

## 目录

|                              |    |
|------------------------------|----|
| 解析概要                         | 2  |
| 一、题目                         | 3  |
| 二、摘要                         | 3  |
| Q1: FAST-LIV02 是什么?          | 3  |
| Q2: FAST-LIV02 解决什么?         | 3  |
| Q3: FAST-LIV02 做了什么?         | 3  |
| Q4: FAST-LIV02 证明了什么?        | 4  |
| 三、引言                         | 5  |
| 1、SLAM 的意义:                  | 5  |
| 2、SLAM 的问题 (FAST-LIV02 的动机): | 5  |
| 3、FAST-LIV02 的思路及意义:         | 5  |
| 4、充分的实验:                     | 6  |
| 四、相关工作                       | 7  |
| 1、关于直接法的简介:                  | 7  |
| 2、多传感器融合 SLAM 简介             | 8  |
| 3、LIVO 的发展时间轴。               | 8  |
| 五、系统概述                       | 10 |
| 六、顺序更新的 ESIKF                | 11 |
| 1、符号和状态转移模型。                 | 11 |
| 2、扫描重合并                      | 12 |
| 3、传播 (前向传播和反向传播)             | 12 |
| 4、顺序更新                       | 14 |
| 七、局部地图                       | 17 |
| 1、地图结构                       | 17 |
| 2、几何 (属性) 构建及更新              | 18 |
| a. 点到地图各节点的分配                | 18 |
| b. 新的八叉树节点生成                 | 18 |
| c. 已有八叉树节点的更新                | 19 |
| 3、视觉地图点生成及更新                 | 19 |
| 4、参考 (图像) 块更新                | 20 |
| 5、法向量细化                      | 21 |
| 八、激光雷达测量模型                   | 23 |
| 1、点到平面的观测模型                  | 23 |
| 2、对激光雷达点不确定性的讨论              | 24 |
| 九、视觉测量模型                     | 26 |
| 1、视觉地图点的选择                   | 26 |
| a. 可视的体素查询                   | 26 |
| b. 按需的投射                     | 26 |
| c. 离群点剔除                     | 27 |
| 2、稀疏直接的视觉观测模型                | 28 |
| a. 简单的光度误差                   | 28 |
| b. Fast-LIV02 的光度误差          | 29 |
| 十、结语                         | 30 |

近年来，同步定位与建图（SLAM）技术在机器人、自动驾驶、增强现实等领域取得了显著进展，尤其是激光-惯性视觉融合系统（LIV）因其鲁棒性和适应性受到了广泛关注。FAST-LIV02 作为该方向的最新研究成果，通过创新的系统设计和算法优化，在精度、效率和鲁棒性上实现了新的突破，为实时 SLAM 系统树立了新的标杆。

在研读和解析 FAST-LIV02 论文的过程中，我深刻感受到作者团队在理论推导与工程实践上的深厚功底。尽管我力求全面、准确地理解论文的贡献与技术细节，但受限于个人水平，解析中难免存在疏漏或理解不足之处。在此，我诚挚欢迎各位同行与读者批评指正，共同探讨这一前沿工作的核心思想与潜在价值。

本文的解析旨在为 SLAM 领域的研究者提供参考，希望能为相关方向的技术发展略尽绵力。若文中观点能为读者带来些许启发，我将倍感荣幸。

## 解析概要

范围：本次解析的内容仅限于实验之前的部分

特点：由于论文长度限制，很多基础知识和扩展无法在 FAST-LIV02 中详细介绍和解释，本文在对论文和源码两者的理解上，对 FAST-LIV02 原理进行补充和讨论，希望帮助读者了解到某个知识来自什么，为什么以及做了什么。

重点：1、关于 ESIKF 的来源；2、点云运动去畸变；3、局部地图管理；4、激光雷达/相机观测模型的雅可比矩阵推导。

## 一、题目

### FAST-LIVO2: Fast, Direct LiDAR-Inertial-Visual Odometry

快速，直接的激光雷达-惯性-视觉里程计

基于对 Mars-Lab 团队及该团队相关工作的了解，笔者从题目中猜测 FAST-LIVO2 是 Fast-LIO 系列的扩展，是对 FAST-LIVO 的改进，其中 Direct 表明该框架的激光和视觉部分采用的是直接法，无需特征提取。

## 二、摘要

笔者将摘要凝练为以下 4 个方面：

- 1、FAST-LIVO2 是什么；
- 2、FAST-LIVO2 解决什么；
- 3、FAST-LIVO2 做了什么；
- 4、FAST-LIVO2 证明了什么。

### Q1: FAST-LIVO2 是什么？

A1: FAST-LIVO2 是一个实时精确且鲁棒的多传感器紧耦合 SLAM 系统，利用误差状态迭代卡尔曼滤波 ESIKF 将激光雷达-惯性-视觉三种异构传感器有效地集成到一起。（笔者注：激光雷达包括重复式扫描的多线旋转激光雷达（Velodyne-16, XT-32, Ouster-64, Pandar128），非重复扫描的固态雷达（Livox-avia），相机包括针孔相机和鱼眼相机。）

### Q2: FAST-LIVO2 解决什么？

A2: a. 激光雷达和相机之间的测量数据异构性；b. 系统的效率；c. 系统的精度；d. 系统鲁棒性。

### Q3: FAST-LIVO2 做了什么？

A3: a. 采用一种顺序更新策略（先运行 LIO，基于 LIO 更新的状态，再运行 VIO），将激光雷达和相机两种异构测量数据有效地融合到一个系统中。

b. 直接配准原始数据，避免复杂耗时的特征提取；激光雷达和相机的测量都被融合到一个统一的体素图（八叉树）中，三维数据的高效管理。

c. 建立激光点到图像块的关联，实现精确的图像对齐；对激光点对应的平面参数进行动态精细化，进一步提升对齐精度。

d. 按需的体素射线投射克服处于激光雷达盲区的情况；相机曝光时间在线估计克服光照不均环境相机曝光时间变化导致的时间戳偏移。

Q4: FAST-LIVO2 证明了什么？

A4: 实时性（效率）：UAV 导航实验；精度/鲁棒性：基于无人机的三维重建；可扩展性：输出结果可直接用于 Nerf 或 3D-GS 渲染。

### 三、引言

笔者将引言凝练为四个部分：

- 1、SLAM 的意义；
- 2、SLAM 的问题（FAST-LIVO2 的动机）；
- 3、FAST-LIVO2 的思路及意义；
- 4、充分的实验。

#### 1、SLAM 的意义：

介绍 SLAM 的应用案例及潜力。

#### 2、SLAM 的问题（FAST-LIVO2 的动机）：

先通过单传感器的问题引出多传感器，再从多传感器的问题引出 FAST-LIVO2。

对单传感器 SLAM 的介绍，a. 视觉 SLAM：具有丰富的纹理信息和场景理解潜力，但缺乏深度信息和对低纹理、光照不均以及噪声的适应能力。b. 激光 SLAM：具备精确的深度信息，但缺乏纹理信息和对结构化环境（隧道，单面墙壁）的适应能力。

多传感器融合 SLAM(激光雷达-惯性-视觉 SLAM: LIVO) 的必要性：更加精确的状态估计，稠密且富含纹理的三维地图，更强的环境适应性（即使单个传感器退化）。

目前 LIVO 仍面临的挑战：a. 板载有效算力下，海量点云的计算效率低下；b. 基于特征提取的方法可降低计算负载，但特征提取的工程技巧繁多且不适应缺乏结构或纹理的环境。c. 统一的地图管理可有效提升系统效率，但如何设计一个适用激光雷达和相机两种异构测量的数据结构。d. 精确的纹理地图需要达到像素级精度，这对工程和算法要求很高（精确的硬件同步，精确的传感器外参，传感器噪声分析，高效的算法设计）。

#### 3、FAST-LIVO2 的思路及意义：

上面这些是问题，也是动机。

FAST-LIVO2 使用 ESIKF 先后对激光模块和视觉模型进行状态更新：用 IMU 的测量进行 ESIKF 的状态预测，获得先验状态；基于该先验状态对激光雷达点云进行反向传播(去畸变)，计算激光雷达测量方程的值(点的平面距离)、进行 ESIKF

的迭代更新、体素八叉树地图(平面先验)更新,获得准后验状态;基于准后验状态,将激光点作为视觉地图点,计算视觉地图点的参考块与视觉地图点在当前图像上投影块的光度误差(测量方程的值),进行 ESIKF 的迭代更新、曝光时间更新、该视觉点的参考块更新,获得后验状态。此外,FAST-LIVO2 采用**按需的体素光线投射**(笔者注:这里不理解也没有关系,后面会详细解释)来克服激光雷达盲区。

贡献: 1、克服异构测量的顺序更新 ESIKF 框架,提升系统鲁棒性。2、**使用及细化激光雷达平面先验(平面参数)来提升精度,而非假设图像块具有一致深度**(笔者注:这里若是不理解也没关系,主要区别在于是否基于平面参数来计算仿射变换阵,这部分可以看代码解析)。3、提出参考块更新策略提升图像对齐精度。4、在线曝光时间估计应对纹理变化。5、按需体素射线增强无激光点时的系统鲁棒性。

#### 4、充分的实验:

涉及公开数据集(Hilti and NTU-VIRAL)的 25 个序列,私有数据集;SLAM 开源 SOTA 算法(R3LIVE, LVI-SAM, FAST-LIO2 等等),定性及定量分析。此外还要三个可用性实验(实时性(效率): UAV 导航实验;精度/鲁棒性: 基于无人机的三维重建;可扩展性: 输出结果可直接用于 Nerf 或 3D-GS 渲染。)

