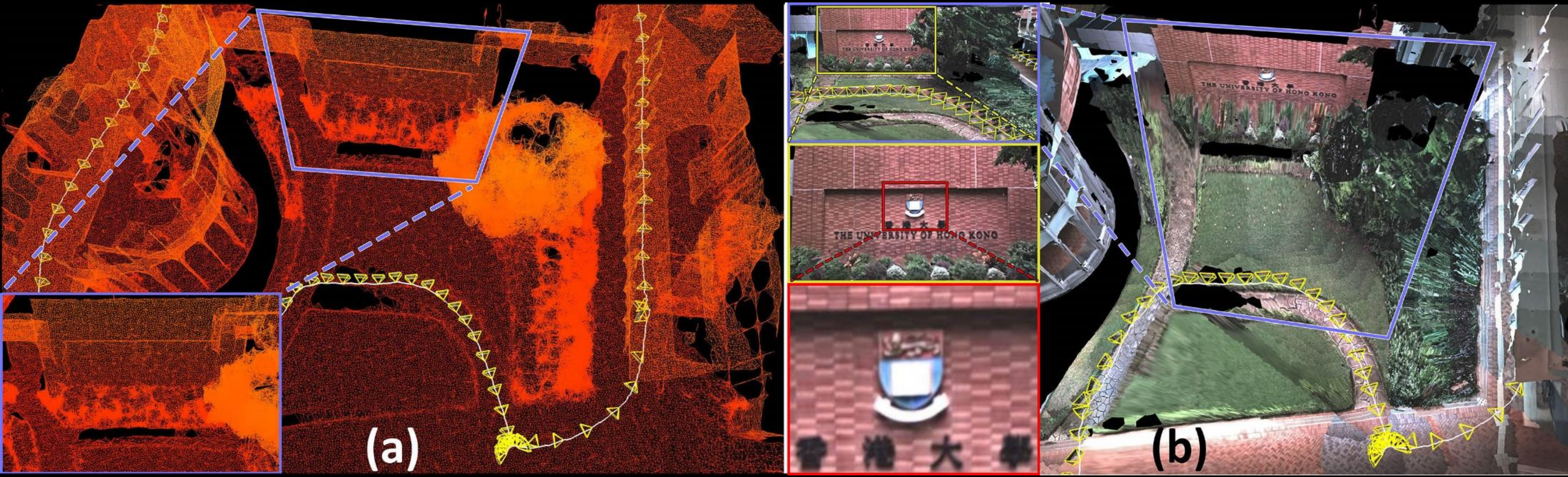
基于多传感器(激光雷达-惯导-视觉)融合的定位和建图系统

主讲人： 林家荣
E-Mail： ziv.lin@connect.hku.hk



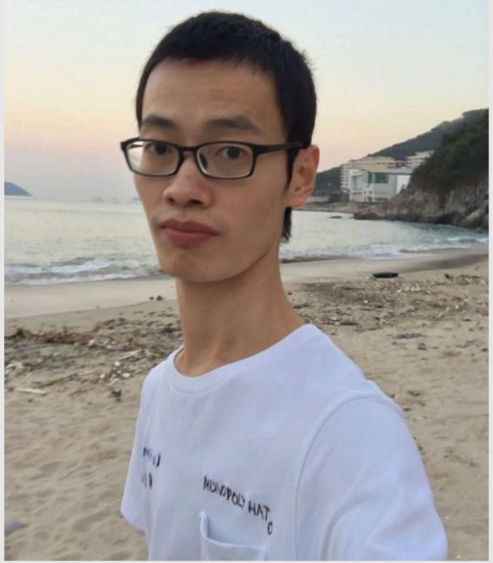
MARS LAB

香 港 大 學
THE UNIVERSITY OF HONG KONG

自我介绍—研究经历

我是香港大学机械工程学院MaRS LAB的在读博士生，师从导师张富，我的主要研究方向是**激光雷达SLAM**以及**多传感器（激光雷达-惯导-视觉）融合**

Jiarong Lin 林家荣



机器人方向-在读博士生

激光 SLAM; 多传感器融合; 三维重建

jiarong.lin@hku.hk

https://github.com/ziv-lin

•研究经历

2019 - 现在 **在读博士生** **香港大学 (HKU)**

HKU MaRS LAB

- **激光 SLAM**: 从事基于激光雷达传感器的同时定位和建图的研究，以第一作者的身份发表了业界首个使用固态激光雷达 SLAM 框架 loam-livox。
- **多传感器融合**: 从事多传感器 (雷达-惯导-视觉) 融合的研究。合作发表了雷达-惯导紧耦合框架 FAST-LIO2，以第一作者的身份发表业界首个开源的雷达-惯导-视觉紧耦合框架 R^2LIVE, 以及首个实时辐射场地图重建框架 R^3LIVE。

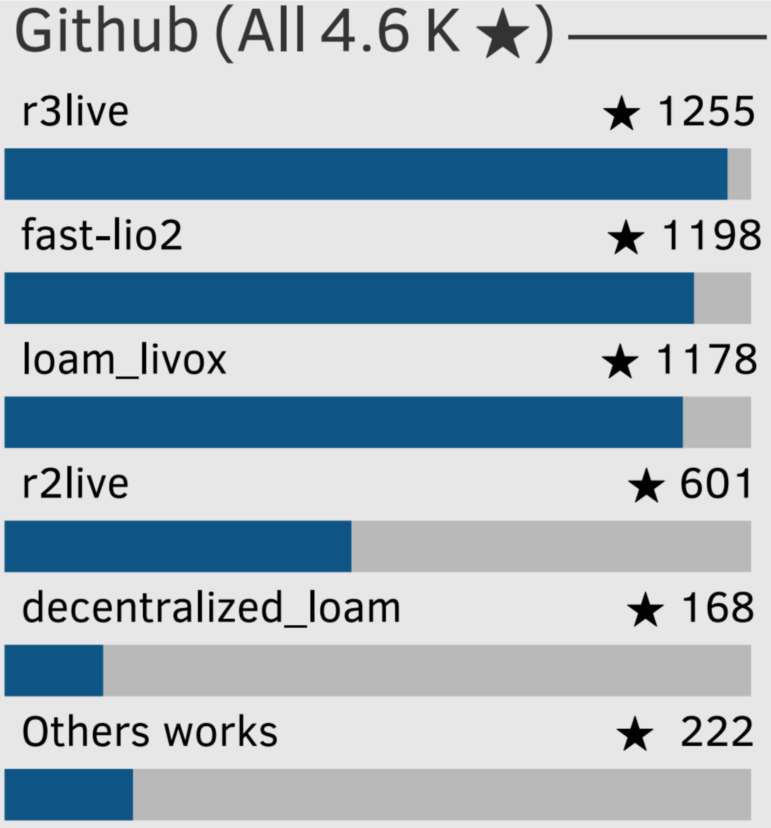
2018 - 2019 **在读博士生** **香港科技大学 (HKUST)**

Robotics Institude

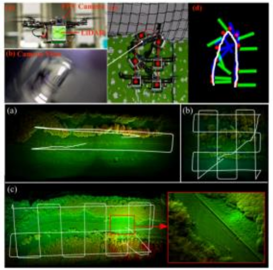
- **无人机运动规划和控制**: 从事全无人机自主导航系统的研究，包括无人机的最优运动规划以及鲁棒控制
- **基于深度强化学习无人机控制**: 从事基于深度强化学习的无人机控制的研究。以第一作者的身份发表了首个使用深度强化学习方法实现的无人机穿越窄斜长缝隙的工作。

自我介绍—主要贡献

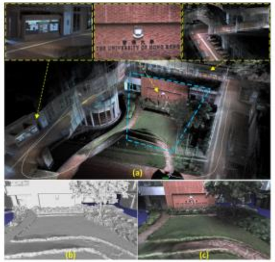
我先后在机器人顶会顶刊ICRA，IROS，RA-L上，以**第一作者**的身份发表了5篇论文，代表性工作有**loam-livox(★1178)，fast-lio2(★1198), R²LIVE(★ 601), R³LIVE(★ 1255)**。我的所有的论文著作都有对应的开源项目，在开源社区累计获得了超过**4.6K的Star**。



R³LIVE++: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package (Under review)
Author: Jiarong Lin and Fu Zhang
Github: https://github.com/hku-mars/r3live (1255 ★)
Introduction: In this work, we proposed a LiDAR-inertial-visual fusion framework termed R3LIVE++ to achieve robust and accurate state estimation while simultaneously reconstructing the radiance map on the fly. R3LIVE++ is developed based on R3LIVE and further improves the accuracy in localization and mapping by accounting for the camera photometric calibration (e.g., non-linear response function and lens vignetting) and the online estimation of camera exposure time. Our quantitative and qualitative results show that our proposed system has significant improvements over others in both accuracy and robustness.



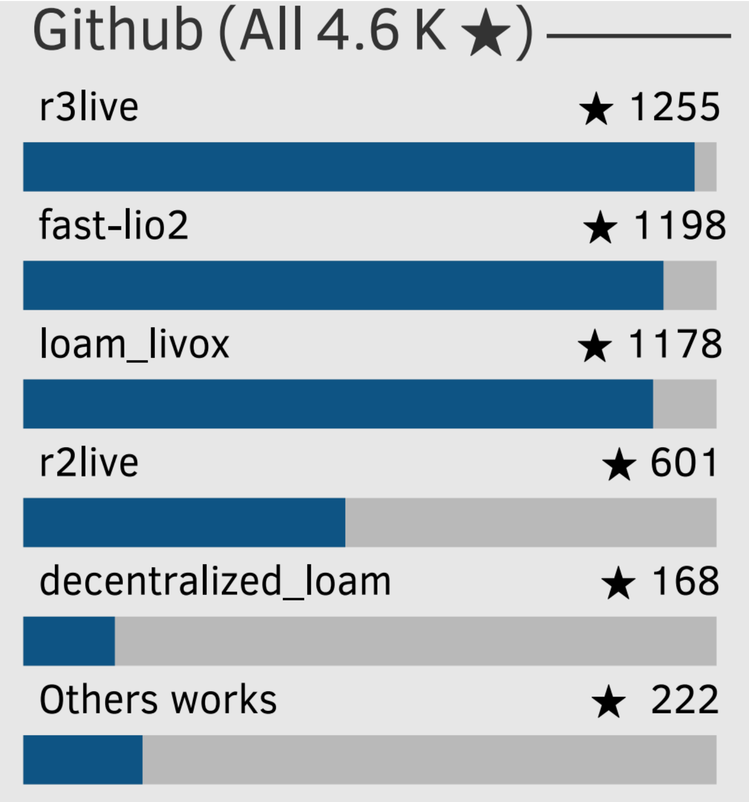
FAST-LIO2: Fast Direct LiDAR-inertial Odometry (Transaction on robotics (TRO 2022))
Author: Wei Xu, Yixi Cai, Dongjiao He, **Jiarong Lin** and Fu Zhang
Github: https://github.com/hku-mars/FAST_LIO (1198 ★)
Introduction: In this paper. we propose a fast, robust, and versatile LiDAR-inertial odometry framework termed FAST-LIO2, which is built upon a highly efficient tightly-coupled iterated Kalman filter and is fast with two key novelties: the direct use of raw point and an incremental k-d tree data structure. FAST-LIO2 is computationally-efficient (e.g., up to 100 Hz odometry and mapping in large outdoor environments), robust (e.g., reliable pose estimation in cluttered indoor environments with rotation up to 1000 deg/s), versatile (i.e., applicable to both multi-line spinning and solid-state LiDARs, UAV and handheld platforms, and Intel and ARM-based processors), while still achieving higher accuracy than existing methods.

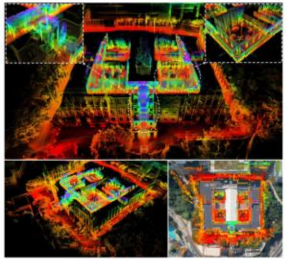


R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package (ICRA 2022)
Author: Jiarong Lin and Fu Zhang
Github: https://github.com/hku-mars/r3live (1255 ★)
Introduction: In this letter, In this letter, we propose a novel LiDAR-Inertial-Visual sensor fusion framework termed R3LIVE, which is developed based on our previous work R2LIVE. R3LIVE is a versatile and well-engineered system toward various possible applications, which can not only serve as a SLAM system for real-time robotic applications but can also reconstruct the dense, precise, RGB-colored 3D maps for applications like surveying and mapping. Moreover, we also develop a series of offline utilities for reconstructing and texturing meshes for various of 3D applications.

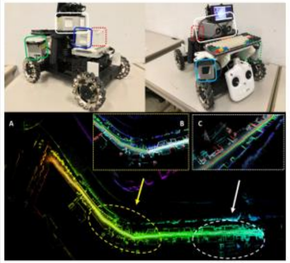
自我介绍—主要贡献

我先后在机器人顶会顶刊ICRA，IROS，RA-L上，以**第一作者**的身份发表了5篇论文，代表性工作有**loam-livox(★1178)，fast-lio2(★1198), R^2LIVE(★ 601), R^3LIVE(★ 1255)**。我的所有的论文著作都有对应的开源项目，在开源社区累计获得了超过**4.6K的Star**。

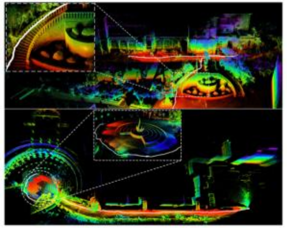




R^2LIVE: A Robust, Real-time, LiDAR-Inertial-Visual tightly-coupled state Estimator and mapping (Robotics and Automation Letters (RA-L 2021))
Author: Jiarong Lin, Chunran Zheng, Wei Xu and Fu Zhang
Github: https://github.com/hku-mars/r2live (601 ★)
Introduction: In this letter, we propose a robust, real-time tightly-coupled multi-sensor fusion framework, which fuses measurement from LiDAR, inertial sensor, and visual camera to achieve robust and accurate state estimation. Our framework estimates the state within the framework of error-state iterated Kalman-filter, and further improves the overall precision with our factor graph optimization. Taking advantage of measurement from all individual sensors, R2LIVE is robust enough to various visual failure, LiDAR-degenerated scenarios, and is able to run in real-time on an on-board computation platform.



A decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs (IROS 2020)
Author: Jiarong Lin, Xiyuan Liu, Fu Zhang
Github: https://github.com/hku-mars/decentralized_loam (168 ★)
Introduction: In this paper, we propose a framework for multiple LiDARs fusion, within this framework, we can not only address the problem of localization and mapping but can also online calibrate the extrinsic of 6-DoF. Our framework is based on an extended Kalman filter but is specially formulated for decentralized implementation. In our experiments, we achieved the accuracy of localization up to 0.2% on the two datasets we collected.



Loam_livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV (ICRA 2020)
Author: Jiarong Lin and Fu Zhang
Github: https://github.com/hku-mars/loam_livox (1178 ★)
Introduction: In this paper, we present a robust, real-time LOAM algorithm for LiDARs with small FoV and irregular samplings. By taking effort on both frontend and back-end, we address several fundamental challenges arising from such LiDARs (i.e. small FoV, irregular LiDAR scanning pattern, motion compensation, and etc), and achieve better performance in both precision and efficiency compared to existing baselines.

自我介绍—主要贡献

我先后在机器人顶会顶刊ICRA，IROS，RA-L上，以**第一作者**的身份发表了5篇论文，代表性工作有**loam-livox(★1178)，fast-lio2(★1198), R²LIVE(★ 601), R³LIVE(★ 1255)**。我的所有的论文著作都有对应的开源项目，在开源社区累计获得了超过**4.6K的Star**。

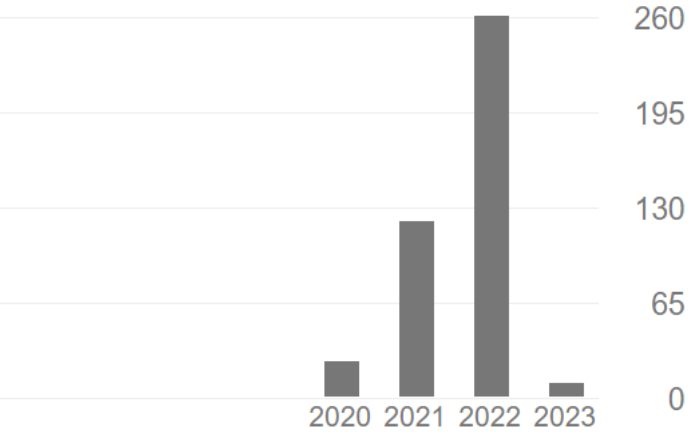
Jiarong lin
已关注

The University of Hong Kong
在 hku.hk 的电子邮件经过验证 - 首页
Robotics LiDAR SLAM Sensor fusion 3D reconstruction

标题	引用次数	年份
Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV J Lin, F Zhang 2020 IEEE International Conference on Robotics and Automation (ICRA), 3126-3131	150	2020
Fast-lio2: Fast direct lidar-inertial odometry W Xu, Y Cai, D He, J Lin, F Zhang IEEE Transactions on Robotics	86	2022
R ² LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping J Lin, C Zheng, W Xu, F Zhang IEEE Robotics and Automation Letters 6 (4), 7469-7476	48	2021
A fast, complete, point cloud based loop closure for LiDAR odometry and mapping J Lin, F Zhang arXiv preprint arXiv:1909.11811	37	2019
R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package J Lin, F Zhang arXiv preprint arXiv:2109.07982	35	2021
Flying through a narrow gap using neural network: an end-to-end planning and control approach J Lin, L Wang, F Gao, S Shen, F Zhang 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems ...	25	2019
A decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs J Lin, X Liu, F Zhang 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems ...	23	2020
Full attitude control of an efficient quadrotor tail-sitter VTOL UAV with flexible modes W Xu, H Gu, Y Qing, J Lin, F Zhang 2019 International Conference on Unmanned Aircraft Systems (ICUAS), 542-550	9	2019
R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator J Lin, F Zhang arXiv preprint arXiv:2209.03666	4	2022

引用次数

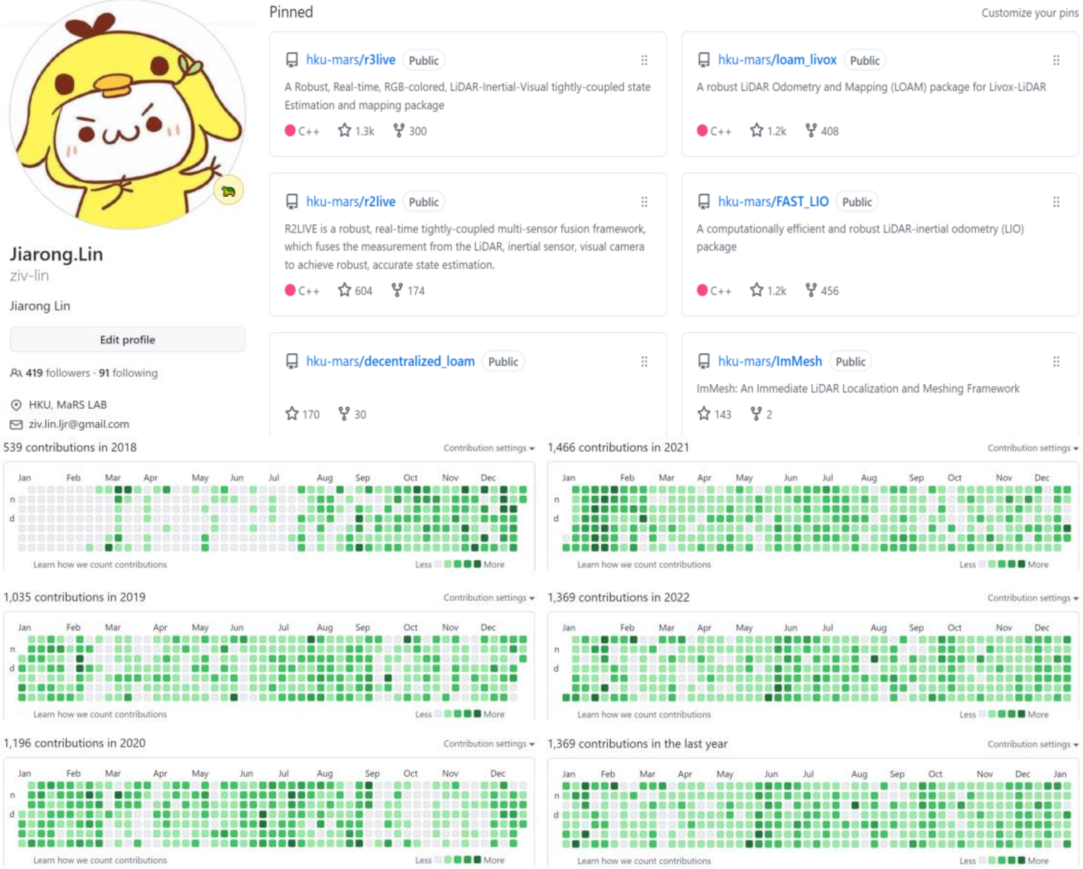
	总计	2018 年至今
引用	418	418
h 指数	8	8
i10 指数	7	7



自我介绍—主要贡献

同时，我也是一个积极的开源社区贡献者，平时也会开源一些小工具，硬件设计平台之类的~
详情请访问我的GitHub主页：https://github.com/ziv-lin

扫描此↑↑↑↑二维码
访问我的GitHub主页



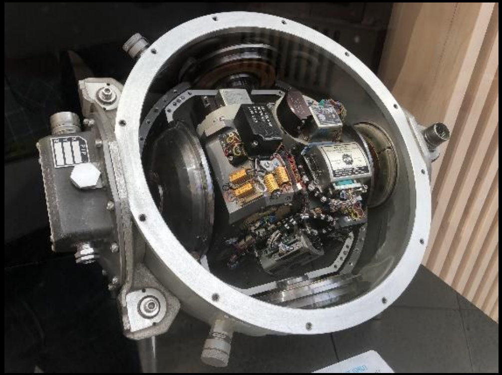
内容简介—内容大纲

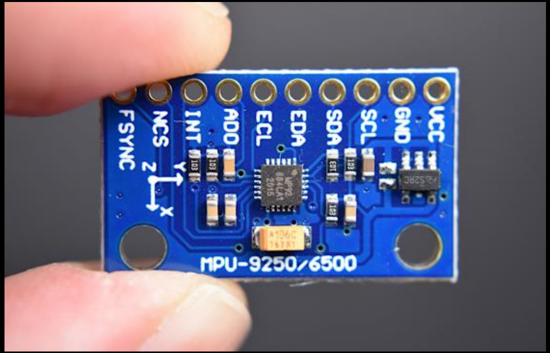
科普向地分享我读博士期间的一些研究成果，**系统性**地介绍我的一些代表性工作。

1. 传感器介绍
2. 激光雷达（-惯导）SLAM
   - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
   - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
   - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R^2LIVE**）
   - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R^3LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
   - ImMesh的介绍和实验演示
   - 基于ImMesh开发的应用
     - 激光雷达点云增强
     - 快速无损的场景纹理重建

1. 传感器介绍
2. 激光雷达（-惯导）SLAM
   - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
   - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
   - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R^2LIVE**）
   - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R^3LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
   - ImMesh的介绍和实验演示
   - 基于ImMesh开发的应用
     - 激光雷达点云增强
     - 快速无损的场景纹理重建

1. 传感器介绍—惯性测量单元(IMU)





由于IMU(加速度计，陀螺仪)由于测量频率高，对运动灵敏，成本低，体积小，抗干扰能力强，因此IMU被广泛应用到各种多传感器融合工作中。但是IMU有以下的局限性：

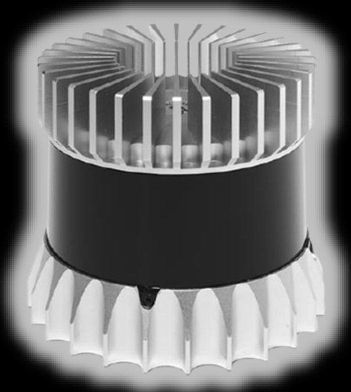
- 测量值存在随机游走的Bias
- 存在累积误差，长时间的积分结果不可靠

1. 传感器介绍—视觉相机

由于视觉相机轻、小、价格便宜，基于视觉相机的SLAM广泛应用于低成本，轻载重的平台（如无人机, AR/VR）上。但是基于相机的SLAM方法受限于：

- 单目相机无法直接获得可靠的3D几何信息
- 基于多视图(双目，从运动恢复结构(SFM))的方法可以获得3D几何信息, 但是需要消耗大量的计算资源, 而且在视觉纹理缺失,大量重复的场景下效果较差
- 基于相机的SLAM方法对环境关照的要求较为苛刻

1. 传感器介绍—激光雷达

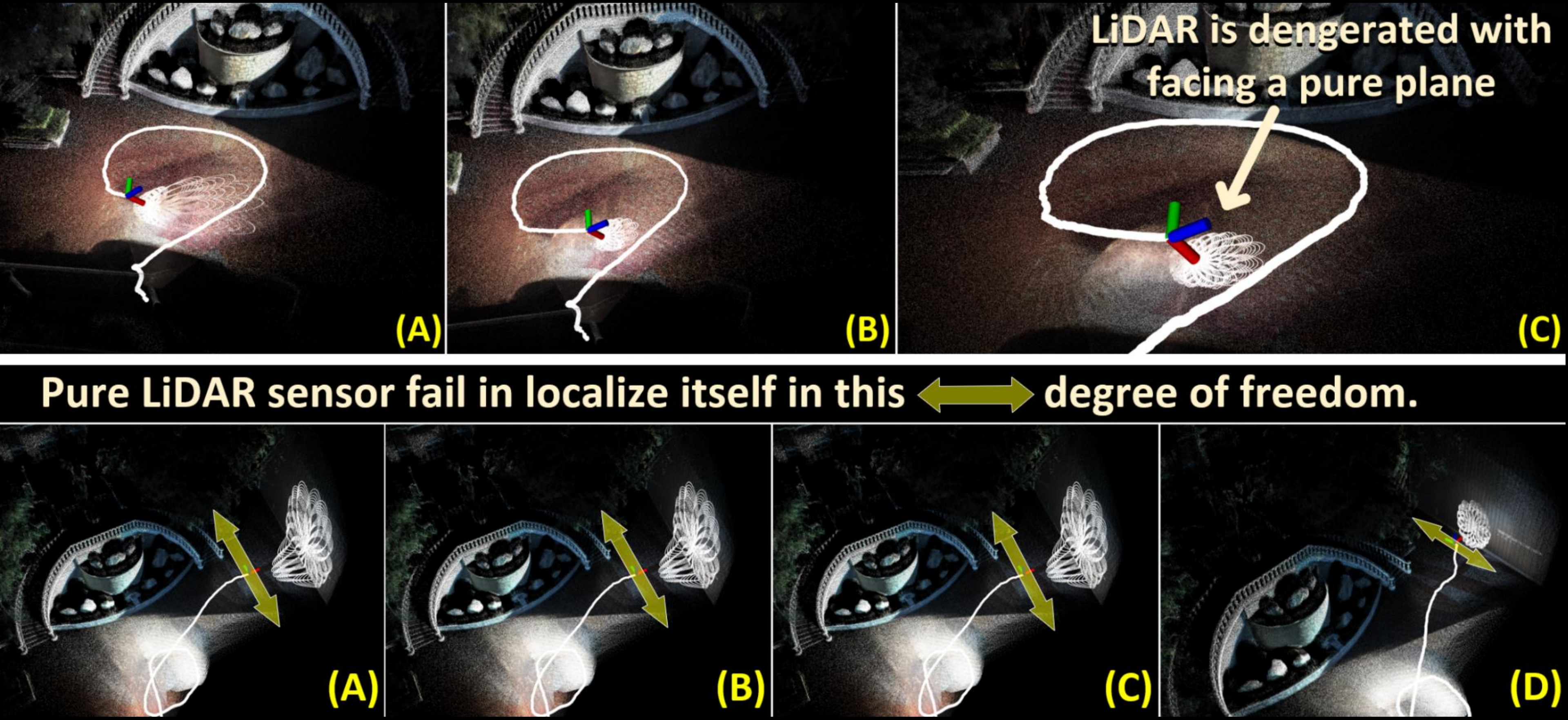


由于激光雷达(LiDAR)传感器可以提供很高精度稠密3D点云，因此激光SLAM方法能提供6自由度的姿态估计，以及能同时重建周围环境高精度的3D地图，但是也受限于：

