

# 滤波算法与SLAM：预备知识及SLAM基础八股文

原创

zengy5

于 2024-05-18 22:12:01 发布

阅读量911

收藏 15

点赞数 16

分类专栏：

滤波算法与SLAM

 文章标签：

算法

人工智能

笔记

机器人

经验分享



滤波算法与SLAM 专栏收录该内容

0 订阅 3 篇文章

## 滤波算法与SLAM

第一章：预备知识（均来自视觉SLAM 十四讲的三、四讲）

本系列并不面向0基础SLAM小白，默认读者拥有一定的slam和机器人开发经验  
至少跑过了基础开源代码如FAST\_LIO或ORB-SLAM，理解坐标系变换等基本知识

### 文章目录

- 滤波算法与SLAM
- 前言
- 一、状态方程与观测方程
- 二、齐次坐标
- 三、刚体变换、相似变换
- 四、升维&降维符号
- 五、其他的旋转表示方法
- 六、李群&李代数
- 七、BCH近似
- 八、李代数求导

### 前言

本章内容包含SLAM基础数学知识，包括旋转矩阵、李群、李代数等  
如需要进一步公式推导或更细致的内容，推荐阅读《视觉SLAM十四讲》第三讲和第四讲

### 一、状态方程与观测方程

定义机器人 在 $k$ 时刻的状态为 $x_k$ ，输入为 $u_k$ ，状态噪声为 $\omega_k$ ，则状态传递满足：

$$x_{k+1} = f(x_k, u_k, \omega_k)$$

设机器人在 $k$ 时刻观测到环境中第 $j$ 个路标 $y$ ，观测噪声为 $\nu_{k,j}$ ，则观测方程定义为：

$$z_{k,j} = h(y_j, x_k, \nu_{k,j})$$

这里的路标在代码中的表现形式取决于传感器的信息维度。如果是相机，则 $y_j$ 实际上代表一个特征点的坐标，如果是雷达，则是三维点。不管如何，义都是：真实世界中某一点在传感器坐标系中的投影。

### 二、齐次坐标

定义三维点 $p = [x, y, z]^T \in \mathbb{R}^3$ ，则其具有唯一的齐次坐标 $\tilde{p} = [x, y, z, 1]^T \in \mathbb{R}^4$

### 三、刚体变换、相似变换

变换矩阵一般记为 $T$ ，按照《视觉SLAM十四讲》的习惯，记 $T_{21}$ 为坐标系1到坐标系2的变换矩阵。

例如：在通用的场景下，设 $\tilde{p}_1, \tilde{p}_2$  两个点，则有 $\tilde{p}_2 = T_{21} \tilde{p}_1$

在多传感器融合的场景下， $T_{LC}$  也可代表相机到雷达的外参矩阵

在先验地图定位的场景下， $T_{k0}$  也可代表 $k$ 时刻机器人在地图中的位姿，虽然一般这种情况论文中会记为 $T_0^k$

具体地， $T$ 一般可以写为

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE(3)$$

其中 $R, t$ 分别代表旋转矩阵和平移向量，满足 $R \in SO(3), t \in \mathbb{R}^3$

则对于 $\tilde{p}_2 = T_{21} \tilde{p}_1$ 也可以写为 $p_2 = R p_1 + t$

相似变换一般记为 $S$ ，写为

$$S = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix} \in Sim(3)$$

在单目视觉SLAM中会常用到，这里不展开

## 四、升维&降维 符号

这里遵循《视觉SLAM十四讲》中对“ $\wedge$ ”符号的定义，对向量 $a = [a_1, a_2, a_3]^T$ 有

$$a^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} = A$$

同理的，对“ $\vee$ ”满足： $A^\vee = a$

这一变换为了方便记住，本文称“ $\wedge$ ”为升维，将 $3 \times 1$ 的向量升维到 $3 \times 3$ 矩阵，“ $\vee$ ”则为降维符号

## 五、其他的旋转表示方法

- 1) 欧拉角：用XYZ三轴上的旋转角度表示一次旋转，roll-pitch-yaw分别对应X-Y-Z轴上的旋转，存在万向锁的问题，一般不用；
- 2) 旋转向量：用一个单位向量 $\vec{n}$ 和一个旋转角 $\theta$ 表示一次旋转，使用罗德里格旋转公式和旋转矩阵转换：

$$R = \cos\theta \cdot I + (1 - \cos\theta)\vec{n} \cdot \vec{n}^T + \sin\theta \cdot \vec{n}^\wedge$$

$$\theta = \arccos \frac{\text{tr}(R)-1}{2}$$

- 3) 四元数：一般记为 $q = [w, x, y, z]^T$ ，代表一次旋转，亦记为 $q = [s, \vec{v}]^T$ 。设 $q_{21}$ 代表 $R_{21}$ 对应的四元数，那么 $p_2, p_1$ 满足：

$$p_2 = R_{21} p_1 = q_{21} p_1 q_{21}^{-1}$$

由四元数计算旋转矩阵通过以下公式：

$$R = \vec{v}\vec{v}^T + s^2 \cdot I + 2s \cdot \vec{v}^\wedge + (\vec{v}^\wedge)^2$$

## 六、李群&李代数

终于到这里了，最麻烦的内容，简单过一下面试能答点就行

李群的定义可以简单理解为：只对一种运算封闭的集合，叫做李群（但是作为普通SLAM玩家不需要管这些）

对于旋转矩阵 $R$ ，属于李群 $SO(3)$ ，满足： $R_1 R_2 \in SO(3)$ ，但是 $R_1 + R_2 \notin SO(3)$ ，同理变换矩阵 $T$ 属于李群 $SE(3)$

对于旋转矩阵，定义李代数 $\phi \in \mathbb{R}^3$ ，那么有

$$R = \exp(\phi^\wedge)$$

这一公式也叫指数变换：已知李代数，求李群

同理有对数变换：已知李群，求李代数

理解起来可能有点乱，代入正常的指数函数就好理解了，知道 $\mathbf{x}$ ，求 $\exp(\mathbf{x})$ 就是对 $\mathbf{x}$ 进行指数变换

