Project 3: Visual-Inertial SLAM

1st Gaopo Huang
UCSD ECE276A
ghuang@ucsd.edu

***Abstract*—I implemented visual-inertial simultaneous localization and mapping (SLAM) using an extended Kalman filter (EKF) in this project. Using synchronized measurements from an inertial measurement unit (IMU) and visual landmark features extracted from a stereo camera provided with its intrinsic and extrinsic calibration equipped on a vehicle, I estimated its trajectory and reconstructed the landmarks of its surrounding environment via two approaches. I first solved this problem by separating out the localization and visual mapping tasks and solving them individually to obtain a dead reckoning result, then compared this result with EKF approach which considers the correlation between the landmarks and the vehicle.**

***Index Terms*—Visual-inertial SLAM, Extended Kalman filter, Pose kinematics**

I. Introduction

The Simultaneous Localization and Mapping (SLAM) problem has been one of the most popular research areas from its coinage. With the breakthrough of robotics and the usage of many related smart devices and observation sensors, the problem of accurately locating the device and building a real-time map of its surrounding environment becomes a popular subject with dense literature in finding a way to best utilize all the collected data and recover the trajectory and the map. Among many proposed solutions, visual-inertial SLAM is one approach that takes advantage of IMU and stereo camera measurements to robustly perform this task. In this project, we discuss how to implement visual-inertial SLAM on a moving vehicle equipped with the camera and IMU.

To give a quick overview, the visual features are extracted from the stereo camera whose intrinsic and extrinsic calibration is given along with the IMU measurement data. We performed Extended Kalman Filter (EKF) with the synchronized data to solve this SLAM problem. At each step, we use the motion model inputted with the IMU sensor data to predict the vehicle location, then we update it factoring in its covariance with the landmark locations estimated from the visual features via the stereo camera model. Following these steps gives real-time vehicle localization and environment mapping.

II. Problem Formulation

A. Motion model

Given the IMU measurement data, linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and rotational velocity $\omega_t \in \mathbb{R}^3$, along with the noise $\mathbf{w}_t \in \mathbb{R}^6$, where the noise $\mathbf{w}_t$ is the stacked noise $[\mathbf{w}_{v,t}^T, \mathbf{w}_{\omega,t}^T]^T$, and the previous IMU pose $T_t \in SE(3)$, assuming that the velocity and the noise is constant within each time step, the IMU pose at the next step can be expressed with the motion model as

$$T_{t+1} = f(T_t, \tau_t, \mathbf{v}_t, \boldsymbol{\omega}_t, \mathbf{w}_t) = T_t \exp(\tau_t(\boldsymbol{\zeta}_t + \mathbf{w}_t)^\wedge) \quad (1)$$

where $\boldsymbol{\zeta}_t \in \mathbb{R}^6$ is the generalized velocity $[\mathbf{v}_t^T, \boldsymbol{\omega}_t^T]^T$, the hat map $\hat{}$ for a $\mathbb{R}^6$ vector is

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

, and $\exp(\cdot)$ is the exponential function for matrix.

B. Observation model

Given the current IMU pose $T_t = {}_WT_{T,t} \in SE(3)$, the world coordinate $\mathbf{m}_i$, the measurement noise $\mathbf{v}_{t,i}$, the intrinsic calibration matrix K_s for the stereo camera, and the relative transformation between camera and IMU ${}_OT_I \in SE(3)$, the observed pixel value at two frames stacked together as $\mathbf{z}_{t,i}$ for this landmark is:

$$\mathbf{z}_{t,i} = h(T_t, \mathbf{m}_i, \mathbf{v}_{t,i}, K_s, {}_OT_I) = K_s \pi({}_OT_I T_t^{-1} \underline{\mathbf{m}}_i) + \mathbf{v_{t,i}} \quad (2)$$

, where $\pi(\mathbf{q}) = \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4$ is the projection function, and $\mathrm{m}_i = \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix}$ is the homogeneous coordinate.

C. Localization

Given the linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3 \forall t$, estimate the IMU trajectory $T_t := {}_WT_{I,t} \in SE(3) \forall t$ using the motion model as described in Eq (1). This is the formulation for the dead reckoning trajectory when no observation is received.

