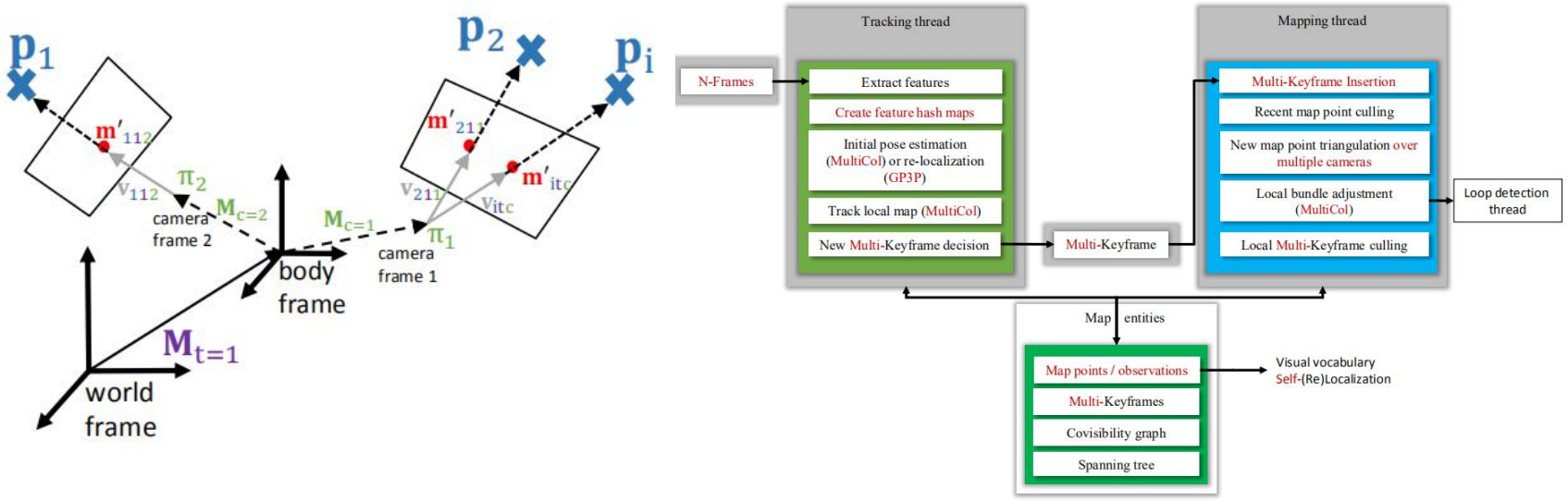
CloudMinds

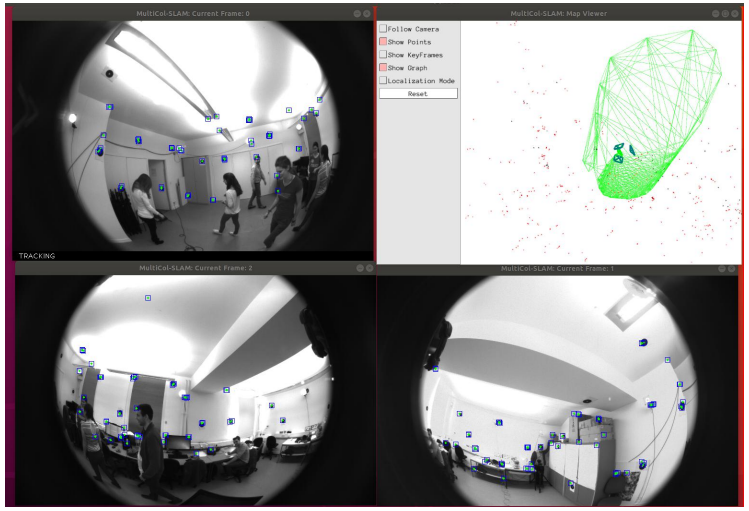
多相机SLAM方案

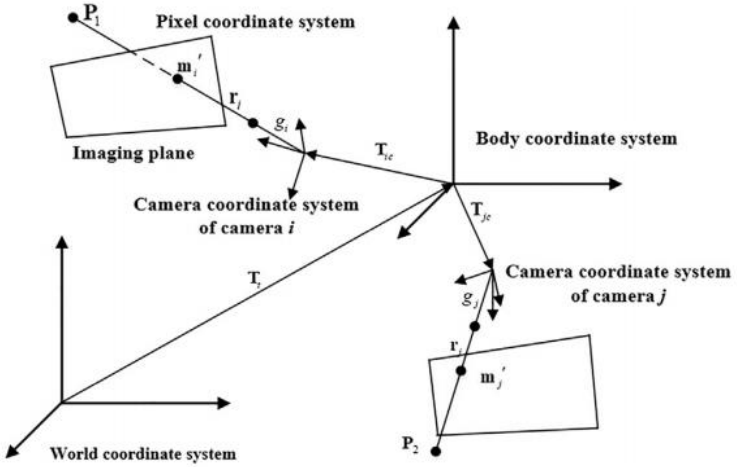
AI-SmartVision-SLAM

2021.2.5

0. 论文方案（多是基于ORB框架和PTAM框架的特征点法）







多相机构建的单帧

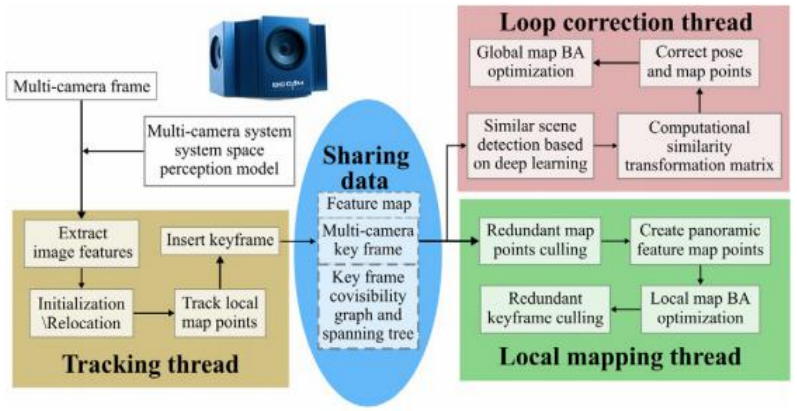


