Continuous-Time Fixed-Lag Smoothing for LiDAR-Inertial-Camera SLAM

Basic information

Published in: IEEE/ASME TRANSACTIONS ON MECHATRONICS
Date Added to IEEE *Xplore*: JULY 2022
Publisher: IEEE

2024/11/8

Xutong Zhang

Abstract

- **Question**
  - To adequately fuse multi-modal sensor measurements received at different time instants and different frequencies, the paper estimate the continuoustime trajectory by fixed-lag smoothing within a factor-graph optimization framework.
- **Method**
  - Adopt continuous-time fixed-lag smoothing method for multi-sensor fusion in a factor-graph optimization framework.
  - Design some LI SLAM and LIC SLAM systems at a variety of sensor combinations and derive the analytical Jacobians for efficient factor-graph optimization.
  - Online calibrate timeoffsets between different sensors.

Introduction

- **SLAM tasks → External signals are unavailable → proprioceptive and exteroceptive**
  - proprioceptive sensors: wheel encoders, inertial measurement unit, magnetometer
  - Exteroceptive: radar, sonar, LiDAR, camera, barometer(气压计), altimeter（测高仪）

- **Timeoffsets generally exist among different sensors**
  - [9] M. Li and A. I. Mourikis. "Online temporal calibration for camera–IMU systems: Theory and algorithms". In: The International Journal of Robotics Research 33.7 (2014), pp. 947–964.
  - [10] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys. "LICFusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking". In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE. 2020, pp. 5112–5119.

Introduction

- **Different sensor frequencies and sampling time instants**
  - Linearly interpolate IMU poses at the sampling time of LiDAR, in order to compensate motion distortion of LiDAR
  - B-spline: Allows pose querying at any time instants without interepolations
    - Two challenges
      - Real-time performance
      - Fixed-lag smoothing: Continues time methods like VIO or LIO rarely consider how to preserve information of old measurements/states.

RELATED WORKS

- **Multi-sensor Fusion for SLAM**
  - 1) LiDAR-Inertial-Camera Fusion for SLAM:
    - Graph-based optimization
      - [6] LVI-SAM: Tightly-coupled lidarvisual-inertial odometry via smoothing and mapping
      - [16] Complementary perception for handheld slam
      - [25] Laser–visual–inertial odometry and mapping with high robustness and low drift
      - [26] Robust high accuracy visual-inertial-laser slam system
      - [29] Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry
      - [30] VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots
      - [32] Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments
      - [33] Lvio-Fusion: A Self-adaptive Multi-sensor Fusion SLAM Framework Using Actorcritic Method

RELATED WORKS

- **Multi-sensor Fusion for SLAM**
  - 1) LiDAR-Inertial-Camera Fusion for SLAM:
    - Graph-based optimization
    - Filter-based methods
      - [4] LIC-Fusion: Lidar-inertial camera odometry
      - [7] R3LIVE: A Robust, Real-time, RGB-colored, LiDAR Inertial-Visual tightly-coupled state Estimation and mapping package
      - [10] LIC-Fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature
      - tracking

RELATED WORKS

- **Multi-sensor Fusion for SLAM**
  - 1) LiDAR-Inertial-Camera Fusion for SLAM:
    - Graph-based optimization
    - filter-based methods
    - Both the graph optimization and filter
      - [31] R 2 LIVE: A Robust, RealTime, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping
    - LIC systems in Tab1.
    - Four typical pipelines in Fig1.

RELATED WORKS

- **Multi-sensor Fusion for SLAM**
  - 1) LiDAR-Inertial-Camera Fusion for SLAM:
    - Graph-based optimization
    - filter-based methods
    - Both the graph optimization
    - LIC systems in Tab1.
    - Four typical pipelines in Fig

Paper	Year	IMU[1]	Camera[2]	LiDAR[3]	Method[4]
V-LOAM [25]	2018-JFR	I1	C1, C3	L1	F1
Wang. [26]	2019-IROS	I2	C1	L1	F2
Khattak. [27]	2019-ICUAS	ROVIO [28]		L1	F2
Lowe. [16]	2018-RAL	I3	C1, C4	L5	F4
LIC-Fusion [4]	2019-IROS	I1	C1	L1	MSCKF
LIC-Fusion2.0 [10]	2020-IROS	I1	C1	L1, L2	MSCKF
LVI-SAM [6]	2021-ICRA	I2	C1, C3	L1	F3
VILENS [29]	2021-RAL	I2	C1, C3	L2	F4
VILENS [30]	2021-Arxiv	I2	C1, C3	L2, L3	F4
R2live [31]	2021-RAL	I1, I2	C1	L1	ESIKF, F4
R3live [7]	2021-Arxiv	I1	C2, C3	L1	ESIKF
Super Odom. [32]	2021-IROS	I2	C1, C3	L4	F3
Lvio-Fusion [33]	2021-IROS	I2	C1	L1	F4

[1] I1=Integration; I2=Preintegration; I3=Raw measurements.
[2] C1=Indirect; C2=Direct; C3=Depth From LiDAR; C4=Depth From Surfel.
[3] L1=LOAM Feature; L2=Tracked Plane/Line; L3=PCA based Feature; L4=PCA based Feature; L5=Surfel.
[4] See Fig. 1 for details.

RELATED WORKS

- **Multi-sensor Fusion for SLAM**
  - 1) LiDAR-Inertial-Camera Fusion for SLAM:
    - Graph-based optimization
    - filter-based methods
    - Both the graph optimization ar
    - LIC systems in Tab1.
    - Four typical pipelines in Fig1.

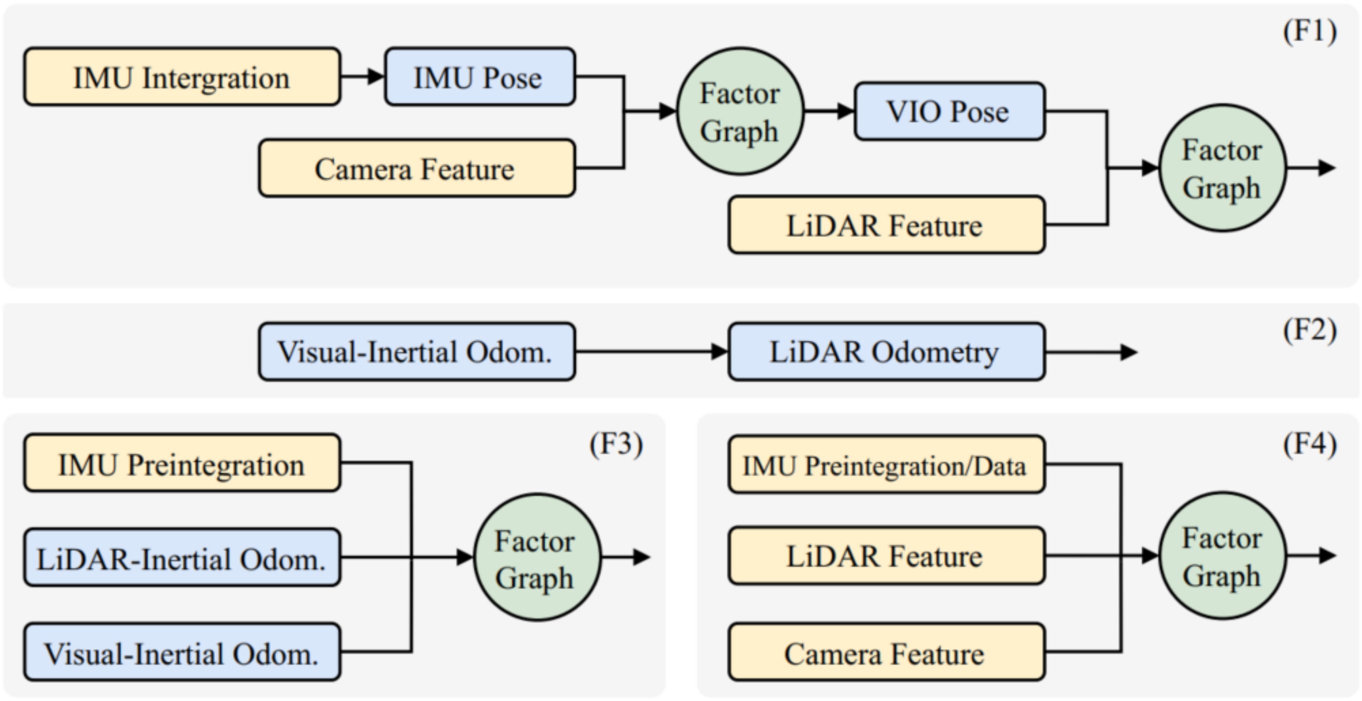


Fig. 1. Four different pipelines of LIC SLAM systems using factor-graph optimization framework. F1, F2, F3 are loosely-coupled methods, and F4 is a tightly-coupled method.