## 四、相关工作

笔者将相关工作凝练为三个部分：

1、关于直接法 SLAM 的简介；

2、LIVO 简介；

3、LIVO 的发展时间轴（时间轴仅基于该论文提及的工作，笔者能力有限，难免出现遗漏）。

### 1、关于直接法的简介：

对特征法和直接法进行了优劣分析。

**视觉直接法的分类：**稠密直接法、半稠密直接法以及稀疏直接法。

稠密直接法（全像素）：应用于 RGB-D 相机，例如《Real-time dense visual tracking under large lighting variations》、《Direct iterative closest point for real-time visual odometry》以及《Robust odometry estimation for RGBD cameras》。

半稠密直接法（显著灰度梯度像素）：例如《LSD-SLAM: Large-scale direct monocular SLAM》、《Semi-dense visual odometry for a monocular camera》。

稀疏直接法（图像块）：几个精挑细选的图像块，例如《Direct sparse odometry》、《SV0: Fast semi-direct monocular visual odometry》

**激光雷达直接法：**没有分类，对扫描进行降采样，构建点到平面约束。例如《FAST-LIO2: Fast direct lidar-inertial odometry》、《Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry》、《Direct lidar-inertial odometry: Lightweight LIO with continuous-time motion correction》以及《D-LIO: Tightly-coupled direct lidar-inertial odometry and mapping》。

**FAST-LIVO2 中的直接法：**对于激光雷达模块，采用《Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry》进行处理（笔者注：使用哈希表和八叉树对体素地图进行管理）。对于视觉模块，采用《SV0: Fast semi-direct monocular visual odometry》进行处理。（笔者注：使用稀疏直接法计算光度误差对位姿扰动的雅克比矩阵，但 FAST-LIVO2 又与 SV0 有坐标系的区别，还需要再处理一步，这可以在代码解析中详细了解）。

## 2、多传感器融合 SLAM 简介

(笔者注：非常适合想快速了解 LIVO 相关工作的小伙伴)

FAST-LIVO2 作者们将 LIVO 的相关工作分为松耦合和紧耦合两类，分类的依据主要有两个方面：a. 状态估计方面：一个传感器的估计是否在另一个传感器模型中作为优化目标；b. 原始测量方面：是否合并了来自不同传感器的原始数据。

## 3、LIVO 的发展时间轴。

首先按照 2 中的分类标准将整个 LIVO 分为了两类，如下图所示：



图 1 LIVO 发展时间轴

对于松耦合方法，从状态和测量两个方面：

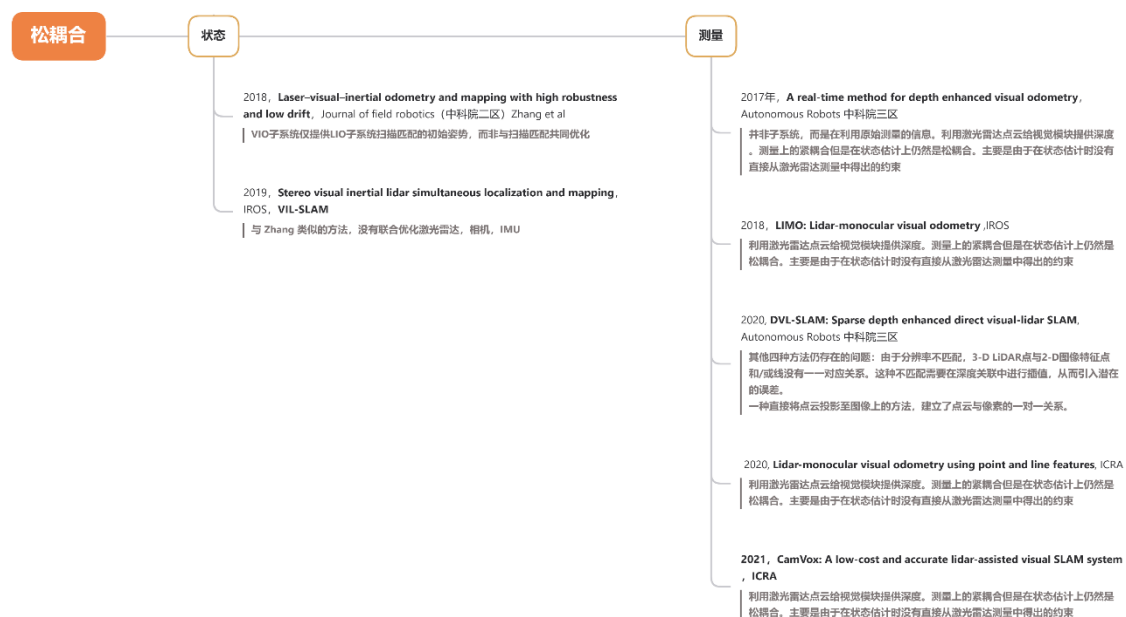


图 2 松耦合方法发展时间轴

对于紧耦合方法，论文从间接法和直接法两方面讨论：

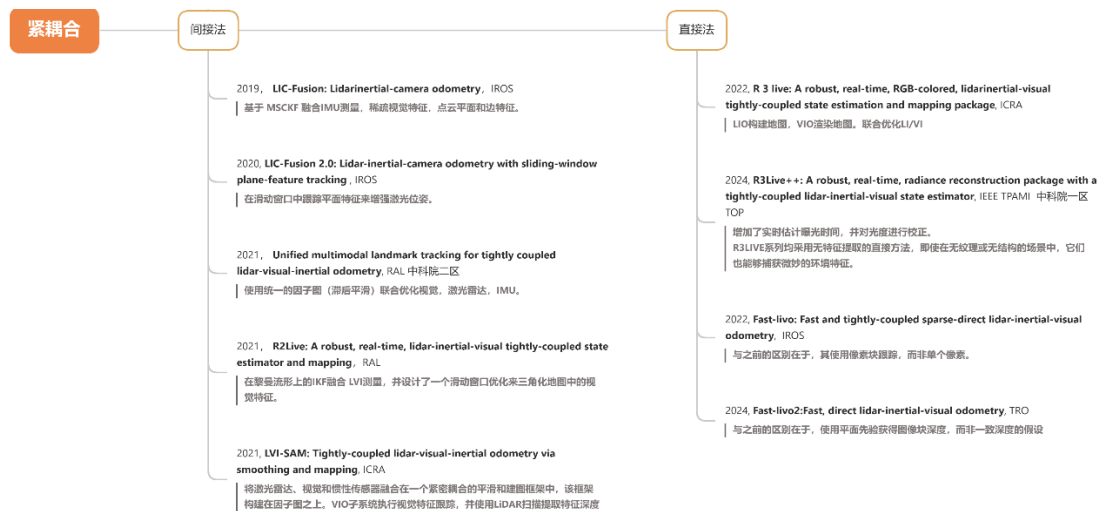


图 3 紧耦合方法发展时间轴

五、系统概述

这一节主要是介绍整个系统的处理流程。

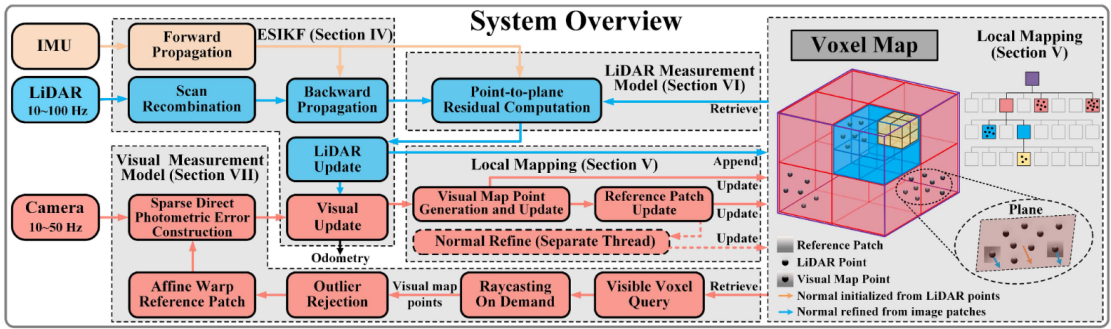


图 4 系统结构图

可以看到系统由四个核心模块组成：1、顺序更新的 ESIKF; 2、局部地图构建; 3、激光雷达观测模型; 4、视觉观测模型。其中 1 主要是关于 ESIKF 的状态预测/更新方程，3 和 4 则是 ESIKF 的状态观测方程。

异步采样的激光雷达点被重合并到相机的采样时刻---》使用 ESIKF 进行顺序更新（先激光雷达后相机），都利用体素地图进行管理---》激光雷达测量模型为帧到地图，点到平面的残差。---》利用可见体素查询和按需光线投射从地图中提取当前 FoV 内的视觉地图点，再计算帧到地图（笔者注：所谓帧到地图更新就是当前帧与参考帧的光度误差随 $G T_I$ 扰动变化的变化。）的图像光度误差实现视觉更新。---》局部地图维护：LIO 更新地图的几何，VIO 更新与地图点关联的当前图像块&参考图像块。（更新的参考图像块在单独的线程中进一步细化它们的法向量。）

