

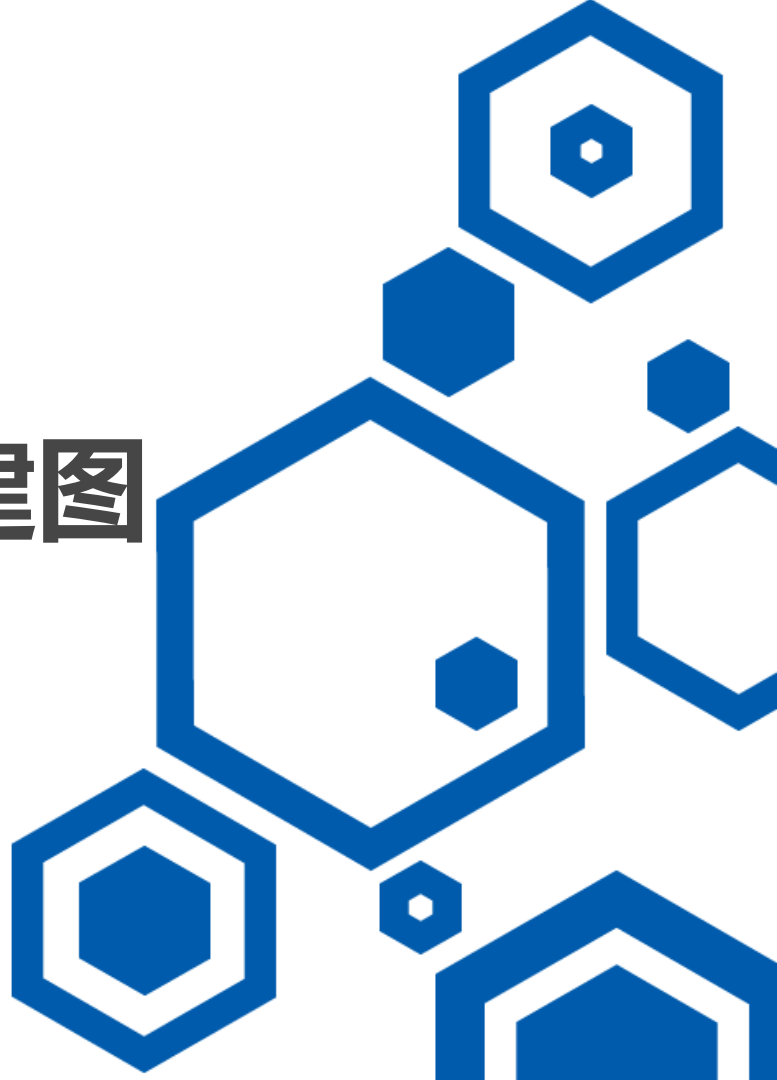
DSO源码解析

——跟踪与建图



主讲人 龚益群

东北大学





预备知识



初始化



跟踪



建图



滑窗优化



总结



跟踪 & 建图



1、运动假设



2、激活点跟踪



3、关键帧筛选



4、候选点跟踪



5、候选点激活



6、总结



跟踪



1、运动假设



2、激活点跟踪



3、关键帧筛选



4、候选点跟踪



5、候选点激活



6、总结

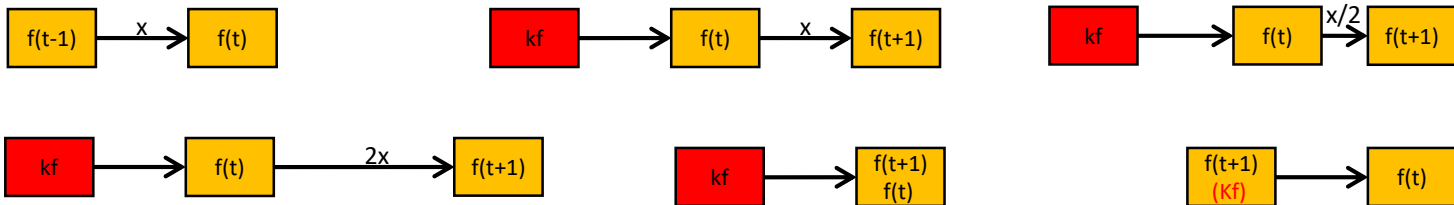


1、运动假设

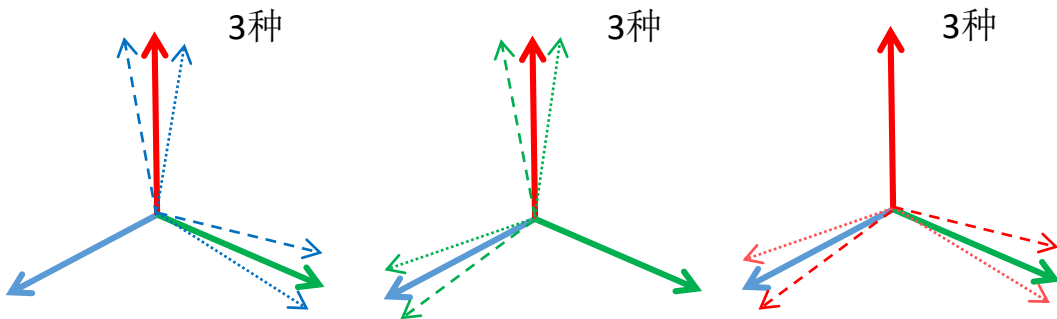
运动假设

● 对最新帧的运动模型进行假设

- 分别假设5种：匀速、半速、倍速、零速、以及从参考帧静止



- 假设匀速基础上有26种旋转运动，例：Quaterniond(1,rotDelta,0,0)





跟踪



1、运动假设



2、激活点跟踪



3、关键帧筛选



4、候选点跟踪