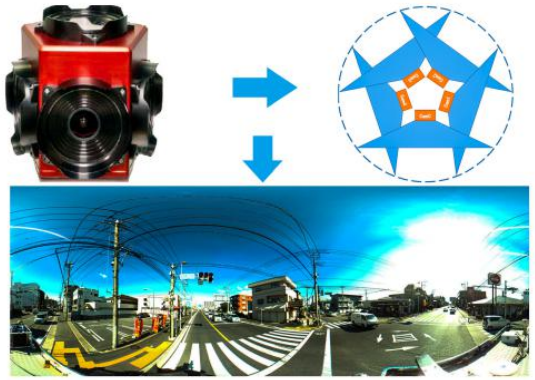
Fig. 3. Complete framework of Multi-camera SLAM system.

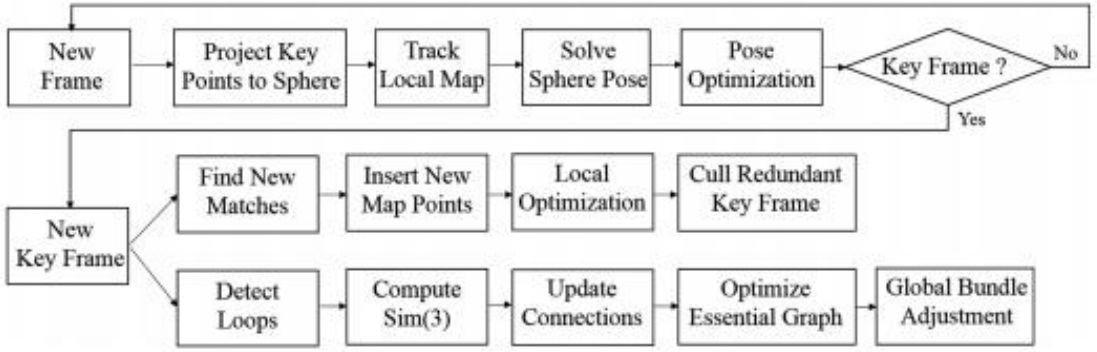
系统流程图

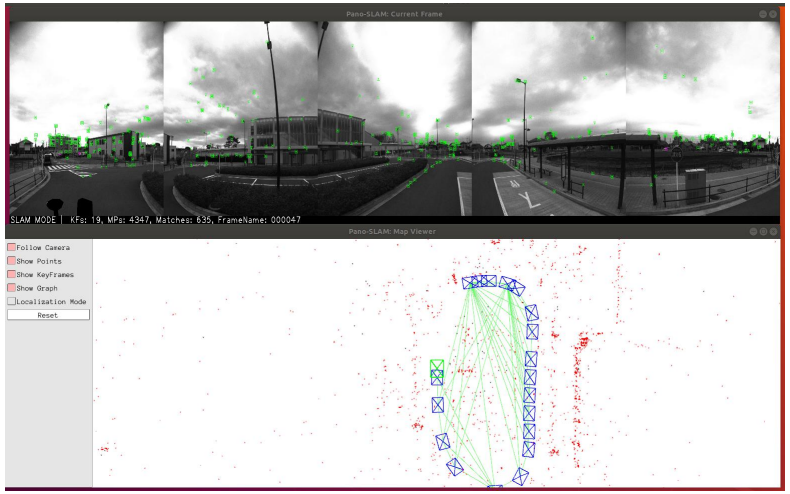
Fig. 7. The vehicle used for collecting data with a panoramic camera and an integrated navigation system.

