating arbitrary trajectory drift.

TABLE I: Transcription table matching the official evaluator report. ✗ indicates that the sequence is excluded from evaluation because it is used as the ground truth trajectory. Note that indicates hidden plots.

| Sequence | RMSE [m] | Mean [m] | Max [m] | Min [m] | Coverage [%] | Poses | Exp. Score |
|---|---|---|---|---|---|---|---|
| floor_1_2025-05-05_run_1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| floor_1_2025-07-07_run_1 | 0.1465 | 0.1264 | 0.4078 | 0.0066 | 100.0000 | 3959 | 94.4015 |
| floor_1_2025-12-02_run_1 | 0.1060 | 0.0969 | 0.2190 | 0.0100 | 100.0000 | 8172 | 95.6537 |
| floor_2_2025-05-05_run_1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| floor_2_2025-10-28_run_1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| floor_2_2025-10-28_run_2 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| floor_2_2025-12-02_run_1 | 0.1354 | 0.1267 | 0.5778 | 0.0109 | 99.9304 | 4306 | 94.3535 |
| floor_2_2025-12-03_run_1 | 0.1883 | 0.1534 | 0.6852 | 0.0097 | 99.9321 | 4413 | 93.2931 |
| floor_3_2025-05-19_run_1 | 0.0907 | 0.0807 | 0.1867 | 0.0063 | 100.0000 | 3383 | 96.3713 |
| floor_3_2025-12-02_run_1 | 0.0866 | 0.0778 | 0.2001 | 0.0042 | 99.9229 | 3890 | 96.4937 |
| floor_4_2025-05-19_run_1 | 0.0783 | 0.0732 | 0.1377 | 0.0118 | 100.0000 | 2616 | 96.6919 |
| floor_4_2025-12-02_run_1 | 0.0918 | 0.0816 | 0.2329 | 0.0042 | 97.0816 | 5522 | 96.3304 |
| floor_5_2025-12-02_run_1 | 0.0837 | 0.0731 | 0.2366 | 0.0065 | 99.9366 | 4728 | 96.7088 |
| floor_6_2025-06-18_run_1 | 0.0588 | 0.0474 | 0.3076 | 0.0097 | 100.0000 | 2029 | 97.8519 |
| floor_6_2025-07-07_run_1 | 0.0693 | 0.0629 | 0.1266 | 0.0096 | 100.0000 | 2104 | 97.1545 |
| floor_6_2025-12-02_run_1 | 0.0883 | 0.0757 | 0.2253 | 0.0052 | 100.0000 | 5041 | 96.5962 |
| floor_6_2025-12-02_run_2 | 0.0882 | 0.0741 | 0.2662 | 0.0026 | 99.9180 | 3654 | 96.6706 |
| floor_7_2025-12-02_run_1 | 0.0889 | 0.0647 | 0.3414 | 0.0015 | 99.9138 | 3476 | 97.1012 |
| floor_7_2025-12-02_run_2 | 0.0749 | 0.0687 | 0.1619 | 0.0027 | 100.0000 | 4539 | 96.8927 |
| floor_7_2025-12-03_run_1 | 0.1134 | 0.0992 | 0.3443 | 0.0078 | 100.0000 | 5877 | 95.5658 |
| floor_EG_2025-10-16_run_1 | 0.0987 | 0.0854 | 0.2563 | 0.0012 | 100.0000 | 7170 | 96.1669 |
| floor_EG_2025-12-02_run_1 | 0.0770 | 0.0611 | 0.3792 | 0.0038 | 99.9191 | 3707 | 97.2465 |
| floor_EG_2025-12-02_run_2 | 0.0667 | 0.0601 | 0.1976 | 0.0038 | 99.8538 | 4781 | 97.2796 |
| floor_UG1_2025-05-19_run_1 | 0.1318 | 0.1159 | 0.2886 | 0.0197 | 100.0000 | 6010 | 94.8434 |
| floor_UG1_2025-06-18_run_1 | 0.1897 | 0.1653 | 0.4832 | 0.0153 | 100.0000 | 4889 | 92.7524 |
| floor_UG1_2025-10-16_run_1 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| floor_UG1_2025-12-02_run_1 | 0.1616 | 0.1457 | 0.3578 | 0.0242 | 99.9603 | 7548 | 93.5574 |
| floor_UG1_2025-12-02_run_2 | 0.1643 | 0.1396 | 0.4202 | 0.0041 | 99.9846 | 6501 | 93.8487 |
| floor_UG1_2025-12-03_run_1 | 0.1093 | 0.1024 | 0.2307 | 0.0101 | 99.8490 | 3967 | 95.4095 |
| floor_UG2_2025-12-02_run_1 | 0.1367 | 0.1200 | 0.3073 | 0.0137 | 100.0000 | 6607 | 94.6656 |
| **Mean Results** | **0.1090** | **0.0951** | **0.3031** | **0.0082** | **99.8481** | | |
| **Final Score** | | | | | | | **2393.9009** |

## II. Experiments

Table I shows that our method achieves a mean position error below 0.1 m across the evaluated sequences.

## References


[1] S. Centanni, Y. Zhang, Y. Tao, J. Kindle, F. Neuhaus *et al.* (2026) Hilti-Trimble-Oxford dataset: 360 visual-inertial benchmark with floor plan priors for slam and localization. [Online]. Available: https://github.com/Hilti-Research/hilti-trimble-slam-challenge-2026

[2] C. Li, C. G. Guo, and C. C. Loy, "Learning to enhance low-light image via zero-reference deep curve estimation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[3] G. Jocher and J. Qiu, "Ultralytics yolo26," 2026. [Online]. Available: https://github.com/ultralytics/ultralytics

[4] Y. Peng, C. Chen, K. Wu, and G. Huang, "sqrt-vins: Robust and ultrafast square-root filter-based 3d motion tracking," *IEEE Transactions on Robotics (TRO)*, oct 2025. [Online]. Available: https://ieeexplore.ieee.org/document/11219345

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[6] I. Suárez, J. M. Buenaposada, and L. Baumela, "Elsed: Enhanced line segment drawing," *Pattern Recognition*, vol. 127, p. 108619, 2022.

[7] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 17 627–17 638.

[8] S. Izquierdo and J. Civera, "Optimal transport aggregation for visual place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[9] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4290–4297.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard *et al.*, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[11] B. Khomutenko, G. Garcia, and P. Martinet, "An enhanced unified camera model," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 137–144, 2015.

[12] Z. Zhang, W. Bai, L. Zhao, and P. Ladosz, "Pl-viwo: A lightweight and robust point-line monocular visual inertial wheel odometry," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 15 205–15 212.