- 激光雷达传感器贵,大,重
- 分辨率较低（和视觉比），刷新率低（~10HZ）
- 帧内点云有运动畸变
- 在环境几何特征不足的情况下会退化

1. 传感器介绍—激光雷达

纯激光雷达的SLAM在几何特征不足的情况下会发生退化现象:



2. 传感器介绍—视觉相机&激光雷达

相机和 LiDAR 是两种**互补**的传感器：相机拍摄2D图形的形式，把现实世界的视觉问题投影到 2D平面上，而 LiDAR则保留着环境的 3D 几何形状。

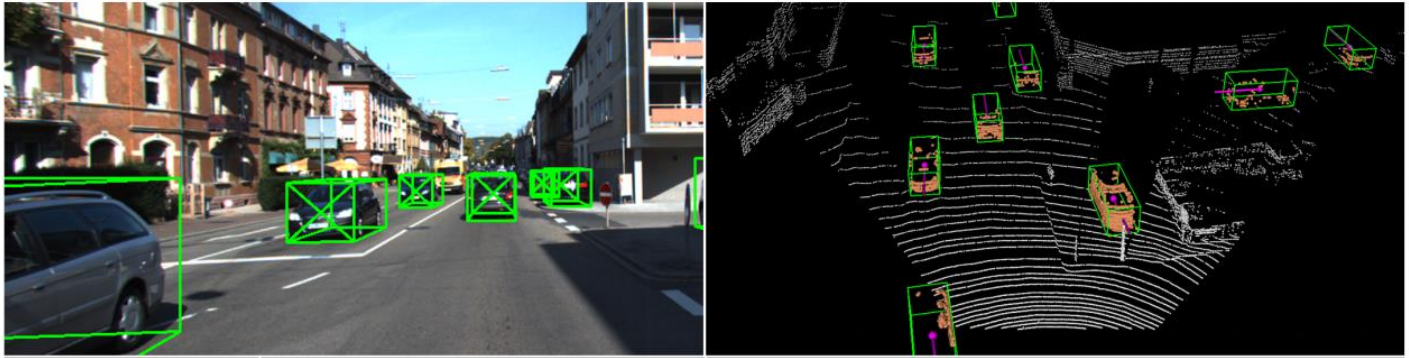


	Image	Point Cloud
Permutation	Ordered	Orderless
Data Structure	Regular	Irregular
Data Type	Discrete	Continuous
Dimension	2D	3D
Coordinates	Projective	Euclidean
Resolution	High	Low

参考文献:

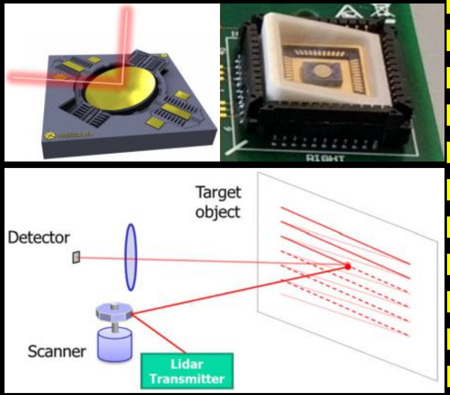
[1] Cui Y, Chen R, Chu W, et al. Deep learning for image and point cloud fusion in autonomous driving: A review

1. 传感器介绍
2. 激光雷达（-惯导）SLAM
    - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
    - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
    - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R²LIVE**）
    - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R³LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
    - ImMesh的介绍和实验演示
    - 基于ImMesh开发的应用
        - 激光雷达点云增强
        - 快速无损的场景纹理重建

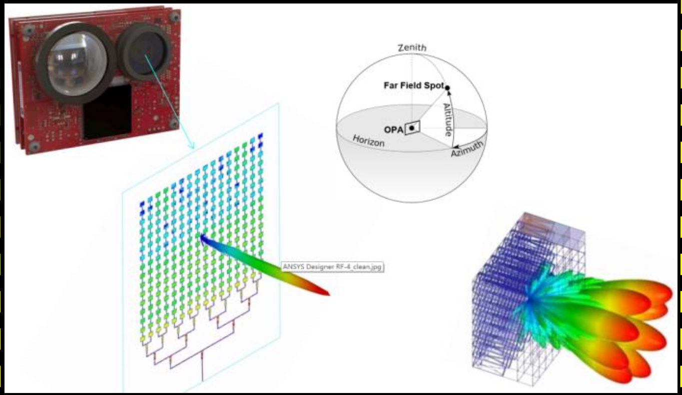
2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：

激光雷达的发展一直致力于降低设备成本同时提高其可靠性。在这种趋势中，一类越来越受关注和发展的就是各类固态LiDAR出现，如微电机系统（MEMS）扫描、光相位阵列（OPA）、Risley棱镜等实现的各类固态激光雷达

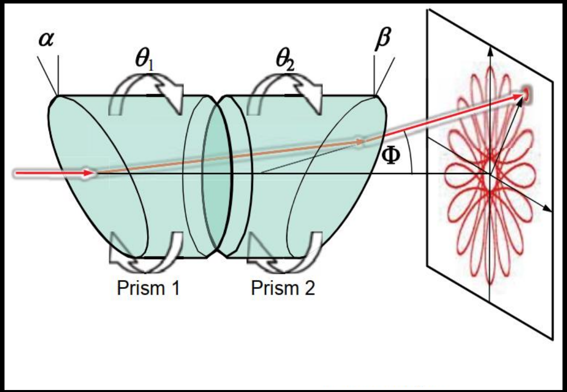
微电机系统（MEMS）扫描



光学相控位阵列（OPA）



旋转棱镜(Risley prism)



Velodyne velarray

QUANERGY *S* Serious

Livox Mid Serious

参考文献:

[1] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV

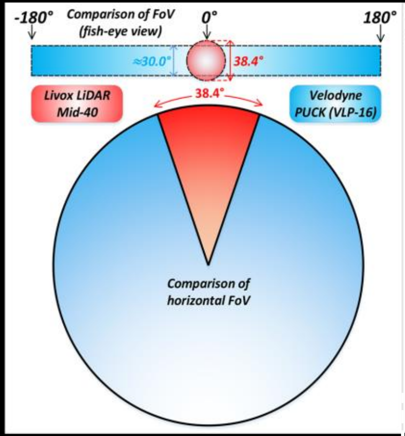
2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：

尽管固态激光雷达在成本上有着无可比拟的优势，但它们也有许多新特性，给基于固态激光雷达导航和建图带来了重大挑战。我们的loam-livox工作以livox-mid40为代表，提出了首个适用于固态激光雷达的SLAM框架。

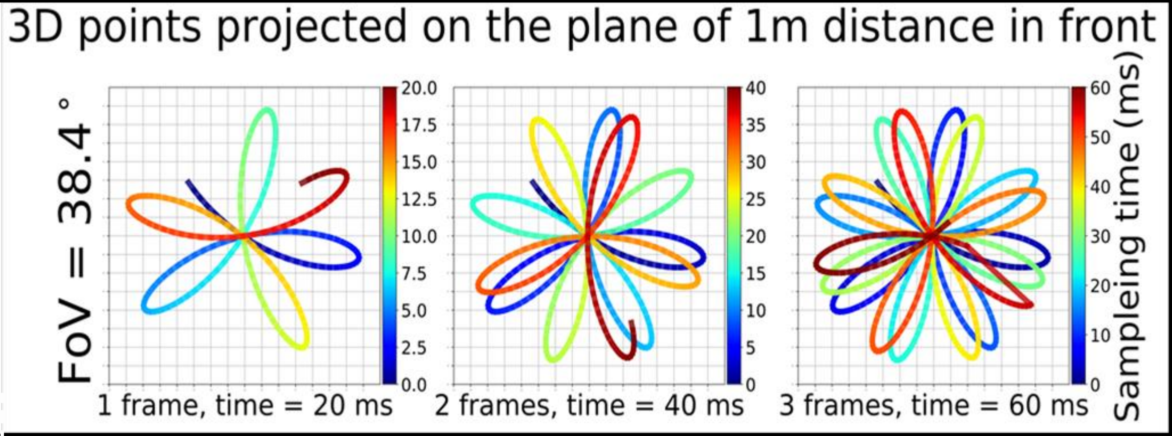
Livox-MID40的特性

(1) 小视场角(FoV)

(2) 不规则扫描模式



FoV of Livox Mid-40 and Velodyne PUCK (VLP-16)



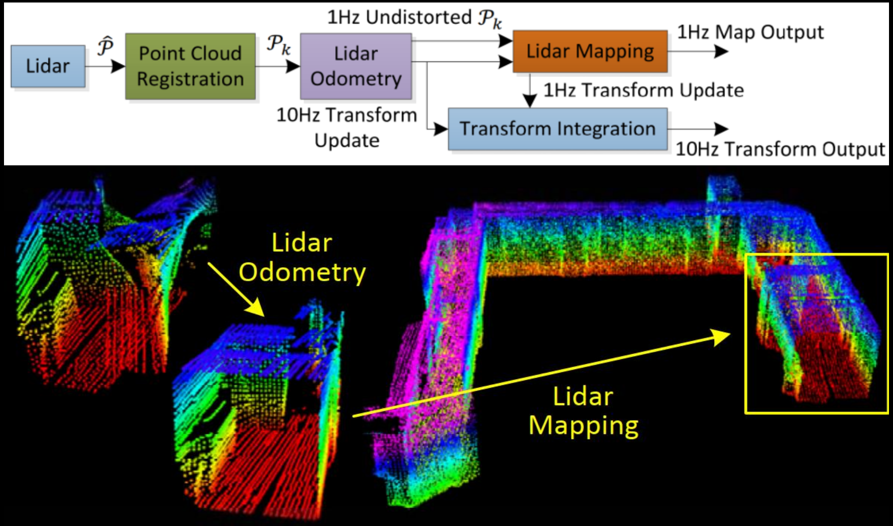
The scanning trajectory of 3D points projected on the plane of 1m distance in front, where the color encodes the sampling time.

参考文献:

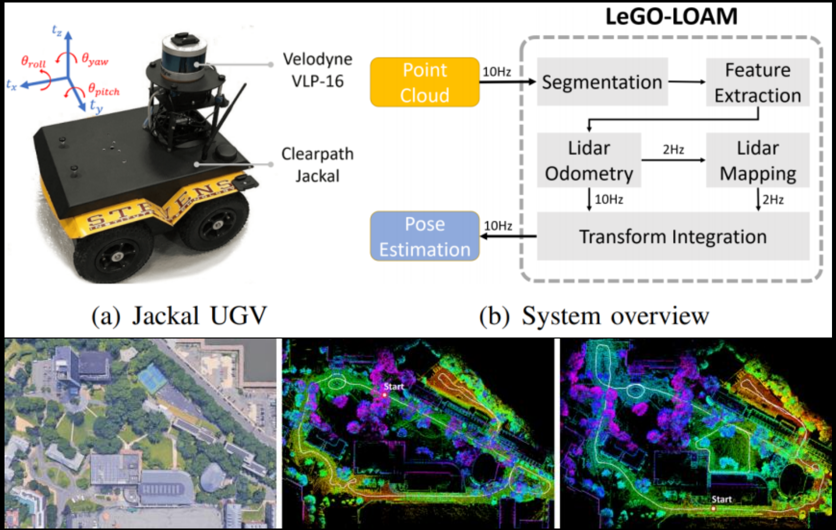
[1] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV

2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：

相关工作(Related work)



[1] J. Zhang *at el,* LOAM



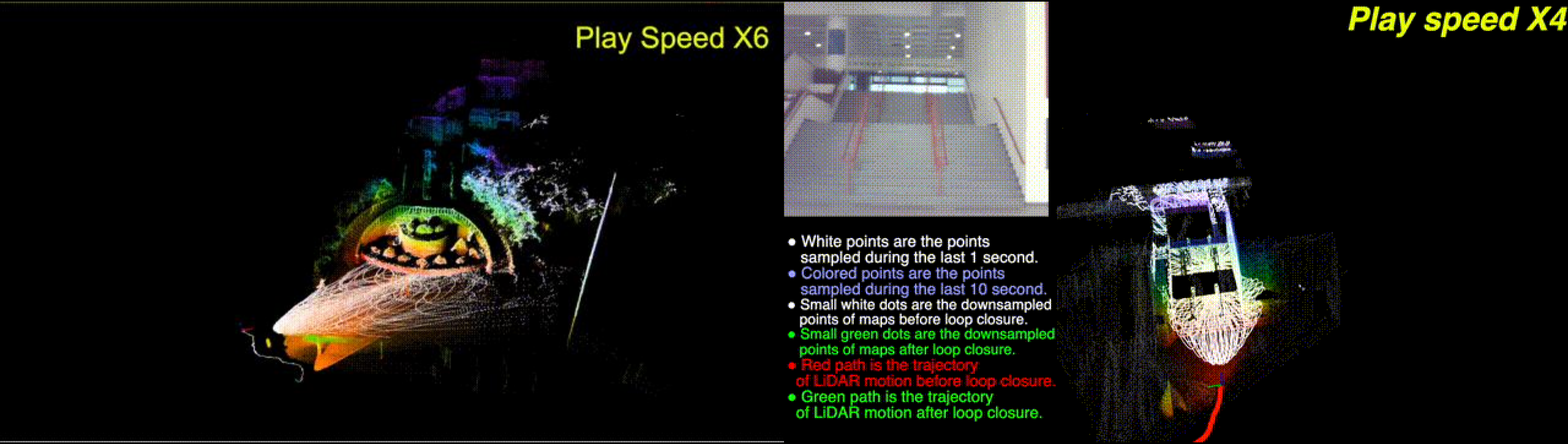
[2] T. Shan *at el, LeGO-LOAM*

参考文献:

[1] Zhang, Ji, and Sanjiv Singh. "LOAM: Lidar Odometry and Mapping in Real-time."

[2] Shan, Tixiao, and Brendan Englot. "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain."

2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：

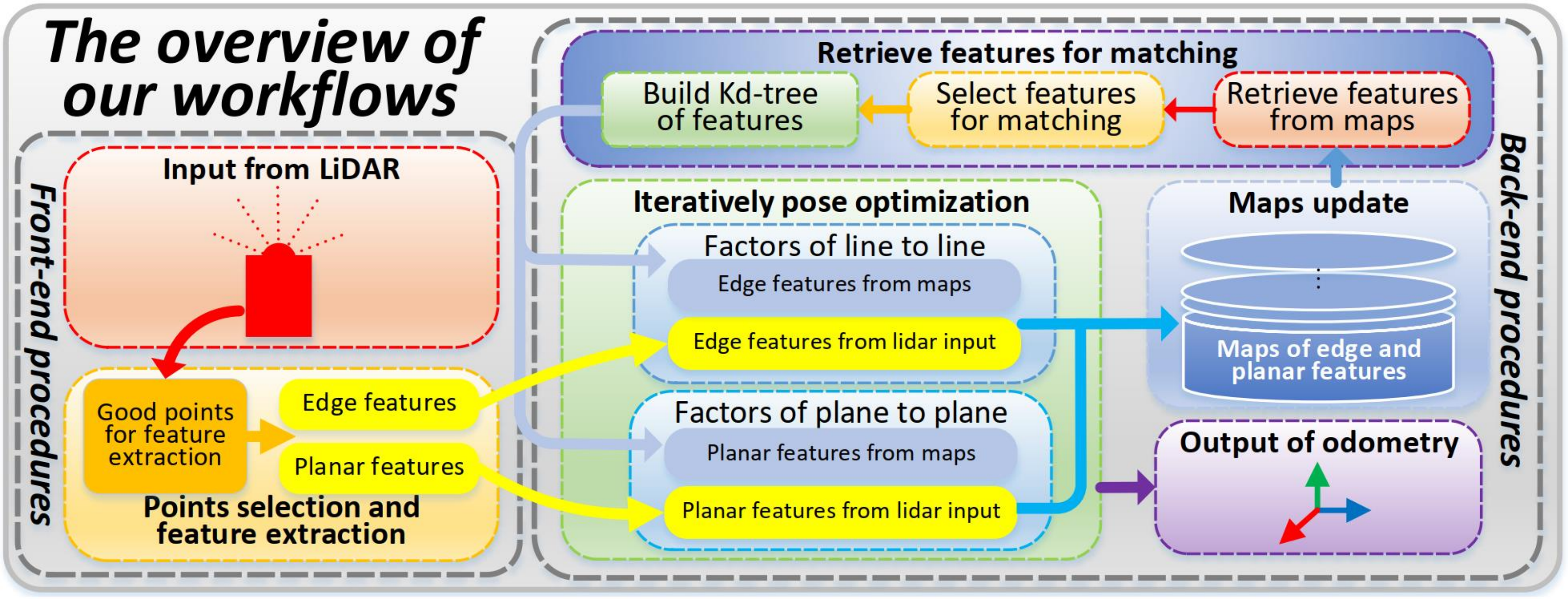


在loam-livox这个工作中，我们实现首个适用于固态激光雷达，实时鲁棒低漂移SLAM系统。他针对小FoV固态雷达在SLAM系统中出现的问题，提出了对应的解决方法: 如小FoV内雷达特征点的提取, 异常值和动态物体的剔除, 帧内点云的去运动模糊。

参考文献:

[1] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV

2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：



系统概览

参考文献:

[1] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV

2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：

特征点提取

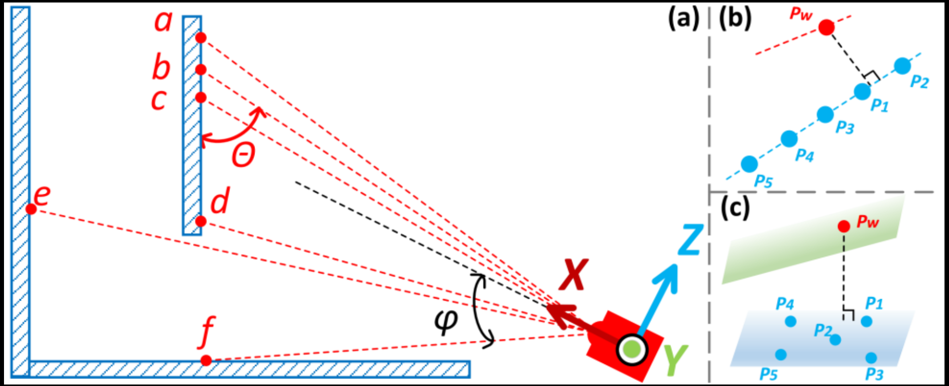


Figure : (a) Illustration of incident angle θ, deflection angle ϕ; (b)residual of edge-to-edge; (c) residual of plane-to-plane.

线到线残差

$$\mathbf{r}_{e2e} = \frac{|(\mathbf{P}_w - \mathbf{P}_5) \times (\mathbf{P}_w - \mathbf{P}_1)|}{|\mathbf{P}_5 - \mathbf{P}_1|}$$

点到面残差

$$\mathbf{r}_{p2p} = \frac{(\mathbf{P}_w - \mathbf{P}_1)^T ((\mathbf{P}_3 - \mathbf{P}_5) \times (\mathbf{P}_3 - \mathbf{P}_1))}{|(\mathbf{P}_3 - \mathbf{P}_5) \times (\mathbf{P}_3 - \mathbf{P}_1)|}$$

迭代姿态估计算法

Algorithm 1: Iterative LiDAR pose optimization

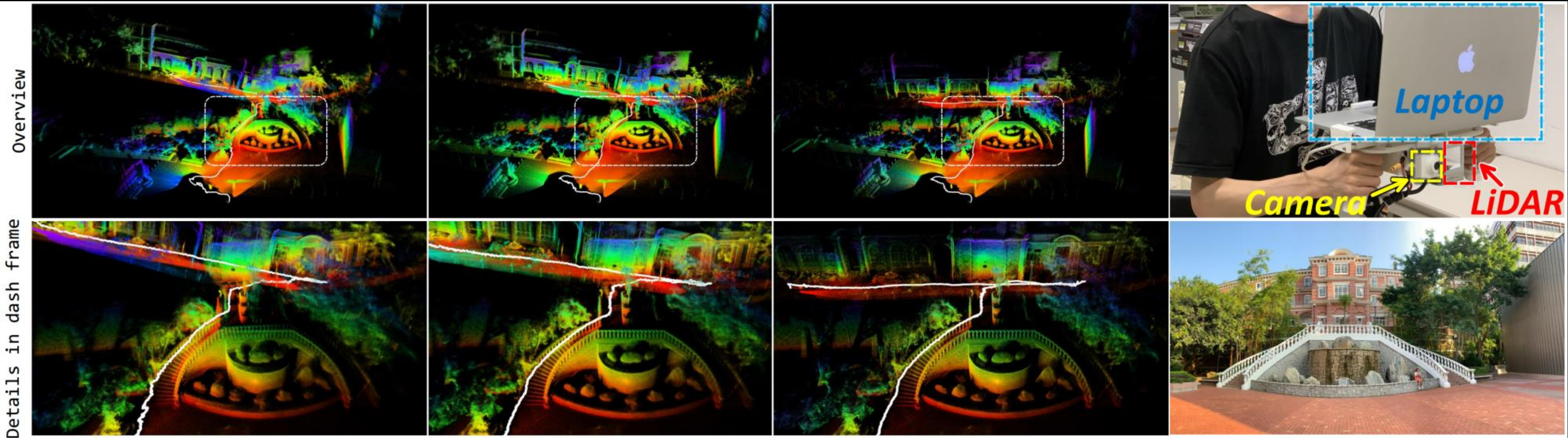
Input : The edge set $\mathcal{E}_k$ and plane set $\mathcal{P}_k$ from the current (sub-) frame; The edge set $\mathcal{E}_m$ and plane set $\mathcal{P}_m$ from maps; The LiDAR pose of the previous frame $(\mathbf{R}_{k-1}, \mathbf{t}_{k-1})$.

Output: The pose of the current frame $(\mathbf{R}_k, \mathbf{t}_k)$.

Start : $\mathbf{R}_k \leftarrow \mathbf{R}_{k-1}$, $\mathbf{t}_k \leftarrow \mathbf{t}_{k-1}$

for *Iterative pose optimization is not converged* **do**
- **for** $\mathbf{p}_l \in \mathcal{E}_k$ **do**
  - Compute $\mathbf{p}_w$ via (4) (or (8)).
  - Find 5 nearest points $\{\mathbf{p}_{1\sim5}\}$ of $\mathbf{p}_w$ in $\mathcal{E}_m$.
  - **if** $\{\mathbf{p}_{1-5}\}$ *are indeed in a line* **then**
    - Add point-to-edge residual $\mathbf{r}_{e2e}$ via (5).
- **for** $\mathbf{p}_l \in \mathcal{P}_k$ **do**
  - Compute $\mathbf{p}_w$ via (4) (or (8)).
  - Find 5 nearest points $\{\mathbf{p}_{1\sim5}\}$ of $\mathbf{p}_w$ in $\mathcal{P}_m$.
  - **if** $\{\mathbf{p}_{1\sim5}\}$ *are indeed a plane* **then**
    - Add point-to-plane residual $\mathbf{r}_{p2p}$ via (6).
- Perform pose optimization with 2 iterations.
- Recompute $\mathbf{r}_{e2e}$ and $\mathbf{r}_{p2p}$, then remove 20% of the biggest residual.
- **for** *a maximal number of iterations* **do**
  - **if** *the nonlinear optimization converges* **then**
    - Break;

2.1 首个基于固态激光雷达的SLAM系统(loam-livox)：



(a) Without motion blur compensation (b) With linear interpolation (c) With piecewise processing (d) Sampling device and scene

点云去帧内运动模糊

参考文献:

[1] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV

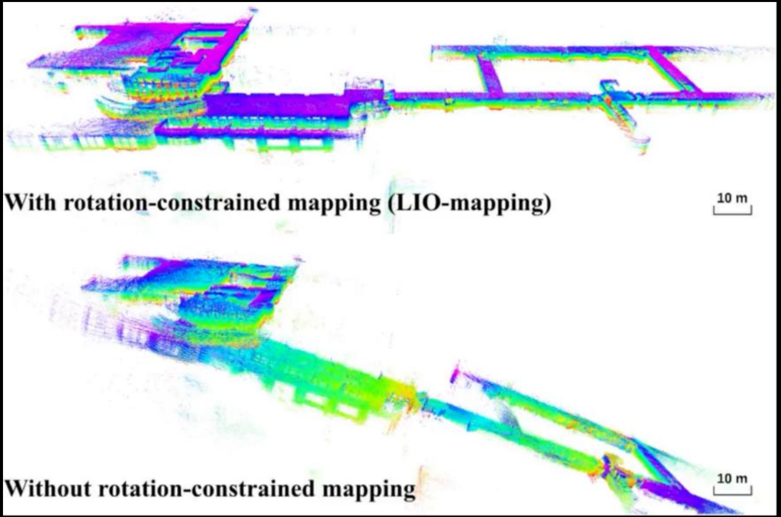
Scene 1: Long corridor in HKU campus

We evaluate the robustness of our algorithm in this scene, which is full of moving pedestrian in a long corridor.

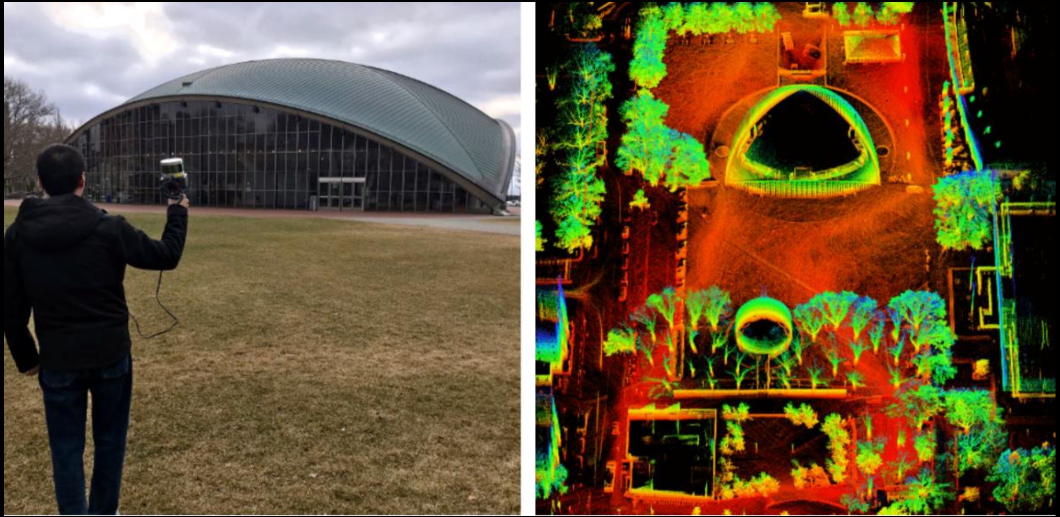
1. 传感器介绍
2. 激光雷达（-惯导）SLAM
   - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
   - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
   - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R^2LIVE**）
   - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R^3LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
   - ImMesh的介绍和实验演示
   - 基于ImMesh开发的应用
     - 激光雷达点云增强
     - 快速无损的场景纹理重建

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

相关工作(Related work)



[1] H. Ye *at el,* Lio-mapping



[2] T. Shan *at el,* LIO-SAM

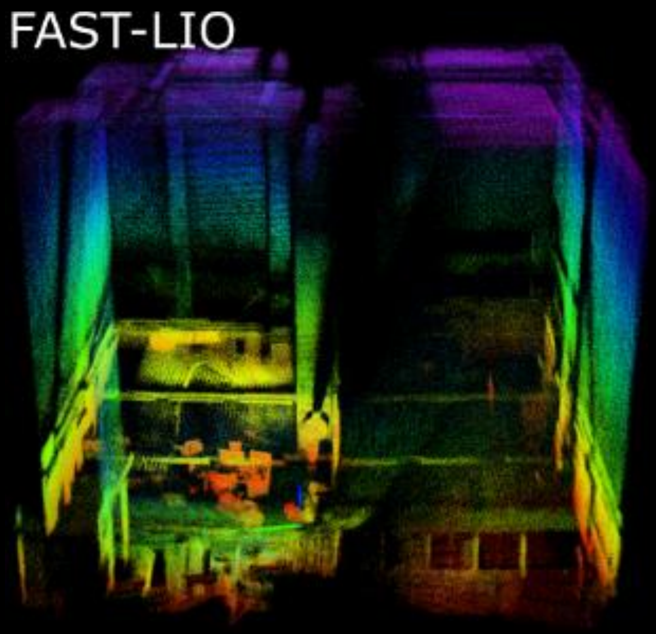
参考文献:

[1] Ye H, Chen Y, Liu M. “Tightly coupled 3d lidar inertial odometry and mapping”

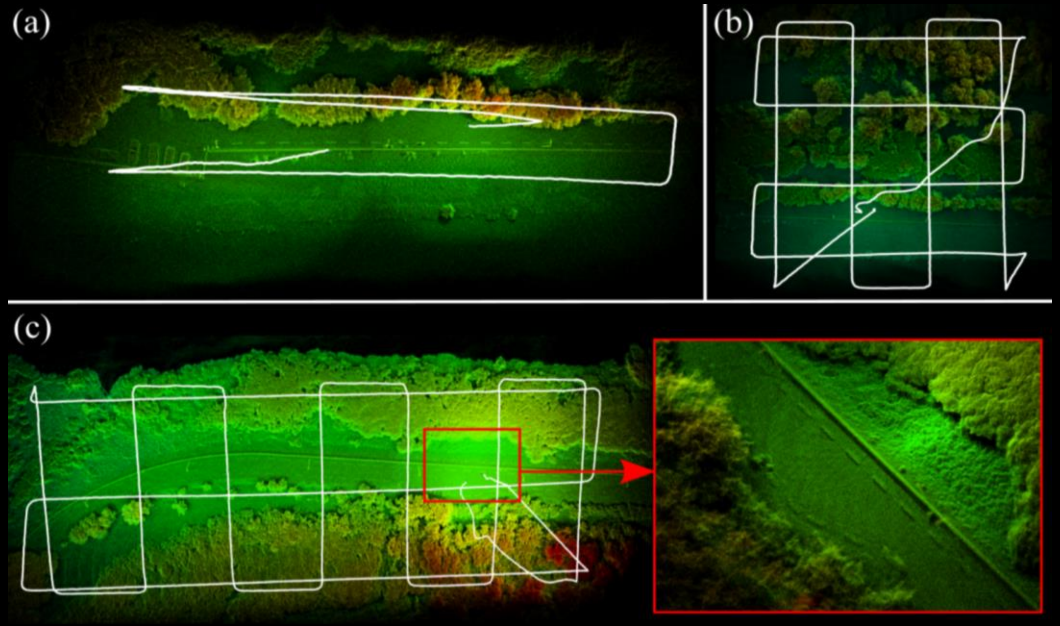
[2] Shan T, Englot B, Meyers D, et al. “**LIO-sam**: Tightly-coupled lidar inertial odometry via smoothing and mapping”

[3] Qin, Chao, et al. “**LINS**: A lidar-inertial state estimator for robust and efficient navigation."

