Project 3: Visual-Inertial SLAM

Behrad Rabiei
Dept. of Electrical and Computer Engineering
University of California San Diego
Email: brabiei@ucsd.edu

***Abstract*—This paper presents a comprehensive report on Project 3 for the ECE 276A course, titled "Sensing and Estimation in Robotics". The core aim of this project was to implement visual-inertial simultaneous localization and mapping (SLAM) using an extended Kalman filter (EKF). We were provided with synchronized measurements from an inertial measurement unit (IMU) and a stereo camera, along with the intrinsic camera calibration and the extrinsic calibration between the two sensors. This calibration specifies the transformation from the left camera frame to the IMU frame. The overall approach begins by implementing only the prediction step of the EKF to estimate motion/trajectory. Subsequently, we implemented the update step of the EKF to refine our initial estimates for the landmark locations. Finally, we combined both steps to implement VI-SLAM using the EKF.**

I. Introduction

In the evolving field of robotics, the ability to accurately perceive and understand the surrounding environment is crucial for the development of autonomous systems. One of the fundamental challenges in this domain is enabling robots to precisely locate themselves and map their surroundings simultaneously, a task known as Simultaneous Localization and Mapping (SLAM). This project delves into the realm of visual-inertial SLAM (VI-SLAM), which integrates visual data from a stereo camera and inertial measurements from an Inertial Measurement Unit (IMU) to achieve this objective. The integration of visual and inertial data offers a robust solution to SLAM by leveraging the complementary nature of the two data types: visual data provides rich environmental details while inertial data offers high-rate motion information. We utilize synchronized IMU measurements—comprising both linear and angular velocities expressed in the body frame of the IMU—and visual feature measurements from detected landmarks to construct and refine a map of the environment while concurrently updating the robot's position and orientation within it. This process is facilitated by the availability of timestamps for each set of measurements, intrinsic camera calibration parameters, and the extrinsic calibration between the IMU and the camera. These elements together form a comprehensive dataset that enables the implementation of an extended Kalman filter (EKF) based VI-SLAM system.

Our dataset includes: linear and angular velocity measurements from the IMU, expressed in the IMU's body frame (despite not being concentric with the vehicle, I assume IMU to be the body frame); pixel coordinates of detected visual features from stereo camera frames, including precomputed correspondences between the left and right cameras; timestamps for each set of measurements; intrinsic calibration details of the stereo camera system; and extrinsic calibration specifying the spatial relationship between the IMU and the left camera. These data are fundamental to our solution, as they enable the fusion of visual and inertial information through an Extended Kalman Filter (EKF), laying the groundwork for our VI-SLAM implementation. The intrinsic calibration helps us understand how the camera lens projects 3D world points onto the 2D image plane, while the extrinsic calibration is critical for transforming measurements between the camera and IMU coordinate frames. This integration of data and calibration aims to precisely track the robot's trajectory and map the environment, showcasing the potential of VI-SLAM in applications ranging from autonomous vehicles to robotic navigation in unknown territories.

II. Problem Formulation

Our goal is to get a reasonably accurate estimate of the location of a robot as it moves around in an unknown environment and to make a map of that environment over time based on data that is attained from our stereo camera.

A. What we Have

*1) **IMU Data**:* These measurements include the linear velocity and angular velocity of the body, both expressed in the body frame of the IMU at time t.

1. **Linear Velocity** (v_t): The linear velocity of the body at time t, expressed in the IMU's body frame. It is a vector in $\mathbb{R}^3$, representing the velocity along the X, Y, and Z axes of the IMU's body frame. Mathematically, it can be represented as:

$$v_t = \begin{bmatrix} v_{t_x} \\ v_{t_y} \\ v_{t_z} \end{bmatrix} \in \mathbb{R}^3$$

where v_{t_x}, v_{t_y}, and v_{t_z} are the components of the linear velocity in the IMU's body frame along its X, Y, and Z axes, respectively.

2. **Angular Velocity** (ω_t): The angular velocity of the body at time t, also expressed in the IMU's body frame. Similar to linear velocity, it is a vector in $\mathbb{R}^3$, which represents the rate of rotation around the X, Y, and Z axes of the IMU's body frame. It is given by:

$$\omega_t = \begin{bmatrix} \omega_{t_x} \\ \omega_{t_y} \\ \omega_{t_z} \end{bmatrix} \in \mathbb{R}^3$$

where ω_{t_x}, ω_{t_y}, and ω_{t_z} denote the angular velocity components around the IMU's body frame X, Y, and Z axes, respectively.

