Visual-Inertial SLAM: Incremental EKF-Based Localization and Mapping from Stereo and IMU

Leo
ECE 276A — Sensing & Estimation in Robotics
University of California, San Diego

I. Introduction

Simultaneous localization and mapping (SLAM) is the problem of jointly estimating a robot's trajectory and a map of its environment from onboard sensor data, without relying on external infrastructure such as GPS. This capability is essential for autonomous navigation in GPS-denied environments—indoor spaces, underground tunnels, or dense urban canyons—where no global position reference is available.

This report presents an incremental, EKF-based visual-inertial SLAM pipeline that fuses body-frame linear and angular velocity measurements from an inertial measurement unit (IMU) with stereo camera feature observations. The pipeline is developed in four progressive stages:

1) **IMU-only prediction** (Task 1): propagate SE(3) poses using the IMU motion model with covariance tracking.
2) **Feature tracking** (Task 2, extra credit): detect, match, and temporally track stereo features on dataset 02 using optical flow.
3) **Landmark mapping** (Task 3): estimate 3D landmark positions via an EKF update step while keeping poses fixed at the IMU predictions.
4) **Visual-inertial SLAM** (Task 4): jointly correct both the trajectory and the landmark map using an information-form pose update fused with landmark EKF updates.

Each stage builds on the previous one, and the report describes the baseline implementation of each component followed by the key improvements that were necessary to achieve stable, accurate results.

II. Problem Formulation

A. State Representation

The robot pose at time t is represented as a rigid-body transformation $T_t \in \mathrm{SE}(3)$:

$$T_t = \begin{bmatrix} R_t & p_t \\ 0 & 1 \end{bmatrix}, \quad R_t \in SO(3),\ p_t \in \mathbb{R}^3, \tag{1}$$

where T_t maps points from the IMU body frame into the world frame. Each landmark is a 3D position $\mathbf{m}_j \in \mathbb{R}^3$ in world coordinates, for $j = 1, \ldots, M$.

B. Motion Model

Given IMU measurements of body-frame linear velocity $v_t \in \mathbb{R}^3$ and angular velocity $\omega_t \in \mathbb{R}^3$, the discrete-time motion model is:

$$T_{t+1} = T_t \exp\big(\hat{\xi}_t\, \tau\big), \tag{2}$$

where $\tau = t_{k+1} - t_k$ is the time step, $\xi_t = [v_t^\top \ \omega_t^\top]^\top \in \mathbb{R}^6$ is the body-frame twist, and $\exp(\cdot)$ denotes the matrix exponential on $\mathrm{se}(3)$.

C. Observation Model

A landmark $\mathbf{m}_j$ observed by a calibrated stereo rig produces a 4-dimensional measurement:

$$z_{t,j} = \pi(\mathbf{m}_j,\, T_t) + n_{t,j}, \quad n_{t,j} \sim \mathcal{N}(0, R), \tag{3}$$

where $z_{t,j} = [u_l,\, v_l,\, u_r,\, v_r]^\top$ contains the left and right pixel coordinates, $R \in \mathbb{R}^{4\times 4}$ is the measurement noise covariance, and π is the stereo projection function. The projection proceeds through three stages:

$$\mathbf{p}^{\text{imu}} = T_t^{-1}\, \tilde{\mathbf{m}}_j, \tag{4}$$

$$\mathbf{p}^{\text{cam}} = {}^{C}T_I\ \mathbf{p}^{\text{imu}}, \tag{5}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u\, x_c/z_c + c_u \\ f_v\, y_c/z_c + c_v \end{bmatrix}, \tag{6}$$

where $\tilde{\mathbf{m}}_j$ is the homogeneous world point, ${}^{C}T_I$ is the extrinsic transform from IMU to camera frame, and $K = \left[\begin{smallmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{smallmatrix}\right]$ is the camera intrinsic matrix.

D. Inputs and Desired Outputs

Inputs: IMU velocities (v_t, ω_t) for $t = 1, \ldots, N$; stereo feature observations $z_{t,j}$ (with -1 for unobserved entries); calibration matrices K_l, K_r, ${}^{C_L}T_I$, ${}^{C_R}T_I$.

Outputs: Estimated trajectory $\{T_t\}_{t=1}^{N}$ and landmark map $\{\mathbf{m}_j\}_{j=1}^{M}$.

III. Technical Approach

A. Extended Kalman Filter

All estimation tasks in this project are solved using the Extended Kalman Filter (EKF). The EKF approximates a nonlinear Bayes filter by linearizing the motion and observation models around the current estimate via first-order Taylor expansion.