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)



[1] FAST-LIO



[2] FAST-LIO2

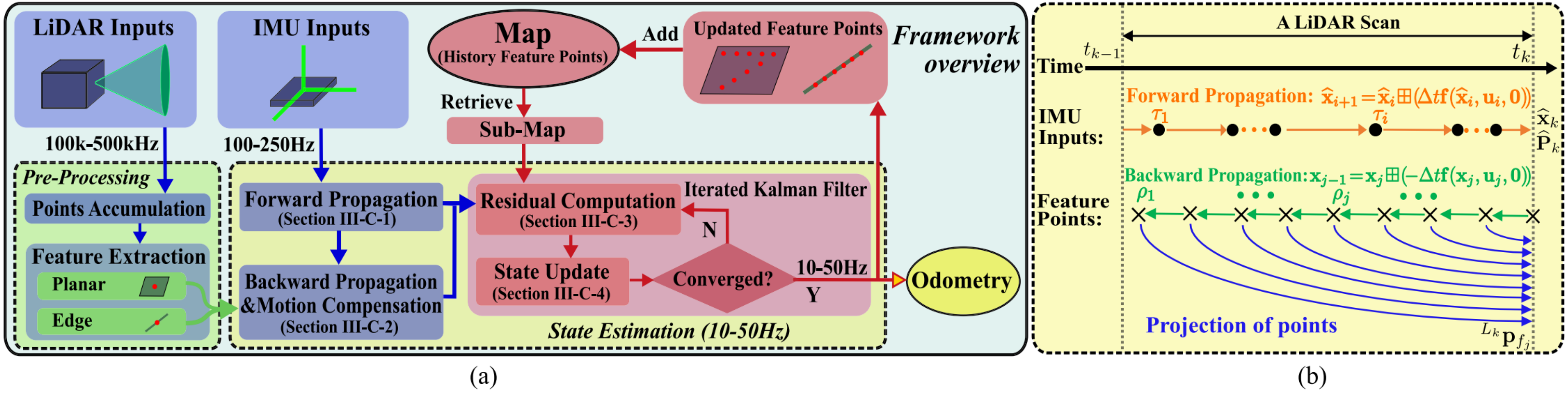
参考文献:

[1] Xu W, Zhang F. “Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter”

[2] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO: 基于紧耦合迭代卡尔曼滤波器的激光雷达-惯导里程计



(a) FAST-LIO 系统框架

(b) IMU back-propagation 去帧内点云模糊

参考文献:

[1] Xu W, Zhang F. “Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO: 等效卡尔曼增益的计算

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

$$\widehat{\mathbf{x}}_k^{\kappa+1} = \widehat{\mathbf{x}}_k^{\kappa} \boxplus \left(-\mathbf{K}\mathbf{z}_k^{\kappa} - (\mathbf{I} - \mathbf{K}\mathbf{H})(\mathbf{J}^{\kappa})^{-1}\left(\widehat{\mathbf{x}}_k^{\kappa} \boxminus \widehat{\mathbf{x}}_k\right)\right). \tag{18}$$

The updated estimate $\widehat{\mathbf{x}}_k^{\kappa+1}$ is then used to compute the residual in Section. III-C3 and repeat the process until convergence (i.e., $\|\widehat{\mathbf{x}}_k^{\kappa+1} \boxminus \widehat{\mathbf{x}}_k^{\kappa}\| < \epsilon$). After convergence, the optimal state estimation and covariance is:

$$\bar{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^{\kappa+1}, \ \bar{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P} \tag{19}$$

A problem with the commonly used Kalman gain form in (18) is that it requires to invert the matrix $\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}$ which is in the dimension of the measurements. In practice, the number of LiDAR feature points are very large in number, inverting a matrix of this size is prohibitive. As such, existing works [22], [27] only use a small number of measurements. In this letter, we show that this limitation can be avoided. The intuition originates from (17) where the cost function is over the state, hence the solution should be calculated with complexity depending on the state dimension. In fact, if directly solving (17), we can obtain the same solution in (18) but with a new form of Kalman gain shown below:

$$\mathbf{K} = \left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}^{-1}\right)^{-1}\mathbf{H}^T\mathbf{R}^{-1}. \tag{20}$$

B. Equivalent Kalman Gain Formula

Based on the matrix inverse lemma [28], we can get:

$$\left(\mathbf{P}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1} = \mathbf{P} - \mathbf{P}\mathbf{H}^T\left(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\right)^{-1}\mathbf{H}\mathbf{P}$$

Substituting above into (20), we can get:

$$\begin{aligned}\mathbf{K} &= \left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}^{-1}\right)^{-1}\mathbf{H}^T\mathbf{R}^{-1} \\ &= \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} - \mathbf{P}\mathbf{H}^T\left(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\right)^{-1}\mathbf{H}\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\end{aligned}$$

Now note that $\mathbf{H}\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} = (\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})\mathbf{R}^{-1} - \mathbf{I}$. Substituting it into above, we can get the standard Kalman gain formula in (18), as shown below.

$$\begin{aligned}\mathbf{K} &= \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{P}\mathbf{H}^T\left(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\right)^{-1} \\ &= \mathbf{P}\mathbf{H}^T\left(\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}\right)^{-1}.\end{aligned}$$

■

参考文献:

[1] Xu W, Zhang F. “Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter”

[2] N. J. Higham, “*Accuracy and Stability of Numerical Algorithms.*”

[3] B. M. Bell and F. W. Cathey, “The iterated kalman filter update as a gauss-newton method”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO: 实验结果

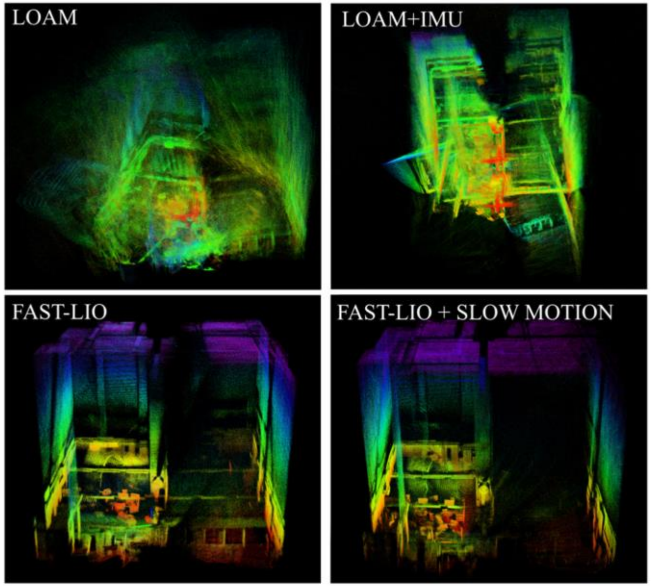


Fig. 5. The Mapping results of different LIO packages in an indoor environment with large rotation speed.

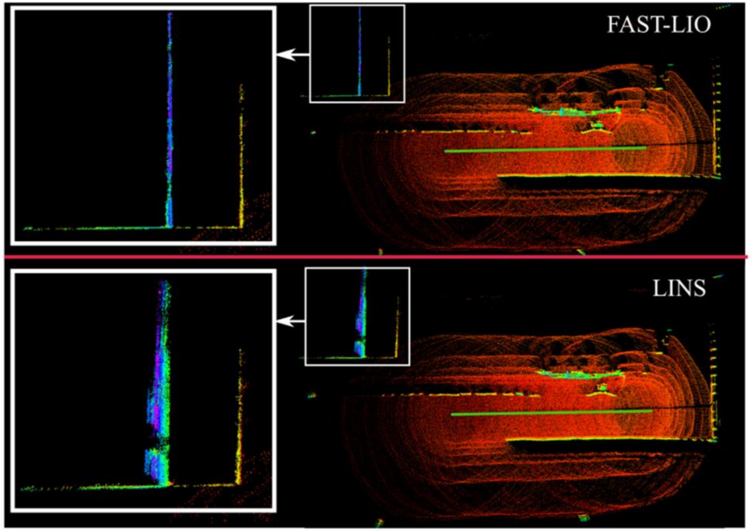


Fig. 7. Comparison between LINS [22] and FAST-LIO. The data is from [22] and is collected by a Velodyne VLP-16 LiDAR and an Xsens MTiG-710 IMU, the green straight line in the center is the odometry output.

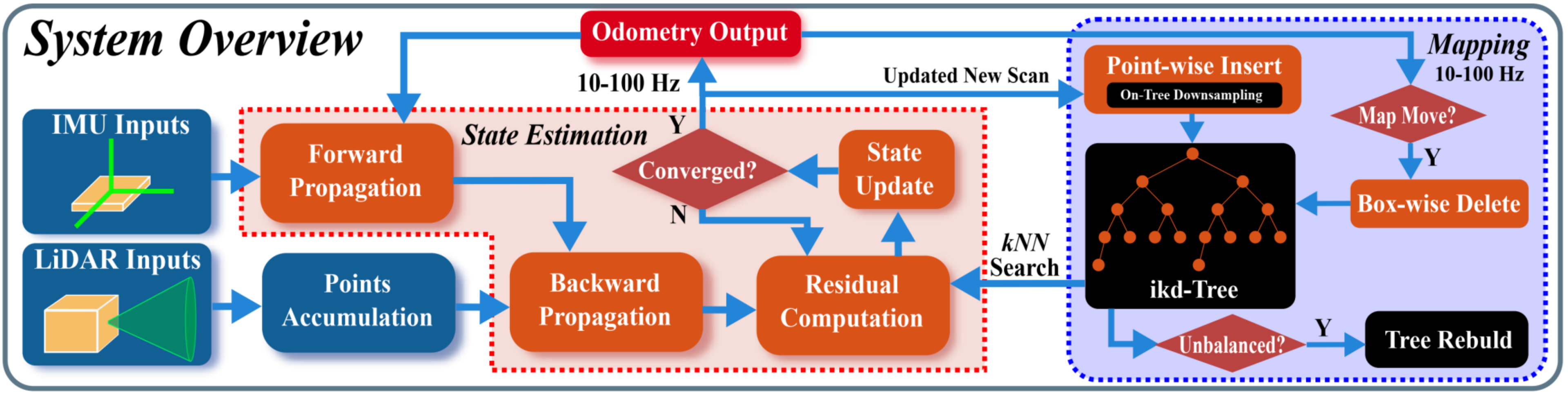
参考文献:

[1] Xu W, Zhang F. "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter"

[2] Qin, Chao, et al. "**LINS**: A lidar-inertial state estimator for robust and efficient navigation."

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计



系统概览

参考文献:

[1] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

[2] Cai Y, Xu W, Zhang F. ikd-Tree: An incremental KD tree for robotic applications

[3] He D, Xu W, Zhang F. Kalman Filters on Differentiable Manifolds

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计

KD Tree:

- 平衡树结构
- K最近邻搜索的最优数据结构
- **不支持增量式更新** ☹

ikd-Tree[2]:

- **支持增量式更新** 😊
  - **动态插入(Insert)**
  - **动态删除(Delete)**
- 动态树平衡 😊
  - 替罪羊树[3]

参考文献:

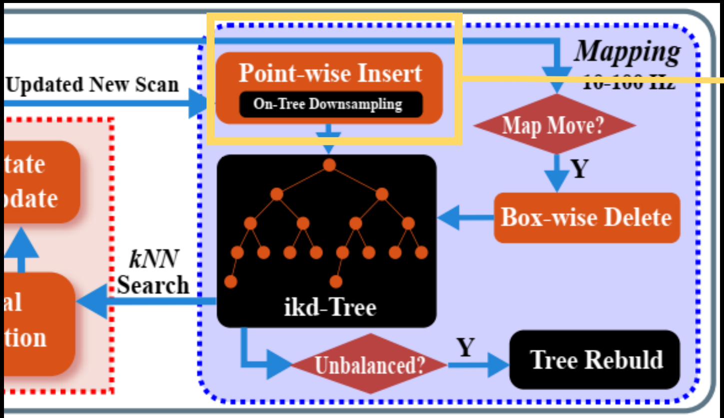
[1] Xu W, Cai Y, He D, Lin, J, Zhang F "Fast-lio2: Fast direct lidar-inertial Odometry"

[2] Cai Y, Xu W, Zhang F. "ikd-Tree: An incremental KD tree for robotic applications"

[3] Galperin I, Rivest R L. "Scapegoat Trees"

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计



ikd-Tree动态点插入

Algorithm 2: Point Insertion with On-tree Downsampling.

```
Input: Downsample Resolution l,
       New Point to Insert p,
       Switch of Parallelly Re-building SW
Algorithm Start
    C_D ← FindCube(l, p)
    p_center ← Center(C_D);
    V ← BoxwiseSearch(RootNode, C_D);
    V.push(p);
    p_nearest ← FindNearest (V, p_center);
    BoxwiseDelete(RootNode, C_D)
    Insert(RootNode, p_nearest, NULL, SW);
Algorithm End

Function Insert(T, p, father, SW)
    if T is empty then
        Initialize(T, p, father);
    else
        ax ← T.axis;
        if p[ax] < T.point[ax] then
            Insert(T.leftchild, p, T, SW);
        else
            Insert(T.rightchild, p, T, SW);
        end
        AttributeUpdate(T);
        Rebalance(T, SW);
    end
End Function
```

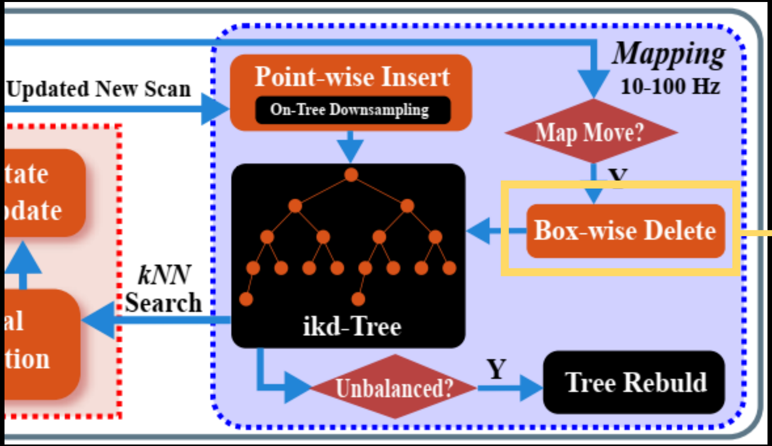

参考文献:

[1] Xu W, Cai Y, He D, Lin, J, Zhang F "Fast-lio2: Fast direct lidar-inertial Odometry"

[2] Cai Y, Xu W, Zhang F. "ikd-Tree: An incremental KD tree for robotic applications"

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计



ikd-Tree区域删除

Algorithm 3: Box-wise Delete.

Input : Operation Cuboid $\mathbf{C}_O$,
k-d Tree Node T,
Switch of Parallelly Re-building SW

Function `BoxwiseDelete(`T, $\mathbf{C}_O$, SW`)`
  $\mathbf{C}_T \leftarrow T.range$;
  if $\mathbf{C}_T \cap \mathbf{C}_O = \varnothing$ **then return**;
  if $\mathbf{C}_T \subseteq \mathbf{C}_O$ **then**
    `T.treedelete`, `T.delete` ← true;
    `T.invalidnum = T.treesize`;
  else
    p ← `T.point`;
    if p $\subset \mathbf{C}_O$ **then** `T.treedelete = ` true;
    `BoxwiseDelete(`$T.leftchild$, $\mathbf{C}_O$, SW`)`;
    `BoxwiseDelete(`$T.rightchild$, $\mathbf{C}_O$, SW`)`;
    `AttributeUpdate(`T`)`;
    `Rebalance(`T, SW`)`;
  end
End Function

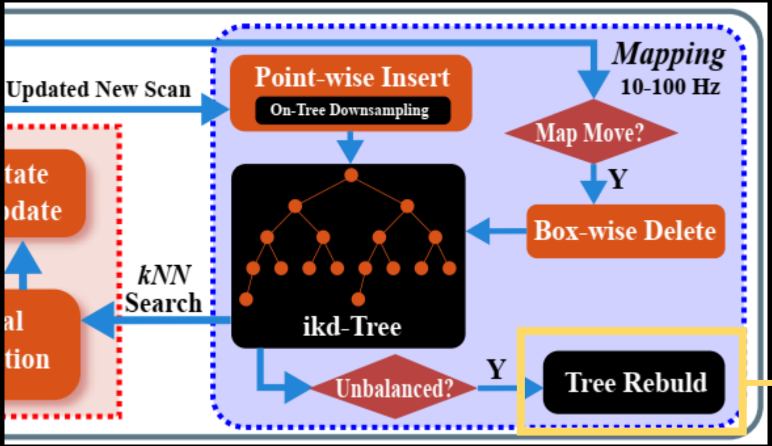
参考文献:

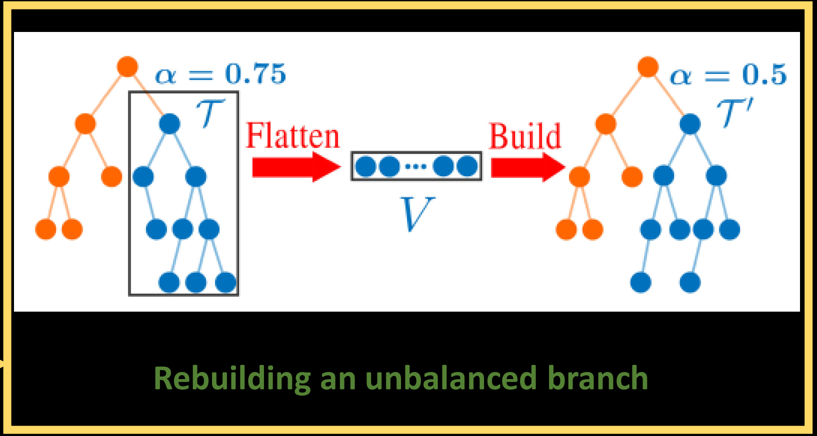
[1] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

[2] Cai Y, Xu W, Zhang F. “ikd-Tree: An incremental KD tree for robotic applications”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计





ikd-Tree动态树平衡

参考文献:

[1] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

[2] Cai Y, Xu W, Zhang F. “ikd-Tree: An incremental KD tree for robotic applications”

[3] Galperin I, Rivest R L. “Scapegoat Trees”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计

	FAST-LIO2 (2000)	FAST-LIO2 (1000)	FAST-LIO2 (800)	FAST-LIO2 (600)	FAST-LIO2 (Feature)	FAST-LIO2 (ARM)	LILI-OM		LIO-SAM		LINS	
	Total	Total	Total	Total	Total	Total	Odo.	Map.	Odo.	Map.	Odo.	Map.
lili_6	13.15	**12.56**	13.22	15.92	15.35	45.58	68.95	58.46	—	—	—	—
lili_7	**16.93**	17.61	20.39	19.72	21.13	65.89	40.01	83.71	—	—	—	—
lili_8	**14.73**	15.31	17.73	17.15	18.37	57.29	61.80	79.11	—	—	—	—
utbm_8	21.72	22.05	21.39	**20.82**	21.16	100.00	65.29	84.76	—	—	37.44	153.92
utbm_9	28.26	25.44	21.41	21.35	**17.46**	91.05	68.94	97.90	—	—	38.82	154.06
utbm_10	23.90	22.48	23.09	20.74	**15.30**	94.62	66.10	97.29	—	—	33.61	166.12
ulhk_4	20.86	20.14	19.96	**20.04**	29.35	91.12	52.40	74.80	39.50	95.29	34.72	93.70
ulhk_5	24.10	23.90	23.96	**23.75**	28.70	68.04	53.56	47.68	25.68	127.63	28.01	99.13
ulhk_6	30.52	31.56	30.15	29.25	31.94	92.38	64.46	70.43	**15.16**	164.36	41.54	199.96
nclt_4	15.65	15.72	15.79	15.75	19.98	69.09	62.49	98.46	**13.38**	184.03	46.43	188.40
nclt_5	16.56	16.60	16.61	16.58	**13.54**	68.95	67.64	83.34	19.09	184.46	47.83	198.88
nclt_6	15.92	15.84	15.83	15.68	**14.72**	66.64	76.10	133.25	×	×	54.48	195.31
nclt_7	16.79	16.87	16.82	16.63	**15.16**	70.24	67.65	81.69	29.50	211.18	56.94	197.71
nclt_8	14.29	14.25	14.32	14.14	**7.94**	57.03	53.54	57.54	16.30	163.09	53.53	144.95
nclt_9	13.73	13.65	13.60	13.64	**10.30**	54.82	42.84	68.86	12.79	118.35	46.12	149.45
nclt_10	21.85	21.79	21.78	21.61	**20.62**	89.65	82.92	130.96	23.13	324.62	83.12	252.68
liosam_1	16.95	14.77	14.65	16.19	15.93	60.60	48.45	84.28	**13.47**	135.39	24.13	179.44
liosam_2	**11.11**	11.47	11.52	11.19	19.68	45.27	42.58	99.01	13.09	154.69	20.71	160.66
liosam_3	19.38	16.64	12.00	13.01	12.37	44.26	38.42	64.02	**11.32**	124.35	40.47	117.25

The bold values stand for the best result of each data sequence.

耗时评估--平均每雷达帧的处理时间

参考文献:

[1] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计

	utbm_8	*utbm_9*	*utbm_10*	*ulhk_4*	*nclt_4*	*nclt_5*	*nclt_6*	*nclt_7*	*nclt_8*	*nclt_9*	*nclt_10*	*liosam_1*
FAST-LIO2 (2000m)	**25.3**	**51.6**	16.89	2.57	3.15	5.41	7.54	2.59	8.21	5.72	1.68	4.62
FAST-LIO2 (1000m)	27.29	**51.6**	**16.8**	2.57	3.21	5.42	7.55	**2.21**	**5.88**	5.56	**1.62**	**4.58**
FAST-LIO2 (800m)	25.8	51.86	17.23	2.57	3.25	**5.4**	7.55	2.49	6.49	5.95	1.67	**4.58**
FAST-LIO2 (600m)	27.75	52.09	17.3	2.57	3.05	5.41	7.58	2.47	6.47	5.8	1.69	**4.58**
FAST-LIO2 (Feature)	27.21	53.81	22.59	2.61	**2.86**	6.43	**7.41**	2.63	9.36	6.09	1.71	7.85
LILI-OM	59.48	782.11	17.59	**2.29**	11.6	11.5	275	13.1	21.5	**5.2**	68.9	18.78
LIO-SAM	—[a]	—	—	3.52	1163	6.82	×[b]	23.3	27.0	7.9	1525	4.75
LINS	48.17	54.35	60.48	3.11	60.8	1135	128.76	397.2	107.3	11.86	3155	880.92

[a] Dataset *utbm* does not produce the attitude quaternion data, which is necessary for LIO-SAM; therefore, LIO-SAM does not work on all the sequences in *utbm* dataset, denoted as –.

[b] ×denotes that the system totally failed.

The bold values stand for the best result of each data sequence.