## 六、顺序更新的 ESIKF

若读者对 ESIKF 熟悉或者对 Fast-Lio 系统熟悉，可跳过此节。若你和我一样是一个小小小白，不妨跟我一起来看看 ESIKF。

这一节主要是介绍状态预测及更新的数学模型。由四个小节构成：1、符号和状态转移模型；2、扫描重合并；3、传播（前向传播和反向传播）；4、顺序更新。

### 1、符号和状态转移模型。

首先对传感器坐标系进行了约定，IMU 系为  $I$ ，全局系为  $G$ 。（笔者注：这里只约定了 IMU 系和全局系，因为整个系统的状态都是基于 IMU 的，激光雷达和相机的观测也都会变换到 IMU 系下）。

其次是离散时间的状态转移方程  $\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus (\Delta t \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i))$ 。

其中  $\mathbf{x}_i$  为状态向量， $\mathbf{u}_i$  为 IMU 的观测， $\mathbf{w}_i$  为过程噪声， $\Delta t$  为 IMU 采样间隔以及矩阵  $\mathbf{f}$  为运动学方程。（笔者注：在推导状态空间方程时，一些方法是从连续时间对运动学方程求导，获得连续时间的状态空间方程，然后对其进行离散化。离散化通常有欧拉积分和中值积分两种。这样的推导其实挺麻烦，因为还要求连续时间的状态空间方程，而且离散化时选的如果是中值积分，还要从 R-v-p 逐个带入。更为方便的办法就是直接把运动学方程离散化，再直接对离散的运动学方程推导离散的状态空间方程。关于状态空间方程的推导，这里推荐读者去[博客](#)了解更多细节。）

**对  $\boxplus$  的解释：**如果是只看了 Fast-LIV02 论文的读者或许会对这个  $\boxplus$  产生困惑，这是个什么符号？其实， $\boxplus$  并不复杂，在论文《FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter》中对其进行过解释，它的出现主要是因为旋转  $R \in SO(3)$  这个状态量，官方的说法是  $SO(3)$  对加法不封闭，说简单点就是：有一个坐标系  $F$ ，先旋转  $R_1^0$  再旋转  $R_2^1$ ，最终得到  $R_2^0$ ，但是  $R_2^0 \neq R_1^0 + R_2^1$ ，而是  $R_2^0 = R_1^0 R_2^1$ ，那么在表示状态增量的时候原来的  $+$  就不合适了，因此使用  $\boxplus$  来表示  $SO(3)$  和  $\mathbb{R}^n$  两种空间的“加”便再合适不过了。（笔者注：其实这是我简化后的解释，对于  $SO(3)$  的“加”和“减”，我觉得读者应当对流形和罗德里格斯公式多多了解一下，这可以从高翔老师的《自动驾驶与机器人种的 SLAM 技术》或《视觉 SLAM 十四讲》学习到。）

## 2、扫描重合并

扫描重合并论文中的说法是：在相机采样时刻将高频、顺序采样的激光雷达原始点分割成不同的激光雷达扫描。这确保了相机和 LiDAR 数据以相同的频率（例如 10Hz）同步，从而允许同时更新状态。

（笔者注：我说说本人一开始的理解和看完代码后的理解。

起初理解：笔者对激光雷达有些了解，激光雷达不是在某时刻同时获得所有点，而是按顺序一个一个的产生，比如一帧有 24000 个点，采集完这 24000 个点的耗时是 100ms，在  $t_{k-1}$  时刻开始采集，这也是一帧点云的帧头时间，在  $t_k$  时刻采集完成，此时这些点合并为一帧。我认为这就是扫描重合并了，但其实不是。

**读完代码理解：**由于相机触发和曝光时间不是绝对稳定，为了获得相机捕获时刻采集的点云，需要对点云的每个点进行判断，点的时间戳在图捕获时间之前的视为当前帧点云，点的时间戳在图像捕获时间之后的视为下一帧点云。这才是重合并。）

这里再多说一点，在复现和阅读源码的时候相信有读者对源码中相机时间需额外加 100ms 表示困惑，这里笔者按个人理解解释一下：在  $t_{k-1}$  时刻激光雷达开始采集，激光雷达采集的还只是第一个点，那么到  $t_k$  时刻，激光雷达才采集完一帧， $t_k$  时刻相机也采集了一个图像，相机驱动通过时间共享文件（郑博自己写的一个文件，用于存放每帧点云的帧头时间）将这一帧点云的帧头  $t_{k-1}$  时刻作为相机的时间戳，并一起给到系统。但实际上相机数据是在  $t_k$  时刻采集的，那么激光雷达的原始时间戳与相机时间戳应该差个 100ms。

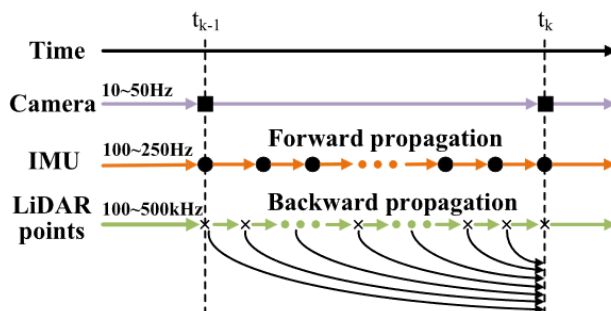


图 5 传感器数据采集时间轴

## 3、传播（前向传播和反向传播）

关于前向传播，本质上就是 EKF 的预测公式：在预测阶段将噪声  $w_i$  (因为无法知道) 设为 0，预测  $t_{k-1}$  到  $t_k$  这段时间每个 IMU 输入时的状态。由 KF 预测可

得此时状态量为，状态量对应的协方差为，这作为更新阶段的先验。

$$\begin{aligned}\hat{\mathbf{x}}_{i+1} &= \hat{\mathbf{x}}_i \oplus (\Delta t \mathbf{f}(\hat{\mathbf{x}}_i, \mathbf{u}_i, \mathbf{0})); \hat{\mathbf{x}}_0 = \bar{\mathbf{x}}_{k-1}, \\ \hat{\mathbf{P}}_{i+1} &= \mathbf{F}_{\tilde{\mathbf{x}}_i} \hat{\mathbf{P}}_i \mathbf{F}_{\tilde{\mathbf{x}}_i}^T + \mathbf{F}_{\mathbf{w}_i} \mathbf{Q}_i \mathbf{F}_{\mathbf{w}_i}^T; \hat{\mathbf{P}}_0 = \bar{\mathbf{P}}_{k-1},\end{aligned}$$

肯定有读者对状态系数阵 $\mathbf{F}_{\tilde{\mathbf{x}}_i}$ 和 $\mathbf{F}_{\mathbf{w}_i}$ 怎么来的好奇，我这里推荐一个[博客](#)，这篇博客对状态系数阵 $\mathbf{F}_{\tilde{\mathbf{x}}_i}$ 和 $\mathbf{F}_{\mathbf{w}_i}$ 推导和具体形式做了简单易懂的解释。

**对于反向传播**，本质上是补偿激光雷达的运动畸变，确保所有的点都是在  $t_k$  时刻观测到的。

什么是运动畸变？正如[览沃干货系列](#)所提到的：激光雷达点云自运动畸变的产生本质上是一帧中每一个点的坐标系不同。如下图，由于激光雷达自身在一帧时间内存在“剧烈”运动，雷达自身分别在三个不同的实际姿态下对三个点进行了扫描。因此在最后得到的点云中，三个点坐标实际处于不同的坐标系，看起来不再共线了：

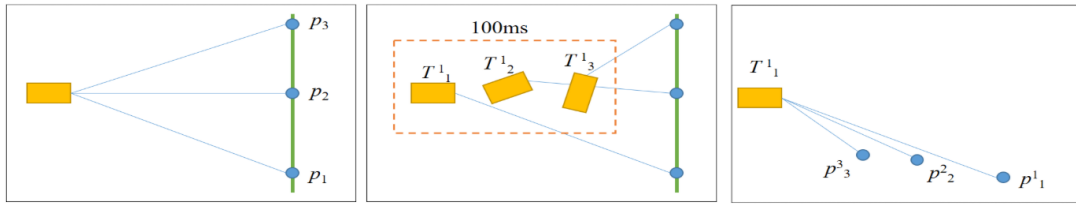


图 6：点云运动畸变示例

解决的办法就是获得  $t_{k-1}$  到  $t_k$  时间段内，每个点相对  $t_k$  时刻的变换阵 ${}^{I_k}\check{T}_{I_j}$ ，然后将每个时刻的点都变换到  $t_k$  时刻下——《FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter》。

首先，建立反向传播的状态转移方程： $\check{\mathbf{x}}_{j-1} = \check{\mathbf{x}}_j \oplus (-\Delta t \mathbf{f}(\check{\mathbf{x}}_j, \mathbf{u}_j, \mathbf{0}))$ 。设  $t_k$  时刻为初始位姿，这里的 $\Delta t$ 为激光采样点之间的时间间隔。

其次，根据 IMU 的采样间隔将  $t_{k-1}$  到  $t_k$  时间段内的点进行分割，存在点云时刻 $p_{j-1} \in [\tau_{i-1}, \tau_i)$ ， $[\tau_{i-1}, \tau_i)$ 为两帧 imu 时间间隔。(以 10hz 的激光雷达(avia)和 200hz 的 IMU 为例，100ms 有 20 个 imu 数据，有一帧激光点云，一帧激光点云有 24000 个点，因此每两个 imu 数据之间存在若干个点。)

