

R3LIVE

论文情况

- 标题: R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package
- 作者: Jiarong Lin, Fu Zhang
- 会议: ICRA, 2022
- 源码: <https://github.com/hku-mars/r3live>

1 Introduction

前作 R2LIVE[1] 提取 LiDAR 和稀疏视觉特征，在误差-状态迭代卡尔曼滤波（ESIKF）框架内通过最小化特征重投影误差来估计状态，以达到实时性能，同时通过滑动窗口优化来提高整体视觉映射精度。R2LIVE 可以在各种具有挑战性的场景中运行，包括激进的运动、传感器故障，甚至是具有许多移动物体和小激光雷达视场的狭窄隧道样环境。

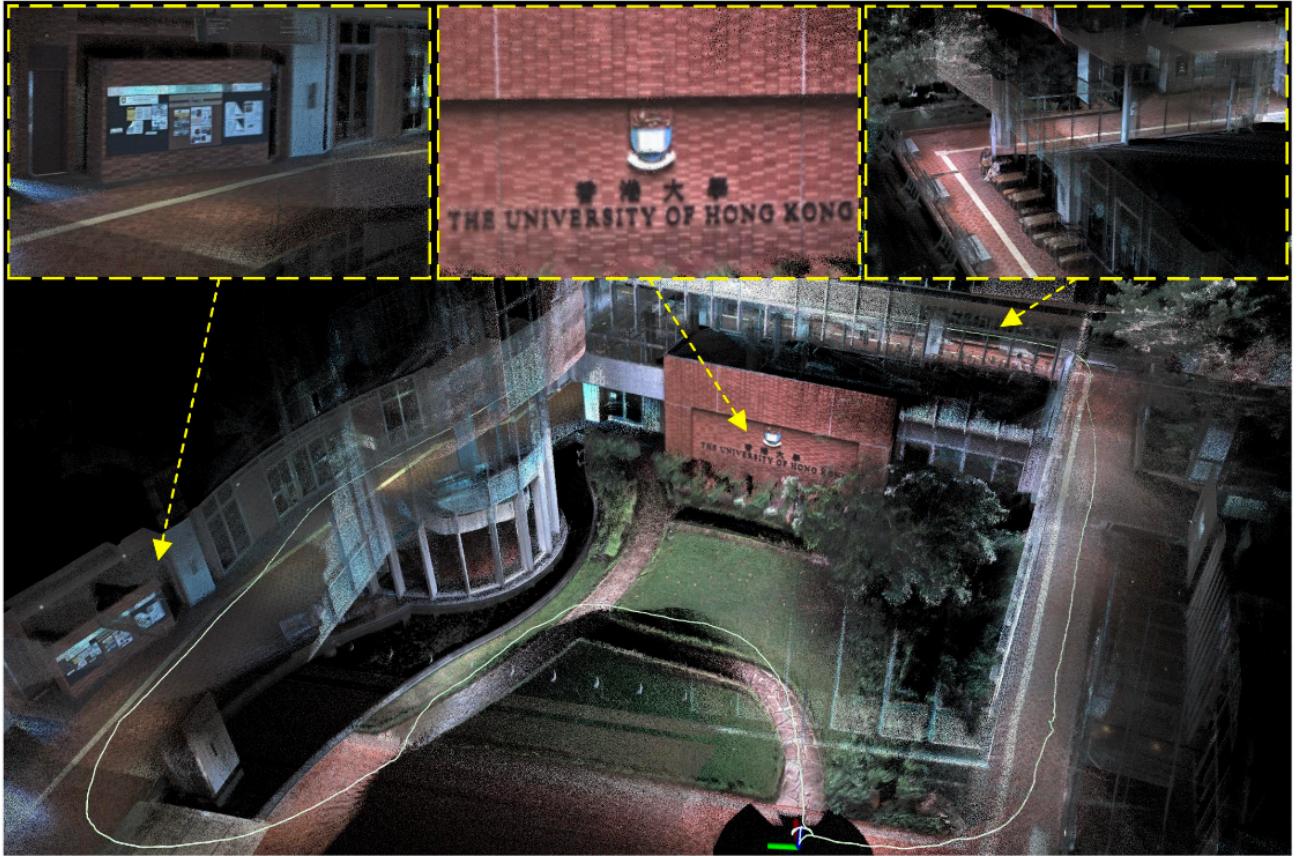


Fig. 1: Our developed R^3LIVE is able to reconstruct a dense, 3D, RGB-colored point cloud of the traveled environment in real-time. The white path is the figure is our traveling trajectory for collecting the data.

为进一步提高之前工作 R2LIVE 的实时性、鲁棒性和准确性，本文提出一种新的传感器融合框架 R3LIVE，与 R2LIVE 具有完全不同的 VIO 子系统：

- 本文提出一种实时同时定位、地图绘制和着色框架。该框架由用于重建几何结构的 LIO 和用于纹理渲染的 VIO 组成。整个系统能够实时重建稠密的 3D RGB 颜色的环境点云，如图-1。
- 基于 RGB 点云地图提出 VIO 系统。VIO 通过最小化观测到的地图点的 RGB 颜色和其在当前帧测量到的颜色的光测误差（photometric error）来估计当前状态。

