

DSO windowed optimization 代码 (4)

5 “step”计算

参考《DSO windowed optimization 公式》，计算各个优化变量的增加量。

公式再写一下：

$$\begin{bmatrix} H_{\rho\rho} & H_{\rho X} \\ H_{X\rho} & H_{XX} \end{bmatrix} \begin{bmatrix} \delta\rho \\ \delta X \end{bmatrix} = - \begin{bmatrix} J_{\rho}^T r \\ J_X^T r \end{bmatrix}$$
$$\begin{bmatrix} H_{\rho\rho} & H_{\rho X} \\ 0 & H_{XX} - H_{X\rho}H_{\rho\rho}^{-1}H_{\rho X} \end{bmatrix} \begin{bmatrix} \delta\rho \\ \delta X \end{bmatrix} = - \begin{bmatrix} J_{\rho}^T r \\ J_X^T r - H_{X\rho}H_{\rho\rho}^{-1}J_{\rho}^T r \end{bmatrix}$$

我们的目标是用上面的第二个方程

$$(H_{XX} - H_{X\rho}H_{\rho\rho}^{-1}H_{\rho X})\delta X = -(J_X^T r - H_{X\rho}H_{\rho\rho}^{-1}J_{\rho}^T r)$$

计算出  $\delta X$ ，再代回第一个方程

$$H_{\rho\rho}\delta\rho + H_{\rho X}\delta X = -J_{\rho}^T r$$

计算  $\delta\rho$ 。

5.1  $\delta X$  计算

这里 `ldlt` 计算

$$(H_{XX} - H_{X\rho}H_{\rho\rho}^{-1}H_{\rho X})(-\delta X) = J_X^T r - H_{X\rho}H_{\rho\rho}^{-1}J_{\rho}^T r$$

的结果 $-\delta X$ 。少了一个负号，所以后面在函数 `EnergyFunctional::resubstituteF_MT` 的计算内参增量和计算帧增量，加了一个负号。这里不要犯迷糊，一开始什么不懂的时候，认为这里 `Engel` 写错了。

这个计算还是很清晰的。

5.2  $\delta\rho$  计算

整理一下，我们要计算的方程是这个：

$$\delta\rho = -H_{\rho\rho}^{-1}(J_{\rho}^T r + H_{\rho X}\delta X)$$

$-H_{\rho\rho}^{-1}$  是一个对角阵，如果看上面方程的一行，得到的结果是

$$\delta\rho^{(j)} = -\left(\sum_{i=1}^N \frac{\partial r^{(i)}}{\partial\rho^{(j)}}^T \frac{\partial r^{(i)}}{\partial\rho^{(j)}}\right)^{-1} \left(\sum_{i=1}^N \frac{\partial r^{(i)}}{\partial\rho^{(j)}}^T r^{(i)} + \sum_{i=1}^N \frac{\partial r^{(i)}}{\partial\rho^{(j)}}^T \left(\frac{\partial r^{(i)}}{\partial C}\delta C + \frac{\partial r^{(i)}}{\partial X_t}\delta X_t + \frac{\partial r^{(i)}}{\partial X_h}\delta X_h\right)\right)$$

(如果  $r^{(i)}$  与  $\rho^{(j)}$  没有关系，导数  $\frac{\partial r^{(i)}}{\partial\rho^{(j)}}$  为 0。)

这个计算比较麻烦，计算过程在函数 `EnergyFunctional::resubstituteFPt` 中，首先在 `EnergyFunctional::resubstituteF_MT` 的这里 准备 `xAd` 数组，这个数组的[h,t]是

$$-\delta X_h^T \frac{\partial X_{th}}{\partial X_h}^T - \delta X_t^T \frac{\partial X_{th}}{\partial X_t}^T$$

嗯，事先把 `adjoint` 导数转换准备好。

接着在这里几行计算  $\delta\rho^{(j)}$ ：

```
float b = p->bdSumF;
b -= xc.dot(p->Hcd_accAF + p->Hcd_accLF);

for(EFResidual* r : p->residualsAll)
{
    if(!r->isActive()) continue;
    b -= xAd[r->hostIDX*nFrames + r->targetIDX] * r->JpJdF;
}

p->data->step = - b*p->HdiF;
```

`p->bdSumF` 对应  $\sum_{i=1}^N \frac{\partial r^{(i)}}{\partial\rho^{(j)}}^T r^{(i)}$ 。

`p->Hcd_accAF + p->Hcd_accLF` 对应  $\sum_{i=1}^N \frac{\partial r^{(i)}}{\partial C}^T \frac{\partial r^{(i)}}{\partial\rho^{(j)}}$ 。


`r->JpJdF` 对应  $\frac{\partial r^{(i)}}{\partial X_{th}}^T \frac{\partial r^{(i)}}{\partial\rho^{(j)}}$ 。

结果就出来了。

分类: [SLAM](#)

好文要顶 关注我 收藏该文

🔥 🗨️

 JingeTU  
关注 - 4  
粉丝 - 71

加关注

« 上一篇: [Adjoint of SE\(3\)](#)  
» 下一篇: [PnP 问题方程怎么列?](#)

posted @ 2018-06-08 21:25 JingeTU 阅读(1063) 评论(2) 编辑 收藏

评论列表

#1楼 2019-03-08 11:46 努力努力君


“ 博主您好，这段时间在看dso，很幸运看到您写的博客，受益匪浅。  
这篇中提到  $\delta X$  的计算，我有些疑问: `ldlt`是对矩阵的分解吧，代码中是通过它直接计算出 $\delta X$ 的吗? 看了下链接处，代码中的计算结果  $\delta X$ 是用 $x$ 表示? 不太懂这行代码所表达的信息: `x = SVecI.asDiagonal() * HFinalScaled.ldlt().solve(SVecI.asDiagonal() * bFinal_top)`; 看不出每部分具体对应哪一项功能，想了解  $\delta X$  的整个计算过程。十分期待博主的解惑!