5、候选点激活



6、总结

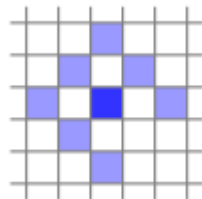
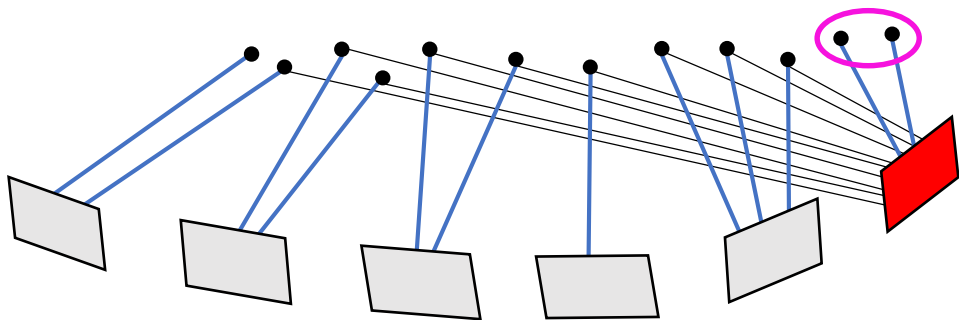


2、激活点跟踪

生成跟踪点

● 生成用于跟踪的半稠密深度地图

- **投影**：将最新关键帧设置为参考帧，并将激活点向最新关键帧投影，并且在金字塔从下向上使用协方差加权生成逆深度。
- **膨胀**：对于0-1层中没有逆深度的点，使用45度的方向点来计算。对于2-4层，使用上下左右的点来进行计算。
- **计算方法**：高斯归一化积，使用协方差加权。



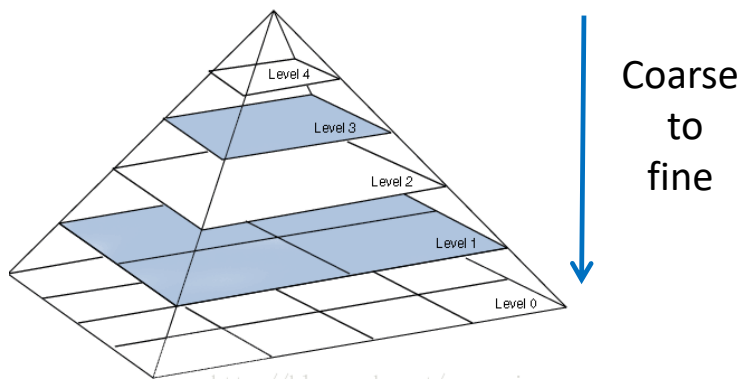


2、激活点跟踪

图像对齐

● 粗到精的跟踪策略

- 从顶层到底层，如果大于能量阈值的点超过60%，则放大外点阈值，然后使用LM进行优化，且该层多优化一遍。
- 如果某一层的能量值大于1.5倍最小值，直接判断为失败，节省时间。
- 如果跟踪成功且第0层好于当前，则保留结果，并更新每一层的最小能量值。
- 第0层最小值小于阈值则停止，把目前最好的值作为下次跟踪的阈值。