就像已知李代数，要对李代数进行指数变换才能得到李群，即：李群= $\exp$ (李代数)

李代数的表示类似李群，即： $R \in SO(3), \phi \in so(3), T \in SE(3), \xi \in se(3)$

李括号用于表示两个李代数的差异性，如果是自身与自身的李括号，则为0（因为没有差别嘛）

## 七、BCH近似

引入李代数和李群，一个原因是旋转矩阵不满足加减法，只能乘除，李代数则有加减法

如果是简单的指数函数  $e^x$ ， $xy = e^{x+y}$  是显而易见的

但是对旋转矩阵是否也满足这个，答案是否定的，所以引入BCH近似，用于表示李群乘法在李代数上的表现

设旋转矩阵  $R$  对应李代数  $\phi$ ，增加一旋转  $\Delta R$ ，对应李代数  $\Delta\phi$

那么对于旋转  $\Delta R \cdot R$ ，设其对应的李代数为  $\phi_{gt}$

则用BCH近似可得

$$\phi_{gt} = J_l^{-1}(\phi)\Delta\phi + \phi$$

这里的雅可比矩阵与指数映射过程有关

## 八、李代数求导

设  $R$  是对点  $p$  的一次旋转，得到结果  $Rp$ ，求其对旋转的偏导数，而旋转矩阵并无加减法计算，无法求导，所以转为对李代数求导  
回想导数的定义：

$$\frac{df(x + \Delta x)}{d(\Delta x)} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

而《视觉SLAM十四讲》中提到的左扰动模型，是对旋转左乘一个微小扰动  $\Delta R$ ，对应的李代数为  $\Delta\phi$ ，那么有

$$\frac{\partial Rp}{\partial \Delta\phi} = \lim_{\Delta\phi \rightarrow 0} \frac{\Delta R \cdot Rp - \Delta Rp}{\Delta\phi} = \lim_{\Delta\phi \rightarrow 0} \frac{\exp(\Delta\phi^\wedge)\exp(\phi^\wedge)p - \exp(\Delta\phi^\wedge)p}{\Delta\phi}$$

计算过程不复杂，最后结果为

$$\frac{\partial Rp}{\partial \Delta\phi} = -(Rp)^\wedge$$

相比直接求导省去一个雅可比矩阵的计算，同样的，对变换矩阵也有：

$$\frac{\partial Tp}{\partial \Delta\xi} = \begin{bmatrix} I & -(Rp + t)^\wedge \\ 0^T & 0^T \end{bmatrix}$$

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心  
家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照  
©1999-2025北京创新乐知网络技术有限公司

1、ORB-SLAM2 为了特征点均匀化做了哪些处理？

构造图像金字塔，对每层金字塔计算目标特征点数目  
对每层图像做均匀分块划分，对每个块提取目标数目特征点  
对所有提取到的的特征点进行四叉树均匀化  
具体做法：

- 1）构建图像金字塔，构建八层的图像金字塔，在每层图像上提取特征点，具体每层提取多少数量的特征点都事先通过缩放因子计算好；
- 2）提取的时候，会将图像划分成一个个网格，分别在网格上进行特征点的提取。这时候特征点的数量往往是超过我们需要的点的数量，且只是初步进行了一个均匀化；
- 3）紧接着会使用四叉树均匀化的方法，这里就涉及到提取器节点的这么一个类。就是最开始整幅图像是一个节点，然后会对该节点进行划分，分成四个，然后将特征点都分别分配到这四个节点。后面就是不断重复类似的操作，划分节点，直到节点数等于所需的特征点数，然后只保留响应值最大的那个特征点。其实这里的操作也有点是把图像进行分块，直到分出来的块数等于所需的节点数，然后只留下质量最好的特征点。做完均匀化之后还会利用灰度质心法计算
- 4）通过上面的步骤就提取完特征点，然后这时候特征点的坐标还是不同图层下的坐标，因此需要将所有特征点坐标转到第 0 层图像下的坐标；
- 5）然后会进行高斯模糊，计算描述子

2、H 矩阵是什么？说出几个典型的使用场景

H 矩阵指单应矩阵，当相机发生纯旋转时，此时平移为零，基础矩阵自由度下降，基础矩阵误差较大，选用单应矩阵来恢复位姿。一般如果所有特征点都落在同一平面上，可以利用这种矩阵进行运动估计。

3、IMU 预积分为什么可以避免重复传播？

IMU 预积分与通常积分相比，解决了初始状态更新时以及加速度计、陀螺仪偏差变化时所导致的积分重新计算问题。其主要思路是计算两个关键帧之间的状态增量，当初始状态变化，就在原来增量基础上乘以初始状态的变化量；当偏差变化时，则通过求取预积分量关于偏差的雅各比，实现预积分的一阶线性近似更新。

4、so(3)和 se(3)的区别是什么，如何在算法中使用？

so3 旋转矩阵的李代数表示，se3 位姿变换矩阵的李代数表示，二者可以通过指数-对数关系实现转换，sophus 中将旋转矩阵利用 Log 函数转换成对应的旋转向量，在算法中扰动模型更新时，常将一个扰动向量通过指数映射关系转换成相应的扰动矩阵，左乘或右乘旋转矩阵实现更新