然后计算各时刻相对  $t_k$  的变换阵：a. 旋转阵 ${}^{I_k}\check{R}_{I_j}$ ；b. 速度 ${}^{I_k}\check{v}_{I_j}$ (用于求平移量)；c. 平移 ${}^{I_k}\check{p}_{I_j}$ 。

a. j 时刻点相对 tk 时刻的旋转阵  ${}^{I_k}\check{\mathbf{R}}_{I_j}$  :

$${}^{I_k}\check{\mathbf{R}}_{I_{j-1}} = {}^{I_k}\check{\mathbf{R}}_{I_j} \text{Exp}((\hat{\mathbf{b}}_{\omega_k} - \omega_{m_{i-1}})\Delta t), \text{ 初始位姿 } {}^{I_k}\mathbf{R}_{I_m} = \mathbf{I},$$

可以发现，这里假设在  $[\tau_{i-1}, \tau_i)$  时间内，角速度是恒定的，用的是  $\tau_{i-1}$  时刻的 IMU 角速度，但偏差依然还是 tk 时刻。

b. j 时刻点相对 tk 时刻的速度  ${}^{I_k}\check{\mathbf{v}}_{I_j}$ :

$$\begin{aligned} {}^{I_k}\check{\mathbf{v}}_{I_{j-1}} &= {}^{I_k}\check{\mathbf{v}}_{I_j} - {}^{I_k}\check{\mathbf{R}}_{I_j}(\mathbf{a}_{m_{i-1}} - \hat{\mathbf{b}}_{\mathbf{a}_k})\Delta t - {}^{I_k}\hat{\mathbf{g}}_k\Delta t, \\ \text{s.f. } {}^{I_k}\check{\mathbf{v}}_{I_m} &= {}^G\hat{\mathbf{R}}_{I_k}^T {}^G\hat{\mathbf{v}}_{I_k}, \quad {}^{I_k}\hat{\mathbf{g}}_k = {}^G\hat{\mathbf{R}}_{I_k}^T {}^G\hat{\mathbf{g}}_k; \end{aligned}$$

这段间隔内的加速度都被假设为  $\tau_{i-1}$  时刻 IMU 的加速度测量（假设加速度是恒定的，这个时间间隔内的所有点的加速度一致）。至于加速度偏差和重力都固定为 tk 时刻的先验值。

c. j 时刻点相对 tk 时刻的平移  ${}^{I_k}\check{\mathbf{p}}_{I_j}$

$${}^{I_k}\check{\mathbf{p}}_{I_{j-1}} = {}^{I_k}\check{\mathbf{p}}_{I_j} - {}^{I_k}\check{\mathbf{v}}_{I_j}\Delta t, \quad \text{s.f. } {}^{I_k}\check{\mathbf{p}}_{I_m} = \mathbf{0};$$

至此，获得了每个点相对 tk 时刻的变换阵  ${}^{I_k}\check{\mathbf{T}}_{I_j}$ ，变换后点： ${}^{L_k}\mathbf{p}_{f_j} = {}^{I_k}\mathbf{T}_L^{-1} {}^{I_k}\check{\mathbf{T}}_{I_j} {}^{I_k}\mathbf{T}_L {}^{L_j}\mathbf{p}_{f_j}$ 。

#### 4、顺序更新

Fast-livo2 作者从贝叶斯概率的角度解释了 Fast-livo2 中涉及的观测（激光雷达/相机，观测的具体形式请参照第八，第九节）和状态在 EKF 更新中的意义。既然说到了概率，那必然涉及分布，请读者注意，下面将有大量分布出没。

通过上面的前向传播，我们获得了先验状态  $\hat{\mathbf{x}}$ （这个先验状态和真实状态  $\mathbf{x}$ （不可知）之间存在一个误差（不可知） $\mathbf{x} \ominus \hat{\mathbf{x}}$ ，这个误差服从分布  $\mathcal{N}(\mathbf{0}, \hat{\mathbf{P}})$ ）、先验状态协方差  $\hat{\mathbf{P}}$  以及基于先验状态计算得到的各观测值（考虑了激光雷达测量噪声  $\mathbf{v}_l \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{v}_l})$  和相机测量噪声  $\mathbf{v}_c \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{v}_c})$ ):

$$\begin{bmatrix} \mathbf{y}_l \\ \mathbf{y}_c \end{bmatrix} = \begin{bmatrix} \mathbf{h}_l(\mathbf{x}, \mathbf{v}_l) \\ \mathbf{h}_c(\mathbf{x}, \mathbf{v}_c) \end{bmatrix}$$

先看看整个顺序更新 ESIKF 在贝叶斯概率中的形式:

$$\begin{aligned}
p(\mathbf{x}|\mathbf{y}_l, \mathbf{y}_c) &\propto p(\mathbf{x}, \mathbf{y}_l, \mathbf{y}_c) = p(\mathbf{y}_c|\mathbf{x}, \mathbf{y}_l) p(\mathbf{x}, \mathbf{y}_l) \\
&= p(\mathbf{y}_c|\mathbf{x}) \underbrace{p(\mathbf{y}_l|\mathbf{x}) p(\mathbf{x})}_{\propto p(\mathbf{x}|\mathbf{y}_l)}.
\end{aligned}$$

可以看到整个顺序更新是由两个贝叶斯概率构成，从右往左分别对应着 LI0 和 VI0。

**LI0 贝叶斯概率**  $p(\mathbf{x}|\mathbf{y}_l) \propto p(\mathbf{y}_l|\mathbf{x})p(\mathbf{x})$ : IMU 的先验状态提供了该状态下的激光雷达观测值，而激光雷达观测值又可以作为约束条件来修正/更新状态。

**VI0 贝叶斯概率**  $p(\mathbf{x}|\mathbf{y}_l, \mathbf{y}_c) \propto p(\mathbf{y}_c|\mathbf{x})p(\mathbf{x}|\mathbf{y}_l)$ : 基于 LI0 模块更新的状态可以获得此状态下相机观测量，而该观测量又能够作为约束条件进一步修正状态。

作者将这两种概率用一种统一的形式表达:  $q(\mathbf{x}|\mathbf{y}) \propto q(\mathbf{y}|\mathbf{x})q(\mathbf{x})$ 。其中  $q(\mathbf{x})$  表示先验状态分布， $q(\mathbf{y}|\mathbf{x})$  表示基于先验状态获得的观测分布。

又进一步对基于先验状态的观测  $\mathbf{y}|\mathbf{x}$  做了展开（主要是将观测方程进行一阶泰勒展开来获得观测方程的系数矩阵  $\mathbf{H}$  以及观测协方差阵  $\mathbf{R}$ （由传感器噪声导致的观测值的不确定性））:

$$\begin{aligned}
\mathbf{y}|\mathbf{x} &\simeq \underbrace{\mathbf{h}(\hat{\mathbf{x}}^\kappa, \mathbf{0})}_{\mathbf{z}^\kappa} + \mathbf{H}^\kappa \delta \mathbf{x}^\kappa + \mathbf{L}^\kappa \mathbf{v} \\
q(\mathbf{y}|\mathbf{x}) &\simeq \mathcal{N}(\mathbf{h}(\hat{\mathbf{x}}^\kappa, \mathbf{0}) + \mathbf{H}^\kappa \delta \mathbf{x}^\kappa, \mathbf{R})
\end{aligned}$$

关于观测方程系数矩阵  $\mathbf{H}$  的具体形式（论文未给出），笔者将在第八和第九节分别介绍。

有了观测方程的系数矩阵  $\mathbf{H}$  以及观测协方差阵  $\mathbf{R}$  便可以进行迭代求 **ESIKF** 的增益和后验状态了:

$$\begin{aligned}
\mathbf{K} &= \left( (\mathbf{H}^\kappa)^T \mathbf{R}^{-1} \mathbf{H}^\kappa + \hat{\mathbf{P}}^{-1} \right)^{-1} (\mathbf{H}^\kappa)^T \mathbf{R}^{-1} \\
\hat{\mathbf{x}}^{\kappa+1} &= \hat{\mathbf{x}}^\kappa \boxplus (-\mathbf{K} \mathbf{z}^\kappa - (\mathbf{I} - \mathbf{K} \mathbf{H}^\kappa) (\hat{\mathbf{x}}^\kappa \boxminus \hat{\mathbf{x}})).
\end{aligned}$$

这里的上标  $k$  代表迭代次数，一般不超过 5 次。增益  $\mathbf{K}$  虽然与 EKF 的一般增益  $\mathbf{K} = \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1}$  形式不同，但实际上是等价的（具体证明可阅读《FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter》），为什么这么写？因为在  $\mathbf{K} = \mathbf{P} \mathbf{H}^T (\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R})^{-1}$  中需要计算  $\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R}$  的逆，而  $\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R}$  是一个  $m \times m$  的矩阵， $m$  是雷达点的数量，而雷达点的数量是非常巨大的，求如此规模矩阵的逆是非常困难的。

协方差则是在迭代完成后更新一次即可（因为迭代的时候根本不涉及协方差）： $\bar{x} = \hat{x}^{k+1}, \bar{P} = (I - KH)\hat{P}$ 。

对于 VIO 部分的更新，笔者这里再多说一点，VIO 模块采用的是多层直接法，即对原图像构建金字塔，因此会对每一层都进行  $\leq 5$  次的迭代，每层的最优状态作为下一层的先验状态继续迭代。

## 七、局部地图

这一节主要是介绍 Fast-LIVO2 的地图数据是怎么构建的，构建元素是什么，怎么更新的，由五小节组成：1、地图结构；2、几何（属性）构建及更新；3、视觉地图点生成及更新；4、参考（图像）块更新；5、法向量细化。

### 1、地图结构

正如前面第四节第一小节中提到了，Fast-LIVO2 的激光雷达点云处理和地图管理部分采用的是论文《Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry》。

