

R²LIVE: A Robust, Real-time, LiDAR-Inertial-Visual tightly-coupled state Estimator and mapping

Jiarong Lin, Chunran Zheng, Wei Xu, and Fu Zhang

Abstract—In this letter, we propose a robust, real-time tightly-coupled multi-sensor fusion framework, which fuses measurements from LiDAR, inertial sensor, and visual camera to achieve robust and accurate state estimation. Our proposed framework is composed of two parts: the filter-based odometry and factor graph optimization. To guarantee real-time performance, we estimate the state within the framework of error-state iterated Kalman-filter, and further improve the overall precision with our factor graph optimization. Taking advantage of measurements from all individual sensors, our algorithm is robust enough to various visual failure, LiDAR-degenerated scenarios, and is able to run in real time on an on-board computation platform, as shown by extensive experiments conducted in indoor, outdoor, and mixed environments of different scale (see attached video¹). Moreover, the results show that our proposed framework can improve the accuracy of state-of-the-art LiDAR-inertial or visual-inertial odometry. To share our findings and to make contributions to the community, we open source our codes on our Github².

Index Terms—SLAM, Mapping, Sensor Fusion.

I. INTRODUCTION

With the capacity of estimating ego-motion in six degrees of freedom (DOF) and simultaneously building dense and high precision maps of surrounding environments, LiDAR-based SLAM has been widely applied in the field of autonomous driving vehicles [1], drones [2, 3], and etc. With the development of LiDAR technologies, the emergence of low-cost LiDARs (e.g., Livox LiDAR [4]) makes LiDAR more accessible. Following this trend, a number of related works [5]–[9] draw the attention of the community to this field of research. However, the accuracy of LiDAR-based SLAM methods would significantly degrade or even fail in those scenarios with few available geometry features, which is more critical for those LiDARs with small FoV [10]. In such scenarios, adding visual features could increase the system’s robustness and accuracy. In this work, we propose a LiDAR-inertial-visual fusion framework to obtain the state estimation of higher robustness and accuracy. The main contributions of our work are:

- We develop a tightly-coupled LiDAR-inertial-visual system for real-time state estimation and mapping. Building on several key techniques from current state-of-the-art

Manuscript received February 24, 2021; Revised May 21, 2021; Accepted June 15, 2021.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers comments. This work is supported by DJI under the grant number 200009538.

J. Lin, C. Zheng, W. Xu and F. Zhang are with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR, China. {jiarong.lin, zhengcr, wuwei, fuzhang}@hku.hk
Digital Object Identifier (DOI): see top of this page.

¹https://youtu.be/9lqRHmLN_MA

²<https://github.com/hku-mars/r2live>

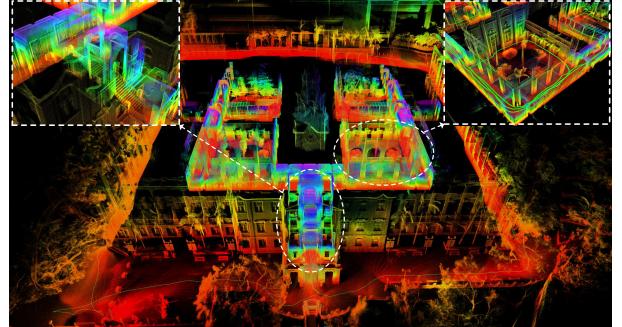


Fig. 1: We use our proposed system to reconstruct a high-precision, large-scale, indoor-outdoor, dense 3D map of the main building of the University of Hong Kong (HKU). The green path is the computed trajectory and the 3D points are colored by height.

LiDAR-inertial and visual-inertial navigation systems, the system consists of a high-rate filter-based odometry and a low-rate factor graph optimization. The filter-based odometry fuses the measurements of LiDAR, inertial, and camera sensors within an error-state iterated Kalman filter to achieve real-time performance. The factor graph optimization refines a local map of keyframe poses and visual landmark positions.

- We conduct various experiments showing that the developed system is able to run in various challenging scenarios with aggressive motion, sensor failure, and even in narrow tunnel-like environments with a large number of moving objects and small LiDAR field of view. It achieves more accurate and robust results than the current existing baselines and is accurate enough to be used to reconstruct large-scale, indoor-outdoor dense 3D maps of building structures (see Fig. 1).
- We open-source the system, which could benefit the whole robotic community and serve as a baseline for comparison in this field of research.

