# Visual-Inertial SLAM via Extended Kalman Filter

Yixin Zhang

## I. INTRODUCTION

Concurrent estate estimation and map reconstruction, through the integration of visual and inertial measurements, has garnered considerable attention within the Robotics and Computer Vision communities. This combination of visual-inertial sensors emerges as an optimal substitute for GPS in scenarios where GPS availability is compromised. The appeal of this sensor suite lies in its compact size, low cost, and the complementary nature of its components. Visual SLAM (Simultaneous Localization and Mapping) excels in environments with distinct visual features, offering precise tracking and detailed map construction. However, its performance is hindered by motion blur, occlusions, and changes in lighting, due to inherent sensor limitations. Conversely, inertial sensors, capable of high-frequency self-state updates, maintain their reliability amidst aggressive movements and contribute to determining the absolute scale of the state, despite their tendency towards noise and rapid divergence. By fusing data from inertial sensors with visual SLAM in a tightly integrated manner, both the robustness and accuracy of state estimation can be significantly enhanced.

Our work introduces a comprehensive approach to visual-inertial Simultaneous Localization and Mapping (SLAM) through the integration of an Extended Kalman Filter (EKF) with SE(3) kinematics and stereo-camera observations. The key contributions of our project are as follows:

- **IMU Localization via EKF Prediction:** We propose an advanced EKF prediction step that utilizes SE(3) kinematics in conjunction with linear and angular velocity measurements from the IMU to precisely estimate the IMU's pose over time. This approach significantly enhances the robustness of pose estimation amidst the inherent noise and bias present in IMU data, establishing a robust foundation for the subsequent mapping and localization processes.
- **Efficient Landmark Mapping via EKF Update:** Acknowledging the computational challenges posed by the processing of extensive visual feature measurements, our methodology introduces an efficient EKF update mechanism for landmark mapping. By selectively leveraging visual observations, we effectively manage computational complexity while achieving high precision in the estimation of landmark positions.
- **Visual-Inertial SLAM Algorithm:** The fusion of the IMU localization and landmark mapping techniques described above results in an exhaustive visual-inertial SLAM algorithm. Incorporating an update step for the IMU pose based on stereo-camera observation models, our algorithm not only tackles computation limitation but also significantly improves the overall accuracy of mapping. This integration marks a considerable advancement in visual-inertial SLAM, offering a robust solution for navigation and mapping in environments where GPS is unavailable.

## II. PROBLEM FORMULATION

This section introduces the robot system and the SLAM algorithm. The robot system discussion focuses on the sensors equipped and data provided. In contrast, the SLAM portion delves into the algorithmic formulation for localization and mapping.

### A. Robot System

The data provided for the estimation process is below, noted that IMU Measurements data and Visual Feature Measurements are synchronized.

- **IMU Measurements:** These consist of the linear velocity $v_t \in \mathbb{R}^3$ and angular velocity $\omega_t \in \mathbb{R}^3$ of the body, with coordinates expressed in the body frame of the IMU.
- **Visual Feature Measurements:** The pixel coordinates $z_t \in \mathbb{R}^{4\times M}$ of detected visual features from $M$ point landmarks come with precomputed correspondences between the left and the right camera frames. For landmarks $i$ that were not observable at time $t$, the measurement is given by $z_{t,i} = [-1 \ \ -1 \ \ -1 \ \ -1]^\mathsf{T}$, indicating a missing observation.
- **Time Stamps:** $\tau_t$ in unix time.

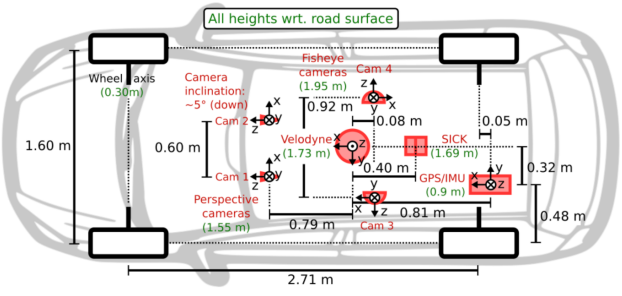


Fig. 1: Robot System and Sensor Configurations

### B. SLAM Framework

This section outlines the fundamental components of a visual-inertial SLAM system, focusing on IMU localization, landmark mapping, and the integration of visual and inertial sensors.

*1) IMU Localization via EKF Prediction:* The process begins with IMU localization, where the system uses data from an Inertial Measurement Unit (IMU). The IMU provides linear acceleration, $\mathbf{a}_t \in \mathbb{R}^3$, and rotational velocity, $\omega_t \in \mathbb{R}^3$. Using these inputs, the system predicts the world-frame IMU pose, $T_I \in SE(3)$, over time through Extended Kalman Filter (EKF) prediction techniques. This step is crucial for understanding the device's movement and orientation within its environment.