TABLE I

RELATED WORKS

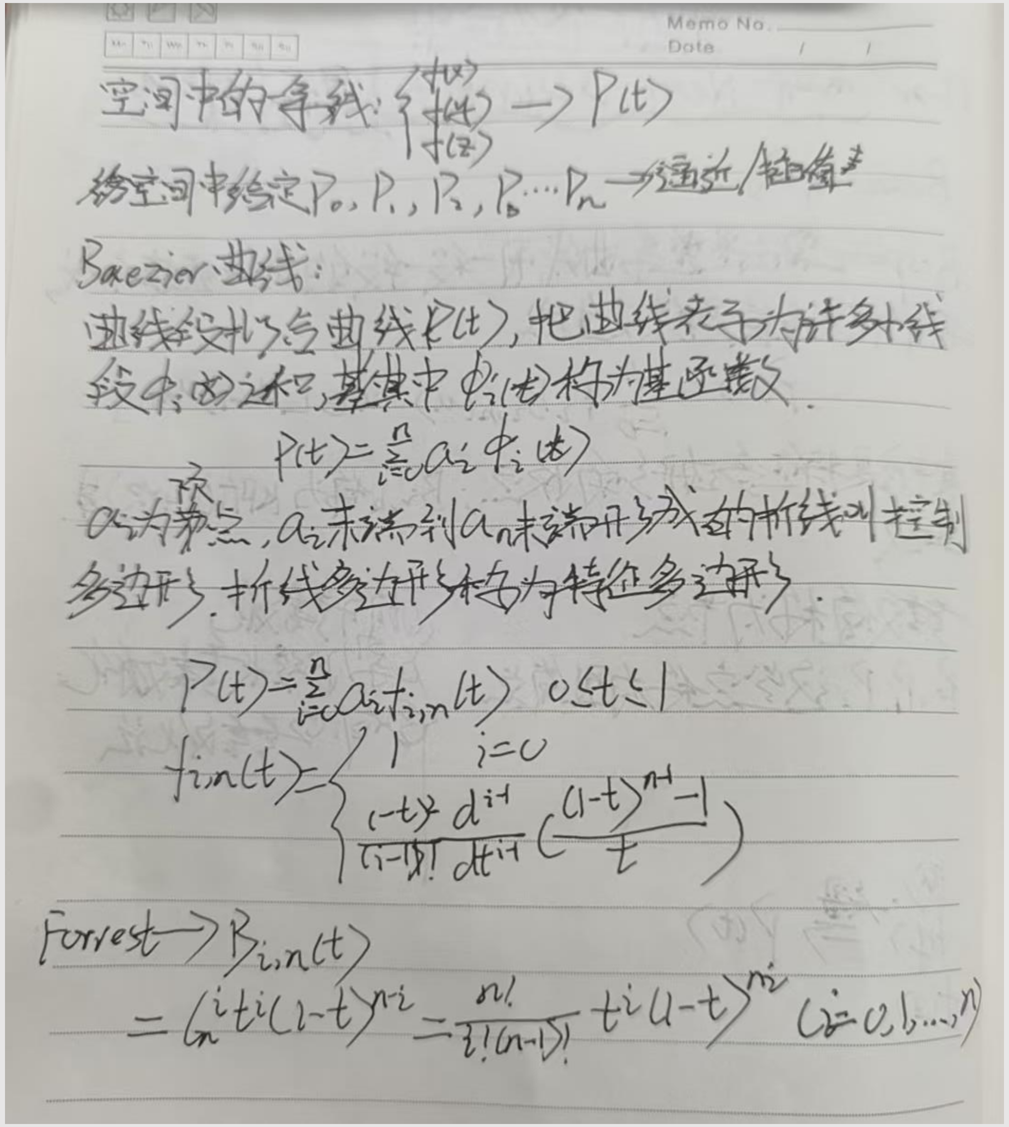
- **Multi-sensor Fusion for SLAM**
  - 1) LiDAR-Inertial-Camera Fusion for SLAM:
  - 2) Multi-LiDAR Fusion for SLAM
    - [8] LOCUS- A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time
    - [36] LOCUS 2.0: Robust and Computationally Efficient Lidar Odometry for Real-Time 3D Mapping
    - [37] MILIOM: Tightly coupled multi-input lidar-inertia odometry and mapping

RELATED WORKS

- **SLAM with Continuous-Time Trajectory**
  - B-spline based continuous-time SLAM problem is first systematically derived in [38]
    - [38] Continuous-time batch estimation using temporal basis functions
  - Thereafter, B-spline has been widely applied to SLAM-relevant applications, however they assume no timeoffset.

PRELIMINARY ON CONTINUOUS-TIME TRAJECTORY

- ***B-spline Trajectory Representation***


空间中的参数曲线 $\begin{cases} x(t) \\ y(t) \\ z(t) \end{cases} \rightarrow P(t)$

给空间中给定 $P_0, P_1, P_2, P_3 \cdots P_n$ → 逼近/插值

Baezier 曲线：

曲线段拟合各曲线 $P(t)$，把曲线表示为许多小线段 $\phi_i(t)$ 之和，其中 $\phi_i(t)$ 称为基函数.

$$P(t) = \sum_{i=0}^{n} a_i \phi_i(t)$$

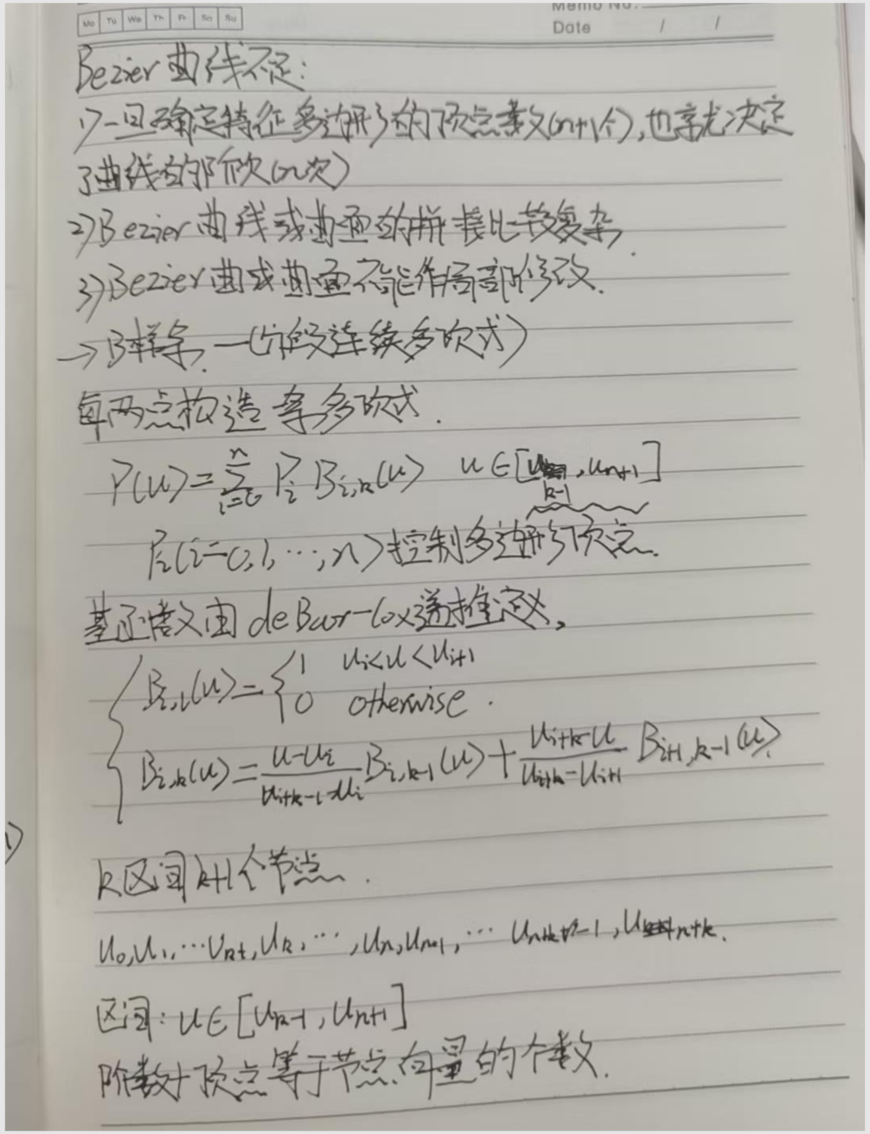
a_i 为顶点，a_0 末端到 a_n 末端形成的折线叫控制多边形. 折线多边形称为特征多边形.

$$P(t) = \sum_{i=0}^{n} a_i f_{i,n}(t) \quad 0 \le t \le 1$$

$$f_{i,n}(t) = \begin{cases} 1 & i=0 \\ \frac{(-t)^i}{(i-1)!} \frac{d^{i-1}}{dt^{i-1}} \left(\frac{(1-t)^{n-1}-1}{t} \right) \end{cases}$$

Forrest → $B_{i,n}(t)$

$$= C_n^i t^i (1-t)^{n-i} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \quad (i=0,1,\ldots,n)$$


Bezier 曲线不足：

1) 一旦确定特征多边形的顶点数（n+1个），也就决定了曲线的阶次（n次）

2) Bezier 曲线或曲面的拼接比较复杂.

3) Bezier 曲线或曲面不能作局部修改.

→ B样条（一分段连续多项式）

每两点构造一多项式.

$$P(u) = \sum_{i=0}^{n} P_i B_{i,k}(u) \quad u \in [u_{k-1}, u_{n+1}]$$

P_i $(i=0,1,\ldots,n)$ 控制多边形顶点.

基函数由 de Boor-Cox 递推定义.

$$\begin{cases} B_{i,1}(u) = \begin{cases} 1 & u_i < u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_{i,k}(u) = \frac{u-u_i}{u_{i+k-1}-u_i} B_{i,k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} B_{i+1,k-1}(u) \end{cases}$$

k 区间 n+k+1 个节点.

$u_0, u_1, \cdots u_{k-1}, u_k, \cdots, u_n, u_{n+1}, \cdots u_{n+k-1}, u_{n+k}$.

区间：$u \in [u_{k-1}, u_{n+1}]$

阶数+顶点数等于节点向量的个数.