These vectors, v_t and ω_t, provide the essential kinematic information about the movement of the robot or device equipped with the IMU, capturing both its translational and rotational dynamics in the 3D space.

*2) **Visual Feature Measurements**:* Visual feature measurements consist of pixel coordinates for detected visual features from point landmarks. These landmarks are observed through both the left and right camera frames, with precomputed correspondences between them to facilitate stereo vision analysis.

1. **Feature Measurements** (z_t): At any given time t, z_t represents the pixel coordinates of visual features detected in the environment, expressed as a matrix in $\mathbb{R}^{4\times M}$, where M is the number of point landmarks. Each column of z_t corresponds to a specific landmark, and each landmark is represented by four values:

$$z_{t,i} = \begin{bmatrix} u^L_{t,i} \\ v^L_{t,i} \\ u^R_{t,i} \\ v^R_{t,i} \end{bmatrix}$$

for the i-th landmark, where: $u^L_{t,i}$ and $v^L_{t,i}$ are the horizontal and vertical pixel coordinates, respectively, in the left camera frame. $u^R_{t,i}$ and $v^R_{t,i}$ are the corresponding horizontal and vertical pixel coordinates in the right camera frame. $i = 1, 2, \ldots, M$, indexing the landmarks.

For landmarks that were not observable at time t, the measurement is denoted by:

$$z_{t,i} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

This notation indicates a missing observation for the i-th landmark.

*3) **Time Stamps**:* Each set of measurements from the IMU and the stereo camera is accompanied by a time stamp, τ_t, which records the specific time at which the measurements were taken. These time stamps are expressed in Unix time, which counts the number of seconds that have elapsed since January 1, 1970 (also known as the Unix epoch). Therefore, for any given measurement at time t, the corresponding time stamp can be defined as:

$$\tau_t = \text{Unix time at } t = \text{seconds since January 1, 1970}$$

The use of Unix time provides a standardized way of timestamping data, ensuring that measurements can be accurately synchronized and ordered in time.

*4) **Intrinsic Calibration**:* The intrinsic calibration of the stereo camera system includes two key components:

1. **Stereo Baseline** (b): The distance in meters between the centers of the left and right cameras.

2. **Camera Calibration Matrix** (K):

$$K = \begin{pmatrix} f_{su} & 0 & c_u \\ 0 & f_{sv} & c_v \\ 0 & 0 & 1 \end{pmatrix}$$

where f_{su} and f_{sv} are the focal lengths of the camera in pixels, scaled by the respective pixel dimensions in the u (horizontal) and v (vertical) directions, and c_u and c_v represent the optical center of the camera in pixel coordinates.

*5) **Extrinsic Calibration**:* The extrinsic calibration describes the spatial relationship between the left camera and the IMU, encapsulated by the transformation $IT_C \in SE(3)$. This transformation is for mapping measurements from the camera frame to the IMU frame and vice versa, enabling the integration of visual and inertial data.

Given that the IMU is mounted upside down on the vehicle, its frame orientation is defined as $x =$ forward, $y =$ right, and $z =$ down. This unconventional orientation may necessitate adjustments when interpreting the IMU's data, especially for trajectory estimation. In this case you can either start your initial orientation as identity but rolled by 180 degrees or flip the IMU values. I made no adjustment as the results matched the video that was provided.

B. Localization

In the EKF prediction step for 3D localization, the state of the robot is updated using the exponential map of the twist expressed in the robot's body frame, with noise w_t assumed to be zero mean Gaussian with covariance W. The state is represented by the homogeneous transformation matrix $\mathbf{T}_t$ and the update rule is given by the matrix exponential of the twist coordinates.

Given: $\mathbf{T}_t$ is the current pose of the robot at time t. $\mathbf{u}_t = \begin{bmatrix} \mathbf{v}_t \\ \boldsymbol{\omega}_t \end{bmatrix}$ is the control input at time t, composed of linear velocity $\mathbf{v}_t$ and angular velocity $\boldsymbol{\omega}_t$. τ_t is the time step between t and $t+1$. $\hat{\mathbf{u}}_t$ is the skew-symmetric matrix (hat operator) for twist coordinates. W is the process noise covariance matrix.