5、VINs-Mono 的初始化和 ORB-SLAM3 的初始化有啥区别？

初始化异同	ORB-SLAM3	VINS-MONO
成功的准则	加速度大于重力的0.5%(引用论文里有)代码中没看到	估计出来的重力大小误差在10%以内，IMU的测量值有一定的变化范围
是否估计加速计 bias	同时估计了加速度计和陀螺仪的bias	不估计加速度计的bias,认为加速度bias与重力（g）值相比很小，难以估计
有无单独估计陀螺仪bias	陀螺仪bias放在目标函数中一起估计	陀螺仪bias单独拿出来估计
尺度恢复策略	初始值设置为1，初始化后100内每隔10秒优化一次尺度	只在初始化的时候恢复该尺度
姿态参数化	SO(3)	基于四元数
惯性系对其方式	优化过程中同时优化惯性系到第一帧的旋转矩阵	优化得到重力在第一帧的测量值，然后对重力重新参数化，以限制其模长，最后重新来一次优化
求解过程	三次非线性优化（LocalMapping, Inertial only optimization, FULL BA)	三次非线性优化(sfm, 求解陀螺仪bias和Full BA), 两次线性求解（求尺度和速度的初值）和求相机-IMU外参
相同点	都以视觉作为基础，最终都对齐到惯性系，都在固定窗口内进行初始化。最后都进行FullBA,完成初始化	

6、为什么单目视觉 SLAM 会有尺度不确定性，而双目视觉 SLAM 却不会有？简述单目视觉 SLAM 尺度不确定的原因？  
从理论上讲，单目视觉 SLAM 的初始化运动是一个 2D-2D 的求解问题，是通过对极几何解决的。我们通过本质或者基础矩阵的分解来得到相机运动，对极约束是等式为 0 的约束，因此具有尺度等价性，由分解出的平移量乘以任

意非零常数任然满足对极约束，因此平移量是具有一个不确定的尺度的，即我们放大任意倍数，都是成立的，导致了单目尺度的不确定性。从直觉感官角度来讲，从一张图像中，我们无法判断图像中物体的实际大小，因为不知道物体的远近。双目相机拍摄一次能够获得具有视差的左右目两张图像，在已知基线的情况下，计算视差，可以通过三角测量原理直接得出深度信息，因此不存在尺度不确定性。

## 7、BA 当中雅可比矩阵的维度如何计算？

如纯视觉情况下，误差量要么是重投影误差，要么是光度误差，维度都为 2。

此时状态量有两个：位姿和空间点，那么空间点维度为 3，李代数表示位姿的话维度为 6。

所以最后雅可比就是 2 维误差量分别对 6 维位姿（ $2 \times 6$ ）、3 维空间点（ $2 \times 3$ ）求导，结果就是  $2 \times 9$ 。

所以最终结论就是：雅可比维度=误差维度  $\times$  状态量维度

## 8、用自己的语言讲一下逆深度的含义和作用？

1.和 3D 位置相比，逆深度维度更小；

2.在实际应用过程中，可能会看到类似天空这样比较远的点，导致  $z$  很大。那么即使产生了微小的变化量，也会给优化带来巨大的干扰。但是使用逆深度就可以这种情况。

3.逆深度的分布更符合高斯分布。

## 9、如何理解视觉重投影误差？

在 SLAM 初始化完成后，我们可以通过世界坐标系中的 3D 点位置和图像中的 2D 点匹配关系，由 PNP 算法初步求解两帧图像间的位姿变换。这个位姿变换并不准确，因此采用视觉重投影误差的方式进行进一步优化。重投影，意为利用我们估计的并不完全准确的位姿对三维空间点再次投影。我们已知特征点在图像中的实际位置，即观测值，利用初步求解的位姿，通过投影关系，再次将世界坐标系中的三维点投影到二维成像平面上，得到特征点像素坐标的估计值，与实际的观测值作差，对于  $n$  个特征点，构建最小二乘问题，认为差值越小，越符合实际的投影，估计越准确，最后得到优化后的位姿。

## 10、ORB-SLAM3 与 VINS-Mono 的前端有何区别？并分析这些区别的优缺点

ORB-SLAM3 跟踪线程的输入为图像帧(单目/双目/RGB-D)及 IMU 数据。对于图像数据，首先对图像进行金字塔分层，对每层金字塔提取 ORB(Fast 角点+二进制描述子)信息，并基于四叉树进行特征点均匀化，同时进行高斯模糊以及去畸变。之后进行地图初始化，ORB-SLAM3 和 ORB-SLAM2 一样都是使用纯视觉的方法进行初始化，再之后开始进行两个阶段的跟踪。第一阶段包括恒速模型跟踪、参考关键帧跟踪以及重定位跟踪，目的是为了跟得上，但估计出的位姿可能没有那么准确。第二阶段是局部地图跟踪，是利用 PNP 进行最小化重投影误差求解，得到更准确的位姿。最后决定是否新建关键帧。对于 IMU 信息，ORB-SLAM3 还需要进行 IMU 预积分，但 IMU 三个阶段的初始化是在局部建图线程完成。此外，在恒速模型跟踪中，如果有 IMU 数据，恒速模型的速度将根据 IMU 信息得到。有 IMU 的话，还会有一个 RECENT\_LOST 来记录刚刚丢失的状态。

VINS-Mono 前端的输入为单目 RGB 图像和 IMU 数据。对于 RGB 信息，首先对图像应用金字塔 LK 光流检测算法，这里金字塔的使用不同于 ORB-SLAM3 那样进行不同分辨率的特征提取，而是避免跟踪陷入局部最优。之后进行特征点均匀化，VINS 的特征点均匀化策略是以特征点为中心画圈，这个圆内不允许提取其他特征点，并使用逐次逼近去畸变。再之后使用 OpenCV 提供的光流跟踪状态位、图像边界、对极约束进行 outlier 的剔除。最终前端输出结果包含特征点像素坐标、去畸变归一化坐标、特征点 ID 以及特征点速度。对于 IMU 信息，VINS-Mono 还需要计算 IMU 预积分。ORB-SLAM3 和 VINS 的预积分都没有依赖 GTSAM，都是自己实现的。