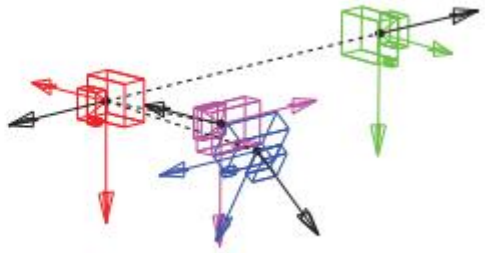
实物图

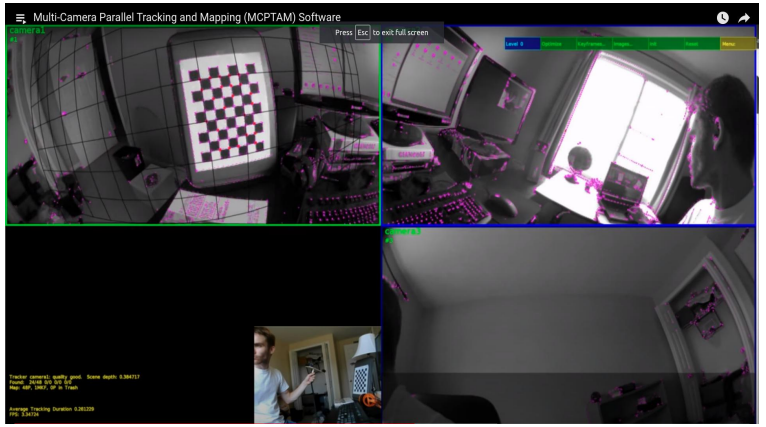
0. 论文方案（多是基于ORB框架和PTAM框架的特征点法）







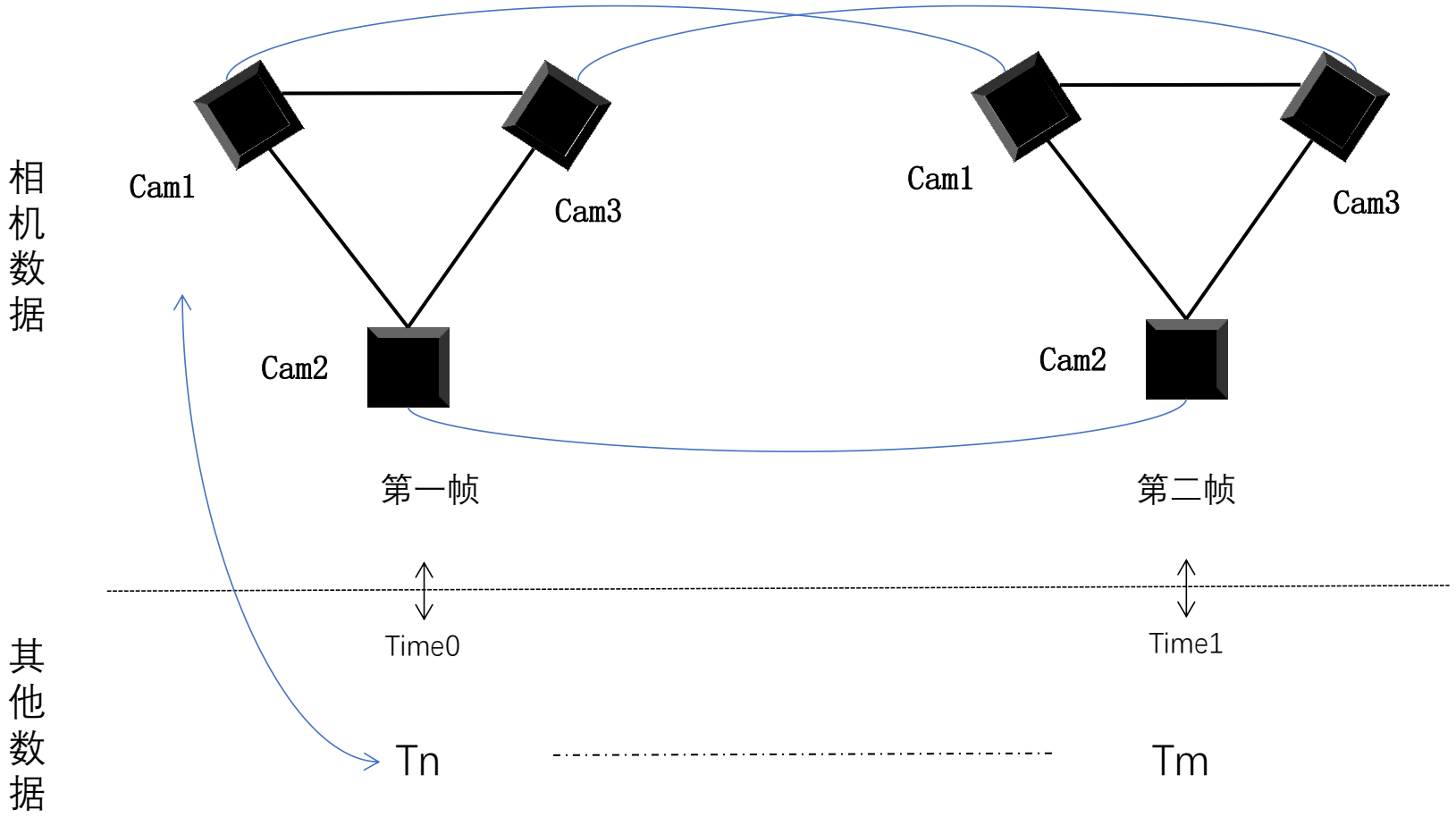




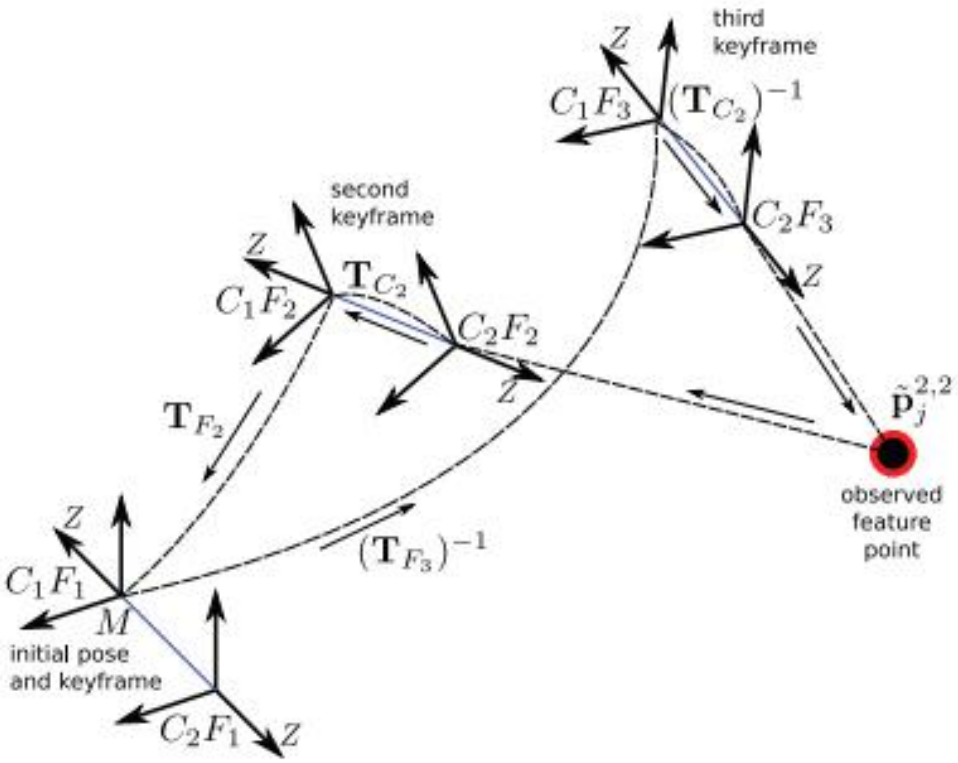
1.初始化

a.尺度问题
b.位姿对齐问题
c. 初始化是否成功

跟踪过程相同



2. 多相机BA优化



以两个相机为例可以拓展为n个相机

观测量：

$$\mathbf{g}_j^{i,\ell}(\mathbf{x}) = \kappa_i\left(\pi_3\left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j^{h,k}\right)\right)$$

待优化量：

$$\delta = \begin{bmatrix}\delta_{F_1}^\top & \dots & \delta_{F_{n_k}}^\top & \delta_{P_1}^\top & \dots & \delta_{P_{n_f}}^\top\end{bmatrix}^\top \in \mathbb{R}^{(6n_k+3n_f)}$$

扰动：

$$\tilde{\mathbf{p}}_j^{i,\ell} = \mathbf{T}_{C_i}^{-1}(\mathbf{T}_{F_\ell}\boxplus_{\mathcal{F}}\delta_{F_\ell})^{-1}(\mathbf{T}_{F_k}\boxplus_{\mathcal{F}}\delta_{F_k})\mathbf{T}_{C_h}\begin{bmatrix}\left(\mathbf{p}_j\boxplus_{\mathcal{P}}\delta_{P_j}\right)^\top & 1\end{bmatrix}^\top$$

