Visual Inertial SLAM with Extended Kalman Filter

1st Zhirui Dai
dept. Electrical and Computer Engineering

UC San Diego
San Diego, United States
zhdai@ucsd.edu

I. Introduction

To enable a robot's autonomous ability, SLAM is one of the most important puzzels to the blueprint. Beginning with completely unknown environment, a robot can use some sensors, e.g. Lidar, Camera, etc. to measure the environment and try to move to explore its surroundings. With probability inference theory, the robot can estimate its position and orientation, and the obstacles in the environment by evaluating data from its sensors, which is exactly what SLAM does.

There are different kinds of estimators available for SLAM, such as particle filter, Kalman filter, etc. In this report, I implemented an SLAM algorithm based on Kalman filter. In this report, a Visual-Inertial SLAM using IMU data and features in stereo view is implemented.

II. Problem Formulation

The overall problem is, given evidence of agent actions and observations, how should the agent sense its state, like position and orientation, and its surrounding, like obstacle distribution, accurately and efficiently. In a dynamic system including both the agent and the environment, the agent state and the environment state may be coupled and the problem becomes very complicated. If we treat this problem as a filtering problem, the agent's target is to filter out states with low probability and take the state with the highest probability as the answer to the problem.

A. Filtering Problem with Markov Assumption

We can use Markov assumption to decouple the agent and the environment to simplify this problem,

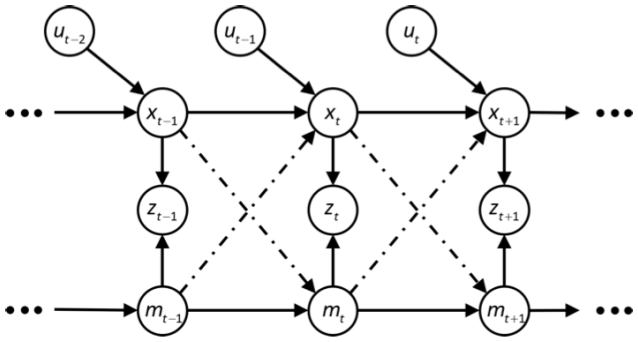

Fig. 1. Graph of Filtering Problem with Markov Assumption, x_t is the robot state, u_t is the control input, z_t is the observation result, and m_t is the environment state.

As figure (1) shows, Markov assumptions simplify the filtering problem to the inference of two sequences of hidden states, $\mathbf{x}_{0:T}$ and $\mathbf{m}_{0:T}$, given the evidence of another two sequential data, $\mathbf{u}_{0:T}$ and $\mathbf{z}_{0:T}$.

The joint distribution is given by

$$\begin{aligned} &p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}, \mathbf{u}_{0:T}) \\ &= p_{0|0}(\mathbf{x}_0) \prod_{t=0}^{\top} p_h(\mathbf{z}_t \mid \mathbf{x}_t) \prod_{t=1}^{\top} p_f(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \end{aligned} \tag{1}$$

B. Bayes Filter

Bayes filter is a probabilistic inference technique for estimating the state of dynamic systems, e.g. the robot pose or the map structure, based on evidences from control inputs and observations using the Markov assumptions and Bayes rule. It is helpful for solving filtering problem. In the theory of Bayesian filter, there are four key elements:

1) Motion Model:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \sim p_f(\cdot \mid \mathbf{x}_t, \mathbf{u}_t) \tag{2}$$

2) Observation Model:

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t) \sim p_h(\cdot \mid \mathbf{x}_t) \tag{3}$$

3) Prediction Step: With the motion model, the prediction step of a Bayes filter is given by

$$\begin{aligned} p_{t+1|t}(\mathbf{x}) &= p(\mathbf{x}_{t+1} \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t}) \\ &= \int p_f(\mathbf{x} \mid \mathbf{s}, \mathbf{u}_t) p_{t|t}(\mathbf{s}) d\mathbf{s} \end{aligned} \tag{4}$$

4) Update Step: With the prediction and the observation model, the agent can correct its prediction by

$$\begin{aligned} p_{t+1|t+1}(\mathbf{x}) &= p(\mathbf{x}_{t+1} \mid \mathbf{z}_{0:t+1}, \mathbf{u}_{0:t}) \\ &= \frac{p_h\left(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}\right) p_{t+1|t}(\mathbf{x}_{t+1})}{p\left(\mathbf{z}_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t}\right)} \end{aligned} \tag{5}$$

C. SLAM

In a pure localization problem, we assume that the environment is known and therefore a Bayes filter can work well. However, in practice, the environment state $\mathbf{m}$ is unknown, which is a mapping problem.

Simultaneous Localization And Mapping (SLAM) is a parameter estimation problem targeting localization $\mathbf{x}_{0:T}$ and mapping $\mathbf{m}$. Given a dataset of the agent inputs $\mathbf{u}_{0:T-1}$ and observations $\mathbf{z}_{0:T}$, a SLAM tries to find the most possible sequence of $\mathbf{x}_{0:T}$ and $\mathbf{m}$.

SLAM can be implemented based on different techniques. For example, early SLAM used Maximum Likelihood Estimation (MLE), Maximum A Posterior (MAP), and Bayes Inference (BI). Nowadays, based on Bayes filter, there are many options for SLAM, such as particle filter, Kalman filter, etc.

A typical cycle of an SLAM algorithm based on Bayes filter or other similar filters is

1) predict agent state $\mathbf{x}_{t+1|t}$ with the control input $\mathbf{u}_t$ based on the observation model
2) correct the predicted agent state as $\mathbf{x}_{t+1|t+1}$ with the observation $\mathbf{z}_{t+1}$ with the observation model and update the environment $\mathbf{m}$ based on $\mathbf{x}_{t+1|t+1}$

III. Technical Approach

A. Kalman Filter

A Kalman filter is a Bayes filter with the following assumptions:

1) Prior: the prior pdf $p_{0|0}$ is Gaussian. Because Gaussian distribution is stable, given $p_{0|0}$ is Gaussian, the prior in the following time steps is also Gaussian

$$\mathbf{x}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}) \tag{6}$$

2) Motion Model: the motion model is linear in the state and affected by Gaussian noise

$$\begin{aligned} &\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) = \mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{u}_t + \mathbf{w}_t, \\ &\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{W}) \end{aligned} \tag{7}$$

$$\begin{aligned} &\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t \sim \mathcal{N}\left(\mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{u}_t, \mathbf{W}\right) \\ &\text{where } \mathbf{F} \in \mathbb{R}^{d_x \times d_x}, \mathbf{G} \in \mathbb{R}^{d_x \times d_u}, \mathbf{W} \in \mathbb{R}^{d_x \times d_x} \end{aligned} \tag{8}$$

3) Observation Model: the observation model is linear in the state and affected by Gaussian noise

$$\begin{aligned} &\mathbf{z}_t = h\left(\mathbf{x}_t, \mathbf{v}_t\right) = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t, \\ &\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{V}) \end{aligned} \tag{9}$$

$$\begin{aligned} &\mathbf{z}_t|\mathbf{x}_t \sim \mathcal{N}\left(\mathbf{H}\mathbf{x}_t, \mathbf{V}\right) \\ &\text{where } \mathbf{H} \in \mathbb{R}^{d_z \times d_x}, \mathbf{V} \in \mathbb{R}^{d_z \times d_z} \end{aligned} \tag{10}$$

4) Noise Independency: the process $\mathbf{w}_t$ and measurement noise $\mathbf{v}_t$ are independent of each other, of the state $\mathbf{x}_t$, and across time.