D. Visual mapping

Given the IMU pose, $T_t := {}_WT_{I,t} \in SE(3)$, the observations of the visual features $\mathbf{z}_t := [\mathbf{z}_{t,1}^T, \ldots, \mathbf{z}_t{}_{t,N_T}^T]^T \in \mathbb{R}^{4N_t} \forall t$, assuming that the data association $\Delta_t : \{1, \ldots, M\} \to \{1, \ldots, N_t\}$ describing the correspondence between the observation $\mathbf{z}_{t,i} \in \mathbb{R}^4$ with the corresponding landmark $\mathbf{m}_j$, $i = \Delta_t(j)$ that generated them at time t is known, and that these landmarks $\mathbf{m}$ are static, then estimate the world coordinates $\mathbf{m} := [\mathbf{m}_1^T, \ldots \mathbf{m}_M^T]^T \in \mathbb{R}^{3M}$ of the landmarks.

E. Visual-inertial SLAM

a) Visual-inertial odometry: Given the linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3 \forall t$, the known world-frame landmark coordinates $\mathbf{m} \in \mathbb{R}^{3M}$, the visual feature observations $\mathbf{z}_{0:T}$ and the data association $\Delta_t : \{1, \ldots, M\} \to \{1, \ldots, N_t\}$ corresponds the landmark j and their observation $\mathbf{z}_{t,i} \in \mathbb{R}^4$ with $i = \Delta_t(j)$ known, estimate the IMU poses $T_t := {}_WT_{I,t} \in SE(3) \forall t = 1, \ldots, T$

b) SLAM: Combined visual-inertial odometry and mapping problem, that is, given the IMU measurement $\mathbf{v}_t$ and $\boldsymbol{\omega}_t \in \mathbb{R}^3$, the features $\mathbf{z}_{t,i} \in \mathbb{R}^4 \forall i = 1, \dots, N_t$, and the corresponding intrinsic and extrinsic stereo camera calibratoin K_s and ${}_O T_I \in SE(3)$, estimate the world-frame IMU pose ${}_W T_{I,t} \in SE(3) \forall t$ over time, and the world-frame coordinates $\mathbf{m}_j \in \mathbb{R}^3 \forall j = 1, \dots, M$ of all landmarks that generated the visual features $\mathbf{z}_{t,i} \in \mathbb{R}^4$.

III. Technical Approach

A. Extended Kalman Filters

A common approach to solve SLAM problem is to predict the highest posterior probability of the location of the subject and its surroundings given the previous control and observations, which is known as Bayes filter. To make the computation actually feasible, Kalman filter is makes additional assumptions that the prior $\Sigma_{t|t}$, motion noise $\mathbf{w}_t$, and observation noises $\mathbf{v}_t$ are all independent Gaussians, and that the motion model and observation model are linear in the state $\mathbf{x}_t$ with the respective noises. However, as Eq (1) and Eq(2) shows, the problem that we are addressing has nonlinear motion and observation models. Therefore, with the linearity assumption violated, we use EKF to tackle this problem.

EKF uses a first-order Taylor series approximation to the motion and observation models around the state and noise means so that we will have a linear approximation to the model and that the rest will follow the regular Kalman filter approach.

Let the motion model be $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$ with zero mean Gaussian noise $\mathbf{w}_t \sim \mathcal{N}(0, W)$, then the prediction step should be

$$\begin{aligned} \boldsymbol{\mu}_{t+1|t} &= f(\boldsymbol{\mu}_{t|t}, \boldsymbol{u}_t, \mathbf{0}) \\ \Sigma_{t+1|t} &= F_t \Sigma_{t+1|t} F_t^T + Q_t W Q_t^T \end{aligned} \tag{3}$$

, where $F_t = \frac{df}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})$, and $Q_t = \frac{df}{d\mathbf{w}}(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0})$.