2、激活点跟踪

图像对齐

● 求导的参数包括：

- j帧的光度参数
- j帧位姿

● 对最新帧位姿求导

- 与初始化中位姿导数相同

$$(d_x f_x \quad d_y f_y) \begin{pmatrix} \frac{d_{\mathbf{p}_i}}{P'_z} & 0 & -\frac{d_{\mathbf{p}_i}}{P'_z} \frac{P'_x}{P'_z} & -\frac{P'_x P'_y}{P'_z} & 1 + \frac{P_x'^2}{P_z'^2} & -\frac{P'_y}{P'_z} \\ 0 & \frac{d_{\mathbf{p}_i}}{P'_z} & -\frac{d_{\mathbf{p}_i}}{P'_z} \frac{P'_y}{P'_z} & -1 - \frac{P_y'^2}{P_z'^2} & \frac{P'_x P'_y}{P_z'^2} & \frac{P'_y}{P'_z} \end{pmatrix}$$

● 对光度参数求导

$$r_k = I_j[\mathbf{p}_j] - \frac{t_j e^{a_j}}{t_i e^{a_i}} I_i[\mathbf{p}_i] - \left(b_j - \frac{t_j e^{a_j}}{t_i e^{a_i}} b_i \right)$$

$$\begin{aligned} J_{\text{photo}} &= \frac{\partial r_k}{\partial \delta_{\text{photo}}} \\ &= \left(\frac{\partial r_k}{\partial \delta a_j} \quad \frac{\partial r_k}{\partial \delta b_j} \right) \\ &= \left(-\frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[\mathbf{p}_i] - b_i) \quad -1 \right) \end{aligned}$$

\uparrow
 a
 \uparrow
 b_0



跟踪



1、运动假设



2、激活点跟踪



3、关键帧筛选



4、候选点跟踪



5、候选点激活



6、总结



3、关键帧筛选

光流判断关键帧

● 光流大小判断关键帧

- 当视角变化时，创建关键帧。使用初始的粗跟踪中流的平均平方来计算

$$f := \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{p} - \mathbf{p}'\|^2 \right)^{\frac{1}{2}} \text{—— 旋转平移}$$

- 相机移动导致的遮挡或去遮挡，需要更多的关键帧（即使f很小），使用没有旋转的平均光流计算

$$f_t := \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{p} - \mathbf{p}'_t\|^2 \right)^{\frac{1}{2}} \text{—— 只有平移}$$

