

# 第五节课习题

高洪臣

2019 年 7 月 18 日

## 基础题

1. 完成单目 Bundle Adjustment 求解器 problem.cc 中的部分代码

- 完成 Problem::MakeHessian() 中信息矩阵 H 的计算

```
// 按照代码顺序
// part 1
H.block(index_i, index_j, dim_i, dim_j).noalias() += hessian;
// part 2
H.block(index_j, index_i, dim_j, dim_i).noalias() += hessian.transpose();
```

- 完成 Problem::SolveLinearSystem() 中 SLAM 问题的求解

```
// 按照代码顺序
// part 1
MatXX Hmm = Hessian_.block(reserve_size, reserve_size, marg_size, marg_size);
MatXX Hpm = Hessian_.block(0, reserve_size, reserve_size, marg_size);
MatXX Hmp = Hessian_.block(reserve_size, 0, marg_size, reserve_size);
VecX bpp = b_.segment(0, reserve_size);
VecX bmm = b_.segment(reserve_size, marg_size);
// part 2
H_pp_schur_ = Hessian_.block(0, 0, reserve_size, reserve_size) - tempH * Hmp;
b_pp_schur_ = bpp - tempH * bmm;
// part 3
delta_x_ll = Hmm_inv * (bmm - Hmp * delta_x_pp);
```

2. 完成滑动窗口算法测试函数

- 完成 Problem::TestMarginalize() 中的代码, 并通过测试

```
// 按照代码顺序
// part 1
H_marg.block(idx, 0, reserve_size - idx - dim, reserve_size) = temp_botRows;
H_marg.block(reserve_size - dim, 0, dim, reserve_size) = temp_rows;
// part 2
Eigen::MatrixXd Arm = H_marg.block(0, n2, n2, m2);
Eigen::MatrixXd Amr = H_marg.block(n2, 0, m2, n2);
Eigen::MatrixXd Arr = H_marg.block(0, 0, n2, n2);
```

3. 运行结果 (完整工程代码见 code 文件夹)

```

Compare MonoBA results after opt...
after opt, point 0 : gt 0.220938 ,noise 0.227057 ,opt 0.220992
after opt, point 1 : gt 0.234336 ,noise 0.314411 ,opt 0.234854
after opt, point 2 : gt 0.142336 ,noise 0.129703 ,opt 0.142666
after opt, point 3 : gt 0.214315 ,noise 0.278486 ,opt 0.214502
after opt, point 4 : gt 0.130629 ,noise 0.130064 ,opt 0.130562
after opt, point 5 : gt 0.191377 ,noise 0.167501 ,opt 0.191892
after opt, point 6 : gt 0.166836 ,noise 0.165906 ,opt 0.167247
after opt, point 7 : gt 0.201627 ,noise 0.225581 ,opt 0.202172
after opt, point 8 : gt 0.167953 ,noise 0.155846 ,opt 0.168029
after opt, point 9 : gt 0.21891 ,noise 0.209697 ,opt 0.219314
after opt, point 10 : gt 0.205719 ,noise 0.14315 ,opt 0.205995
after opt, point 11 : gt 0.127916 ,noise 0.122109 ,opt 0.127908
after opt, point 12 : gt 0.167904 ,noise 0.143334 ,opt 0.168228
after opt, point 13 : gt 0.216712 ,noise 0.18526 ,opt 0.216866
after opt, point 14 : gt 0.180009 ,noise 0.184249 ,opt 0.180036
after opt, point 15 : gt 0.226935 ,noise 0.245716 ,opt 0.227491
after opt, point 16 : gt 0.157432 ,noise 0.176529 ,opt 0.157589
after opt, point 17 : gt 0.182452 ,noise 0.14729 ,opt 0.182444
after opt, point 18 : gt 0.155701 ,noise 0.182258 ,opt 0.155769
after opt, point 19 : gt 0.14646 ,noise 0.240649 ,opt 0.14677
----- pose translation -----
translation after opt: 0 :-0.000478009 0.00115904 0.000366508 || gt: 0 0 0
translation after opt: 1 :-1.06959 4.00018 0.863877 || gt: -1.0718 4 0.866025
translation after opt: 2 :-4.00232 6.92678 0.867244 || gt: -4 6.9282 0.866025
----- TEST Marg: before marg-----
100 -100 0
-100 136.111 -11.1111
0 -11.1111 11.1111
----- TEST Marg: 将变量移动到右下角-----
100 0 -100
0 11.1111 -11.1111
-100 -11.1111 136.111
----- TEST Marg: after marg-----
26.5306 -8.16327
-8.16327 10.2041

