ACDC-VSLAM: Adaptive Constraints for Dual-Fisheye-Camera-Based Visual-Inertial SLAM With Point and Line Features (HILTI×TRIMBLE SLAM Challenge 2026)

Jinwoo Jeon[1*], Dong-Uk Seo[1*], Junwan Choi[1*], Seunghun Lee[1*], *Student Member, IEEE*, Jihun Nam[1*], Hyungtae Lim[2*], *Member, IEEE*, and Hyun Myung[1*], *Senior Member, IEEE*

***Abstract*— We present *ACDC-VSLAM*, short for *Adaptive Constraints for Dual-Fisheye-Camera-Based Visual-Inertial SLAM,* developed as our submission to the HILTI×TRIMBLE SLAM Challenge 2026. The challenge dataset poses three coupled difficulties: (i) wide field-of-view (FOV) fisheye distortion that biases pixel-space feature selection, (ii) textureless indoor scenes that yield short and unreliable point tracks, and (iii) a non-overlapping front-rear camera configuration that prevents inter-camera triangulation and biases revisits toward the cross-camera regime. To address these difficulties, ACDC-VSLAM combines (i) a geometry-guided point-and-line frontend with spherical equal-area bucketing for balanced angular coverage, (ii) a local bundle adjustment module that reweights visual reprojection residuals by observation frequency and depth while incorporating stable line constraints, and (iii) a two-stage loop-closure module that recovers intra- and cross-camera revisits while adaptively selecting the correction manifold among $\mathrm{SE}(3)$, $\mathrm{Sim}(3)$, and 4-DoF for each candidate. Together, these components yield consistently strong scores on the official HILTI×TRIMBLE SLAM Challenge 2026 benchmark.**

I. Overview of the HILTI×TRIMBLE SLAM Challenge 2026

The HILTI×TRIMBLE SLAM Challenge 2026 dataset is recorded with a custom handheld rig and presents three distinctive properties that strongly shape the design space of any SLAM system targeting this benchmark.

(P1) Wide-FOV fisheye cameras. The rig is equipped with wide-FOV fisheye cameras whose angular pixel density is highly non-uniform. Standard pixel-space feature selection therefore over-samples the image center, under-represents the distorted periphery, and yields unreliable depth and parallax for peripheral observations.

(P2) Textureless indoor scenes. Some sequences are captured in industrial and architectural indoor environments dominated by white walls, repetitive structures, and uniform floors. In such regions, point-only frontends produce short and unstable tracks, weakening the visual constraints supplied to the optimizer.

(P3) Non-overlapping front-rear camera configuration. Unlike standard stereo rigs whose cameras are configured to look in the same direction with overlapping views, the two fisheye cameras face opposite directions with negligible FOV overlap. This makes conventional inter-camera triangulation infeasible and biases revisits toward the *cross-camera* regime, in which the rear camera at the current time must be matched against a front-camera view from an earlier traversal of the same place, or vice versa.

We design *ACDC-VSLAM*, which is an abbreviation for Adaptive Constraints for Dual-Fisheye-Camera-Based Visual-Inertial SLAM, around these three properties. The frontend addresses wide-FOV fisheye distortion (P1) and textureless indoor scenes (P2) through spherical feature selection and line features; the local bundle adjustment further suppresses unreliable peripheral, low-parallax, and short-track constraints while incorporating line constraints; adaptive rear-camera management suppresses weak constraints caused by the non-overlapping front-rear camera configuration (P3); and the backend handles cross-camera revisits under this non-overlapping setting through two-stage loop detection and adaptive manifold switching.

II. ACDC-VSLAM Approach

ACDC-VSLAM is an optimization-based dual-fisheye visual-inertial SLAM system built on [1] and organized into three main modules: a fisheye-aware frontend that provides balanced angular feature coverage and line-feature support (Sec. II.A), a local bundle adjustment that reweights residuals to suppress unreliable observations (Sec. II.B), and a backend that performs intra- and cross-camera loop closure and adapts active constraints according to camera geometry and observation reliability (Sec. II.C). Each module is designed to address one or more of the challenges introduced in Sec. I.

A. Frontend

The frontend addresses the non-uniform pixel density of fisheye images (P1) by selecting features on the unit sphere rather than in the distorted image plane, and addresses the lack of texture (P2) by augmenting point tracks with spherical line features.

1) Spherical Equal-Area Feature Bucketing (addressing (P1)): Unlike pinhole images, fisheye images compress peripheral viewing directions: near the image boundary, a wide range of rays is mapped to a relatively small image region. This non-uniform projection leaves the image center

*All authors equally contributed to this project.

[1]The School of Electrical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Daejeon, 34141, Republic of Korea, {zinuok, dongukseo, joseph3155, shleee, namjh1228, hmyung}@kaist.ac.kr

[2]KAIST InnoCORE LLM, KAIST, Daejeon, 34141, Republic of Korea, shapelim@kaist.ac.kr

with denser pixel sampling per unit angular area. Consequently, uniform keypoint selection in pixel space tends to over-sample the image center while under-representing peripheral regions in angular coverage.