- 曝光时间变化明显的，创建新的关键帧。通过相对亮度因子测量

$$a := |\log(e^{a_j - a_i} t_j t_i^{-1})|$$



预备知识



初始化



跟踪



建图



滑窗优化



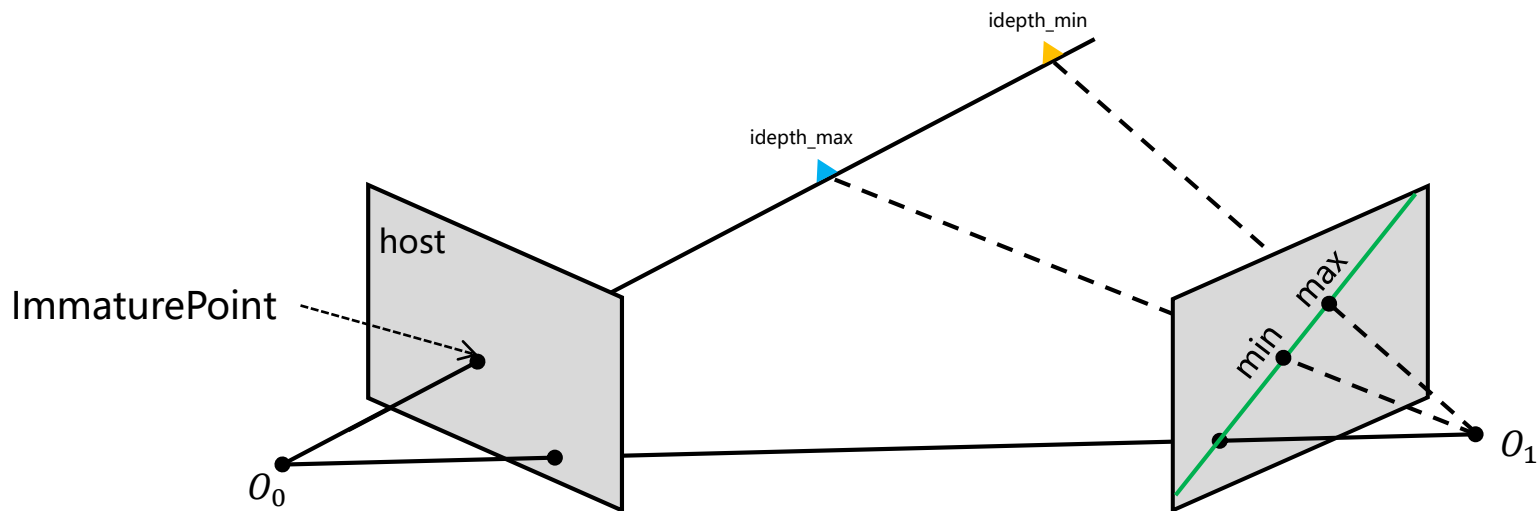
总结

- 1、运动假设
- 2、激活点跟踪
- 3、关键帧筛选
- 4、候选点跟踪**
- 5、候选点激活
- 6、总结



● 投影计算极线范围

- 对于有depth_max直接投影
- 对于没有depth_max会给定一个范围





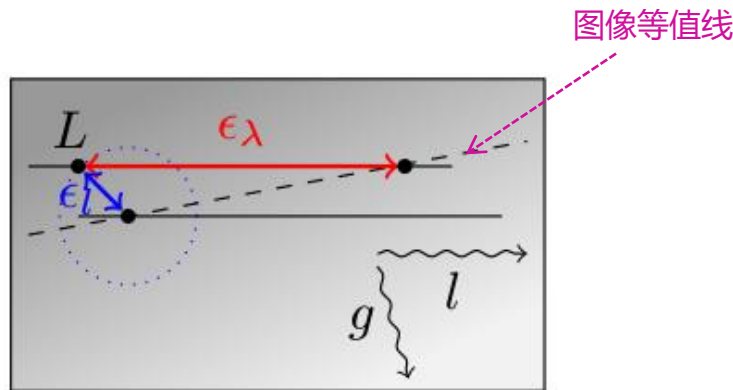
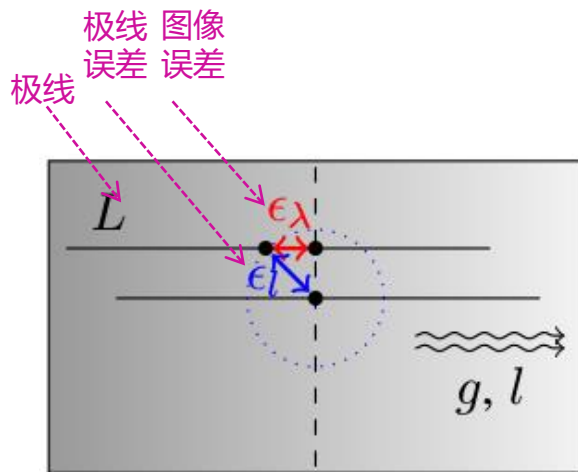
4、候选点跟踪

判断误差

● 计算极线搜索的误差

- 像素梯度和极线方向夹角越大得到的图像误差会越大，极线搜索的效果也不好

$$l_0 + \lambda^* \begin{pmatrix} l_x \\ l_y \end{pmatrix} \stackrel{!}{=} g_0 + \gamma \begin{pmatrix} -g_y \\ g_x \end{pmatrix}, \quad \gamma \in \mathbb{R}$$