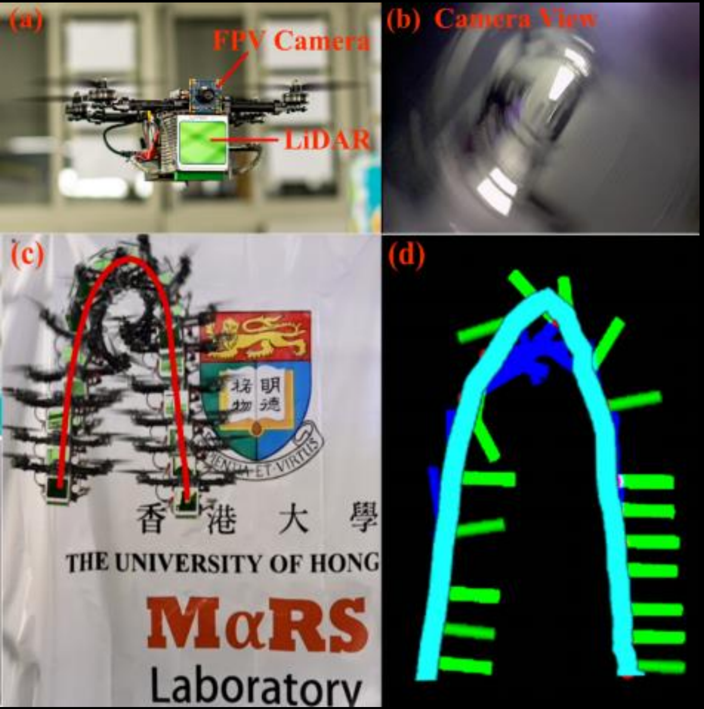
精度评估—绝对位移误差(RMSE)

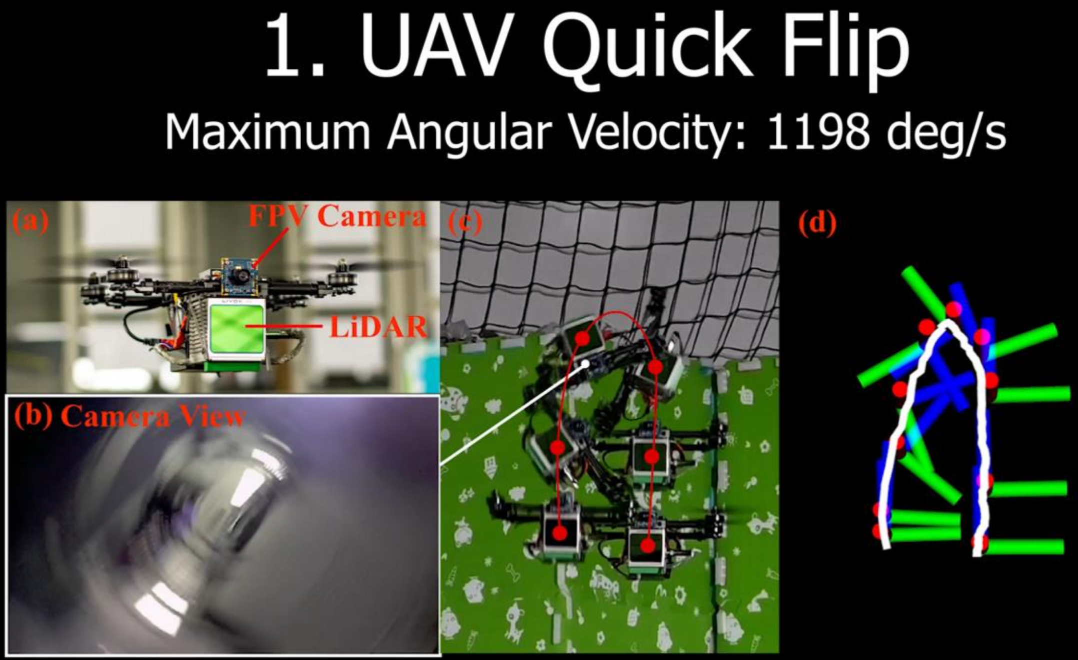
参考文献:

[1] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计





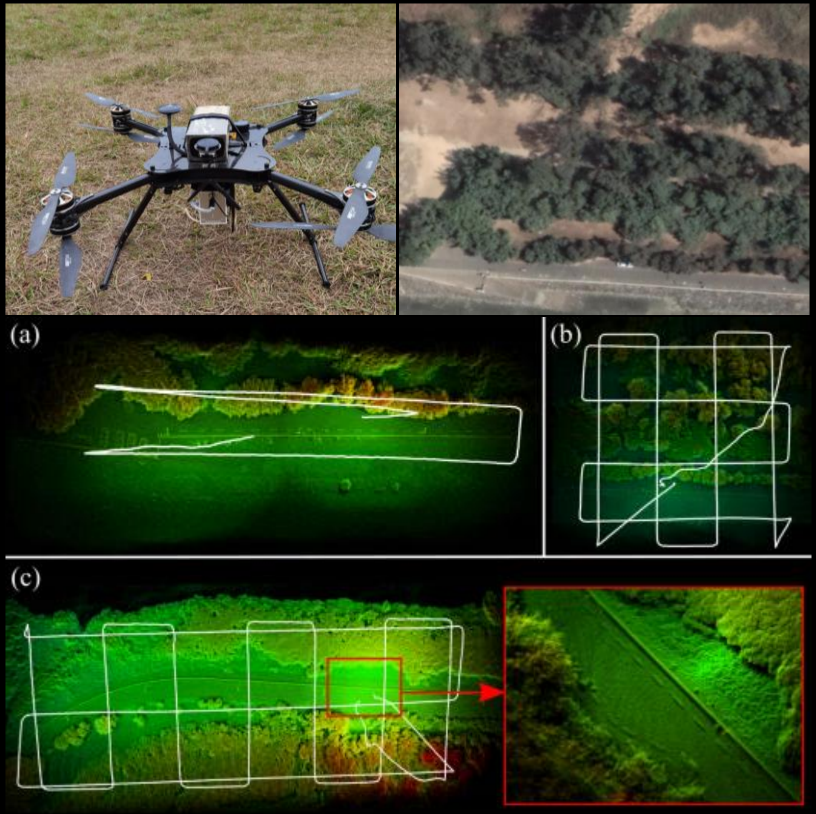
真机实验

参考文献:

[1] Xu W, Cai Y, He D, Lin, J, Zhang F “Fast-lio2: Fast direct lidar-inertial Odometry”

2.2 激光雷达-惯导紧耦合的里程计(FAST-LIO)

FAST-LIO2: 快速、直接激光雷达-惯导紧耦合里程计



野外测量

2.3 里程计(FAST-LIO) 在无人机上的应用

雷达无人机在线规避动态细小障碍物

Avoiding dynamic small obstacles with onboard sensing and computating on aerial robots

Fanze Kong*, Wei Xu*,Yixi Cai and Fu Zhang

香港大學
Univerisity of Hong Kong

MaRS
Laboratory

参考文献:

[1] Kong, Fanze, et al. "Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots."

2.3 里程计(FAST-LIO) 在无人机上的应用

雷达无人机在线高速避障飞行

参考文献:

[1] Ren, Yunfan, et al. "Bubble Planner: Planning High-speed Smooth Quadrotor Trajectories using Receding Corridors."

2.3 里程计(FAST-LIO) 在无人机上的应用

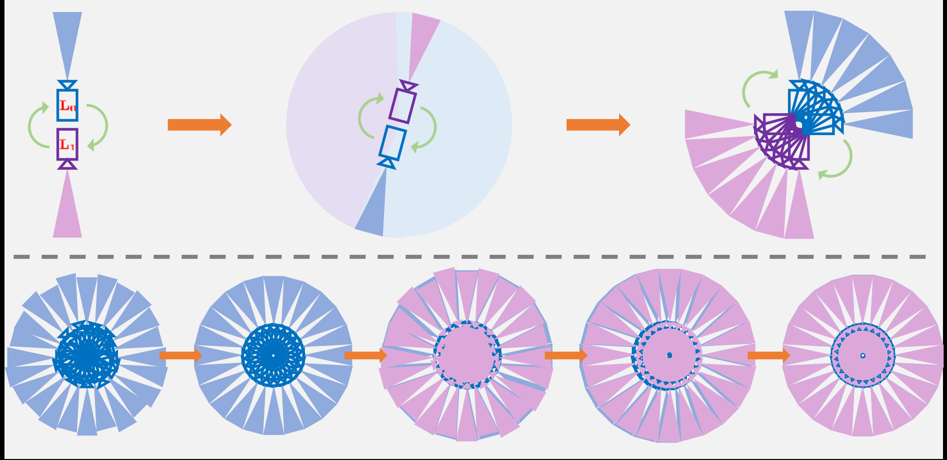
雷达无人机在线全姿态运动规划

参考文献:

[1] Ren, Yunfan, et al. "Online Whole-body Motion Planning for Quadrotor using Multi-resolution Search."

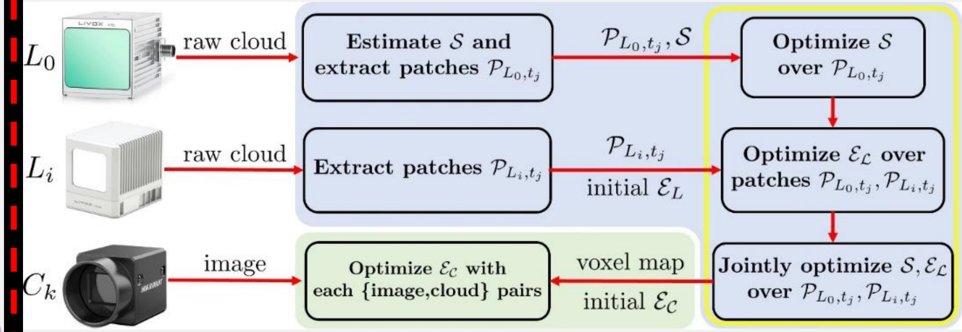
2.3 扩展工作：多个激光雷达-相机之间的外参标定(mlcc)

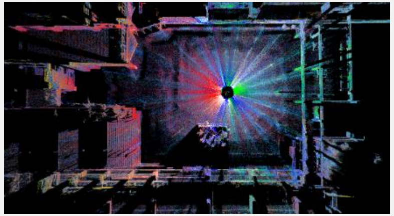
多雷达外参标定：通过旋转产生共视关系，通过最小化不同雷达点云之间点到面距离来优化雷达外参[1][3]



L0雷达位姿优化建图　　L1雷达外参粗优化　　雷达外参位姿联合优化

多雷达-相机外参标定：通过自适应体素提取雷达连续边缘特征，通过投影雷达边缘特征到相机平面对齐来优化雷达-相机外参[1][2]





多雷达优化后建图

多雷达相机联合优化后建图

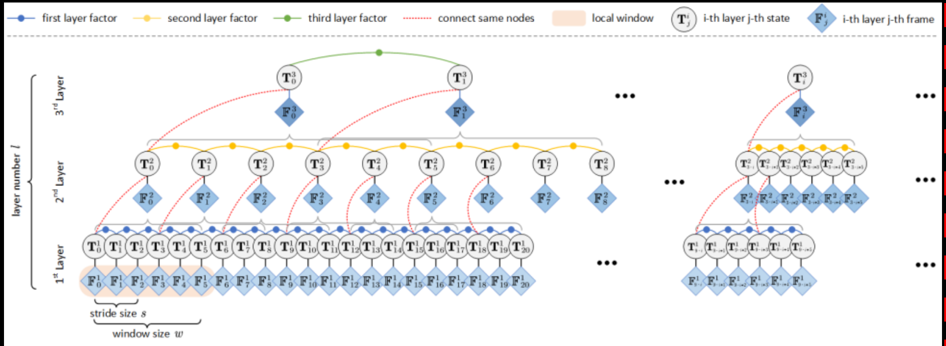
参考文献:

[1] X. Liu and F. Zhang, “Extrinsic Calibration of Multiple LiDARs of Small FoV in Targetless Environments”.

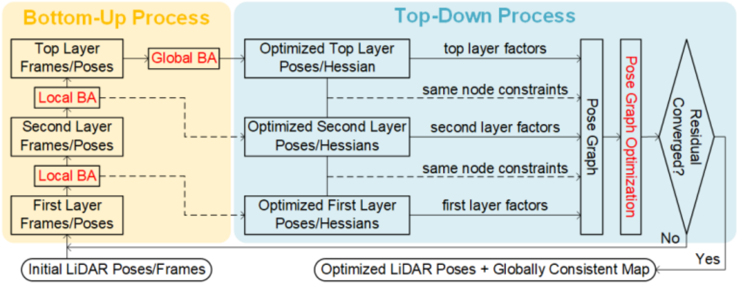
[2] C. Yuan, X. Liu, X. Hong and F. Zhang, “Pixel-Level Extrinsic Self Calibration of High Resolution LiDAR and Camera in Targetless Environments”.

[3] X. Liu, C. Yuan and F. Zhang, “Targetless Extrinsic Calibration of Multiple Small FoV LiDARs and Cameras Using Adaptive Voxelization”.

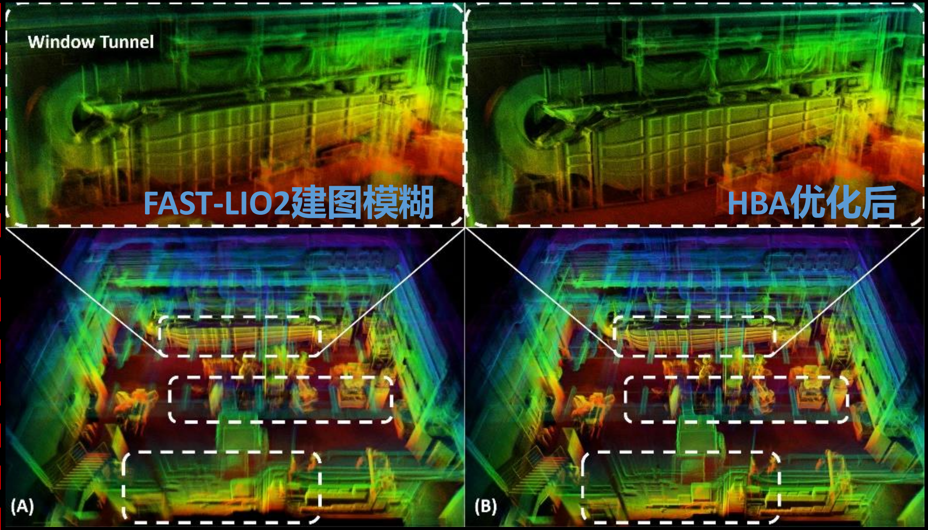
2.3 扩展工作：适用于大场景的激光雷达高精建图(HBA)

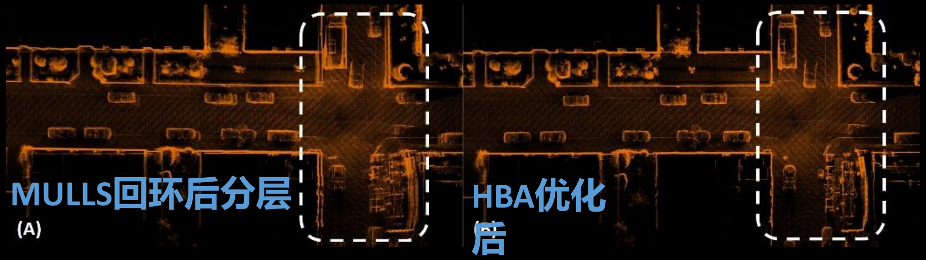


通过类似金字塔一样的设计，将大场景全局BA转换成多个小场景BA，精度保持不变，同时大幅降低耗时[1][2]



自下而上的金字塔BA和自上而下的位姿图优化



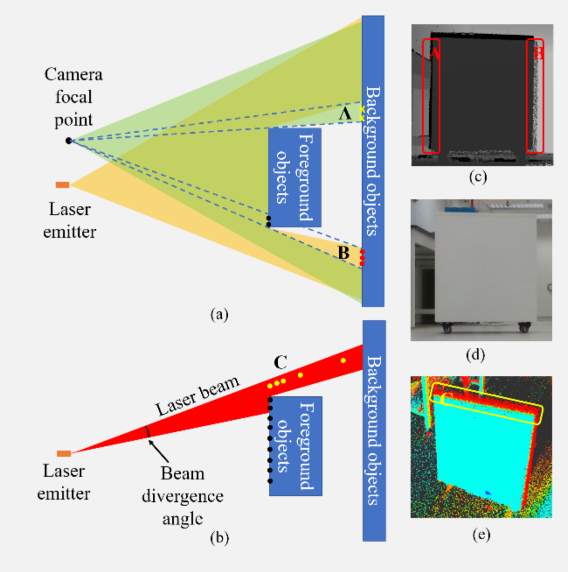


参考文献:

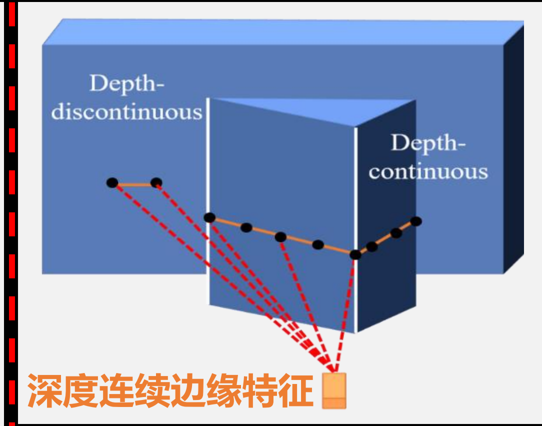
[1] X. Liu, Z. Liu, F. Kong and F. Zhang, “Large-Scale LiDAR Consistent Mapping using Hierarchical LiDAR Bundle Adjustment”.

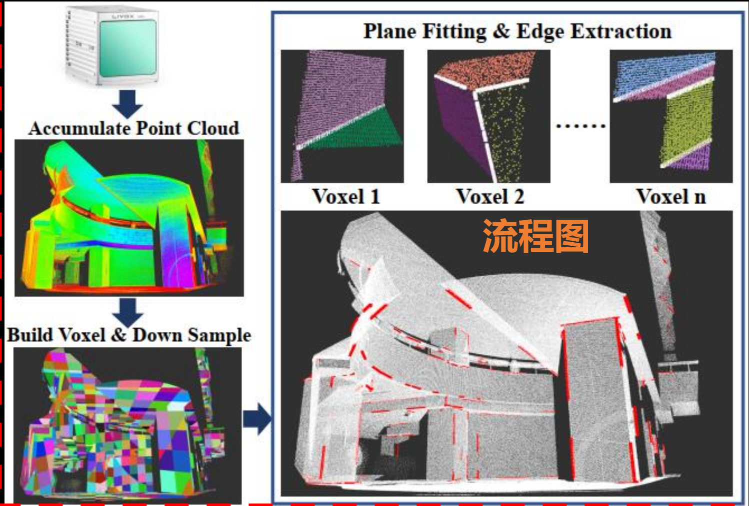
[2] Z. Liu, X. Liu and F. Zhang, “Efficient and Consistent Bundle Adjustment on Lidar Point Clouds”.

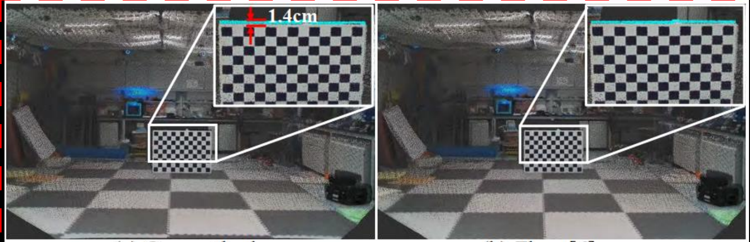
2.3 扩展工作：激光雷达-相机之间的外参标定(livox_camera_calib)



雷达深度不连续边缘特征点在与图片匹配时会产生较大误差







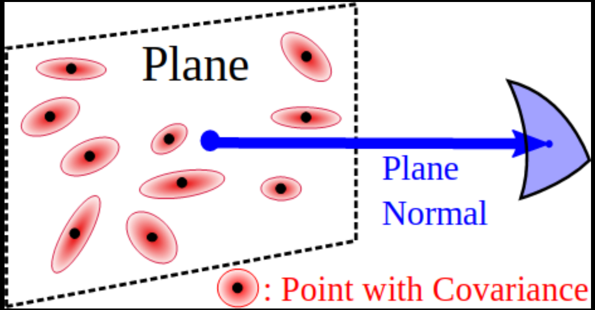
(a) Our method (b) Zhou [6]

我们方法 标定板方法

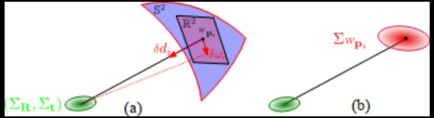
参考文献:

[1] C. Yuan, X. Liu, X. Hong and F. Zhang, “Pixel-Level Extrinsic Self Calibration of High Resolution LiDAR and Camera in Targetless Environments”.

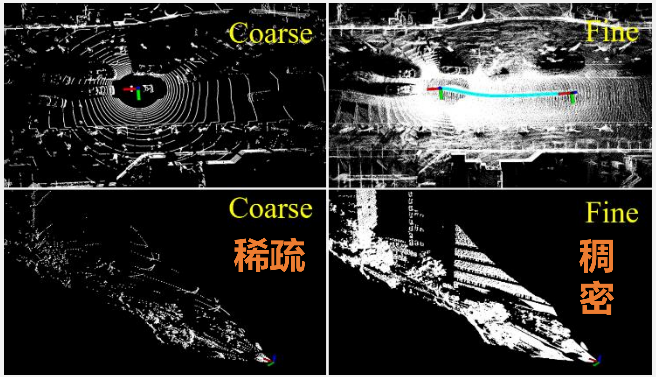
2.3 扩展工作：高效自适应体素激光雷达里程计(VoxelMap)



每个voxel内的平面特征噪声由每个点的噪声及雷达位姿噪声构成



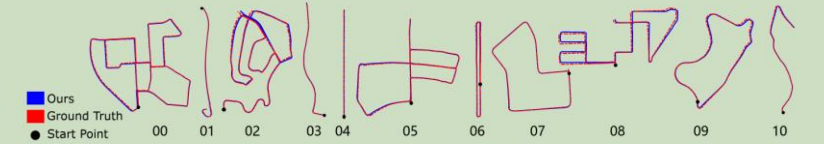
平面法向量估计不确定性，平面深度不确定性，雷达点测量不确定性



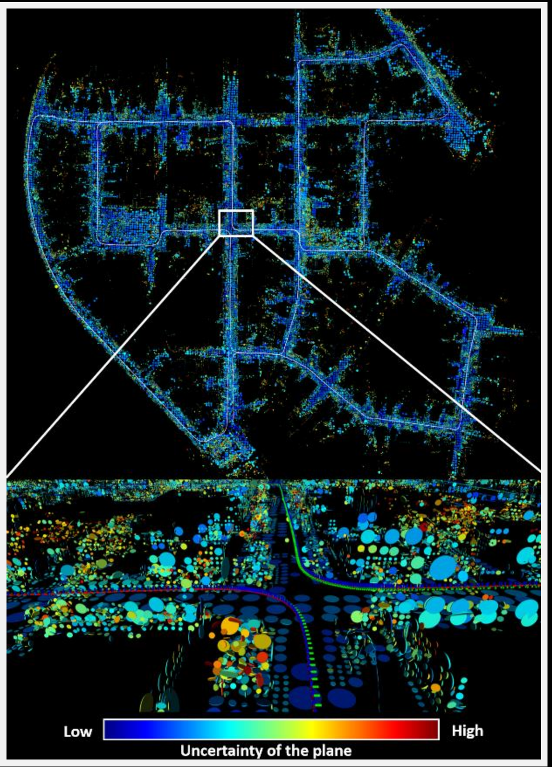
雷达点稀疏时，先建立粗略估计的平面特征地图，随着点增加，不断完善估计

TABLE II: Accuracy (ATE in meters) Comparison on KITTI Odometry Training Sequences

Approach (Total length [m])	00 (3724)	01 (2453)	02 (5067)	03 (560)	04 (393)	05 (2205)	06 (1232)	07 (694)	08 (3222)	09 (1705)	10 (919)	Avg. of all scans [deg]/[m]
Ours (full)	**0.9/2.8**	1.9/7.8	**1.7/6.1**	1.2/**0.7**	0.6/**0.3**	0.8/**1.2**	**0.4/0.4**	0.7/0.7	**1.1/2.3**	**1.0/1.9**	1.0/1.1	**1.2/2.9**
Ours (w/o adaptive)	1.1/3.1	2.0/8.2	1.7/7.1	1.2/0.7	0.7/0/5	0.8/1.3	0.5/0.5	0.9/0.7	1.8/3.1	1.1/2.0	1.3/1.2	1.3/3.4
Ours (w/o prob.)	2.2/3.9	2.5/9.5	1.9/8.1	1.4/0.9	0.7/0.4	1.0/1.2	0.5/0.6	1.0/0.6	3.0 /6.8	1.2/1.9	1.2/1.5	1.8/4.5
LiTAMIN2 [24]	1.6/5.8	3.5/15.9	2.7/10.7	2.6/0.8	2.3/0.7	1.1/2.4	1.1/0.9	1.0/0.6	1.3/2.5	1.7/2.1	1.2/**1.0**	1.8/5.1
MULLS [30]	1.7/6.1	**1.0/2.4**	2.4/10.6	**0.7**/0.7	**0.2**/0.9	1.0/2.4	0.4/0.6	**0.5**/0.6	1.9/4.3	1.4/2.5	**0.5**/1.1	1.5/4.8
Suma [14]	1.0/2.9	3.2/13.8	2.2/8.4	1.5/0.9	1.8/0.4	**0.7**/1.2	0.4/0.4	0.7/**0.5**	1.5/2.8	1.1/2.9	0.8/1.3	1.4/3.9
FAST-LIO2 [13]	2.0/3.8	2.5/11.9	2.8/12.8	1.2/0.8	0.7/0.5	1.2/2.1	1.1/1.2	1.2/0.8	1.8/3.0	1.4/2.0	1.1/1.5	1.8/4.9
Lego-Loam [7]	2.8/6.3	3.8/119.4	4.1/14.7	4.1/0.9	3.3/0.8	1.9/2.8	1.4/0.8	1.5/0.7	2.5/3.5	2.2/2.1	1.9/1.8	2.8/11.1



KITTI数据集轨迹精度



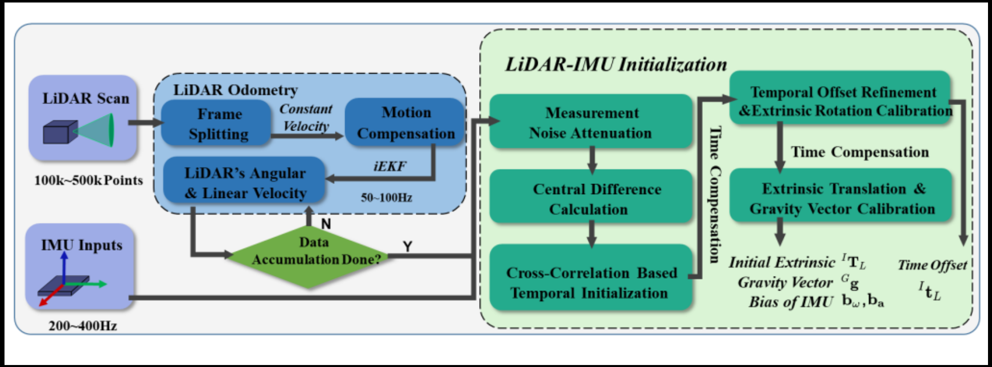
地图使用带概率估计的平面特征表示

参考文献:

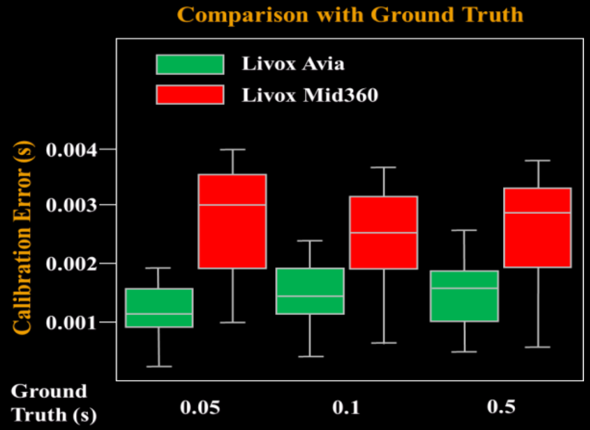
[1] C. Yuan, W. Xu, X. Liu, X. Hong and F. Zhang, “Efficient and Probabilistic Adaptive Voxel Mapping for Accurate Online LiDAR Odometry”.

2.3 扩展工作：鲁棒、实时的激光雷达IMU初始化方法(LI-Init)

算法流程

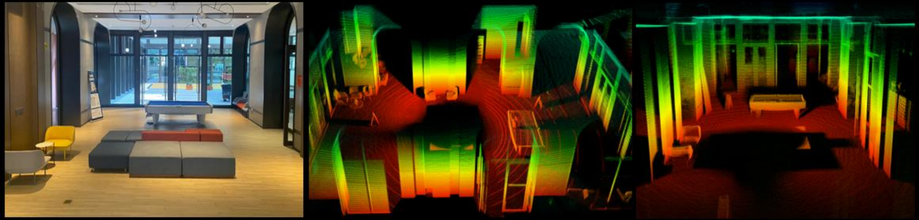


时差标定



外参标定

Method	Rotation Error(deg)	Translation Error (m)	Processing Time (s)	Length of Dataset
Ours	0.6208	0.0162	10.2	40 seconds
LI-Calib [2]	1.0375	×	332.6	
Target-free [3]	0.8483	0.0187	115.7	



The calibration scene. Accurate pointcloud map constructed by our LiDAR odometry.

[1] Zhu, Fangcheng, Yunfan Ren, and Fu Zhang. "Robust real-time lidar-inertial initialization." 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022.

开源项目地址：https://github.com/hku-mars/LiDAR_IMU_Init

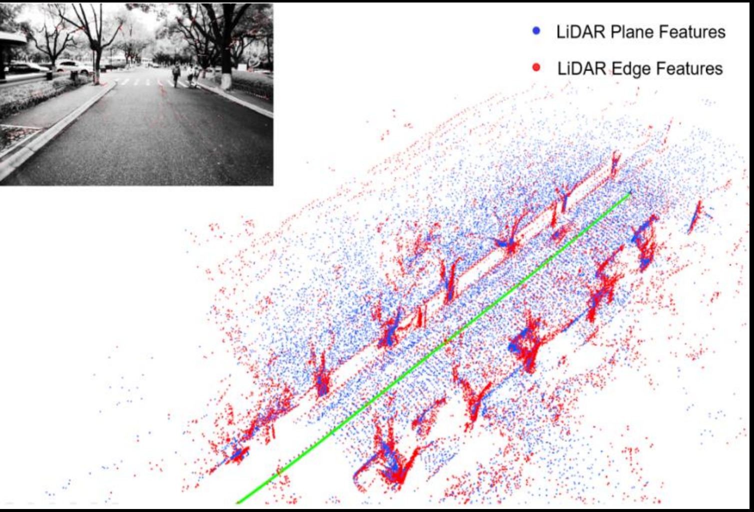
[2] J. Lv, J. Xu, K. Hu, Y. Liu, and X. Zuo. Target-less calibration of lidar-imu system based on continuous-time batch estimation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 9968–9975. IEEE, 2020

[3] S. Mishra, G. Pandey, and S. Saripalli. Target-free extrinsic calibration of a 3d-lidar and an imu. In 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pages 1–7. IEEE, 2021.