II. RELATED WORK

In this section, we review existing works closely related to our work, including LiDAR-only odometry and mapping, LiDAR-Inertial fusion and LiDAR-Inertial-Visual methods.

A. LiDAR Odometry and Mapping

Zhang *et al* [11] first proposed a LiDAR odometry and mapping framework, LOAM, that combines ICP method [12] with point-to-plane and point-to-edge distance. It achieves good odometry and mapping performance by running the two modules at different rates. To make the algorithm run in real time at a computation limited platform, Shan *et al* [13] propose a lightweight and ground-optimized LOAM (LeGO-LOAM), which discards unreliable features in the step of ground plane

segmentation. These works [11, 13] are mainly based on multi-line spinning LiDARs. Our previous works [10, 14] develop an accurate and robust algorithm by considering the low-level physical properties of solid-state LiDARs with small FOV. However, these methods solely based on LiDAR measurements are very vulnerable to degenerated scenarios with few geometric features.

B. LiDAR-Inertial Odometry

The existing works of LiDAR-Inertial fusion can be categorized into two classes: loosely-coupled and the tightly-coupled. Loosely-coupled methods deal with two sensors separately to infer their motion constraints while tightly-coupled approaches directly fuse LiDAR and inertial measurements through joint-optimization. In LOAM [11], IMU is used to de-skew the LiDAR scan and give a motion prior for scan-to-scan matching. It is a loosely-coupled method since the IMU is not engaged in the scan registration. Compared with loosely-coupled methods, tightly-coupled methods show higher robustness and accuracy, therefore drawing increasing research interests recently. For example, authors in [15] propose LIOM which uses a graph optimization based on the priors from LiDAR-Inertial odometry and a rotation-constrained refinement method. Compared with the former algorithm, LIO-SAM [16] optimizes a sliding window of keyframe poses in a factor graph to achieve higher accuracy. Similarly, Li *et al.* propose LiLi-OM [17] for both conventional and solid-state LiDARs based on sliding window optimization. LINS [18] is the first tightly-coupled LIO that solves the 6 DOF ego-motion via iterated Kalman filtering. To lower the high computation load in calculating the Kalman gain, our previous work FAST-LIO [5] proposes a new formula of the Kalman gain computation, the resultant computation complexity depends on the state dimension instead of measurement dimension. The work achieves up to 50 Hz odometry and mapping rate while running on embedded computers onboard a UAV.

C. LiDAR-Inertial-Visual Odometry

On the basis of LiDAR-Inertial methods, LiDAR-Inertial-Visual odometry incorporating measurements from visual sensors shows higher robustness and accuracy. Zhang and Singh in [19] propose a LiDAR-inertial-visual system that uses a loosely-coupled Visual-Inertial odometry (VIO) as the motion model for initializing the LiDAR mapping subsystem. Similarly, Shao *et al* in [20] propose a stereo visual-inertial LiDAR SLAM that incorporates the tightly-coupled stereo visual-inertial odometry with the LiDAR mapping and LiDAR enhanced visual loop closure. The overall system is still a loosely-coupled LiDAR-inertial-visual fusion since the LiDAR data are not jointly optimized together with the visual-inertial measurements. There are also some RGB-D-inertial SLAM systems, such as [20, 21]. Being designed for RGB-D cameras, their effectiveness for LiDARs are not verified due to the significant difference in measurement pattern, range and density between RGB-D cameras and LiDAR sensors. In the work of [22], the LiDAR measurements are used to provide depth information for camera images, forming a system similar

to RGB-D camera that can leverage existing visual SLAM works such as ORB-SLAM2 [23]. This is also a loosely-coupled method as it ignores the direct constraints on state imposed by LiDAR measurements. Zuo *et al* [24] propose a LIC-fusion framework combining IMU measurements, sparse visual features, and LiDAR plane and edge features with online spatial and temporal calibration based on the MSCKF framework, which exhibits higher accuracy and robustness than other state-of-the-art methods in their experiment results. In quick succession, their further work termed LIC-Fusion 2.0 [25] refines a novel plane-feature tracking algorithm across multiple LiDAR scans within a sliding-window to make LiDAR scan matching more robust.

Our proposed system tightly fuses the data of different types of sensors, estimates the state within the framework of error-state iterated Kalman-filter achieving real-time performance, meanwhile improves the overall accuracy with a factor graph optimization. Various real-world dataset results show that our system is more accurate and robust than current state-of-the-art LiDAR-inertial (FAST-LIO [5]), visual-inertial (VINS-mono [26]), and LiDAR-visual systems (CamVox [22]). We also open source on our implementation on Github³, which are not available for prior LiDAR-inertial-visual SLAM systems [19, 20, 24, 25].