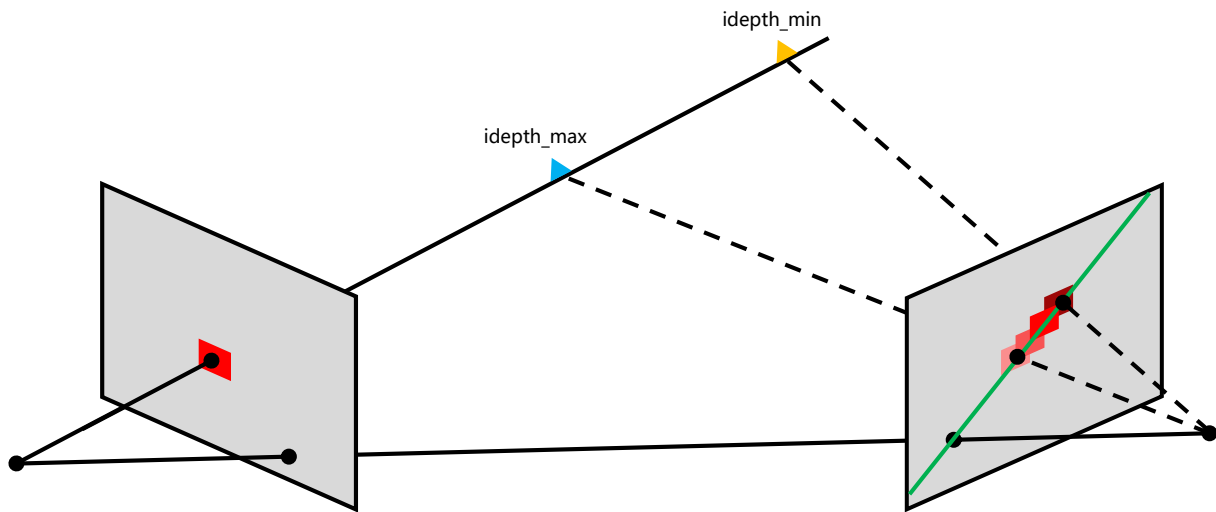


4、候选点跟踪

极线搜索1

● 粗略的极线搜索策略

- 在极线上按照一定步长移动，找到像素差最小的位置和第二小的位置，二者比值来评定质量



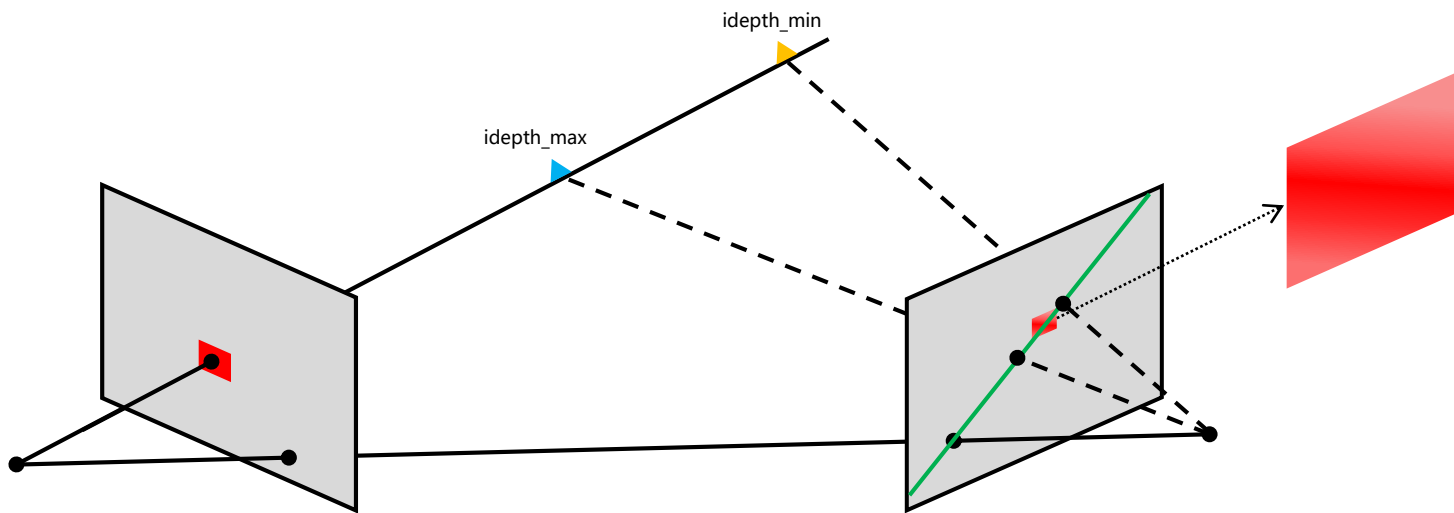


4、候选点跟踪

极线搜索2

- 精确极线搜索策略

- 以上一步的结果为初值进行GN优化, Jacobian是图像梯度在极线上的投影。





4、候选点跟踪

求逆深度范围

● 得到匹配位置后，计算逆深度范围

- 匹配像素点和逆深度的关系如下

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \Pi_c \left(KR \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + Kt \cdot d_p \right)$$

$$= \Pi_c \left(\begin{bmatrix} pr_x \\ pr_y \\ pr_z \end{bmatrix} + d_p \cdot \begin{bmatrix} kt_x \\ kt_y \\ kt_z \end{bmatrix} \right)$$