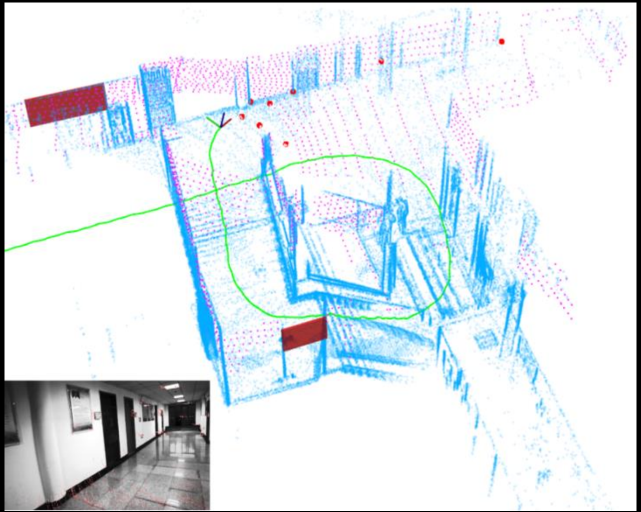
1. 传感器介绍
2. 激光雷达（-惯导）SLAM
   - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
   - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
   - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R²LIVE**）
   - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R³LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
   - ImMesh的介绍和实验演示
   - 基于ImMesh开发的应用
     - ➢ 激光雷达点云增强
     - ➢ 快速无损的场景纹理重建

3.1 多传感器（激光雷达-惯导-视觉）融合

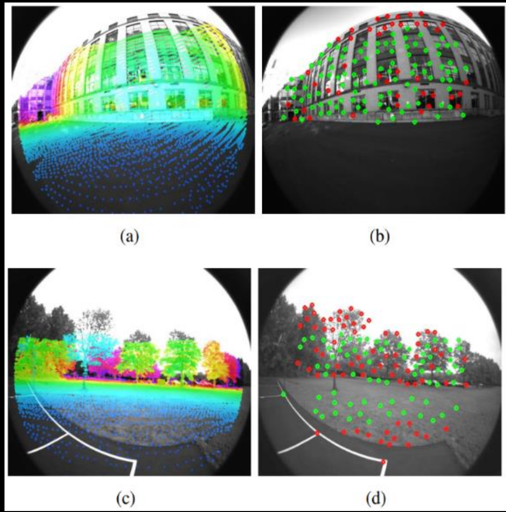
相关工作



[1] LIC-Fusion



[2] LIC-Fusion 2.0



[3] LVI-SAM

参考文献：

[1] Zuo X, Geneva P, Lee W, et al. “Lic-fusion: Lidar-inertial-camera Odometry”

[2] Zuo X, Yang Y, Geneva P, et al. “Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking”

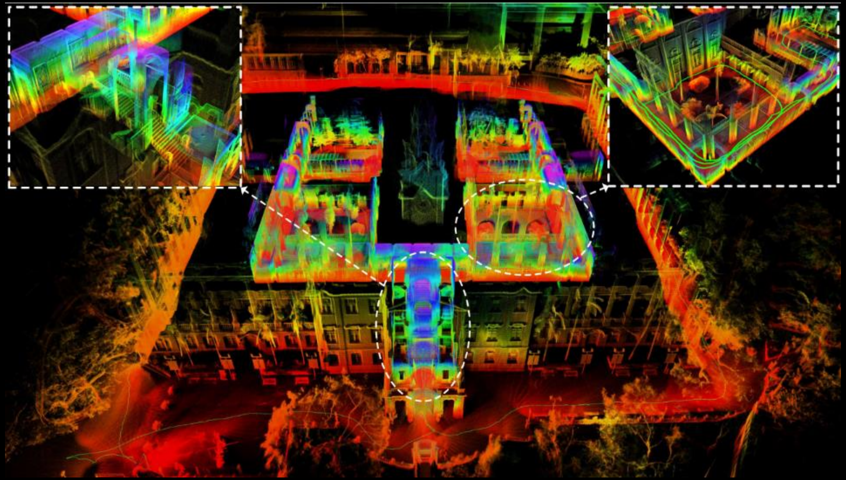
[3] Shan T, Englot B, Ratti C, et al. “Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping”

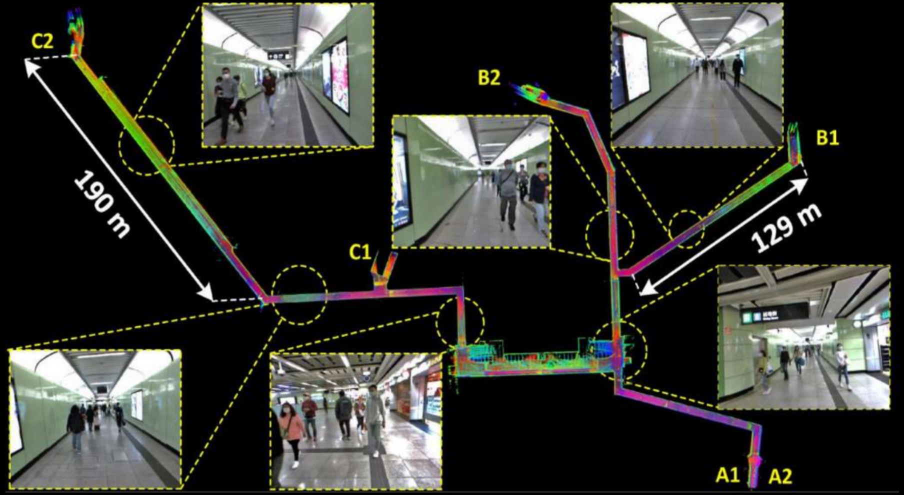
3.1. R^2LIVE: A Robust, Real-time, LiDAR-Inertial-Visual tightly-coupled state Estimator and mapping

Jiarong Lin, Chunrang Zheng, Wei Xu and Fu Zhang

All our codes will be available at:

https://github.com/hku-mars/r2live





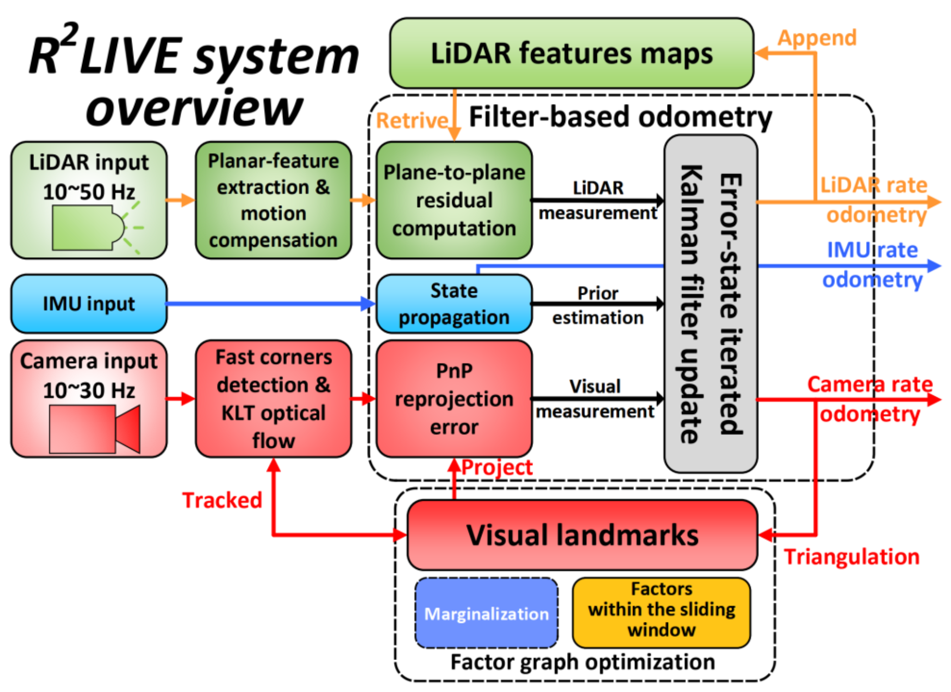
MαRS Laboratory

香 港 大 學

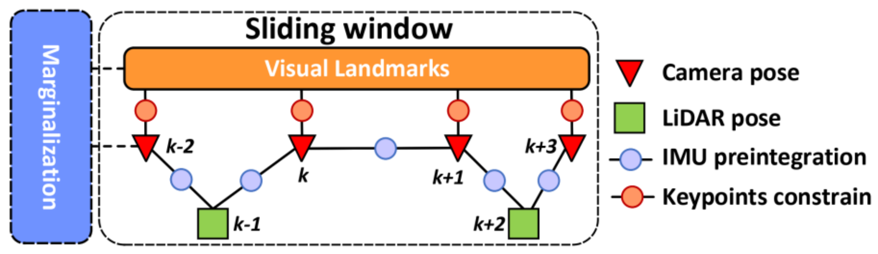
THE UNIVERSITY OF HONG KONG

3.1 多传感器（激光雷达-惯导-视觉）融合(R²LIVE)

R²LIVE: 实时鲁棒激光雷达-惯导-视觉紧耦合SLAM框架



（系统框图）



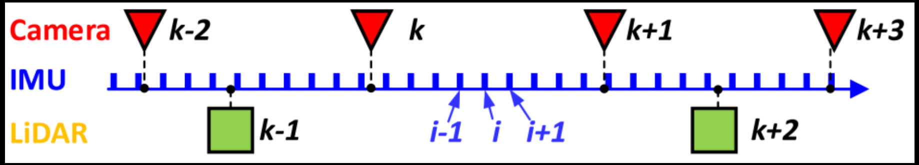
（后端图优化框图）

参考文献：

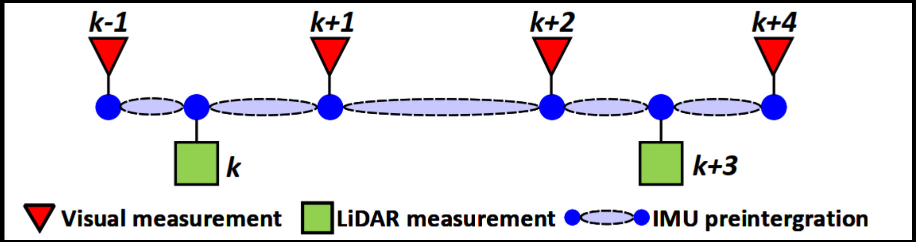
[1] Lin J, Zheng C, Xu W, et al. “R²LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping”

3.1 多传感器（激光雷达-惯导-视觉）融合(R^2LIVE)

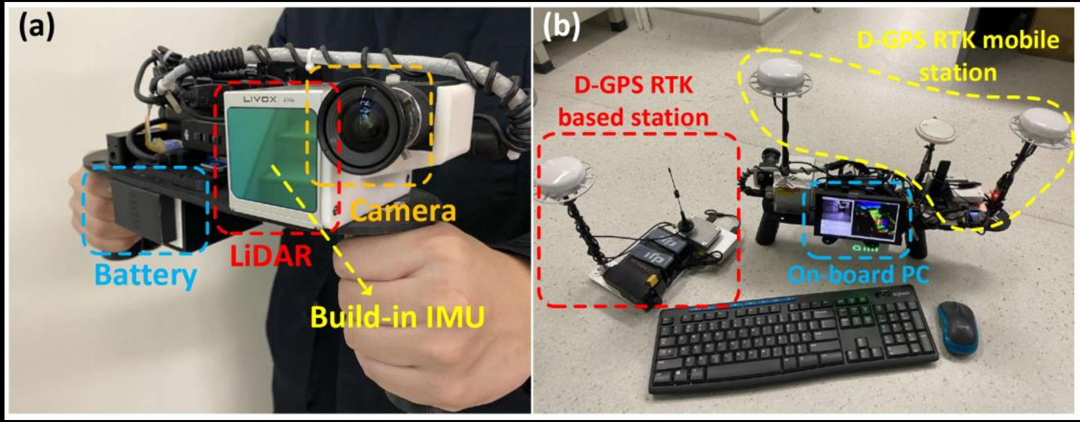
R^2LIVE: 实时鲁棒激光雷达-惯导-视觉紧耦合SLAM框架



(传感器时序图)



(迭代卡尔曼滤波器更新示意图)



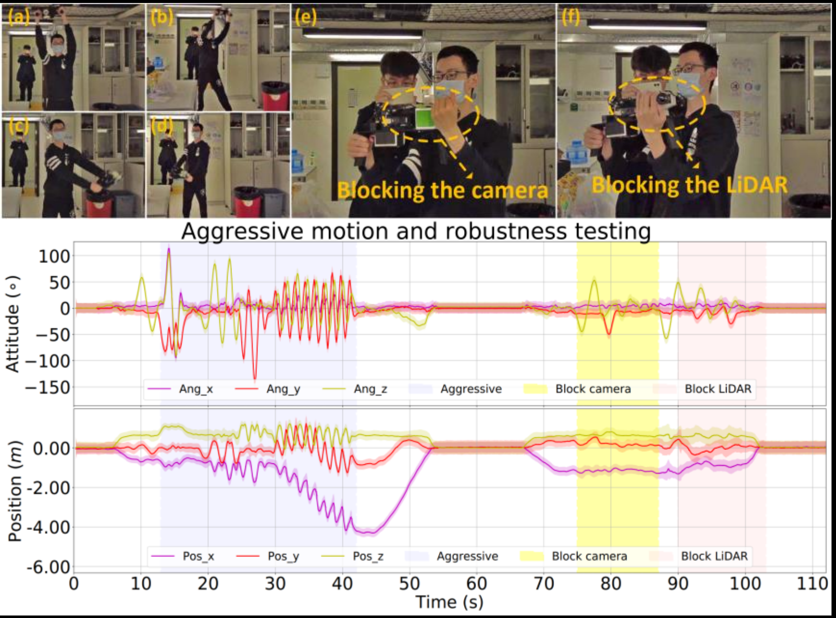
(实验设备)

参考文献：

[1] Lin J, Zheng C, Xu W, et al. "R^2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping"

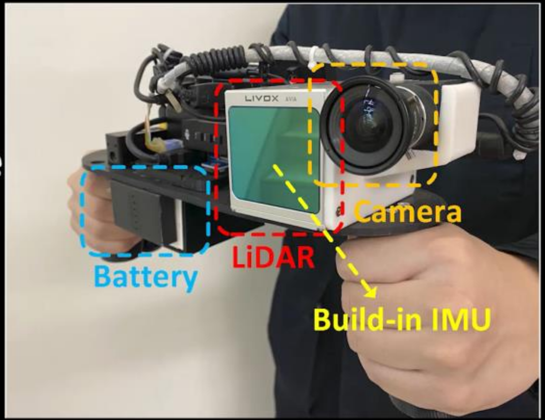
3.1 多传感器（激光雷达-惯导-视觉）融合(R^2LIVE)

R^2LIVE: 实时鲁棒激光雷达-惯导-视觉紧耦合SLAM框架



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

Our handheld device for data collection



（鲁棒性实验测试—状态估计结果）

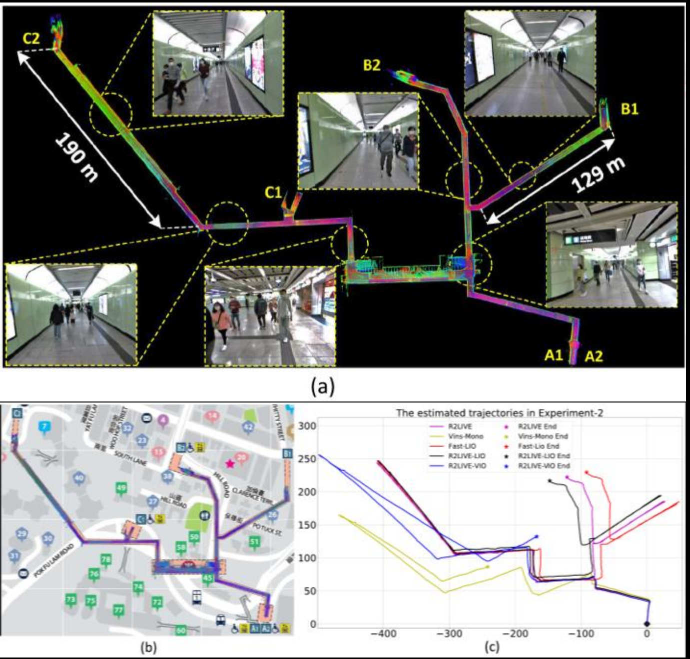
（鲁棒性实验测试—视频）

参考文献：

[1] Lin J, Zheng C, Xu W, et al. "R^2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping"

3.1 多传感器（激光雷达-惯导-视觉）融合(R^2LIVE)

R^2LIVE: 实时鲁棒激光雷达-惯导-视觉紧耦合SLAM框架



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

（鲁棒性实验测试—地铁长走廊测试）

参考文献：

[1] Lin J, Zheng C, Xu W, et al. “R^2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping”

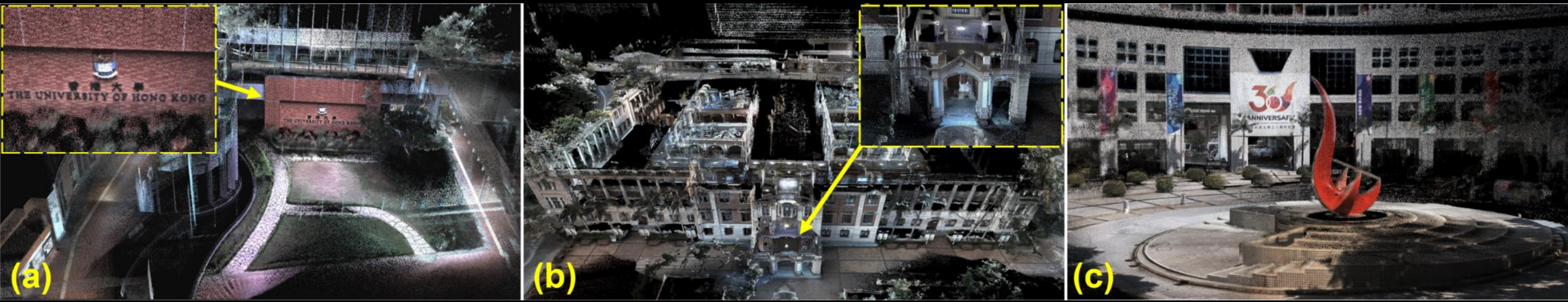
1. 传感器介绍
2. 激光雷达（-惯导）SLAM
   - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
   - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
   - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R²LIVE**）
   - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R³LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
   - ImMesh的介绍和实验演示
   - 基于ImMesh开发的应用
     - 激光雷达点云增强
     - 快速无损的场景纹理重建

3.2 R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator

Jiarong Lin** and **Fu Zhang

All our codes are now available at:

https://github.com/hku-mars/r3live



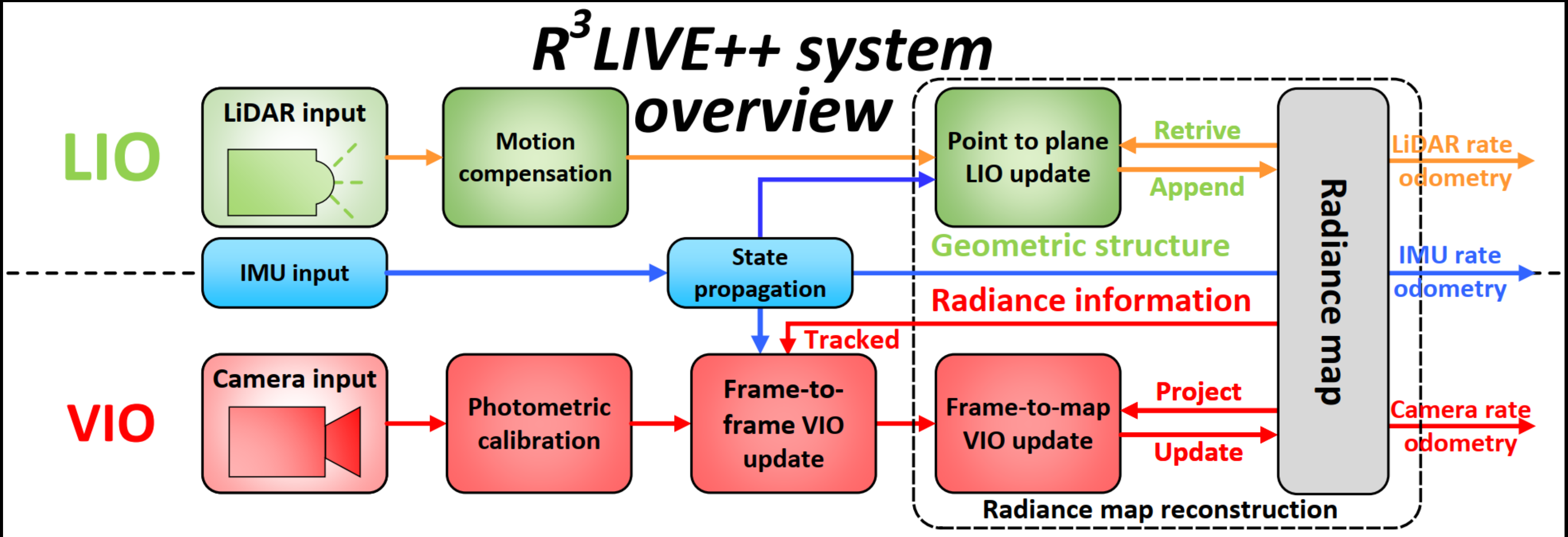
MARS LAB

香港大學

THE UNIVERSITY OF HONG KONG

3.2 多传感器（激光雷达-惯导-视觉）融合(R³LIVE++)

R³LIVE++: 基于激光雷达-惯导-视觉的实时辐射场地图重建



（R³LIVE++系统框图）

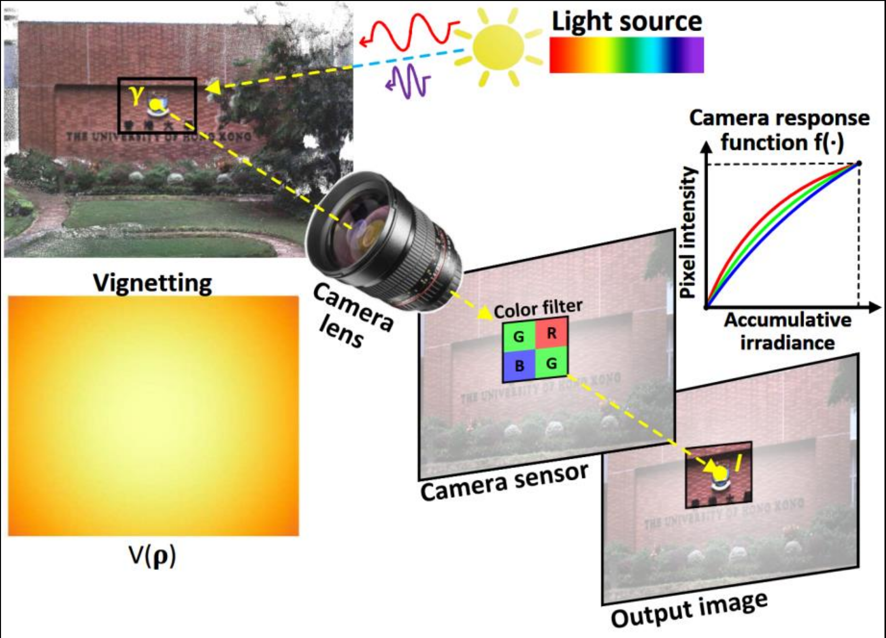
参考文献：

[1] Lin J, Zhang F. “R³LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package”

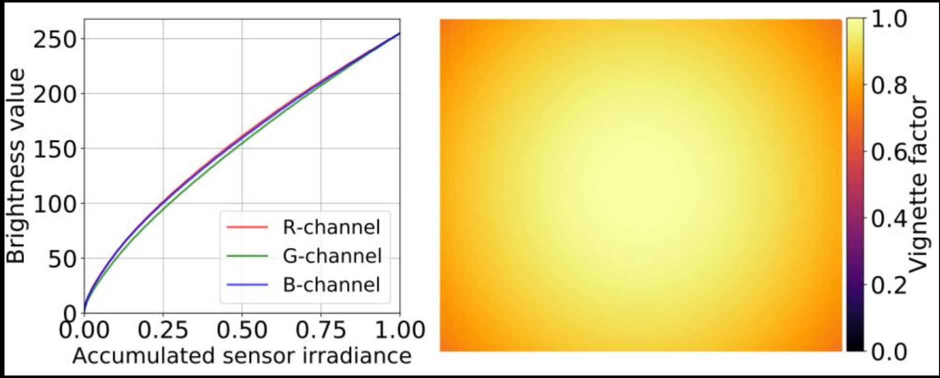
[2] Lin J, Zhang F. “R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator”

3.2 多传感器（激光雷达-惯导-视觉）融合(R^3LIVE++)

R^3LIVE++: 基于激光雷达-惯导-视觉的实时辐射场地图重建



(R³LIVE++光度学模型)



(彩色相机光度学标定)

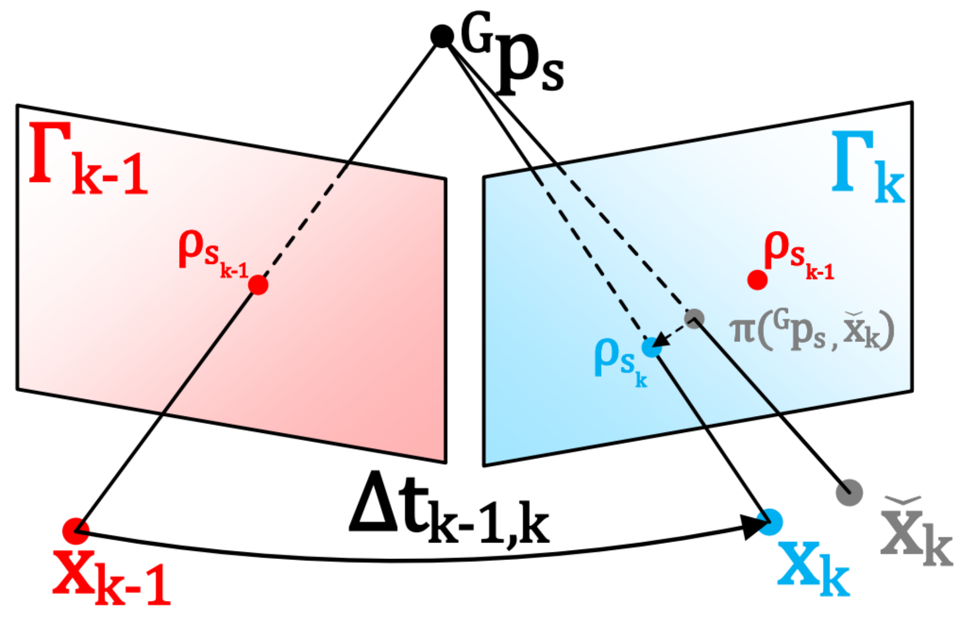
参考文献：

[1] Lin J, Zhang F. "R³LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package"