这是一种自适应体素结构的地图。有读者就疑问了，什么是体素结构，什么又是自适应？

**什么是体素结构：**整个地图由  $0.5\text{m} \times 0.5\text{m} \times 0.5\text{m}$  大小的体素块构成，所有的体素块由哈希表进行组织，实现快速检索（每个体素对应一个哈希表的索引，同时每个体素还对应着一个八叉树的根节点）。

**什么是自适应：**现实环境中的平面大小是不一的，有的一个体素只有一个平面（这种就没有再细分的必要），有的呢一个体素有好几个平面（这个体素没法拟合出一个平面），这时候就需要对这个体素细分，先分八个子节点，判断每个子节点能否拟合出平面，能拟合则停止，不能拟合的子节点继续细分，非为八个孙节点，能拟合出平面最好，不能拟合也停止（再分影响效率，意义也不大）。

这种根节点、子节点和孙节点的不同大小使其可以代表不同尺度的局部平面，因此适应具有不同结构的环境。

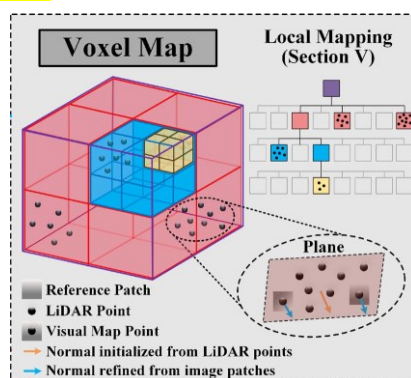


图 7 体素地图及八叉树节点

知道了什么是自适应体素结构，我们再来看看结构里面的每个节点吧：每个节点（根/子/孙）都可以代表一个局部平面（如果能拟合出来的话），并存储这个

平面的一些属性（平面中心点，平面法向量，平面不确定性等等）以及该平面上的雷达原始点。

通过对这些雷达原始点进行筛选（比如使用 `vk::shiTomasiScore` 函数计算该点在图像投影点的分数值），可以将其视为**视觉地图点**，同时这些点将与三层 8x8 的图像块（金字塔）相关联。收敛（比如该视觉点对应的平面法向量更新量很小时就是收敛）的视觉地图点仅与参考块关联，未收敛的点与参考块和其他可见块都关联。

此外，为了防止地图无限制增长，以当前位置为中心设置了一个局部地图范围。

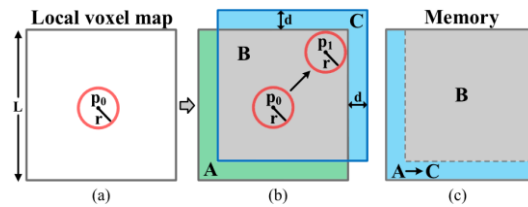


图 8 滑动地图

设置了一个初始地图范围(a)后，就在这个范围进行探测和建图，当雷达探测范围触碰到范围边界，局部地图范围会在该方向上移动  $d$  距离。存储的 A 处点云数据将被重置为 C 处的数据。每次 ESIKF 更新后就检查一次地图是否移动。

## 2、几何（属性）构建及更新

本小节跟激光雷达关联甚大，笔者将其总结为三点：a. 点到地图各节点的分配；b. 新的八叉树节点生成；c. 已有八叉树节点的更新。

### a. 点到地图各节点的分配

每次 ESIKF 更新后，将每个激光雷达点注册到全局地图中，并确定每个点位于哪个体素内。判断这个体素有没有对应的八叉树根节点，如果没有就初始化一个新的根体素，如果有就添加点到已有体素。

### b. 新的八叉树节点生成

判断体素内的点是否位于平面上。如果是，则计算平面的相关属性：**平面的中心点  $\mathbf{q} = \bar{\mathbf{p}}$ ，法向量  $\mathbf{n}$ ，平面协方差  $\Sigma_{\mathbf{n}, \mathbf{q}}$** 。平面协方差被用于表示平面的不确定性，这受位姿估计不确定性和测量噪声影响。如果不是，则将体素继续细分为八个较小的子节点（子节点再分为孙节点），直到确定各节点中的点集可形成平面

或达到最大层。如果到最大层，还没有平面，那么子/孙节点中的点将被丢弃（笔者注：这里的丢弃不是至直接把点云缓存器中的点删除了，而是节点中的点向量成员不保留点了）。

### c. 已有八叉树节点的更新

判断新来的点和之前存有的点能否构成平面。如果可以则重新计算一下加入新点后平面的相关属性。如果不能，则重复 b 中的细分过程。一旦平面参数是收敛（平面法向量更新很小）的，该平面将被视为成熟面，平面的相关属性将被视为固定，平面上的新点将被丢弃。

此外，对于成熟的平面，50 个最近的激光雷达点被作为候选点用于生成视觉地图点。对于不成熟的平面，所有点都作为候选点。视觉地图点生成过程中会确定其中的一些候选点作为视觉地图点并将图像块与它们关联用于图片对齐（这句话可以和本节的第一点中关于视觉地图点的生成关联上）。

## 3、视觉地图点生成及更新

这一小节将告诉读者，视觉地图点是怎么生成及更新的。（笔者注：这一小节不太好解释，总感觉怎么说都不清楚）

首先是筛选：需要筛选才能执行生成视觉地图点操作。主要通过两个条件来筛选：a. 这些候选点在当前的相机视角下是可见的（将世界系下的这些点根据变换阵  $T_{c-w}$  变换到当前相机坐标系，保留相机系下正前方的点，再根据相机内参投影到像素坐标系，保留像素坐标在图像边界内的点。）；b. 再找投影到图像边界内的点的像素值是否具有显著的像素梯度。

在视觉更新之后，将这些候选点投射到当前图像上，并保留每个体素中距离平面最近的候选点。（笔者注：保留最近点的操作并不在源码函数 `generateVisualMapPoints(img,pg)` 中，而是在该函数之前的函数 `retrieveFromVisualSparseMap(img, pg, feat_map)` 中实现）。

生成视觉地图点：生成这一步其实没啥讲究，只要点符合筛选条件，就用这个点的属性生成一个视觉地图点即可。（至于上面说的保留最近点的操作，笔者这里再详细说一说：假设整个图像可以看作是由大小为  $30\text{pixel} \times 30\text{pixel}$  的栅格单元拼成的，通过计算视觉地图点的像素坐标可以知道这个视觉地图点位于哪个栅格单元，如果有视觉地图点位于这个栅格，后面又有视觉地图点位于这个栅格，

那么这个栅格只跟距离图像最近的视觉地图点建立关联。)

投影到该栅格的视觉地图点将与**当前图像块** (8 pixel \*8 pixel, 按金字塔层级来说, 每一层都有一个图像块, 每个块大小一样)、估计的当前状态 (即帧姿态和曝光时间) 以及该点的平面法向量相关联。如果栅格单元不包含投影的任何视觉地图点, 将使用具有最高灰度梯度的候选点生成一个新的视觉地图点并执行上述的关联操作。

**需要注意的是:** 视觉点与图像块的关联并非每帧都做, 而是需要以下两个条件。a. 超过 20 帧没有添加新的块到视觉地图点了; b. 视觉地图点在当前帧中的像素位置比其在上一个块的位置相差至少 40 像素 (运动幅度较大, 点在屏幕里移动了一大段距离)。

#### 4、参考 (图像) 块更新

什么是图像块? 就是以视觉地图点投影的像素坐标为中心, 划一个 8 pixel \*8 pixel 的块。

视觉地图点在图像上的投影坐标为中心划个块, 将块推入到一个图像块向量中, 从该向量选出一个质量最好的块作为参考快。

怎么选呢? Fast-LIV02 基于光度相似性和视角给每个块  $\mathbf{f}$  评分, 分数最高的设为参考块。

这里先给出整个分数计算公式, 再介绍各符号含义:

$$\begin{aligned} \text{NCC}(\mathbf{f}, \mathbf{g}) &= \frac{\sum_{x,y} [\mathbf{f}(x,y) - \bar{\mathbf{f}}] [\mathbf{g}(x,y) - \bar{\mathbf{g}}]}{\sqrt{\sum_{x,y} [\mathbf{f}(x,y) - \bar{\mathbf{f}}]^2 \sum_{x,y} [\mathbf{g}(x,y) - \bar{\mathbf{g}}]^2}} \\ c &= \frac{\mathbf{n} \cdot \mathbf{p}}{\|\mathbf{p}\|}, \quad \omega_1 = \frac{1}{1 + e^{\text{tr}(\Sigma_{\mathbf{n}})}} \\ S &= (1 - \omega_1) \cdot \frac{1}{n} \sum_{i=1}^n \text{NCC}(\mathbf{f}, \mathbf{g}_i) + \omega_1 \cdot c \quad (12) \end{aligned}$$

NCC: 用于在 0 层金字塔测量块  $\mathbf{f}$  和块  $\mathbf{g}$  的相似性。0 层金字塔具备最高的分辨率。在此前对两个块均进行了均值减法处理-先减去块的亮度均值 (消除光照差异影响), 再计算相关性。

$c$ : 表示法向量  $\mathbf{n}$  与被评估块  $\mathbf{f}$  的观察方向 (参考块所在的相机系原点到视觉地图点的向量)  $\mathbf{p} / \|\mathbf{p}\|$  之间的余弦相似度 (夹角余弦值)。