*2) Landmark Mapping via EKF Update:* Assuming the IMU pose, $T_t \in SE(3)$, is known, the next step focuses on mapping environmental landmarks. Given observations $\mathbf{z}t := \begin{bmatrix} \mathbf{z}t, 1^\top & \cdots & \mathbf{z}_{t,N_t}^\top \end{bmatrix}^\top \in \mathbb{R}^{4N_t}$ over time, the goal is to estimate the coordinates $\mathbf{m} := \begin{bmatrix} \mathbf{m}_1^\top & \cdots & \mathbf{m}_M^\top \end{bmatrix}^\top \in \mathbb{R}^{3M}$ of landmarks. This estimation relies on known or externally provided data association $\Delta_t : 1,\ldots,M \to 1,\ldots,N_t$, indicating the correspondence between landmarks and observations at each time point. Since landmarks are assumed to be static, no motion model or prediction step is required for $\mathbf{m}$.

*3) Visual-Inertial SLAM Algorithm:* The Visual-Inertial SLAM Algorithm integrates data from both the IMU and a camera. The IMU provides linear acceleration, $\mathbf{a}_t \in \mathbb{R}^3$, and rotational velocity, $\omega t \in \mathbb{R}^3$. The camera captures features, $\mathbf{z}t, i \in \mathbb{R}^4$, representing left and right image pixels for $i = 1,\ldots,N_t$. Using this information, the algorithm outputs the world-frame IMU pose, $T_I \in SE(3)$, over time, and the world-frame coordinates, $\mathbf{m}j \in \mathbb{R}^3$, of point landmarks corresponding to the visual features $\mathbf{z}t, i \in \mathbb{R}^4$ for $j = 1,\ldots,M$. This comprehensive approach allows for accurate navigation and mapping in complex environments.

## III. TECHNICAL APPROACH

### A. IMU Localization via EKF Prediction

Within our project, we aim to estimate the evolving pose of an IMU over time, leveraging a kinematic model that encapsulates both the deterministic and stochastic behaviors of its movements.

*1) Pose Kinematics with Perturbation:* We define the continuous-time motion model for the IMU pose $T_t$, inclusive of noise $\mathbf{w}(t)$, via the differential equation:

$$\dot{T} = T(\hat{\mathbf{u}} + \hat{\mathbf{w}}), \quad \mathbf{u}(t) := \begin{bmatrix} \mathbf{v}(t) \\ \omega(t) \end{bmatrix} \in \mathbb{R}^6,$$

where $\mathbf{v}(t)$ and $\omega(t)$ represent the linear velocity and angular velocity, respectively. To model the pose $T$ as a Gaussian distribution, we express it as a nominal pose $\mu \in SE(3)$, perturbed by a small deviation $\hat{\delta} \in \mathfrak{se}(3)$:

$$T = \mu \exp(\hat{\delta}) \approx \mu(I + \hat{\delta}\mu),$$

Transitioning to a discrete-time framework with a timestep $\tau_t$, the model is discretized as follows:

$$\begin{aligned} \text{Nominal:} \quad & \mu_{t+1} = \mu_t \exp(\tau_t \hat{\mathbf{u}}_t), \\ \text{Perturbation:} \quad & \delta\mu_{t+1} = \exp(-\tau_t \hat{\mathbf{u}}_t)\, \delta\mu_t + \mathbf{w}_t. \end{aligned}$$

The input $\hat{\mathbf{u}}$ is structured as:

$$\hat{\mathbf{u}} := \begin{bmatrix} \hat{\omega} & \hat{\mathbf{v}} \\ 0 & \hat{\omega} \end{bmatrix} \in \mathbb{R}^{6\times 6}.$$

*2) Summary of EKF Prediction Step:* For the EKF prediction step, starting from the prior distribution $T_t | \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$, we incorporate both nominal and perturbation dynamics over the interval $\tau_t$:

$$\begin{aligned} \mu_{t+1|t} &= \mu_{t|t} \exp(\tau_t \hat{\mathbf{u}}_t), \\ \delta\mu_{t+1|t} &= \exp(-\tau_t \hat{\mathbf{u}}_t)\, \delta\mu_{t|t} + \mathbf{w}_t. \end{aligned}$$

To account for motion model noise $\mathbf{w}_t \sim \mathcal{N}(0, W)$ and potential biases in IMU measurements, the covariance matrix is updated as:

$$\Sigma_{t+1|t} = \exp(-\tau_t \hat{\mathbf{u}}_t)\, \Sigma_{t|t} \exp(-\tau_t \hat{\mathbf{u}}_t)^\top + W,$$

thus integrating the IMU's measurement inaccuracies and biases into the prediction phase, thereby fortifying the pose estimation's reliability.