With the above assumptions, we can derive the prediction step and update step of a Kalman filter:

5) Prediction Step: with (4, 7), we get

$$\begin{aligned} p_{t+1|t}(\mathbf{x}) &= \int \phi\left(\mathbf{x}; \mathbf{F}\mathbf{s} + \mathbf{G}\mathbf{u}_t, \mathbf{W}\right) \phi\left(\mathbf{s}; \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}\right) d\mathbf{s} \\ &= \phi\left(\mathbf{x}; \mathbf{F}\boldsymbol{\mu}_{t|t} + \mathbf{G}\mathbf{u}_t, \mathbf{F}\boldsymbol{\Sigma}_{t|t}\mathbf{F}^\top + \mathbf{W}\right) \end{aligned} \tag{11}$$

Therefore, in the prediction step, we predict the mean and covariance of the agent state $\mathbf{x}_{t+1}$

$$\boldsymbol{\mu}_{t+1|t} = \mathbf{F}\boldsymbol{\mu}_{t|t} + \mathbf{G}\mathbf{u}_t \tag{12}$$

$$\boldsymbol{\Sigma}_{t+1|t} = \mathbf{F}\boldsymbol{\Sigma}_{t|t}\mathbf{F}^\top + \mathbf{W} \tag{13}$$

6) Update Step: with (5, 9, 11), given observation $\mathbf{z}_{t+1}$ and the observation model p_h, we update the pdf of $\mathbf{x}_{t+1}$ by

$$\begin{aligned} p_{t+1|t+1}(\mathbf{x}) &= \frac{\phi\left(\mathbf{z}_{t+1}; \mathbf{H}\mathbf{x}, \mathbf{V}\right) \phi\left(\mathbf{x}; \boldsymbol{\mu}_{t+1|t}, \boldsymbol{\Sigma}_{t+1|t}\right)}{\int \phi\left(\mathbf{z}_{t+1}; \mathbf{H}\mathbf{s}, \mathbf{V}\right) \phi\left(\mathbf{s}; \boldsymbol{\mu}_{t+1|t}, \boldsymbol{\Sigma}_{t+1|t}\right) d\mathbf{s}} \\ &= \phi\left(\mathbf{x}; \boldsymbol{\mu}_{t+1|t+1}, \boldsymbol{\Sigma}_{t+1|t+1}\right) \end{aligned} \tag{14}$$

$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} + \boldsymbol{K}_{t+1|t}\left(\mathbf{z}_{t+1} - \mathbf{H}\boldsymbol{\mu}_{t+1|t}\right) \tag{15}$$

$$\boldsymbol{\Sigma}_{t+1|t+1} = \left(\mathbf{I} - \boldsymbol{K}_{t+1|t}\mathbf{H}\right)\boldsymbol{\Sigma}_{t+1|t} \tag{16}$$

$$\boldsymbol{K}_{t+1|t} = \boldsymbol{\Sigma}_{t+1|t}\mathbf{H}^\top\left(\mathbf{H}\boldsymbol{\Sigma}_{t+1|t}\mathbf{H}^\top + \mathbf{V}\right)^{-1} \tag{17}$$

We can see that $\boldsymbol{K}_{t+1|t}$ is the confidence of this update step, which is expressed as a formula of covariance $\boldsymbol{\Sigma}_{t+1|t}$, $\mathbf{V}$, and the transformation $\mathbf{H}$ between agent state and observation. The agent state is corrected by the difference between the observation $\mathbf{z}_{t+1}$ and predicted observation $\mathbf{H}\boldsymbol{\mu}_{t+1|t}$ with confidence $\boldsymbol{K}_{t+1|t}$, which is called Kalman gain.

7) Information Filter: If we let

$$\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1} \tag{18}$$

$$\boldsymbol{\nu} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \boldsymbol{\Omega}\boldsymbol{\mu} \tag{19}$$

the update step of Kalman filter can become very simple.
prediction:

$$\boldsymbol{\nu}_{t+1|t} = (\mathbf{I} - \mathbf{C}_{t+1|t})\mathbf{F}^{-\top}\boldsymbol{\nu}_{t|t} + \boldsymbol{\Omega}_{t+1|t}\mathbf{G}\mathbf{u}_t \tag{20}$$

$$\boldsymbol{\Omega}_{t+1|t} = \mathbf{C}_{t|t}\mathbf{W}^{-1} = \mathbf{W}^{-1}\mathbf{C}_{t|t}^\top \tag{21}$$

gain:

$$\mathbf{C}_{t|t} = \mathbf{F}^{-\top}\boldsymbol{\Omega}_{t|t}\mathbf{F}^{-1}\left(\mathbf{F}^{-\top}\boldsymbol{\Omega}_{t|t}\mathbf{F}^{-1} + \mathbf{W}^{-1}\right)^{-1} \tag{22}$$

update:

$$\boldsymbol{\nu}_{t+1|t+1} = \boldsymbol{\nu}_{t+1|t} + \mathbf{H}^\top\mathbf{V}^{-1}\mathbf{z}_{t+1} \tag{23}$$

$$\boldsymbol{\Omega}_{t+1|t+1} = \boldsymbol{\Omega}_{t+1|t} + \mathbf{H}^\top\mathbf{V}^{-1}\mathbf{H} \tag{24}$$

B. Nonlinear Kalman Filter

Kalman filter has a strong assumption on the linearity of the motion model and the observation model. But in most cases, the system is not linear and it can no longer be evaluated in closed form.