PRELIMINARY ON CONTINUOUS-TIME TRAJECTORY

- ***B-spline Trajectory Representation***
- ***Time Derivatives of B-spline***

$$\underbrace{\mathbf{p}(t)}_{3\times 1} = \underbrace{\begin{bmatrix} \mathbf{p}_i & \mathbf{d}_1^i & \cdots & \mathbf{d}_{k-1}^i \end{bmatrix}}_{3\times k} \underbrace{\widetilde{\mathbf{M}}^{(k)}}_{k\times k} \underbrace{\mathbf{u}}_{k\times 1} \quad (1)$$

$$\mathbf{u} = \begin{bmatrix} 1 & u & \cdots & u^{k-1} \end{bmatrix}, u = (t - t_i)/(t_{i+1} - t_i)$$

with difference vectors $\mathbf{d}_j^i = \mathbf{p}_{i+j} - \mathbf{p}_{i+j-1} \in \mathbb{R}^3$. The cumulative spline matrix $\widetilde{\mathbf{M}}^{(k)}$ of uniform B-spline only depends on the B-spline order. We further define $\boldsymbol{\lambda}(t) = \widetilde{\mathbf{M}}^{(k)}\mathbf{u}$ thus Eq. (1) can be written as

$$\mathbf{p}(t) = \mathbf{p}_i + \sum_{j=1}^{k-1} \lambda_j(t) \cdot \mathbf{d}_j^i \,. \quad (2)$$

PRELIMINARY ON CONTINUOUS-TIME TRAJECTORY

- ***B-spline Trajectory Representation***
- *Time Derivatives of B-spline*

$$\mathbf{R}(t) = \mathbf{R}_i \cdot \prod_{j=1}^{k-1} \operatorname{Exp}\left(\lambda_j(t) \cdot \operatorname{Log}\left(\mathbf{R}_{i+j-1}^{-1}\mathbf{R}_{i+j}\right)\right) \tag{3}$$

where $\mathbf{R}_i \in SO(3)$ are the control points for rotation. The difference vector between two rotations is defined as $\mathbf{d}_j^i = \operatorname{Log}\left(\mathbf{R}_{i+j-1}^{-1}\mathbf{R}_{i+j}\right) \in \mathbb{R}^3$ and $\mathbf{A}_j(t) = \operatorname{Exp}\left(\lambda_j(t) \cdot \mathbf{d}_j\right)$ where omitting the i to simplify notation, Eq. (3) can be written in the following concise equation:

$$\mathbf{R}(t) = \mathbf{R}_i \cdot \prod_{j=1}^{k-1} \mathbf{A}_j(t)\,. \tag{4}$$

In this paper, we select cubic (degree = 3) B-spline and the corresponding cumulative spline matrix $\widetilde{\mathbf{M}}^{(k)}$ is

$$\widetilde{\mathbf{M}}^{(4)} = \frac{1}{6}\begin{bmatrix} 6 & 5 & 1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix}. \tag{5}$$

PRELIMINARY ON CONTINUOUS-TIME TRAJECTORY

- *B-spline Trajectory Representation*
- ***Time Derivatives of B-spline***

$$^{G}\mathbf{v}(t) = {}^{G}\dot{\mathbf{p}}_I(t) = \sum_{j=1}^{3} \dot{\lambda}_j(t) \cdot \mathbf{d}_j^i, \tag{6}$$

$$^{G}\mathbf{a}(t) = {}^{G}\ddot{\mathbf{p}}_I(t) = \sum_{j=1}^{3} \ddot{\lambda}_j(t) \cdot \mathbf{d}_j^i, \tag{7}$$

$$^{G}_{I}\dot{\mathbf{R}}(t) = \mathbf{R}_i \left(\dot{\mathbf{A}}_1 \mathbf{A}_2 \mathbf{A}_3 + \mathbf{A}_1 \dot{\mathbf{A}}_2 \mathbf{A}_3 + \mathbf{A}_1 \mathbf{A}_2 \dot{\mathbf{A}}_3 \right) \tag{8}$$

Where $\dot{\mathbf{A}}_j = \mathrm{Exp}\left(\dot{\lambda}_j(t) \cdot \mathbf{d}_j\right)$. Continuous-time trajectory of IMU in global frame {G} is denoted as $^{G}_{I}\mathbf{T}(t) = [^{G}_{I}\mathbf{R}(t), {}^{G}\mathbf{p}_I(t)]$

$$^{I}\mathbf{a}(t) = {}^{G}_{I}\mathbf{R}^{\top}(t) \left({}^{G}\mathbf{a}(t) - {}^{G}\mathbf{g} \right) \tag{9}$$

$$^{I}\boldsymbol{\omega}(t) = {}^{G}_{I}\mathbf{R}^{\top}(t) \cdot {}^{G}_{I}\dot{\mathbf{R}}(t) \tag{10}$$

Where $^{G}\mathbf{g} \in \mathbb{R}^3$ denotes the gravity vector in global frame.

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- ***Factor-Graph Optimization***
- *LiDAR Factor*
- *IMU Factor and Bias Factor*
- *Visual Factor*
- *Marginalization*

TABLE II
NOTATIONS GLOSSARY

Symbols	Meaning
$\mathbf{\Phi}(t_{\kappa-1}, t_\kappa)$	control points of B-splines in $[t_{\kappa-1}, t_\kappa)$
$\mathbf{\Phi}_R(t_\kappa)/\mathbf{\Phi}_p(t_\kappa)$	involved orientation/position control points at t_κ
$\mathbf{b}_\omega^\kappa, \mathbf{b}_a^\kappa$	biases of temporal sliding window in $[t_{\kappa-1}, t_\kappa)$
t_L, t_I, t_C	timeoffsets of LiDAR, IMU and camera, respectively
t	timestamp of trajectory or sensor measurements
$\tau = t + t_{\text{offset}}$	corrected timestamp of sensor measurements

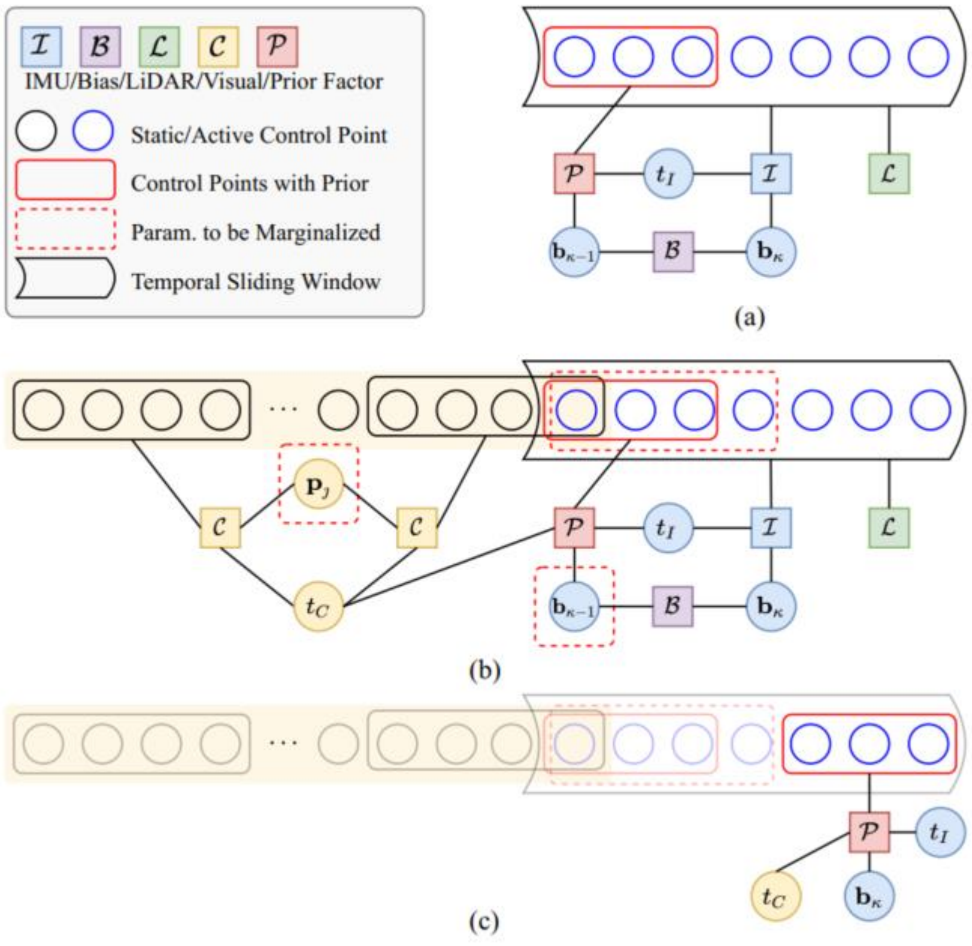


Fig. 2. Factor graphs of multi-sensor fusion. (a) A typical factor graph of LI system fusion. (b) A typical factor graph of LIC system fusion. Active control points are to be optimized, while static control points remain constant. The control points with yellow background are involved in visual keyframe sliding window. (c) After marginalization of *(b)*, the induced prior factor is involved with the latest control points, latest bias and timeoffsets.

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- ***LiDAR Factor***
- *IMU Factor and Bias Factor*
- *Visual Factor*
- *Marginalization*

A LiDAR point measurement ${}^{L}\mathbf{p}_t$ with noise $\mathbf{n}_L$, measured at time t_ℓ, is associated with a 3D plane in closest point parameterization [14, 45], ${}^{G}\boldsymbol{\pi} = {}^{G}d_\pi {}^{G}\mathbf{n}_\pi$, where ${}^{G}d_\pi$ and ${}^{G}\mathbf{n}_\pi$ denote the distance of the plane to origin and unit normal vector, respectively. We can transform LiDAR point to global frame by

$$ {}^{G}\hat{\mathbf{p}}_\ell = {}^{G}_{L}\mathbf{R}(\tau_\ell)\left({}^{L}\mathbf{p}_\ell + \mathbf{n}_L\right) + {}^{G}\mathbf{p}_L(\tau_\ell), \tag{13} $$