Let the observation model be $\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t)$ with zero mean Gaussian noise $\mathbf{v}_t \sim \mathcal{N}(0, V)$, then the Kalman gain can be computed as

$$K_{t+1|t} = \Sigma_{t+1|t} H_{t+1}^T (H_{t+1} \Sigma_{t+1|t} H_{t+1}^T + R_{t+1} V R_{t+1}^T)^{-1} \tag{4}$$

, where $H_t = \frac{dh}{d\mathbf{x}}(\boldsymbol{\mu}_{t|t-1}, \mathbf{0})$, and $R_t = \frac{dh}{d\mathbf{v}}(\boldsymbol{\mu}_{t|t-1}, \mathbf{0})$. The updated mean and covariance can then be computed as,

$$\begin{aligned} \boldsymbol{\mu}_{t+1|t+1} &= \boldsymbol{\mu}_{t+1|t} + K_{t+1|t}(z_{t+1} - h(\boldsymbol{\mu}_{t+1|t}, 0)) \\ \Sigma_{t+1|t+1} &= (I - K_{t+1|t} H_{t+1}) \Sigma_{t+1|t} \end{aligned} \tag{5}$$

B. IMU localization via EKF prediction

For each prediction step, the IMU location will be computed with $SE(3)$ kinematics equation, which is described by the motion model in Eq(1). Assume that the noise $\mathbf{w}_t \sim \mathcal{N}(0, W)$. Since the noise is a zero mean vector, the mean will just be $\boldsymbol{\mu}_{t+1|t} = \boldsymbol{\mu}_{t|t} \exp(\tau_t \hat{\mathbf{u}}_t)$, where $\hat{\mathbf{u}}_t$ is the hat map of the generalized velocity as described when defining the motion model. Since $T \in SE(3)$, the Gaussian distribution perturbation $\delta\boldsymbol{\mu}$ has to be defined over Lie algebra as $T = \boldsymbol{\mu} \exp(\hat{\delta\boldsymbol{\mu}}) \approx \boldsymbol{\mu}(I + \hat{\delta\boldsymbol{\mu}})$. The gradient of T can be computed with perturbation, and the perturbation would be $\delta\boldsymbol{\mu}_{t+1} = \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t) \delta\boldsymbol{\mu}_t + \mathbf{w}_t$. Therefore, for the covariance prediction in Eq(3), $F_t = \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t), Q_t = I$, and the prediction step equations are:

$$\begin{aligned} \boldsymbol{\mu}_{t+1|t} &= \boldsymbol{\mu}_{t|t} \exp(\tau_t \hat{\mathbf{u}}_t) \\ \Sigma_{t+1|t} &= \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t) \Sigma_{t|t} \exp(-\tau_t \overset{\curlywedge}{\mathbf{u}}_t)^T + W \end{aligned} \tag{6}$$

C. Landmark mapping via EKF update

Consider only mapping the landmark $\mathbf{m}_j$, since these landmarks are static, we only need to implement the EKF update step. Using the observation model described in Eq (2), we need to compute its derivative and Q for the EKF update equation. At each step, we stack all observed features as $4N_t$ vector, then $\mathbf{z_t} = K_s \pi({}_O T_I T_t^{-1} \underline{\mathbf{m}}_i) + \mathbf{v_t}$, where $v_t \sim \mathcal{N}(0, I \bigotimes V)$ Using chain rule, we can compute the jacobian for π as

$$\frac{d\pi}{d\mathbf{q}}(\mathbf{q}) = \frac{1}{q_3} \begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \tag{7}$$

After computing the full derivative,

$$H_{t+1} = K_s \frac{d\pi}{d\mathbf{q}}({}_O T_I T_{t+1}^{-1} \underline{\boldsymbol{\mu}}_t) {}_O T_I T_{t+1}^{-1} P^T$$

for corresponding landmark and visual features $\Delta_t(j) = i$ and 0 otherwise. The predicted observation should be

$$\overset{\sim}{\mathbf{z}}_{t+1,i} = K_s \pi({}_O T_I T_{t+1}^{-1} \underline{\boldsymbol{\mu}}_t), \forall i$$

In addition, in implementation, some visual features observed can be outliers that are extremely far away from the observation point and only cause drift in the update. These outliers should be removed. The simplest way that we implemented is to simply decide a threshold ϵ that rejects any observation point $\overset{\sim}{\mathbf{z}}_{t+1,i} > \epsilon$ before computing the update matrices.