The EKF prediction update is then:

1. **State Update**:

$$\mathbf{T}_{t+1|t} = \mathbf{T}_{t|t} \exp(\tau_t \hat{\mathbf{u}}_t)$$

Here, $\exp(\cdot)$ denotes the matrix exponential, and $\hat{\mathbf{u}}_t$ is constructed from the control inputs as:

$$\hat{\mathbf{u}}_t = \begin{bmatrix} \hat{\omega}_t & \mathbf{v}_t \\ \mathbf{0}^T & 0 \end{bmatrix}$$

with $\hat{\omega}_t$ being the 3×3 skew-symmetric matrix form of $\boldsymbol{\omega}_t$.

2. **Covariance Update**:

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

where $\Sigma_{t|t}$ is the current covariance estimate, and W represents the process noise. The operation $\exp(-\tau_t \hat{\mathbf{u}}_t)$ affects the covariance by rotating and translating it according to the twist, taking into account the uncertainty in the robot's motion.

For the $\hat{\mathbf{u}}_t$ in $\mathbb{R}^{6\times 6}$, it's given by:

$$\hat{\mathbf{u}}_t = \begin{bmatrix} \hat{\omega}_t & \hat{\mathbf{v}}_t \\ \mathbf{0} & \hat{\omega}_t \end{bmatrix}$$

where $\hat{\mathbf{v}}_t$ is the 3×3 skew-symmetric matrix form of $\mathbf{v}_t$, and the second $\hat{\omega}_t$ on the bottom right is the same as the top left.

This formulation is for the prediction step of an EKF for 3D localization, where the pose is represented by a 4×4 homogeneous transformation matrix and the control input is a twist in $\mathbb{R}^6$. The prediction step advances the state estimate and updates its covariance, preparing for the next measurement update in the SLAM process.

C. Mapping

We can formulate Mapping like this. Prior mean of the map's features $\mu_t \in \mathbb{R}^{3M}$ and covariance matrix $\Sigma_t \in \mathbb{R}^{3M \times 3M}$. Stereo calibration matrix K_s, extrinsics $oT_l \in SE(3)$, and IMU pose T_{t+1}. New observations $z_{t+1} \in \mathbb{R}^{4N_{t+1}}$.

Mapping Update Process:

Predicted Observation: For each landmark i at time $t+1$, compute the predicted observation $\hat{z}_{t+1,i}$:

$$\hat{z}_{t+1,i} = K_s \pi(oT_l T_{t+1}^{-1} \mu_{t,j})$$

with $\mu_{t,j}$ representing the state estimate for the landmark corresponding to the i-th observation.

Jacobian Calculation: The Jacobian $H_{t+1,i,j}$ of the predicted observation $\hat{z}_{t+1,i}$ with respect to the j-th map feature m_j is:

$$H_{t+1,i,j} = \begin{cases} K_s \frac{d\pi}{dq} \left(oT_l T_{t+1}^{-1} \mu_{t,j} \right) oT_l T_{t+1}^{-1} P^T & \text{if } \Delta_{t+1}(j) = i, \\ 0 & \text{otherwise.} \end{cases}$$

EKF Update: Update the map estimates using the Kalman gain K_{t+1}, the actual observations z_{t+1}, and the predicted observations $\hat{z}_{t+1}$:

$$K_{t+1} = \Sigma_t H_{t+1}^T \left(H_{t+1} \Sigma_t H_{t+1}^T + I \otimes V \right)^{-1}$$
$$\mu_{t+1} = \mu_t + K_{t+1} \left(z_{t+1} - \hat{z}_{t+1} \right)$$
$$\Sigma_{t+1} = \left(I - K_{t+1} H_{t+1} \right) \Sigma_t$$

where V represents the measurement noise covariance, I is the identity matrix, and $\otimes$ denotes the Kronecker product. H_{t+1} is assembled from the individual $H_{t+1,i,j}$ Jacobians for each observed landmark.

This process refines the map's feature estimates in the robot's environment based on new visual observations, improving the map's accuracy as the robot gathers more data over time.

D. SLAM

SLAM combines both the localization process and the mapping. Given the trajectory prediction and landmark update processes, we need to define a prediction step for landmark positions and an update step for pose estimation.

1) Landmark Prediction: In the problem of triangulating the position of a landmark from two cameras, the objective is to determine the 3D coordinates of a point m in the reference frame of the first camera, using the pixel coordinates z_1 and z_2 from both cameras, along with the known relative rotation R and translation p between the cameras.

