Two-Stage Visual-Inertial SLAM with Sliding-Window Filtering and Global Bundle Adjustment for the Hilti × Trimble SLAM Challenge 2026

Arda C. Demirtaş*[1], Berkay Şekeroglu*[1], and Atakan Topaloglu*[1]

***Abstract*—This report summarizes our submission to the Hilti × Trimble SLAM Challenge 2026, covering both the SLAM and Localization tasks. We developed a two-stage visual-inertial SLAM pipeline: first, a causal multi-camera MSCKF-based estimator produces an initial trajectory and feature tracks; second, an offline visual-inertial bundle adjustment refines all trajectories using the full sequence. During development, the main bottleneck was not only drift, but reliable initialization and sustained feature tracking in dynamic-start, low-texture, and low-light sequences. Our SLAM submission relies on dynamic initialization, reduced playback rate for stable tracking, and per-sequence selection among a small set of robustness-oriented filter configurations. For the Localization task, we register the SLAM trajectories to the floorplan coordinate frame using a single cam0 anchor pose provided per sequence, applying an SE(2) transform derived from the IMU-to-camera extrinsic calibration and the ground-truth initialisation pose.**

***Index Terms*—Visual-inertial odometry, SLAM, MSCKF, bundle adjustment, IMU preintegration, OpenVINS, COLMAP, floorplan localization, map alignment, Hilti SLAM Challenge.**

I. Introduction

THIS report describes our approach for the Hilti × Trimble SLAM Challenge 2026 trajectory-estimation task. Since the SLAM task evaluates trajectories in any preferred reference frame, we focused on producing geometrically consistent trajectories with high temporal coverage and low drift.

The dataset exposed two dominant issues. First, several sequences were difficult to initialize because they began with motion, low visual texture, or rapid viewpoint changes. Second, even when initialization succeeded, feature-poor regions caused drift that a causal sliding-window estimator could not fully recover.

Our final pipeline addresses these issues with a two-stage design. We use OpenVINS [1] as the front-end estimator because it provides robust MSCKF-based visual-inertial filtering and exposes the feature tracks required for later refinement. We then apply an offline visual-inertial bundle adjustment stage to all trajectories to reduce residual drift, using the COLMAP visual-inertial fork [2].

*All authors contributed equally to this submission.

[1]The authors are with the Department of Information Technology and Electrical Engineering (D-ITET), ETH Zürich, 8092 Zürich, Switzerland. {ademirtas, bsekeroglu, atopaloglu}@ethz.ch

Manuscript submitted to the Hilti × Trimble SLAM Challenge 2026.

The final system consists of five steps:

1) Run an MSCKF visual-inertial filter with KLT tracking across both fisheye cameras and IMU propagation at 1000 Hz.
2) Generate multiple candidate trajectories using robustness-oriented filter configurations.
3) Select the best candidate trajectory for each sequence.
4) Reconstruct feature tracks from the filter output.
5) Refine all trajectories using offline visual-inertial bundle adjustment.

II. Filter Initialization and Configuration

A. Initialization Failure Modes

Initialization was the primary failure mode across the 30 sequences. The static initializer requires feature tracks to survive for at least half the initialization window in the older half of the sliding window before attempting state recovery. Fast motion or low visual texture breaks KLT tracks before they age sufficiently, blocking initialization indefinitely.

A representative case is `floor_EG/2025-12-02/run_2`. The static initializer waits for a still-to-moving jerk event that does not occur until approximately 59 s into the recording, when the operator adjusts their grip. Switching to dynamic initialization with a 1.0 s window allowed the filter to trigger on the early camera-shake motion, reducing the initialization delay to 5.6 s.

B. Dynamic Initialization

Dynamic initialization triggers on generally excited angular velocity rather than requiring a static-to-moving transition. This was the single most effective intervention, converting several delayed or failed runs into complete trajectories. It is particularly important for sequences that begin while the operator is already walking.

C. Initialization Window Duration

A shorter initialization window (1.0 s versus the 3.0 s default) reduces the track-survival requirement at a minor cost to inertial alignment quality. On dynamic-start and low-texture sequences this trade-off was clearly beneficial.

D. Reduced Playback Rate

At full playback speed, the CPU could not consistently process every image frame at the nominal 30 Hz camera rate, reaching only 16–23 frames per second. This forced the optical-flow tracker to operate on larger apparent inter-frame displacements, shortening feature tracks and degrading both initialization and sustained estimation. Processing at 0.5× playback speed restored near-nominal tracking performance and increased pose output density from approximately 16–23 Hz to approximately 29 Hz.