- 将矩阵形式展开成方程组

$$u' = \frac{pr_x + d_p \cdot kt_x}{pr_z + d_p \cdot kt_z}$$

$$v' = \frac{pr_y + d_p \cdot kt_y}{pr_z + d_p \cdot kt_z}$$

- 将矩阵形式展开成方程组

$$d_p = \frac{u' \cdot pr_z - pr_x}{kt_x - u' \cdot kt_z}$$

$$d_p = \frac{v' \cdot pr_z - pr_y}{kt_y - v' \cdot kt_z}$$

- 给匹配点加上误差

$$d_{pmin} = \frac{(v' - error \cdot d_y) \cdot pr_z - pr_y}{kt_y - (v' - error \cdot d_y) \cdot kt_z} \quad d_{pmin} = \frac{(u' - error \cdot d_x) \cdot pr_z - pr_x}{kt_x - (u' - error \cdot d_x) \cdot kt_z}$$

$$d_{pmax} = \frac{(v' + error \cdot d_y) \cdot pr_z - pr_y}{kt_y - (v' + error \cdot d_y) \cdot kt_z} \quad d_{pmax} = \frac{(u' + error \cdot d_x) \cdot pr_z - pr_x}{kt_x - (u' + error \cdot d_x) \cdot kt_z}$$

- 1、运动假设
- 2、激活点跟踪
- 3、关键帧筛选
- 4、候选点跟踪
- 5、候选点激活**
- 6、总结

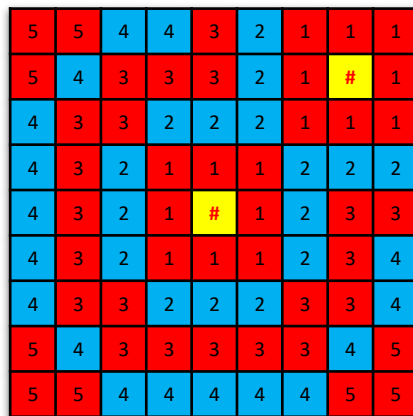
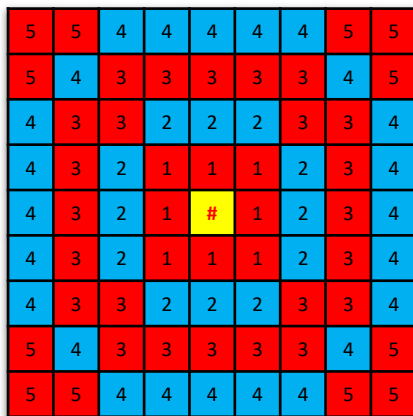


5、候选点激活

距离地图

- 计算图像上每个像素到激活像素点的距离

- 在最新关键帧的金字塔第一层上计算
- 点附近的一层为1，第二层为2，.....以此类推
- 目的是计算每个像素到最近的激活像素点的距离





5、候选点激活

候选点选择

- 将符合激活条件的未成熟点挑选出来
 - 点被更新过且不是外点
 - 上一次更新得到的像素误差区间小于8
 - 在粗略的极线搜索过程中得到的质量，即次小误差比最小误差比值大于3
 - 最大最小逆深度的和为正数
 - 向距离地图上投影，距离数值大于阈值，该阈值和已激活点数目和点类型有关
 - 不满足要求的候选点就删除



5、候选点激活

Structure优化

- 使用LM算法优化选出的候选点逆深度

- 将候选点向滑窗内关键帧投影，构建残差，计算正规方程来优化逆深度
- 与初始化中逆深度导数相同

$$\frac{1}{P'_z} d_x f_x \left(t_x - \frac{P'_x}{P'_z} t_z \right) + \frac{1}{P'_z} d_y f_y \left(t_y - \frac{P'_y}{P'_z} t_z \right)$$