To find m, we solve for the unknown depth λ_1 from the first camera using the pixel coordinates and the relative camera transformation. The key equation derived for m is:

$$m = \frac{a^T a}{a^T b} z_1$$

where a and b are given by:

$$a = R^T z_1 - e_3^T R^T z_1 z_2$$
$$b = R^T p - e_3^T R^T p z_2$$

These equations relate the pixel coordinates in each camera's image plane to the 3D point m by solving for the scale factor λ_1, which allows us to back-project the 2D coordinates to 3D space. The vectors a and b encapsulate the geometry of the stereo camera setup and the observed pixel coordinates, ultimately enabling the calculation of the landmark's position.

2) Pose Update: The aim is to refine the estimate of the robot's pose $\mu_{t+1|t}$ in the SE(3) space using new observations and the known correspondences between landmarks and their observed positions in the image frame. The process is underpinned by a Gaussian prior with mean $\mu_{t+1|t}$ and covariance $\Sigma_{t+1|t}$.

Prior: Gaussian with mean $\mu_{t+1|t} \in SE(3)$ and covariance $\Sigma_{t+1|t} \in \mathbb{R}^{6\times6}$. Known quantities: Stereo calibration matrix K_s, extrinsics $oT_l \in SE(3)$, and the positions of landmarks $m \in \mathbb{R}^{3M}$. New observations $z_{t+1} \in \mathbb{R}^{4N_{t+1}}$.

Process:

1. **Predicted Observation**: Generate the predicted observation $\hat{z}_{t+1,i}$ for each landmark i using the transformation between the robot's pose and the landmarks:

$$\hat{z}_{t+1,i} = K_s \pi(oT_l \mu_{t+1|t}^{-1} m_i)$$

where π is the projection function from 3D space to the camera's image plane, and m_i is the position of the i-th landmark.

2. **Jacobian Calculation**: Compute the Jacobian $H_{t+1,i}$ of the predicted observation with respect to the robot's pose at $\mu_{t+1|t}$:

$$H_{t+1,i} = -K_s \frac{d\pi}{dq} (oT_l \mu_{t+1|t}^{-1} m_i) oT_l (\mu_{t+1|t}^{-1} m_i) \emptyset$$

The $\emptyset$ operation represents the operation on slide 26 of lecture 12. I couldn't figure out how to write it in latex.

3. **EKF Update**: Perform the update step to refine the robot's pose estimate:

$$K_{t+1} = \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + I \otimes V)^{-1}$$

$$\mu_{t+1|t+1} = \mu_{t+1|t} \exp((K_{t+1}(z_{t+1} - \hat{z}_{t+1}))^\wedge)$$

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

Here, V is the measurement noise covariance, I is the identity matrix, and $\otimes$ is the Kronecker product. The $\exp$ and $\wedge$ operations convert the twist vector into the corresponding matrix in SE(3). This process uses the new observations to update the estimate of the robot's pose, accounting for the motion and sensor measurement uncertainties. The resulting $\mu_{t+1|t+1}$ and $\Sigma_{t+1|t+1}$ represent the updated mean and covariance of the pose in SE(3) space, reflecting a more accurate estimate given the new data.

III. Technical Approach

A. IMU Localization via EKF Prediction

In IMU Localization via the Extended Kalman Filter (EKF) Prediction, we develop a motion model to predict the robot's pose and uncertainty over time based on inertial measurement unit (IMU) data. The prediction phase utilizes linear and angular velocities (v_t and ω_t) from the IMU to estimate the robot's motion. This process involves two key steps:

1. **State Prediction**: The robot's pose is updated using the exponential map. We compute a matrix ζ incorporating the angular velocity ω_t into a skew-symmetric matrix and the linear velocity v_t into its last column. The new pose, $\mathbf{T}_{t+1|t}$, is calculated by multiplying the current pose, $\mathbf{T}_{t|t}$, with the exponential of ζ times the timestep (τ_t), representing the robot's movement over the interval.

2. **Covariance Prediction**: The uncertainty in the robot's pose is updated to reflect the prediction's imprecision and the inherent noise in the IMU measurements. This is done by updating the covariance matrix, $\Sigma_{t+1|t}$, using the matrix exponential of a constructed matrix u_{hat} that combines the angular and linear velocities, and then adding process noise W, which accounts for the uncertainty in the motion model. The initial pose, $\mathbf{T}_{t|t}$, and covariance, $\Sigma_{t|t}$, are set to an identity matrix and a zero matrix, respectively, indicating the start of the prediction process with no prior movement. The process iteratively updates the robot's predicted state and the associated covariance for each timestep based on the IMU data, producing a trajectory that reflects both the estimated path and the uncertainty around it.