E. Robustness-Oriented Filter Settings

We designed five filter variants around the baseline configuration. The useful changes were: increasing the number of tracked features and persistent SLAM landmarks; extending the clone horizon; applying CLAHE contrast normalization; inflating IMU process noise; and enabling zero-velocity updates (ZUPT).

Inflating the IMU process noise covariance down-weights the inertial contribution to the EKF update, allowing visual measurements to dominate when inertial biases are poorly observable (for example, slow walking in low-contrast basement corridors). The trade-off is increased susceptibility to IMU integration drift under fast motion.

Increasing tracked features to 800 and SLAM landmarks to 100 widens the reprojection constraint set. Extending the clone horizon to 15 allows the filter to accumulate longer-baseline observations, improving triangulation depth in texture-sparse regions.

ZUPT constrains velocity and rotation rate to zero during detected stationary intervals, strongly anchoring bias estimation. In practice it had limited impact because most sequences involve continuous walking with few clear stationary periods.

III. Per-Sequence Selection

No single filter configuration was best for all sequences. We generated multiple candidate trajectories per run and selected the final one using three criteria.

For the five released ground-truth sequences, selection was based on ATE after SE(3) alignment. For the remaining sequences, we used trajectory plausibility and multi-configuration consensus. Specifically, we checked whether the horizontal extent was consistent with the expected floor geometry and whether vertical drift remained below approximately 2 m (elevated vertical drift reliably indicates unconstrained IMU bias divergence in monocular mode). When three or more independent configurations converged on the same trajectory shape, that shape was accepted as correct.

Table I summarizes the final selected configuration types.

Dynamic initialization at 0.5× playback accounts for 77% of the final submission.

TABLE I
Final Submission Configuration Types

Configuration type	Runs
Dynamic init, 1.0 s window, 0.5× playback	13
Dynamic init, 3.0 s window, 0.5× playback	10
Baseline, full playback speed	3
Inflated IMU process noise, 0.5× playback	2
High-feature, extended clone horizon, 0.5× playback	2

TABLE II
Effect of Offline VI-BA (averaged over 5 GT sequences)

Method	ATE RMSE (m)
Filter baseline (mean, 5 GT runs)	0.327
After VI-BA (mean, 5 GT runs)	**0.262**
Mean score improvement	**+2.4 pts**

TABLE III
Per-Run Results on the Five Ground-Truth Sequences

Sequence	ATE RMSE (m)		Score	
	Filter	VI-BA	Filter	VI-BA
floor_1 / 2025-05-05 / run_1	0.300	**0.233**	85.7	**88.3**
floor_2 / 2025-05-05 / run_1	0.446	**0.390**	81.4	**83.3**
floor_2 / 2025-10-28 / run_1	0.248	**0.184**	89.7	**92.5**
floor_2 / 2025-10-28 / run_2	0.262	**0.192**	89.5	**91.9**
floor_UG1 / 2025-10-16 / run_1	0.378	**0.311**	82.5	**85.1**
Mean	0.327	**0.262**	85.8	**88.2**

IV. Offline Visual-Inertial Bundle Adjustment

We applied an offline visual-inertial bundle adjustment to all runs to refine the filter output. The optimization is seeded directly from the filter: the feature publisher outputs, at each camera frame, the 3D position, pixel coordinates, and stable track ID of every active KLT-tracked feature. Accumulating these messages reconstructs the full track history without additional feature extraction or descriptor matching. A typical run yields approximately 20k–50k triangulated 3D points and 800k 2D reprojection observations.

The bundle adjustment minimizes the joint cost

$$\min_{\mathbf{T},\mathbf{p},\mathbf{b}} \sum_{(i,j)} \rho\left(\left\|\mathbf{r}^{ij}_{\text{reproj}}\right\|^2_{\Sigma_c}\right) + \sum_{k} \left\|\mathbf{r}^{k}_{\text{IMU}}\right\|^2_{\Sigma_{\text{IMU}}}, \qquad (1)$$

where $\mathbf{T}$ are camera poses, $\mathbf{p}$ are landmark positions, $\mathbf{b}$ denotes inertial states and biases, $\mathbf{r}^{ij}_{\text{reproj}}$ are per-observation reprojection residuals, $\mathbf{r}^{k}_{\text{IMU}}$ are IMU preintegration residuals computed using the on-manifold analytical formulation of Forster et al. [3], and $\rho(\cdot)$ is a Huber loss. The optimization variables are refined jointly without fixing the filter poses.