支持(0) 反对(0)

#2楼 [楼主 🧑🏻] 2019-03-24 14:58 JingeTU

“ @ 努力努力君  
你好。  
1. `HFinalScaled.ldlt()` 可以查 `eigen` 的文档，是一种 `Cholesky` 矩阵分解，将 `HFinalScaled` 分解为 `L D L^T` 的形式，方便 `x` 的求解。`L` 指 `lower triangular matrix`，`D` 指 `diagonal matrix`，出现了三角阵就可以分步求解 `x` 的各个元素。`H x = b => L (D L^T x) = b`，令 `D L^T x = y`，则 `L y = b`，容易求解 `y`（依次求解 `y_1, y_2, ..., y_n`）。`D L^T x = y => L^T x = D^{-1} y`，`D` 是对角阵，对角元素求倒数，就是逆。接着算 `x`（依次求解 `x_n, x_{n-1}, ..., x_1`）。  
2. `SVecI.asDiagonal()` 是协方差矩阵，这是一个对角阵，也就是只存在方差，认为协方差为 0。协方差阵是对各项误差的加权，最终最小化各项误差的加权和。  
3. 你在代码中看到的 `1 + lambda` 是 `LM` 方法。

支持(0) 反对(0)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)， [访问](#) 网站首页。

【推荐】 了解你才能更懂你，博客园首发问卷调查，助力社区新升级  
【推荐】 超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库  
【推荐】 技术人必备的17组成长笔记+1500道面试题

立即参与

有道智云·AI开放平台

有道智云周年庆  
API服务大放送!

服务类型

文本翻译、图片翻译、语音翻译  
OCR识别、语音合成、语音识别  
语音评测、拍照搜题、整页拍摄

相关博文：  
· [DSO windowed optimization 公式](#)  
· [DSO 优化代码中的 Schur Complement](#)  
· [DSO windowed optimization 代码 \(1\)](#)  
· [直接法光度误差导数推导](#)  
· [AdjointofSE\(3\)](#)  
» 更多推荐...

最新 IT 新闻：  
· 日烧6000万 免费模式能否颠覆To B市场?  
· 明星直播带不动货，销售额水分高达99%  
· 腾讯云与工业富联合作升级，联手打造新基建领域的数字“灯塔工厂”标杆  
· 抖音出海，可能比华为更难  
· 理想VS蔚来，首轮PK谁赢了?  
» 更多新闻...

公告

昵称: JingeTU  
园龄: 3年5个月  
粉丝: 71  
关注: 4  
+加关注

< 2020年7月 >						
日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

搜索

找找看

谷歌搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

随笔分类

LeetCode(7)  
SLAM(38)

随笔档案

2020年5月(2)  
2019年12月(1)  
2019年10月(3)  
2019年8月(1)  
2019年5月(1)  
2019年4月(1)  
2018年6月(1)  
2018年5月(2)  
2018年3月(3)  
2018年1月(5)  
2017年11月(1)  
2017年10月(1)  
2017年9月(5)  
2017年8月(1)  
2017年3月(5)  
2017年2月(13)

最新评论

1. Re:直接法光度误差导数推导  
@OldYangtze 求导链式法则，公式 (34) 最后一行。你是问这? ...  
--JingeTU

2. Re:直接法光度误差导数推导  
大佬（3 6）公式是怎么得出来的?  
--OldYangtze

3. Re:DSO 代码框架  
大神你好，请教一个问题。在 `trackNewestCoarse()`这个函数中有一句 `float extrapFac = 1; if(lambda < lambdaExtrapolationLimit) ...`  
--胡公子

4. Re:DSO windowed optimization 公式  
@daichujiang 没有计算吗? 你看这一行。具体的你去确认这附近的代码，我很久没看了。...  
--JingeTU

5. Re:DSO windowed optimization 公式  
@JingeTU 是我理解有问题，这里ksi包含8个参数（se(3)和两个光度参数）。这里还有个问题，相对光度参数delta(bij)对host帧光度参数的伴随，代码中只求取了对bi的导数，而没有求对...  
--daichujiang

阅读排行榜

1. 矩阵求导的思考(6054)
2. DSO 代码框架(5954)
3. IMU 预积分推导(5464)
4. ORB\_SLAM2 源码阅读 ORB\_SLAM2::ORBextractor(4262)
5. 直接法光度误差导数推导(3545)

评论排行榜

1. DSO 代码框架(10)
2. DSO windowed optimization 公式(9)
3. IMU 预积分推导(6)
4. Adjoint of SE(3)(5)
5. 直接法光度误差导数推导(5)

推荐排行榜

1. DSO 代码框架(4)
2. 【SLAM】安装 g2o\_viewer(3)
3. 直接法光度误差导数推导(3)
4. DSO windowed optimization 代码 (1)(2)
5. DSO windowed optimization 代码 (3)(2)