B. Landmark Mapping via EKF Update

In the landmark mapping process of Visual-Inertial SLAM, we use stereo camera images to triangulate and update the positions of landmarks in the environment. This process is outlined in steps as follows:

1. **Initialization**: Landmark positions are initialized to zero in a global array meant to hold their 3D coordinates.

2. **Observation Processing**: At each time step, we filter valid observations from the stereo images, discarding any that don't meet our criteria for valid landmarks.

3. **Triangulation**: For each valid observation, we convert pixel coordinates to camera coordinates using the inverse of the camera's intrinsic matrix K. We then triangulate the 3D position of each landmark by solving for its position based on the geometry of the stereo setup (given by the baseline b and the relative orientation and position of the two cameras).

4. **World Frame Transformation**: The 3D positions are transformed from the camera frame to the world frame using the robot's pose estimated from IMU data and the extrinsic calibration between the IMU and the camera.

5. **Map Update**: The global map of landmark positions is updated with these newly calculated positions, refining the map's accuracy over time.

6. **Filtering**: Implausible landmark positions (such as ones that give a negative z value in optical frame or if the point is to far from the robot i.e. the norm of the position is greater than 500m) are filtered out to maintain the integrity of the map.

C. Visual-inertial SLAM

Note: the noise value for the plots is, diag([1e-4,1e-4,1e-4,1e-8,1e-8,1e-8]) for data set 10 and diag([2e-4,2e-4,2e-4,5e-8,5e-8,5e-8]) for dataset 03.

In the technical approach for Visual-Inertial SLAM (VI-SLAM), we integrate the processes of IMU localization and landmark mapping to achieve simultaneous localization and mapping. This approach involves updating both the robot's trajectory and the map of the environment using visual and inertial data. The key steps in the VI-SLAM process, as illustrated by the provided code snippet, are detailed below:

1. **Initialization**: The covariance matrices for landmarks (σ_{LL}) and the robot's pose (σ_{RR}), along with their cross-covariance (σ_{LR} and σ_{RL}), are initialized to reflect initial uncertainties. - The mean state vector (MU) is initialized, combining the landmarks' positions (μ_L) and the robot's pose (μ_R).

2. **Motion Model Update**: - For each time step, the robot's pose is updated using the motion model derived from IMU data (linear and angular velocities). This update affects the pose part of the covariance matrix (σ_{RR}), as well as its cross-covariance with the landmarks' positions (σ_{LR} and σ_{RL}). - The trajectory of the robot (T) is updated accordingly.

3. **Landmark Observation and Update**: - At each time step, valid visual observations of landmarks are identified from the stereo images. For each observed landmark, its position is estimated or updated based on triangulation from the stereo image data. - A Jacobian matrix (H) for the observation model is constructed, relating the changes in observed landmark positions to changes in the landmarks' states and the robot's pose.

4. **EKF Update**: - The Extended Kalman Filter (EKF) update step is performed using the observation model. This involves calculating the Kalman gain (K_{gain}), updating the mean state vector (MU), and refining the combined covariance matrix ($SIGMA$). - This update step adjusts both the landmarks' estimated positions and the robot's pose based on the new observations, integrating visual and inertial data to enhance SLAM accuracy.

5. **Trajectory Correction**: - Corrections to the robot's trajectory are applied based on the outcomes of the EKF

update. This involves updating the robot's pose (T) using the corrected state estimates, ensuring the trajectory aligns with both the inertial measurements and the observed landmark positions.

6. **Covariance Matrix Refinement**: - Following the EKF update, the covariance matrix ($SIGMA$) is refined to reflect the updated uncertainties in both the landmarks' positions and the robot's pose. This refinement accounts for the new information gained from the visual observations and adjusts the SLAM process's overall uncertainty.

The VI-SLAM process iteratively refines the map of the environment and the robot's trajectory by continuously integrating and updating based on new inertial and visual data.