$\omega_1$ : 表示权重, 其中  $\text{tr}(\Sigma_{\mathbf{n}})$ , 为法向量协方差的迹。

$S$ : 综合分数, 加权求和得到。(EKF 又何尝不是一种加权求和呢, 权重是  $K$ )。

整体流程就是将关联到同一个地图点的所有块都作为一个待评估块遍历一遍, 选一个分数最高的。

这种分数评估机制有什么好处呢: a. 参考块与其余大多数块相似。(这是 MVS[47] 中用于避免选择动态物体上补丁的技术。); b. 参考块的视角方向与平面正交的, 以最高分辨率保留纹理细节。

(笔者注: 一个好的工作, 必然会考虑到各种细节, Fast-LIVO2 能够在农田这种退化的环境精确运行, 属实是情理之中了。)

## 5、法向量细化

读到这里的读者应该也都有这么一个概念: 每个视觉地图点都位于一个局部平面 (不管是成熟还是不成熟的) 上。

为什么要细化法向量呢, 这里其实涉及到 VIO 中的观测方程——**光度误差**。

在本文的开头提到, Fast-LIVO2 的 VIO 采用的是直接法 (参考视觉 SLAM 十四讲的 8.4 节), 直接法需要计算两个像素的光度误差: 像素  $a$  通过仿射变换  $A$  找到其在另一个图像上对应的像素  $b$ , 仿射变换  $A$  越准确找到的  $b$  就越准确, 光度误差就越小。而仿射变换  $A$  是通过单应阵实现的 ([单应阵](#): 空间中同一平面上的三维点在两张图像中的对应关系), 单应阵又与平面法向量相关。因此, 通过细化平面法向量就能获得更精确的单应阵, 进而得到更精确的仿射变换阵  $A$ 。

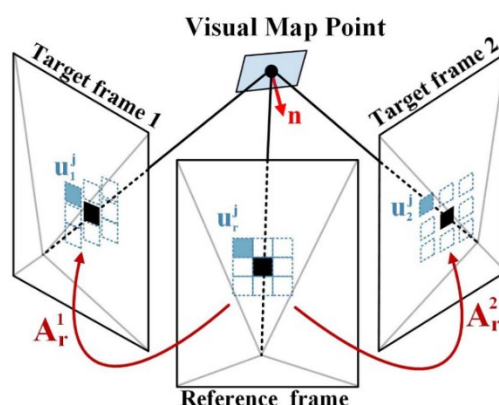


图 9 仿射变换效果

仿射变换公式:

$$u_i^j = A_r^i u_r^j$$

$$A_r^i = P \left( I_i R_{I_r} + I_i t_{I_r} \frac{1}{I_r n^T \cdot I_r p} I_r n^T \right) P^{-1}$$

其中  $P$  为相机内参,  ${}^l_i\mathbf{R}_{I_r}$  和  ${}^l_i\mathbf{t}_{I_r}$  为参考块到目标块的变换位姿,  $I_r\mathbf{n}^T$  为参考块的相机系下的平面法向量,  $I_r\mathbf{p}$  为参考块的相机系下视觉地图点。

基于光度误差的法向量精细化 (优化):

$$I_r\mathbf{n}^* = \arg \min_{I_r\mathbf{n} \in \mathbb{S}^2} \sum_{i \in S} \sum_{j=1}^{N^2} \left\| \tau_i \mathbf{I}_i (\mathbf{A}_r^i \mathbf{u}_r^j) - \tau_r \mathbf{I}_r (\mathbf{u}_r^j) \right\|_2$$

计算参考块与每个目标帧对应块的每个像素值的光度误差之和。

优化问题的形式变换 (提高计算效率):

首先将涉及到法向量的项整理到一起, 设为  $M$ :

$$\mathbf{M} \triangleq \frac{1}{I_r\mathbf{n}^T \cdot I_r\mathbf{p}} I_r\mathbf{n} \in \mathbb{R}^3$$

可以看到待优化变量都在  $M$  上, 直接对  $M$  构建约束然后优化即可:

$$I_r\mathbf{p} \cdot \mathbf{M} = 1$$

(笔者注: 挺疑惑的, 这玩意直接整成闭式解的形式了? 若是笔者理解有误, 希望读者指出! 感谢!)

根据上面的约束可以获得  $M$  的具体形式:

$$\mathbf{M} = \begin{bmatrix} M_x \\ M_y \\ \frac{1}{I_r\mathbf{p}_z} - \frac{I_r\mathbf{p}_x}{I_r\mathbf{p}_z} M_x - \frac{I_r\mathbf{p}_y}{I_r\mathbf{p}_z} M_y \end{bmatrix} = \mathbf{B}\mathbf{m} + \mathbf{b}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{I_r\mathbf{p}_x}{I_r\mathbf{p}_z} & -\frac{I_r\mathbf{p}_y}{I_r\mathbf{p}_z} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{I_r\mathbf{p}_z} \end{bmatrix}, \mathbf{m} = \begin{bmatrix} M_x \\ M_y \end{bmatrix} \in \mathbb{R}^2$$

线性代数里的  $Ax+B=C$  的形式,  $m$  就是待求的  $x$ 。

求得  $m$  之后自然就得到法向量(笔者注: 得到的似乎是归一化后的法向量):

$$I_r\mathbf{n}^* = \frac{\mathbf{M}^*}{\|\mathbf{M}^*\|}, \mathbf{M}^* = \mathbf{B}\mathbf{m}^* + \mathbf{b}.$$

## 八、激光雷达测量模型

这一节是对第六节中激光雷达项的观测方程  $y_l = h_l(x, v_l)$  展开讨论，主要有两个小节：1、点到平面的观测模型；2、对激光雷达点不确定性的讨论。

### 1、点到平面的观测模型

基于先验状态  $(G\hat{\mathbf{T}}_I^\kappa)$  可以将激光雷达系下的点变换到世界系下：

$${}^G\hat{\mathbf{p}}_j^\kappa = {}^G\hat{\mathbf{T}}_I^\kappa {}^I\mathbf{T}_L {}^L\mathbf{p}_j$$

根据世界系下的点坐标找到其对应的八叉树节点（根/子/孙），没找到或对应节点不是平面就算了，否则就计算点到平面的距离：

$$\text{先讨论理想情况： } 0 = (\mathbf{n}_j^{gt})^T ({}^G\mathbf{T}_I^T {}^L\mathbf{p}_j^{gt} - \mathbf{q}_j^{gt})$$

然而受激光雷达测量噪声和平面不确定性影响，这个点和平面不可能是  $\mathbf{p}_j^{gt}$ 、 $\mathbf{n}_j^{gt}$  和  $\mathbf{q}_j^{gt}$ 。考虑这些扰动的影响（观测噪声  $\mathbf{v}_l = (\delta^L\mathbf{p}_j, \delta\mathbf{n}_j, \delta\mathbf{q}_j)$ ），实际上的观测方程如下：

$$\underbrace{0}_{y_l} = \underbrace{(\mathbf{n}_j \boxminus \delta\mathbf{n}_j)^T ({}^G\mathbf{T}_I^T {}^L\mathbf{p}_j - \mathbf{q}_j - \delta\mathbf{q}_j)}_{h_l(\mathbf{x}, \mathbf{v}_l)}$$

至此我们建立了关于观测噪声的观测方程，还记得第六节提到的观测系数阵  $\mathbf{H}$  和观测协方差阵  $\mathbf{R}$  嘛？现在可以求观测协方差阵  $\mathbf{R}$  了。（笔者笔者！那  $\mathbf{H}$  阵呢？不要急，如果将状态误差和观测噪声写到一起，然后同时求  $\mathbf{H}$  和  $\mathbf{R}$ ，会比较冗杂，笔者担心会让读者反而不易理解怎么求  $\mathbf{R}$  和  $\mathbf{H}$ 。）

为了可读性，笔者将观测方程中的  ${}^G\mathbf{T}_I^T {}^L\mathbf{p}_j - \delta^L\mathbf{p}_j$  写为  ${}^G\mathbf{p}_j - \delta^G\mathbf{p}_j$ 。我们在  $\mathbf{v}_l = (\delta^G\mathbf{p}_j, \delta\mathbf{n}_j, \delta\mathbf{q}_j)$  处对  $h_l(x, v_l)$  进行一阶泰勒展开（笔者注：多元函数的泰勒展开大家还记得么， $f(x^1, x^2, \dots, x^n) = f(x_k^1, x_k^2, \dots, x_k^n) + \sum_{i=1}^n (x^i - x_k^i) f'_{x^i}(x_k^1, x_k^2, \dots, x_k^n) + o^n$ ）：

$$\approx \underbrace{\mathbf{n}_j^T ({}^G\mathbf{p}_j - \mathbf{q}_j)}_0 + \underbrace{\mathbf{J}_{\mathbf{n}_j} \delta\mathbf{n}_j + \mathbf{J}_{\mathbf{q}_j} \delta\mathbf{q}_j + \mathbf{J}_{{}^G\mathbf{p}_j} \delta^G\mathbf{p}_j}_{\mathbf{v}_l}$$

关于观测的协方差阵  $\mathbf{R}$  可以写为：  $\Sigma_{\mathbf{v}_l} = \mathbf{J}_{\mathbf{v}_l} \Sigma_{\mathbf{n}_j, \mathbf{q}_j, {}^G\mathbf{p}_j} \mathbf{J}_{\mathbf{v}_l}^T$