2 The System Overview

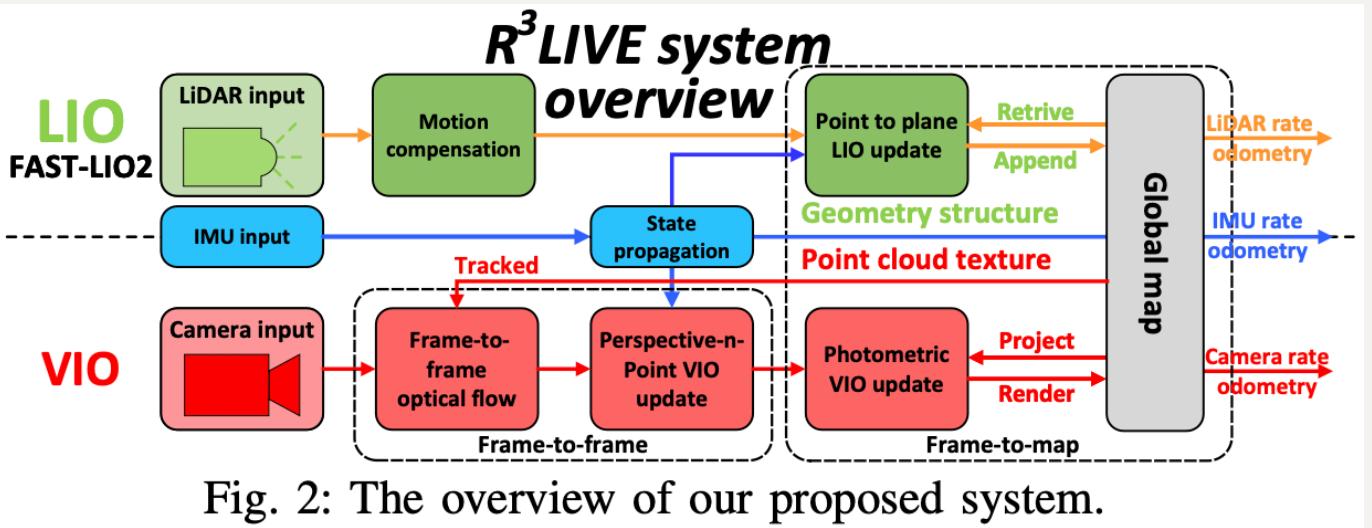


Fig. 2: The overview of our proposed system.

系统如图-2，框架包含两个子系统：LIO 子系统和 VIO 子系统。

- LIO 子系统构建全局地图的几何结构，对输入的 LiDAR 扫描进行配准，并通过最小化点面残差来估计系统状态。
- VIO 子系统构建地图的纹理，用输入图像渲染每个点的 RGB 颜色，通过最小化帧到帧的 PnP 重投影误差和帧到地图的光度误差来更新系统状态。

3 Notation

符号表达列表：

- \boxplus/\boxminus : 向量运算符号
- ${}^G(\cdot)$: (\cdot) 的全局帧值
- ${}^C(\cdot)$: (\cdot) 的当前帧值
- $\text{Exp}(\cdot)/\text{Log}(\cdot)$: 旋转向量和旋转矩阵的罗德里格斯转换
- $\delta(\cdot)$: (\cdot) 的残差
- $\Sigma_{(\cdot)}$: 向量 (\cdot) 的协方差矩阵
- $\mathbf{b}_g, \mathbf{b}_a$: 陀螺仪和加速度计的偏置
- ${}^G\mathbf{g}$: 全局系下的重力加速度
- ${}^G\mathbf{v}$: 全局系下的线速度
- $({}^G\mathbf{R}_I, {}^G\mathbf{p}_I)$: IMU 在全局系下的旋转和位置
- $({}^I\mathbf{R}_C, {}^I\mathbf{p}_C)$: IMU 和相机间的外参
- \mathbf{x} : 全状态向量

- $\hat{\mathbf{x}}$: \mathbf{x} 的先验
- $\check{\mathbf{x}}$: 在每一次 ESIKF 迭代中 \mathbf{x} 的当前状态

3.1 State Vector

定义完全状态 $\mathbf{x} \in \mathbb{R}^{29}$ 为:

$$\mathbf{x} = [{}^G\mathbf{R}_I^T, {}^G\mathbf{p}_I^T, {}^G\mathbf{v}^T, \mathbf{b_g}^T, \mathbf{b_a}^T, {}^G\mathbf{g}^T, {}^I\mathbf{R}_C^T, {}^I\mathbf{p}_C^T, {}^I\mathbf{t}_C, \phi^T]^T \quad (1)$$

${}^I\mathbf{t}_C$ 为 IMU 和相机间的时间偏移, $\phi = [f_x, f_y, c_x, c_y]^T$ 为相机内参。

3.2 Maps Representation

地图由体素和点构成:

(1) 体素

体素大小为 $0.1m \times 0.1m \times 0.1m$, 如果体素在近期 (如 1 秒内) 添加过新的点, 则标记为 *activated*, 否则标记为 *deactivated*。

(2) 点

一个点 \mathbf{P} 为向量

$$\mathbf{P} = [{}^Gp_x, {}^Gp_y, {}^Gp_z, c_r, c_g, c_b]^T = [{}^G\mathbf{p}^T, \mathbf{c}^T]^T \quad (2)$$

${}^G\mathbf{p}^T = [{}^Gp_x, {}^Gp_y, {}^Gp_z]^T$ 为全局 3D 坐标, $\mathbf{c} = [c_r, c_g, c_b]$ 为点的 RGB 色彩。同时记录点 3×3 的坐标协方差 Σ_p 和色彩协方差 Σ_c , 以及点被创建和渲染的时间戳。

4 LiDAR-Inertial Odometry Subsystem

如图-2，R3LIVE 的 LIO 子系统[1, 2] 构建全局地图的几何结构：

- 对于传入的雷达扫描，帧内连续运动造成的运动失真由 IMU 反向传播补偿；
- 然后，利用最小化点面残差的 ESIKF 估计系统状态；
- 最后，该扫描的点被添加到全局地图中，并标记相应的体素为 *activated* 或 *deactivated*。

全局地图中累积的 3D 点形成几何结构，它也用于为 VIO 子系统提供深度。

5 Visual-Inertial Odometry Subsystem

VIO 子系统渲染全局地图的纹理，通过最小化光度误差来估计系统状态：将一定数量的被跟踪的点从全局地图投影到当前图像，然后通过最小化这些点的光度误差，在 ESIKF 内迭代估计系统状态。

被跟踪的地图点是稀疏的，最小化这些稀疏点上的光度误差通常需要建立输入图像的金字塔。然而，金字塔不是平移或旋转不变的，这也是需要估计的。在提出的框架中，利用单个地图点的颜色来计算光度误差。