Given a prior $\mathbf{x}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}(\boldsymbol{\mu}_{t|t},\, \Sigma_{t|t})$, a nonlinear motion model $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$ with process noise $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, W)$, and a nonlinear observation model $\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t)$ with measurement noise $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, V)$, the EKF proceeds as follows.

Linearization. Define the Jacobians at the current estimate:

$$F_t := \frac{df}{d\mathbf{x}}\Big|_{(\boldsymbol{\mu}_{t|t},\, \mathbf{u}_t,\, \mathbf{0})}, \quad Q_t := \frac{df}{d\mathbf{w}}\Big|_{(\boldsymbol{\mu}_{t|t},\, \mathbf{u}_t,\, \mathbf{0})}, \tag{7}$$

$$H_{t+1} := \frac{dh}{d\mathbf{x}}\Big|_{(\boldsymbol{\mu}_{t+1|t},\, \mathbf{0})}, \quad R_{t+1} := \frac{dh}{d\mathbf{v}}\Big|_{(\boldsymbol{\mu}_{t+1|t},\, \mathbf{0})}. \tag{8}$$

Prediction step:

$$\boldsymbol{\mu}_{t+1|t} = f(\boldsymbol{\mu}_{t|t},\, \mathbf{u}_t,\, \mathbf{0}), \tag{9}$$

$$\Sigma_{t+1|t} = F_t\, \Sigma_{t|t}\, F_t^\top + Q_t\, W\, Q_t^\top. \tag{10}$$

Update step:

$$K_{t+1} = \Sigma_{t+1|t}\, H_{t+1}^\top \big(H_{t+1}\, \Sigma_{t+1|t}\, H_{t+1}^\top + R_{t+1}\, V\, R_{t+1}^\top\big)^{-1}, \tag{11}$$

$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} + K_{t+1}\big(\mathbf{z}_{t+1} - h(\boldsymbol{\mu}_{t+1|t},\, \mathbf{0})\big), \tag{12}$$

$$\Sigma_{t+1|t+1} = (I - K_{t+1}\, H_{t+1})\, \Sigma_{t+1|t}. \tag{13}$$

The following subsections describe how this general EKF framework is instantiated for each task: the choice of state, the specific forms of f, h, and their Jacobians, and the practical considerations that arise when the state lives on $\mathrm{SE}(3)$ rather than in $\mathbb{R}^n$.

B. IMU EKF Prediction (Task 1)

The pose is propagated via the exponential map on $\mathrm{SE}(3)$ (Eq. 2). For near-zero angular velocity ($\|\omega_t \tau\| < 10^{-12}$), we fall back to a pure translation update to avoid numerical issues in the matrix exponential.

The 6×6 pose covariance Σ_t is propagated using the first-order linearization:

$$\Sigma_{t+1} = F_t\, \Sigma_t\, F_t^\top + Q_t, \tag{14}$$

where $F_t = I_6 - \mathrm{ad}(\xi_t\, \tau)$ is the adjoint-based state transition matrix, $\mathrm{ad}(\cdot)$ is the adjoint representation of the Lie algebra $\mathrm{se}(3)$, and $Q_t = \mathrm{diag}(\sigma_v^2,\, \sigma_v^2,\, \sigma_v^2,\, \sigma_w^2,\, \sigma_w^2,\, \sigma_w^2)\, \tau^2$.

C. Stereo Observation Model

Separate camera projections. Rather than using a single stereo projection matrix, we project each landmark separately through the left and right cameras using their respective extrinsics ${}^{C_L}T_I$ and ${}^{C_R}T_I$ and intrinsics K_l, K_r. This design handles the case where the stereo baseline is not perfectly horizontal, which would violate the assumptions of a single rectified stereo model.

Bearing triangulation with quality gates. New landmarks are initialized via bearing measurement triangulation: pixel coordinates are back-projected into ray directions using each camera's intrinsics and extrinsics, and the 3D point is estimated as the midpoint of the shortest segment between the two rays (closed-form least squares). To reject poor initializations, three quality gates are applied:

1) *Disparity check*: $|u_l - u_r| \geq d_{\min}$ (default 2.0 px) rejects points at near-infinity depth.
2) *Depth bounds*: the camera-frame depth z_c must lie in $[0.5, 80]$ m for both cameras.
3) *Reprojection error*: the initial reprojection error must be below 5.0 px.