The update step for these landmarks just follow Eq(4) and Eq(5) with $h(\boldsymbol{\mu}_{t+1|t}, 0) = \overset{\sim}{\mathbf{z}}_{\mathbf{t+1}}$ and $R = I$.

D. Visual-inertial SLAM

Visual-inertial SLAM is then combining the prediction step and the landmark update step from the previous sections while adding an update step for the IMU pose $T_t \in SE(3)$ based on the stereo-camera observation model and the landmark coordinates.

The prediction step is exactly the same as before except that the covariance of the landmark and the IMU pose needs to updated with $\Sigma_{RL} = F \Sigma_{RL}$, where Σ_{RL} represents the cross-covariance between the robot and landmarks.

For the update step, Σ_{RL} also needs to be updated, using pose perturbation $\delta\boldsymbol{\mu}$ to derive the jacobian, we can obtain

$$H_{t+1,i} = -K_s \frac{d\pi}{d\mathbf{q}}({}_O T_I \boldsymbol{\mu}_{t+1|t}^T \underline{\mathbf{m}_j}) {}_O T_I (\boldsymbol{\mu}_{t+1|t}^T \underline{\mathbf{m}_j})^{\odot} \in \mathbb{R}^{4 \times 6}$$

,where

$$\begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} = \begin{bmatrix} I & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 6}$$

Stacking all H_i for a given time step t, we can get $H_{t+1} \in \mathbb{R}^{4N_t \times 6}$, which corresponds to H_{RL}. Stacking $[H_{LL}, H_{RL}]$ to obtain H_{SLAM} and we can calculate the overall Kalman gain by plugging it into Eq(4) and obtain $K_{t+1} = \begin{bmatrix} K_{LL} \\ K_{RL} \end{bmatrix} \in \mathbb{R}^{6 \times 4N_t}$, where K_{LL} is the kalman gain for the landmarks, and K_{RL} is the kalman gain for the robot given the landmarks. We can then use K_{LL} to compute the EKF update step for the landmarks exactly as in the last section with visual mapping alone. The covariance can be updated altogether with

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

to keep track all of $\Sigma_{RL}, \Sigma_{RR,LL}$ in one covariance matrix $\Sigma_{SLAM} \in \mathbb{R}^{3M \times 3M}$.

However, for the robot mean, since we are operating in the $SE(3)$ space, the mean update needs to use the adjoint with

$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} \exp((K_{t+1}(\mathbf{z}_{t+1} - \widetilde{\mathbf{z}}_{t+1}))^\wedge)$$

IV. Results

A. IMU localization via EKF prediction

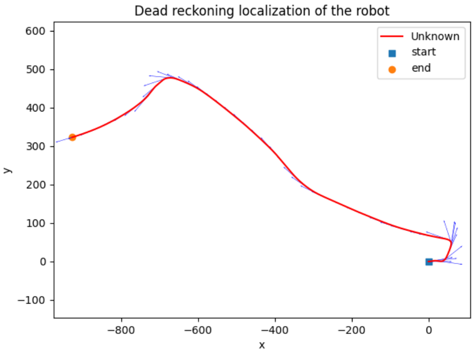

Fig. 1. Dataset 03 dead reckoning trajectory IMU localization

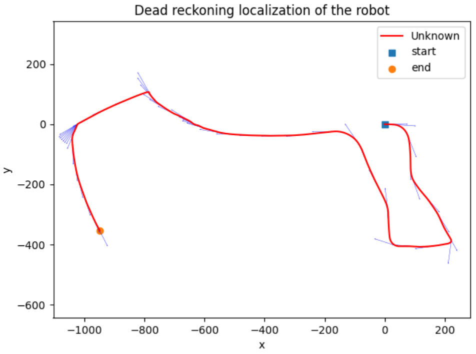

Fig. 2. Dataset 10 dead reckoning trajectory IMU localization

The EKF prediction steps performed on the IMU measurements alone. The result is relatively stable but there is no way to observe its accuracy.

