ORBSLAM3 系列代码讲解

优化问题 专题一

主 讲 人：魏宏宇

公 众 号：3D视觉工坊

主要内容

1 滤波与非线性优化

2 SLAM十四讲中的非线性优化算法重点总结

3 讨论与交流
（假期过后如何快速进入科研状态）

1 滤波与非线性优化

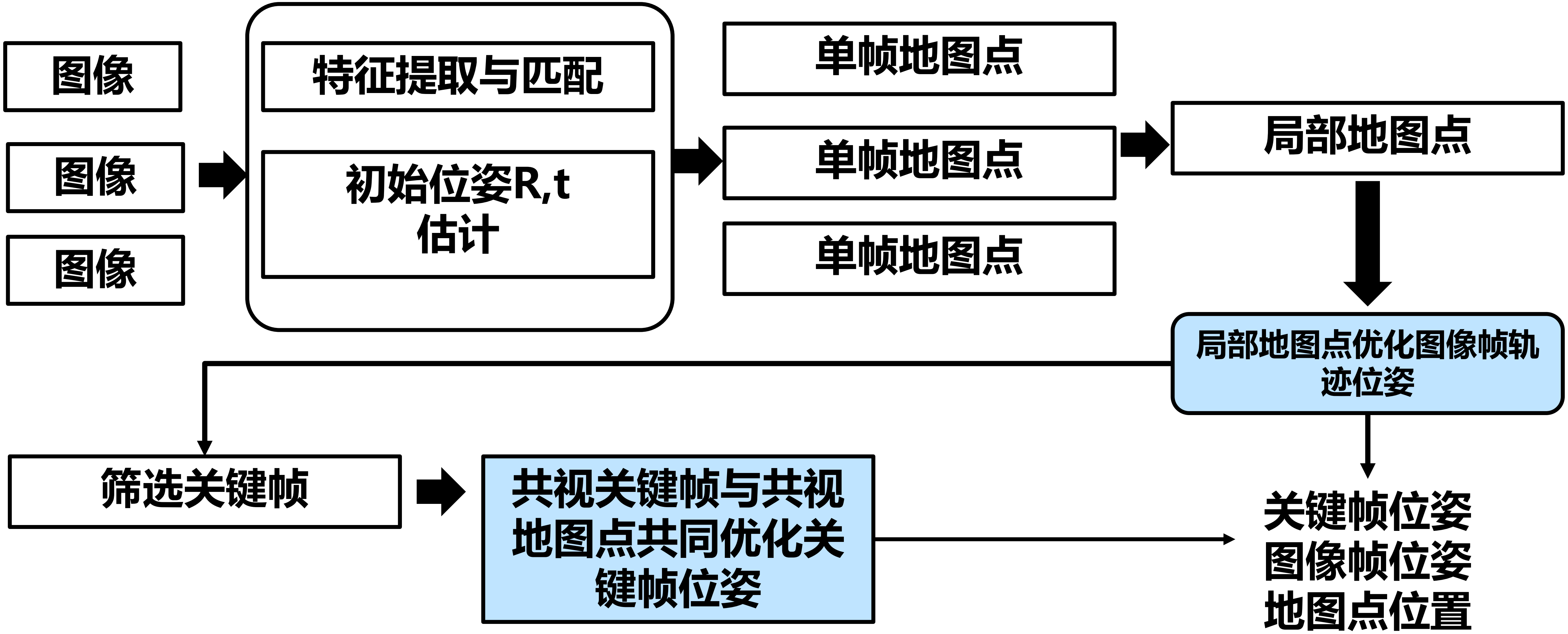
SLAM系统根据后端优化的方式，可分为基于滤波的SLAM和基于优化的SLAM。其中基于优化的SLAM指的是基于非线性优化的SLAM算法。