However, given a motion model $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$ where $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{W})$ and an observation model $\mathbf{z}_{t+1} = h(\mathbf{x}_{t+1}, \mathbf{v}_{t+1})$ where $\mathbf{v}_{t+1} \sim \mathcal{N}(0, \mathbf{V})$, we can force the predicted and updated pdfs to be Gaussian by evaluating their first and second moments and approximating them with Gaussian s with the same moments. To calculate these moments, we need to use lots of integrals, which is infeasible in discrete cases.

Extended and Unscented Kalman filters (EKF and UKF) are two different methods to approximate those integrals. The EKF uses a first-order Taylor series approximation while the UKF uses a set of sigma points that capture the mean and covariance of the prior Gaussian pdfs to approximate the integrals via a sum. In this report, I used EKF to implement the SLAM.

C. Extended Kalman Filter

Prediction Step

Let $\mathbf{F}_t = \frac{df}{d\mathbf{x}}\left(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}\right)$ and $\mathbf{Q}_t = \frac{df}{d\mathbf{w}}\left(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}\right)$ so that we can linearize the motion model by:

$$f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \approx f\left(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, \mathbf{0}\right) + \mathbf{F}_t\left(\mathbf{x}_t - \boldsymbol{\mu}_{t|t}\right) + \mathbf{Q}_t\mathbf{w}_t$$

then, we can calculate the predicted mean and covariance in closed form:

$$\boldsymbol{\mu}_{t+1|t} \approx f\left(\boldsymbol{\mu}_{t|t}, \mathbf{u}_t, 0\right) \tag{25}$$

$$\boldsymbol{\Sigma}_{t+1|t} \approx \mathbf{F}_t\boldsymbol{\Sigma}_{t|t}\mathbf{F}_t^\top + \mathbf{Q}_t\mathbf{W}\mathbf{Q}_t^\top \tag{26}$$

Update Step

Let $\mathbf{H}_{t+1} = \frac{dh}{d\mathbf{x}}\left(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}\right)$ and $\mathbf{R}_{t+1} = \frac{dh}{d\mathbf{v}}\left(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}\right)$ so that the linearization of the observation model is:

$$\begin{aligned} &h(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \\ &\approx h\left(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}\right) + \mathbf{H}_{t+1}\left(\mathbf{x}_{t+1} - \boldsymbol{\mu}_{t+1|t}\right) + \mathbf{R}_{t+1}\mathbf{v}_{t+1}\end{aligned}$$

Then, we can use Gaussian distribution to approximate the joint distribution of $\mathbf{x}_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t}$ and $\mathbf{z}_{t+1}$:

$$\begin{pmatrix} \mathbf{x}_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t} \\ \mathbf{z}_{t+1} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{t+1|t} \\ \mathbf{m}_{t+1|t} \end{pmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{t+1|t} & \mathbf{C}_{t+1|t} \\ \mathbf{C}_{t+1|t}^\top & \mathbf{S}_{t+1|t} \end{bmatrix}\right) \tag{27}$$

$$\mathbf{m}_{t+1|t} \approx h\left(\boldsymbol{\mu}_{t+1|t}, \mathbf{0}\right) \tag{28}$$

$$\mathbf{S}_{t+1|t} \approx \mathbf{H}_{t+1}\boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1}^\top + \mathbf{R}_{t+1}\mathbf{V}\mathbf{R}_{t+1}^\top \tag{29}$$

$$\mathbf{C}_{t+1|t} \approx \boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1}^\top \tag{30}$$

Similarly, we can derive the update step

$$\boldsymbol{\mu}_{t+1|t+1} = \boldsymbol{\mu}_{t+1|t} + \mathbf{K}_{t+1|t}\left(\mathbf{z}_{t+1} - \mathbf{m}_{t+1|t}\right) \tag{31}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{t+1|t+1} &= \boldsymbol{\Sigma}_{t+1|t} - \mathbf{K}_{t+1|t}\mathbf{S}_{t+1|t}\mathbf{K}_{t+1|t}^\top \\ &= \left(\mathbf{I} - \boldsymbol{K}_{t+1|t}\mathbf{H}\right)\boldsymbol{\Sigma}_{t+1|t}\end{aligned} \tag{32}$$

$$\begin{aligned}\mathbf{K}_{t+1|t} &= \mathbf{C}_{t+1|t}\mathbf{S}_{t+1|t}^{-1} \\ &= \boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1}^\top\left(\mathbf{H}_{t+1}\boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1}^\top + \mathbf{R}_{t+1}\mathbf{V}\mathbf{R}_{t+1}^\top\right)^{-1}\end{aligned} \tag{33}$$

We can see that (32) is the same as (16).

D. SO(3) and SE(3) Geometry and Kinematics with Matrix Lie Group and Lie Algebra

1) SO(3): A rotation matrix is an element of the Special Orthogonal Group, $SO(3)$:

$$\mathbf{R} \in SO(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} \middle| \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\right\} \tag{34}$$

which can be represented by $\boldsymbol{\theta} \in \mathbb{R}^3$

$$\mathbf{R} = \exp(\hat{\boldsymbol{\theta}}) = \mathbf{I} + \hat{\boldsymbol{\theta}} + \frac{1}{2!}\hat{\boldsymbol{\theta}}^2 + \frac{1}{3!}\hat{\boldsymbol{\theta}}^3 + ... \tag{35}$$

$\boldsymbol{\theta}$ specifies an axis $\boldsymbol{\eta} = \frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||}$ and a rotation angle $||\boldsymbol{\theta}||$.

2) SE(3): The pose $\mathbf{T}$ of a rigid body can be described by a matrix in the Special Euclidean Group, $SE(3)$:

$$SE(3) := \left\{T := \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{bmatrix} \middle| \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^3\right\} \subset \mathbb{R}^{4\times 4} \tag{36}$$

3) Matrix Lie Group and Lie Algebra: $SO(3)$ and $SE(3)$ are matrix Lie groups. The Lie algebra of $SO(3)$ is $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ for $SE(3)$. And the exponential map relates a matrix Lie group to its Lie algebra:

$$\mathbf{R} = \exp(\hat{\boldsymbol{\theta}}) = \sum_{n=0}^{\infty}\frac{1}{n!}\hat{\boldsymbol{\theta}}^n \tag{37}$$

$$\boldsymbol{\theta} = \log(\mathbf{R})^\vee = \left[\sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{n}(\mathbf{A} - \mathbf{I})^n\right]^\vee \tag{38}$$

4) Exponential Map from $\mathfrak{so}(3)$ to $SO(3)$: Some key equations we will use:

- **Rodrigues Formula**:

$$\hat{\boldsymbol{\theta}}^{2n+1} = (-\boldsymbol{\theta}^\top\boldsymbol{\theta})^n\hat{\boldsymbol{\theta}} \tag{39}$$