The principal limitation is that KLT tracks carry no feature descriptors. When a track is lost due to occlusion or fast motion, the physical point re-appears with a new track ID, breaking the observation chain. This weakens the global reprojection term, particularly in environments with heavy occlusion or repetitive structure.

Table II summarises the aggregate VI-BA improvement over the filter baseline on the five ground-truth sequences.

V. Results

Table III reports per-run performance on the five released ground-truth sequences after SE(3) alignment.

VI. Localization Task

Each SLAM trajectory is expressed in an arbitrary global frame; we register it to the floorplan map using the single ground-truth cam0 anchor pose $\mathbf{T}^{\text{map}}_{\text{cam}_0,\text{gt}}$ provided per sequence.

SLAM poses are output in the IMU frame, so the anchor SLAM pose is first converted to cam0 using the Kalibr extrinsic $\mathbf{T}^{\text{imu}}_{\text{cam}_0}$:

$$\mathbf{T}^{\text{global}}_{\text{cam}_0,k} = \mathbf{T}^{\text{global}}_{\text{imu},k}\,\mathbf{T}^{\text{imu}}_{\text{cam}_0}. \tag{2}$$

The global-to-map transform follows as

$$\mathbf{T}^{\text{map}}_{\text{global}} = \mathbf{T}^{\text{map}}_{\text{cam}_0,\text{gt}}\left(\mathbf{T}^{\text{global}}_{\text{cam}_0,k}\right)^{-1}. \tag{3}$$

Because the SLAM estimator maintains a gravity-aligned frame, the residual pitch and roll of $\mathbf{T}^{\text{map}}_{\text{global}}$ are noise. We zero them, retaining only the yaw ψ and translation $\mathbf{t}$, and apply the resulting planar transform to every pose in the trajectory:

$$\mathbf{T}^{\text{map}}_{\text{cam}_0,i} = \begin{bmatrix}\mathbf{R}_z(\psi) & \mathbf{t}\\ \mathbf{0}^\top & 1\end{bmatrix}\mathbf{T}^{\text{global}}_{\text{imu},i}\,\mathbf{T}^{\text{imu}}_{\text{cam}_0}. \tag{4}$$

For sequences where the filter initializes after the anchor epoch, we decouple the rotation and translation estimates: the yaw is derived from the first filter-initialized pose, while the translation is set so that the anchor-epoch SLAM position maps exactly to the ground-truth map position.

VII. Discussion

Robust initialization mattered more than back-end complexity. Dynamic initialization was the most impactful change, converting delayed or failed runs into complete trajectories. The reduced playback rate was equally important in practice: processing frames faster than the optical-flow tracker can reliably handle degrades track quality and ultimately causes filter divergence, even though the underlying data is identical.

The offline bundle-adjustment stage confirmed that full-trajectory optimization recovers drift that is irrecoverable in a sliding-window filter. The main limitation is the reliance on KLT tracks without descriptors. Replacing the KLT front-end with descriptor-based local features and re-identification across track breaks (for example, ALIKED with LightGlue matching) would yield longer, more constraining observation chains and likely improve the back-end substantially, particularly in the occluded basement environments.

VIII. Conclusion

We presented a two-stage visual-inertial SLAM pipeline combining an MSCKF sliding-window filter with offline global visual-inertial bundle adjustment. The most effective practical decisions were dynamic initialization, reduced playback speed for stable feature tracking, and per-sequence selection among a small number of targeted filter variants. For the Localization task, SLAM trajectories are registered to the floorplan frame via a gravity-consistent planar transform derived from the IMU-to-camera extrinsic calibration and a single ground-truth anchor pose per sequence.

Acknowledgment

The authors thank their supervisors Zador Pataki (zador.pataki@inf.ethz.ch), Xudong Jiang (xudong.jiang@inf.ethz.ch), and Shaohui Liu (shaohui.liu@inf.ethz.ch) of the Department of Computer Science, ETH Zürich, and Paul-Edouard Sarlin (psarlin@google.com) of Google, for their guidance throughout this project. We also thank the organizers of the Hilti × Trimble SLAM Challenge 2026.

References

[1] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom. Workshop on Open Source Systems in Robotics*, 2020. https://github.com/rpng/open_vins

[2] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[3] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.