为了确保鲁棒和快速收敛，设计了一个两步管道，如图-2：

- 首先利用帧到帧的光流跟踪地图点，并通过最小化 PnP 投影误差优化系统状态。
- 然后，通过最小化被跟踪点之间的帧到地图的光度误差进一步优化系统状态估计。
- 利用融合的状态估计和原始输入图像，进行纹理渲染以更新全局地图中点的颜色。

5.1 Frame-to-Frame Visual-Inertial Odometry

假设从前一帧 \mathbf{I}_{k-1} 跟踪到 m 个地图点 $\mathcal{P} = \mathbf{P}_1, \dots, \mathbf{P}_m$, 这些点在 \mathbf{I}_{k-1} 上的投影为 $\{\rho_{1_{k-1}}, \dots, \rho_{m_{k-1}}\}$, 使用 LK 光流寻找它们在当前帧 \mathbf{I}_k 的位置。然后使用 ESIKF 优化投影误差估计状态。

(1) PnP 投影误差

如图-3, 以第 s 个点 $\mathbf{P}_s = [{}^G \mathbf{p}_s^T, \mathbf{c}^T]^T$ 为例。

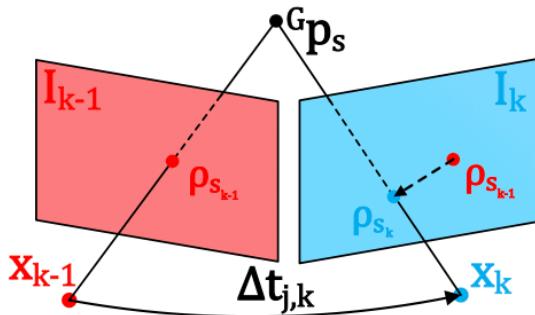


Fig. 3: Our frame-to-frame ESIKF VIO update the system's state with minimizing the PnP projection error.

投影误差 $\mathbf{r}(\check{\mathbf{x}}_k, \rho_{s_k}, {}^G \mathbf{p}_s)$ 定义为:

$${}^C \mathbf{p}_s = [{}^C \mathbf{p}_{x_s}, {}^C \mathbf{p}_{y_s}, {}^C \mathbf{p}_{z_s}] = \left({}^G \check{\mathbf{R}}_{I_k} \cdot {}^I \check{\mathbf{R}}_{C_k} \right)^T \cdot {}^G \mathbf{p}_s - {}^I \check{\mathbf{R}}_{C_k}^T \cdot {}^I \check{\mathbf{p}}_{C_k} - \left({}^G \check{\mathbf{R}}_{I_k} \cdot {}^I \check{\mathbf{R}}_{C_k} \right)^T \cdot {}^G \check{\mathbf{p}}_{I_k} \quad (3)$$

$$\mathbf{r}(\check{\mathbf{x}}_k, \rho_{s_k}, {}^G \mathbf{p}_s) = \rho_{s_k} - \pi({}^C \mathbf{p}_s, \check{\mathbf{x}}_k) \quad (4)$$

$$\begin{aligned} \pi({}^C \mathbf{p}_s, \check{\mathbf{x}}_k) &= \left[\check{f}_{x_k} \frac{{}^C \mathbf{p}_{x_s}}{{}^C \mathbf{p}_{z_s}} + \check{c}_{x_k}, \check{f}_{y_k} \frac{{}^C \mathbf{p}_{y_s}}{{}^C \mathbf{p}_{z_s}} + \check{c}_{y_k} \right]^T \\ &\quad + \frac{{}^I \check{t}_{C_k}}{\Delta t_{k-1, k}} (\rho_{s_k} - \rho_{s_{k-1}}) \end{aligned} \quad (5)$$

$\Delta t_{k-1, k}$ 为 \mathbf{I}_{k-1} 和 \mathbf{I}_k 之间的时间区间, (5) 的前一部分为小孔投影函数, 后一部分为矫正因子[3]。

式(4)中的测量噪声包含像素追踪误差 ρ_{s_k} 和地图点位置误差 ${}^G \mathbf{p}_s$:

$${}^G \mathbf{p}_s = {}^G \mathbf{p}_s^{\text{gt}} + \mathbf{n}_{\mathbf{p}_s}, \mathbf{n}_{\mathbf{p}_s} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{n}_{\mathbf{p}_s}}) \quad (6)$$

$$\boldsymbol{\rho}_{s_k} = \boldsymbol{\rho}_{s_k}^{\text{gt}} + \mathbf{n}_{\boldsymbol{\rho}_{s_k}}, \mathbf{n}_{\boldsymbol{\rho}_{s_k}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{n}_{\boldsymbol{\rho}_{s_k}}}) \quad (7)$$

\mathbf{p}_s^{gt} 和 $\boldsymbol{\rho}_{s_k}^{\text{gt}}$ 为 ${}^G\mathbf{p}_s$ 和 $\boldsymbol{\rho}_{s_k}$ 的真实值。因此，对 0 值残差 $\mathbf{r}(\mathbf{x}_k, \boldsymbol{\rho}_{s_k}^{\text{gt}}, {}^G\mathbf{p}_s^{\text{gt}})$ 一阶泰勒展开，得到：

$$\mathbf{0} = \mathbf{r}(\mathbf{x}_k, \boldsymbol{\rho}_{s_k}^{\text{gt}}, {}^G\mathbf{p}_s^{\text{gt}}) \approx \mathbf{r}(\check{\mathbf{x}}_k, \boldsymbol{\rho}_{s_k}, {}^G\mathbf{p}_s) + \mathbf{H}_s^r \delta \check{\mathbf{x}}_k + \boldsymbol{\alpha}_s \quad (8)$$