具体的优缺点上，ORB-SLAM3 是对每一帧图像都提取 Fast 角点和 BRIEF 描述子，并进行描述子匹配。而 BRIEF 描述子的计算和匹配非常耗时。虽然 ORB-SLAM3 使用了 SearchByBoW 等策略加速匹配，但仍然需要较多的计算时间。相比之下，LK 光流只需要提取每帧图像中跟丢的特征点，计算时间大大降低。不过 ORB 特征的全局一致性更好，在中期、长期数据关联上要优于光流法。我觉得 VINS 课程讲师的描述很贴切：“VINS 更适合一条路跑到头进行定位建图，ORB-SLAM3 更适合具有回环的轨迹”

## 11、Ceres solver 怎么加残差的权重，这样做的原理是？

残差加权重的话，需要在代价函数表达式里加入信息矩阵。但 Ceres Solver 中只接受最小二乘优化，也就是说代价函数定义（自动求导和数值求导是重载括号运算符 `bool operator()`，解析求导是重载虚函数 `virtual bool Evaluate()`）里没办法直接写信息矩阵表达式。因此需要对信息矩阵进行 Cholesky 分解，把分解后的矩阵跟原来的残差乘到一起变为新的残差。

## 12、特征点法 SLAM 中，如何增强特征描述子之间的区分度？

ORB 中有个方法采用运动模型，利用上一帧位姿粗略估计当前帧位姿，然后利用 3D-2D 的匹配，计算投影点的附近圆形区域内所有特征点的匹配程度，减少匹配的计算量

提升精度方面，最近看到一个 GMS 的匹配算法。考虑特征点附近区域内特征描述子点来提高特征描述子的区分度

## 13、在没有 groundtruth 情况下如何判断自己的置信度？

走一个回环到原地 看看误差

如果是室外环境的话，可以用谷歌地图，像 VINS 论文里在港科大校园里跑的那样。有雷达的话可以把雷达轨迹当作视觉的参考。然后就是找同类型的主流算法做对比了，跟主流算法轨迹差不多应该也能说明精度。

## 14、光流跟踪在快速运动过程中，跟踪容易失败，有什么解决办法？

硬件加持：

采用全局快门相机；

引入 imu（解决相机过速退化的外部硬件首选），通过 imu 得到一个相对准确的位姿，这样光流法跟踪最小化光度误差的时候可以得到一个较好的像素点位置初值。

算法加持：

加上光度标定（参考 dso）；

引入线性约束（参考直接法点线 slam 的工作，如 edplvo）

参考 svo 直接法 + 间接法模式。

利用多层金字塔光流跟踪，先在分辨率小的金字塔层上进行光流跟踪得到在该层金字塔上的最优像素点坐标，然后恢复到下一层金字塔表示的像素点坐标，下一层金字塔以该像素点坐标为初值，在进行光流跟踪，以此类推，直到最底层金字塔

## 15、简述一下 PnP 原理，适用的场景是什么？

PnP（Perspective-n-Point）是求解 3D 到 2D 点对运动的方法。它描述了当我们知道  $n$  个 3D 空间点以及它们的投影位置时，如何估计相机所在的位姿。两张图像中，其中一张特征点的 3D 位置已知，那么最少只需三个点对（需要至少一个额外点验证结果）就可以估计相机运动。特征点的 3D 位置可以由三角化，或者由 RGB-D 相机的深度图确定。因此，在双目或 RGB-D 的视觉里程计中，我们可以直接使用 PnP 估计相机运动。而在单目视觉里程计中，必须先进行初始化，然后才能使用 PnP。3D-2D 方法不需要使用对极约束，又可以在很少的匹配点中获得较好的运动估计，是最重要的一种姿态估计方法。

## 16、简述如何在 slam 框架基础上添加语义分割网络？

最简单的思路就是做动态特征点剔除，在图像帧上使用 YOLO 等目标检测网络，或者使用 SegNet 等语义分割网络检测动态物体，对于候选区域就不再提取特征点了，然后进行运动一致性检查进一步提高检测精度。

这种方法高效、鲁棒。缺点主要有两个，一个是只能检测训练过的类别，对于未训练的运动目标无法检测。另一个是很难检测目前静止的潜在运动对象。

## 17、ORB-SLAM 中的 B 是什么？ORB 和 FAST 的区别？

ORB 的 B 指的是 **brief** 描述子，是一个二进制的描述子，计算和匹配的速度都很快。区别在于 ORB 特征描述子具备旋转不变性，而 FAST 不具备。并且 orb-slam 利用灰度质心法计算了旋转不变性，利用金字塔在不同分辨率图像上进行特征提取和匹配。

### 17.1 ORB 特征的旋转不变性是如何做的，BRIEF 算子是怎么提取的？

ORB 特征点由关键点和描述子两部分组成，关键点称为“Oriented FAST”，是一种改进的 FAST 角点，相较于原版的 FAST，ORB 通过灰度质心法计算特征点的主方向（对特征点周围像素的坐标根据灰度值加权平均得到中心坐标）然

后计算旋转后的 BRIEF 描述子。

BRIEF 描述子通过对前一步关键点的周围图像区域选取 128 对点，根据每对点灰度值大小关系分别取 0、1，得到 128 维 0、1 组成的向量即描述子。

## 18、回环检测如何理解？请简述你熟悉的开源框架实现流程

SLAM 运行过程中会产生轨迹漂移，回环检测就是当机器人回到以前运行过的场景时，利用约束将轨迹漂移拉回来。视觉 SLAM 回环的主流方法就是词袋，即用词袋建立数据库，看关键帧中有没有在数据库中有比较“接近”的关键帧。但是词袋的使用方式有所不同。