III. THE OVERVIEW OF OUR SYSTEM

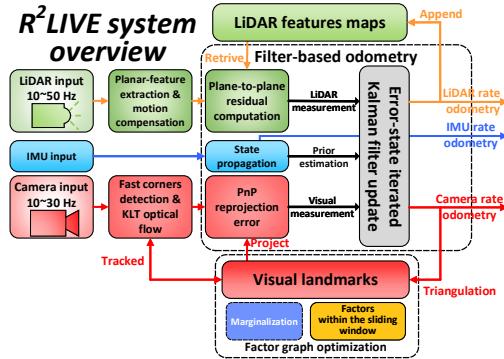


Fig. 2: The overview of our system.

The overview of our system is shown in Fig. 2. For each scan of LiDAR input, we extract planar feature points, compensate the motion distortion (i.e., de-skewing), and compute the point-to-plane residual. Similarly, for an incoming image, we detect fast corners, track them with a map of visual landmarks, and compute the re-projection error. Then, the LiDAR point-to-plane residual and the image re-projection errors are tightly fused with the IMU propagation in an error-state iterated Kalman filter (see Section IV). The fused pose is then used to append new points in current LiDAR scan to point-cloud map and triangulate new visual landmarks. To further improve the accuracy of visual measurements, we leverage the factor graph optimization to refine the visual landmarks within a local sliding window (see Section V).

³<https://github.com/hku-mars/r2live>



Fig. 9: We evaluate our algorithm in a Hong Kong MTR station consisting of cluttered lobby and very long narrow tunnels, as shown in (a). The tunnel is up to 190 meters long and is filled with moving pedestrians, making it extremely challenging for both LiDAR-based and camera-based SLAM methods. (b): the map built by our system is well aligned with the street map of the MTR station. (c) Trajectory comparison among our system “R2LIVE”, the LiDAR-inertial system “Fast-LIO” , and visual-inertial system “VINS-Mono”. “R2LIVE-LIO” and “R2LIVE-VIO” are of the LiDAR-inertial and visual-inertial subsystem of our proposed system. The starting point is marked with ♦ while the ending point of each trajectory is marked with ★. “VINS-Mono” stopped at middle way due to the failure of feature tracking.

Our handheld device for data collection is shown in Fig. 6 (a), which includes the power supply unit, the onboard DJI *manifold-2c*⁷ computation platform (equipped with an *Intel i7-8550u* CPU and 8 GB RAM), a *FLIR Blackfly BFS-u3-13y3c* global shutter camera, and a *LiVOX AVIA*⁸ LiDAR. The FoV of the camera is $82.9^\circ \times 66.5^\circ$ while the FoV of LiDAR is $70.4^\circ \times 77.2^\circ$. To quantitatively evaluate the precision of our algorithm (Section VI-E), we install a Differential-GPS (D-GPS) real-time kinematic (RTK) system⁹ on our device, shown in Fig. 6 (b).



Fig. 7: We evaluate the robustness of our algorithm under aggressive motion (sequence in a~d) and sensor failure by intentionally blocking the camera (e) and LiDAR sensor (f).

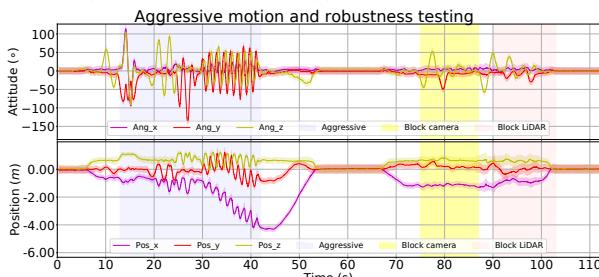


Fig. 8: Our estimated pose and its 3σ bound with 5 times amplification for better visualization (the light-colored area around the trajectory) of Experiment-1. The shaded area in blue, yellow and red represent the phase of aggressive motion, camera-failure and LiDAR-failure, respectively. In all phases, the estimated trajectory can closely track the actual motion.

B. Experiment-1: Robustness evaluation with aggressive motion and sensor failure

In this experiment, we evaluate the robustness of our algorithm in aggressive motion (see Fig. 7 (a~d)), where the maximum angular speed reaches up to $300^\circ/s$ (see the gyroscope readings in Fig. 8). In addition, we also simulate

drastic sensor failures by intentionally blocking the camera (see Fig. 7(e)) and LiDAR (see Fig. 7(f)).