滤波SLAM源于传统导航与状态估计，通过利用前一时刻的状态信息推断/估计当前时刻的状态，并利用协方差实现状态与轨迹的更新。由于与当前时刻的估计量有关的只是前一时刻的状态，因此基于滤波的SLAM算法往往由于其较低的计算量而获得较好的轨迹估计实时性。另外滤波框架相对于非线性框架具有更好的扩展性，易于与其他更多传感器进行组合导航与定位。总结来说，滤波SLAM在大型导航与定位系统（如自动驾驶、全源导航等）中使用频率更高。

非线性SLAM算法主要是从概率学的角度出发，利用最大似然概率或最大后验概率的思想来实现当前状态的估计和优化。非线性SLAM算法一方面利用迭代增量最小二乘的方式获得更加准确的估计，一方面通过滑动窗多状态联合优化与估计的方式实现状态的更新与优化。由于非线性SLAM算法利用局部多状态实现精确状态估计，因此不可避免存在计算量大、实时性略低、内存空间占比大的问题，但随着稀疏性和边缘化的提出，计算量问题得到了一定的缓解，极大程度上扩展了非线性优化的使用率。从需求角度分析，非线性SLAM更多用于小型导航与定位系统（如无人机、AGV等）。

滤波SLAM：MSCKF、ROVIO、S-MSCKF、其他基于UKF、IEKF、EKF等的SLAM算法

优化SLAM：OKVIS、ORBSLAM系列、DSO、SVO、VINS-MONO等


图像
图像
图像
特征提取与匹配
初始位姿R,t
估计
单帧地图点
单帧地图点
单帧地图点
局部地图点
局部地图点优化图像帧轨迹位姿
筛选关键帧
共视关键帧与共视地图点共同优化关键帧位姿
关键帧位姿
图像帧位姿
地图点位置