ORB-SLAM3 的闭环线程流程：

首先检测缓冲队列中是否有新的关键帧进来；

然后检测共同区域，如果检测到公共区域并且公共区域发生在当前地图和非活跃地图就执行地图合并，如果公共区域在当前地图内，就执行闭环。

具体的公共区域检测流程也和 ORB-SLAM2 不一样，ORB2 是连续 3 个闭环候选关键帧组都检测到了回环关系才认为是回环，要检查时间连续性和几何连续性，延时较高，容易漏掉回环。而 ORB3 的重点是检查几何一致性，主要是说当前关键帧的 5 个共视关键帧只要有 3 个和候选关键帧组匹配成功，就认为检测到了共同区域，如果不够 3 个，再检查后续新进来关键帧的时间一致性，以略高的计算成本为代价，提高了召回率和精度。之后利用共视关系来调整位姿和地图点。

VINS-Mono 的闭环检测：

首先对所有新老角点进行 BRIEF 描述，计算当前帧与词袋的相似度分数，进行闭环一致性检测，获得闭环的候选帧。当检测到闭环后，开启快速重定位，将之前帧的位姿和相关特征点作为视觉约束项，加到后端非线性优化的整体目标函数中，根据优化得到的相对位姿关系，对滑窗内所有帧进行调整。当从滑窗内滑出的帧与数据库中的帧为闭环帧时，则对数据库中的所有帧进行闭环优化。

## 19、说说多目相机的标定原理和主要流程

对于双目相机来说，首先要对左右目的两个相机进行单目标定，得到各自的内参矩阵和畸变系数。然后标定左右目相机之间的外参，当两个相机位于同一平面时，旋转矩阵可近似为单位阵，标定平移外参可得到基线长度  $b$ 。

主要流程为：

准备一张棋盘格，粘贴于墙面，用直尺测量黑白方格的真实物理长度。

使用双目相机分别从不同角度拍摄得到一系列棋盘格图像。

进行左目相机标定，得到左目内参矩阵  $K_1$ 、左目畸变系数  $D_1$ 。

进行右目相机标定，得到右目内参矩阵  $K_2$ 、右目畸变系数  $D_2$ 。

将标定得到的参数  $K_1$ 、 $K_2$ 、 $D_1$ 、 $D_2$  作为输入，再同时利用左右目对齐后的棋盘格图片，调用 OpenCV 的 stereoCalibrate 函数，输出左右目的旋转矩阵  $R$ 、平移向量  $T$ 。

基于标定得到的  $K_1$ 、 $K_2$ 、 $D_1$ 、 $D_2$ 、 $R$ 、 $T$  进行双目视觉的图像校正。

## 20、用自己的语言描述下预积分原理和过程？

IMU 可以获得每一时刻的加速度和角速度，通过积分就可以得到两帧之间的由 IMU 测出的位移和旋转。但 IMU 的频率远高于相机，在基于优化的 VIO 算法中，当被估计的状态量不断调整时，每次调整都需要在它们之间重新积分，传递 IMU 测量值，这个过程会导致计算量的爆炸式增长。因此提出了 IMU 预积分，希望对 IMU 的相对测量进行处理，使它与绝对位姿解耦，或者只要线性运算就可以进行矫正，从而避免绝对位姿被优化时进行重复积分。

ORB-SLAM3 没有依赖 GTSAM 库，整个预积分的过程都是自己实现的。具体思路为：

由积分引出预积分，得到旋转、速度、位置的预积分表达式，重点是消除第  $i$  时刻对积分的影响并保留重力、噪声分离，推导出标定好的 imu 噪声对预积分的影响。

噪声递推，之前的推导结果要么是求和，要么是多次积分导致每次新来一个数据都需要从头计算，这给计算平台带来资源的浪费，因此要推出误差的递推形式。还需要计算协方差矩阵



由于偏置在 VIO 算法中会作为状态量来优化，因此要推出当偏置变化时直接求得新的预积分结果。

定义残差对于旋转、速度、位置的残差。

## 21、光流法和直接法的区别

光流法和直接法都是基于灰度不变假设。光流法提取关键点后，根据灰度不变假设得到关于像素点速度的欠定方程，之后通过对极约束、PNP、ICP 计算  $R$ 、 $t$ 。但光流法只是一个粗略匹配，很容易陷入局部极值（VINS 使用金字塔来改善），直接法相当于在光流基础上对前面估计的相机位姿进一步优化，即将下一帧位置假设为  $1/z * K(RP+t)$ ，之后利用光度误差来优化  $R$ 、 $t$ 。

光流法和直接法根据提取关键点数量可分为稀疏、半稠密、稠密。相较于特征点法来说，提取速度很快，但要求相机运动较慢，或者采样频率较高，对光照变化也不够鲁棒。

## 22、谈一谈你对边缘化的理解和它的意义？

SLAM 属于渐进式算法，需要使用一定数目的关键帧进行 BA 运算以求得位姿和地图点。但随着关键帧数目的增多，滑窗（或者说 ORB-SLAM3 的共视图）的计算量会越来越大。因此需要想办法去掉老的关键帧，来给新关键帧腾地方。直接做法是把最老帧以及相关地图点直接丢弃，但最老帧可能包含了很强的共视关系，直接删除可能会导致 BA 效果变差。

而边缘化就是指，既从滑窗/共视图里剔除了老的关键帧，又能保留这个关键帧的约束效果，其核心算法是舒尔补。总体来说，就是本身要被扔掉的最老帧，他可能和后面的帧有约束关系（一起看到某个地图点，和下一帧之间有 IMU 约束等），扔掉这个第 0 帧的数据的话，这些约束信息需要被保留下来。相当于建立起地图点和地图点之间的约束关系。

## 23、剔除离群点有哪些方法？

只对多个图像帧都观测到的点进行 BA 运算，观测太少的点就不参与优化了

ORB-SLAM2/3 中对于深度超过 40 倍基线的点，由于估计的平移分量非常不准，因此认为是外点，剔除在重投影计算时，如果投影误差超过一定阈值，则不让这个点参与优化，ORB-SLAM2/3 是通过卡方检验实现 VINS-Fusion 将追踪到的光流再反向追踪回来，然后比较追踪回来的特征点与原先的特征点之间的距离，过大就认为是外点去掉