To obtain a more balanced angular distribution, we first back-project each keypoint to a unit bearing vector (x, y, z) in the camera frame. This bearing is represented using spherical coordinates (θ, φ), where the polar angle $\theta = \arccos(z)$ and the azimuthal angle $\varphi = \operatorname{atan2}(y, x)$. These coordinates are then mapped to a 2D equal-area domain $(\tilde{u}, \tilde{v})$ where equal-sized regions correspond to equal angular areas on the unit sphere. Specifically, we perform quadtree-based bucketing in this domain and retain the keypoint with the highest response within each cell. Because the bucketing is performed after the equal-area transformation, each cell corresponds to a comparable angular region rather than a comparable pixel-space region.

2) Line Feature Detection and Tracking (addressing (P2)): As noted in (P2), the HILTI sequences include textureless indoor scenes where point tracks are often short and noisy. To improve robustness, we use line segments as additional structural cues, especially in weakly textured regions such as walls and corridors. For wide-FOV fisheye images, we adopt the spherical line representation [2], where each segment is formed by lifting its endpoint bearings onto the unit sphere and using the normal of its associated great circle. Line segments are detected using ELSED [3], filtered by angular length, and tracked using descriptor matching with geometric gating based on direction, spherical normal alignment, and length consistency. Only temporally stable and geodesically consistent tracks are used as line constraints, reducing false matches from repetitive indoor structures common in (P2).

3) Floor-Aware Landmark Rejection: In addition, we apply gravity-aligned height filtering to suppress geometrically inconsistent landmarks. After gravity initialization, landmark heights are evaluated in the gravity-aligned world frame, and landmarks whose estimated heights fall far outside a plausible vertical range are treated as unreliable and rejected during matching and optimization.

B. Local Bundle Adjustment

Building on the frontend, the local bundle adjustment further suppresses unreliable constraints originating from (P1) and (P2): peripheral fisheye observations carry larger reprojection noise, and texture-poor scenes produce many short and low-parallax tracks. We address these issues in two ways: by reweighting point reprojection residuals according to observation reliability (Sec. II.B.1) and by adding spherical line residuals as additional geometric constraints (Sec. II.B.2).

1) Reweighting for Visual Reprojection Residuals: To suppress noisy short tracks and distant low-parallax constraints, we reweight each point reprojection residual according to the landmark track length and its camera-frame depth. This encourages the optimization to rely more on stable, informative observations while reducing the influence of noisy or weak constraints.

Landmark track-length reweighting. In SLAM, the number of observations associated with a landmark corresponds to the length of its feature track, i.e., the number of frames in which the same feature is repeatedly observed from different viewpoints. Landmarks with longer tracks are repeatedly verified across frames and usually provide more stable pose constraints than short-lived landmarks. We therefore increase the relative contribution of long-track landmarks while reducing the dominance of numerous short tracks that often appear in texture-poor scenes (P2).

Inverse-depth reweighting. Inverse-depth weighting reflects the fact that points with smaller camera-frame depth are generally more reliable, i.e., closer points induce stronger parallax and better-conditioned reprojection constraints. Accordingly, we preserve nearby observations and gradually attenuate the contribution of distant ones, whose constraints are weakened by low parallax and peripheral distortion (P1).

2) Line Feature Optimization: To complement reweighted point residuals in texture-poor scenes (P2), we use stable line tracks as additional constraints in local bundle adjustment.

For each tracked line segment, a candidate 3D line is initialized from the intersection of two corresponding interpretation planes. The candidate is then filtered by geometric checks, such as baseline, depth, and line length, so that only well-conditioned line constraints are introduced into the optimizer.

During local bundle adjustment, each accepted line track is parameterized by two 3D endpoint landmarks and refined jointly with the camera poses and point landmarks. Line residuals are robustly weighted according to track age and line reprojection consistency, allowing reliable long-lived lines to stabilize the optimization while suppressing poorly fitted line tracks.

3) Adaptive Rear-Camera Constraint Selection: Under the non-overlapping front-rear configuration (P3), we empirically found that using rear-camera observations is not always beneficial during forward-dominant motion. Because the rear camera mainly observes previously traversed regions, its observations can become redundant and often form shorter-lived or more weakly conditioned constraints than those from the front camera, which observes the direction of upcoming motion.

We therefore selectively deactivate rear-camera observations at keyframe boundaries when the estimated rig motion is predominantly aligned with the front-camera viewing direction and the front camera provides sufficient reliable features. The rear camera is reactivated as soon as either condition fails, i.e., when rotational motion makes the rear view useful again or when the front camera alone provides too few reliable features in texture-poor regions. This deactivation is temporary and does not discard rear-camera images from the system. It only prevents rear-camera observations from being used for local matching, triangulation, and bundle adjustment, while feature detection and per-camera BoW registration continue. Thus, rear-camera frames can later be recovered for loop closure through on-demand temporal triangulation with nearby active keyframes. This temporal

triangulation uses nearby keyframes and is distinct from inter-camera triangulation between the non-overlapping front and rear cameras.