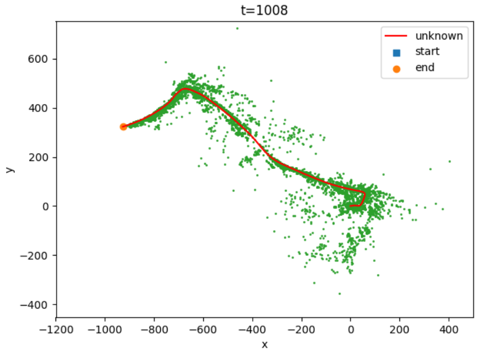

Fig. 3. Dataset 03 dead reckoning landmark mapping

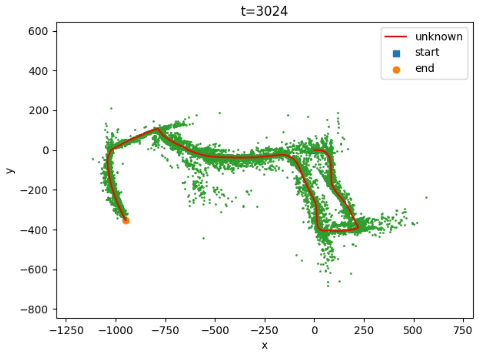

Fig. 4. Dataset 10 dead reckoning landmark mapping

B. Landmark mapping via EKF update

The EKF update step performed on the visual features to map the world coordinates of the landmarks that generated these visual features. Note that the outlier observations have been removed from this result with a distance threshold of 1000 meters from the observation point. From observation, there are still observed landmarks that are relatively far away but more than 99% of the observed points are valid landmarks close to the IMU trajectory.

C. Visual-Inertial SLAM

a) Final result: Combining the prediction step and update step in visual-inertial SLAM with both the vehicle and the landmarks, the covariance between them are captured and thus the trajectory and the landmarks will have a more accurate representation comparing to the dead trajectory result. In this result, the figure, we used noise W in motion model with linear velocity of 1e-2, angular velocity of 1e-5, the noise V in observation model of 5 pixel unit. The landmark covariance is initialized to 2.0 since we are not certain of the precision of landmarks. As the result suggests, the overall trajectory drifts a bit, particularly to where large magnitude of rotation occurs. The covariance of landmarks more accurately correct the vehicle trajectories.

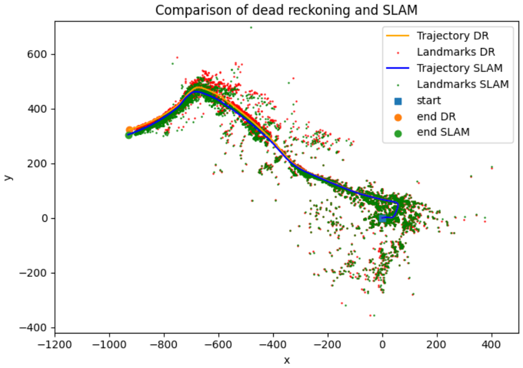

Fig. 5. Dataset 03 Visual-inertial SLAM vs. Dead reckoning

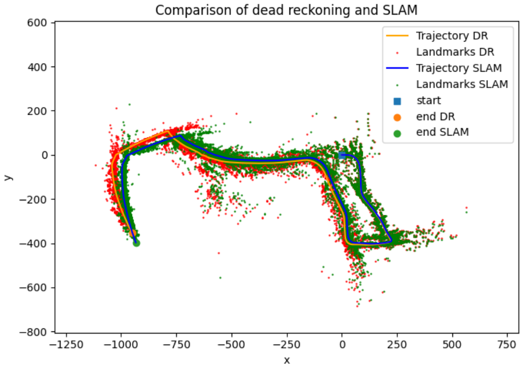

Fig. 6. Dataset 10 Visual-inertial SLAM vs. Dead reckoning

b) Noise tuning: I also tuned the noise with several configuration. Specifically, noise W in motion model with [1, 1e-2] in linear velocity, [1e-2, 1e-5] in angular velocity; noise V in observation with $[2, 5]$, and initial landmark covariance [0.5, 2.0]. As the result shows, the noise tuning can be very effective in tuning the trajectory and landmark locations as well. This should depend on whether we trust IMU measurement or the camera measurement in field.