VINS-Mono 中通过 opencv 的 findFundamentalMat 函数自带的 ransac 来消除外点

鲁棒核函数

MAD(median absolute deviation)算法：即绝对中位值偏差，其大致思想是通过判断每一个元素与中位值的偏差是否处于合理的范围内来判断该元素是否为离群值。

标准差法- $3\sigma$  法：因标准差本身可以体现数据的离散程度，和 MAD 算法类似，只是标准差法用到的不是中位值，而是均值，并且离群阈值为  $3\sigma$ ，即  $|x_{out} - x_{mean}| > 3\sigma$ 。

百分位法：百分位算法类似于比赛中“去掉几个最高分，去掉几个最低分”的做法，基本步骤如下：1)将数据值进行排序，一般用升序排序，这里排序函数可自定义；2)去掉排位百分位高于 90%或排位百分位低于 10%的数据值。

## 24、SLAM 后端不收敛怎么办？

第一次进入后端优化就不收敛，这时候就应该分析优化变量和约束条件的理论和 code 是否正确：

前面优化收敛，突然当前优化不收敛，这时候需要判断下是否加入了错误的回环约束：删除回环约束，重新本次优化，如果 OK，则应该是回环错误。

迭代次数不够，增加迭代次数。

优化方法的问题：使用高斯牛顿法时，可能出现  $J^T J$  ( $H$  的近似) 为奇异矩阵或者病态的情况，此时增量稳定性较差，导致算法不收敛。（残差函数在局部不像一个二次函数，泰勒展开不是很准确），导致算法不收敛。而且当步长较大时，也无法保证收敛性改进更换增量方程的求解方法，如果使用的是高斯牛顿法，可尝试使用 LM 法算法。

## 25、在机器人仅定位阶段，如果实际地图与加载地图发生较大差异，应该如何处理？

可以减小环境匹配约束（如 scan-to-map）的权重，具体可以通过设置核函数：环境匹配残差较小时正常引入环境约束，过大时根据残差与设定阈值的比例设置权重。

## 26、PnP 位姿最少需要几个点？有几组解？误差来源于哪里？



PnP 位姿估计最少需要 7 个点（3 对点以及一个验证点），通过 SVD 分解求解位姿时会得到 4 组解。

误差来源主要是：

特征匹配正确性（数据关联正确程度）

测量数据的噪声情况（特征点是否去畸变，是否坐标精确）

其次光照变化、运动变化都会带来误差。

27、描述一下关键帧是什么？有什么用？如何选择关键帧？

关键帧是在一系列普通帧中选择出的最有代表性的一帧。

降低信息冗余度。

减少待优化的帧数，节省计算资源。

选取关键帧时进行了筛选，防止无用或错误信息进入优化过程。

选取的指标主要有（都要满足）：

距离上一关键帧的帧数是否足够多（时间）。

距离最近关键帧的距离是否足够远（空间）/运动。比如相邻帧根据 pose 计算运动的相对大小，可以是位移也可以是旋转或者两个都考虑，运动足够大（超过一定阈值）就新建一个关键帧，这种方法比第一种好。但问题是如果对着同一个物体来回扫就会出现大量相似关键帧。

跟踪质量（主要根据跟踪过程中搜索到的点数和搜索的点数比例）/共视特征点

28、ORB-SLAM3 中如何存储关键帧、地图点及数据结构？

关键帧和地图点都是以类的方式存储的。

关键帧中除了包括位姿之外，还包括相机内参，以及从图像帧提取的所有 ORB 特征（不管是否已经关联了地图点云，这些特征点都已经被畸变矫正过）。

地图点中除了存储空间点坐标，同时还存储了 3D 点的描述子（其实就是 BRIEF 描述子），用来快速进行与特征点的匹配，同时还用了一个 map 存储了与其有观测关系的关键帧以及其在关键帧中的 Index 等等。

29、ORB-SLAM 单目初始化的时候为什么要同时计算 H 矩阵和 F 矩阵？

H 矩阵和 F 矩阵适用于当前场景的不同状态。

F 矩阵（基础矩阵）在纯旋转以及空间点在一个平面上的情况时效果不好，H 矩阵（单应矩阵）推导即针对特征点在同一平面时计算出的，一定程度上弥补其缺陷。

因此当计算出两个矩阵的值后分别计算重投影误差得到分数，比较哪个矩阵的得分占比更高则取哪个矩阵恢复位姿，在 ORB-SLAM 系列中作者倾向于采取 H 矩阵恢复，当比值大于 0.4 时即选择 H 矩阵。

30、ORB-SLAM 的共视图是什么结构？内部如何存储的？

ORB-SLAM 中由关键帧构成了几种重要的图结构，包括共视图、本质图和生成树；

共视图是无向加权图，每个顶点都是关键帧。如果两个关键帧之间满足一定的共视关系（至少有 15 个共视地图点），则它们就会连成一条边，边的权重是共视点数；

ORB-SLAM 使用两个链表 `list<KeyFrame*>`、`list<int>` 分别保存按照权重从大到小排列后的共视关键帧及其权重。

31、ORB-SLAM3 中特征点是如何均匀化的？你还知道哪些均匀化方法？各有什么优劣

构建图像金字塔，构建八层的图像金字塔，在每层图像上提取特征点，具体每层提取多少数量的特征点都事先通过缩放因子计算好；

在提取阶段将每层图像金字塔分割为一个个网格，分别在每个网格内提取特征点并采用非极大值抑制。（对关键点进行非极大值抑制，即抑制掉那些周围有更强响应的关键点，只保留那些相对来说比较独特或者突出的关键点。这样可以使得检测到的关键点更加鲁棒，同时也能够减少冗余的关键点，提高算法的效率。）