其中：  $\mathbf{J}_{\mathbf{v}_l} = [\mathbf{J}_{\mathbf{n}_j}, \mathbf{J}_{\mathbf{q}_j}, \mathbf{J}_{{}^G\mathbf{p}_j}] = [({}^G\mathbf{p}_j - \mathbf{q}_j)^T, -\mathbf{n}_j^T, \mathbf{n}_j^T]$ ；

$$\Sigma_{\mathbf{n}_j, \mathbf{q}_j, {}^G\mathbf{p}_j} = \begin{bmatrix} \Sigma_{\mathbf{n}_j, \mathbf{q}_j} & 0 \\ 0 & \Sigma_{G\mathbf{p}_j} \end{bmatrix}$$

获得了观测协方差阵  $\mathbf{R}$  后，可以说怎么获得观测系数阵  $\mathbf{H}$  了。其实和观测协方差阵类似，观测协方差阵是对测量增加了一个扰动  $\mathbf{v}_l$ ，观测系数阵  $\mathbf{H}$  则是对状态增加一个扰动（误差） $\delta\mathbf{x}$ ，建立关于状态扰动  $\delta\mathbf{x}$  的观测方程  $\mathbf{h}(\mathbf{x} + \delta\mathbf{x}, 0)$ ，对其进行展开：

$$\mathbf{h}(\mathbf{x} + \delta\mathbf{x}, 0) = \mathbf{n}_j^T ({}^G\mathbf{R}_l \text{Exp}(\delta {}^G\mathbf{r}_l) ({}^l\mathbf{R}_L {}^L\mathbf{p}_j + {}^l\mathbf{t}_L) + {}^G\mathbf{t}_l + \delta {}^G\mathbf{t}_l - \mathbf{q}_j)$$

其中旋转比较麻烦，要对旋转求导得将其转为矢量形式，这里笔者套用罗德里格斯公式将旋转扰动进行一阶展开：

$$\begin{aligned} & {}^G\mathbf{R}_l \text{Exp}(\delta {}^G\mathbf{r}_l) {}^l\mathbf{p}_j \\ & \approx {}^G\mathbf{R}_l (\mathbf{I} + [\delta {}^G\mathbf{r}_l]_{\times}) {}^l\mathbf{p}_j \\ & = {}^G\mathbf{R}_l {}^l\mathbf{p}_j - {}^G\mathbf{R}_l [{}^l\mathbf{p}_j]_{\times} \delta {}^G\mathbf{r}_l \\ & {}^l\mathbf{p}_j = {}^l\mathbf{R}_L {}^L\mathbf{p}_j + {}^l\mathbf{t}_L \end{aligned}$$

带入到观测方程可得：

$$\begin{aligned} \mathbf{h}(\mathbf{x} + \delta\mathbf{x}, 0) & = \mathbf{n}_j^T ({}^G\mathbf{R}_l \text{Exp}(\delta {}^G\mathbf{r}_l) {}^l\mathbf{p}_j + {}^G\mathbf{t}_l + \delta {}^G\mathbf{t}_l - \mathbf{q}_j) \\ & = \mathbf{n}_j^T ({}^G\mathbf{R}_l {}^l\mathbf{p}_j - {}^G\mathbf{R}_l [{}^l\mathbf{p}_j]_{\times} \delta {}^G\mathbf{r}_l + {}^G\mathbf{t}_l + \delta {}^G\mathbf{t}_l - \mathbf{q}_j) \\ & = \mathbf{n}_j^T ({}^G\mathbf{R}_l {}^l\mathbf{p}_j + {}^G\mathbf{t}_l - \mathbf{q}_j) + \mathbf{n}_j^T (-{}^G\mathbf{R}_l [{}^l\mathbf{p}_j]_{\times} \delta {}^G\mathbf{r}_l + \delta {}^G\mathbf{t}_l) \\ & = \mathbf{n}_j^T ({}^G\mathbf{p}_j - \mathbf{q}_j) + \mathbf{n}_j^T [-{}^G\mathbf{R}_l [{}^l\mathbf{p}_j]_{\times} \quad \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 13}] \delta \mathbf{x} \end{aligned}$$

这样我们就获得了观测系数阵  $\mathbf{H} = \mathbf{n}_j^T [-{}^G\mathbf{R}_l [{}^l\mathbf{p}_j]_{\times} \quad \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 13}]$ 。

请读者把获得观测协方差阵和观测系数阵的过程串起来看，是不是也发现了华点：这不就是多元函数  $\mathbf{h}(\mathbf{x} + \delta\mathbf{x}, \mathbf{v}_l)$  对  $\delta\mathbf{x}$  和  $\mathbf{v}_l$  求偏导嘛！

## 2、对激光雷达点不确定性的讨论

在上面的激光雷达观测方程中，我们提到了激光雷达点的扰动（噪声） $\delta {}^L\mathbf{p}_j$ ，那么这个噪声到底是怎么引起的捏？

这个噪声主要来自三个方面：a. 激光飞行时间导致的测距不确定性  $\delta d$ ；b. 激光雷达轴承编码误差导致的发射方向不确定性  $\delta\omega$ ；c. 激光束的发散角  $\theta$ 。如下图：

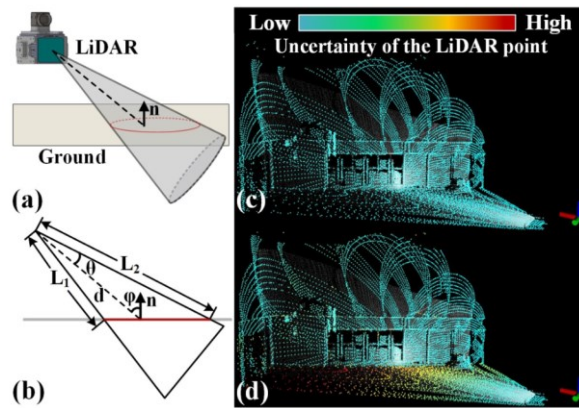


图 10 激光雷达发散角

这个图的意思是，除了激光飞行时间会导致 $\delta d$ ，激光束的发散角也对导致 $\delta d$ :

$$\delta d = L_2 - L_1 = d \left( \frac{\cos \varphi}{\cos (\theta + \varphi)} - \frac{\cos \varphi}{\cos (\theta - \varphi)} \right)$$

这个公式没有别的意思，就是告诉读者测距不确定性 $\delta d$ 与发散角 $\theta$ 关系，采数据时尽量避免激光雷达发散方向与平面太平行的动作。

## 九、视觉测量模型

这一节是对第六节中视觉项的观测方程 $y_c = h_c(x, v_c)$ 展开讨论，主要有两个小节：1、视觉地图点的选择；2、稀疏直接的视觉观测模型。

### 1、视觉地图点的选择

选择合适的视觉地图点对下面执行光度误差至关重要，（笔者注：好的视觉地图点拥有像素梯度显著的块，光度分布合适）。

选择的流程是：首先使用哈希表查询和体素光线投射把相机视野内的地图点提取出来（叫做视觉子地图），然后从这个视觉子地图选择视觉地图点并剔除离群点。

本小节有三点组成：a. 可视的体素查询；b. 按需的投射；c. 离群点剔除。

#### a. 可视的体素查询

虽说使用哈希表可以提高查询效率，但是对个体素地图进行查询仍是非常具有挑战性的，因此 Fast-livo2 通过将当前帧激光雷达点云转到世界系下，并用哈希查询这这些点的体素位置，快速获得候选体素（如果相机 FoV 与 LiDAR FoV 有很大重叠，则相机 FoV 中的视觉子地图点也基本位于这些体素中）。如果两帧图像之间视场重叠较大，也对上一帧图像中通过相同体素查询与光线投射判定为可见的地图点所占据的体素进行筛选。最终，通过提取这两类体素所包含的地图点，并经过视场范围验证后，获得当前视觉地图点。

#### b. 按需的投射

什么情况下需要？当雷达距离目标太近而没有点返回时或者摄像机 FOV 可能不会被激光雷达 FOV 完全覆盖。

笔者将其整理为 5 小点：i. 图像栅格划分；ii. 对未被占据的栅格进行射线投射；iii. 降低计算量；iv. 每个采样点处理；v. 所有未被占据栅格元处理完成。

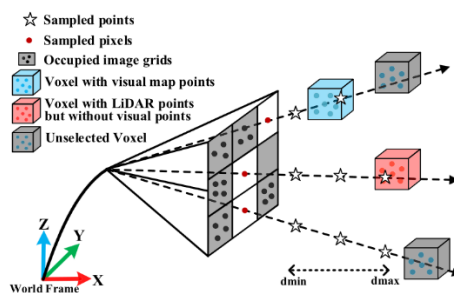


图 11 体素点投射

i. 图像栅格划分：将图像划分为  $30 \times 30$  的均匀栅格元，将体素查询获得的视觉地图点投影到栅格元上。

ii. 对未被占据的栅格进行射线投射：对于每个未被视觉地图点占据的栅格元，光线沿着中心像素向后投射，其中采样点沿着光线在深度方向上从  $d_{\min}$  到  $d_{\max}$  均匀分布；(笔者注：如果大家对相机系点投影到像素面的操作不陌生的话，这可以理解为逆操作，然后按照一定间隔设置深度值即可)

iii. 降低计算量：相机机体系下每个光线上的采样点的位置都被预先算好了。

iv. 每个采样点处理：对每个采样点，评估其对应的体素状态。若该体素包含位于此栅格内的地图点，则将这些地图点纳入视觉子地图，并终止当前射线的处理。否则：继续射线上的下一个采样点，直到达到最大深度  $D_{\max}$ 。

v. 所有未被占据栅格元处理完成：获得了一组在整个图像上分布的视觉地图点集合。每个栅格都有对应的视觉地图点。

### c. 离群点剔除

为什么要剔除？i. 在当前帧中被遮挡；ii. 存在深度不连续性；iii. 视觉地图点的参考块/当前帧采集视角过大——所有这些情况都会严重降低图像对齐精度。

#### i. 在当前帧中被遮挡

在雷达  $L10$  更新后，将视觉子地图中的所有视觉地图点用位姿投影至当前帧，并在每个栅格中保留深度最小的点。

#### ii. 存在深度不连续性

将当前雷达帧的点投影至当前帧生成深度图，用  $9 \times 9$  的图像块在深度图中比较视觉地图点的深度，并判断遮挡和深度变化（变化不能  $> 0.5\text{m}$ ）。被占据的/深度不连续的点将被剔除。