The results of our estimated attitude and position are shown in Fig. 8, where we shade the different testing phases with different colors. It is seen that our estimated trajectory can tightly track the actual motion even in severe rotation and translation. Even when the camera or LiDAR is occasionally blocked, the estimated trajectory is still very smooth and exhibits no noticeable degradation. These results show that our algorithm is robust to endure aggressive motion and sensor failures. We refer readers to the accompanying video on YouTube¹⁰ showing details of the experiment.

C. Experiment-2: Robustness evaluation in a narrow tunnel-like environments with moving objects

In this experiment, we challenge one of the most difficult scenarios in the scope of camera-based and LiDAR-based SLAM, a MTR station, the *HKU station*¹¹. It is a typical narrow tunnel-like environment (see the left figure of Fig. 9 (a)), with the maximum length reaching 190 meters. Moreover, there are many moving pedestrians frequently showing in the sensor views. All these factors make the SLAM extremely challenging: 1) the long tunnel structure significantly reduces the geometric features for LiDAR-based SLAM, especially when walking along the tunnel or turning the LiDAR around at one end of the tunnel; 2) The highly repeated visual feature (pattern on the floor) easily fail the matching and tracking of visual features in visual SLAM; 3) The many moving pedestrians could cause outliers to the already few point- and visual-features.

Despite these challenges, our system is robust enough to survive in this scene. Due to the lack of GPS ground-truth, we align our point cloud data with the *HKU station street map*¹² to verify its long-term accuracy. As shown in Fig. 9 (b), our mapped point cloud matches the station street map tightly, demonstrating that our localization is of high robustness and accuracy. On the other hand, as shown in Fig. 9 (c), the

⁷<https://www.dji.com/manifold-2>

⁸<https://www.livoxtech.com/avia>

⁹<https://www.dji.com/d-rtk>

¹⁰https://youtu.be/9lqRHmlN_MA

¹¹https://en.wikipedia.org/wiki/HKU_station

¹²<https://www.mtr.com.hk/archive/en/services/maps/hku.pdf>

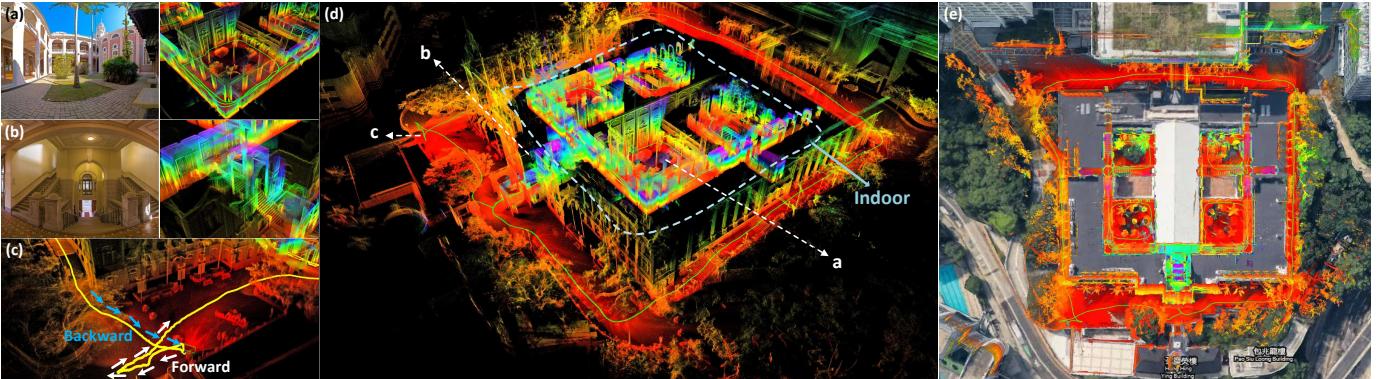


Fig. 10: The reconstructed 3D maps in Experiment-3 are shown in (d), and the detail point cloud with the correspondence panorama images are shown in (a) and (b). (c) shows that our algorithm can close the loop by itself (returning the starting point) without any additional processing (e.g. loop closure). In (e), we merge our map with the satellite image to further examine the accuracy of our system.



Fig. 11: Five different traveling trajectories in Experiments-3. Without any explicit loop closure, all of them can close the loop (i.e. return to zero) after traveling around 900 meters.

trajectory estimated by Vins-Mono¹³, Fast-LIO¹⁴, our LiDAR-inertial and visual-inertial subsystem significantly drift.