进一步用四叉树对上述提取的特征点均匀化。这里用到提取器节点的这么一个类。就是最开始整幅图像是一个节点，然后会对该节点进行划分，分成四个，然后将特征点都分别分配到这四个节点。后面就是不断重复类似的操作，划分节点，直到节点数等于所需的特征点数，然后只保留响应值最大的那个特征点。

vins:设置 mask 限制特征点之间的距离,在现有的特征点范围内画一个 mask 在这个 mask 范围内不再提取特征点。

32、SLAM 里多传感器松耦合与紧耦合的区别？

以 IMU 和相机两种传感器为例。视觉传感器和惯性传感器有一定的互补性，IMU 更适合估计短时间的高速运动，视觉传感器更适合估计长时间低速的运动。

两种耦合方式的区别：

松耦合：IMU 和相机分别进行自身运动估计，然后对位姿估计结果进行融合。松耦合情况下，存在两个优化问题，即视觉测量优化问题和惯性测量优化问题。两个优化问题分别利用视觉测量信息（如检测到的特征点）和惯性测量信息（加速度、角速度）对各自的状态向量进行优化，最终结果只是把两个优化过程得到的位姿估计结果进行融合。

(2) 紧耦合：IMU 和相机共同构建运动方程和观测方程，再进行状态估计。紧耦合方式只存在一个优化问题，它将视觉和惯性放在同一个状态向量中，利用视觉测量信息（如特征点）和惯性测量信息构建（如加速度、角速度）包含视觉残差和惯性测量残差的误差项，同时优化视觉状态变量和惯性测量状态变量。

一般采取紧耦合方案，因为这样可以更好的利用两类传感器的互补性质，使两类传感器相互校正，达到更好的效果。

对于更多传感器，紧耦合只是根据各传感器的特点，在系统状态向量中加入新的状态，同时利用各传感器的测量信息构建各自的误差项，最终通过使总误差最小，对状态向量进行优化。

33、请用自己的语言阐述 BA 优化原理？

BA 是指通过调整相机的位姿和特征点的 3D 位置，使总体的重投影误差最小。

从非线性优化的角度来看，代价函数是特征点的像素坐标与三维点经过投影后的像素坐标进行作差得到的重投影误差，优化变量是相机位姿和三维点位置，优化方法一般可以采取高斯牛顿法(GN) 或列文伯格-马夸尔特法(LM)。

BA 优化的过程是通过对最小二乘形式的代价函数进行求解，将相机位姿和三维点位置(优化变量) 从粗略的初始值调整到更精确的值。

34、ORB-SLAM 怎么克服尺度漂移问题？

单目 vo 的尺度漂移是由于，尺度不确定，以及位姿估计，地图点的估计。位姿估计和地图点估计相辅相成，计算误差，测量误差，会在这个过程中一步一步的传播下去。

要解决这个尺度漂移的话问题，可以考虑中断这个误差传播链条，比如双目，rgbd 相机可以直接计算特征点的深度，所以地图点的误差就被固定在一定范围，从而限制了整个估计的误差传播发散。

同时 imu 传感器可以提供绝对尺度，虽然 imu 位姿长时间发散严重，但是短时间内的位姿估计可以认为误差有个上限，从而 vio 也可以在位姿估计上限制误差传播。

35、ORB-SLAM3 回环过程是怎么实现的，要估计哪些量？

检查是否有新关键帧，如果检测到公共区域，非活跃地图则地图融合，活跃地图则闭环

NewDetectCommonRegions()寻找后选关键帧，找到当前帧共有公共单词的关键帧，计算得分，得到三个组的最高分关键帧。

定义局部窗口，在窗口内找到候选帧和当前关键帧匹配的地图点。

使用最近邻 ORB 特征点对之间的几何约束关系，通过 RANSAC 算法估计当前帧与之前某一帧之间的运动模型

将之前所有关键帧及其之间的相对位姿组成的图形优化，以最大化一个目标函数，有重投影误差、平移和旋转的惩罚项

如果回路检测失败，则尝试通过当前帧与之前某一关键帧之间的 RANSAC 检测和全局优化来重定位，即计算当前帧的绝对位姿。

36、ORB-SLAM 的哪个部分最耗时？有什么加速建议？

全流程 BA 优化最耗时，但优化是单独一个线程，且不要求实时进行，所以没有特别强的加速需求。一般加速考虑从前端跟踪部分着手，特征提取方面 ORB 速度已经是目前比较快的策略，但特征匹配比较耗时，可以考虑采用 GMS 等更新的特征匹配算法替换。

37、什么情况下无法正确计算 E 矩阵，为什么？

当相机做纯旋转，平移为 0 的时候无法计算 E 矩阵，因为推导本质矩阵 E 和基础矩阵 F 的过程是通过两不同位置对

同一 3D 点观测列两组方程，联立推导出对极约束，而当没有平移时不满足不同位置，推到中的  $t=0$ ，本质矩阵  $E$  也为 0，无法计算。

### 38、SLAM 与 SFM 的区别？

区别：

slam 的目标是定位，SFM 的目标是 3D 重建；

SLAM 要求实时，数据是线性有序的，无法一次获得所有图像，部分 SLAM 算法会丢失过去的部分信息；基于图像 SFM 不要求实时，数据是无序的，可以一次输入所有图像，利用所有信息

SLAM 是个动态问题，会涉及滤波，运动学相关的知识，而 SFM 主要涉及的还是图像处理的知识；

联系：

基本理论都是一致 - 多视图几何。传统方法都是做特征点提取与匹配，都需要最小化视觉投影误差。

### 39、选取某个你熟悉的 SLAM 算法，谈谈可以提升算法的运行速度的思路？

ORB-SLAM2 和 3 系列在特征匹配部分可以采用 GMS 算法，准确率更高速度也能适当提升；另外前端跟踪部分可以用光流法跟踪普通帧，关键帧仍采用特征点进行跟踪，这个应该已经有论文实现了；但对这种方法应该可以调整关键帧插入策略，在用光流法跟踪大部分普通帧之后可以根据设计当前环境的光流跟踪难度系数，避免出现光流跟不上的情况。