```

图 1 没有 fix 第一帧和第二帧

```

Compare MonoBA results after opt...
after opt, point 0 : gt 0.220938 ,noise 0.227057 ,opt 0.220909
after opt, point 1 : gt 0.234336 ,noise 0.314411 ,opt 0.234374
after opt, point 2 : gt 0.142336 ,noise 0.129703 ,opt 0.142353
after opt, point 3 : gt 0.214315 ,noise 0.278486 ,opt 0.214501
after opt, point 4 : gt 0.130629 ,noise 0.130064 ,opt 0.130511
after opt, point 5 : gt 0.191377 ,noise 0.167501 ,opt 0.191539
after opt, point 6 : gt 0.166836 ,noise 0.165906 ,opt 0.166965
after opt, point 7 : gt 0.201627 ,noise 0.225581 ,opt 0.201859
after opt, point 8 : gt 0.167953 ,noise 0.155846 ,opt 0.167965
after opt, point 9 : gt 0.21891 ,noise 0.209697 ,opt 0.218834
after opt, point 10 : gt 0.205719 ,noise 0.14315 ,opt 0.205683
after opt, point 11 : gt 0.127916 ,noise 0.122109 ,opt 0.127751
after opt, point 12 : gt 0.167904 ,noise 0.143334 ,opt 0.167924
after opt, point 13 : gt 0.216712 ,noise 0.18526 ,opt 0.216885
after opt, point 14 : gt 0.180009 ,noise 0.184249 ,opt 0.179961
after opt, point 15 : gt 0.226935 ,noise 0.245716 ,opt 0.227114
after opt, point 16 : gt 0.157432 ,noise 0.176529 ,opt 0.157529
after opt, point 17 : gt 0.182452 ,noise 0.14729 ,opt 0.1823
after opt, point 18 : gt 0.155701 ,noise 0.182258 ,opt 0.155627
after opt, point 19 : gt 0.14646 ,noise 0.240649 ,opt 0.146533
----- pose translation -----
translation after opt: 0 :0 0 0 || gt: 0 0 0
translation after opt: 1 : -1.0718 4 0.866025 || gt: -1.0718 4 0.866025
translation after opt: 2 :-3.99917 6.92852 0.859878 || gt: -4 6.9282 0.866025
----- TEST Marg: before marg-----
100 -100 0
-100 136.111 -11.1111
0 -11.1111 11.1111
----- TEST Marg: 将变量移动到右下角-----
100 0 -100
0 11.1111 -11.1111
-100 -11.1111 136.111
----- TEST Marg: after marg-----
26.5306 -8.16327
-8.16327 10.2041

```

图 2 fix 第一帧和第二帧

## 提升题

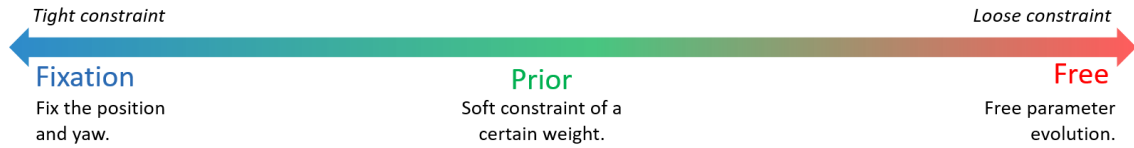
paper reading [1], 请总结论文：优化过程中处理 H 自由度的不同操作方式。总结内容包括：具体处理方式，实验效果，结论。