- 要保证能量下降，并且逆深度的协方差足够小，至少在一帧上的残差比较小
- 然后将计算得到的点构造成为PointHessian

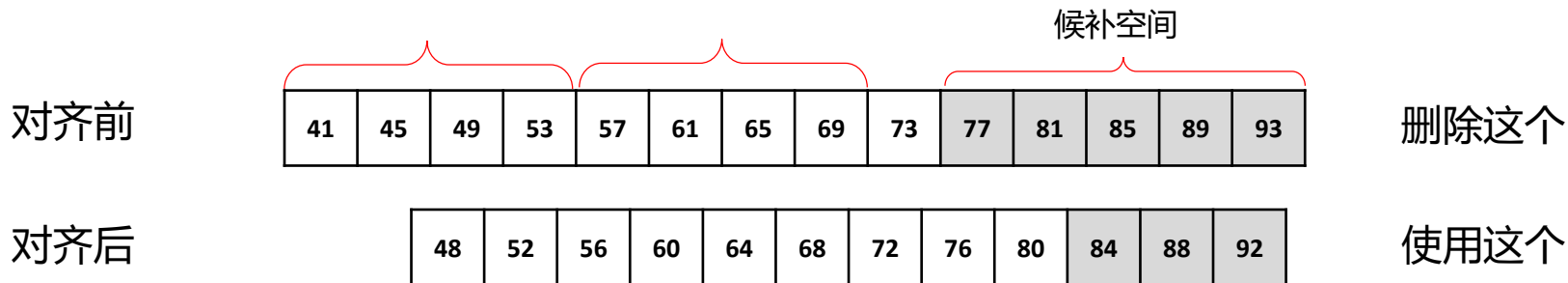


5、候选点激活

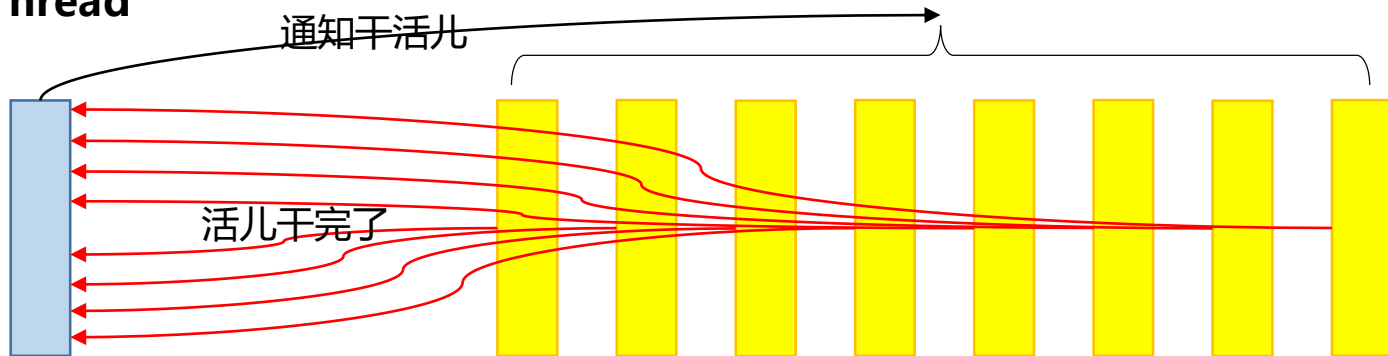
加速方法

● SSE

- **加速原理：**16个字节（128位）的数据可以同时计算，如4个float（4字节）



● Multi-Thread





- Code

CoarseTracker.cpp

ImmaturePoint.cpp

FullSystem.cpp

FullSystemOptPoint.cpp

IndexThreadReduce.h



跟踪 & 建图



1、运动假设



2、激活点跟踪



3、关键帧筛选



4、候选点跟踪



5、候选点激活



6、总结



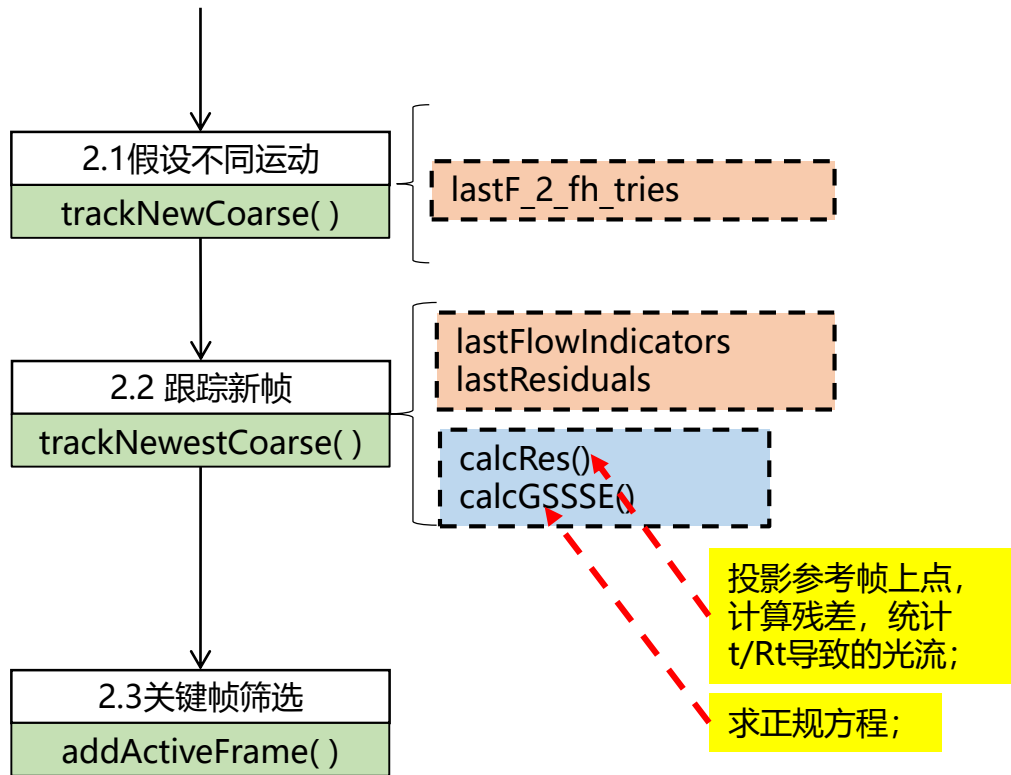
总结

跟踪

- 分别假设**5种**：匀速、半速、倍速、零速、以及从参考帧就没动。
- 另假设匀速基础上有**26种**旋转运动。例： $\text{Quaterniond}(1, \text{rotDelta}, 0, 0)$

- 由粗到精迭代优化，如果大于能量阈值的点超过60%，则**放大阈值**，且该层**多优化一遍**。
- 如果**某一层**的能量值大于1.5倍最小值，直接判断为失败，结束节省时间。