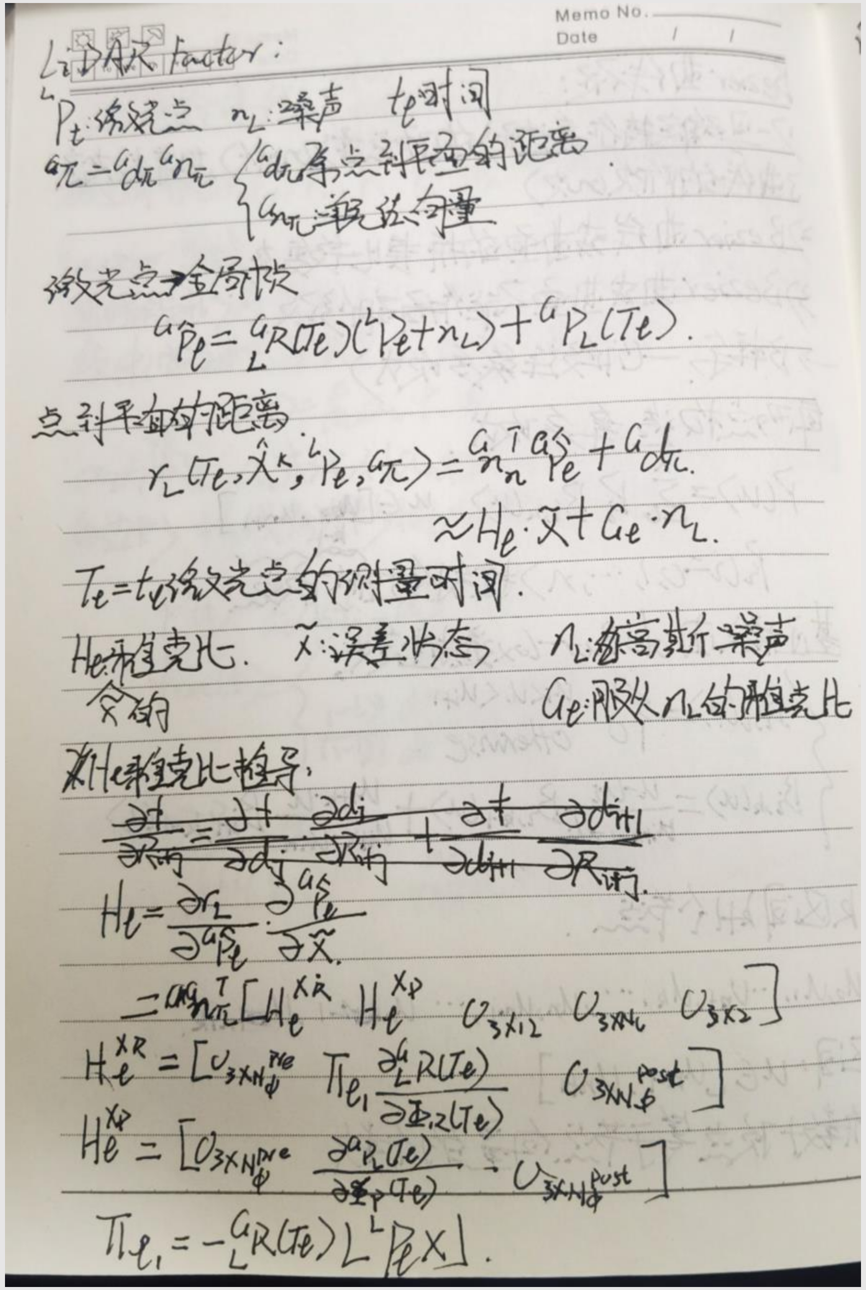
and the point-to-plane distance is given by:

$$ \begin{aligned} \mathbf{r}_L(\tau_\ell, \hat{\mathcal{X}}^{\kappa}, {}^{L}\mathbf{p}_\ell, {}^{G}\boldsymbol{\pi}) &= {}^{G}\mathbf{n}_\pi^{\top} {}^{G}\hat{\mathbf{p}}_\ell + {}^{G}d_\pi \\ &\approx \mathbf{H}_\ell \cdot \tilde{\mathbf{x}} + \mathbf{G}_\ell \cdot \mathbf{n}_L. \end{aligned} \tag{14} $$

where $\tau_\ell = t_\ell$ is the measure time of LiDAR point, $\mathbf{H}_\ell$ is Jacobian matrix w.r.t error state $\tilde{\mathbf{x}}$ (see our supplementary file for detail). $\mathbf{n}_L$ is assumed to be under independent and identically distributed (i.i.d.) white Gaussian noise in our experiments. $\mathbf{G}_\ell$ is the Jacobian with respect to $\mathbf{n}_L$ which can be easily computed.

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- ***LiDAR Factor***
- *IMU Factor and Bias Factor*
- *Visual Factor*
- *Marginalization*



Memo No.
Date / /

LiDAR factor:

${}^{L}p_t$: 激光点 n_L: 噪声 t_ℓ: 时间

${}^{G}\pi_\ell = {}^{G}d_{\pi_\ell}\,{}^{G}n_{\pi_\ell}$ ${}^{G}d_{\pi_\ell}$: 原点到平面的距离. ${}^{G}n_{\pi_\ell}$: 单位法向量

激光点→全局快

$${}^{G}\hat{p}_\ell = {}^{G}_{L}R(\tau_\ell)({}^{L}p_\ell + n_L) + {}^{G}p_L(\tau_\ell).$$

点到平面的距离:

$$r_L(\tau_\ell, \hat{x}^k, {}^{L}p_\ell, {}^{G}\pi_\ell) = {}^{G}n_{\pi_\ell}^T\,{}^{G}\hat{p}_\ell + {}^{G}d_{\pi_\ell}$$

$$\approx H_\ell \cdot \tilde{x} + G_\ell \cdot n_L.$$

$\tau_\ell = t_\ell$ 激光点的测量时间.

H_ℓ: 雅克比. $\tilde{x}$: 误差状态 n_L: 高斯噪声

交的 G_ℓ: 服从 n_L 的雅克比

对 H_ℓ 求雅克比推导:

$$\cancel{\frac{\partial f}{\partial R_{i}} = \frac{\partial f}{\partial d_j}\cdot\frac{\partial d_j}{\partial R_{i}} + \frac{\partial f}{\partial d_{i+1}}\cdot\frac{\partial d_{i+1}}{\partial R_{i}}}$$

$$H_\ell = \frac{\partial r_L}{\partial {}^{G}\hat{p}_\ell}\cdot\frac{\partial {}^{G}\hat{p}_\ell}{\partial \tilde{x}}$$

$$= {}^{G}n_{\pi_\ell}^T\left[H_\ell^{x_R}\quad H_\ell^{x_p}\quad 0_{3\times 12}\quad 0_{3\times N_c}\quad 0_{3\times 2}\right]$$

$$H_\ell^{x_R} = \left[0_{3\times N_\phi^{pre}}\quad \Pi_{\ell_1}\frac{\partial {}^{G}_{L}R(\tau_\ell)}{\partial \Phi_{i,2}(\tau_\ell)}\quad 0_{3\times N_\phi^{post}}\right]$$

$$H_\ell^{x_p} = \left[0_{3\times N_\phi^{pre}}\quad \frac{\partial {}^{G}p_L(\tau_\ell)}{\partial \Phi_p(\tau_\ell)}\quad 0_{3\times N_\phi^{post}}\right]$$

$$\Pi_{\ell_1} = -{}^{G}_{L}R(\tau_\ell)\lfloor {}^{L}p_\ell \times\rfloor.$$

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- *LiDAR Factor*
- ***IMU Factor and Bias Factor***
- *Visual Factor*
- *Marginalization*

Considering raw IMU measurements at t_m with angular velocity ${}^I\boldsymbol{\omega}_m$ and linear acceleration ${}^I\mathbf{a}_m$, and the true angular velocity and linear acceleration are denoted by ${}^I\boldsymbol{\omega}$ and ${}^I\mathbf{a}$, respectively. The following equations hold:

$$ {}^I\boldsymbol{\omega}_m = {}^I\boldsymbol{\omega}(t) + \mathbf{b}_\omega(t) + \mathbf{n}_\omega \tag{15} $$

$$ {}^I\mathbf{a}_m(t) = {}^G_I\mathbf{R}^\top(t)\left({}^G\mathbf{a}(t) - {}^G\mathbf{g}\right) + \mathbf{b}_a(t) + \mathbf{n}_a \tag{16} $$

$$ \dot{\mathbf{b}}_\omega(t) = \mathbf{n}_{b_\omega}, \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{b_a} \tag{17} $$

where $\mathbf{n}_\omega$, $\mathbf{n}_a$ are zero-mean Gaussian white noise. The gyroscope bias $\mathbf{b}_\omega$ and accelerometer bias $\mathbf{b}_a$ are modeled as random walks, driving by the white Gaussian noises $\mathbf{n}_{b_\omega}$ and $\mathbf{n}_{b_a}$, respectively.

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- *LiDAR Factor*
- ***IMU Factor and Bias Factor***
- *Visual Factor*
- *Marginalization*

and $\mathbf{n}_{b_a}$, respectively. With the timeoffset t_I of IMU between LiDAR, we have the following IMU factor

$$\begin{aligned} &\mathbf{r}_I(\tau_m, \hat{\mathcal{X}}^\kappa, {}^I\boldsymbol{\omega}_m, {}^I\mathbf{a}_m) \\ &= \begin{bmatrix} {}^I\boldsymbol{\omega}(\tau_m) - {}^I\boldsymbol{\omega}_m + \mathbf{b}_\omega^\kappa \\ {}^I\mathbf{a}(\tau_m) - {}^I\mathbf{a}_m + \mathbf{b}_a^\kappa \end{bmatrix} + \begin{bmatrix} \mathbf{n}_\omega \\ \mathbf{n}_a \end{bmatrix} \\ &\approx \mathbf{H}_{I_m}^\kappa \cdot \tilde{\mathbf{x}} + \mathbf{G}_{I_m} \cdot \mathbf{n}_I, \end{aligned} \tag{18}$$

and bias factor

$$\begin{aligned} \mathbf{r}_{I_b}(\hat{\mathcal{X}}^\kappa) &= \begin{bmatrix} \mathbf{b}_\omega^\kappa - \mathbf{b}_\omega^{\kappa-1} \\ \mathbf{b}_a^\kappa - \mathbf{b}_a^{\kappa-1} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{b_\omega} \\ \mathbf{n}_{b_a} \end{bmatrix} \\ &\approx \mathbf{H}_{I_b}^\kappa \cdot \tilde{\mathbf{x}} + \mathbf{G}_{I_b} \cdot \mathbf{n}_{I_b} \end{aligned} \tag{19}$$

where $\tau_m = t_m + t_I$ is the corrected IMU timestamp, and $\mathbf{H}_{I_m}^\kappa, \mathbf{H}_{I_b}^\kappa$ are Jacobian matrices with respect to states (see supplementary file), and $\mathbf{G}_{I_m}, \mathbf{G}_{I_b}$ are Jacobian matrices with respect to noise. By substituting the derivative of continuous-time trajectory (Eq. (9)) at time instant τ_m into Eq. (18), we can optimize the continuous-time trajectory by raw IMU measurements directly, avoiding the efforts of IMU propagation or pre-integration.

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- *LiDAR Factor*
- ***IMU Factor and Bias Factor***
- *Visual Factor*
- *Marginalization*

周四

1.组会: Continuous-Time Fixed-Lag Smoothing.