### B. Efficient Landmark Mapping via EKF Update

Now, addressing the mapping-only challenge, we assume access to the IMU pose over time and camera feature data, aiming to incrementally build a map.

*1) Observation Model and Its Jacobian:* We utilize the camera observation model, which incorporates measurement noise $\mathbf{v}_{t,i} \sim \mathcal{N}(0, V)$, expressed as:

$$\mathbf{z}_{t,i} = h(T_t, \mathbf{m}_j) + \mathbf{v}_{t,i} := K_s \pi\left(o T_I T_t^{-1} \underline{\mathbf{m}}_j\right) + \mathbf{v}_{t,i},$$

where $\underline{\mathbf{m}}_j$ denotes the homogeneous coordinates of the point $\mathbf{m}_j$:

$$\underline{\mathbf{m}}_j := \begin{bmatrix} \mathbf{m}_j \\ 1 \end{bmatrix}.$$

The projection function $\pi(\mathbf{q})$ and its derivative are defined as:

$$\pi(\mathbf{q}) := \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4, \quad \frac{d\pi}{d\mathbf{q}}(\mathbf{q}) = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}.$$

Aggregating all observations into a $4N_t$ vector at time $t$, we represent them as:

$$\mathbf{z}_t = K_s \pi\left(o T_I T_t^{-1} \underline{\mathbf{m}}\right) + \mathbf{v}_t,$$

$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, I \otimes V), \quad I \otimes V := \begin{bmatrix} V & & \\ & \ddots & \\ & & V \end{bmatrix}.$$

To determine the Stereo Camera Jacobian, we apply the chain rule as follows:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{m}_j} h(T_{t+1}, \mathbf{m}_j) &= K_s \frac{\partial \pi}{\partial \mathbf{q}} \left( o T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) \frac{\partial}{\partial \mathbf{m}_j} \left( o T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) \\ &= K_s \frac{\partial \pi}{\partial \mathbf{q}} \left( o T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) o T_I T_{t+1}^{-1} \frac{\partial \underline{\mathbf{m}}_j}{\partial \mathbf{m}_j} \\ &= K_s \frac{\partial \pi}{\partial \mathbf{q}} \left( o T_I T_{t+1}^{-1} \underline{\mathbf{m}}_j \right) o T_I T_{t+1}^{-1} P^\top,\end{aligned}$$

where $P = \begin{bmatrix} I & 0 \end{bmatrix} \in \mathbb{R}^{3\times 4}$. This formulation meticulously applies the chain rule to compute the derivative necessary for mapping construction.

*2) Summary of the Mapping Update Step:* In the mapping update step, we commence with the prior assumption that the map points $\mathbf{m}$, given the observations up to time $t$, follow a Gaussian distribution: $\mathbf{m}|\mathbf{z}_{0:t} \sim \mathcal{N}(\mu_t, \Sigma_t)$. This distribution is characterized by a mean vector $\mu_t \in \mathbb{R}^{3M}$ and a covariance matrix $\Sigma_t \in \mathbb{R}^{3M\times 3M}$, where $M$ represents the number of map points.

Upon receiving a new observation $\mathbf{z}_{t+1} \in \mathbb{R}^{4N_{t+1}}$, the Extended Kalman Filter (EKF) update step is executed as follows:

1) Compute the Kalman gain $K_{t+1}$, which balances the prior estimate uncertainty and the measurement uncertainty:

$$K_{t+1} = \Sigma_t H_{t+1}^\top \left( H_{t+1} \Sigma_t H_{t+1}^\top + I \otimes V \right)^{-1},$$

where $H_{t+1}$ is the Jacobian of the observation model with respect to the map points, and $I \otimes V$ denotes the measurement noise covariance matrix.

2) Update the mean vector $\mu_{t+1}$ of the map points, integrating the discrepancy between the new observation $\mathbf{z}_{t+1}$ and the predicted observation $\tilde{\mathbf{z}}_{t+1}$, derived from the current estimate:

$$\mu_{t+1} = \mu_t + K_{t+1}(\mathbf{z}_{t+1} - \underbrace{K_s \pi \left( o T_I T_{t+1}^{-1} \underline{\mu}_t \right)}_{\tilde{\mathbf{z}}_{t+1}}),$$

where $\tilde{\mathbf{z}}_{t+1}$ represents the expected observation based on the current map estimate and the IMU pose.