- 如果跟踪成功且**第0层**好于当前，则保留结果，并更新每一层的最小能量值。
- **第0层**最小值小于阈值则停止，把目前最好的值作为**下次跟踪的阈值**。

- 可以设置定时插入
- 通过**像素移动**的大小判断，位移较大、平移+旋转较大、曝光变化大、跟踪得到的能量变化大，则插入（论文中说只考虑位移，有效处理遮挡/去遮挡）



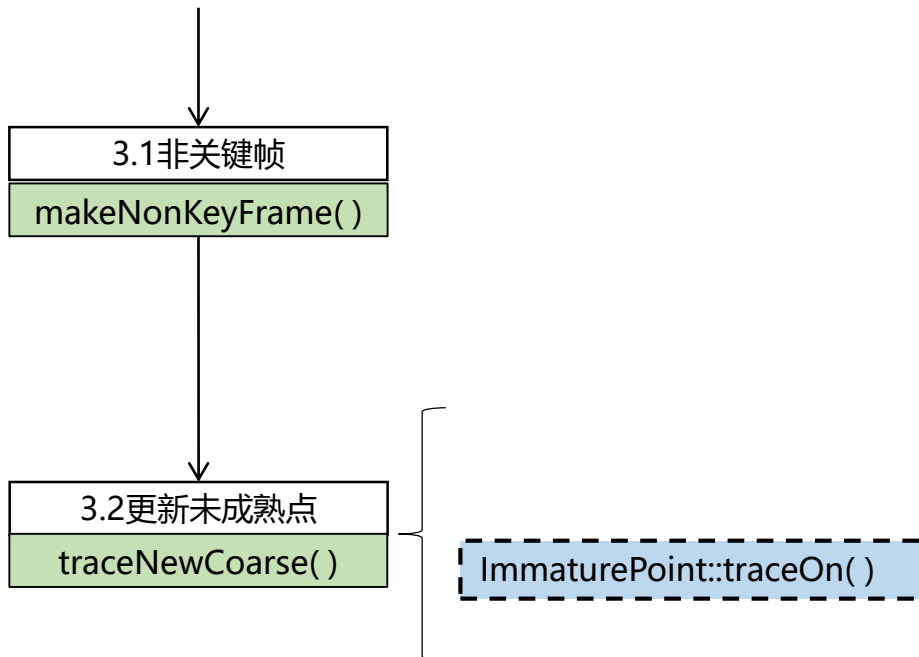


总结

建图

- 更新未成熟点ImmaturePoint的逆深度范围后，删除fh，只保留FrameShell。

- 根据逆深度范围得到极线搜索的范围
- 计算图像梯度和极线夹角的大小，如果太大则误差会很大
- 在极线上按照一定步长进行搜索能量最小的位置，和大于设置半径(2)的第二小的位置，后/前作为质量，越大越好
- 沿着极线进行GN优化，直到增量足够小
- 根据搜索得到的投影位置计算新的逆深度范围





Reference

- Engel J, Sturm J, Cremers D. Semi-dense visual odometry for a monocular camera[C]//Proceedings of the IEEE international conference on computer vision. 2013: 1449-1456.



结语

感谢各位聆听！
Thanks for Listening