答：

单目 VO 有 7 个不可观的自由度（双目 VO 有 6 个），VIO 有 4 个不可观的自由度（全局位置和 yaw 不可观测），称为**规范自由度 (gauge freedom)**。

VIO 优化时，需要特别处理这 4 个自由度，通常有三种方法：固定这四个自由度、给这四个自由度加先验、任意优化这四个自由度最后 reset。

### 1. 具体处理方式



#### (a) Gauge Fixation: fixing the initial state

固定第一帧相机的位置  $p_0$  和偏航角 yaw，将其设定为常量，限制规范自由度

$$p_0 = p_0^0, \quad \Delta\phi_{0z} = 0$$

等价于，对应的视觉残差的雅克比矩阵为  $\mathbf{0}$

$$J_{p_0} = \mathbf{0}, \quad J_{\Delta\phi_{0z}} = \mathbf{0}$$

#### (b) Gauge Prior: adding a prior to the initial state

使用合适的先验协方差矩阵  $\Sigma_0^P$ ，给视觉残差最小二乘添加惩罚

$$\|r_0^P\|_{\Sigma_0^P}^2$$

通常，

$$\Sigma_0^P = \sigma_0^2 I$$

#### (c) Free Gauge: allowing the parameters to evolve freely during optimization

使用伪逆或添加阻尼项（LM）处理 Hessian 矩阵，然后再将所有帧重置伪逆

$$\Delta x = H^+ b$$

添加阻尼项

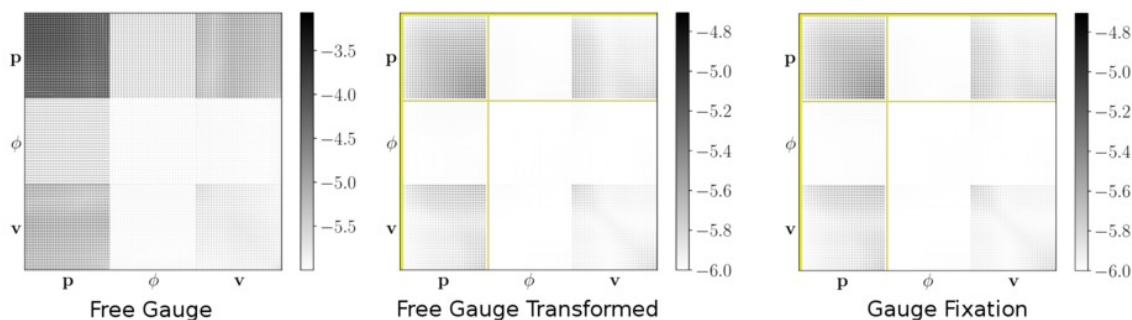
$$\Delta x = (H + \mu\lambda)^{-1} b$$

### 2. 实验效果

选择合适的先验权重：

- Accuracy：当先验权重超过某个阈值后，估计误差的 RMSE 稳定在一个值
- Computational Cost：当先验权重超过某个阈值后，迭代次数和收敛时间也会稳定

选择合适的先验权重后，**Gauge Prior** 和 **Gauge Fixation** 的精度和计算代价几乎相同。协方差比较：



### 3. 结论

- **Accuracy**: 三种方式精度都差不多
- **Efficiency**: **Free Gauge**, 即任意优化然后 reset 的操作, 收敛所需要的迭代次数较少, 速度快一点, 效率较高; 不过没有一个参考系, 信息矩阵的逆得到的协方差没有太多的意义; **Gauge Prior** 需要选择合适的先验协方差权重, 来避免计算代价的增加
- **Covariance**: 三种方式的协方差是紧密相关的, free gauge 的协方差可通过线性变换变为 gauge fixation 的协方差

### 4. 代码

协方差变换的代码: `rpg_vi_cov_transformation`

## 参考文献

- [1] Zichao Zhang, Guillermo Gallego, and Davide Scaramuzza. On the comparison of gauge freedom handling in optimization-based visual-inertial state estimation. *IEEE Robotics and Automation Letters*, 3(3):2710–2717, 2018.