3) Finally, update the covariance matrix $\Sigma_{t+1}$ to reflect the reduced uncertainty following the incorporation of the new observation:

$$\Sigma_{t+1} = (I - K_{t+1} H_{t+1}) \Sigma_t,$$

thus completing the update step and refining the map estimate with the latest observation.

This sequential update process progressively enhances the accuracy of the map by assimilating new observations and adjusting the estimates of map points accordingly.

*3) Map Initialization:* In the context of landmark mapping, we encounter scenarios where certain landmarks have not been observed previously. To initialize these landmarks, we employ Bearing Measurement Triangulation, aiming to determine the coordinates of a point $\mathbf{m} \in \mathbb{R}^3$ as observed by two cameras from the perspective of the first camera.

Assuming that we possess pixel coordinates $\mathbf{z}_1 \in \mathbb{R}^2$ and $\mathbf{z}_2 \in \mathbb{R}^2$ from two calibrated cameras. These cameras have a known relative transformation, denoted by $\mathbf{p} \in \mathbb{R}^3$ (the translation of camera 2 relative to camera 1) and $R \in SO(3)$ (the rotation of camera 2 relative to camera 1). The relationship between the observed pixel coordinates and the real-world position of the landmark can be described as follows:

$$\lambda_1 \underline{z}_1 = \mathbf{m}, \qquad \lambda_1 = \mathbf{e}_3^\top \mathbf{m}, \tag{1}$$

$$\lambda_2 \underline{z}_2 = R^\top (\mathbf{m} - \mathbf{p}), \qquad \lambda_2 = \mathbf{e}_3^\top R^\top (\mathbf{m} - \mathbf{p}), \tag{2}$$

where $\lambda_2$ can be expressed as a function of $\lambda_1$:

$$\lambda_2 = \lambda_1 \mathbf{e}_3^\top \mathbf{R}^\top \underline{\mathbf{z}}_1 - \mathbf{e}_3^\top \mathbf{R}^\top \mathbf{p},$$

leading to the equation:

$$\left( \lambda_1 \mathbf{e}_3^\top \mathbf{R}^\top \underline{\mathbf{z}}_1 - \mathbf{e}_3^\top \mathbf{R}^\top \mathbf{p} \right) \underline{z}_2 = \lambda_1 \mathbf{R}^\top \underline{\mathbf{z}}_1 - \mathbf{R}^\top \mathbf{p},$$

and further simplified to:

$$\left( \mathbf{R}^\top \mathbf{p} - \mathbf{e}_3^\top \mathbf{R}^\top \mathbf{p} \underline{z}_2 \right) \frac{1}{\lambda_1} = \left( \mathbf{R}^\top \underline{\mathbf{z}}_1 - \mathbf{e}_3^\top \mathbf{R}^\top \underline{\mathbf{z}}_1 \underline{z}_2 \right),$$

denoting $\mathbf{a} = \mathbf{R}^\top \mathbf{p} - \mathbf{e}_3^\top \mathbf{R}^\top \mathbf{p} \underline{z}_2$ and $\mathbf{b} = \mathbf{R}^\top \underline{\mathbf{z}}_1 - \mathbf{e}_3^\top \mathbf{R}^\top \underline{\mathbf{z}}_1 \underline{z}_2$, we obtain:

$$\frac{1}{\lambda_1} = \frac{\mathbf{a}^\top \mathbf{b}}{\mathbf{a}^\top \mathbf{a}} \quad \Rightarrow \quad \mathbf{m} = \frac{\mathbf{a}^\top \mathbf{a}}{\mathbf{a}^\top \mathbf{b}} \underline{z}_1.$$

These equations facilitate the initialization of landmarks that have not been previously observed. By solving for $\mathbf{m}$, we can determine the spatial coordinates of new landmarks relative to the first camera's frame, leveraging the geometric relationships imposed by the pixel coordinates observed from both cameras.

*4) Implementation Details in Python:* In the map initialization process, landmarks not previously observed are introduced through bearing measurement triangulation. This approach facilitates the addition of new landmarks to our map, utilizing data derived from observations.