其中 $\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_s})$ 且：

$$\mathbf{H}_s^r = \frac{\partial \mathbf{r}_c(\check{\mathbf{x}}_k \boxplus \delta \check{\mathbf{x}}_k, \boldsymbol{\rho}_{s_k}, {}^G\mathbf{p}_s)}{\partial \delta \check{\mathbf{x}}_k} \Bigg|_{\delta \check{\mathbf{x}}_k = \mathbf{0}} \quad (9)$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_s} = \boldsymbol{\Sigma}_{\mathbf{n}_{\boldsymbol{\rho}_{s_k}}} + \mathbf{F}_{\mathbf{p}_s}^r \boldsymbol{\Sigma}_{\mathbf{p}_s} \mathbf{F}_{\mathbf{p}_s}^{r T}, \quad \mathbf{F}_{\mathbf{p}_s}^r = \frac{\partial \mathbf{r}(\check{\mathbf{x}}_k, \boldsymbol{\rho}_{s_k}, {}^G\mathbf{p}_s)}{\partial {}^G\mathbf{p}_s} \quad (10)$$

(2) Frame-to-Frame VIO ESIKF

式(8)构成 \mathbf{x}_k 的观测分布 ($\delta \check{\mathbf{x}}_k \triangleq \mathbf{x}_k \boxminus \check{\mathbf{x}}_k$)，可以根据 IMU 的先验分布获得 $\delta \check{\mathbf{x}}_k$ 的极大后验估计：

$$\begin{aligned} \min_{\delta \check{\mathbf{x}}_k} & \left(\|\check{\mathbf{x}}_k \boxminus \hat{\mathbf{x}}_k + \mathcal{H} \delta \check{\mathbf{x}}_k\|_{\boldsymbol{\Sigma}_{\delta \check{\mathbf{x}}_k}}^2 \right. \\ & \left. + \sum_{s=1}^m \|\mathbf{r}(\check{\mathbf{x}}_k, \boldsymbol{\rho}_{s_k}, {}^G\mathbf{p}_s) + \mathbf{H}_s^r \delta \check{\mathbf{x}}_k\|_{\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_s}}^2 \right) \end{aligned} \quad (11)$$

其中 $\|\mathbf{x}\|_{\boldsymbol{\Sigma}}^2 = \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}$ ， $\hat{\mathbf{x}}_k$ 为 IMU 传播来的状态估计， \mathcal{H} 为将 $\hat{\mathbf{x}}_k$ 投影到 $\check{\mathbf{x}}_k$ 时状态误差的雅可比矩阵。(11) 的前半部分推导可以在 R2LIVE 中找到。

注意：

$$\mathbf{H} = [\mathbf{H}_1^{rT}, \dots, \mathbf{H}_m^{rT}]^T, \quad \mathbf{R} = \text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_1}, \dots, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_m}) \quad (12)$$

$$\check{\mathbf{z}}_k = [\mathbf{r}(\check{\mathbf{x}}_k, \boldsymbol{\rho}_{1_k}, {}^G\mathbf{p}_1), \dots, \mathbf{r}(\check{\mathbf{x}}_k, \boldsymbol{\rho}_{m_k}, {}^G\mathbf{p}_m)]^T \quad (13)$$

$$\mathbf{P} = (\mathcal{H})^{-1} \boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_k} (\mathcal{H})^{-T} \quad (14)$$

根据 [4]，有卡尔曼增益：

$$\mathbf{K} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}^{-1})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \quad (15)$$

状态估计的更新为：

$$\check{\mathbf{x}}_k = \check{\mathbf{x}}_k \boxplus (-\mathbf{K} \check{\mathbf{z}}_k - (\mathbf{I} - \mathbf{K}\mathbf{H}) \mathcal{H}^{-1}(\check{\mathbf{x}}_k \boxminus \hat{\mathbf{x}}_k)) \quad (16)$$

上述过程直到收敛。请注意，这种迭代卡尔曼滤波器等同于高斯-牛顿优化。

5.2 Frame-to-Map Visual-Inertial Odometry

(1) Frame-to-Map Photometric Update

经过帧到帧的 VIO 更新，获得了一个良好的估计状态 $\check{\mathbf{x}}_k$ ，然后通过最小化跟踪点光度误差进行帧到地图点 VIO 更新。

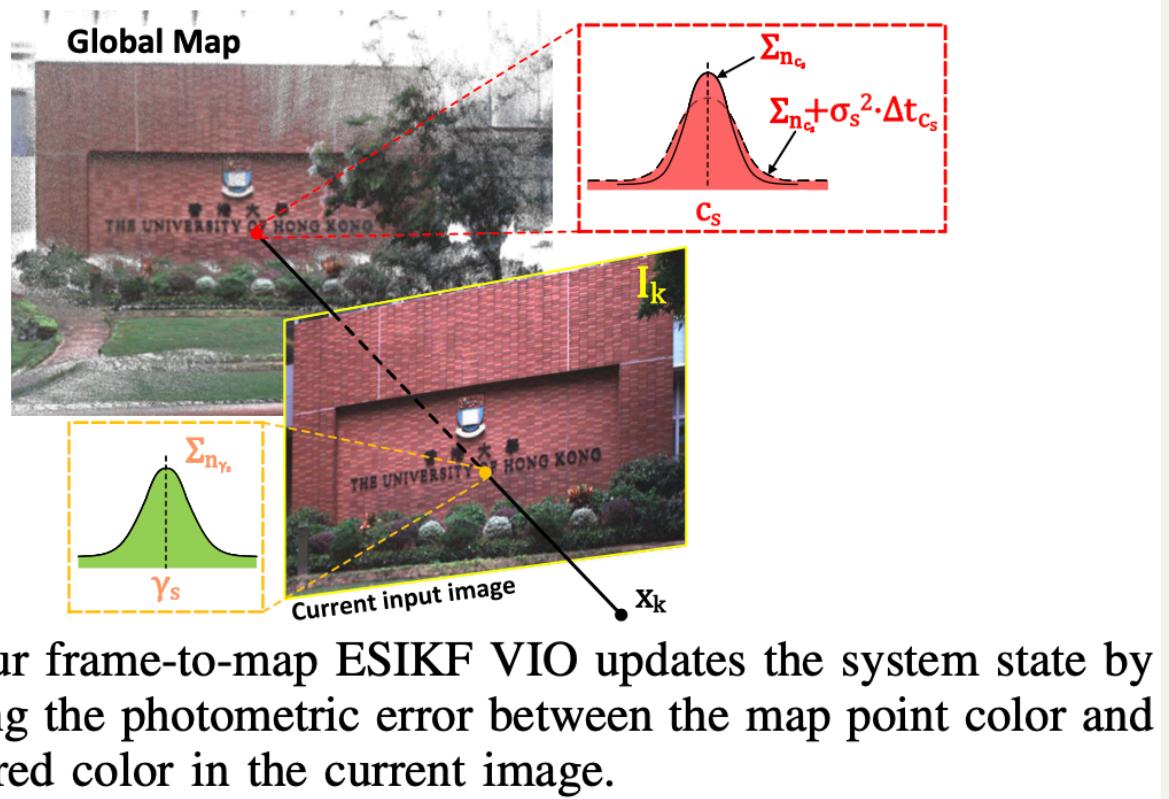


Fig. 4: Our frame-to-map ESIKF VIO updates the system state by minimizing the photometric error between the map point color and its measured color in the current image.