C. Backend Optimization

The backend addresses the non-overlapping front-rear configuration (P3) on two fronts. First, loop candidates in this configuration fall into three geometric cases: Case A is a rigid same-direction revisit, Case B is a scale-drifted cross-camera revisit, and Case C is a degenerate match in a texture-poor region. We separate these cases because they provide different levels of geometric support and therefore should not be constrained by a single fixed correction model: a rigid model cannot absorb accumulated scale drift, while a high-DoF model can overfit weak or poorly conditioned correspondences. We therefore adaptively select the correction manifold per candidate (Sec. II.C.1). Second, because conventional same-camera retrieval systematically misses cross-camera revisits in the non-overlapping setup, we detect loops in two stages over a per-camera BoW database (Sec. II.C.2).

1) Adaptive Manifold-Switching Loop Closure: We operationalize the three cases above by selecting the loop-closure manifold from diagnostics computed on the geometrically verified inlier set. For Case A, when the candidate shows no significant scale drift and the inlier consensus supports a stable 6-DoF estimate, we use a rigid $\mathrm{SE}(3)$ correction. For Case B, when the scale ratio estimated from 3D-3D landmark correspondences, together with the loop path length, reliably indicates accumulated scale drift, we use a $\mathrm{Sim}(3)$ correction. For Case C, when the correspondence quality is insufficient for a stable 6-DoF estimate but still supports reduced visual correction, we leverage the IMU-stabilized roll and pitch and fall back to a 4-DoF (yaw and translation) correction. If even this reduced consensus is insufficient or the resulting correction exceeds a consistency budget, the loop is rejected to preserve trajectory integrity.

2) Two-Stage Dual Loop Detection: In the non-overlapping front-rear configuration (P3), the dominant loop signal is often *cross-camera*: with rapid rotational motion, the rear camera at the current time tends to revisit what the front camera saw earlier, or vice versa, so same-camera-only retrieval systematically misses these front-rear revisits. We address this with a two-stage loop detector built on a per-camera bag-of-words (BoW) database. The first stage performs *intra-camera retrieval*, matching each current camera against its own past descriptors to recover conventional same-direction revisits at low cost. When the first stage finds no reliable candidate, the second stage performs *inter-camera retrieval*, querying each current camera against the opposite camera's database to recover the cross-camera revisits unique to the dual-fisheye configuration. In short, the first stage handles the common case efficiently, while the second stage captures the geometry-specific revisits that a single-stage same-camera search would miss.

III. Experimental Results

As presented in Table I, ACDC-VSLAM scores above 95 on most floors, while the underground `floor_UG1` sequences remain the hardest cases because long featureless corridors and limited revisits stress both the frontend and loop closure.

TABLE I. Per-sequence scores of ACDC-VSLAM on the HILTI×TRIMBLE SLAM Challenge 2026 benchmark, as reported by the official evaluation server.

Scene	Sequence	Score
`floor_1`	2025-07-07_run_1	95.90
	2025-12-02_run_1	96.03
`floor_2`	2025-12-02_run_1	96.95
	2025-12-03_run_1	95.55
`floor_3`	2025-05-19_run_1	95.21
	2025-12-02_run_1	97.36
`floor_4`	2025-05-19_run_1	96.66
	2025-12-02_run_1	97.10
`floor_5`	2025-12-02_run_1	97.71
`floor_6`	2025-06-18_run_1	97.41
	2025-07-07_run_1	97.68
	2025-12-02_run_1	97.56
	2025-12-02_run_2	97.72
`floor_7`	2025-12-02_run_1	97.49
	2025-12-02_run_2	97.63
	2025-12-03_run_1	95.85
`floor_EG`	2025-10-16_run_1	96.84
	2025-12-02_run_1	97.73
	2025-12-02_run_2	97.20
`floor_UG1`	2025-05-19_run_1	93.88
	2025-06-18_run_1	92.30
	2025-12-02_run_1	95.43
	2025-12-02_run_2	94.55
	2025-12-03_run_1	97.47
`floor_UG2`	2025-12-02_run_1	94.83
Total		**2410.00**

IV. Conclusion

ACDC-VSLAM addresses fisheye distortion (P1), textureless scenes (P2), and non-overlapping front-rear cameras (P3) by combining a spherical point-and-line frontend, robust local bundle adjustment, and adaptive cross-camera loop closure, achieving high official scores on the benchmark.

References

[1] S. Boche, J. Jung, S. B. Laina, and S. Leutenegger, "OKVIS2-X: Open keyframe-based visual-inertial SLAM configurable with dense depth or LiDAR, and GNSS," *IEEE Transactions on Robotics*, 2025.

[2] Z. Wang, K. Yang, H. Shi, Y. Zhang, Z. Xu, F. Gao, and K. Wang, "LF-PGVIO: A visual-inertial-odometry framework for large field-of-view cameras using points and geodesic segments," *IEEE Trans. Intell. Veh.*, 2024.

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