Realtime tracking with Monado's Basalt on 360 FoV fisheye cameras

Mattis Krauch[*,1] Mateo de Mayo[*,1,2] Jonathan Eichhorn[1,2] Tomás Santucci[3]
Dominik Muhle[1,2] Taihú Pire[3,4] Daniel Cremers[1,2]

I. Introduction

Wea present a modified version of Monado's Basalt VI-SLAM system [1] [2], commonly targeted towards XR devices like AR glasses and VR headsets. We pick it due to its strong performance and efficient cpu-only design, which allows us to implement heavier methods on top. The resulting system remains performant and fully real-time,while producing good trajectories on the available sequences.

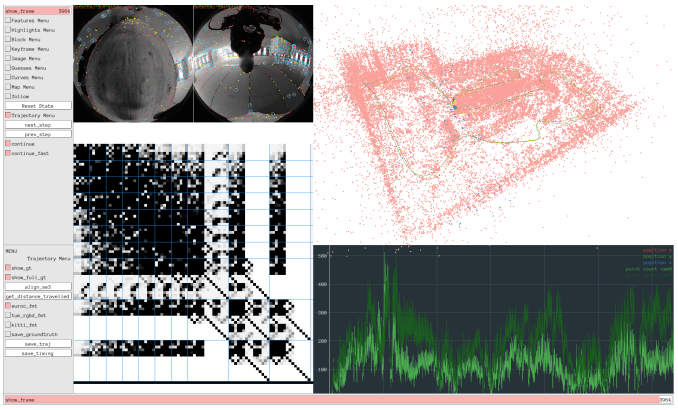

Figure 1. User interface for development of the system

II. Pipeline

Our system consist of a visual-inertial odometry based on Basalt, with extensive improvements from Monado's fork [1] and additional modifications to improve robustness on the challenge sequences.

We operate on raw images from the fisheye cameras without undistortion, which preserves all pixel information and avoids additional preprocessing cost. Our frontend uses a patch-based pyramidal Lucas–Kanade optical flow tracker to propagate features between frames. Outlier rejection methods like IMU-PARSAC [3] are used to improve robustness against spurious measurements (e.g., moving operator, flickering lights).

A loop closure thread runs in parallel to the main VIO pipeline, which detects loop closures and performs pose graph optimization to reduce drift using Ceres. The loop closure thread operates asynchronously and does not introduce latency to the main VIO pipeline, which remains real-time.

The VIO backend performs nonlinear optimization using reprojection terms, IMU factors, and marginalization priors. We use Basalt's native implementation with improvements from Monado's fork.

Table I
ATE and RTE results on the five challenge sequences.

Sequence	ATE (m)	RTE (m)
HF01D0505R1	0.733	0.008634
HF02D0505R1	0.178	0.304322
HF02D1028R1	0.162	0.459010
HF02D1028R2	0.296	0.462755
HFU1D1016R1	1.250	0.008710
Mean	0.524	0.248686

III. Results

We take advantage of Basalt's efficient design and add let loose with heavier settings, while keeping the system real-time, averaging ~25ms per frame as can be seen in Table II. The system runs on an older generation i5 13600 CPU, limited to 4 threads for each run. Figure 2 shows the per-frame estimation time across on sequence (`Floor 1 05-05 run 1`) sequences, averaging averages 28.62ms. The memory footprint is more than doubled than original basalt, but still stits at just above 1GB. We combine runs from two settings (one with more aggressive outlier rejection and one with more conservative settings) to achieve the best possible results on the sequences, which is reflected in the reported mean values.

We evaluate the method on the five publicly available sequences from the challenge benchmark and show results in Table I. Absolute trajectory error (ATE) is reported as the primary global metric. In addition, we also report relative trajectory error (RTE with delta=6) to capture local consistency and short-horizon drift, which is often omitted in SLAM benchmark reporting but is useful for understanding estimator quality beyond global alignment.

IV. Conclusion

The Hilti-Tremble SLAM Challenge 2026 provides a valuable real-world benchmark for visual-inertial SLAM

[1]Computer Vision Group - Technical University of Munich
[2]Munich Center for Machine Learning
[3]Universidad Nacional de Rosario
[4]CONICET-UNR CIFASIS

Table II
FRAME ESTIMATION TIME AND PEAK MEMORY USAGE ARE SIGNIFICANTLY BIGGER THAN ORIGINAL BASALT.

Sequence	Frame Time (ms)	Peak Memory (MB)
HF01D0505R1	25.54 ± 18.66	1109.1
HF02D0505R1	25.12 ± 18.21	1108.6
HF02D1028R1	26.95 ± 24.08	1108.8
HF02D1028R2	23.22 ± 15.98	1108.9
HFU1D1016R1	23.77 ± 17.20	1108.8
Mean	24.92 ± 18.82	1108.8

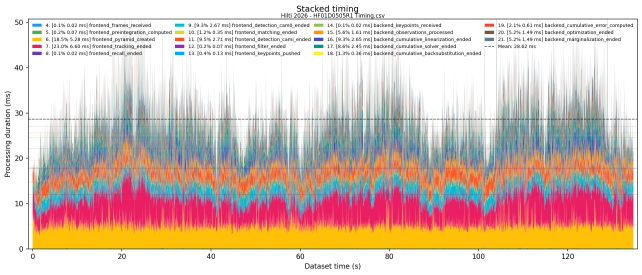

Figure 2. Figure 3

systems and a very nice sensor setup that is relevant for many practical applications. Leveraging an XR-focused system based on Basalt demonstrates good performance on the available sequences. By leveraging raw image tracking, robust outlier rejection, and efficient optimization, we achieve competitive accuracy while maintaining real-time performance on a CPU-only setup.

[1] M. de Mayo, D. Cremers, and T. Pire, "The Monado SLAM Dataset for Egocentric Visual-Inertial Tracking," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 13111–13118.

[2] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-Inertial Mapping with Non-Linear Factor Recovery," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 422–429, Apr. 2020.

[3] J. Li, X. Pan, G. Huang, Z. Zhang, N. Wang, H. Bao, and G. Zhang, "RD-VIO: Robust Visual-Inertial Odometry for Mobile Augmented Reality in Dynamic Environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 10, pp. 6941–6955, Oct. 2024.