如图-4，以第 s 个跟踪点 $\mathbf{P}_s \in \mathcal{P}$ 为例，其帧到地图点误差 $\mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_s, \mathbf{c}_s)$ 为：

$$\mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_s, \mathbf{c}_s) = \mathbf{c}_s - \boldsymbol{\gamma}_s \quad (17)$$

\mathbf{c}_s 为存储在全局地图的点的颜色， $\boldsymbol{\gamma}_s$ 为在当前帧 \mathbf{I}_k 中观测到的点的颜色。为得到 $\boldsymbol{\gamma}_s$ 及其协方差 $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}_s}$ ，在 \mathbf{I}_k 中预测点的位置 $\tilde{\rho} = \pi({}^C\mathbf{p}_s, \check{\mathbf{x}}_k)$ ，然后线性插值根据邻居像素得到点的 RGB 颜色。

考虑噪声：

$$\boldsymbol{\gamma}_s = \boldsymbol{\gamma}_s^{\text{gt}} + \mathbf{n}_{\boldsymbol{\gamma}_s}, \mathbf{n}_{\boldsymbol{\gamma}_s} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{n}_{\boldsymbol{\gamma}_s}}) \quad (18)$$

$$\mathbf{c}_s = \mathbf{c}_s^{\text{gt}} + \mathbf{n}_{\mathbf{c}_s} + \boldsymbol{\eta}_{\mathbf{c}_s}, \mathbf{n}_{\mathbf{c}_s} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{n}_{\mathbf{c}_s}}), \boldsymbol{\eta}_{\mathbf{c}_s} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_s^2 \cdot \Delta t_{\mathbf{c}_s}) \quad (19)$$

$\Delta t_{\mathbf{c}_s}$ 为 \mathbf{P}_s 于当前帧相对于前一次访问的时间间隔。 $\mathbf{n}_{\mathbf{c}_s}$ 为估计误差， $\boldsymbol{\eta}_{\mathbf{c}_s}$ 为随机游走误差。

联合 (17-19)，得到 0 值残差 $\mathbf{o}(\mathbf{x}_k, {}^G\mathbf{p}_s^{\text{gt}}, \mathbf{c}_s^{\text{gt}})$ 的一阶泰勒展开：

$$\mathbf{0} = \mathbf{o}(\mathbf{x}_k, {}^G\mathbf{p}_s^{\text{gt}}, \mathbf{c}_s^{\text{gt}}) \approx \mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_s, \mathbf{c}_s) + \mathbf{H}_s^o \delta \check{\mathbf{x}}_k + \boldsymbol{\beta}_s \quad (20)$$

其中 $\boldsymbol{\beta}_s \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_s})$ 且：

$$\mathbf{H}_s^o = \left. \frac{\partial \mathbf{o}(\check{\mathbf{x}}_k \boxplus \delta \check{\mathbf{x}}_k, {}^G\mathbf{p}_s, \mathbf{c}_s)}{\partial \delta \check{\mathbf{x}}_k} \right|_{\delta \check{\mathbf{x}}_k = \mathbf{0}} \quad (21)$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_s} = \boldsymbol{\Sigma}_{\mathbf{n}_{\mathbf{c}_s}} + \boldsymbol{\sigma}_s^2 \cdot \Delta t_{\mathbf{c}_s} + \boldsymbol{\Sigma}_{\mathbf{n}_{\boldsymbol{\gamma}_s}} + \mathbf{F}_{\mathbf{p}_s}^o \boldsymbol{\Sigma}_{\mathbf{p}_s} \mathbf{F}_{\mathbf{p}_s}^{oT} \quad (22)$$

$$\mathbf{F}_{\mathbf{p}_s}^o = \frac{\partial \mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_s, \mathbf{c}_s)}{\partial {}^G\mathbf{p}_s} \quad (23)$$

(2) Frame-to-Map VIO ESIKF Update

(20) 构成了 $\delta \check{\mathbf{x}}_k$ 的另一个观测分布，其结合 IMU 传播的先验以获得 $\delta \check{\mathbf{x}}_k$ 的最大后验估计：

$$\begin{aligned} & \min_{\delta \check{\mathbf{x}}_k} \left(\left\| \check{\mathbf{x}}_k \boxminus \hat{\mathbf{x}}_k + \mathcal{H} \delta \check{\mathbf{x}}_k \right\|_{\boldsymbol{\Sigma}_{\delta \check{\mathbf{x}}_k}}^2 \right. \\ & \quad \left. + \sum_{s=1}^m \left\| \mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_s, \mathbf{c}_s) + \mathbf{H}_s^o \delta \check{\mathbf{x}}_k \right\|_{\boldsymbol{\Sigma}_{\boldsymbol{\beta}_s}}^2 \right) \end{aligned} \quad (24)$$

$\mathbf{H}, \mathbf{R}, \check{\mathbf{z}}_k, \mathbf{P}$ 类似于 (12-14)：

$$\mathbf{H} = [\mathbf{H}_1^{oT}, \dots, \mathbf{H}_m^{oT}]^T, \quad \mathbf{R} = \text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}_1}, \dots, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_m}) \quad (25)$$

$$\check{\mathbf{z}}_k = [\mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_1, \mathbf{c}_1), \dots, \mathbf{o}(\check{\mathbf{x}}_k, {}^G\mathbf{p}_m, \mathbf{c}_m)]^T \quad (26)$$

$$\mathbf{P} = (\mathcal{H})^{-1} \boldsymbol{\Sigma}_{\delta \check{\mathbf{x}}_k} (\mathcal{H})^{-T} \quad (27)$$