For every update step, a crucial operation is the computation of the observation model Jacobian, denoted as $H_{t+1} \in \mathbb{R}^{4N_t \times 3M}$, and evaluated at $\mu_t$. This Jacobian matrix comprises block elements $H_{t+1,i,j} \in \mathbb{R}^{4\times 3}$, each corresponding to the partial derivatives of the observation model with respect to the position of a specific landmark. Given the computational intensity of this task, we implement a strategy to efficiently manage the complexity. During each update step, we initially identify the landmarks relevant to the current observation, denoted by $\mu_t'$ and $\Sigma_t'$. This selective approach significantly reduces the computational load by concentrating on a relevant subset of landmarks. Consequently, we compute a reduced Jacobian matrix $H_{t+1}' \in \mathbb{R}^{4N_t \times 3N_t}$, where $H_{t+1,i,j}$ includes only those landmarks directly associated with the current observation. Upon ascertaining the updated values $\mu_{t+1}'$ and $\Sigma_{t+1}'$ for the associated landmarks, we reintegrate these values into the original state vectors and covariance

matrices. Specifically, we update the respective components in $\mu_t$ and $\Sigma_t$ to obtain the new estimates $\mu_{t+1}$ and $\Sigma_{t+1}$. This methodology ensures that our map and landmark estimates remain current and precise, incorporating the latest observations while maintaining computational efficiency.

### C. Visual-Inertial SLAM Algorithm

In this section, we address the complex task of concurrently estimating the IMU trajectory and constructing a landmark map based on time-series data from IMU measurements and visual feature observations. Our approach builds on previously established methods for IMU prediction and visual map updates, with a specific focus on incorporating the IMU update step in response to feature observations.

*1) IMU EKF Update:* The IMU pose at time $t+1$, conditioned on all preceding observations and control inputs, is presumed to follow a Gaussian distribution: $T_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim \mathcal{N}\left(\mu_{t+1|t}, \Sigma_{t+1|t}\right)$. This distribution is defined by a mean pose $\mu_{t+1|t} \in SE(3)$ and a $6\times 6$ covariance matrix $\Sigma_{t+1|t}$, encapsulating the uncertainty associated with the pose estimate.

**Observation Model:** Echoing the visual mapping scenario, our observation model for feature measurements concentrates on the IMU pose $T_{t+1} \in SE(3)$ rather than the landmark positions. Integrated with measurement noise $\mathbf{v}_{t+1,i} \sim \mathcal{N}(0,V)$, the model is articulated as follows:

$$\begin{aligned}\mathbf{z}_{t+1,i} &= h\left(T_{t+1}, \mathbf{m}_j\right) + \mathbf{v}_{t+1,i} \\ &= K_s \pi\left(oT_I T_{t+1}^{-1} \underline{m}_j\right) + \mathbf{v}_{t+1,i},\end{aligned}$$

representing the projection of the $j$-th landmark position $\mathbf{m}_j$ into the image frame at time $t+1$, influenced by the current IMU pose and observed with additive Gaussian noise.

Further simplification and linearization yield:

$$\begin{aligned}\mathbf{z}_{t+1,i} &= K_s \pi\left(oT_I\left(\mu_{t+1|t}\exp(\hat{\delta\mu})\right)^{-1}\underline{\mathbf{m}}_j\right) + \mathbf{v}_{t+1,i} \\ &\approx K_s \pi\left(oT_I(I-\hat{\delta\mu})\mu_{t+1|t}^{-1}\underline{\mathbf{m}}_j\right) + \mathbf{v}_{t+1,i} \\ &= K_s \pi\left(oT_I \mu_{t+1|t}^{-1}\underline{\mathbf{m}}_j\right) \\ &\quad - K_s \pi\left(oT_I\left(\mu_{t+1|t}^{-1}\underline{\mathbf{m}}_j\right)^{\odot}\delta\mu\right) + \mathbf{v}_{t+1,i} \\ &\approx \underbrace{K_s \pi\left(oT_I \mu_{t+1|t}^{-1}\underline{\mathbf{m}}_j\right)}_{\tilde{\mathbf{z}}_{t+1,i}} \\ &\quad \underbrace{-K_s \frac{d\pi}{d\mathbf{q}}\left(oT_I \mu_{t+1|t}^{-1}\underline{\mathbf{m}}_j\right) oT_I\left(\mu_{t+1|t}^{-1}\underline{\mathbf{m}}_j\right)^{\odot}}_{H_{t+1,i}}\delta\mu + \mathbf{v}_{t+1,i}.\end{aligned}$$

**Observation Model Jacobian:** The Jacobian $H_{t+1} \in \mathbb{R}^{4N_{t+1}\times 6}$, crucial for the update mechanism, is evaluated at the mean IMU pose $\mu_{t+1|t}$. It quantitatively relates how alterations in the IMU pose affect the perceived features in the image frame.