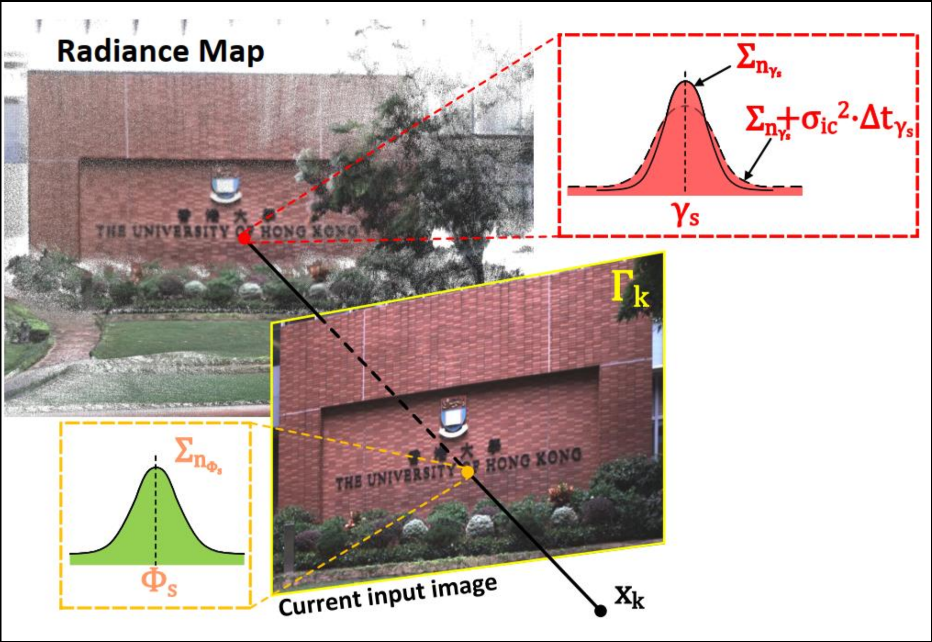
[2] Lin J, Zhang F. "R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator"

3.2 多传感器（激光雷达-惯导-视觉）融合(R^3LIVE++)

R^3LIVE++: 基于激光雷达-惯导-视觉的实时辐射场地图重建



（Frame-to-Frame VIO更新）



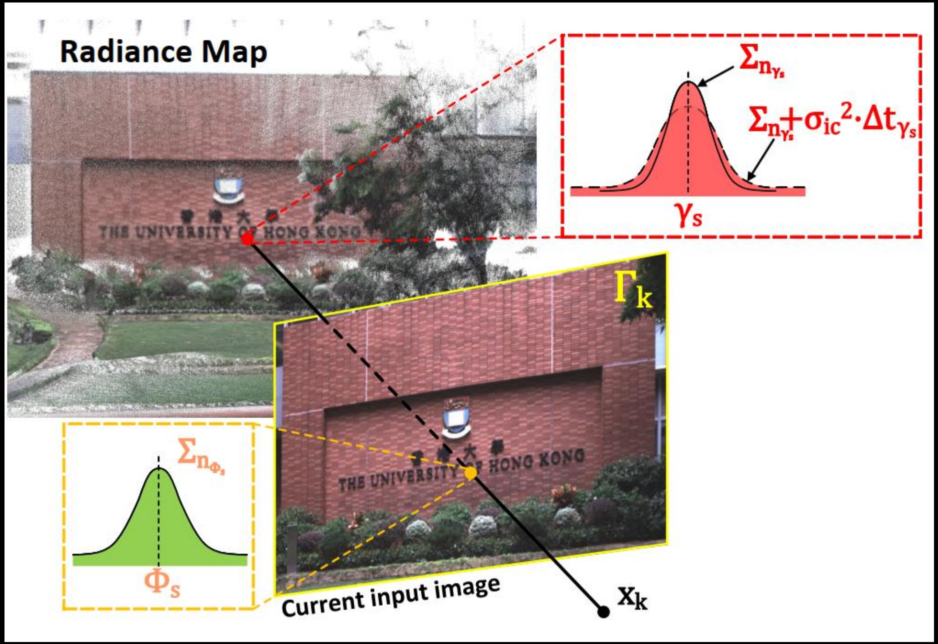
（Frame-to-Map VIO更新）

参考文献：

[1] Lin J, Zhang F. “R^3LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator”

3.2 多传感器（激光雷达-惯导-视觉）融合(R^3LIVE++)

R^3LIVE++: 基于激光雷达-惯导-视觉的实时辐射场地图重建



（Frame-to-Map VIO更新）

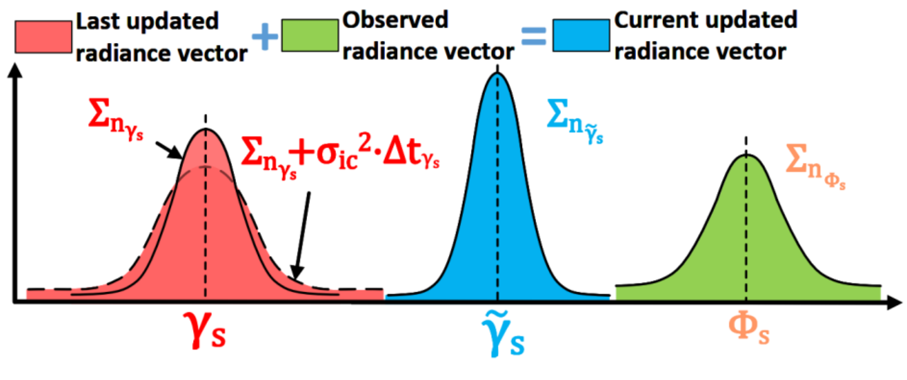


Fig. 6 – We update the radiance γ_s of a map point via Bayesian update.

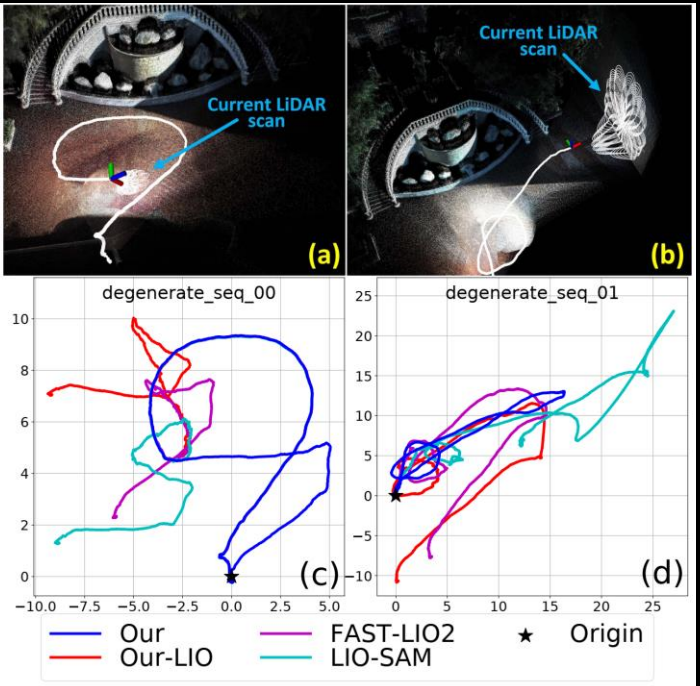
（地图点辐射值的更新）

参考文献：

[1] Lin J, Zhang F. "R^3LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator"

3.2 多传感器（激光雷达-惯导-视觉）融合(R³LIVE++)

R³LIVE++: 基于激光雷达-惯导-视觉的实时辐射场地图重建



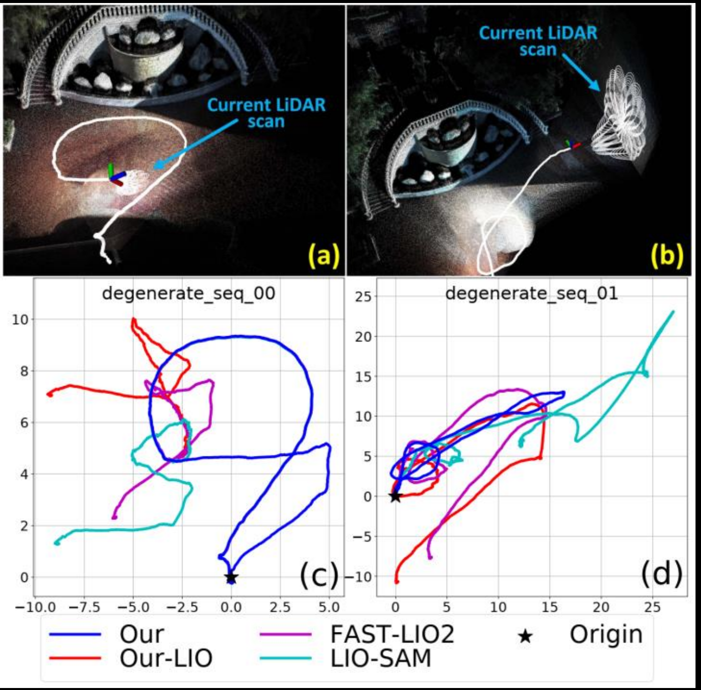
（抗激光雷达退化—测试1）

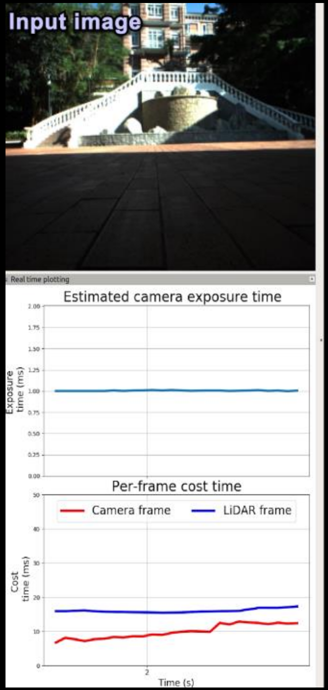
参考文献：

[1] Lin J, Zhang F. “R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator”

3.2 多传感器（激光雷达-惯导-视觉）融合(R³LIVE++)

R³LIVE++: 基于激光雷达-惯导-视觉的实时辐射场地图重建



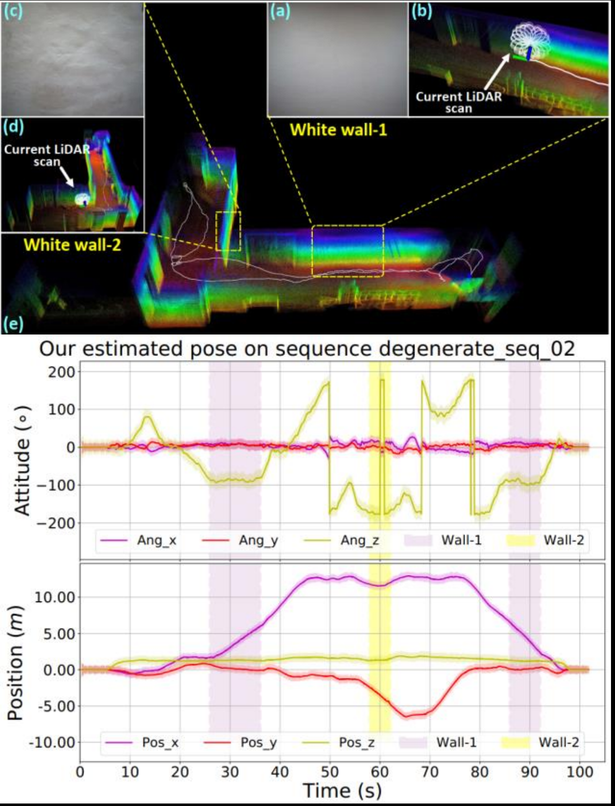


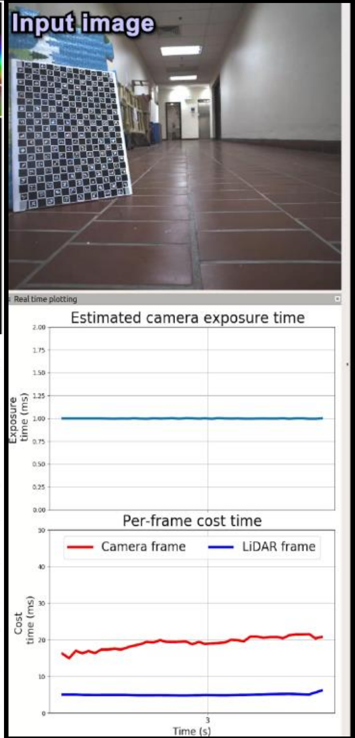
（抗激光雷达退化—测试2）

参考文献：

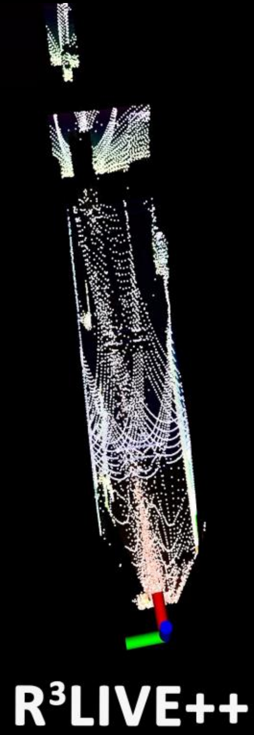
[1] Lin J, Zhang F. “R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator”

3.2 多传感器（激光雷达-惯导-视觉）融合($R^3LIVE++$)





Play X 10

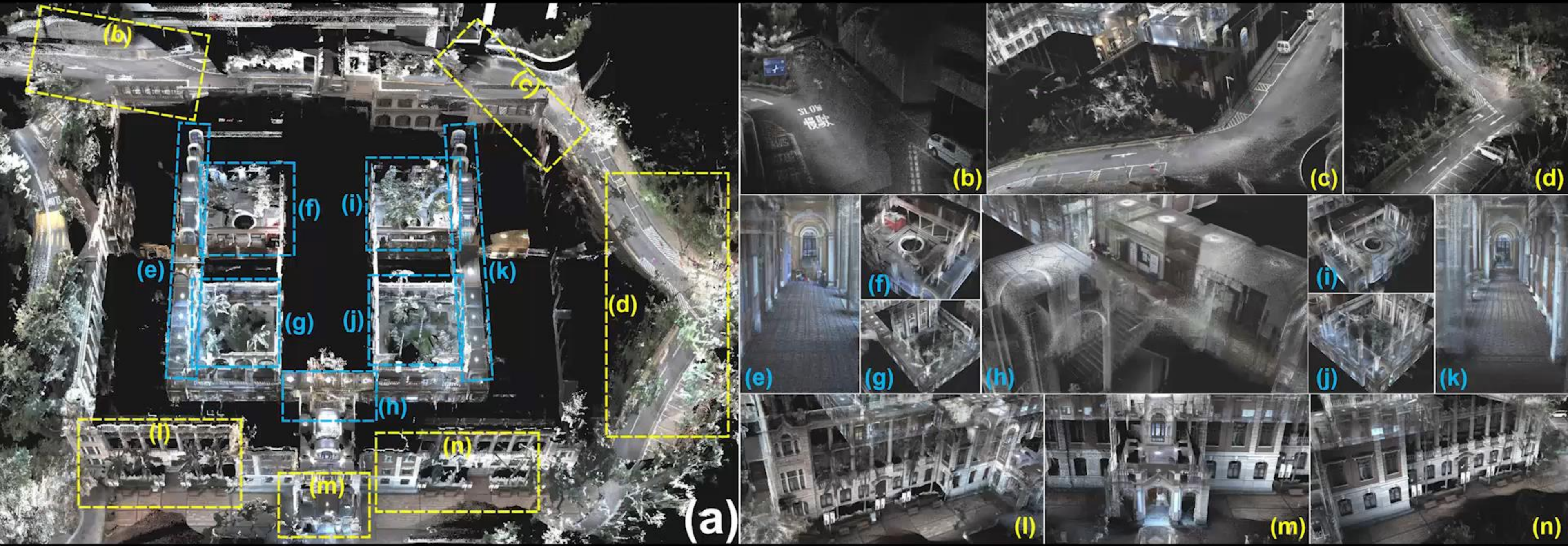


（激光雷达退化，同时弱视觉纹理场景测试）

参考文献：

[1] Lin J, Zhang F. “R³LIVE++: A Robust, Real-time, Radiance reconstruction package with a tightly-coupled LiDAR-Inertial-Visual state Estimator”

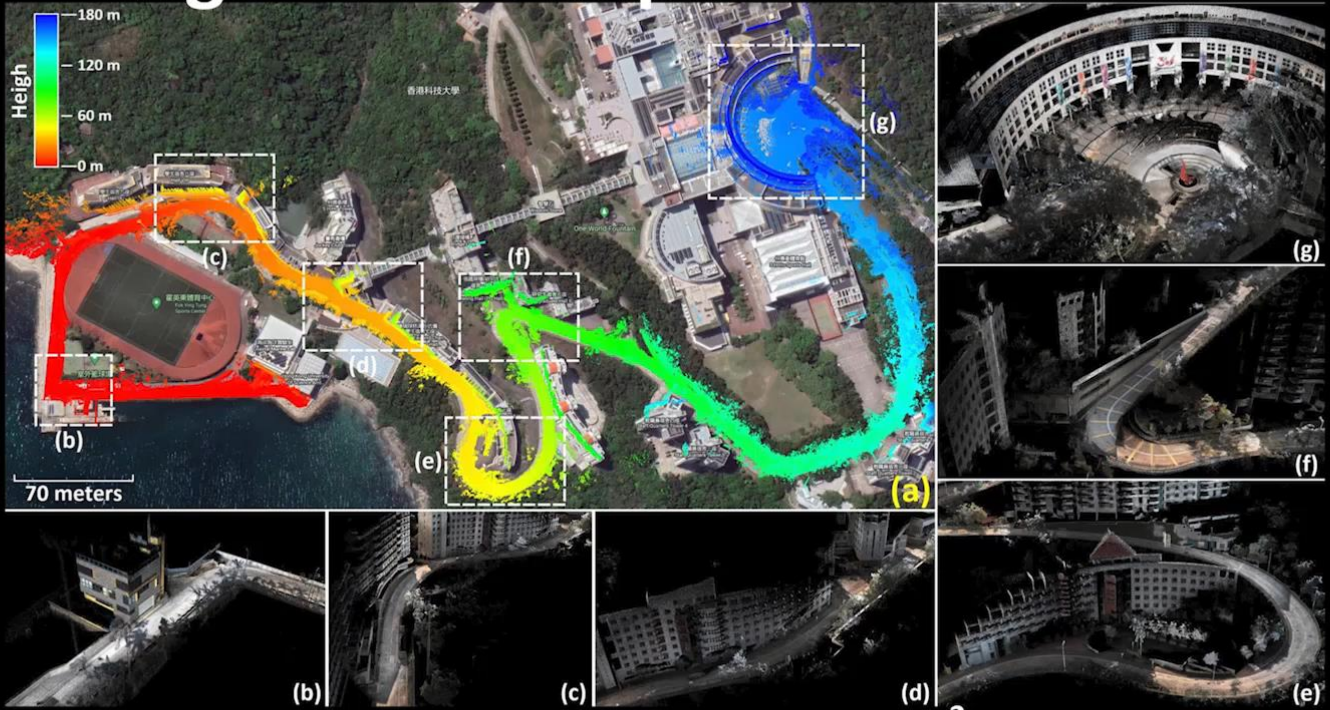
Demo-2: Online radiance map reconstruction in a large-scale indoor & outdoor scenario



(Sequence "hku_main_building" of R^3LIVE-dataset)

In this demo, we show the real-time process of R^3LIVE++ in reconstructing the large-scale, indoor-outdoor radiance map of HKU main building.

Demo-3: Online radiance map reconstruction in a large-scale campus environment



(Sequence “hkust_campus_seq_03” of R^3LIVE-dataset)

In this demo, we show the real-time process of R^3LIVE++ in reconstructing the large-scale HKUST campus, with the traveling length reaching 2.1 Km.

Application-1: Offline mesh reconstruction and texturing

While R^3LIVE++ reconstructs the radiance map in real-time, we also develop software utilities to mesh and texture the reconstructed map offline.

Application-3: We use the maps reconstructed by R^{3}LIVE++ for developing the video game for desktop PC

In this video game, the player is fighting against the dragon with shoting the rubber balls in the campus of HKUST.

Application-4: We use the maps reconstructed by R[3]LIVE++ to build the car and drone simulator with AirSim[1]

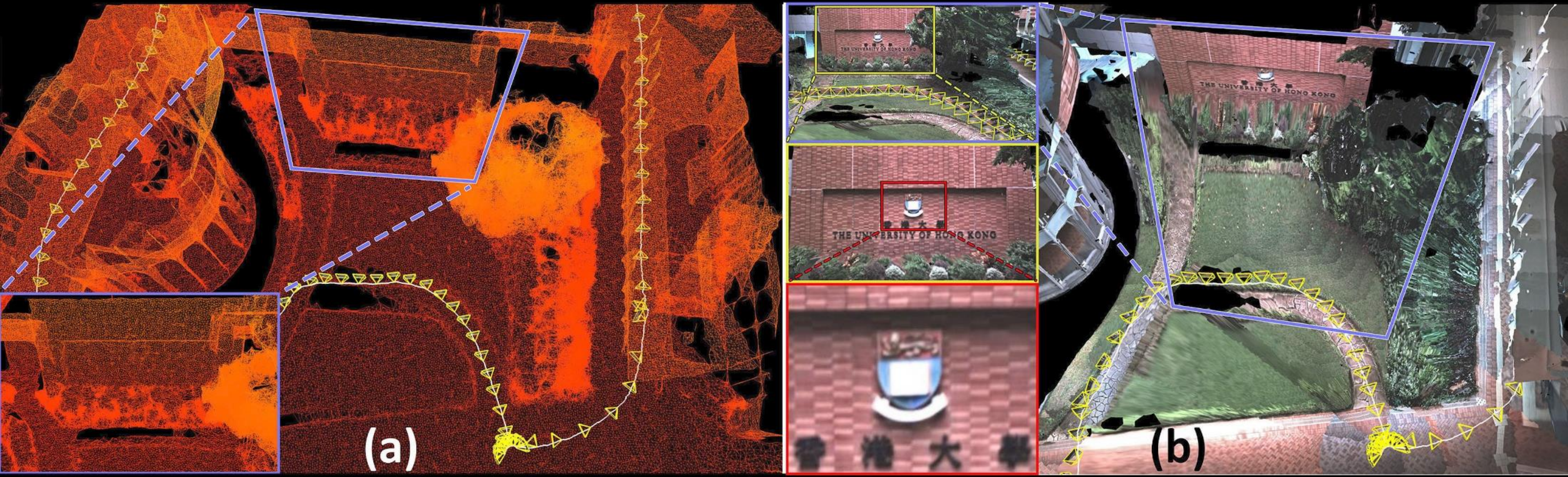
[1] AirSim: https://microsoft.github.io/AirSim

1. 传感器介绍
2. 激光雷达（-惯导）SLAM
   - 首个基于固态激光雷达的SLAM系统（**loam-livox**）
   - 激光雷达-惯导紧耦合的里程计（**FAST-LIO**）
3. 多传感器（激光雷达-惯导-视觉）融合
   - 首个开源的激光雷达-惯导-视觉多传感器紧耦合方案（**R²LIVE**）
   - 基于激光雷达-惯导-视觉的实时真彩地图重建（**R³LIVE**）
4. 激光雷达的实时定位和网格（mesh）重建系统 （**ImMesh**）
   - ImMesh的介绍和实验演示
   - 基于ImMesh开发的应用
     - 激光雷达点云增强
     - 快速无损的场景纹理重建

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

Jiarong Lin*, Chongjiang Yuan*, Yixi Cai, Haotian Li, Yuying Zou, Xiaoping Hong and Fu Zhang

https://github.com/hku-mars/ImMesh



MARS LAB 香港大學 THE UNIVERSITY OF HONG KONG 南方科技大學 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY ISEE

参考文献：

[1] Lin J, Yuan C, Cai Y, Li H, et al. “An Immediate LiDAR Localization and Meshing Framework”

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

相关工作(离线方法)—泊松表面重建

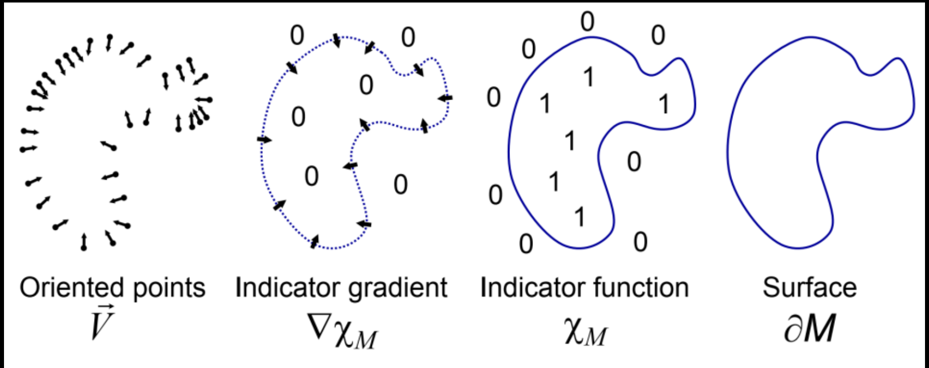
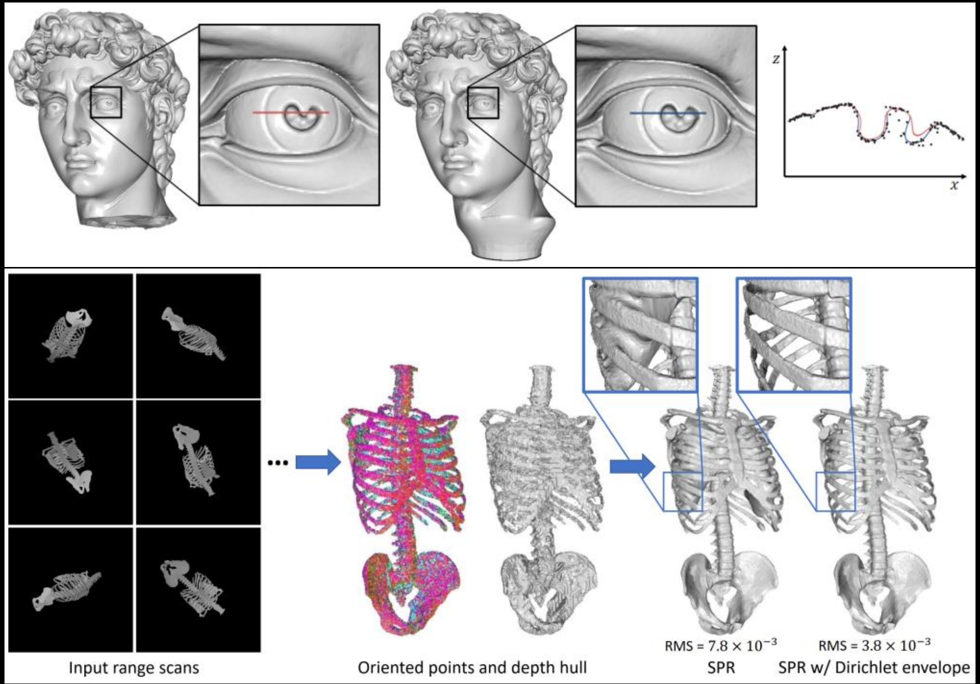


Figure 1: *Intuitive illustration of Poisson reconstruction in 2D.*

(泊松重建2D示意图)



(使用泊松重建从3D扫描点云中重建Mesh模型)

参考文献:

[1] Kazhdan, Michael, Matthew B, and Hugues H. "Poisson surface reconstruction."

[2] Kazhdan, M, and Hugues H. "Screened poisson surface reconstruction."