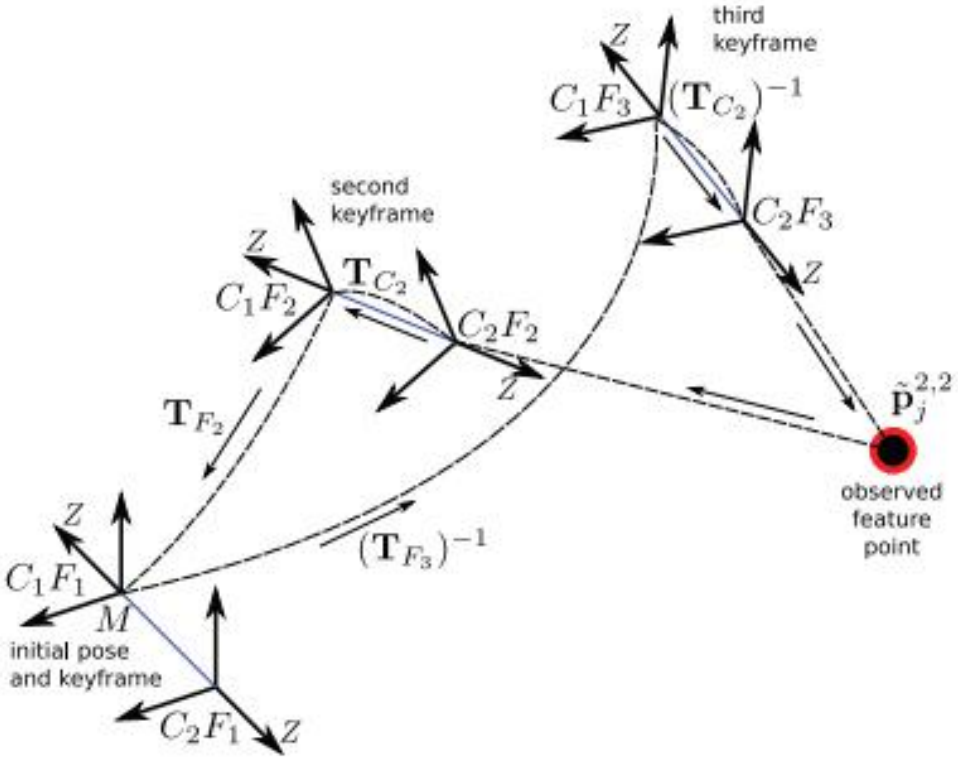
误差对状态量的导数（两部分）：

$$\mathbf{J}_j^{i,\ell} = \left.\frac{\partial \mathbf{g}_j^{i,\ell}(\tilde{\mathbf{x}}\boxplus_{\mathcal{S}}\delta)}{\partial\delta}\right|_{\delta=\mathbf{0}} = \mathbf{H}_j^{i,\ell}\mathbf{G}_j^{i,\ell}$$

where

$$\mathbf{H}_j^{i,\ell} = \left.\frac{\partial\kappa_i(\pi_3(\tilde{\mathbf{p}}_j^{i,\ell}))}{\partial\tilde{\mathbf{p}}_j^{i,\ell}}\right|_{\delta=\mathbf{0}} \quad \text{and} \quad \mathbf{G}_j^{i,\ell} = \left.\frac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\delta}\right|_{\delta=\mathbf{0}}$$

2. 多相机BA优化



以两个相机为例可以拓展为n个相机

导数中的非零块：

$$\mathbf{G}_j^{i,\ell} = \begin{bmatrix} \mathbf{0} & \frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \delta_{F_k}} & \mathbf{0} & \dots & \mathbf{0} & \frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \delta_{F_\ell}} & \mathbf{0} & \dots & \mathbf{0} & \frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \delta_{P_j}} & \mathbf{0} \end{bmatrix}$$

主帧和观测帧导数：

$$\frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \delta_{F_k}} = \left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\right) \frac{\partial \mathbf{T}_{\mathcal{F}}(\delta_{F_k})}{\partial \delta_{F_k}} \left(\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j\right)$$

$$= (\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1} \begin{bmatrix} \mathbf{I}_{3\times 3} & -\left[\mathbf{p}_j^M\right]_\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}$$

$$\frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \delta_{F_\ell}} = \left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\right) \frac{\partial \left(\mathbf{T}_{\mathcal{F}}(\delta_{F_\ell})^{-1}\right)}{\partial \delta_{F_\ell}} \left(\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j\right.$$

$$= \left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\right) \frac{\partial \mathbf{T}_{\mathcal{F}}(-\delta_{F_\ell})}{\partial \delta_{F_\ell}} \left(\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j\right)$$

$$= (\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1} \begin{bmatrix} -\mathbf{I}_{3\times 3} & \left[\mathbf{p}_j^M\right]_\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}$$