D. Experiment-3: High precision mapping large-scale indoor & outdoor urban environment

In this experiment, we show that our proposed method is able to reconstruct a dense 3D, high precision, large-scale indoor-outdoor map of urban environments. We collect data of the *HKU main building* exterior and interior 5 times with slightly different traveling paths but all starting from and ending at the same location (see Fig. 11). What worth to be mentioned is, without any additional processing (i.e., loop detection), all of our 5 trajectories can still close the loop by themselves (see Fig. 10 (c) and Fig. 11) after traveling around 900 meters, which demonstrates that our proposed method is extremely low drift. The real-time reconstructed 3D maps along one path is shown in Fig. 10, in which the clear and high-quality point cloud demonstrates that our proposed method is of high accuracy. Finally, we overlay our map on the satellite image and find them match tightly (see Fig. 10 (e)).

E. Experiment-4: Quantitative evaluation of precision using D-GPS RTK

In this experiment, we collect two fast-rotating (rotation speed reaching $130^\circ/s$ and $200^\circ/s$) sequences with a real-

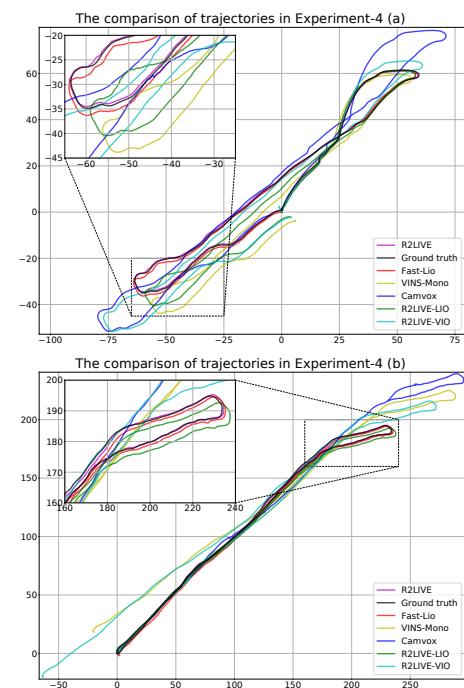


Fig. 12: Comparison results on two sequences with D-GPS ground-truth in Experiments-4.

time Differential-GPS (D-GPS) Kinematic (RTK) system providing the ground-truth trajectory. We compare the trajectory estimated by VINS-Mono (IMU+Camera) [26], Fast-LIO (IMU+LiDAR) [5], Camvox (loosely-coupled LiDAR+Visual) [22] with the ground-truth in Fig. 12, it is seen that the trajectory estimated by our factor graph optimization agrees the best with the ground-truth trajectory in both sequences. To have more quantitative comparisons, we compute the relative rotation error (RPE) and relative translation error (RTE) [30] over all possible sub-sequences of length (50,100,150,...,300) meters are shown in Table. I.

F. Run time analysis

The average time consumption in experiment 1~4 are listed in TABLE. II, which demonstrates that R²LIVE can achieve real-time localization on both the desktop PC and embedded computation platform. Notice that the factor graph optimization runs on a separate thread and therefore is allowed to run at a lower rate.

¹³<https://github.com/HKUST-Aerial-Robotics/VINS-Mono>

¹⁴https://github.com/hku-mars/FAST_LIO/

For conveniently, we omit the $(\cdot)|_{\delta\check{\mathbf{x}}_{k+1}=0}$ in the following derivation, and we have:

$$\mathbf{H}_s^c = -\frac{\partial \boldsymbol{\pi}({}^C \mathbf{P}_s)}{\partial {}^C \mathbf{P}_s} \cdot \frac{\partial \mathbf{P}_C(\check{\mathbf{x}}_{k+1} \boxplus \delta\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s)}{\partial \delta\check{\mathbf{x}}_{k+1}} \quad (\text{S8})$$

$$\mathbf{F}_{\mathbf{P}_s} = -\frac{\partial \boldsymbol{\pi}({}^C \mathbf{P}_s)}{\partial {}^C \mathbf{P}_s} \cdot \frac{\partial \mathbf{P}_C(\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s)}{\partial {}^G \mathbf{P}_s} \quad (\text{S9})$$

where:

$$\frac{\partial \boldsymbol{\pi}({}^C \mathbf{P}_s)}{\partial {}^C \mathbf{P}_s} = \frac{1}{{}^C P_{sz}} \begin{bmatrix} f_x & 0 & -f_x \frac{{}^C P_{sx}}{{}^C P_{sz}} \\ 0 & f_y & -f_y \frac{{}^C P_{sy}}{{}^C P_{sz}} \end{bmatrix} \quad (\text{S10})$$

$$\frac{\partial \mathbf{P}_b(\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s)}{\partial {}^G \mathbf{P}_s} = \left({}^G \check{\mathbf{R}}_{I_{k+1}} {}^I \check{\mathbf{R}}_C \right)^T \quad (\text{S11})$$