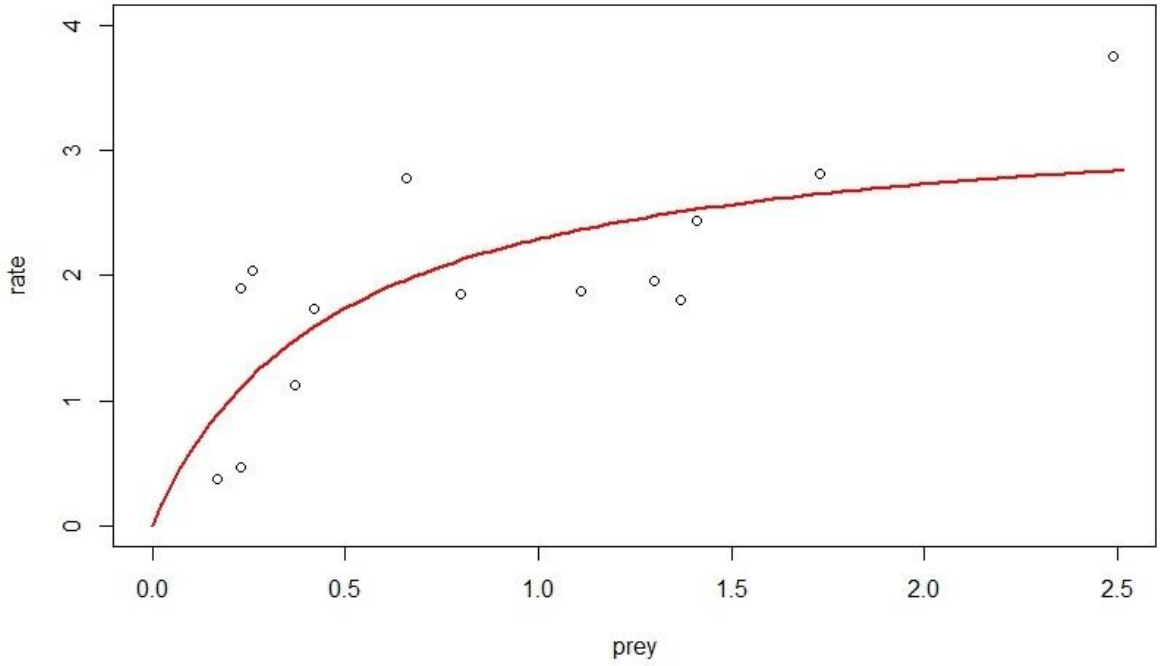
2

SLAM十四讲中的非线性优化算法重点总结

非线性优化相关章节（以第一版为例）

- 第六讲：非线性优化
  - 6.1 状态估计问题
  - 6.2 非线性最小二乘
- 第七讲：视觉里程计1
  - 7.7 3D-2D：PnP
  - 7.8 实践：求解PnP
  - 7.9 3D-3D：ICP
  - 7.10 实践：求解ICP
- 第九讲：设计前端
  - 9.3 改进：优化PnP的结果
- 第十讲：后端1
- 第十一讲：后端2

非线性优化解决的问题？



SLAM优化问题的实质：利用观测量（特征点）得到的3D地图点实现载体运动轨迹的曲线拟合

非线性最小二乘 ⇒ 非线性轨迹优化及地图点位置优化

简单数学问题 复杂工程问题

抽象形式

$$\min_{x} \frac{1}{2}\|f(x)\|_2^2.$$

$f(x)$是一个与x有关的非线性函数。求解的问题是自变量x为多少时，$f(x)$可以取得最小值，当$f(x)$为最小时，此时的x为最优解

具象化

$$y_i = \boxed{\exp\left(x_1 P_i^2 + x_2 P_i + x_3\right) + \omega}$$

实际上P_i位置对应的y_i

利用估计的x_1, x_2, x_3获得的理想中的y_i

存在一系列离散的点(P_i, y_i),试图利用x_1, x_2, x_3拟合一条曲线表征P与y之间的变化规律，其中ω表示拟合过程中或P_i的噪声，噪声通常是呈高斯分布规律的。

由于噪声及x_1, x_2, x_3的不准确，理想y_i(右侧)和实际y_i(左侧) 之间无法真正相等，因此始终会存在一个误差e，即：

$$e = y_i - \exp\left(x_1 P_i^2 + x_2 P_i + x_3\right) + \omega = f(x)$$

为获得更加符合变化规律的拟合曲线，我们需要尽可能调整x_1, x_2, x_3的取值，降低误差e, 从而获得多维函数$f(x)$的极小值点。

非线性优化的求解方式

$$e = y_i - \exp(x_1 P_i^2 + x_2 P_i + x_3) + \omega = f(x)$$

从数学的引入一下，函数$f(x)$ 的极值点通常出现在$f(x)$的导数为0的地方，即$f'(x) = 0$，若$f(x)$为简单的线性函数，我们通常直接对其进行求导，获得导数为0的点所对应的 f(x)，然后比较$f(x)$的值，取最小处即可。

但若函数较复杂且非线性较明显，很难实现准确的导数求解，因此，往往我们使用的是迭代的方式，从某个初始值出发（如初始情况下，令$x_1 = x_2 = x_3$=0），每次进行**迭代更新**（$x_1 = \mathrm{x}_1 + \boldsymbol{\Delta x_1}$, $x_2 = \mathrm{x}_2 + \boldsymbol{\Delta x_2}$, $x_3 = \mathrm{x}_3 + \boldsymbol{\Delta x_3}$），向$f(\mathrm{x})$的最小值无限逼近，从而得到最佳的$x_1, x_2, x_3$。

1. 给定某个初始值 $\boldsymbol{x}_0$。
2. 对于第 k 次迭代，寻找一个增量 $\Delta \boldsymbol{x}_k$，使得 $\|f(\boldsymbol{x}_k + \Delta \boldsymbol{x}_k)\|_2^2$ 达到极小值。
3. 若 $\Delta \boldsymbol{x}_k$ 足够小，则停止。
4. 否则，令 $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta \boldsymbol{x}_k$，返回 2.