The overall EKF IMU update step is summarized as:

$$\begin{aligned}K_{t+1} &= \Sigma_{t+1|t} H_{t+1}^{\top}\left(H_{t+1}\Sigma_{t+1|t}H_{t+1}^{\top} + I\otimes V\right)^{-1}, \\ \mu_{t+1|t+1} &= \mu_{t+1|t}\exp\left(\left(K_{t+1}\left(\mathbf{z}_{t+1} - \tilde{\mathbf{z}}_{t+1}\right)\right)^{\wedge}\right), \\ \Sigma_{t+1|t+1} &= \left(I - K_{t+1}H_{t+1}\right)\Sigma_{t+1|t}.\end{aligned}$$

These steps underpin the IMU update in the Visual-Inertial SLAM algorithm, facilitating the integration of feature observations to refine both the IMU trajectory estimate and the landmark map concurrently.

### D. Imlemention Detials

In this section, we address the integration of IMU readings with visual observations, a crucial component for achieving precise and robust pose estimation. A notable preprocessing step involves adjusting velocity readings, which are initially provided in the vehicle's coordinate frame. To align these with the IMU's frame of reference, we implement a conversion that flips the y and z components of the velocity data. Additionally, we initialize the initial pose as $\mathrm{diag}(1,-1,-1,1)$, denoting the IMU's original orientation in world coordinates.

The procedure for each time step is as follows:

1) **IMU Prediction Step:** The process begins with the IMU prediction step, wherein the IMU's motion is estimated based on its readings.
2) **Visual Mapping Update Strategy:** Utilizing the visual mapping update strategy, the map is refined with observations of new landmarks. It is imperative to filter out landmarks significantly distant from the IMU, as these are likely outliers that could adversely impact the pose estimation's accuracy.
3) **IMU Update Step:** After refining the map and excluding potential outliers, the IMU update step is performed. This step integrates the visual observations with the predicted IMU state, thereby enhancing the pose estimation.

By meticulously transforming the velocity readings and judiciously managing landmark data, in conjunction with merging IMU predictions with visual observations, we ensure that the fusion of these data sources significantly improves the system's capability to estimate the IMU trajectory and maintain an accurate environmental map. This methodology highlights the significance of data preprocessing and selective integration in visual-inertial SLAM.

### E. Memory Management

Effective memory management is paramount in our SLAM system to ensure efficient processing and optimal performance.

*1) Using Windows 11 to Increase Virtual Memory:* To augment the system's capability to handle large datasets typical in SLAM applications, increasing the virtual memory in Windows 11 can be beneficial. Virtual memory acts as an extension of the physical memory, allowing the system to manage more data concurrently.

*2) Downsampling the Landmarks:* To further optimize memory usage, downsampling the landmarks is a critical strategy. This process involves reducing the number of landmarks used in the SLAM algorithm without significantly compromising the accuracy of the map or the trajectory estimation. Downsampling can be achieved by:

- Prioritizing landmarks based on their observational frequency and geometrical distribution across the map.
- Merging nearby landmarks that are likely to represent the same physical point in the environment.
- Removing outliers or landmarks that have been observed fewer times, as they are more prone to errors and less informative.

Implementing these strategies helps in managing the computational load and memory requirements, ensuring smoother and more efficient SLAM processing.

## IV. Results and Comparative Analysis

In our Visual-Inertial SLAM framework, the articulation of noise within the motion and observation models is crucial for optimizing estimation accuracy. The motion noise is characterized by a diagonal matrix: Motion Noise = np.diag([1e-3, 1e-3, 1e-2, 1e-1, 1e-1, 1e-4]), where the variances are allocated to different motion dimensions, reflecting a higher confidence in the IMU's linear motion measurements compared to rotational motions, with a particular emphasis on the yaw rate ($w_z$) due to the robot's planar movement.

The observation noise, set as a scalar value of 4, accounts for the generally higher uncertainty in camera measurements relative to the IMU. This parameterization, especially the prioritization of $v_x$, $v_y$, and $w_z$, highlights the need for precise handling of translational and rotational motions pertinent to navigating roadways. These noise models are integral to the balance and integration of IMU and camera data within our SLAM system.