- Exponential Map:

$$\mathbf{R} = \exp(\hat{\boldsymbol{\theta}}) = \mathbf{I} + \left(\frac{\sin||\boldsymbol{\theta}||}{||\boldsymbol{\theta}||}\right)\hat{\boldsymbol{\theta}} + \left(\frac{1 - \cos(||\boldsymbol{\theta}||)}{||\boldsymbol{\theta}||^2}\right)\hat{\boldsymbol{\theta}}^2 \tag{40}$$

5) Logarithm Map from $SO(3)$ to $\mathfrak{so}(3)$: because the exponential map is surjective but not injective, $\forall\, \mathbf{R} \in SO(3)$, there exists a non-unique $\boldsymbol{\theta} \in \mathbb{R}^3$ such that $\mathbf{R} = \exp(\hat{\boldsymbol{\theta}})$. In this report, I used the following logarithm map:

$$\theta = ||\boldsymbol{\theta}|| = \arccos\left(\frac{\text{tr}(\mathbf{R}) - 1}{2}\right) \tag{41}$$

$$\boldsymbol{\eta} = \frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||} = \frac{1}{2\sin(||\boldsymbol{\theta}||)}\begin{bmatrix}\mathbf{R}_{32} - \mathbf{R}_{23} \\ \mathbf{R}_{13} - \mathbf{R}_{31} \\ \mathbf{R}_{21} - \mathbf{R}_{12}\end{bmatrix} \tag{42}$$

$$\hat{\boldsymbol{\theta}} = \log(\mathbf{R}) = \frac{||\boldsymbol{\theta}||}{2\sin(||\boldsymbol{\theta}||)}(\mathbf{R} - \mathbf{R}^\top) \tag{43}$$

Note that there is a singularity at $\theta = 0$.

6) Derivative and Perturbation in $SO(3)$: when linearizing a non-linear dynamic system, we sometimes need to calculate the derivative of a rotation result $\mathbf{Rs}$ with respect to the rotation $\mathbf{R}$ because $\boldsymbol{\theta}$ corresponding to $\mathbf{R}$ can be a parameter of the motion model.

To calculate the derivative, we need to introduce the left Jacobian of $SO(3)$ under left multiplication:

$$\begin{aligned} J_L(\boldsymbol{\theta}) &= \sum_{n=0}^{\infty} \frac{1}{(n+1)!}\left(\hat{\boldsymbol{\theta}}\right)^n \\ &= \mathbf{I} + \left(\frac{1 - \cos||\boldsymbol{\theta}||}{||\boldsymbol{\theta}||^2}\right)\hat{\boldsymbol{\theta}} + \left(\frac{||\boldsymbol{\theta}|| - \sin||\boldsymbol{\theta}||}{||\boldsymbol{\theta}||^3}\right)\hat{\boldsymbol{\theta}}^2 \\ &\approx \mathbf{I} + \frac{1}{2}\hat{\boldsymbol{\theta}} \end{aligned} \tag{44}$$

$$\mathbf{R} = \mathbf{I} + \hat{\boldsymbol{\theta}} J_L(\boldsymbol{\theta}) \tag{45}$$

Baker-Campbell-Hausdorff Formula:

$$\exp\left((\boldsymbol{\theta} + \delta\boldsymbol{\theta})^\wedge\right) \approx \exp\left((J_L(\boldsymbol{\theta})\delta\boldsymbol{\theta})^\wedge\right)\exp(\hat{\boldsymbol{\theta}}) \tag{46}$$

So, the directional derivative is

$$\frac{\partial \mathbf{Rs}}{\partial \boldsymbol{\theta}} = -(\mathbf{Rs})^\wedge J_L(\boldsymbol{\theta}) \tag{47}$$

$$\exp\left((\boldsymbol{\theta} + \delta\boldsymbol{\theta})^\wedge\right)\mathbf{s} = R\mathbf{s}\underbrace{-(R\mathbf{s})^\wedge J_L(\boldsymbol{\theta})}_{\frac{\partial \mathbf{Rs}}{\partial \boldsymbol{\theta}}}\delta\boldsymbol{\theta} = \mathbf{Rs} + \frac{\partial \mathbf{Rs}}{\partial \boldsymbol{\theta}}\delta\boldsymbol{\theta} \tag{48}$$

E. Visual-Inertial SLAM with EKF

With camera and IMU data, we can predict and update agent localization and mapping with EKF. The goal is to output the agent pose ${}_w\mathbf{T}_I \in SE(3)$ overtime and the world-frame coordinates of the point landmarks $\mathbf{m} \in \mathbb{R}^{3\times M}$ that generated the features $\mathbf{z}_t$.

1) Stereo Camera: In this report, I used stereo camera, of which the calibration matrix $\mathbf{M}$ of intrinsic parameters is:

$$\mathbf{M} = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \tag{49}$$

2) Visual Mapping: By assuming the localization is known, we can consider the mapping-only problem. i.e. given the inverse IMU pose

$$\mathbf{U}_t =_w \mathbf{T}_I^{-1} \in SE(3)$$

and the visual feature observations $\mathbf{z}_{0:T}$, estimate the homogeneous coordinates $\mathbf{m} \in \mathbb{R}^{4\times L}$ in the world frame of the L landmarks that generated the visual observations.

The landmarks are assumed to be static and their data association at each time t is also known.

With the prior

$$\mathbf{m}|\mathbf{z}_{0:t} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t) \text{ with } \boldsymbol{\mu}_t \in \mathbb{R}^{3L} \text{ and } \boldsymbol{\Sigma}_t \in \mathbb{R}^{3L\times 3L} \tag{50}$$

the observation model

$$\begin{aligned} &\mathbf{z}_{t,i} = h(\mathbf{U}_t, \mathbf{m}_j) + \mathbf{v}_{t,i} = \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\mathbf{m}}_j\right) + \mathbf{v}_{t,i} \\ &\mathbf{v}_{t,i} \sim \mathcal{N}(0, \mathbf{V}) \end{aligned} \tag{51}$$

where $\underline{\mathbf{m}}$ is the homogeneous form of $\mathbf{m}$, the projection function and its derivative are

$$\pi(\mathbf{q}) = \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^4 \tag{52}$$

$$\frac{d\pi}{d\mathbf{q}}(\mathbf{q}) = \frac{1}{q_3}\begin{bmatrix} 1 & 0 & -\frac{q_1}{q_3} & 0 \\ 0 & 1 & -\frac{q_2}{q_3} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{q_4}{q_3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{53}$$