关键是获得一个合适的增量

高斯牛顿法

$$\min_{x} \frac{1}{2}\|f(\boldsymbol{x})\|_2^2.$$

我们试图借助Δx 获得最小的$f(x)$，因此应将$f(x)$ 改写成关于Δx的式子

1、将$f(x)$进行一阶泰勒展开

$$f(\boldsymbol{x}+\Delta\boldsymbol{x}) \approx f(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})\Delta\boldsymbol{x}.$$

上式在给定初始值x的条件下转化为下面的公式

$$\Delta\boldsymbol{x}^* = \arg\min_{\Delta\boldsymbol{x}} \frac{1}{2}\|f(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})\Delta\boldsymbol{x}\|^2.$$

2、求导

此时待求解的公式中变量变为了Δx，因此将二范数展开并求解关于Δx的导数，使其为0，可得到

$$\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{J}(\boldsymbol{x})\Delta\boldsymbol{x} = -\boldsymbol{J}(\boldsymbol{x})^T f(\boldsymbol{x}).$$

高斯牛顿法

$$\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}) \Delta \boldsymbol{x} = -\boldsymbol{J}(\boldsymbol{x})^T f(\boldsymbol{x}).$$

注意：此时的自变量为Δx，应变量为$f(x)$，因此此公式完全是一个关于Δx的线性方程

3、定义增量方程/正规方程

用标准且统一的符号重写上式：

$$\boldsymbol{H} = \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}),\ \boldsymbol{g} = -\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x})$$

即：

$$\boldsymbol{H}\Delta\boldsymbol{x} = \boldsymbol{g}$$

4、更新x

求解$H\Delta x = g$后获得初始值为x时的Δx，更新x得到新的x、$f(x)$以及$J(x)$：

5、利用更新后的x继续迭代，直到Δx足够小（可以通过前面的分析知道当Δx足够小的时候，$f(x)$也足够小），拟合曲线非常接近真值

高斯牛顿法 → 列文伯格-马夸尔特法

$$\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}) \Delta \boldsymbol{x} = -\boldsymbol{J}(\boldsymbol{x})^T f(\boldsymbol{x}).$$

$$\boldsymbol{H}\Delta\boldsymbol{x} = \boldsymbol{g}$$

GN中，要想求得Δx，需要保证H是正定且可逆的，但实际中$J^T J$大多是半正定的（$J^T J \geq 0$），从而使得无法获得理想的Δx
因此产生大量GN的变种，从而保证最佳效果的优化

LM在一定程度上修正了这些问题，通常相比于GN，LM更加鲁棒，收敛速度相比于GN较慢一些，通常也被称为阻尼牛顿法

列文伯格-马夸尔特法

$$f(\boldsymbol{x}+\Delta\boldsymbol{x}) \approx f(\boldsymbol{x})+\boldsymbol{J}(\boldsymbol{x})\Delta\boldsymbol{x}.$$

回想GN的第一步，在x附近泰勒展开，因此我们其实求解的是以初值x为中心的邻域范围中的函数最优解，我们需要保证求解的Δx一定是在初值的邻域范围内，Δx不能太大而造成近似的不准确。
于是，LM的方法就是为Δx添加一个信赖区域，从而保证近似的有效性

那么如何确定这个信赖区域的范围呢？一个比较好的方法是根据我们的近似模型跟实际函数之间的差异来确定这个范围：如果差异小，我们就让范围尽可能大；如果差异大，我们就缩小这个近似范围。因此，考虑使用

$$\rho=\frac{f(\boldsymbol{x}+\Delta\boldsymbol{x})-f(\boldsymbol{x})}{\boldsymbol{J}(\boldsymbol{x})\Delta\boldsymbol{x}}.$$

来判断泰勒近似是否够好。ρ 的分子是实际函数下降的值，分母是近似模型下降的值。如果 ρ 接近于 1，则近似是好的。如果 ρ 太小，说明实际减小的值远少于近似减小的值，则认为近似比较差，需要缩小近似范围。反之，如果 ρ 比较大，则说明实际下降的比预计的更大，我们可以放大近似范围。