### A. IMU Prediction Step

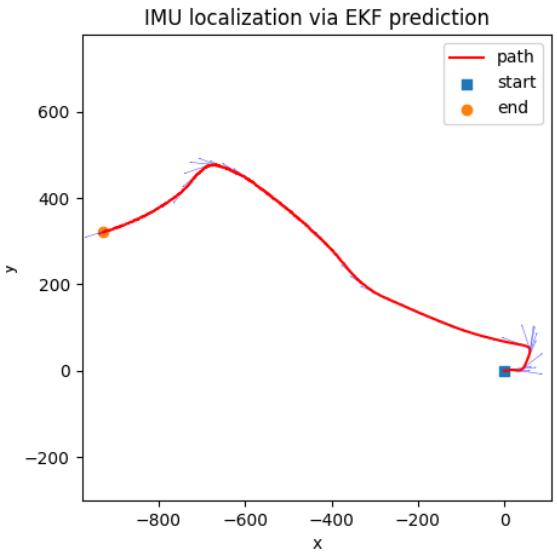


Fig. 2: Dataset 03

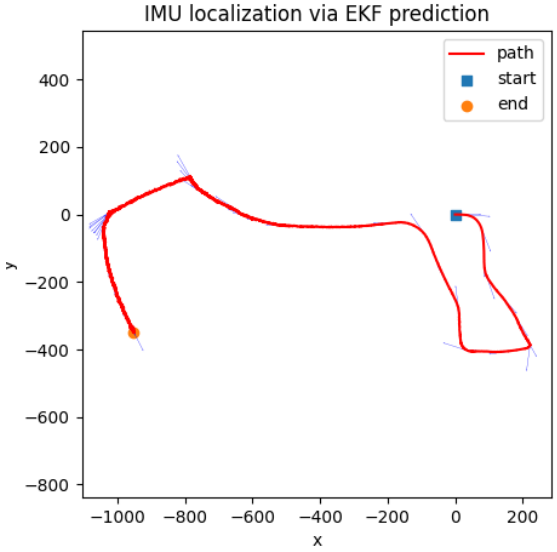


Fig. 3: Dataset 10

### B. Map Intialization

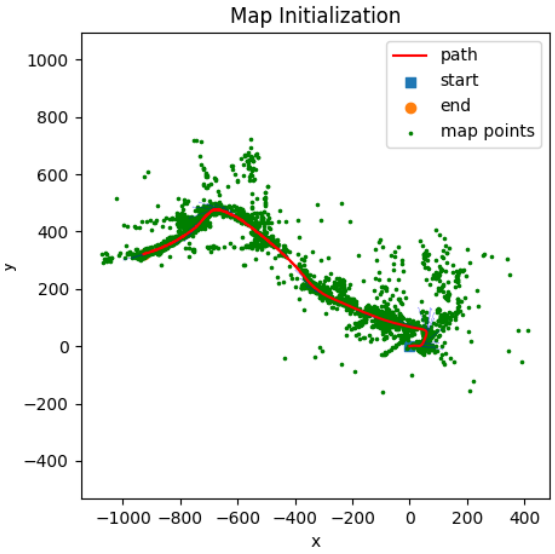


Fig. 4: Dataset 03

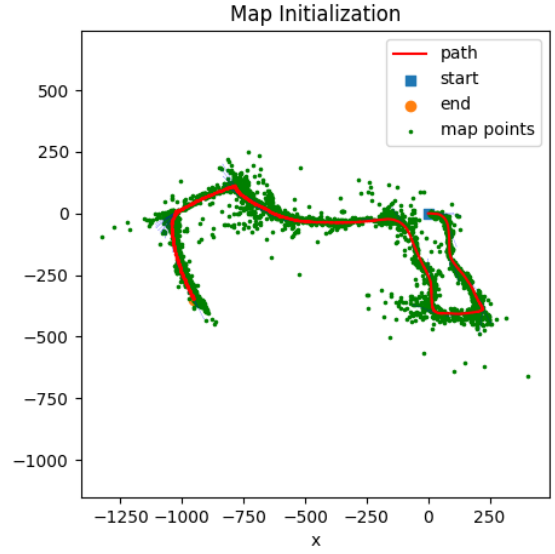


Fig. 5: Dataset 10

### C. Visual Mapping Update

Following the visual mapping phase, we observe a dispersion of all landmarks, a consequence of continuously incorporating observation noise into their estimations. This dispersion is an expected outcome as each addition of noise incrementally adjusts the perceived location of landmarks, leading to a more spread out distribution. It's important to note that outliers, specifically those significantly distant from the expected positions, are systematically excluded from our analysis to maintain the integrity and accuracy of our mapping process.