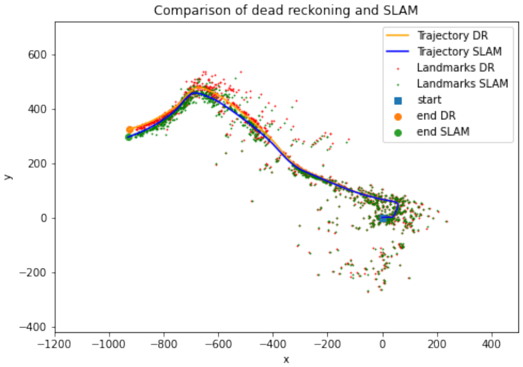

Fig. 7. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 2, \Sigma_{LL} = 0.5$

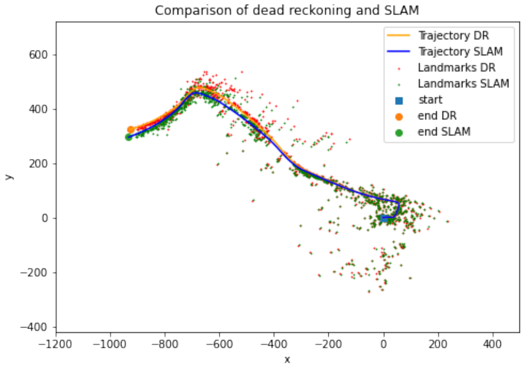

Fig. 8. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 2, \Sigma_{LL} = 2.0$

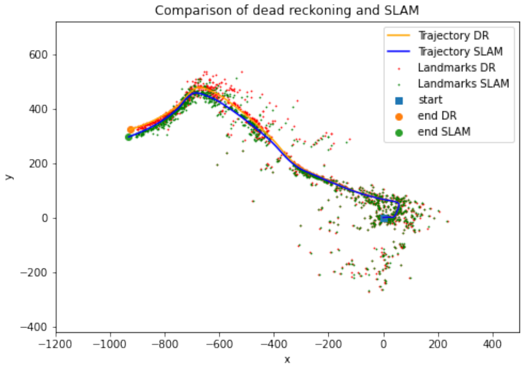

Fig. 9. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 5, \Sigma_{LL} = 0.5$

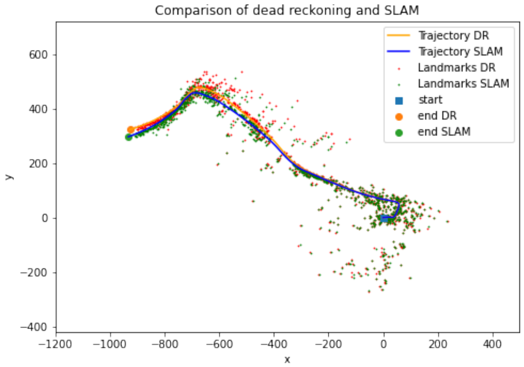

Fig. 10. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 5, \Sigma_{LL} = 2.0$

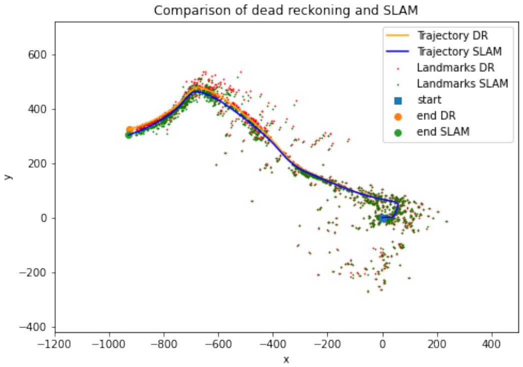

Fig. 11. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 2, \Sigma_{LL} = 0.5$