#### iii. 视觉地图点的参考块/当前帧采集视角过大

移除参考块与当前块的视角太大（法向量与从视觉地图点到块光学中心的方向之间的角度超过  $80^\circ$ ）的点。

至此，用于计算光度误差的点已全部筛选完毕。

## 2、稀疏直接的视觉观测模型

这一部分是稀疏直接法（笔者注：读过源码的读者或许会对源码中多出的两行有些疑惑，笔者在这里也会进行解释。）

在直接法之前，通常采用的是特征点匹配法，特征点匹配法是利用 ORB 等特征知道了两个对应点，然后用重投影误差来优化位姿（先给对应点，再优化位姿）。而直接法是不知对应点的，因此逻辑上是反过来的（先给位姿初值，然后求对应点，再优化位姿）：先给一个位姿初始值（LIO 提供的状态），当前图像基于这个位姿初始值估计它在另一个图像上的对应点，再计算两个点的光度误差来优化位姿（基于光度不变假设）。直接法就是遵循这样的思路得到的结果。

笔者想循序渐进的向大家介绍光度误差，因此先介绍简单的光度误差，再介绍 Fast-LIVO2 的光度误差，并评价两者异同。

### a. 简单的光度误差

（笔者注：推荐大家去学习高翔老师的直接法推导[博客](#)）

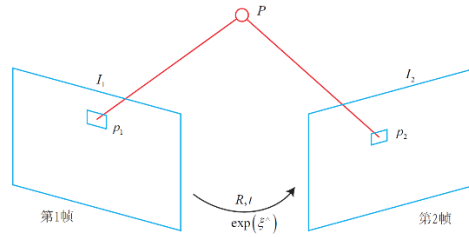


图 12 直接法示意

已知一个空间三维点  $P$ ，根据相机内参可以知道其在  $I_1$  上的像素位置  $p_1$ 。已知  $I_2$  相对  $I_1$  的位姿初始值  $(R, t)$ ，结合相机内参也可以知道  $P$  在  $I_2$  上的对应像素  $p_2$ （利用位姿将  $P$  转到  $I_2$  相机系下，再用内参投影）。

建立两个光度之间的误差（观测方程）： $e = I_1(p_1) - I_2(p_2)$

优化的目标函数： $\min_{\xi} J(\xi) = e^T e$

与 LIO 中的观测方程一样，我们关心观测量是如何随位姿扰动变化的（求对位姿扰动的雅可比）。考虑到位姿中的旋转，这里对旋转使用一阶近似将其转到  $R^n$  空间。

$$\begin{aligned} e(\xi \oplus \delta\xi) &= I_1(\pi(P)) - I_2(\pi(\exp(\delta\xi) \exp(\xi) P)) \\ &\approx I_1(\pi(P)) - I_2(\pi((1 + \delta\xi) \exp(\xi) P)) \\ &= I_1(\pi(P)) - I_2(\pi(\exp(\xi) P) + \pi(\delta\xi \exp(\xi) P)) \end{aligned}$$

记  $q = \delta\xi \exp(\xi) P$  (  $P$  在第二个相机坐标系下的坐标 ),  $u = \pi(q)$ 。其中  $\pi$  为投影方程, 将三维点投影至像素。

在扰动  $\delta\xi$  处进行一阶泰勒展开 (这里需要用的链式法则):

$$e(\xi \oplus \delta\xi) = I_1(\pi(P)) - I_2(\pi(\exp(\xi^\wedge) P)) - \frac{\partial I_2}{\partial u} \frac{\partial u}{\partial q} \frac{\partial q}{\partial \delta\xi} \delta\xi$$

其中:  $\frac{\partial I_2}{\partial u}$  为  $u$  处的像素梯度;  $\frac{\partial u}{\partial q}$  为投影方程关于相机坐标系下的三维点的导数;  $\frac{\partial q}{\partial \delta\xi} = [I, -q^\wedge]$  为变换后的三维点对扰动的导数,  $\wedge$  为向量求其反对称阵 ( 对扰动求导的原理建议学习《视觉 SLAM14 讲》的 4.3 节 )。

$$\frac{\partial u}{\partial q} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \end{bmatrix} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} \end{bmatrix}$$

至此, 光度误差关于位姿扰动的雅可比阵就求得了  $H = \frac{\partial I_2}{\partial u} \frac{\partial u}{\partial q} \frac{\partial q}{\partial \delta\xi}$

## b. Fast-LIVO2 的光度误差

直接贴出 Fast-LIVO2 的光度误差公式:

$$0 = \tau_k \mathbf{I}_k^{gt} \underbrace{(\pi({}^C \mathbf{T}_l ({}^G \mathbf{T}_l)^{-1} \mathbf{p}_i) + \Delta \mathbf{u})}_{\mathbf{u}_i} - \tau_r \mathbf{I}_r^{gt} \underbrace{(\pi({}^C \mathbf{T}_G^r \mathbf{p}_i) + \mathbf{A}_i^r \Delta \mathbf{u})}_{\mathbf{u}_i'}$$

可以看到 Fast-LIVO2 的光度误差公式比上面讨论的似乎多了点东西。

首先是  $\Delta \mathbf{u}$ 、 $\mathbf{A}_i^r \Delta \mathbf{u}$ 。还记得前面提及的图像块么, Fast-LIVO2 不是只计算视觉地图点对应像素的光度误差, 而是以视觉地图点投影像素为中心的  $8 \times 8$  像素块的光度误差,  $\Delta \mathbf{u}$  就是中心到块中像素的偏移,  $\mathbf{A}_i^r \Delta \mathbf{u}$  中心到参考块中像素的偏移。并不影响雅可比的计算。

然后是  ${}^C \mathbf{T}_l$ 。这也是代码中多了两行的原因:

关于Fast-LIVO2中直接法的雅可比矩阵与十四讲中直接法的对应关系

问题: 什么Fast-LIVO2比十四讲多了最后两行代码, 怎么推导的?

FAST-LIVO2

```
void VIOManager::updateState(cv::Mat img, int level)
{
    Jimg << du, dv;
    Jimg = Jimg * state->inv_expo_time;
    Jimg = Jimg * inv_scale;
    Jdphi = Jimg * Jdpi * p_hat;
    Jdp = -Jimg * Jdpi;
    JdR = Jdphi * Jdphi_dR + Jdp * Jdp_dR;
    Jdt = Jdp * Jdp_dt;

    Jdphi_dR = Rci; Jdp_dR = -Rci * tmp;
    tmp << SKEW_SYM_MATRX(Pic);
    Jdp_dt = Rci * Rwi.transpose();

    Rci和Pic为imu与相机系的外参。
}
```

十四讲

1.  $\frac{\partial I_2}{\partial u}$  为  $u$  处的像素梯度;
2.  $\frac{\partial u}{\partial q}$  为投影方程关于相机坐标系下的三维点的导数。
3.  $\frac{\partial q}{\partial \delta\xi}$  为变换后的三维点对变换的导数,  $\frac{\partial q}{\partial \delta\xi} = [I, -q^\wedge]$

$\mathbf{J} = -\frac{\partial I_2}{\partial u} \frac{\partial u}{\partial \delta\xi}$

?

回看一下 a. 简单的光度误差中的参考系，我们发现所求的位姿扰动是相机系下的，即  $T_{c-w}$ 。而 FAST-LIVO2 中的位姿扰动是 imu 相对世界系  $T_{w-i}$ ，所以前五行代码实现的是光度误差对  $T_{c-w}$  扰动的雅可比，不是对  $T_{w-i}$  扰动的。

在李群 SE(3) 中，要将一个李代数扰动从一个坐标系转换到另一个坐标系需要一个伴随矩阵。直接用 Deepseek 计算：

$$\delta T_{c,w} = \text{Ad}_{T_{c,i}}(\delta T_{w,i})$$

$$\text{Ad}_{T_{c,i}} = \begin{bmatrix} R_{c,i} & 0 \\ [t_{c,i}]^\times R_{c,i} & R_{c,i} \end{bmatrix}$$

其实可以从链式法则的角度去想： $T_{c-w}$  扰动是关于  $T_{w-i}$  扰动的函数，因此根据链式法制，再求  $T_{c-w}$  扰动对  $T_{w-i}$  扰动的偏导，结果也是这个伴随矩阵。最后两行就是做了这么一个事。

因此，在正常求完光度误差的雅可比后，再乘以该伴随矩阵，就是对  $T_{w-i}$  扰动的雅可比阵了！（笔者注：FAST-LIVO2 采用的是多层直接法，会在每个金字塔层上执行观测的雅可比计算并进行 VIO 更新。）

## 十、结语

关于实验部分，笔者就不解析啦，直接翻译就能看懂。感谢所有能读到这里的读者，也非常感谢郑纯然博士及其所在团队开源的如此优秀的工作！

祝大家越来越强，咱们代码解析再见！

哈尔滨工程大学-智能科学与工程学院

潘峰

2025.05.29