IV. Results

The plots for the trajectory estimate, landmark locations, and the SLAM for both datasets are shown in figures 1 through 6. Figure 1 shows our initial trajectory estimate using only the predict portion of EKF. Then in figure 2 we can see the landmark initializations versus the updated locations after running EKF overlaid on our trajectory estimate in part 1. We can see that the locations have changed after running EKF. The degree to which the points change depends on the amount of noise we assume in our sensor. Finally in figure 3 we can see the result of the visual inertial SLAM with before and after for both trajectory and landmark locations. This part is also very sensitive to the noise you assume for the sensor. In Figure 3 we see that the trajectory deviates more when we select a larger W value than the one in Figure 6. In Figure 6 the noise is set very low and thus very little deviation for the trajectory (there is still some deviation in both trajectory and landmark location).

V. Conclusion

In conclusion, the comprehensive exploration of visual-inertial simultaneous localization and mapping (VI-SLAM) within this study demonstrates importance of sensing and estimation. By integrating data from an inertial measurement unit (IMU) and a stereo camera, and leveraging the strengths of the extended Kalman filter (EKF), this project successfully achieved accurate localization and mapping in an unknown environment. The results validate the efficacy of the approach in handling the complexities of VI-SLAM. Notably, the sensitivity of the SLAM process to assumed sensor noise underscores the importance of precise sensor modeling and calibration. The implementation showcases not only the potential of EKF-based VI-SLAM in enhancing autonomous robotic navigation but also highlights areas for further research, particularly in optimizing sensor noise models to improve the robustness and accuracy of localization and mapping.