2.阅读文章篇.

3.改代码.

IMU Factor and Bias Factor:

$$r_I(T_m, \hat{X}^k, {}^I w_m, {}^I a_m)$$

$$= \begin{bmatrix} {}^I w(T_m) - {}^I w_m + b_w^k \\ {}^I a(T_m) - {}^I a_m + b_a^k \end{bmatrix} + \begin{bmatrix} n_w \\ n_a \end{bmatrix}$$

$$\approx H_{I_m}^k \cdot \tilde{X} + G_{I_m} \cdot n_I,$$

$$r_{I_b}(\hat{X}^k) = \begin{bmatrix} b_w^k - b_w^{k-1} \\ b_a^k - b_a^{k-1} \end{bmatrix} + \begin{bmatrix} n_{bw} \\ n_{ba} \end{bmatrix}$$

$$\approx H_{I_b}^k \cdot \tilde{X} + G_{I_b} \cdot n_{I_b}.$$

Visual Factor.

P_j: landmark　F_a: keyframe　t_a: timestamp

F_b　t_b

λ_j: 初始化逆深度.

ρ_j^a: 带有n_c的F_a系统观测值

F_b中估计位置为:

$$\hat{\rho}_j^b = {}^G_C T(T_b)^T \cdot {}^G_C T(T_a) \cdot \frac{1}{\lambda} \pi_c(\rho_j^a + n_c)$$

$\pi_c(\cdot)$: 投影模型

$$r_c(T_b, \hat{X}^k, \rho_j^b) = \begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix} \left(\frac{\hat{\rho}_j^b}{e_3^T \hat{\rho}_j^b} - \pi_c(\rho_j^b + n_c) \right)$$

$$\approx H_j^b \cdot \tilde{X} + G_j^b \cdot n_c.$$

Marginalization:

$$\begin{bmatrix} H_{\alpha\alpha} & H_{\alpha\beta} \\ H_{\beta\alpha} & H_{\beta\beta} \end{bmatrix} \begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} = \begin{bmatrix} b_\alpha \\ b_\beta \end{bmatrix}$$

X_α: 保留参数　X_β: 边缘化参数

$$(H_{\alpha\alpha} - H_{\alpha\beta} H_{\beta\beta}^{-1} H_{\beta\alpha}) X_\alpha = (X_\alpha - H_{\alpha\beta} H_{\beta\beta}^{-1} X_\beta)$$

$$\hat{H}_{\alpha\alpha} X_\alpha = \hat{b}_\alpha.$$

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- *LiDAR Factor*
- *IMU Factor and Bias Factor*
- ***Visual Factor***
- *Marginalization*

A landmark $\mathbf{p}_j$ observed in its anchor keyframe $\mathcal{F}_a$ at timestamp t_a and observed again in frame $\mathcal{F}_b$ at timestamp t_b, can be given the initialized inverse depth as λ_j through triangulation (as Sec. V-B1). Let $\boldsymbol{\rho}_j^a$ denotes 2D raw observation in $\mathcal{F}_a$ with noise $\mathbf{n}_c$, the estimated position of landmark in frame $\mathcal{F}_b$ is

$$\hat{\mathbf{p}}_j^b = {}_C^G\mathbf{T}(\tau_b)^\top \cdot {}_C^G\mathbf{T}(\tau_a) \cdot \frac{1}{\lambda_j} \pi_c(\boldsymbol{\rho}_j^a + \mathbf{n}_c) \tag{20}$$

where $\pi_c(\cdot)$ denotes the back projection which transforms a pixel to the normalized image plane. τ_a, τ_b are corrected timestamps to remove timeoffsets. The corresponding visual factor based on reprojection error is defined as:

$$\begin{aligned} \mathbf{r}_c(\tau_b, \hat{\mathcal{X}}^\kappa, \boldsymbol{\rho}_j^b) &= \begin{bmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \end{bmatrix} \left(\frac{\hat{\mathbf{p}}_j^b}{\mathbf{e}_3^\top \hat{\mathbf{p}}_j^b} - \pi_c\left(\boldsymbol{\rho}_j^b + \mathbf{n}_c\right) \right) \\ &\approx \mathbf{H}_j^b \cdot \tilde{\mathbf{x}} + \mathbf{G}_j^b \cdot \mathbf{n}_c \end{aligned} \tag{21}$$

where $\mathbf{e}_i$ denotes a 3×1 vector with its i-th element to be 1 and the others to be 0, and $\boldsymbol{\rho}_j^b$ describes 2D raw observation in $\mathcal{F}_b$. $\mathbf{H}_j^b$ denotes Jacobian matrix (see supplementary file).

CONTINUOUS-TIME FIXED-LAG SMOOTHING

- *Factor-Graph Optimization*
- *LiDAR Factor*
- *IMU Factor and Bias Factor*
- *Visual Factor*
- ***Marginalization***

$$\begin{bmatrix} \mathbf{H}_{\alpha\alpha} & \mathbf{H}_{\alpha\beta} \\ \mathbf{H}_{\beta\alpha} & \mathbf{H}_{\beta\beta} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\alpha} \\ \mathbf{x}_{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\alpha} \\ \mathbf{b}_{\beta} \end{bmatrix}$$

where we organize the reserved parameters in $\mathbf{x}_{\alpha}$, and $\mathbf{x}_{\beta}$ will be marginalized. Using the Schur-Complement [46], the following equation holds: 舒尔补:求解大型线性方程

$$\left(\mathbf{H}_{\alpha\alpha} - \mathbf{H}_{\alpha\beta}\mathbf{H}_{\beta\beta}^{-1}\mathbf{H}_{\beta\alpha}\right)\mathbf{x}_{\alpha} = \left(\mathbf{x}_{\alpha} - \mathbf{H}_{\alpha\beta}\mathbf{H}_{\beta\beta}^{-1}\mathbf{x}_{\beta}\right) \quad (23)$$

and by introducing new notations, we can denote the above equation by:

$$\hat{\mathbf{H}}_{\alpha\alpha}\mathbf{x}_{\alpha} = \hat{\mathbf{b}}_{\alpha} \quad (24)$$

$$\mathcal{X}_{prior}^{\kappa} = \{\mathbf{\Phi}(t_{\kappa-1}, t_{\kappa}) \cap \mathbf{\Phi}(t_{\kappa}, t_{\kappa+1}), \mathbf{b}_{\omega}^{\kappa}, \mathbf{b}_{a}^{\kappa}, t_I, t_C\}.$$

where $\mathbf{\Phi}(t_{\kappa-1}, t_{\kappa}) \cap \mathbf{\Phi}(t_{\kappa}, t_{\kappa+1})$ denotes the affected control points in next sliding window. The prior factor induced from

MULTI-SENSOR FUSION

- *LiDAR-Inertial System*
  - LiDAR Measurement Processing
  - LI Temporal Sliding Window
- *Visual System*
  - Visual Front End
  - Visual Keyframe Sliding Window
- *Extra Implementation Details*
  - Initialization
  - Online Calibration of Timeoffset
  - Loop Closure

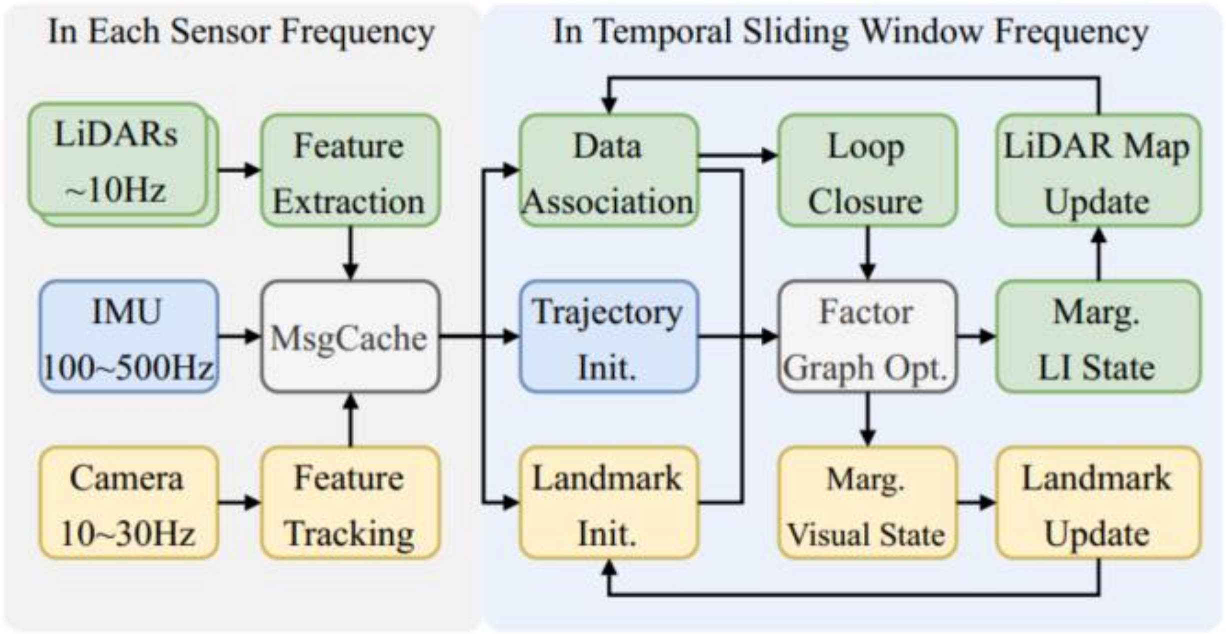


Fig. 3. The pipeline of the proposed LiDAR-Inertial-Camera fusion system. Raw IMU measurements, features of each LiDAR and tracked features of camera are cached in the MsgCache module, and measurements are fed to the sliding window at $\frac{1}{\eta\Delta t}$ Hz. After factor graph optimization, we separately marginalize LI state and visual state, and update local LiDAR map and visual landmarks. More details are provided in Sec. V.