[3] Kazhdan M, Chuang M, Rusinkiewicz S, et al. "Poisson surface reconstruction with envelope constraints"

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

相关工作(离线方法)—Delaunay triangulation and graph cut(三角剖分+图分割)

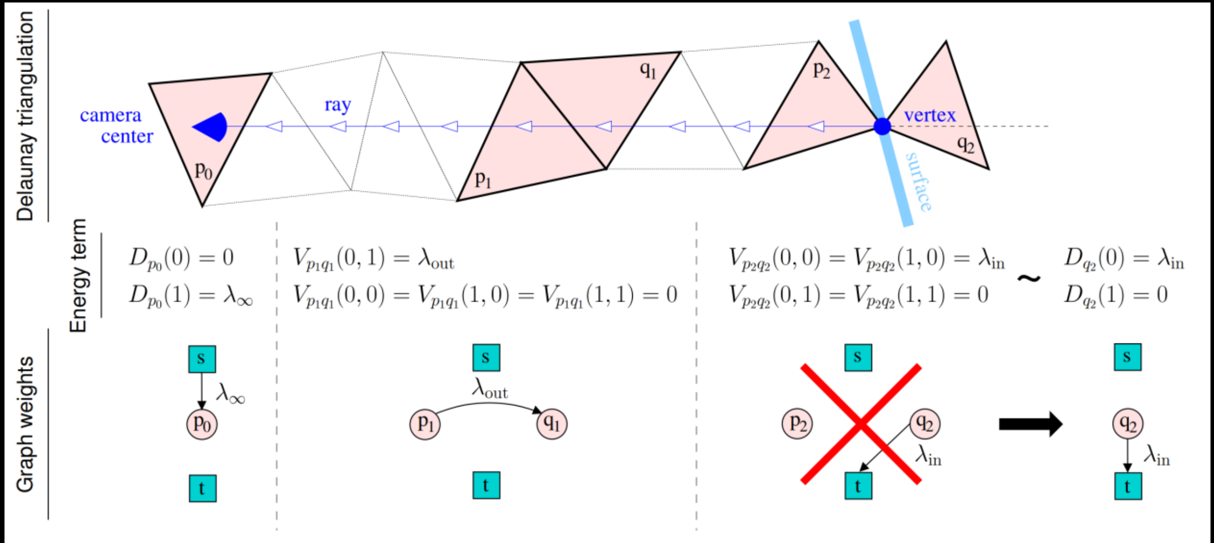
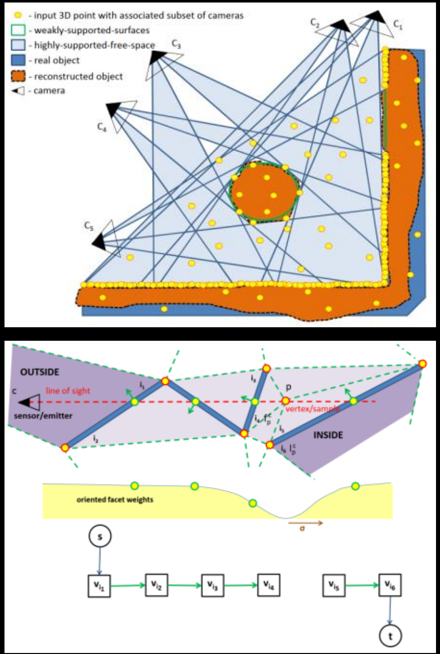


Figure 3. A ray emanating from a vertex to a camera center (and the putative surface), the corresponding visibility-related energy term that penalizes the number of intersections with the ray (the label 0 means *s* / “outside” and the label 1 means *t* / “inside”) and the edge weights of the crossed tetrahedra in the graph.



(Labatut P, et al.)　　(Jancosek M, et al)

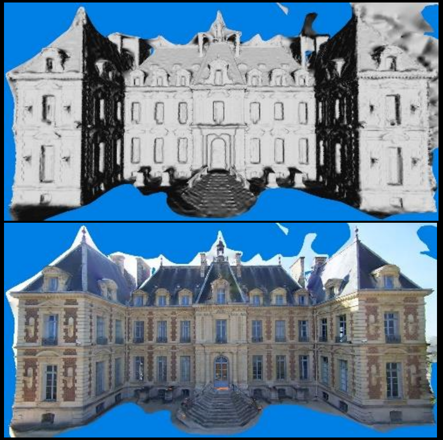
参考文献：

[1] Labatut P, Pons J P, Keriven R. “Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts

[2] Jancosek M, Pajdla T. “Multi-view reconstruction preserving weakly-supported surfaces”

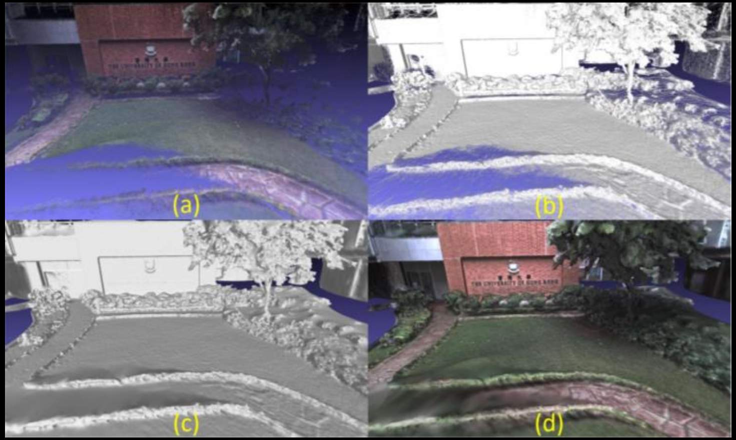
4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

相关工作(离线方法)—Delaunay triangulation and graph cut(三角剖分+图分割)



(openMVG--PMVS/CMVS)

(openMVS)



(R^3LIVE)

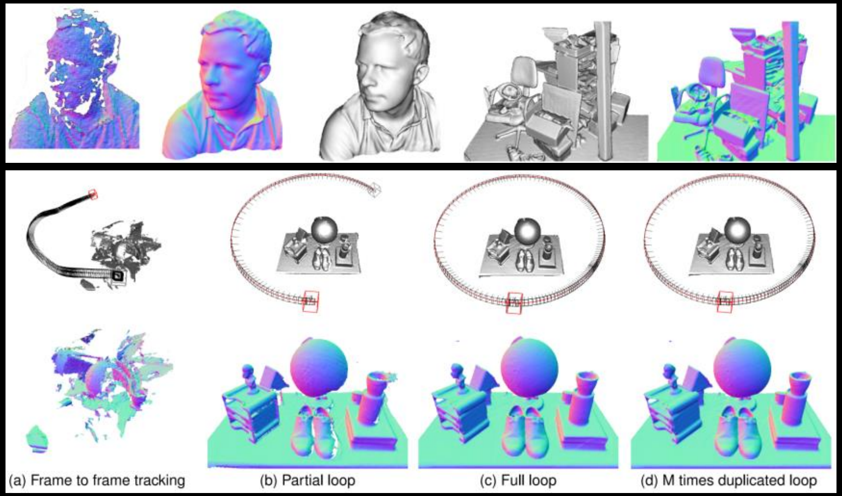
开源实现：

[1] openMVG： https://github.com/openMVG/openMVG

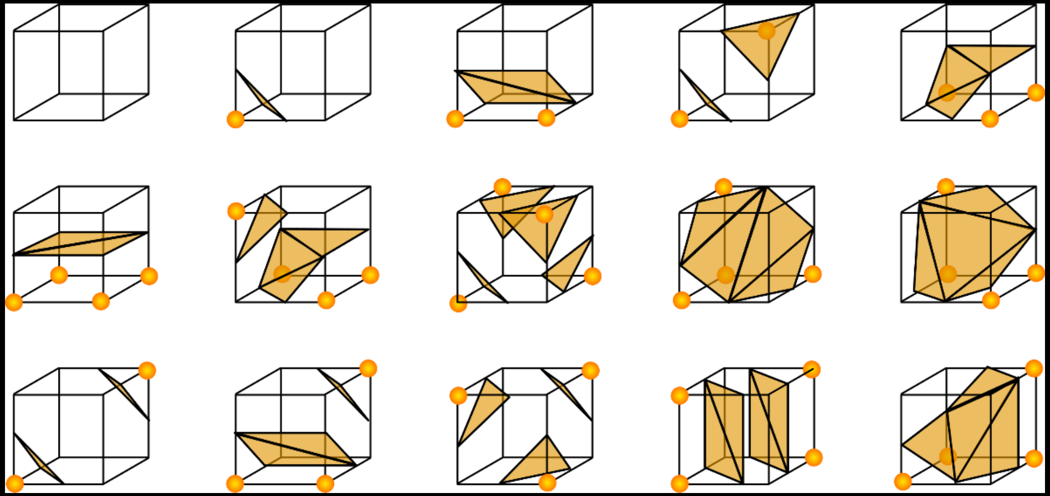
[2] openMVS： https://github.com/cdcseacave/openMVS

[3] R^3LIVE: https://github.com/hku-mars/r3live

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)
相关工作(在线方法)—TSDF+Marching cubes



(KinectFusion)

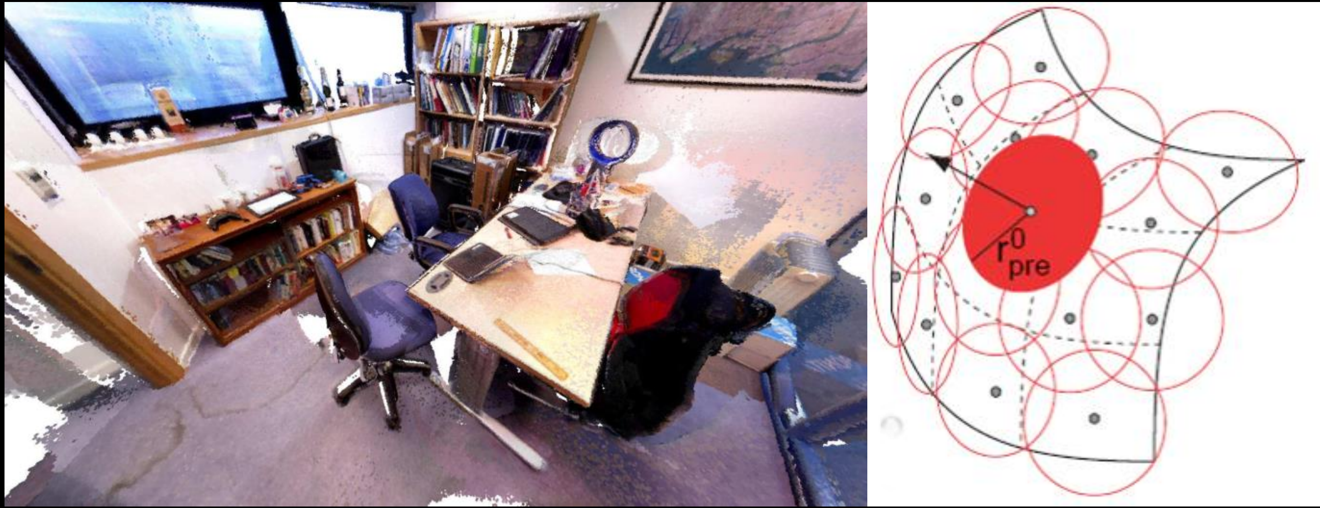


(Marching cubes)

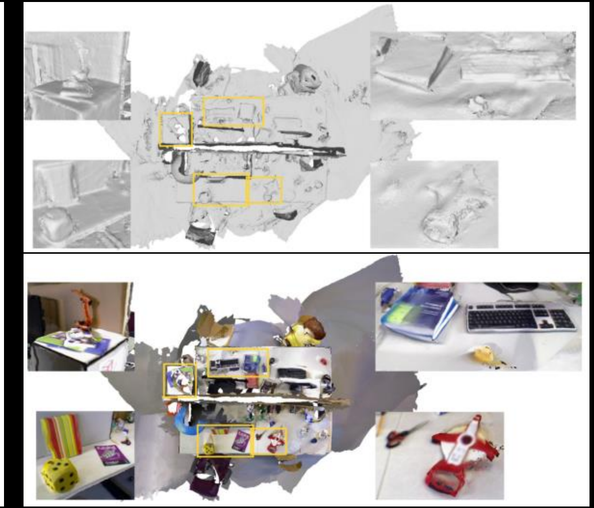
参考文献：
[1] Newcombe R A, Izadi S, Hilliges O, et al. “Kinectfusion: Real-time dense surface mapping and tracking”
[2] Lorensen W E, Cline H E. “Marching cubes: A high resolution 3D surface construction algorithm”

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

相关工作(在线方法)—Point- Surfels- based



(ElasticFusion)



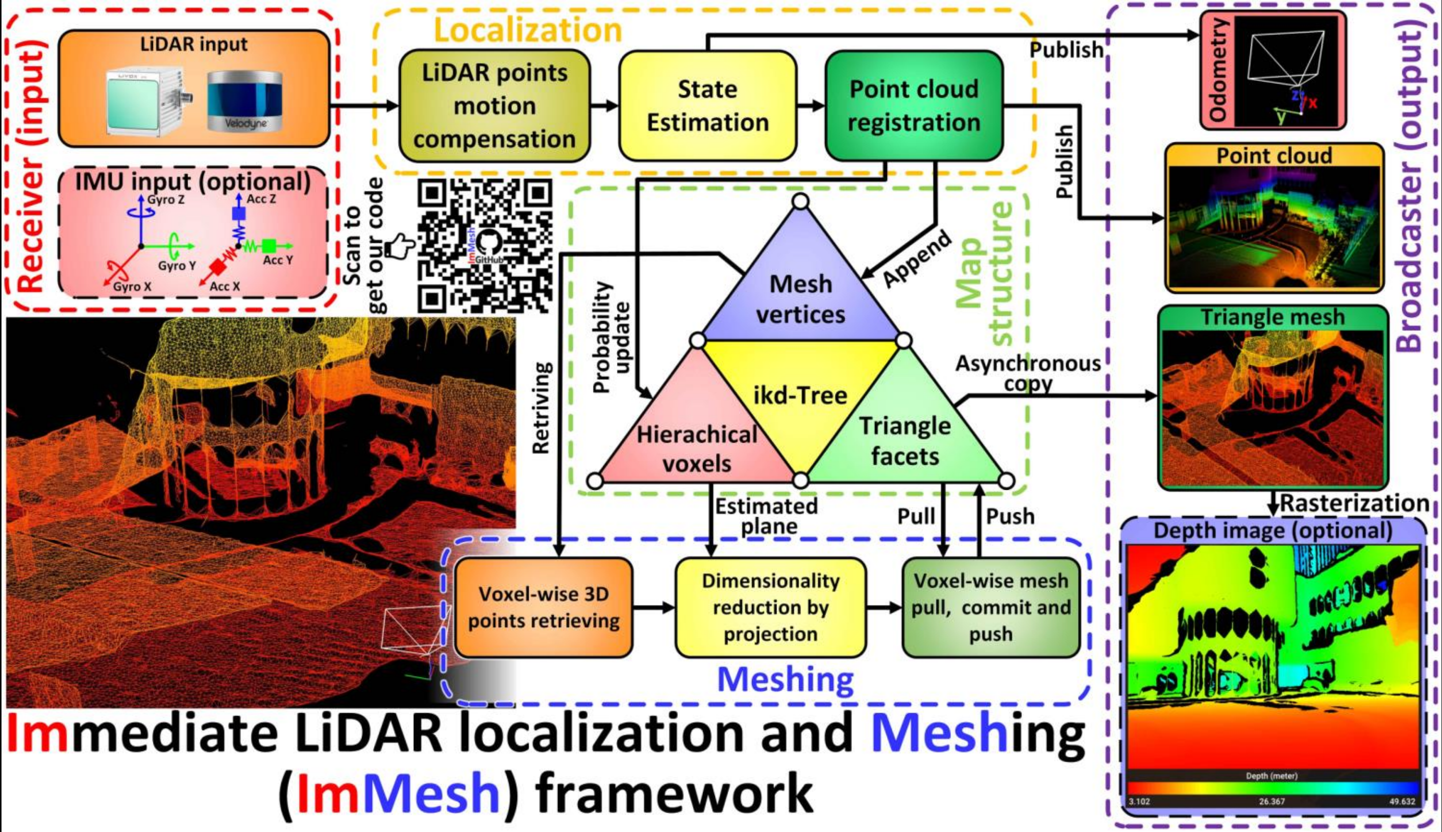
(SurfelMeshing)

参考文献：

[1] Whelan T, Leutenegger S, Salas-Moreno R, et al. “ElasticFusion: Dense SLAM without a pose graph”

[2] Schöps T, Sattler T, Pollefeys M. “Surfelmeshing: Online surfel-based mesh reconstruction”

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

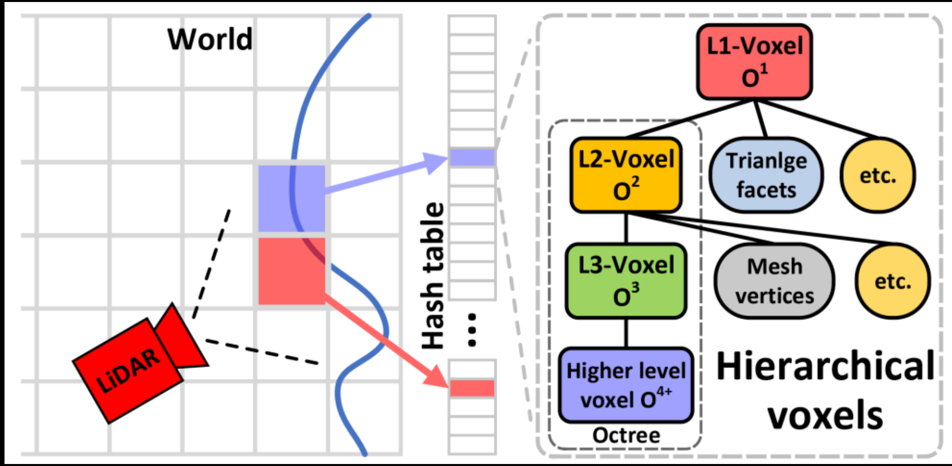


(ImMesh系统框图)

参考文献：

[1] Lin J, Yuan C, Cai Y, Li H, et al. “An Immediate LiDAR Localization and Meshing Framework”

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)



(分层体素)

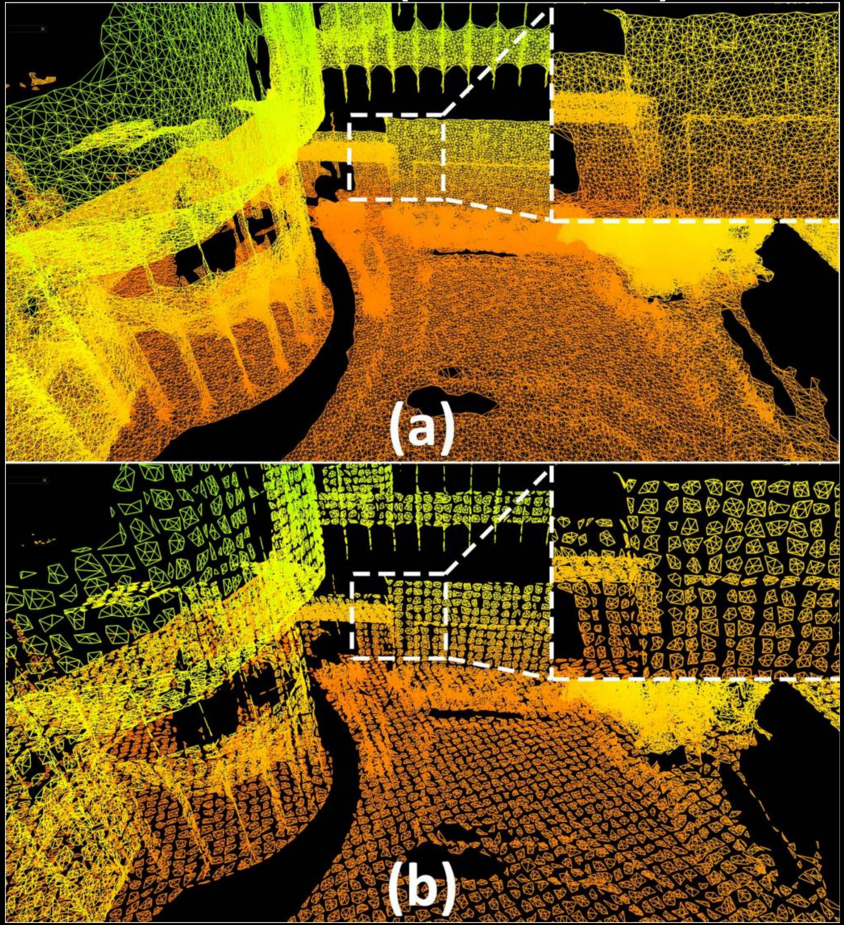
Algorithm 1: Voxel-wise vertex retrieval of $\mathbf{O}_i^2$

Input : The *activated* voxel $\mathbf{O}_i^2$
Output: The retrieved vertex set $\mathcal{V}_i$
Start : Copy all in-voxel pointer list to $\mathcal{V}_i^{\text{In}}$.
$\mathcal{V}_i = \mathcal{V}_i^{\text{In}}$.

foreach $\mathbf{V}_{i_j} \in \mathcal{V}_i^{\text{In}}$ **do**
  $\tilde{\mathcal{V}}_{i_j}$ = `RadiusSearch`($\mathbf{V}_{i_j}, d_r$)
  foreach $\mathbf{V} \in \tilde{\mathcal{V}}_{i_j}$ **do**
    if $\mathbf{V} \notin \mathcal{V}_i$ **then**
      $\mathcal{V}_i = \mathcal{V}_i \cup \mathbf{V}$

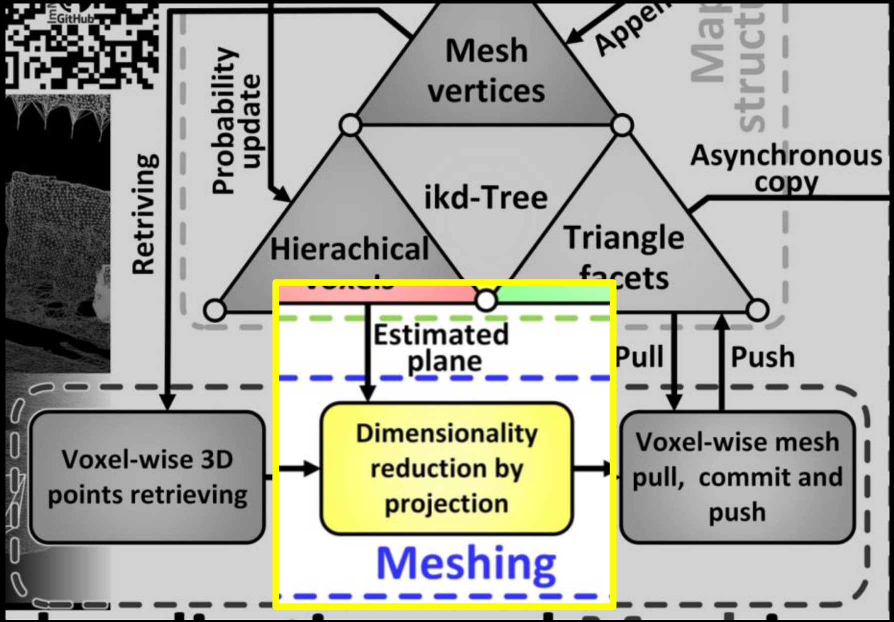
Return: The retrived vertex set $\mathcal{V}_i$ after dilation

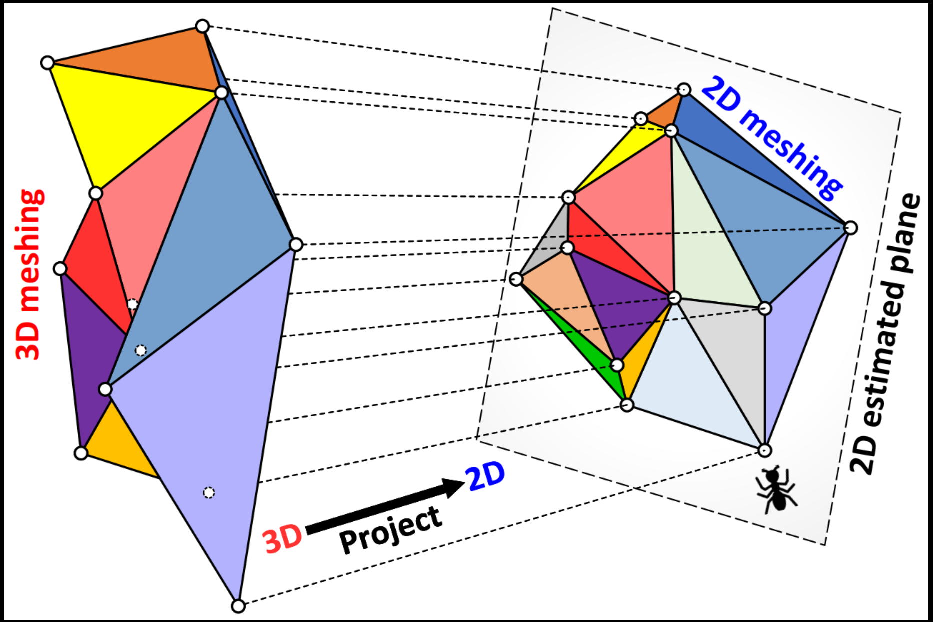
(体素内点的检索，膨胀算法)



(顶点集膨胀)

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)





(通过投影对Meshing问题降维处理)

参考文献:
[1] Lin J, Yuan C, Cai Y, Li H, et al. "An Immediate LiDAR Localization and Meshing Framework"

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

Algorithm 2: Voxel-wise mesh pull.

Input : The retrieved vertex set $\mathcal{V}_i$ from Algorithm 1
Output: The pulled triangle facets $\mathcal{T}_i^{\texttt{Pull}}$
Start : $\mathcal{T}_i^{\texttt{Pull}} = \{\texttt{null}\}$

foreach $\mathbf{V}_j \in \mathcal{V}_i$ **do**
  Get all vertices related triangle set $\mathcal{T}_{\mathbf{V}_j} = \texttt{Tri}(\mathbf{V}_j)$
  foreach $\mathbf{T}_k \in \mathcal{T}_{\mathbf{V}_j}$ **do**
    Get triangle vertex index $\{\alpha, \beta, \gamma\} = \texttt{Pts\_id}(\mathbf{T}_k)$
    if $(\mathbf{V}_\alpha \in \mathcal{V}_i)$ *and* $(\mathbf{V}_\beta \in \mathcal{V}_i)$ *and* $(\mathbf{V}_\gamma \in \mathcal{V}_i)$ **then**
      $\mathcal{T}_i^{\texttt{Pull}} = \mathcal{T}_i^{\texttt{Pull}} \cup \mathbf{T}_k$

Return: The pulled triangle facets $\mathcal{T}_i^{\texttt{Pull}}$

(Mesh Pull 拉取)

Algorithm 3: Voxel-wise mesh commit.

Input : The pulled triangle facets $\mathcal{T}_i^{\texttt{Pull}}$ from Algorithm 2
  The reconstructed triangle facets $\mathcal{T}_i$
Output: The triangle facets to be added $\mathcal{T}_i^{\texttt{Add}}$.
  The triangle facets to be erased $\mathcal{T}_i^{\texttt{Erase}}$.
Start : $\mathcal{T}_i^{\texttt{Add}} = \{\texttt{null}\}$, $\mathcal{T}_i^{\texttt{Erase}} = \{\texttt{null}\}$

foreach $\mathbf{T}_j \in \mathcal{T}_i$ **do**
  if $\mathbf{T}_j \notin \mathcal{T}_i^{\texttt{Pull}}$ **then**
    $\mathcal{T}_i^{\texttt{Add}} = \mathcal{T}_i^{\texttt{Add}} \cup \mathbf{T}_j$

foreach $\mathbf{T}_j \in \mathcal{T}_i^{\texttt{Pull}}$ **do**
  if $\mathbf{T}_j \notin \mathcal{T}_i$ **then**
    $\mathcal{T}_i^{\texttt{Erase}} = \mathcal{T}_i^{\texttt{Erase}} \cup \mathbf{T}_j$

Return: The triangle facets to be added $\mathcal{T}_i^{\texttt{Add}}$ and erased $\mathcal{T}_i^{\texttt{Erase}}$.

(Mesh commit 提交)

Algorithm 4: Voxel-wise mesh push.

Input : The triangle facets that need to erased $\mathcal{T}_i^{\texttt{Erase}}$.
  The triangle facets that need to added $\mathcal{T}_i^{\texttt{Add}}$.

Function `Add_triangle`$(\mathbf{T}_j)$:
  Get point indices $\{\alpha, \beta, \gamma\} = \texttt{Id}(\mathbf{T}_j)$
  Construct triangle $\mathbf{T}_j^G = \texttt{Tri}(\alpha, \beta, \gamma)$ in global map.
  Calculate the center of $\mathbf{T}_j^G$:
    $\texttt{Center}(\mathbf{T}_j^G) = (\mathbf{V}_\alpha + \mathbf{V}_\beta + \mathbf{V}_\gamma)/3$
  Find the $L1$-Voxel $\mathbf{V}^1$ that $\texttt{Center}(\mathbf{T}_j^G)$ located in:
    $\mathbf{V}^1 = \texttt{Get\_L1\_voxel}(\texttt{Hash}(\texttt{Center}(\mathbf{T}_j^G)))$
  Set the status flag of $\mathbf{V}^1$ to *Sync-required* (Section IV-D2).
  Add $\texttt{Ptr}(\mathbf{T}_j^G)$ to triangle list of $L1$-Voxel $\mathbf{V}^1$.
  Add $\texttt{Ptr}(\mathbf{T}_j^G)$ to triangle list of points $\mathbf{V}_\alpha$, $\mathbf{V}_\beta$, $\mathbf{V}_\gamma$.