列文伯格-马夸尔特法

根据上面的分析，我们可以知道，当利用LM的方法时，我们不仅需要获得一个较准确的初值x_0，还应该给定一个初始的优化半径μ_0
优化半径的作用在于使得$\|\boldsymbol{D}\Delta x\|^2 \le \mu$成为约束条件，从而在求解$\Delta x$时，构造一个有界优化

另外，每次在更新时，根据ρ值的大小，需要同时优化半径μ和估计量x

我们把增量限定于一个半径为 μ 的球中，认为只在这个球内才是有效的。带上 $\boldsymbol{D}$ 之后，这个球可以看成一个椭球。在 Levenberg 提出的优化方法中，把 $\boldsymbol{D}$ 取成单位阵 $\boldsymbol{I}$，相当于直接把 $\Delta \boldsymbol{x}$ 约束在一个球中。随后， Marqaurdt 提出将 $\boldsymbol{D}$ 取成非负数对角阵——实际中通常用 $\boldsymbol{J}_T\boldsymbol{J}$ 的对角元素平方根，使得在梯度小的维度上约束范围更大一些。

列文伯格-马夸尔特法

1. 给定初始值 $\boldsymbol{x}_0$，以及初始优化半径 μ。
2. 对于第 k 次迭代，求解：

带不等式约束的优化问题

$$\min_{\Delta \boldsymbol{x}_k} \frac{1}{2}\left\|f\left(\boldsymbol{x}_k\right)+\boldsymbol{J}\left(\boldsymbol{x}_k\right) \Delta \boldsymbol{x}_k\right\|^2, \quad s.t. \left\|\boldsymbol{D} \Delta \boldsymbol{x}_k\right\|^2 \leq \mu,$$

   这里 μ 是信赖区域的半径

3. 计算 ρ。
4. 若 $\rho > \frac{3}{4}$，则 $\mu = 2\mu$；
5. 若 $\rho < \frac{1}{4}$，则 $\mu = 0.5\mu$；
6. 如果 ρ 大于某阈值，认为近似可行。令 $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta \boldsymbol{x}_k$。
7. 判断算法是否收敛。如不收敛则返回 2，否则结束。

列文伯格-马夸尔特法

$$\min_{\Delta \boldsymbol{x}_k} \frac{1}{2}\left\|f\left(\boldsymbol{x}_k\right)+\boldsymbol{J}\left(\boldsymbol{x}_k\right) \Delta \boldsymbol{x}_k\right\|^2, \quad s.t. \left\|\boldsymbol{D} \Delta \boldsymbol{x}_k\right\|^2 \leq \mu,$$

带不等式约束的优化问题可利用拉格朗日乘子转化为无约束的优化问题：

$$\min_{\Delta \boldsymbol{x}_k} \frac{1}{2}\left\|f\left(\boldsymbol{x}_k\right)+\boldsymbol{J}\left(\boldsymbol{x}_k\right) \Delta \boldsymbol{x}_k\right\|^2+\frac{\lambda}{2}\|\boldsymbol{D} \Delta \boldsymbol{x}\|^2 .$$

接下来的步骤和GN一样，范数展开然后计算增量方程：

$$\left(\boldsymbol{H}+\lambda \boldsymbol{D}^T \boldsymbol{D}\right) \Delta \boldsymbol{x}=\boldsymbol{g} .$$

3

讨论与交流

（假期过后如何快速进入科研状态）

欢迎关注3D视觉工坊

公众号

我们这里有3D视觉算法、SLAM、点云处理、三维重建、计算机视觉、深度学习、自动驾驶、图像处理、技术干货以及前沿paper分享！

交流群请添加客服

如果你也想成为主讲人，欢迎加入我们。

➢ 报名方式：请发送邮件至vision3d@yeah.net

微信扫码加入星球

客服微信，咨询课程

3D视觉工坊知识星球

- 课程PPT和注释代码
- 补充知识点　PDF版和视频版
- 答疑

感谢聆听

Thanks for Listening