MULTI-SENSOR FUSION

- ***LiDAR-Inertial System***
  - **LiDAR Measurement Processing: feature extraction, data association, and local map management**
    - Compute the curvature → planar features → the continuous-time trajectory compensate the points of motion distortion → point-to-plane data association → optimize estimation
  - **LI Temporal Sliding Window**
    - Propose a temporal sliding window within a constant time duration. The continuous-time trajectory of LI system is optimized and updated every delta t seconds

MULTI-SENSOR FUSION

- ***LiDAR-Inertial System***
  - **LiDAR Measurement Processing**
  - **LI Temporal Sliding Window**
- ***Visual System***
  - **Visual Front End**
    - Extract corner features → triangulate query current best-estimated continuous-time trajectory
  - **Visual Keyframe Sliding Window**

MULTI-SENSOR FUSION

- ***Visual System***
  - **Visual Front End**
  - **Visual Keyframe Sliding Window**
    - Triangulation needs to maintain a visual keyframe sliding window with a constant number of keyframes, in contrast to the constant time duration for LI temporal sliding window.
    - In practice, they determine the trajectory optimization range based on the temporal sliding window of the LI system and define the control points to be optimized as active control points.

MULTI-SENSOR FUSION

- ***Extra Implementation Details***
  - **Initialization**
    - Stationary: The new IMU biases are initialized to the value of the previous temporal sliding window bias, and the new control points are first assigned values of the neighboring control points and further initialized via factor graph optimization.
    - Motion: Use the raw IMU measurements
  - **Online Calibration of Timeoffset**
  - **Loop Closure**

MULTI-SENSOR FUSION

- ***Extra Implementation Details***
  - **Initialization**
  - **Online Calibration of Timeoffset**
    - The paper choose the LiDAR sensor as the time baseline. The main reason of choosing LiDAR as the base sensor is that local LiDAR map needs to be maintained
  - **Loop Closure**
    - Utilize Euclidean distance-based loop closure detection method.
    - After a loop closure optimization, the paper remove the prior information of states since the current best-estimated states may be away from the linearized points, resulting in inappropriate prior constraints.

EXPRTIMENT

- Datasets
  - VIRAL (Visual-Inertial-Ranging-Lidar Dataset)
  - NCD (Newer College Dataset)
  - LVI-SAM (handheld and Jackal)
  - YQ (self-collected YuQuan Dataset)
  - Vicon Room (self-collected Vicon Room)
- Evaluation:
  - Absolute Pose Error (APE)

EXPRTIMENT

- **LiDAR-Inertial Fusion: CLIO**
  - The results on VIRAL dataset are shown in Tab. III

TABLE III

THE APE (RMSE, METER) RESULTS ON VIRAL DATASET. THE BEST RESULT IS IN BOLD, AND THE SECOND BEST IS UNDERLINED.

Method	Sensor[1]	eee_01 (237m)	eee_02 (171m)	eee_03 (128m)	nya_01 (160m)	nya_02 (249m)	nya_03 (315m)	sbs_01 (202m)	sbs_02 (184m)	sbs_03 (199m)	average
LIO-SAM[2] [50]	L, I	0.075	0.069	0.101	0.076	0.090	0.137	0.089	0.083	0.140	0.096
MILIOM (horz. LiDAR)[2] [37]	L, I	0.104	0.065	0.063	0.083	0.072	0.058	0.076	0.081	0.088	0.077
VIRAL (horz. LiDAR)[2] [51]	L, I	0.064	0.051	0.060	0.063	0.042	0.039	0.051	0.056	0.060	0.054
CLINS (w/o loop) [17]	L, I	0.059	0.030	0.029	**0.034**	**0.040**	**0.039**	0.029	0.031	0.033	0.036
CLIO (w/o loop)	L, I	**0.030**	**0.023**	0.028	0.042	0.053	0.042	**0.028**	0.032	**0.030**	**0.034**
CLIC (w/o loop)	L, I, C	0.030	0.029	**0.028**	0.040	0.054	0.041	0.029	**0.031**	0.033	0.035
MILIOM (2 LiDARs)[2] [37]	L2, I	0.067	0.066	0.052	0.057	0.067	0.042	0.066	0.082	0.093	0.066
VIRAL (2 LiDARs)[2] [51]	L2, I, C	0.060	0.058	0.037	0.051	0.043	**0.032**	0.048	0.062	0.054	0.049
CLIO2 (w/o loop)	L2, I	0.040	**0.021**	0.031	0.030	0.037	0.034	**0.033**	0.037	0.044	0.034
CLIC2 (w/o loop)	L2, I, C	**0.038**	0.025	**0.030**	**0.029**	**0.036**	0.035	0.034	**0.035**	**0.043**	**0.034**

(1) Sensors L,I,C are abbreviations of LiDAR, IMU, camera, respectively, L2 represents using two LiDAR sensors.

(2) Results are from [51]. The horz. LiDAR in table means only the horizontal LiDAR is used for odometry.

EXPRTIMENT

- **LiDAR-Inertial Fusion: CLIO**
  - Evaluations in large-scale scenarios on the LVI-SAM dataset and NCD dataset (Tab V and IV)

TABLE V
THE APE (RMSE, METER) RESULTS ON LVI-SAM DATASET.

Method	Sensor	Handheld (1642s)	Jackal (2182s)
LIO-SAM (w/o loop)	L, I	53.62	3.54
CLIO (w/o loop)	L, I	fail	3.43
LVI-SAM (w/o loop)	L, I, C	7.87	4.05
CLIC (w/o loop)	L, I, C	**2.56**	**2.55**
LIO-SAM (w/ loop)	L, I	fail	1.52
CLIO (w/ loop)	L, I	fail	1.01
LVI-SAM (w/ loop)	L, I, C	0.83	**0.67**
CLIC (w/ loop)	L, I, C	0.65	0.88
CLIC (w/ loop, w/ calib)	L, I, C	**0.56**	0.84

EXPRTIMENT

- **LiDAR-Inertial Fusion: CLIO**
  - Evaluations in large-scale scenarios on the LVI-SAM dataset and NCD dataset (Tab V and IV)

TABLE IV
THE APE (RMSE, METER) RESULTS ON NCD DATASET.

Method	NCD_01 (1530s / 1609m)	NCD_02 (2656s / 3063m)	NCD_06 (120s / 97m)
LIO-SAM (w/o loop)	1.660	**2.305**	0.272
CLIO (w/o loop)	**0.792**	2.686	**0.091**
LIO-SAM (w/ loop)	0.544	0.592	0.272
CLIO (w/ loop)	**0.408**	**0.381**	**0.091**

EXPRTIMENT

- **LiDAR-Inertial Fusion: CLIO**
  - The pose estimation accuracy of a system with the fusion of IMU and two LiDARs (Tab III)

TABLE III

THE APE (RMSE, METER) RESULTS ON VIRAL DATASET. THE BEST RESULT IS IN BOLD, AND THE SECOND BEST IS UNDERLINED.

Method	Sensor(1)	eee_01 (237m)	eee_02 (171m)	eee_03 (128m)	nya_01 (160m)	nya_02 (249m)	nya_03 (315m)	sbs_01 (202m)	sbs_02 (184m)	sbs_03 (199m)	average
LIO-SAM(2) [50]	L, I	0.075	0.069	0.101	0.076	0.090	0.137	0.089	0.083	0.140	0.096
MILIOM (horz. LiDAR)(2) [37]	L, I	0.104	0.065	0.063	0.083	0.072	0.058	0.076	0.081	0.088	0.077
VIRAL (horz. LiDAR)(2) [51]	L, I	0.064	0.051	0.060	0.063	0.042	0.039	0.051	0.056	0.060	0.054
CLINS (w/o loop) [17]	L, I	0.059	0.030	0.029	**0.034**	**0.040**	**0.039**	0.029	0.031	0.033	0.036
CLIO (w/o loop)	L, I	**0.030**	**0.023**	0.028	0.042	0.053	0.042	**0.028**	0.032	**0.030**	**0.034**
CLIC (w/o loop)	L, I, C	0.030	0.029	**0.028**	0.040	0.054	0.041	0.029	**0.031**	0.033	0.035
MILIOM (2 LiDARs)(2) [37]	L2, I	0.067	0.066	0.052	0.057	0.067	0.042	0.066	0.082	0.093	0.066
VIRAL (2 LiDARs)(2) [51]	L2, I, C	0.060	0.058	0.037	0.051	0.043	**0.032**	0.048	0.062	0.054	0.049
CLIO2 (w/o loop)	L2, I	0.040	**0.021**	0.031	0.030	0.037	0.034	**0.033**	0.037	0.044	0.034
CLIC2 (w/o loop)	L2, I, C	**0.038**	0.025	**0.030**	**0.029**	**0.036**	0.035	0.034	**0.035**	**0.043**	**0.034**

(1) Sensors L,I,C are abbreviations of LiDAR, IMU, camera, respectively, L2 represents using two LiDAR sensors.

(2) Results are from [51]. The horz. LiDAR in table means only the horizontal LiDAR is used for odometry.

EXPRTIMENT

- **LiDAR-Inertial-Camera Fusion: CLIC**
  - The evaluation results on our self collected indoor and outdoor datasets are shown in Tab. VI and VII

TABLE VI

THE APE (RMSE, METER) RESULTS ON YQ DATASET (OUTDOOR). THE BEST RESULT OF LIC SYSTEM WITHOUT (OR WITH) LOOP CLOSURE IS UNDERLINED (OR IN BOLD).

Sequence	LVI-SAM (w/o loop)	LIC-Fusion 2.0 (w/o loop)	CLIC (w/o loop)	LVI-SAM (w/ loop)	CLIC (w/ loop)
YQ-01 (1005m)	<u>1.614</u>	3.300	1.826	**1.227**	1.537
YQ-02 (1021m)	1.790	1.804	<u>1.626</u>	1.610	**1.363**
YQ-03 (1058m)	3.017	2.798	<u>2.616</u>	1.886	**1.701**
YQ-04 (1233m)	3.163	2.697	<u>2.450</u>	2.402	**1.917**
YQ-05 (673m)	1.721	1.550	<u>1.537</u>	8.465	**1.439**
YQ-06 (1644m)	3.753	3.862	<u>3.082</u>	3.682	**1.607**
YQ-07 (414m)	0.800	1.306	<u>0.761</u>	0.795	**0.761**

EXPRTIMENT

- **LiDAR-Inertial-Camera Fusion: CLIC**
  - The evaluation results on our self collected indoor and outdoor datasets are shown in Tab. VI and VII

TABLE VII

THE APE (RMSE, METER) RESULTS ON VICON ROOM DATASET (INDOOR). THE BEST RESULT IS IN BOLD.