### 40、说出 ORB-SLAM 的原版加速的一些 track？

求本质和单应矩阵开多线程加速；

计算关键点灰度质心方向时，利用圆的对称性每次循环求解对称的两组值；

最后使用 FastAtan2 函数加速计算角度；

通过近邻窗口半径（提取特征时预先分配至网格）、投影后坐标窗口附近、Bow 词袋等几种方法加速特征匹配；

每次先将某些变量的倒数值计算出来存放至 inverse 变量，减少函数循环计算时除法的计算量；

循环或函数开始前先把 vector 等变量 reserve 扩容至足够大，防止循环时因容量不够发生临时扩容；

计算双目特征匹配时，只在某近邻像素行范围内查找，不全局暴力匹配；

访问多线程加锁数据时，快速解锁，避免长时间独占共享数据；

正确选用合适的容器，如存储特征点的 vector（频繁随机访问，变化较小；提取完成后很少再进行插入和删除）；存储共视关键帧的 map 容器（单次查询是否为共视帧的性能稳定，个人猜测不用 unordered map 的原因是求共视关键帧排序时需要使用迭代器迭代，而有序 map 迭代性能差）；四叉树均匀化分配特征点以及记录连接关键帧、局部地图点（如 ILocalMapPoints）时使用 list 链表（不经常随机访问，而是遍历迭代访问，经常需要插入，list 插入效率高）；Map 中的地图点和关键帧用 set 容器，因为经常需要做插入删除动作（印象中 orbslam2 这里地图点的删除没实现，可能会造成内存泄漏）set 有序，无重复元素，方便转 vector（返回关键帧集合转化的 vector 给可视化和 BA 优化程序用）

### 41、ceres 库为什么可以自动求解雅克比？

Ceres 库中的自动求导是基于 Jet 类型实现的，利用链式法则来计算函数的导数。在计算图中，每个节点代表一个计算步骤，节点的输入是前一步的输出，输出是当前步骤的计算结果，节点是 Jet 类型，其向量部分存储该点处函数的导数值

### 42、SLAM 过程中如何降低动态物体的影响？

利用深度学习技术识别图像中特定的物体，结合运动物体的先验知识（汽车、行人、动物），去除潜在的动态区域，该方法是最为简单高效的方案，也是目前主流的技术方案。但对于移动的椅子、停在路边的汽车等，该方法往往会做出错误的判断。

多视图几何检测动态特征点，原理是用多帧图像的位姿约束，剔除误差较大的特征点。常用的方法为极线约束。

对于两张图像，稠密光流能很好的描述每个像素在二维平面中运动的情况，是检测运动区域很好的方法，一般来说运动物体的区域所产生的光流会远高于静态的背景区域。

### 43、解释相机内参、外参的含义。图片放大两倍，内参如何变化？

相机内参在相机出厂之后是固定的，有  $f_x$ ,  $f_y$ ,  $c_x$ ,  $c_y$  四个参数，相机内参描述了从图像坐标系到像素坐标系之间

的关系，即进行单位变换和平移。

在图像坐标系中坐标  $x,y$  的单位是米，像素坐标系中单位是像素:  $fx, fy$  是将单位为米的  $f$  缩放为单位为像素；  
在图像坐标系中图像中心为坐标系原点，像素坐标系中左上角为坐标系原点:  $cx, cy$  分别代表两个坐标系原点在  $x,y$  方向上的平移，一般是图像长和宽的一半。

四个参数的单位均为像素。所以当图片方法两倍，四个内参数均放大两倍。

相机外参即相机的位姿  $R, t$ ，描述的是世界坐标系和相机坐标系之间的关系，外参会随着相机运动发生改变。

#### 44、左乘和右乘一个扰动是相同的吗？请解释原因

定义 Global 坐标系为固定坐标系，如导航中常用的东北天坐标系；local 坐标系为绑定在运动体上的坐标系，是随动的。

一般旋转扰动的定义是在机体 local 坐标系上添加一个小的旋转  $\theta$ ，这样扰动的旋转角比较小，可以保证比较好的线性而且避免奇异性。

当 pose 表达在 Global 坐标系时，根据旋转的叠加方式，添加的是右扰动；

当 pose 表达在 local 坐标系时，根据旋转的叠加方式，添加的是左扰动。

原因见下图：

$$\text{已知: } R_{L'}^L = \exp([\theta_{L'}^L]^\times).$$

$$\text{Global坐标系下右扰动: } R_{L'}^G = R_L^G R_{L'}^L = R_L^G (I + [\theta_{L'}^L]^\times)$$

$$\text{local坐标系下左扰动: } R_G^{L'} = R_L^{L'} R_G^L = (I + [\theta_{L'}^L]^\times) R_G^L = (I - [\theta_{L'}^L]^\times) R_G^L$$

CSDN @double立

#### 45、BA 和 pose graph 优化的区别是什么？

BA: 是把所有的三维点和位姿放在一起作为自变量进行非线性优化；

pose graph: 是优化所有的位姿，目的是将位姿之间的不同方式计算得到约束残差最小；

本质上 BA 包含了 pose graph 优化，两种优化方法都会转换为求解正规方程，两者形式完全一样；

pose graph 与 BA 的区别，仅仅体现在 BA 中使用三维点间接地表达 pose 间的约束关系。

#### 46、ceres 库中 huber 什么作用？是怎么设置的？

Huber 是设置 LossFunction 时设置的，是核函数的一种，主要就是减少异常值对优化结果的影响，一般定义语句就是 `loss_function = new ceres::HuberLoss(value)`

#### 47、单目地图初始化过程对相机运动有哪些要求？如何能够稳定快速的初始化？

对相机的运动要求：单目初始化应该避免纯旋转，要有一定程度上的平移。但是还需要注意，平移不能太小，如果太小会导致较大的深度不确定性；如果平移太大的话，图像内容变化大的话由可能导致特征匹配成功数量降低。