By stacking all the N_t observations as homogeneous coordinates of all the L landmarks as a $4N_t$ vector at time t, we can calculate all the observations at the same time:

$$\mathbf{z}_t = \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\mathbf{m}}\right) + \mathbf{v}_t \tag{54}$$

$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}\otimes\mathbf{V}) \tag{55}$$

$$\mathbf{I}\otimes\mathbf{V} = \begin{bmatrix}\mathbf{V} & & \\ & \ldots & \\ & & \mathbf{V}\end{bmatrix} \tag{56}$$

Visual Mapping via the EKF

According to (31, 32, 33), with (54, 55, 56), we can derive the EKF update step for visual mapping:

$$\hat{\mathbf{z}}_{t,i} = \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\boldsymbol{\mu}}_{t,j}\right) \in \mathbb{R}^4,\ \mathbf{z}_t \in \mathbb{R}^{4\times N_t} \tag{57}$$

$$\mathbf{K}_t = \boldsymbol{\Sigma}_t\mathbf{H}_t^\top\left(\mathbf{H}_t\boldsymbol{\Sigma}_t\mathbf{H}_t^\top + \mathbf{I}\otimes\mathbf{V}\right)^{-1} \tag{58}$$

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - \hat{\mathbf{z}}_t) \tag{59}$$

$$\boldsymbol{\Sigma}_{t+1} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\boldsymbol{\Sigma}_t \tag{60}$$

where $\boldsymbol{\mu}_t$ is the estimation of landmark positions in implementation, $\mathbf{H}_t \in \mathbb{R}^{4N_t\times 3L}$ is the Jacobian of the observation model, and its element corresponding to observations i and

different landmarks j is $\mathbf{H}_{t,i,j} \in \mathbb{R}^{4\times 3}$. By considering a perturbation $\delta\mathbf{m}_{t,j}$ for the position of landmark j:

$$\mathbf{m}_j = \boldsymbol{\mu}_{t,j} + \delta\boldsymbol{\mu}_{t,j} \tag{61}$$

and using the first-order Tayler series approximation, we can derive $\mathbf{H}_t$

$$\begin{aligned}
\mathbf{z}_{t,i} &= \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\left(\boldsymbol{\mu}_{t,j} + \delta\boldsymbol{\mu}_{t,j}\right)}\right) + \mathbf{v}_{t,i} \\
&= \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_t\left(\underline{\boldsymbol{\mu}_{t,j}} + \mathbf{P}^\top\delta\boldsymbol{\mu}_{t,j}\right)\right) + \mathbf{v}_{t,i} \\
&\approx \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\boldsymbol{\mu}_{t,j}}\right) \\
&+ \underbrace{\mathbf{M}\frac{d\pi}{d\mathbf{q}}\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\boldsymbol{\mu}_{t,j}}\right){}_O\mathbf{T}_I\mathbf{U}_t\mathbf{P}^\top\delta\boldsymbol{\mu}_{t,j}}_{\mathbf{H}_{t,i,j}} + \mathbf{v}_{t,i}
\end{aligned} \tag{62}$$

$$\mathbf{H}_{t,i,j} = \begin{cases} \mathbf{M}\frac{d\pi}{d\mathbf{q}}\left({}_O\mathbf{T}_I\mathbf{U}_t\underline{\boldsymbol{\mu}_{t,j}}\right){}_O\mathbf{T}_I\mathbf{U}_t\mathbf{P}^\top \\ \text{observation i and its landmark j} \\ \mathbf{0} \in \mathbb{R}^{4\times 3} \text{ otherwise} \end{cases} \tag{63}$$

3) Visual-Inertial Odometry: Considering the localization-only problem, we employ the following assumptions:

- linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ is available
- the world-frame landmark coordinates $\mathbf{m} \in \mathbb{R}^{3\times M}$ are known
- the association between L landmarks and N_t observations at time t is known.
- the IMU measurements $\mathbf{u}_{0:T}$ with $\mathbf{u}_t = [\mathbf{v}_t^\top, \boldsymbol{\omega}_t^\top]^\top$ and the visual feature observations $\mathbf{z}_{0:T}$ are given.

and our goal is to estimate the inverse IMU pose $\mathbf{U}_t = {}_W\mathbf{T}_{I,t}^{-1}$

At time t, the prior of the pose is given by

$$\mathbf{U}_t \mid \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}\right) \tag{64}$$

where $\boldsymbol{\mu}_{t|t} \in SE(3)$ and $\boldsymbol{\Sigma}_{t|t} \in \mathbb{R}^{6\times 6}$

Since $\mathbf{U}_t$ is the inverse IMU pose, with the IMU measurement $\mathbf{u}_t$, the motion model is defined as

$$\mathbf{U}_{t+1} = \exp\left(-\tau\left(\mathbf{u}_t + \mathbf{w}_t\right)^\wedge\right)\mathbf{U}_t \tag{65}$$

where τ is the time discretization and $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{W})$ is the noise.

Visual-Inertial Odometry via the EKF

By expressing the pose as a nominal pose $\boldsymbol{\mu} \in SE(3)$ and small perturbation $\delta\hat{\boldsymbol{\mu}} \in \mathfrak{se}(3)$:

$$\mathbf{U} = \exp\left(\delta\hat{\boldsymbol{\mu}}\right)\boldsymbol{\mu}$$

we can separate the effect of the noise $\mathbf{w}_t$ from the motion of the deterministic part of $\mathbf{U}_t$ and therefore re-write the motion model in terms of nominal kinematics of the mean of $\mathbf{U}_t$ and zero-mean perturbation kinematics:

$$\begin{aligned}
\boldsymbol{\mu}_{t+1|t} &= \exp\left(-\tau\hat{\mathbf{u}}_t\right)\boldsymbol{\mu}_{t|t} \\
\delta\boldsymbol{\mu}_{t+1|t} &= \exp\left(-\tau\hat{\mathbf{u}}_t^\wedge\right)\delta\boldsymbol{\mu}_{t|t} + \mathbf{w}_t
\end{aligned}$$