3d点导数：

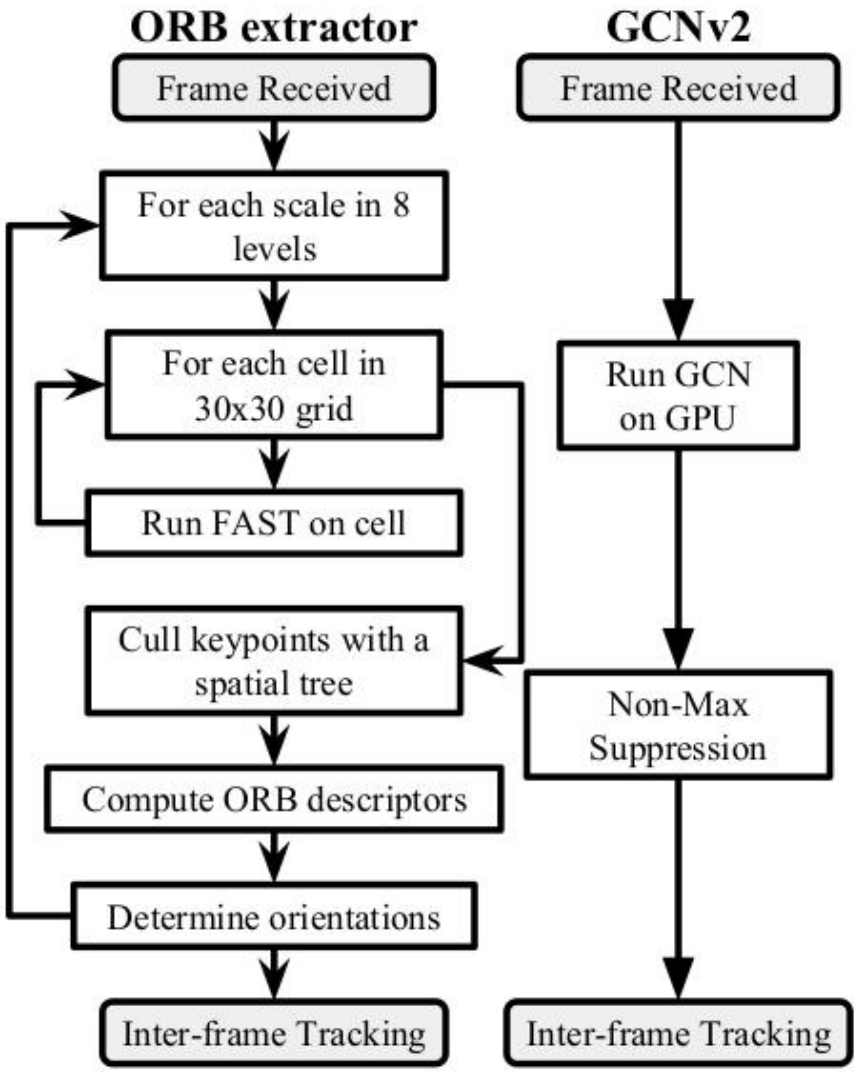
$$\frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \delta_{P_j}} = \|\mathbf{p}_j\|(\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\mathbf{T}_{F_k}\mathbf{T}_{C_h} \begin{bmatrix} \mathcal{R}P_j(\mathbf{p}_j) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{0}_{1\times 3} \end{bmatrix}$$