Seq.	LVI-SAM (w/o loop)	LIC-Fusion 2.0 (w/o loop)	CLIC (w/o loop)
Seq1 (43m)	0.393	**0.033**	0.080
Seq2 (84m)	0.232	0.096	**0.073**
Seq3 (34m)	0.364	**0.052**	0.073
Seq4 (53m)	0.395	0.092	**0.089**
Seq5 (50m)	0.155	**0.044**	0.171
Seq6 (88m)	0.459	**0.046**	0.128

EXPRTIMENT

- **LiDAR-Inertial-Camera Fusion: CLIC**
  - Some representative estimated trajectories of different methods aligned with the ground truth trajectories.

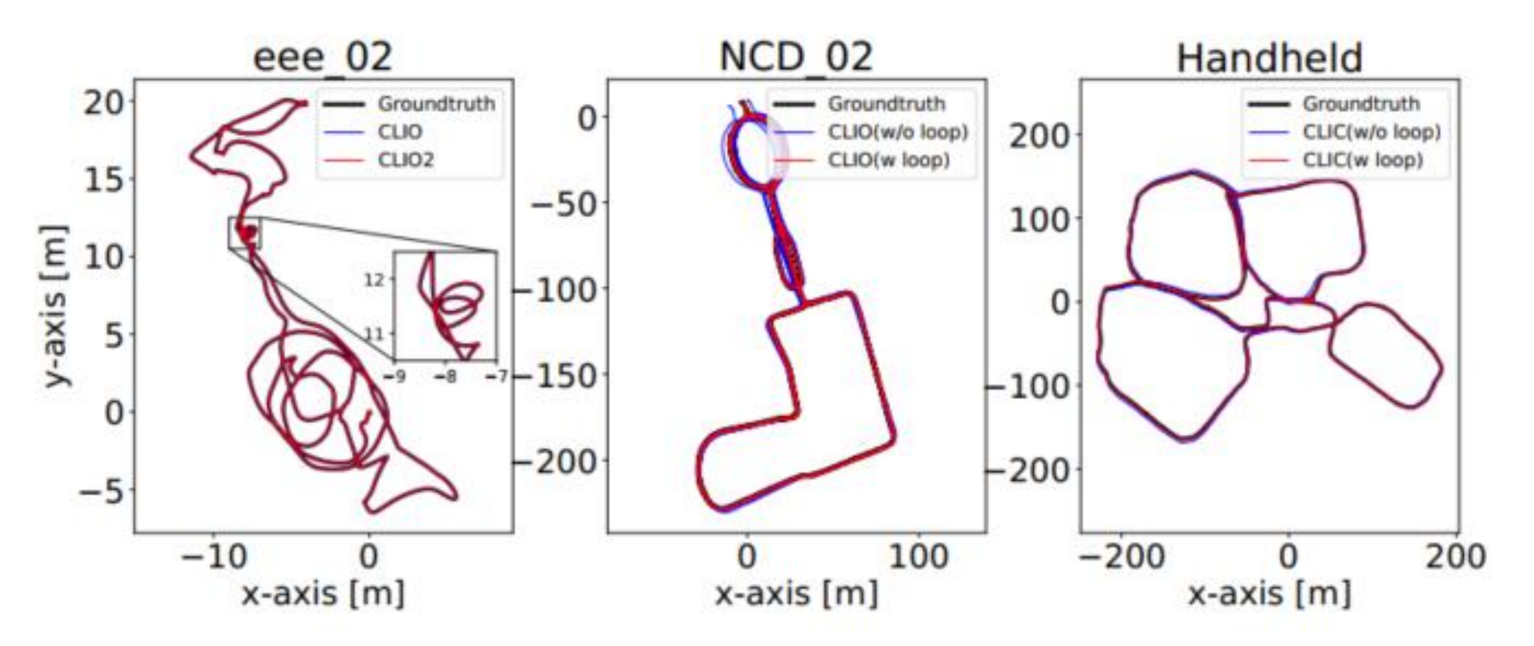


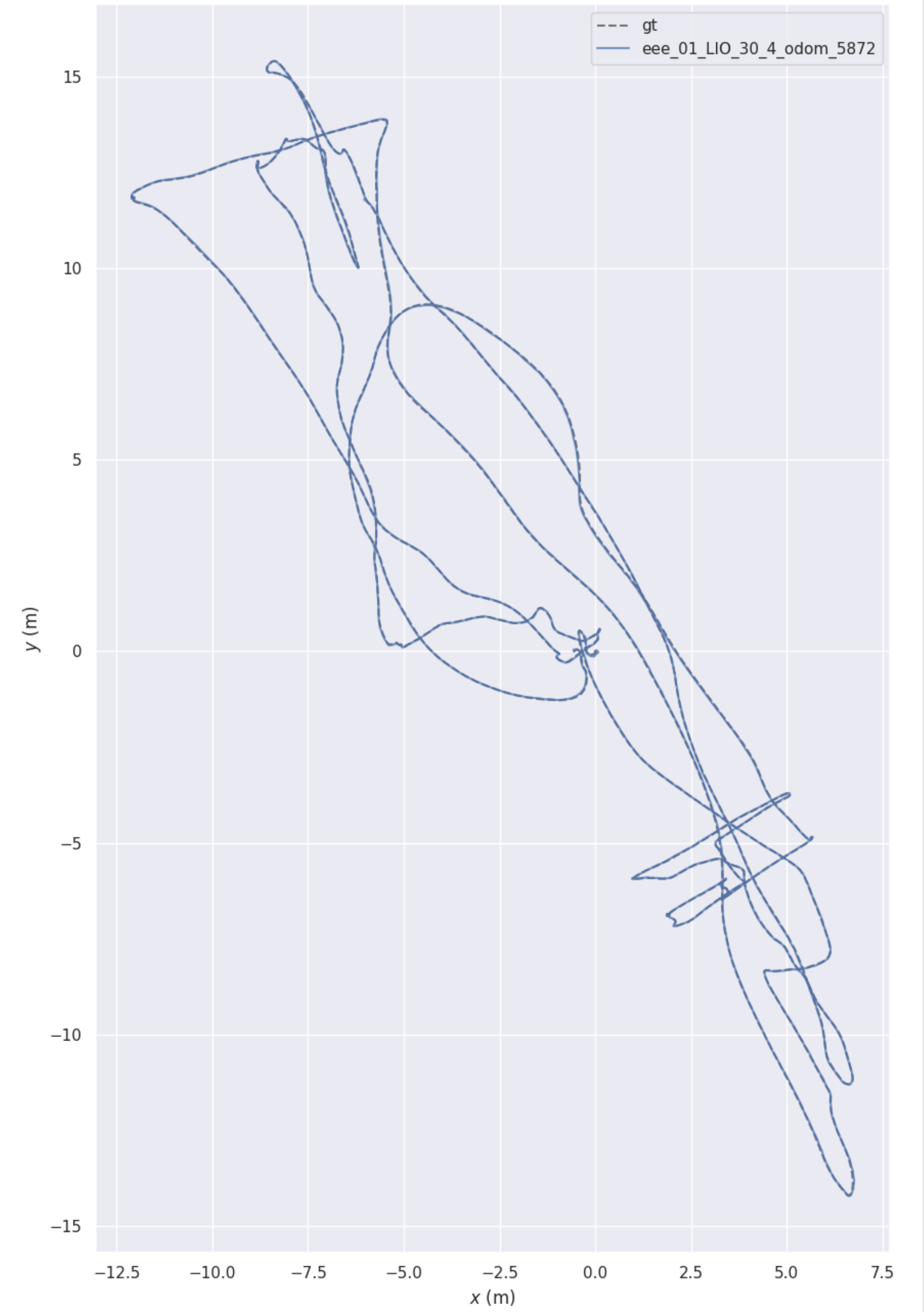
Fig. 9. The estimated trajectories compared to the groundtruth in eee_02 sequence of VIRAL dataset, NCD_02 sequence of NCD dataset and Handheld sequence of LVI-SAM dataset.

EXPRTIMENT

- **LiDAR-Inertial-Camera Fusion: CLIC**
  - Some representative estimated trajectories of different methods aligned with the ground truth trajectories.

```
dell@dell-Precision-3660:~/ORB_SLAM3$ evo_ape tum /home/dell/catkin_clic/src/clic/data/eee_01_LIO_30_4_od
--------------------------------------------------------------------------------
Loaded 39759 stamps and poses from: /home/dell/catkin_clic/src/clic/data/eee_01_LIO_30_4_odom_5872.txt
Loaded 6616 stamps and poses from: /home/dell/catkin_clic/src/clic/eee_01/ntuviral_gt-master/eee_01/gt.txt
--------------------------------------------------------------------------------
Synchronizing trajectories...
Found 6616 of max. 6616 possible matching timestamps between...
        /home/dell/catkin_clic/src/clic/data/eee_01_LIO_30_4_odom_5872.txt
and:    /home/dell/catkin_clic/src/clic/eee_01/ntuviral_gt-master/eee_01/gt.txt
..with max. time diff.: 0.01 (s) and time offset: 0.0 (s).
--------------------------------------------------------------------------------
Aligning using Umeyama's method...
Rotation of alignment:
[[-0.89335988 -0.44882241 -0.02160007]
 [ 0.44903519 -0.89349381 -0.00601761]
 [-0.0165987  -0.01507508  0.99974858]]
Translation of alignment:
[-3.09376701e-01  1.20267539e-04  1.92186907e-01]
Scale correction: 1.0
--------------------------------------------------------------------------------
Compared 6616 absolute pose pairs.
Calculating APE for translation part pose relation...
--------------------------------------------------------------------------------
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

       max  0.135643
      mean  0.026870
    median  0.024768
       min  0.001313
      rmse  0.030499
       sse  6.154271
       std  0.014429
```