Prediction Step

So, according to (25, 26), $\mathbf{F}_t = \exp\left(-\tau\hat{\mathbf{u}}_t^\wedge\right)$, the EKF prediction step is given by

$$\boldsymbol{\mu}_{t+1|t} = \exp\left(-\tau\hat{\mathbf{u}}_t\right)\boldsymbol{\mu}_{t|t} \tag{66}$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{t+1|t} &= \mathbb{E}\left[\delta\boldsymbol{\mu}_{t+1|t}\delta\boldsymbol{\mu}_{t+1|t}^\top\right] \\
&= \exp\left(-\tau\hat{\mathbf{u}}_t^\wedge\right)\boldsymbol{\Sigma}_{t|t}\exp\left(-\tau\hat{\mathbf{u}}_t^\wedge\right)^\top + \mathbf{W}
\end{aligned} \tag{67}$$

where

$$\hat{\mathbf{u}}_t^\wedge = \begin{bmatrix} \hat{\omega}_t & \hat{\mathbf{v}}_t \\ 0 & \hat{\omega}_t \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{68}$$

$$\hat{\boldsymbol{\mu}} = \begin{bmatrix} \hat{\omega}_t & \mathbf{v}_t \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{69}$$

Update Step

For the update step, with the prior $\mathbf{U}_{t+1}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t} \sim \mathcal{N}\left(\boldsymbol{\mu}_{t+1|t}, \boldsymbol{\Sigma}_{t+1|t}\right)$ and the observation model

$$\mathbf{z}_{t+1,i} = h(\mathbf{U}_{t+1}, \mathbf{m}_j) + \mathbf{v}_{t+1,i} \tag{70}$$

$$= \mathbf{M}\pi\left({}_O\mathbf{T}_I\mathbf{U}_{t+1}\underline{\mathbf{m}_j}\right) + \mathbf{v}_{t+1,j} \tag{71}$$

which is the same as the one in the visual mapping problem but with a different variable of interest, $\mathbf{U}_{t+1}$. So, we need the observation model Jacobian $\mathbf{H}_{t+1|t} \in \mathbb{R}^{4N_t\times 6}$ with respect to the inverse IMU pose $\mathbf{U}_t$ at $\boldsymbol{\mu}_{t+1|t}$

$$\mathbf{H}_{i,t+1|t} = \mathbf{M}\frac{d\pi}{d\mathbf{q}}\left({}_O\mathbf{T}_I\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right){}_O\mathbf{T}_I\left(\boldsymbol{\mu}_{t+1|t}\mathbf{m}_j\right)^\odot \in \mathbb{R}^{4\times 6} \tag{72}$$

where for homogeneous coordinates $\mathbf{s} \in \mathbb{R}^4$ and $\hat{\boldsymbol{\xi}} \in \mathfrak{se}(3)$:

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

Therefore, the update step is

$$\mathbf{K}_{t+1|t} = \boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1|t}^\top\left(\mathbf{H}_{t+1|t}\boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1|t}^\top + \mathbf{I}\otimes\mathbf{V}\right)^{-1} \tag{73}$$

$$\boldsymbol{\mu}_{t+1|t+1} = \exp\left(\left(\mathbf{K}_{t+1|t}\left(\mathbf{z}_{t+1} - \hat{\mathbf{z}}_{t+1}\right)\right)^\wedge\right)\boldsymbol{\mu}_{t+1|t} \tag{74}$$

$$\boldsymbol{\Sigma}_{t+1|t+1} = \left(\mathbf{I} - \mathbf{K}_{t+1|t}\mathbf{H}_{t+1|t}\right)\boldsymbol{\Sigma}_{t+1|t} \tag{75}$$

where

$$\mathbf{H}_{t+1|t} = \begin{bmatrix} \mathbf{H}_{1,t+1|t} \\ \vdots \\ \mathbf{H}_{N_{t+1},t+1|t} \end{bmatrix} \tag{76}$$

4) Localization and Mapping: Now, combine the visual mapping and visual-inertial odometry:

a. robot state: $\boldsymbol{\mu}_{t|t}^{P} \in \mathbb{R}^{4\times 4}, \boldsymbol{\Sigma}_{t|t}^{P} \in \mathbb{R}^{6\times 6}$
b. landmark state: $\boldsymbol{\mu}_{t|t}^{L} \in \mathbb{R}^{4\times L}, \boldsymbol{\Sigma}_{t|t}^{L} \in \mathbb{R}^{3L\times 3L}$
c. system state covariance:

$$\boldsymbol{\Sigma}_{t|t} = \begin{bmatrix} \boldsymbol{\Sigma}_{t|t}^{P} & \mathbf{C}_{t|t} \\ \mathbf{C}_{t|t}^{T} & \boldsymbol{\Sigma}_{t|t}^{L} \end{bmatrix} \in \mathbb{R}^{(6+3L)\times(6+3L)} \tag{77}$$

Prediction Step

To predict the robot state and the landmark state, we still use (66) to predict the robot state. As for the landmarks' states, we do not need to predict their states.

To predict the system state covariance, we use the following equation:

$$\boldsymbol{\Sigma}_{t+1|t} = \begin{bmatrix} \exp\left(-\tau\hat{\boldsymbol{\mu}}_t\right) & 0 \\ 0^{\top} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\Sigma}_{t|t} \begin{bmatrix} \exp\left(-\tau\hat{\boldsymbol{\mu}}_t\right) & 0 \\ 0^{\top} & \boldsymbol{I} \end{bmatrix}^{\top} + \begin{bmatrix} \mathbf{W} & 0 \\ 0 & 0 \end{bmatrix} \tag{78}$$

Update Step

In the update step, we still use those equations described in the former two small sections. But before using those equations, we need to construct the $\mathbf{H}$ for the system from $\mathbf{H}^{L}$ and $\mathbf{H}^{P}$, which are respectively the landmark's and the robot state's $\mathbf{H}$ matrix.

- robot state: $\mathbf{H}^{P} \in \mathbb{R}^{4N_t\times 6}$
- landmark state: $\mathbf{H}^{L} \in \mathbb{R}^{4N_t\times 3L}$
- system state: $\mathbf{H} = \left[\mathbf{H}^{P}, \mathbf{H}^{L}\right] \in \mathbb{R}^{4N_T\times(6+3L)}$

Then, we can calculate the Kalman gain for the system using:

$$\mathbf{K}_{t+1|t} = \boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1|t}^{\top}\left(\mathbf{H}_{t+1|t}\boldsymbol{\Sigma}_{t+1|t}\mathbf{H}_{t+1|t}^{\top} + \mathbf{I}\otimes\mathbf{V}\right)^{-1}$$

which is the same as the equation in the visual mapping problem.

Then, we can split $\mathbf{K}_{t+1|t}$ into two parts:

$$\mathbf{K}_{t+1|t} = \begin{bmatrix} \mathbf{K}_{t+1|t}^{P} \\ \mathbf{K}_{t+1|t}^{L} \end{bmatrix}, \ \mathbf{K}_{t+1|t}^{P} \in \mathbb{R}^{6\times 4N_t}, \mathbf{K}_{t+1|t}^{L} \in \mathbb{R}^{3L\times 4N_t} \tag{79}$$

and update the robot states and the landmarks' states with these two Kalman gains.