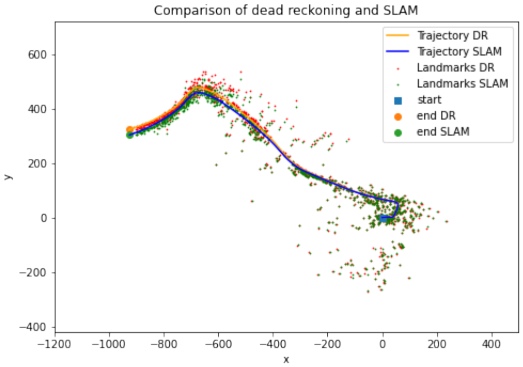

Fig. 12. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 2, \Sigma_{LL} = 2.0$

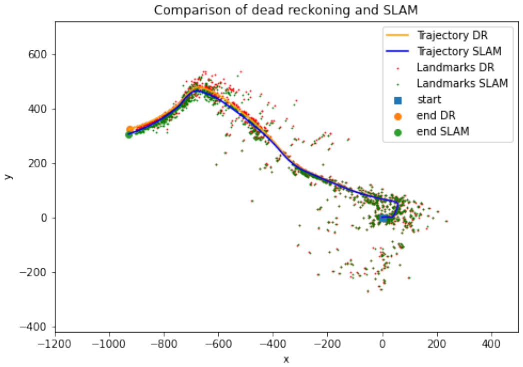

Fig. 13. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 5, \Sigma_{LL} = 0.5$

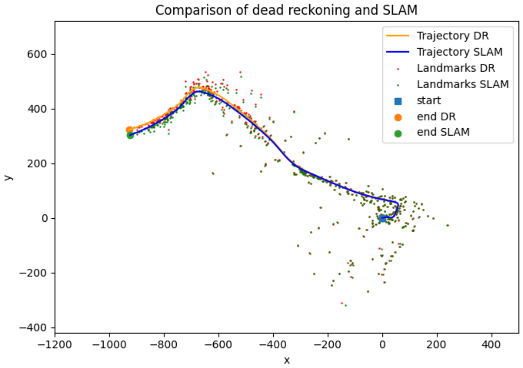

Fig. 14. Dataset 03 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 5, \Sigma_{LL} = 2.0$

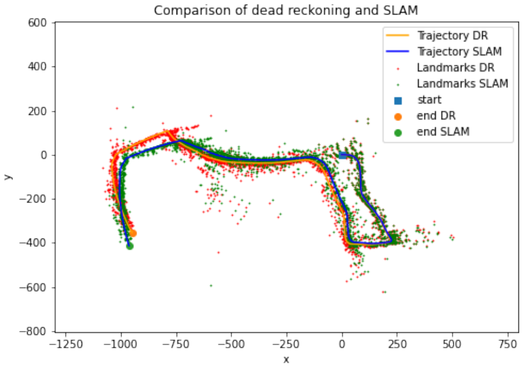

Fig. 15. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 2, \Sigma_{LL} = 0.5$

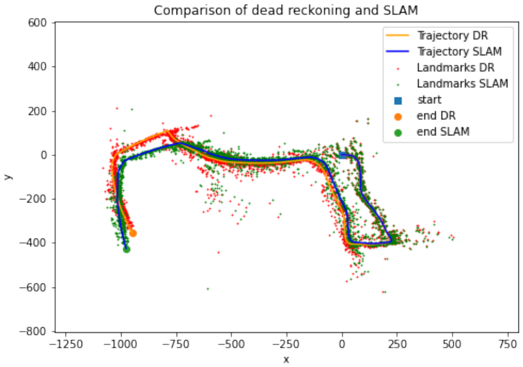

Fig. 16. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 2, \Sigma_{LL} = 2.0$

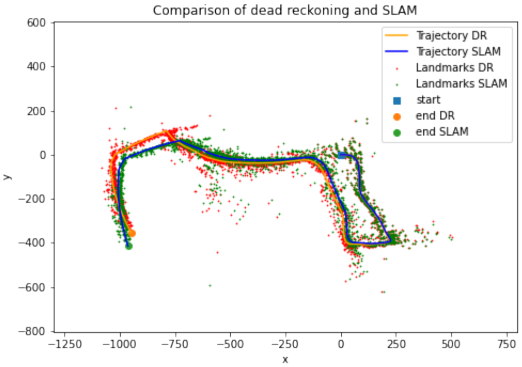

Fig. 17. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 5, \Sigma_{LL} = 0.5$