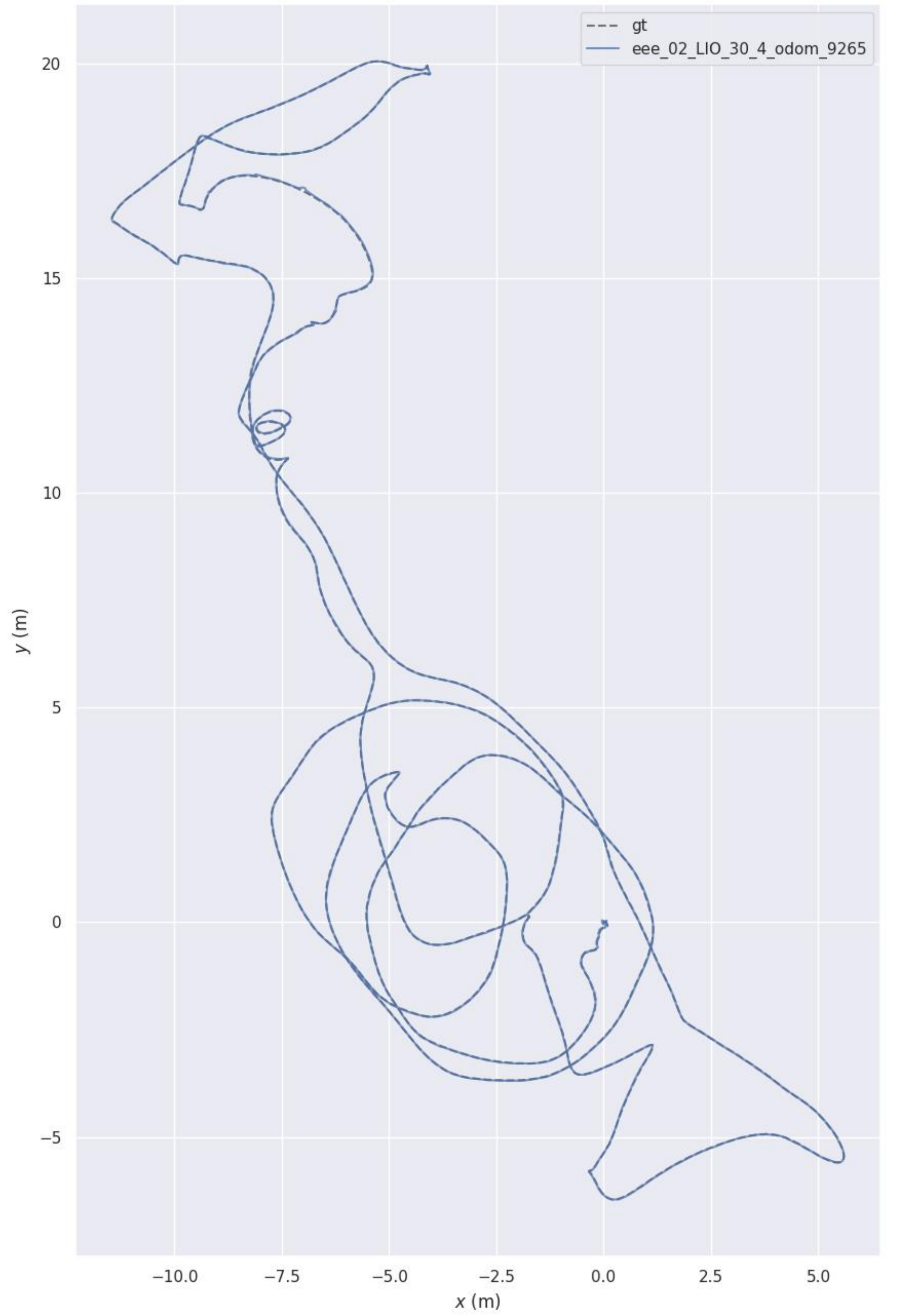


EXPRTIMENT

- **LiDAR-Inertial-Camera Fusion: CLIC**
  - Some representative estimated trajectories of different methods aligned with the ground truth trajectories.

```
--------------------------------------------------------------------------------
Loaded 31983 stamps and poses from: /home/dell/catkin_clic/src/clic/data/eee_02_LIO_30_4_odom_9265.txt
Loaded 5512 stamps and poses from: /home/dell/catkin_clic/src/clic/ntuviral_gt-master/eee_02/gt.txt
--------------------------------------------------------------------------------
Synchronizing trajectories...
Found 5512 of max. 5512 possible matching timestamps between...
        /home/dell/catkin_clic/src/clic/data/eee_02_LIO_30_4_odom_9265.txt
and:    /home/dell/catkin_clic/src/clic/ntuviral_gt-master/eee_02/gt.txt
..with max. time diff.: 0.01 (s) and time offset: 0.0 (s).
--------------------------------------------------------------------------------
Aligning using Umeyama's method...
Rotation of alignment:
[[ 0.45962738 -0.88759719 -0.03023068]
 [ 0.88805513  0.45971639  0.00434913]
 [ 0.01003726 -0.02884549  0.99953349]]
Translation of alignment:
[-0.30124371  0.00663976  0.20136034]
Scale correction: 1.0
--------------------------------------------------------------------------------
Compared 5512 absolute pose pairs.
Calculating APE for translation part pose relation...
--------------------------------------------------------------------------------
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

      max   0.101847
     mean   0.020504
   median   0.016569
      min   0.000335
     rmse   0.025052
      sse   3.459391
      std   0.014395
```



EXPRTIMENT

- **Online Temporal Calibration**
  - Shows the error of estimated timeoffset over time.

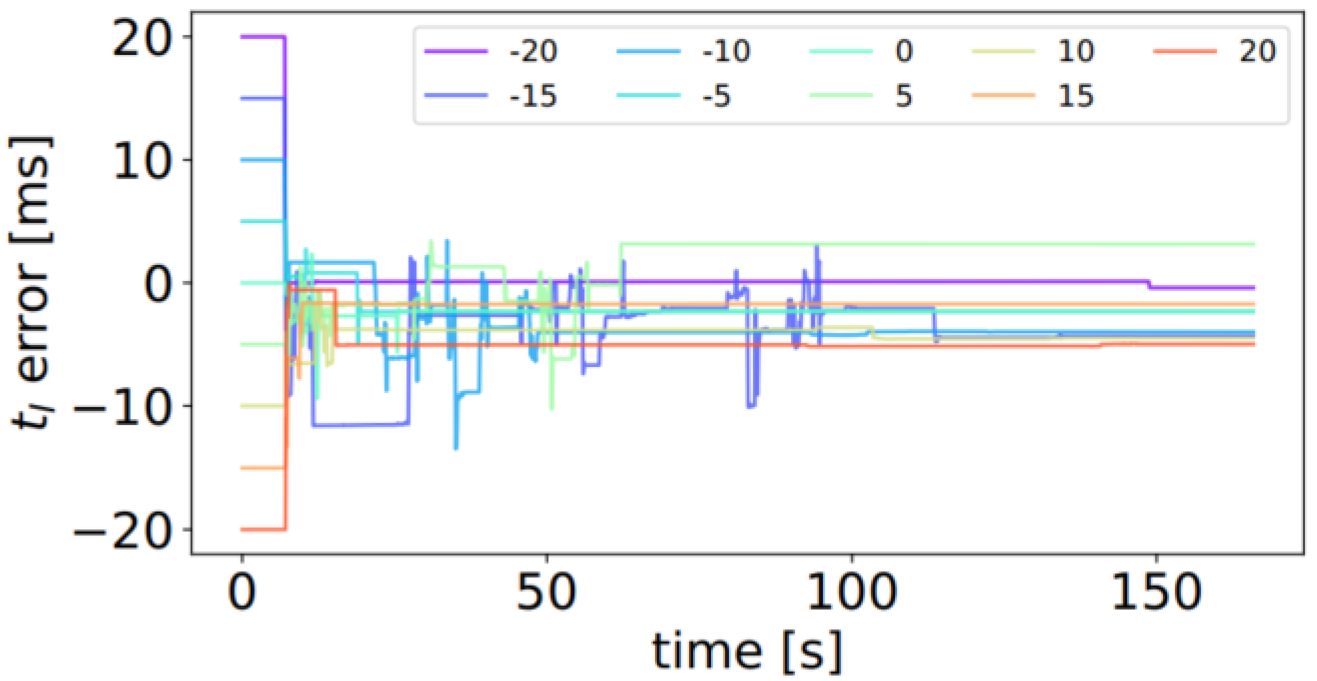


Fig. 11. Temporal calibration error of CLIC system in NCD_01 sequence of NCD dataset. Although we start from various initial values of timeoffsets, the estimates of timeoffsets are able to quickly converge.

EXPRTIMENT

- **Runtime analysis**
  - The method is implemented in C++ and executed on the desktop PC with an Intel i7-7700K and 32GB RAM, and Tab. VIII summarizes the time consumption of CLINS , CLIO and CLIC.

TABLE VIII
TIMING OF DIFFERENT MODULES OF CLINS, CLIO AND CLIC IN EEE_01 SEQUENCE WITH A DURATION OF 397 SECONDS.

	CLINS	CLIO	CLIC
Update Local Map	17.39	11.56	11.52
Update Trajectory	1184.34	58.55	80.64
Update Prior	0.00	8.45	8.44
Others	400.13	139.27	193.97
Total Time Cost (second)	1601.86	217.82	294.57

Conclusion

This paper exploits continuous-time fixed-lag smoothing for asynchronous multi-sensor fusion in a factor graph framework. Specifically, they propose to probabilistically marginalize old states and measurement out of the sliding window, and derive analytic Jacobians for continuous-time optimization. On line temporal calibration between sensors is also naturally supported in the continuous-time estimator

Advantages

- Propose continuous-time fixed-lag smoothing method for multi-sensor in a factor-graph optimization framework.
- Design LI SLAM and LIC SLAM systems at a variety of sensor combinations.
- Timeoffsets between different sensors can be online calibrated.

Problems

- CONTINUOUS-TIME FIXED-LAG SMOOTHING
  - Derivation of the factor graph formula

THANK YOU FOR LISTENING

QUESTIONS?

SUGGESTIONS?

CRITICS?