Analytic Jacobians. Two Jacobians are derived analytically:

- $\partial z/\partial \mathbf{m}$ (4×3): the observation Jacobian with respect to the landmark position, used in the landmark EKF update. Derived via the quotient rule applied to the pinhole projection through each camera's 3×4 projection matrix $P = K\,[{}^{C}T_I\, T_t^{-1}]_{3\times 4}$.
- $\partial z/\partial \xi$ (4×6): the observation Jacobian with respect to the pose perturbation, used in the SLAM pose update. Derived via the chain rule:

$$\frac{\partial z}{\partial \xi} = \frac{\partial z}{\partial \mathbf{p}^{\text{cam}}} \cdot \frac{\partial \mathbf{p}^{\text{cam}}}{\partial \xi}, \tag{15}$$

where $\frac{\partial \mathbf{p}^{\text{cam}}}{\partial \xi} = -R_{CI}\,[I_3 \mid -[\mathbf{p}^{\text{imu}}]_\times]$ uses the right perturbation convention on $\mathrm{SE}(3)$.

D. Landmark Mapping EKF (Task 3)

Each landmark maintains an independent 3×3 covariance matrix, avoiding the $\mathcal{O}(M^2)$ cost of a joint covariance. The update follows the standard EKF form:

$$K_j = P_j H_j^\top (H_j P_j H_j^\top + R)^{-1}, \tag{16}$$

$$\mathbf{m}_j \leftarrow \mathbf{m}_j + K_j(z - \hat{z}), \tag{17}$$

$$P_j \leftarrow (I - K_j H_j)\, P_j. \tag{18}$$

Depth-bounds checking is applied after each update to reject landmarks that move outside the valid range $[0.5, 80]$ m in camera-frame depth.

Problem: covariance collapse. In the baseline implementation (zero landmark process noise), the covariance P_j shrinks rapidly after the first few updates. Since the IMU-predicted poses carry unmodeled drift, subsequent observations produce large innovations relative to the overly tight P_j. The Kalman gain becomes negligibly small and the filter effectively stops updating, producing a sparse map with only ∼5,000 effective updates across the full sequence despite hundreds of thousands of observations.

Improvement: landmark process noise. To address covariance collapse, a small additive process noise $Q_{\text{lm}} = 0.01 \cdot I_3$ is applied to each observed landmark's covariance at every timestep:

$$P_j \leftarrow P_j + Q_{\text{lm}}. \tag{19}$$

This inflates P_j just enough to keep the Kalman gain responsive, allowing the filter to absorb the effective pose drift without destabilizing. The result was dramatic: successful updates increased from ∼5,000 to ∼65,000–155,000 depending on the dataset, and the landmark map became dense and coherent.

E. Visual-Inertial SLAM (Task 4)

The full VIO SLAM pipeline runs three phases per timestep:

Phase 1: IMU prediction. The pose and covariance are propagated using the same EKF prediction step from Task 1, starting from the previous SLAM-corrected pose rather than the raw IMU prediction.

Phase 2: Batch pose update via information form. Well-established landmarks (those with ≥ 6 observations and covariance trace < 15.0) are used to correct the current pose. The update accumulates contributions in information form:

$$\Lambda = \sum_j \frac{1}{\sigma_{\text{pose}}^2} H_j^\top H_j, \tag{20}$$

$$\eta = \sum_j \frac{1}{\sigma_{\text{pose}}^2} H_j^\top (z_j - \hat{z}_j), \tag{21}$$

where $H_j = \partial z / \partial \xi$ is the pose Jacobian. The pose correction is then:

$$\delta\xi = (\Sigma_t^{-1} + \Lambda)^{-1} \eta, \quad T_t \leftarrow T_t \cdot \exp(\widehat{\delta\xi}). \tag{22}$$

Improvement 1: phase reordering. The pose update is performed *before* the landmark update, so that landmark initialization and refinement use the best available (corrected) pose. This significantly improves landmark accuracy because DLT triangulation is highly sensitive to pose errors.

Improvement 2: conservative tuning. Several mechanisms prevent overcorrection:

- A high pose measurement noise $\sigma_{\text{pose}} = 25.0\,\text{px}$ (vs. $\sigma_{\text{meas}} = 2.0\,\text{px}$ for landmarks) damps the correction magnitude.
- An innovation norm filter rejects landmarks with $\|z - \hat{z}\| > 15.0\,\text{px}$ before they enter the information sum.
- A hard cap $\|\delta\xi\| \leq 0.15$ clips the per-step correction norm, preventing single-step jumps.
- The minimum observation count of 6 ensures only converged landmarks influence the pose.