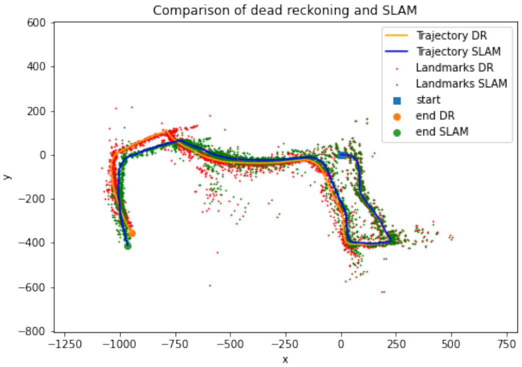

Fig. 18. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1, W_\omega = 1e-2, V = 5, \Sigma_{LL} = 2.0$

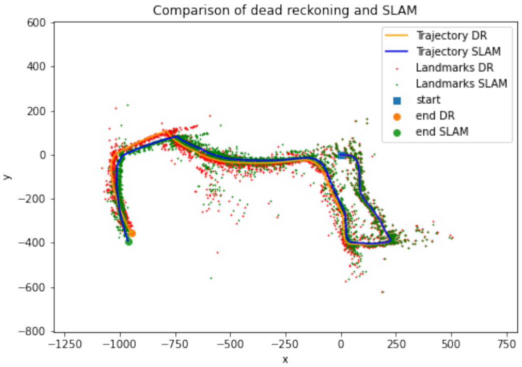

Fig. 19. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 2, \Sigma_{LL} = 0.5$

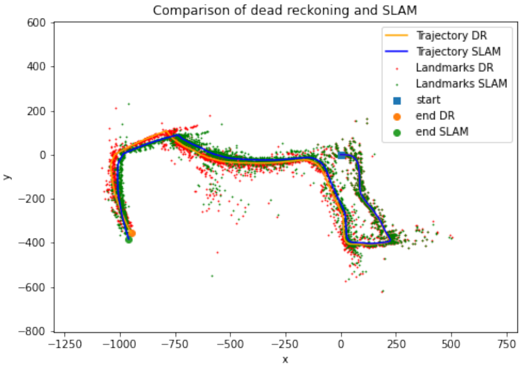

Fig. 20. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 2, \Sigma_{LL} = 2.0$

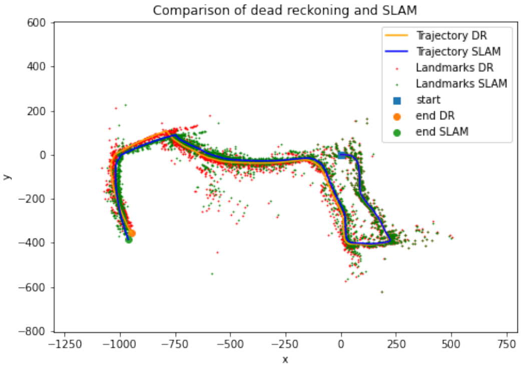

Fig. 21. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 5, \Sigma_{LL} = 0.5$

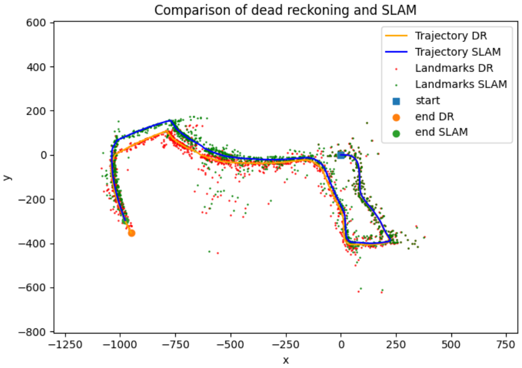

Fig. 22. Dataset 10 Visual-inertial SLAM vs. Dead reckoning with $W_v = 1e-2, W_\omega = 1e-5, V = 5, \Sigma_{LL} = 2.0$