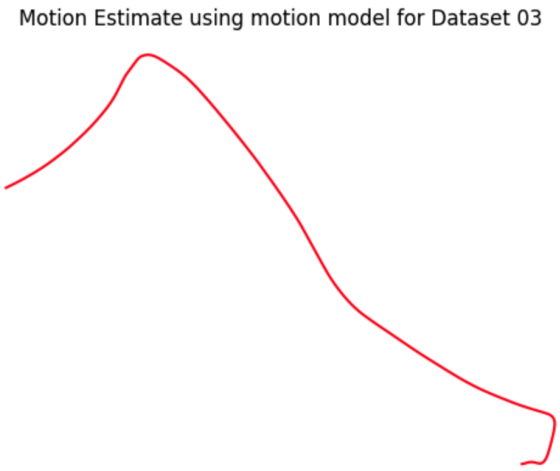

Fig. 1. Trajectory estimate for dataset 03

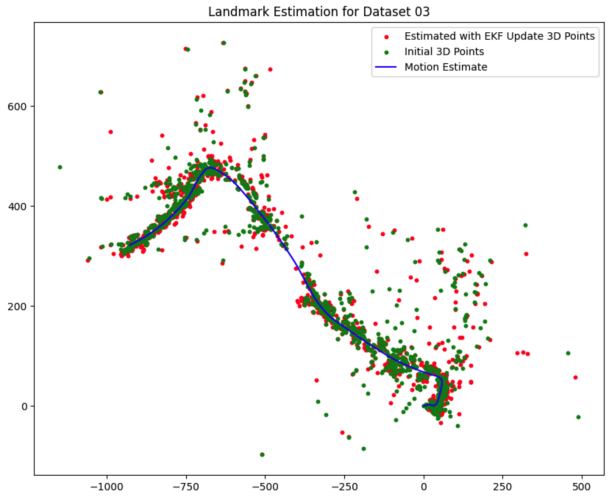

Fig. 2. landmark estimate for dataset 03

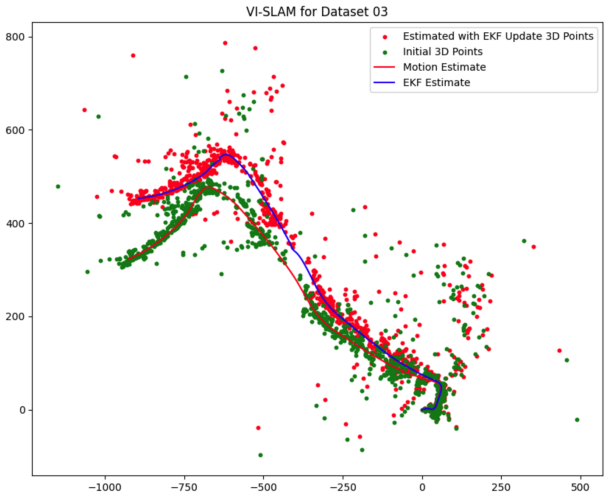

Fig. 3. VI-SLAM for dataset 03

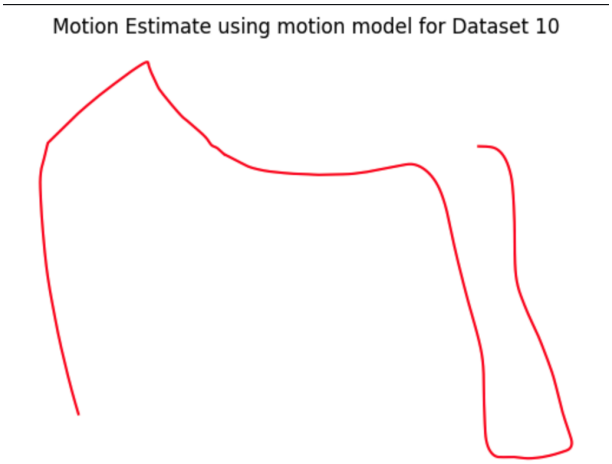

Fig. 4. Trajectory estimate for dataset 10

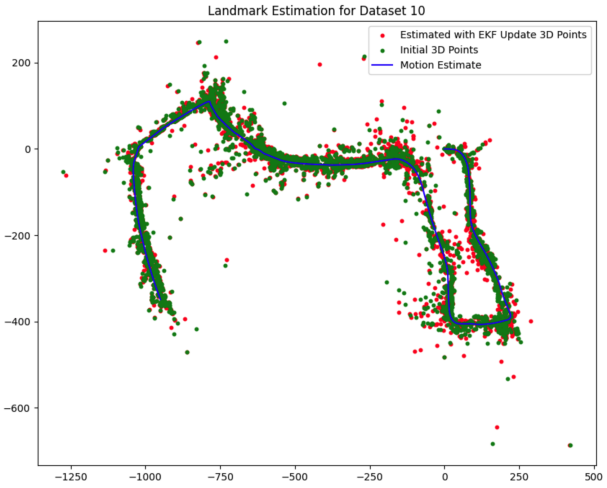

Fig. 5. landmark estimate for dataset 10

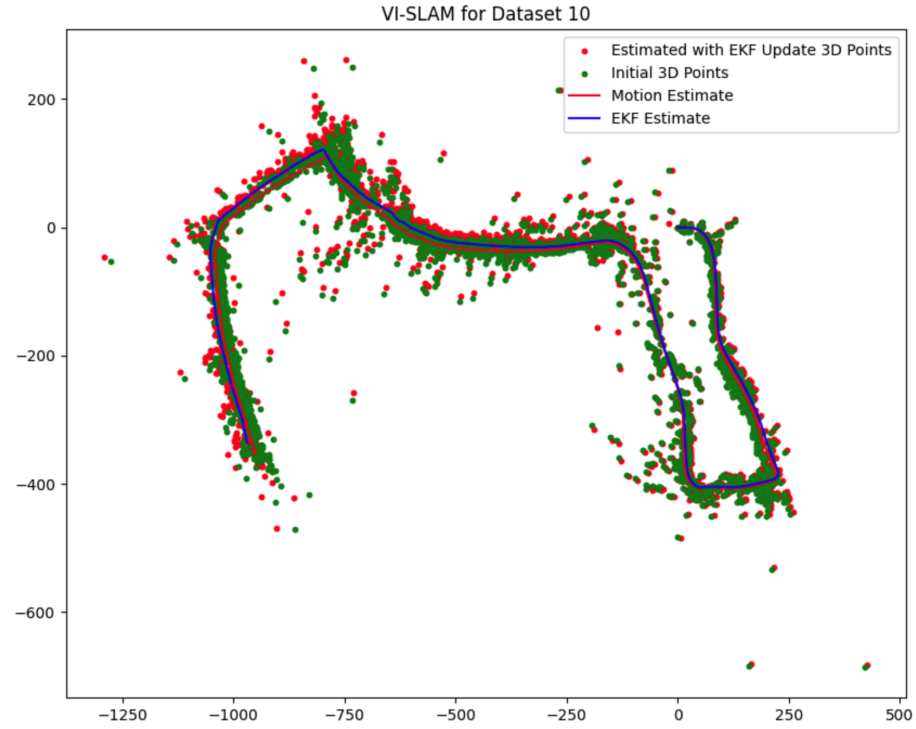

Fig. 6. VI-SLAM for dataset 10

REFERENCES

[1] https://en.wikipedia.org/wiki/Spherical coordinate system
[2] https://natanaso.github.io/ece276a/

CONTENTS