3. GCNv2: Efficient Correspondence Prediction for Real-Time SLAM

论文声称能够在嵌入式平台实时运行

此代码同样基于ORBSLAM(2019年开源

参考之前基于ORBSLAM框架的多目SLAM是否能够基于此代码？
速度比supperPoint要快



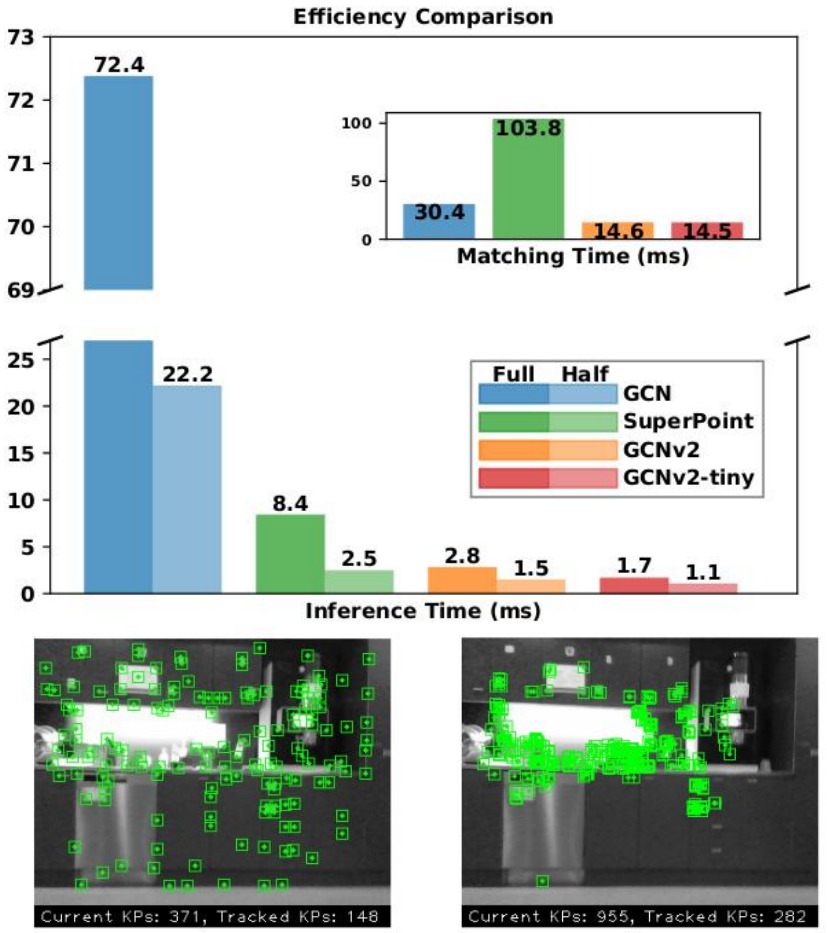
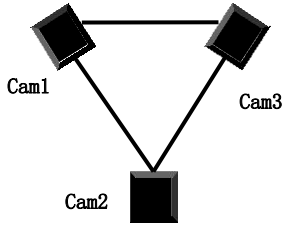


Fig. 1: The top figure shows our drone preforming position hold using GCN-SLAM. The figures below show the intermediate output for comparison of binary features, ORB and GCNv2, in ORB-SLAM2 and GCN-SLAM respectively. GCNv2 (left) tends to predict more repeatable and evenly distributed features compared with ORB (right.)

3. 基于多相机的DSO(参考前面的基于ORBSLAM框架的)

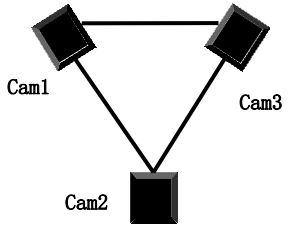
CloudMinds

框架不变更多的数据的处理



第一帧

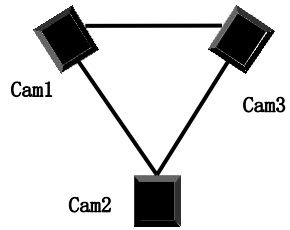
提取各个图像中的梯度点构建makeKNN(),潜在的成熟点和未成熟点（开始之前是不是可以做一下弱纹理检测？）.



第二帧

生成成熟点构建残差·（选择成熟点个数多的那个作为主导的camera?)