Key insight: process noise helps mapping, hurts SLAM. While landmark process noise ($Q_{\text{lm}} = 0.01 \cdot I_3$) dramatically improves standalone landmark mapping (Task 3), it is set to zero in the full SLAM pipeline. In SLAM, pose corrections keep the trajectory consistent with observations, so additional landmark process noise inflates uncertainty unnecessarily and can cause the pose update to overcorrect.

Phase 3: Landmark initialization and update. After the pose is corrected, new landmarks are initialized via DLT triangulation and existing landmarks are updated using the same EKF equations as in Task 3. The corrected pose from Phase 2 is used, improving triangulation accuracy.

F. Feature Tracking (Task 2, Extra Credit)

For dataset 02, which provides raw stereo images instead of precomputed features, a custom feature tracking pipeline produces the $(4 \times M \times T)$ observation tensor.

Detection. New features are detected using Shi-Tomasi corners (`goodFeaturesToTrack`) with grid-based spatial balancing: the image is divided into a 4×6 grid with equal detection quota per cell, ensuring uniform coverage.

Temporal tracking. Features are tracked across consecutive left frames using Lucas-Kanade (LK) pyramidal optical flow. A forward-backward consistency check (threshold 3.0 px) rejects tracks that are not temporally stable.

Stereo matching. For each tracked left-image feature, the corresponding right-image point is found via LK flow from left to right. Validity is enforced by:

- Epipolar y-consistency: $|v_l - v_r| \leq 3.0\,\text{px}$.
- Minimum disparity: $|u_l - u_r| \geq 0.1\,\text{px}$.
- Bidirectional roundtrip check (threshold 1.0 px).

Problem: ID reuse corrupts the observation tensor. In the initial implementation, a slot-pool approach recycled feature IDs when tracks were lost. This caused different physical points to share the same column j in the tensor, which catastrophically confused the EKF—it would attempt to update a landmark initialized from one physical point with observations from a completely different point, leading to divergence.

Fix: monotonic ID counter. Replacing the slot-pool with a monotonic counter ensures each column in the output tensor corresponds to exactly one physical landmark across the entire sequence. Combined with an increased feature budget (250 active tracks, 15,000 total IDs) and the relaxed forward-backward threshold, this produced a clean observation tensor that downstream EKF modules could process correctly.

IV. Results

A. Task 1: IMU-Only Trajectory

Fig. 1 shows the IMU-predicted trajectories for all three datasets. The poses are propagated purely from IMU velocities without any visual correction. As expected, the trajectories exhibit unbounded drift over time due to the open-loop integration of noisy IMU measurements.

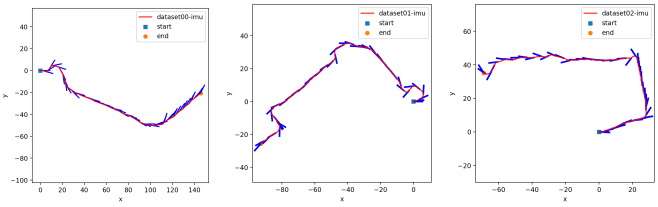

Fig. 1. IMU-only predicted trajectories for datasets 00 (left), 01 (center), and 02 (right). Arrows indicate heading direction.

B. Task 3: Landmark Mapping with Fixed Poses

Fig. 2 shows the landmark maps generated with poses fixed at the IMU predictions. The landmark process noise $Q_{\text{lm}} = 0.01 \cdot I_3$ was critical for preventing covariance collapse and achieving dense, coherent maps. Table I summarizes the mapping statistics.

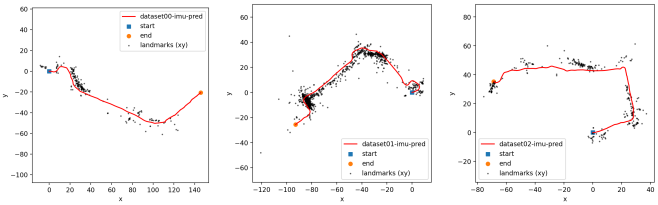

Fig. 2. Landmark maps (black dots) overlaid on IMU trajectories for datasets 00 (left), 01 (center), and 02 (right).

TABLE I
LANDMARK MAPPING STATISTICS ACROSS DATASETS (WITH $Q_{\text{LM}} = 0.01 \cdot I_3$, $d_{\min} = 2.0$ PX).