According to Section A, we have the following approximation of $\mathbf{P}_C(\check{\mathbf{x}}_{k+1} \boxplus \delta\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s)$:

$$\begin{aligned} & \mathbf{P}_C(\check{\mathbf{x}}_{k+1} \boxplus \delta\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s) \\ &= \left({}^G \check{\mathbf{R}}_{I_{k+1}} \text{Exp}\left({}^G \delta\check{\mathbf{r}}_{I_{k+1}}\right) {}^I \check{\mathbf{R}}_{C_{k+1}} \text{Exp}\left({}^I \delta\check{\mathbf{r}}_{C_{k+1}}\right) \right)^T {}^G \mathbf{P}_s \\ & - \left({}^I \check{\mathbf{R}}_{C_{k+1}} \text{Exp}\left({}^I \delta\check{\mathbf{r}}_{C_{k+1}}\right) \right)^T \left({}^I \check{\mathbf{p}}_{C_{k+1}} + {}^I \delta\check{\mathbf{p}}_{C_{k+1}} \right) \\ & - \left({}^G \check{\mathbf{R}}_{I_{k+1}} \text{Exp}\left({}^G \delta\check{\mathbf{r}}_{I_{k+1}}\right) {}^I \check{\mathbf{R}}_{C_{k+1}} \text{Exp}\left({}^I \delta\check{\mathbf{r}}_{C_{k+1}}\right) \right)^T \left({}^G \check{\mathbf{p}}_{I_{k+1}} + {}^G \delta\check{\mathbf{p}}_{I_{k+1}} \right) \\ & \approx \mathbf{P}_b(\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s) - \left({}^I \check{\mathbf{R}}_{C_{k+1}} \right)^T \left({}^G \check{\mathbf{R}}_{I_{k+1}} \right)^T {}^G \delta\check{\mathbf{p}}_{I_{k+1}} \\ & + \left({}^I \check{\mathbf{R}}_{C_{k+1}} \right)^T \left[{}^G \check{\mathbf{R}}_{I_{k+1}}^T ({}^G \mathbf{P}_s - {}^G \check{\mathbf{p}}_{I_{k+1}}) \right]_x {}^G \delta\check{\mathbf{r}}_{I_{k+1}} \\ & + \left[\mathbf{P}_b(\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s) \right]_x {}^I \delta\check{\mathbf{r}}_{C_{k+1}} - \left({}^I \check{\mathbf{R}}_C \right)^T {}^I \delta\check{\mathbf{p}}_{C_{k+1}} \end{aligned} \quad (\text{S12})$$

With this, we can derive:

$$\frac{\partial \mathbf{P}_C(\check{\mathbf{x}}_{k+1} \boxplus \delta\check{\mathbf{x}}_{k+1}, {}^G \mathbf{P}_s)}{\partial \delta\check{\mathbf{x}}_{k+1}} = [\mathbf{M}_A \ \mathbf{M}_B \ \mathbf{M}_C \ \mathbf{M}_D \ \mathbf{0}_{3 \times 12}] \quad (\text{S13})$$

$$\begin{aligned} \mathbf{M}_A &= \left({}^I \check{\mathbf{R}}_{C_{k+1}} \right)^T \left[{}^G \check{\mathbf{R}}_{I_{k+1}}^T ({}^G \mathbf{P}_s - {}^G \check{\mathbf{p}}_{I_{k+1}}) \right]_x \\ \mathbf{M}_B &= - \left({}^I \check{\mathbf{R}}_{C_{k+1}} \right)^T \left({}^G \check{\mathbf{R}}_{I_{k+1}} \right)^T \\ \mathbf{M}_C &= \left[{}^C \mathbf{P}_s \right]_x \\ \mathbf{M}_D &= - \left({}^I \check{\mathbf{R}}_C \right)^T \end{aligned} \quad (\text{S14})$$

Substituting (S10), (S11) and (S13) into (S8) and (S9), we finish the computation of \mathbf{H}_s^c and $\mathbf{F}_{\mathbf{P}_s}$.