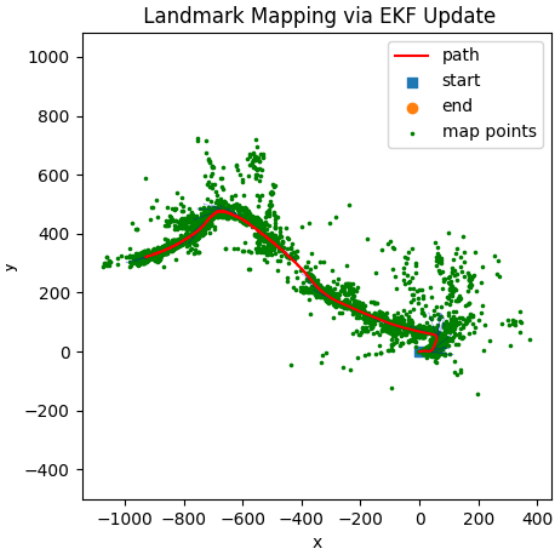


Fig. 6: Dataset 03

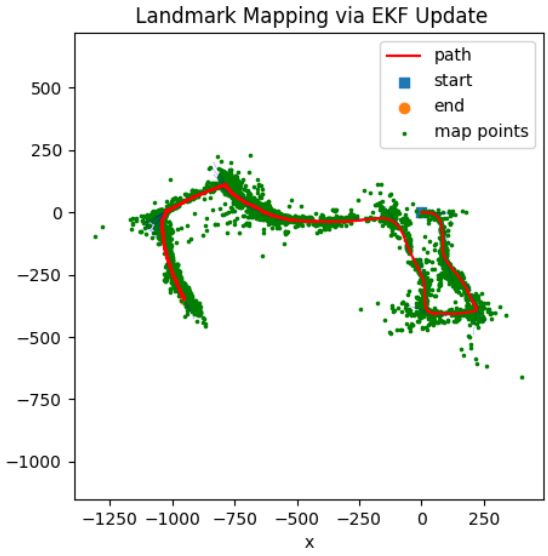


Fig. 7: Dataset 10

### D. Overall SLAM Algorithm

In our SLAM implementation, we effectively mitigate cumulative error by harnessing the complementary strengths of IMU and camera data. This fusion results in a refined trajectory and map that not only appear coherent but also align closely with the video evidence. This synergy between inertial measurements and visual cues ensures our reconstructed path and environmental mapping are both robust and consistent, showcasing the practicality and accuracy of our approach in navigating and understanding complex spaces.

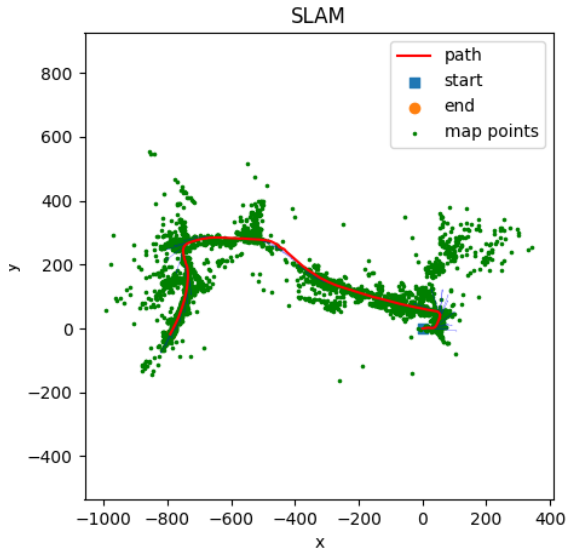


Fig. 8: Dataset 03

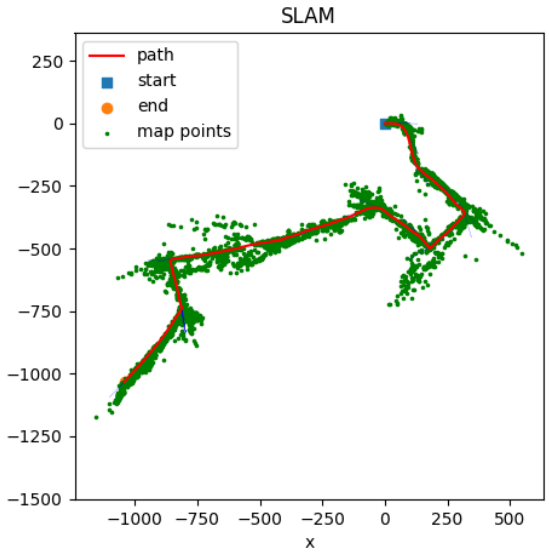


Fig. 9: Dataset 10

## V. CONCLUSIONS

In conclusion, this study presents a comprehensive approach to Visual-Inertial SLAM, addressing key challenges in trajectory estimation and landmark mapping through the integration of IMU readings and visual observations. By refining the IMU EKF update with precise observation models, implementing memory management strategies such as increasing virtual memory in Windows 11, and optimizing landmark data through downsampling, we significantly enhance the system's accuracy and efficiency. These innovations not only underscore the importance of data preprocessing and selective integration in SLAM but also pave the way for more reliable and computationally efficient navigation and mapping technologies in dynamic environments.

## ACKNOWLEDGMENT

The author would also like to thank for the constructive advice from Professor Nikolay Atanasov, all teaching assistants, and everyone who shared their ideas on Piazza and during office hours.