IV. Results and Discussion

A. Localization Only

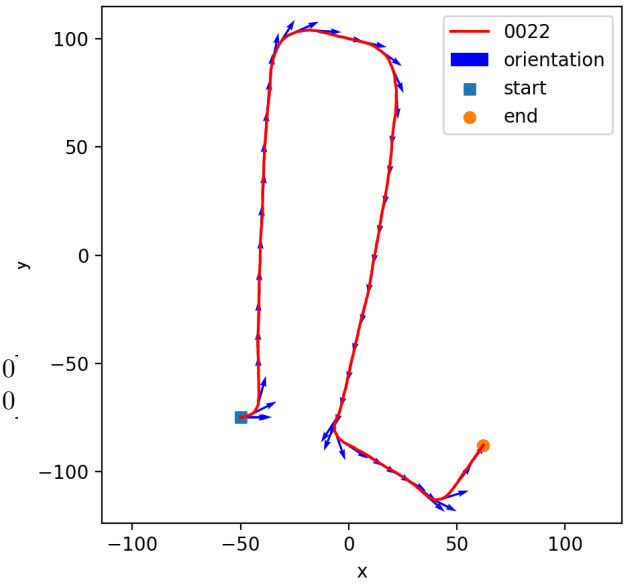

Fig. 2. Dataset 0022

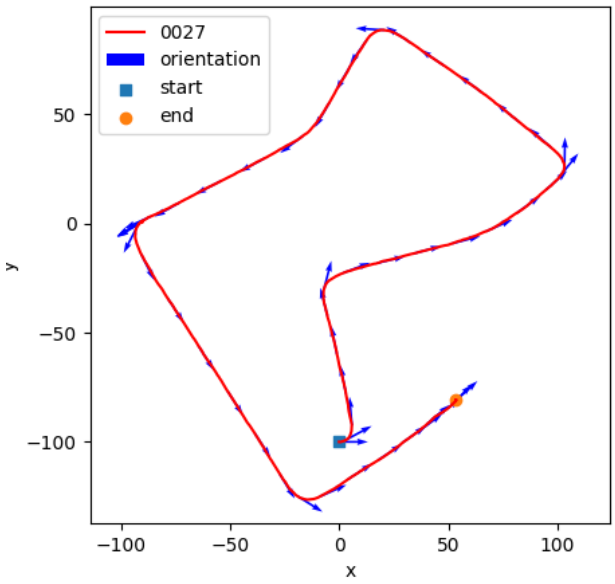

Fig. 3. Dataset 0027

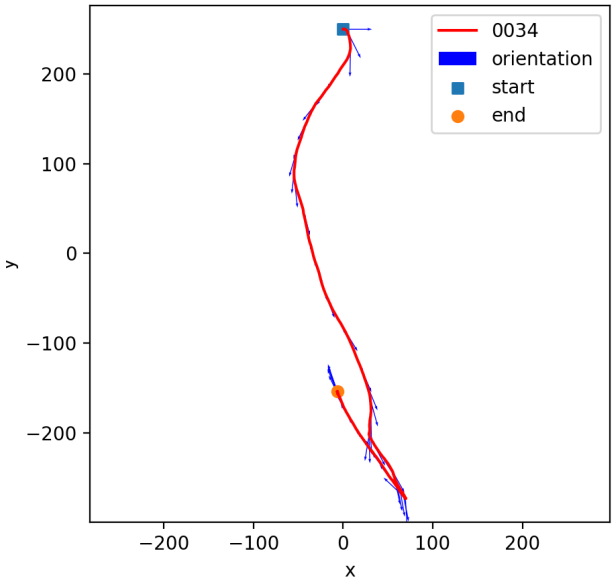

Fig. 4. Dataset 0034

Compared with the videos of these datasets, we find that with only the localization, the path's shape is correct overall. But at the position of turning, the predicted turning angle is not accurate enough. And the IMU data's noise is cumulated because of the absence of update step. The result of datatset 0027 shows that the path, which should be closed, is not closed because the inaccuracy of IMU data.