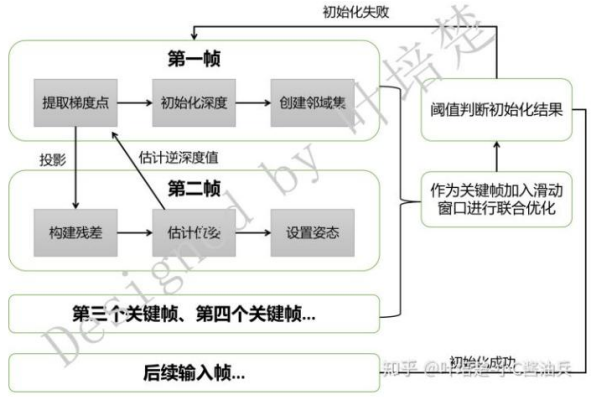
运动跟踪尝试多种运动模型

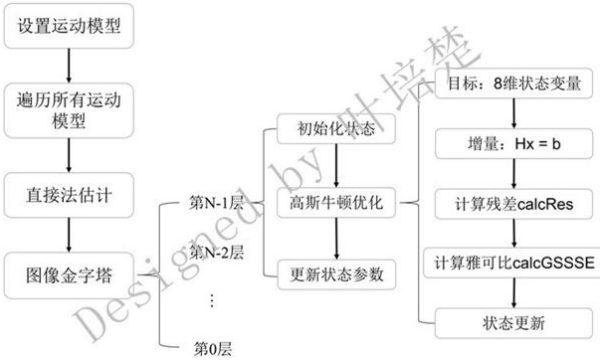


第n帧

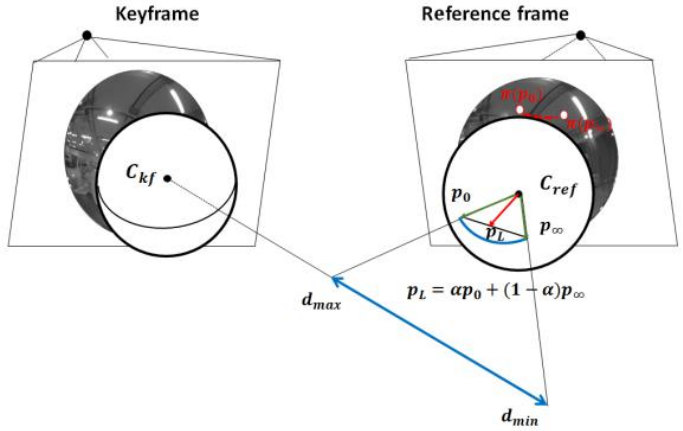
makeNoKeyFrame
makeKeyFrame

后端窗口优化以及边缘化





参考单目鱼眼DSO需要注意的是：投影到归一化球面用到对极曲线



深度滤波极限搜索会有不同

个人理解：多目的SLAM大体框架不会变动主要是数据关联以及求导

多目对尺度漂移的影响

零空间

所谓零空间，在数学上是指方程

$$\mathbf{H}\mathbf{x} = \mathbf{0} \tag{1}$$

的解所形成的空间$\{\mathbf{x}|\mathbf{H}\mathbf{x} = \mathbf{0}\}$。很显然，当矩阵$\mathbf{H}$的行列式等于0时，方程有无数解，这些解就构成了所谓的零空间。
在DSO中求解的增量方程，由于绝对尺度这个信息的缺失，对于同一个增量方程，显然有不同的状态量$\mathbf{x}_1, \mathbf{x}_2$满足

$$\mathbf{H}\mathbf{x}_1 = \mathbf{b} \tag{2}$$

$$\mathbf{H}\mathbf{x}_2 = \mathbf{b} \tag{3}$$

两式一减，就得到了公式（1）的形式：

$$\mathbf{H}(\mathbf{x}_2 - \mathbf{x}_1) = \mathbf{0} \tag{4}$$

DSO采用的滑动窗口法进行优化，当前面的关键帧离开滑动窗口后，最多只能通过少量的边缘化的点提供一定的先验信息。在某些情况下比如转弯过快等，关键帧生成速度很快，前面的帧来不及留下足够多的先验信息时（因为前面帧的点大多投影不到现在的帧上，构不成残差项）

DSO中的两个消除方法：FEJ和增量正交化

图1　相机一个急转弯后尺度突然坍塌

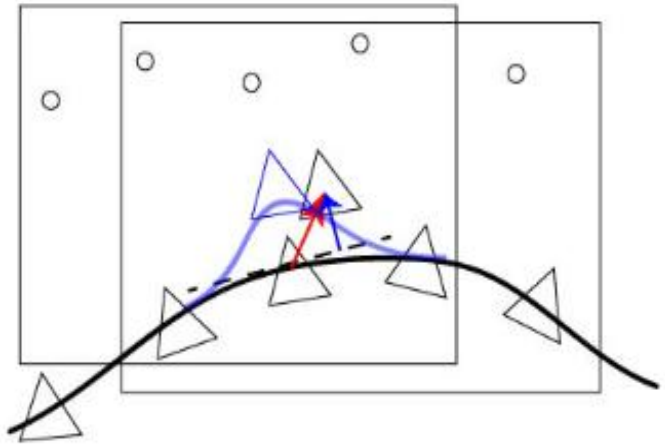


图2 增量正交化来缓解零空间漂移

尺度问题：是不是能够做一下相机和激光的标定，利用激光数据作为两帧之间的一个先验

多目提供更多的先验信息一定程度消除漂移

1. 参考资料以及其它问题

CloudMinds

1. Multi-Camera Tracking and Mapping for Unmanned Aerial Vehicles in Unstructured Environments (2015)

2. Multi-camera parallel tracking and mapping with non-overlapping fields of view（2014）

3. MULTICOL-SLAM – A MODULAR REAL-TIME MULTI-CAMERA SLAM SYSTEM (2016)

4.Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles (2017)

5.Panoramic SLAM from a multiple fisheye camera rig (2020)

6.Multi-camera visual SLAM for off-road navigation（2020）

多相机标定： 现在使用的标定工具不能支持多个相机一点公共视角都没有的情况

相机同步： 有一篇论文中稍微提了一下(硬件同步和软件同步)

依托框架： 实际操作在那种方法上改

效率问题： 要求的帧率

CloudMinds

THANKS

Operating Smart Robots for People