然后执行如同 (15) 和 (16) 的状态更新。

执行帧到地图的 VIO ESIKF 更新直到收敛。然后，融合状态估计用于：

- 渲染地图的纹理；
- 更新当前跟踪点集 \mathcal{P} ，以便下一帧使用；
- 作为 LIO 或 VIO 更新的下一帧中 IMU 传播的起点。

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k, \quad \Sigma_{\delta\check{\mathbf{x}}_k} = (\mathbf{I} - \mathbf{K}\mathbf{H})\Sigma_{\delta\check{\mathbf{x}}_k} \quad (28)$$

5.3 Render the Texture of Global Map

在更新从帧到地图的 VIO 之后，得到当前帧的精确位置，然后执行渲染函数来更新映射点的颜色。

首先，使用所有在 *activated* 体素的点，记为 $\zeta = \{\mathbf{P}_1, \dots, \mathbf{P}_n\}$ 。以第 s 个点 $\mathbf{P}_s = [{}^G\mathbf{p}_s^T, \mathbf{c}_s^T]^T$ 的颜色更新为例。如果 \mathbf{P}_s 落在当前帧 \mathbf{I}_k ，则可以获得其观测颜色 γ_s （通过线性插值获得）和协方差 $\Sigma_{\mathbf{n}_{\gamma_s}}$ 。新观测的点的颜色通过贝叶斯更新与存在地图上的已有颜色 \mathbf{c}_s 融合：

$$\Sigma_{\mathbf{n}_{\tilde{\mathbf{c}}_s}} = \left((\Sigma_{\mathbf{n}_{\mathbf{c}_s}} + \sigma_s^2 \cdot \Delta t_{\mathbf{c}_s})^{-1} + \Sigma_{\mathbf{n}_{\gamma_s}}^{-1} \right)^{-1} \quad (29)$$

$$\tilde{\mathbf{c}}_s = \left((\Sigma_{\mathbf{n}_{\mathbf{c}_s}} + \sigma_s^2 \cdot \Delta t_{\mathbf{c}_s})^{-1} \mathbf{c}_s + \Sigma_{\mathbf{n}_{\gamma_s}}^{-1} \gamma_s \right)^{-1} \Sigma_{\mathbf{n}_{\tilde{\mathbf{c}}_s}} \quad (30)$$

$$\mathbf{c}_s = \tilde{\mathbf{c}}_s, \Sigma_{\mathbf{n}_{\mathbf{c}_s}} = \Sigma_{\mathbf{n}_{\tilde{\mathbf{c}}_s}} \quad (31)$$

5.4 Update of the Tracking Points of VIO Subsystem

纹理渲染完成后，对跟踪到的点集 \mathcal{P} 进行更新：

- 首先，对 \mathcal{P} 中投影误差 (4) 或光度误差 (17) 较大的点进行删除，同时删除不属于 \mathbf{I}_k 的点。
- 其次，将 ζ 中的每个点投影到当前图像 \mathbf{I}_k 中，如果附近（例如半径为 50 像素）没有其他跟踪点，则将其添加到 \mathcal{P} 中。

6 Experiments and Results

6.1 Setup of the Experiments



Fig. 5: Our handheld device for data collection. (a) shows our minimum system. (b) shows an additional D-GPS RTK system and an ArUco marker board used to evaluate the system accuracy.

- DJI manifold-2c 计算平台
- FLIR Blackly BFS-u3-13y3c 相机
- LiVOX AVIA LiDAR
- 为了定量评估算法的精度,在设备上安装了差分 GPS RTK 系统

对于(1)中 \mathbf{x} 的初始化, 除外参和内参 ϕ , ${}^I\mathbf{R}_C^T$, ${}^I\mathbf{p}_C^T$ 外, 其余设为 0。