Function `Erase_triangle`$(\mathbf{T}_j)$:
  Get point indices $\{\alpha, \beta, \gamma\} = \texttt{Id}(\mathbf{T}_j)$
  Remove $\texttt{Ptr}(\mathbf{T}_j^G)$ in triangle list of points $\mathbf{V}_\alpha$, $\mathbf{V}_\beta$, $\mathbf{V}_\gamma$.
  Find the $L1$-Voxel $\mathbf{V}^1$ with $\texttt{Center}(\mathbf{T}_j^G)$ via (9):
    $\mathbf{V}^1 = \texttt{Get\_L1\_voxel}(\texttt{Hash}(\texttt{Center}(\mathbf{T}_j^G)))$
  Set the status flag of $\mathbf{V}^1$ to *Sync-required* (Section IV-D2).
  Remove $\texttt{Ptr}(\mathbf{T}_j^G)$ from triangle list of $L1$-Voxel $\mathbf{V}^1$.
  Delete triangle $\mathbf{T}_j^G$ from memory.

foreach $\mathbf{T}_j \in \mathcal{T}_i^{\texttt{Add}}$ **do**
  `Add_triangle`$(\mathbf{T}_j)$

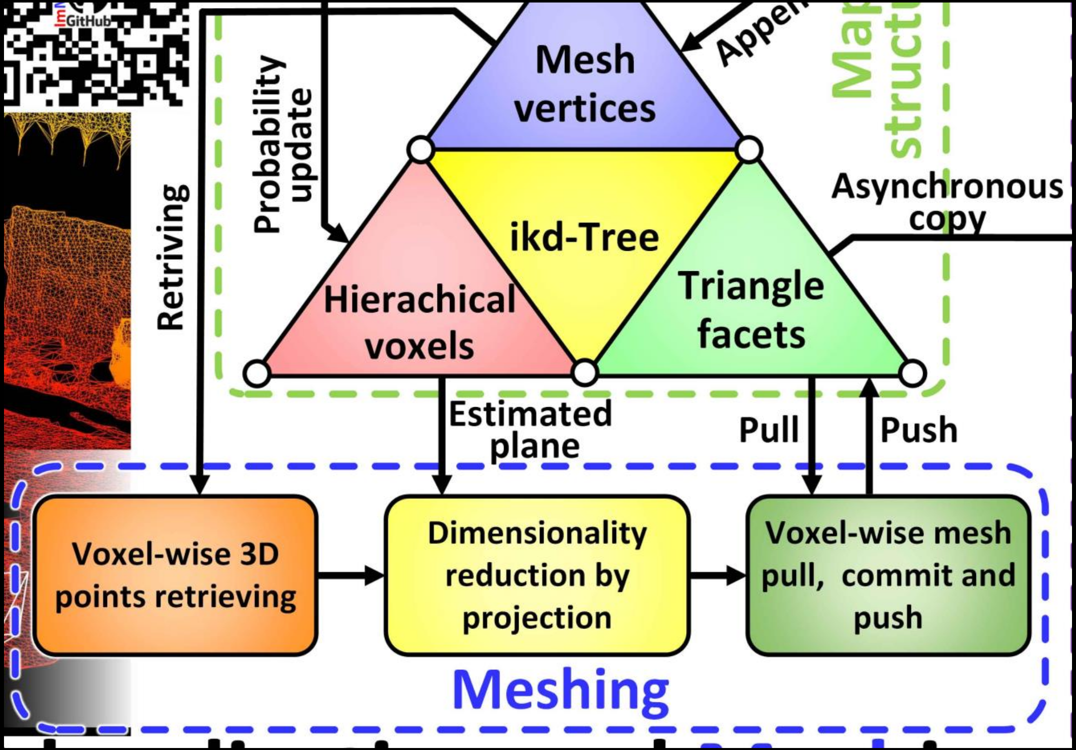
foreach $\mathbf{T}_j \in \mathcal{T}_i^{\texttt{Erase}}$ **do**
  `Erase_triangle`$(\mathbf{T}_j)$

(Mesh Push 推送)

参考文献：

[1] Lin J, Yuan C, Cai Y, Li H, et al. “An Immediate LiDAR Localization and Meshing Framework”

4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)



(Meshing模块)

Algorithm 5: The full meshing process of each update of LiDAR scan

Input : The set of $L2$-Voxels $\mathcal{V}^2 = \{\mathbf{O}_1^2, \mathbf{O}_2^2, ..., \mathbf{O}_m^2\}$ that *activated* in Section V-C

Start : The triangle facets that need to added $\mathcal{T}^{\text{Add}} = \{\text{null}\}$, and to be erased in this update $\mathcal{T}^{\text{Erase}} = \{\text{null}\}$.

1 **foreach** $\mathbf{O}_i^2 \in \mathcal{V}^2$ **do in parallel**
2 Retrieve vertices $\mathcal{V}_i$ with Algorithm 1.
3 Reconstruct the triangle facets $\mathcal{T}_i$ with $\mathcal{V}_i$ (Section VI-D2),
4 Performing voxel-wise mesh pull (Algorithm 2) to get $\mathcal{T}_i^{\text{Pull}}$. ▷ // Mesh *pull*
5 Performing voxel-wise mesh commit (Algorithm 3) to get the triangle facets that need to be added $\mathcal{T}_i^{\text{Add}}$ and erased $\mathcal{T}_i^{\text{Erase}}$. ▷ // Mesh *commit*
6 $\mathcal{T}^{\text{Add}} = \mathcal{T}^{\text{Add}} \bigcup \mathcal{T}_i^{\text{Add}}$, $\mathcal{T}^{\text{Erase}} = \mathcal{T}^{\text{Erase}} \bigcup \mathcal{T}_i^{\text{Erase}}$

/* === Mesh *push* start === */
7 **foreach** $\mathbf{T}_j \in \mathcal{T}^{\text{Add}}$ **do**
8 Add_triangle($\mathbf{T}_j$) ▷ // In Algorithm 4
9 **foreach** $\mathbf{T}_j \in \mathcal{T}^{\text{Erase}}$ **do**
10 Erase_triangle($\mathbf{T}_j$) ▷ // In Algorithm 4
/* === Mesh *push* end === */
11 **foreach** $\mathbf{O}_i^2 \in \mathcal{V}^2$ **do**
12 Reset status of $\mathbf{O}_i^2$ as *deactived*.

/* *Remark 1*: Line 1~6 are done in parallel for better real-time performance (as mentioned in Section VI-F). */
/* *Remark 2*: The mesh push step Line 7~10 is different with the voxel-wise operations in Algorithm 4. The $\mathcal{T}^{\text{Add}}$ and $\mathcal{T}^{\text{Erase}}$ are processed after the parallelism to avoid possible conflicts when operating the same data (i.e., triangle facets in our *mapping* module) (Line 1~6). */

(增量式meshing的全算法)

参考文献：

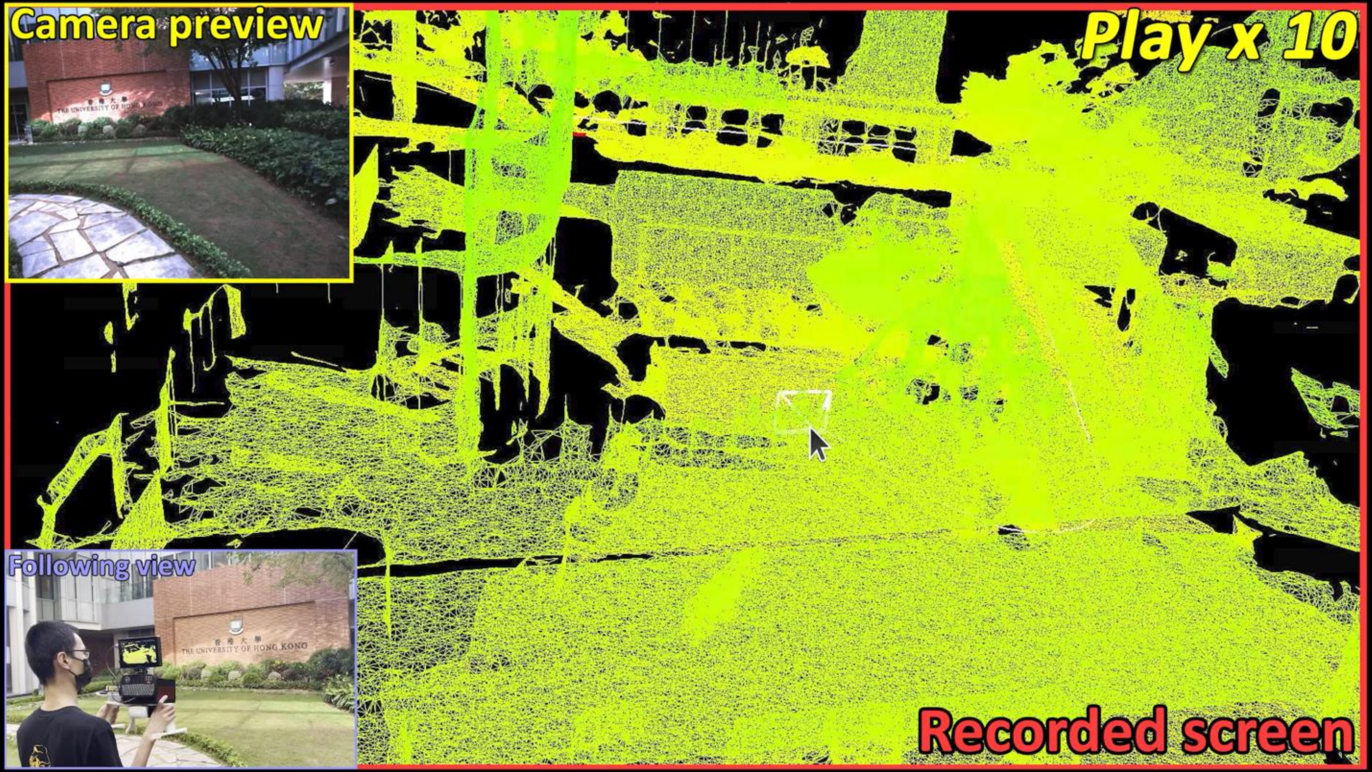
[1] Lin J, Yuan C, Cai Y, Li H, et al. “An Immediate LiDAR Localization and Meshing Framework”

Experiment-1: ImMesh for immediate mesh reconstruction (Trial #1)

(a) show our handheld device for data collection and online mesh reconstruction. In this experiment, we present three time-aligned views of different sources including: a screen-recorded view (in red), a camera preview (in yellow), and a third-person view (in blue), as shown in (b).

Experiment-1: ImMesh for immediate mesh reconstruction (Trial #2)

In this experiment, we present three time-aligned views of different sources including: a screen-recorded view (in red), a camera preview (in yellow), and a third-person view (in blue).



4.1 基于激光雷达的即时定位和网格重建系统 (ImMesh)

TABLE I: The specifications of LiDARs in four datasets

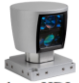
Dataset	Kitti	NCLT	NTU VIRAL	R^3LIVE
LiDAR	 Velodyne HDL-64E	 Velodyne HDL-32E	 Ouster OS1-16 Gen1	 Livox Avia
Scanning mechanism	Mechanical, spinning 64-line	Mechanical, spinning 32-line	Mechanical, spinning 16-line	Solid-state, Risley's prism
Field of View (Horizontal° × Vertical°)	$360.0^\circ \times 26.8^\circ$	$360.0^\circ \times 41.3^\circ$	$360.0^\circ \times 33.2^\circ$	$70.4^\circ \times 77.2^\circ$
Points per second[1]	1,333,312	695,000	327,680	240,000
Price	$ 75,000	$ 8,800	$ 3,500	$ 1,599

TABLE IV: The average/maximum time of *meshing* and *localization* module for processing each LiDAR scan in four datasets.

	Kitti mean/max	NCLT mean/max	NTU VIRAL mean/max	R^3LIVE mean/max
Meshing (ms)	31.3 / 34.5	24.2 / 25.4	9.8 / 17.2	25.3 / 33.6
Localization (ms)	42.2 / 56.9	22.3 / 26.8	11.9 / 17.2	16.6 / 21.0

TABLE II: This table shows the detailed information (e.g., length, duration, scenarios) of each testing sequence, the time consumption of ImMesh in processing a LiDAR scan, and the number of vertices and facets of each reconstructed mesh in Experiment-2.

Sequece	Traveling length (m)	Durations (s)	LiDAR frames	Meshing mean/Std (ms)	Localization mean/Std (ms)	Number of vertices (k)	Number of facets(k)	Scenarios
Kitti_00	3,724.2	456	4,541	32.1 / 12.0	49.0 / 11.7	3,339.4	7,692.7	Urban city
Kitti_01	2,453.2	146	1,101	34.5 / 10.5	51.1 / 18.5	2,033.0	4,046.8	High way
Kitti_02	5,058.9	509	4,661	33.5 / 7.0	36.2 / 9.5	4,390.3	10,028.1	Residential
Kitti_03	560.9	88	801	28 / 7.1	49.0 / 12.2	730.0	1,550.8	Countryside; Road
Kitti_04	393.6	27	271	30.1 / 9.4	42.4 / 12.9	411.7	850.6	Urban city; Road
Kitti_05	2,205.6	303	2,761	29.6 / 8.2	38.7 / 11.5	2,167.4	4,950.3	Residential
Kitti_06	1,232.9	123	1,101	23.1 / 5.6	56.9 / 9.7	886.1	1,889.4	Urban city
Kitti_07	2,453.2	114	1,101	20.7 / 7.4	31.3 / 8.6	764.4	1,710.5	Urban city
Kitti_08	3,222.8	441	4,071	32.4 / 7.8	45.7 / 17.7	3,559.1	7,936.3	Urban city
Kitti_09	1,705.1	171	1,591	34.5 / 7.5	43.1 / 19.2	1,827.4	4,127.5	Countryside; Road
Kitti_10	919.5	132	1,201	23.4 / 6.9	30.9 / 11.9	939.6	2,096.5	Residential
NCLT 2012-01-15	7,499.8	6739	66,889	26.3 / 14.1	21.3 / 9.8	9,659.7	26,608.3	Campus; Indoor
NCLT 2012-04-29	3,183.1	2598	25,819	25.4 / 13.9	19.1 / 5.4	4,820.9	13,483.9	Campus
NCLT 2012-06-15	4,085.9	3310	32,954	24.5 / 14.4	22.3 / 7.7	6,361.0	17,473.5	Campus
NCLT 2013-01-10	1,132.3	1024	10,212	20.2 / 12.5	19.3 / 6.5	2,020.6	5,495.8	Campus
NCLT 2013-04-05	4,523.6	4167	41,651	20.6 / 13.8	26.8 / 11.7	9,582.3	23,982.4	Campus
NTU VIRAL eee_01	265.3	398	3,987	11.2 / 6.7	14.5 / 3.4	597.6	1,380.3	Aerial; Outdoor
NTU VIRAL nya_01	200.6	396	3,949	9.4 / 5.3	10.2 / 1.7	536.8	1,247.6	Aerial; Indoor
NTU VIRAL rtp_01	449.6	482	4,615	12.1 / 8.5	10.9 / 2.6	719.2	2,030.5	Aerial; Outdoor
NTU VIRAL sbs_01	222.1	354	3,542	11.4 / 8.0	17.2 / 3.2	472.5	1,150.4	Aerial; Outdoor
NTU VIRAL tnp_01	319.4	583	5,795	6.3 / 3.7	8.8 / 1.2	155.5	414.0	Aerial; Indoor
R^3LIVE hku_campus_00	190.6	202	2,022	12.0 / 7.3	11.5 / 3.2	587.1	1,236.9	Campus
R^3LIVE hku_campus_01	374.6	304	3,043	20.4 / 12.6	17.2 / 6.9	1,323.4	2,862.9	Campus
R^3LIVE hku_campus_02	354.3	323	3,236	13.5 / 6.4	11.9 / 2.8	867.9	1,913.6	Campus
R^3LIVE hku_campus_03	181.2	173	1,737	12.2 / 5.7	11.3 / 2.9	550.0	1,130.6	Campus
R^3LIVE hku_main_building	1,036.9	1170	11,703	16.9 / 14.3	12.5 / 8.0	3,031.2	6,803.6	Indoor; Outdoor
R^3LIVE hku_park_00	247.3	228	2,285	30.1 / 15.9	12.6 / 3.7	919.5	2,380.2	Cluttered field
R^3LIVE hku_park_01	401.8	351	3,520	31.5 / 12.2	12.6 / 3.9	1,673.0	3,964.8	Cluttered field
R^3LIVE hkust_campus_00	1,317.2	1073	10,732	26.0 / 12.8	18.0 / 7.6	4,916.7	11,246.8	Campus
R^3LIVE hkust_campus_01	1,524.3	1162	11,629	27.1 / 13.9	16.8 / 6.7	5,353.1	12,638.1	Campus
R^3LIVE hkust_campus_02	2,112.2	1618	4,787	26.7 / 14.5	20.3 / 6.1	1,991.6	4,653.5	Campus
R^3LIVE hkust_campus_03	503.8	478	16,181	33.6 / 13.3	21.0 / 5.3	7,673.8	18,247.3	Campus

(在四个公开数据集上的测试)

参考文献：

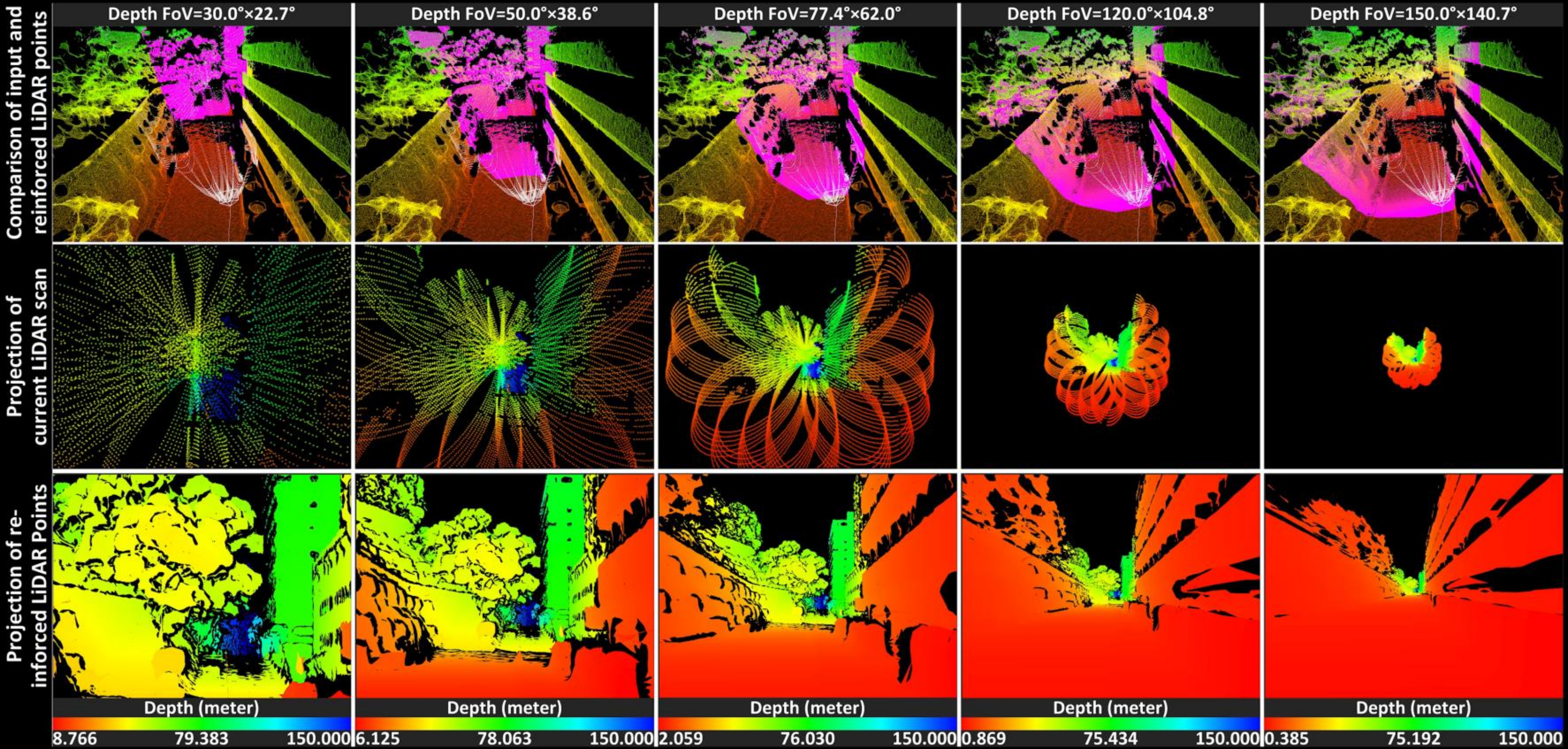
[1] Lin J, Yuan C, Cai Y, Li H, et al. "An Immediate LiDAR Localization and Meshing Framework"

Experiment-2: Evaluation of ImMesh on Kitti dataset

In this experiment, we extensively tested ImMesh with public datasets. This video show the online mesh reconstruction process of ImMesh that tested on Kitti_00 sequence.

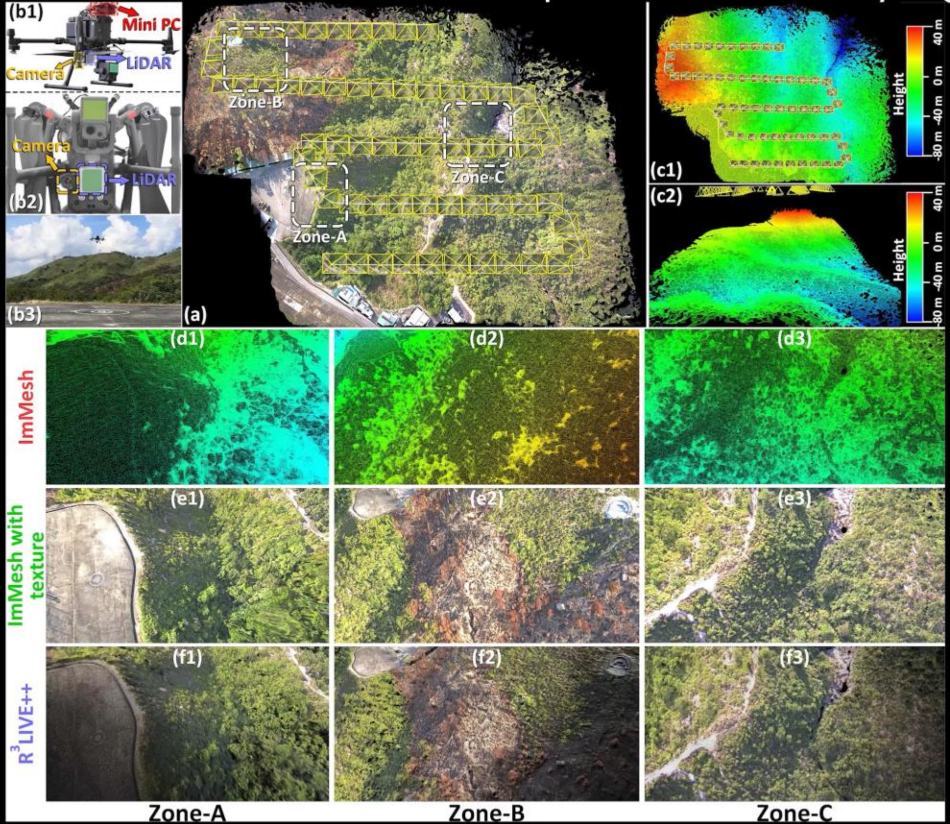
Application-1: ImMesh for LiDAR point cloud reinforement

In this application, we show how ImMesh can be applied for LiDAR point cloud reinforcement, which can output the reinforced points in regular pattern, and with higher density and wider FoV compared to raw LiDAR scan.



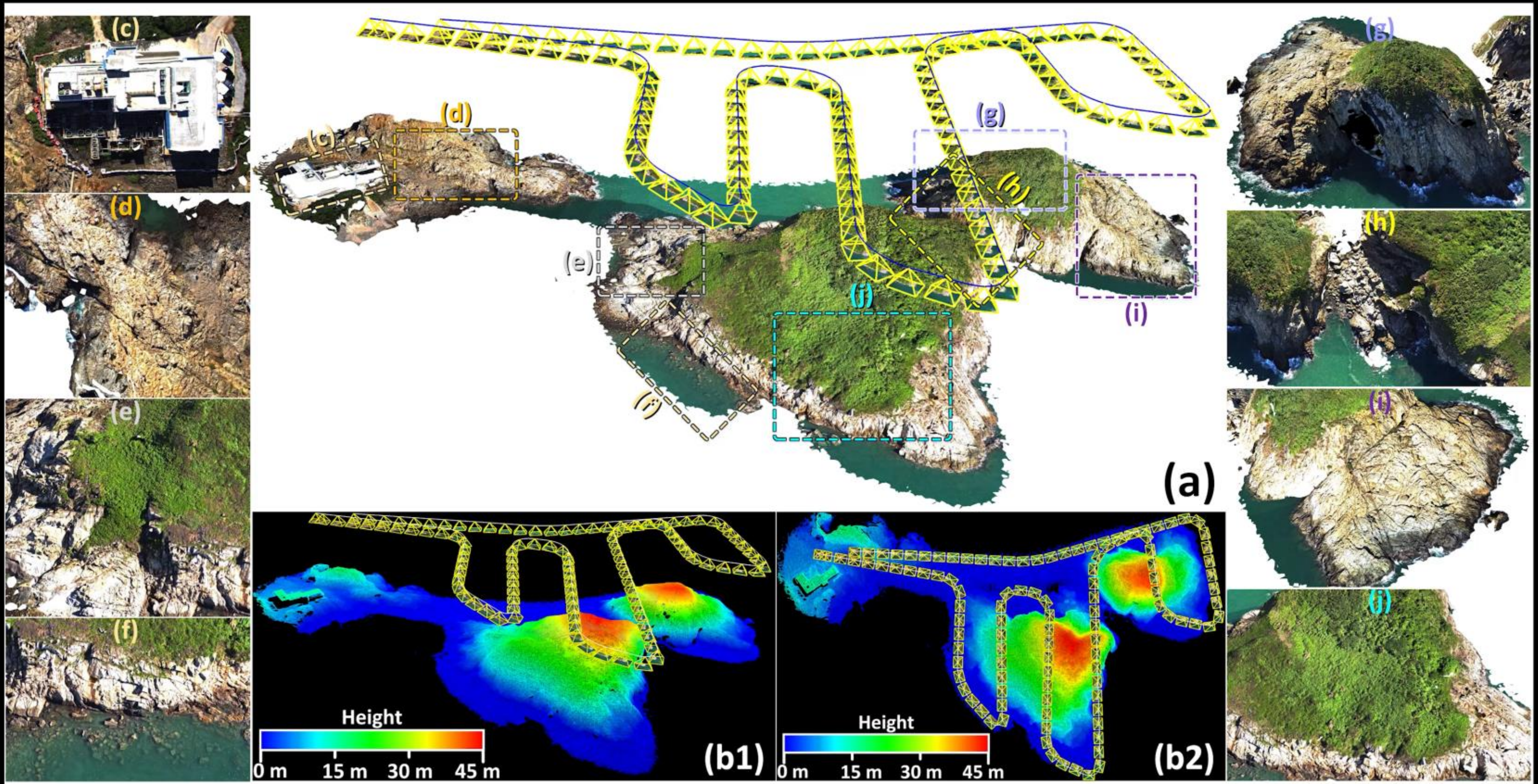
Application-2: ImMesh for rapid, lossless texture reconstruction (Trail #1)

In this application, we show how ImMesh can be applied in applications of lossless texture reconstruction for rapid field surveying.



Application-2: ImMesh for rapid, lossless texture reconstruction (Trail #2)

In this application, we show how ImMesh can be applied in applications of lossless texture reconstruction for rapid field surveying.



谢谢

扫描此↑↑↑二维码
访问我的GitHub主页

扫描此↑↑↑二维码
访问我们实验室主页