	Dataset 00	Dataset 01	Dataset 02
Initialized	289	1,410	360
Reliable	218	1,057	321
Updates	5,425	8,111	23,541
Reproj median (px)	31.18	20.95	4.79

C. Task 4: Visual-Inertial SLAM

Fig. 3 shows the SLAM-corrected trajectories (green) overlaid on IMU-only predictions (blue). The visual corrections pull the trajectory toward geometrically consistent positions, reducing drift particularly in the x-y plane. Conservative tuning prevents overcorrection artifacts.

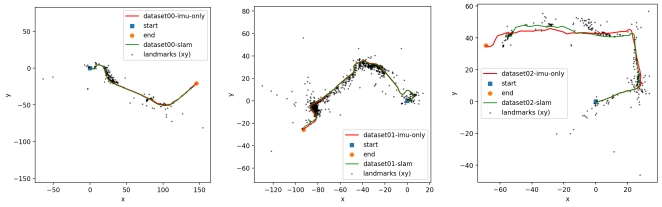

Fig. 3. SLAM results for datasets 00 (left), 01 (center), and 02 (right). Blue: IMU-only prediction. Green: SLAM-corrected trajectory. Black: landmark map.

TABLE II
SLAM TRAJECTORY DEVIATION FROM IMU-ONLY (METERS).

	Dataset 00	Dataset 01	Dataset 02
Mean deviation	1.41	0.54	4.15
Max deviation	4.77	1.98	13.92
Pose updates	308	445	4,068
Reliable landmarks	215	1,017	324

TABLE III
KEY SLAM TUNING PARAMETERS AND THEIR ROLES.

Parameter	Value	Role
σ_{pose}	25.0 px	Damps pose correction magnitude
Min observations	6	Ensures landmark convergence
$\|\delta\xi\|_{\max}$	0.15	Clips per-step correction
Innovation filter	15.0 px	Rejects large residuals
Max pose landmarks	50	Limits batch size
Landmark Q_{lm}	0.0	Disabled (see Sec. III-D)

D. Task 2: Feature Tracking (Extra Credit)

Fig. 4 compares the landmark map and SLAM trajectory produced using tracked features on dataset 02 versus the provided precomputed features shown in Figs. 2 and 3 (rightmost panels).

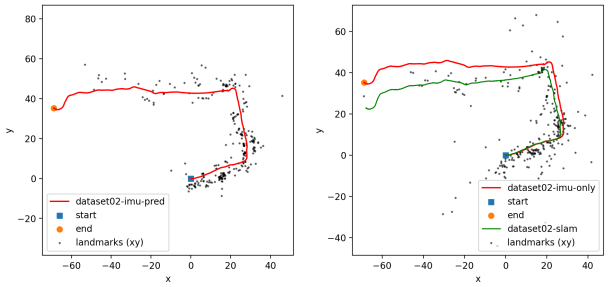

Fig. 4. Results on dataset 02 using custom-tracked features. Left: landmark map. Right: SLAM trajectory and map.

The tracked features produce qualitatively similar results to the provided features, validating the tracking pipeline. The monotonic ID counter fix was essential—the slot-pool reuse bug produced unusable maps before correction.

E. Discussion

What worked. Analytic Jacobians for both landmark and pose updates avoided the cost and inaccuracy of numerical differentiation. Phase reordering (pose correction before landmark update) in the SLAM pipeline improved map quality by ensuring triangulation uses the best available pose. Conservative pose correction tuning (high σ_{pose}, innovation filter, norm cap) prevented overcorrection while still providing meaningful drift reduction.

What did not work initially. The most impactful failure was covariance collapse in the landmark EKF: without process noise, the covariance shrank so quickly that the Kalman gain became negligible and the filter effectively stopped updating. Adding $Q_{\text{lm}} = 0.01 \cdot I_3$ resolved this entirely. In the feature tracker, ID reuse corrupted the observation tensor and prevented the downstream EKF from converging. Both issues had subtle symptoms (the filter ran without errors but produced sparse or chaotic maps) that required careful diagnosis.

Limitations. The z-coordinate of landmarks remains unreliable because the datasets have limited vertical motion, providing poor observability in that direction. The per-landmark independent covariance assumption prevents cross-landmark correlations from being exploited, which a full joint filter or graph-based approach would capture. Loop closure is not implemented, so accumulated drift in long sequences cannot be corrected retroactively.