6.2 Robustness Evaluation in Simultaneously LiDAR Degenerated and Visual Texture-Less Environments

同时在 LiDAR 退化和视觉无纹理环境下进行鲁棒性评估, 如图-6。

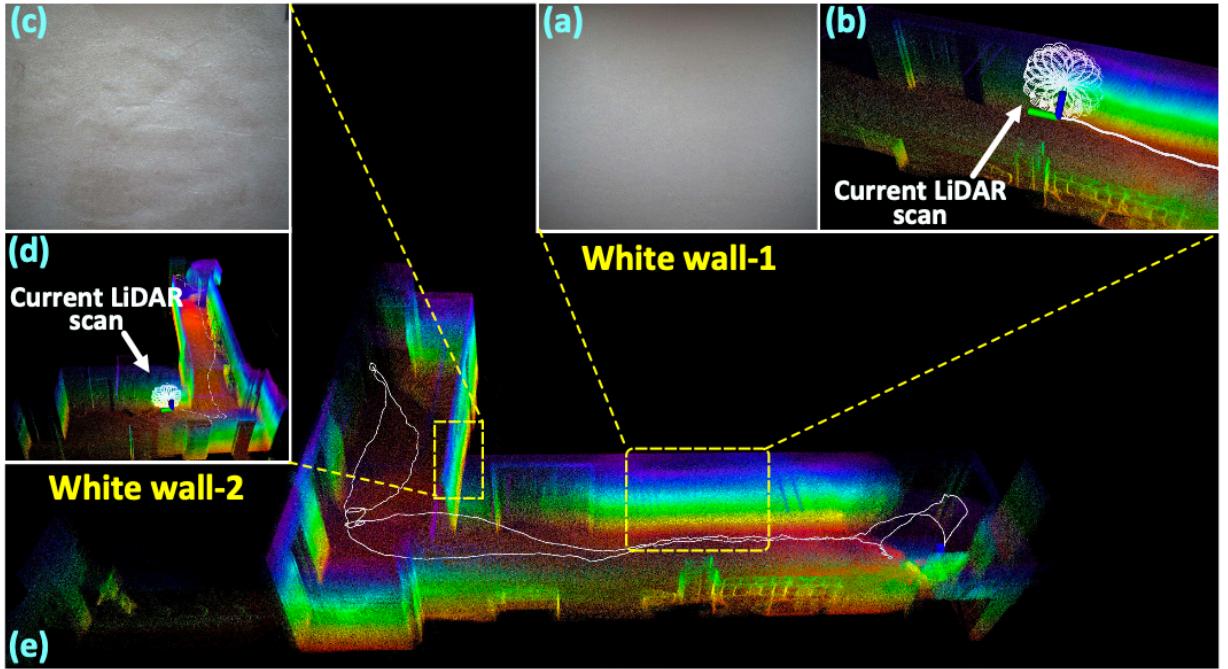


Fig. 6: We test our algorithm in simultaneously LiDAR degenerated and visual texture-less scenarios.

图-8 显示了估计姿态，通过“墙-1”和“墙-2”的阶段分别被紫色和黄色阴影。

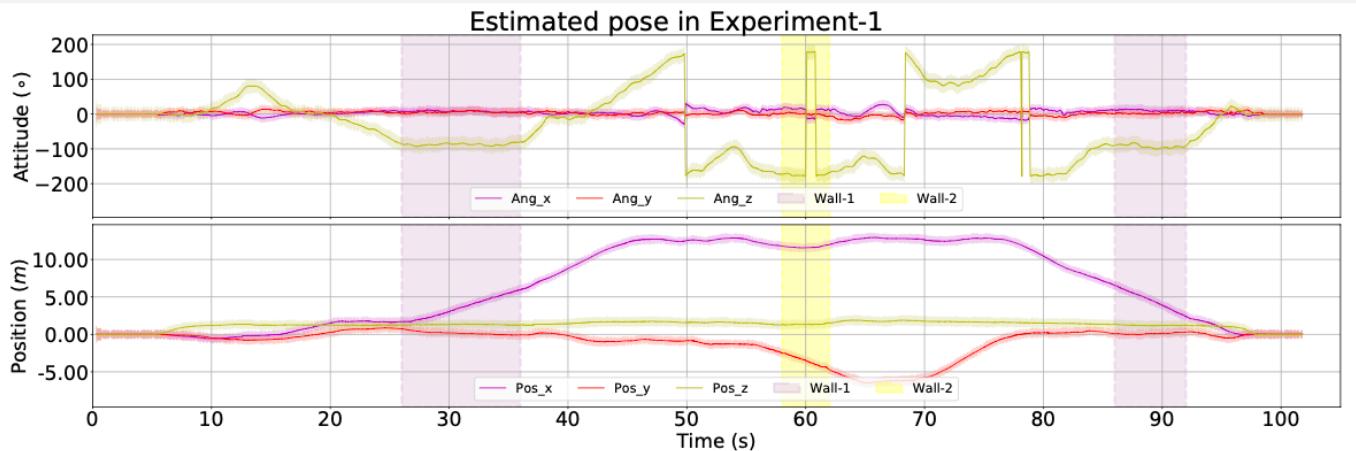


Fig. 8: The estimated poses and their 3σ bound with 5 times amplification for better visualization (the light-colored area around the trajectory) of Experiment-1. The shaded areas in purple and yellow are the phases of the sensors facing against the white “wall-1” and “wall-2”, respectively.