B. Mapping Only

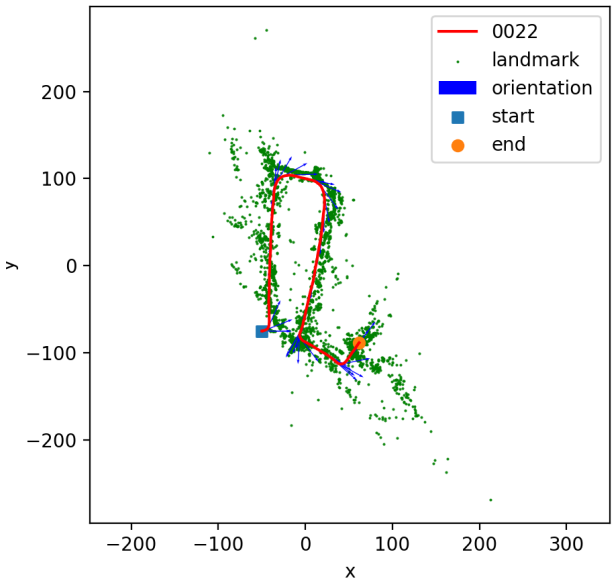

Fig. 5. Dataset 0022

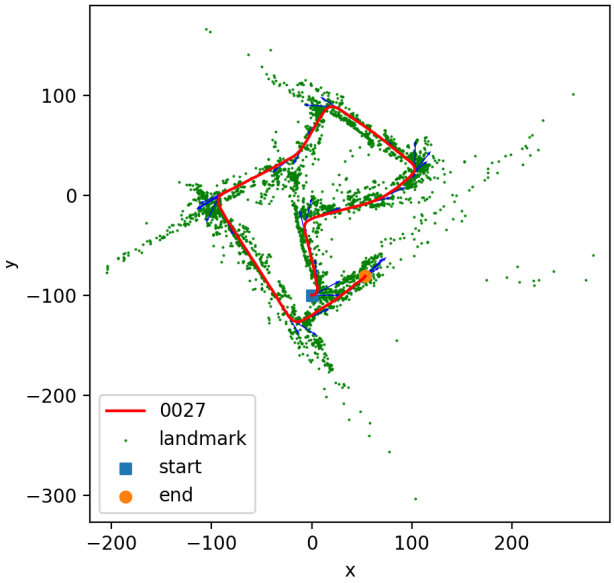

Fig. 6. Dataset 0027

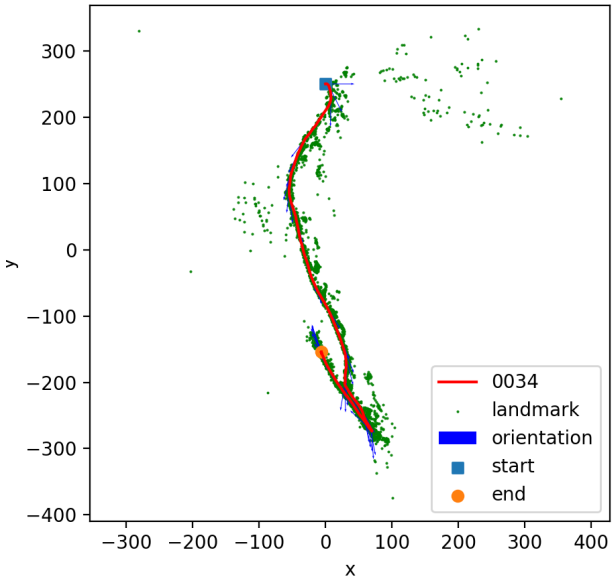

Fig. 7. Dataset 0034

With the predicted pose from the localization only part, we can see that the landmarks' positions are not well estimated.

C. SLAM

By combining the localization and the mapping, making them correct each other, we can see that visual-inertial SLAM is very robust. With only one parameter setup,

$$\mathbf{W} = \begin{bmatrix} 8.e^{-6} & & & & & \\ & 8.e^{-6} & & & & \\ & & 8.e^{-6} & & & \\ & & & 2.e^{-10} & & \\ & & & & 2.e^{-10} & \\ & & & & & 2.e^{-10} \end{bmatrix} \tag{80}$$

$$\mathbf{V} = \mathbf{I} \tag{81}$$

,

this SLAM algorithm can generate the paths' of all the dataset shown in the following images. From the error curve of pose and landmark in every dataset's result, we can also see that the SLAM can handle some occasional errors from the camera side.

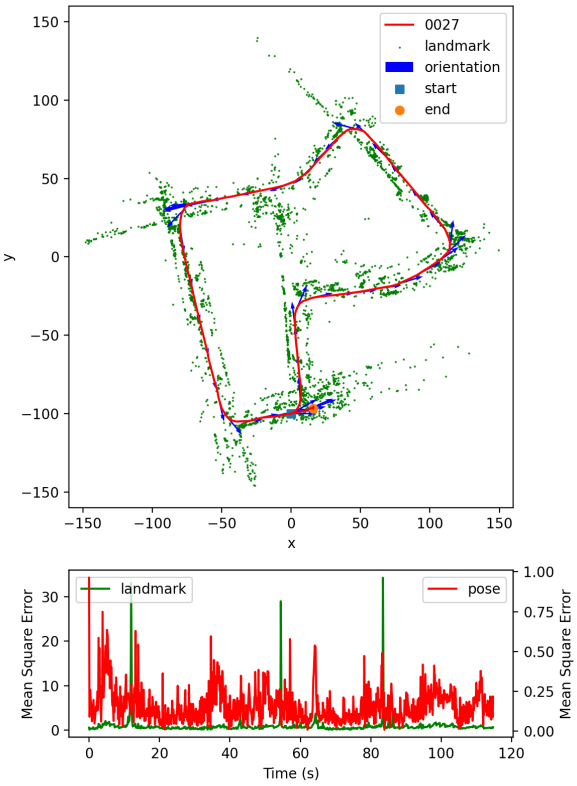

Fig. 9. Dataset 0027

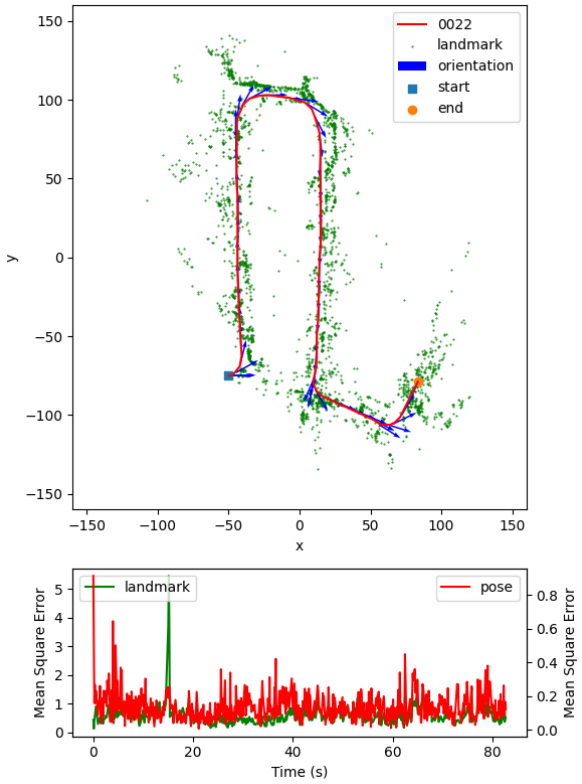

Fig. 8. Dataset 0022

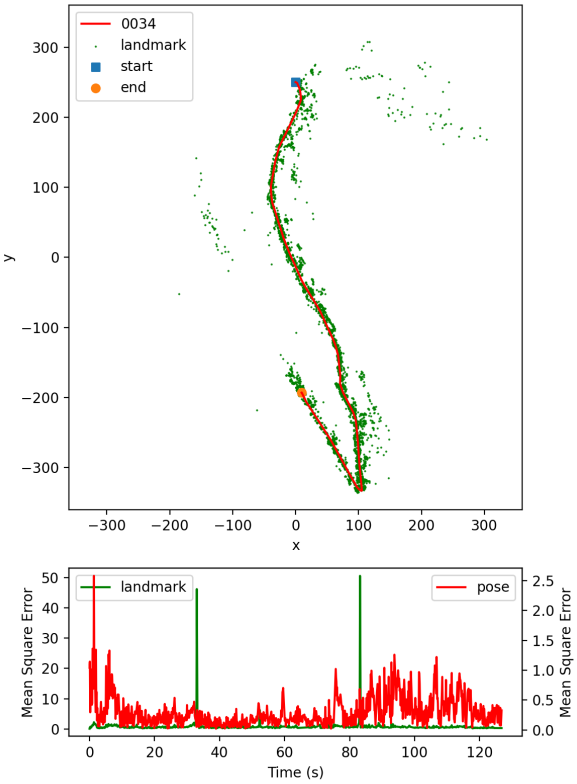

Fig. 10. Dataset 0034