6.3 High Precision Mapping Large-Scale Indoor & Outdoor Urban Environment

在香港科技大学校园内收集了 4 条不同旅行轨迹 (Traj 1~4) 数据。这些轨迹的鸟瞰图 (在 XY 平面) 如图-7(f)。

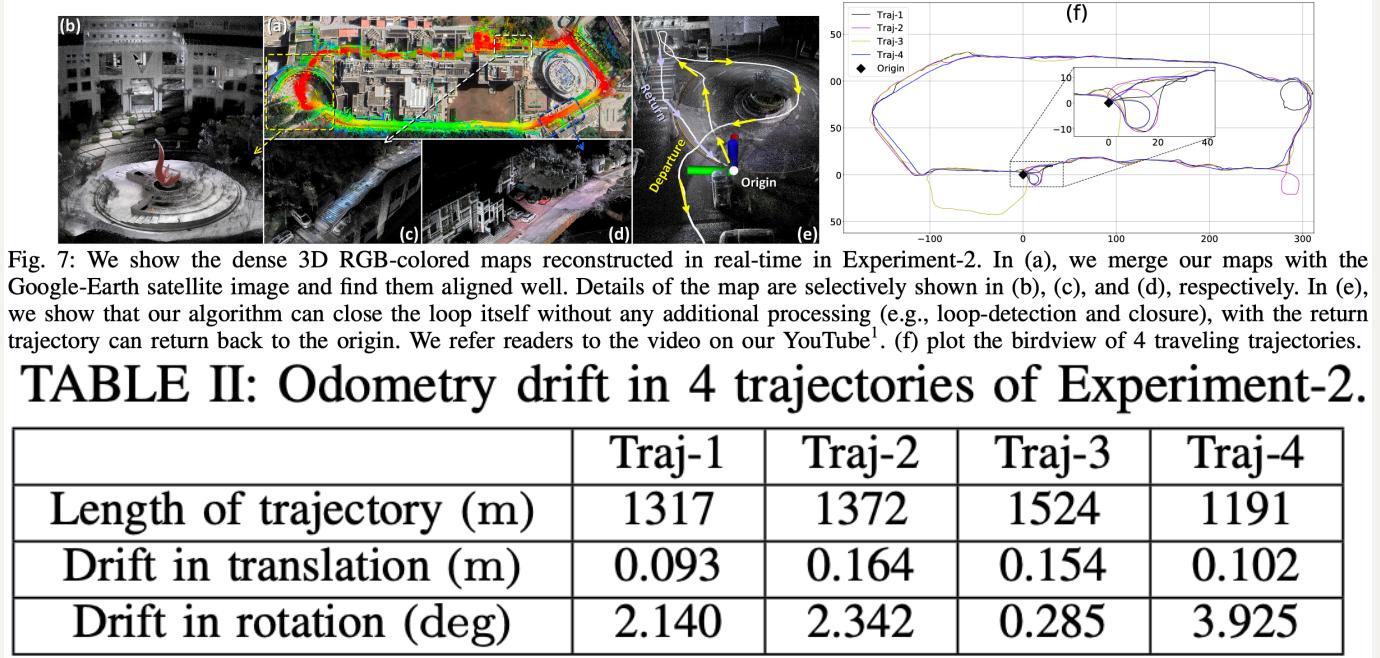


Fig. 7: We show the dense 3D RGB-colored maps reconstructed in real-time in Experiment-2. In (a), we merge our maps with the Google-Earth satellite image and find them aligned well. Details of the map are selectively shown in (b), (c), and (d), respectively. In (e), we show that our algorithm can close the loop itself without any additional processing (e.g., loop-detection and closure), with the return trajectory can return back to the origin. We refer readers to the video on our YouTube¹. (f) plot the birdview of 4 traveling trajectories.

TABLE II: Odometry drift in 4 trajectories of Experiment-2.

	Traj-1	Traj-2	Traj-3	Traj-4
Length of trajectory (m)	1317	1372	1524	1191
Drift in translation (m)	0.093	0.164	0.154	0.102
Drift in rotation (deg)	2.140	2.342	0.285	3.925

6.4 Quantitative Evaluation of Precision Using D-GPS RTK

在海港用一个实时差分 GPS 运动学系统收集了两个序列的数据，提供了真实轨迹。在这两个序列中，传感器通常面对许多行人，偶尔也会面对开阔的海域，在那里激光雷达的测量数据很少。

TABLE III: The relative pose error in Experiment-3. In configuration “R3LIVE-HiRES”, we set the input image resolution as 1280×1024 and the minimum distance (i.e., the resolution of point cloud map) between two points in maps as 0.01 m. In configuration “R3LIVE-RT”, we set the input image resolution as 320×256 and the minimum distance between two points in maps as 0.10 m

Relative pose error in Experiments-3 (a)						
Sub-sequence RRE/RTE	50 m deg / %	100 m deg / %	150 m deg / %	200 m deg / %	250 m deg / %	300 m deg / %
R3LIVE-HiRES	0.99 / 2.25	0.53 / 1.02	0.46 / 0.60	0.29 / 0.31	0.24 / 0.31	0.21 / 0.17
R3LIVE-RT	1.48 / 2.21	0.54 / 1.06	0.49 / 0.60	0.31 / 0.41	0.25 / 0.31	0.22 / 0.23
LVI-SAM	2.11 / 13.73	1.04 / 6.69	0.83 / 5.07	0.60 / 3.86	0.47 / 2.98	0.43 / 2.40
R2LIVE	1.21 / 2.47	0.61 / 1.02	0.59 / 0.60	0.34 / 0.34	0.37 / 0.33	0.34 / 0.21
FAST-LIO2	1.36 / 2.35	0.66 / 0.82	0.47 / 0.60	0.37 / 0.36	0.37 / 0.31	0.21 / 0.20
VINS-Mono	3.03 / 10.41	1.70 / 7.63	1.15 / 6.36	0.89 / 4.34	0.73 / 3.08	0.59 / 2.31

Relative pose error in Experiments-3 (b)						
Sub-sequence RRE/RTE	50 m deg / %	100 m deg / %	150 m deg / %	200 m deg / %	250 m deg / %	300 m deg / %
R3LIVE-HiRES	1.06 / 1.98	0.58 / 1.34	0.43 / 0.83	0.32 / 0.59	0.27 / 0.27	0.25 / 0.16
R3LIVE-RT	0.98 / 2.08	0.55 / 1.61	0.47 / 0.99	0.39 / 0.65	0.33 / 0.33	0.25 / 0.14
LVI-SAM	2.04 / 3.33	1.05 / 2.37	0.81 / 1.53	0.57 / 1.38	0.49 / 1.13	0.44 / 1.01
R2LIVE	1.13 / 2.06	0.65 / 1.45	0.49 / 0.88	0.31 / 0.57	0.26 / 0.30	0.29 / 0.16
FAST-LIO2	1.31 / 2.22	0.69 / 1.34	0.49 / 0.89	0.31 / 0.59	0.26 / 0.31	0.25 / 0.29
VINS-Mono	2.44 / 9.35	1.31 / 7.28	0.99 / 4.63	0.66 / 3.68	0.56 / 3.24	0.48 / 2.73

6.5 Runtime

对比 PC (i7-9700K, 32 GB RAM) 和板载 UAV (“OB”, i7-8550u, 8GB RAM)。Pc res 表示点云地图分辨率。

TABLE IV: The average time consumption on two different platforms with different configurations.

VIO per-frame cost time										LIO per-frame cost time	
Image size	320×256			640×512			1280×1024				
Pc res (m)	0.10	0.05	0.01	0.10	0.05	0.01	0.10	0.05	0.01		
On PC (ms)	7.08	8.43	16.78	10.38	11.74	20.67	12.13	13.19	23.17	18.40	
On OB (ms)	15.30	21.69	43.79	23.71	34.70	47.99	30.38	42.43	61.34	39.56	

参考

- [1] J. Lin, C. Zheng, W. Xu, and F. Zhang, “R2live: A robust, real- time, lidar- inertial-visual tightly-coupled state estimator and mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.
- [2] W. Xu, Y. Cai, D. He, J. Lin and F. Zhang, "FAST-LIO2: Fast Direct LiDAR- Inertial Odometry," in *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053- 2073, Aug. 2022, doi: 10.1109/TRO.2022.3141876.
- [3] T. Qin and S. Shen, “Online temporal calibration for monocular visual-inertial systems,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3662– 3669.
- [4] W. Xu and F. Zhang, “Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter,” *IEEE Robotics and Automation Letters*, 2021, in press.