深蓝学院
shenlanxueyuan.com

第十四章：后端优化实践：逐行手写求解器作业思路提示

主讲人 Horizon

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：作业完成情况
- 第二部分：作业内容提示

作业完成情况

- 基础编程题：

  大家都完成得不错。

- 总结论文：

  对三种不同方法的做法和结果分析总结都很用心。

- 提升编程题（为前两帧添加先验约束）：

  为前两帧增加先验，和固定前两帧，是两个不同的做法。

纲要

- 第一部分：作业完成情况
- 第二部分：作业内容提示

作业基础题

完成单目 BA 求解器 problem.cc 中的部分代码。

- 完成 Problem::MakeHessian() 中信息矩阵 H 的计算。
- 完成 Problem::SolveLinearSystem() 中 SLAM 问题的求解。

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。

位置 - MakeHessian()

```
MatXX hessian = JtW * jacobian_j;
// 所有的信息矩阵叠加起来
H.block(index_i, index_j, dim_i, dim_j).noalias() += hessian;    // TODO:: home work
if (j != i) {
    // 对称的下三角
    H.block(index_j, index_i, dim_j, dim_i).noalias() += hessian.transpose();    // TODO:: home work
}
```

位置 - SolveLinearSystem()

```
/* Problem::SetOrdering 中定义的 pose 在前, landmark 在后 */
MatXX Hmm = Hessian_.block(reserve_size, reserve_size, marg_size, marg_size);   // TODO:: home work
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);
```

```
// 计算 schur 补
MatXX tempH = Hpm * Hmm_inv;
H_pp_schur_ = Hessian_.block(0, 0, reserve_size, reserve_size) - tempH * Hmp;   // TODO:: home work
b_pp_schur_ = bpp - tempH * bmm;
```

```
VecX delta_x_ll(marg_size);
/* 将求解出来的 delta_x_pp 代回到分块矩阵中, 构造 delta_x_ll 的方程, 求解 delta_x_ll */
delta_x_ll = PCGSolver(Hmm, bmm - Hmp * delta_x_pp, n); // TODO:: home work
delta_x_.tail(marg_size) = delta_x_ll;
```

位置 - TestMarginalize()

```cpp
// 准备工作:  move the marg pose to the Hmm bottown right
// 将 row i 移动矩阵最下面
Eigen::MatrixXd temp_rows = H_marg.block(idx, 0, dim, reserve_size);    // TODO:: home work
Eigen::MatrixXd temp_botRows = H_marg.block(idx + dim, 0, reserve_size - idx - dim, reserve_size);
/* temp_rows 是矩阵 H_marg 的第 i 行, temp_botRows 是矩阵 H_marg 的第 i+1 行一直到最后一行 */
H_marg.block(reserve_size - dim, 0, dim, reserve_size) = temp_rows;
H_marg.block(idx, 0, reserve_size - idx - dim, reserve_size) = temp_botRows;

// 将 col i 移动矩阵最右边
Eigen::MatrixXd temp_cols = H_marg.block(0, idx, reserve_size, dim);
Eigen::MatrixXd temp_rightCols = H_marg.block(0, idx + dim, reserve_size, reserve_size - idx - dim);
H_marg.block(0, idx, reserve_size, reserve_size - idx - dim) = temp_rightCols;
H_marg.block(0, reserve_size - dim, reserve_size, dim) = temp_cols;
```

```cpp
// 完成舒尔补操作
Eigen::MatrixXd Arm = H_marg.block(0, n2, n2, m2);  // TODO:: home work
Eigen::MatrixXd Amr = H_marg.block(n2, 0, m2, n2);
Eigen::MatrixXd Arr = H_marg.block(0, 0, n2, n2);
```

作业提升题第一题

- 请总结论文 “On the comparison of gauge freedom handling in optimization-based visual-inertial state estimation”：优化过程中处理 H 自由度的不同操作方式。
- 内容包括：具体处理方式，实验效果，结论。
- 目的：讨论使用最小二乘法解决 VIO 问题时处理不可观自由度（Gauge Freedom）的三种处理方法和性能对比。
- 不可观自由度在 VIO 问题中为全局位置和全局航向角。

作业提升题第一题

深蓝学院 shenlanxueyuan.com

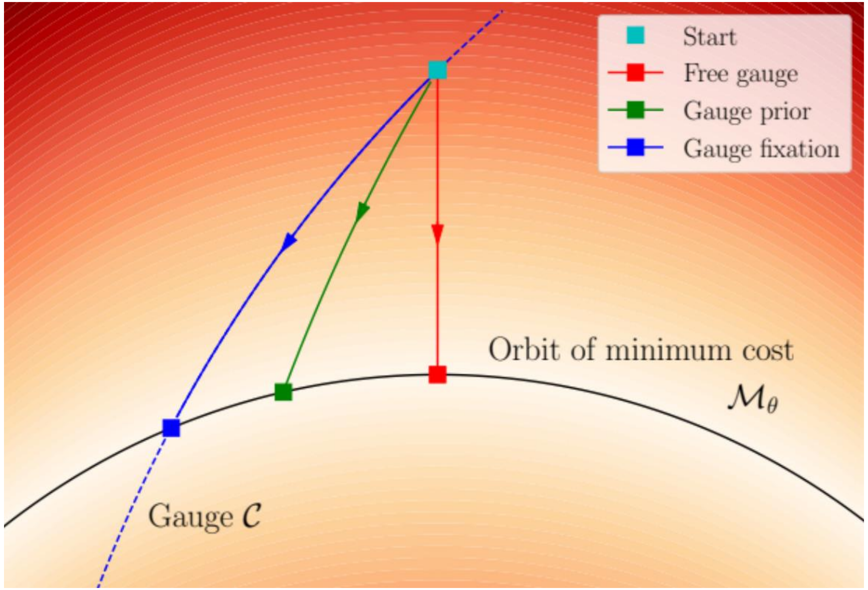
处理不可观自由度的三种方法：

- Fixed gauge：固定第一帧的位置和航向角，这相当于引入了新的测量信息，使得原本不可观的状态变得可观，并且使得信息矩阵满秩，可求逆得到唯一最优解；
- Free gauge：不处理不可观状态，得到的解会在其零空间内漂移。信息矩阵不满秩，可以采用广义逆得到最小范数的最小二乘解；
- Gauge prior：介于上面两种方法之间，通过改变残差的权重，添加额外的信息以处理不可观状态，使得信息矩阵满秩。

作业提升题第一题

三种方法的图形化对比：

- Fixed gauge：将解限定在某一个流形上，得到最优解；
- Free gauge：获得最小范数的最小二乘解，即“垂线段”最短；
- Gauge prior：介于上面两种方法之间。



作业提升题第一题

三种方法的具体操作：

- Fixed gauge：对应雅可比矩阵清零，或者将状态量不可观的变量直接剔除，或者将 Gauge prior 的先验权重设置为无穷大；
- Free gauge：使用 Moore-Penrose 广义逆求解得到范数最小的最小二乘解，或者给 H 矩阵添加阻尼，即 LM 算法，或者将 Gauge prior 的先验权重设置为无穷小；
- Gauge prior：添加先验边（Prior edge），即额外的惩罚项。

作业提升题第一题

三种方法的实验结果与结论：

- 先验权重：不同先验权值对求解精度没有明显影响，但对计算成本有一定影响。需要正确选择权值以保持较小的计算成本。
- 精度与计算成本：三种方法的精度几乎相同。Gauge prior 方法需要正确选择权值以保持较小的计算成本；Free gauge 迭代次数更少，计算更快，可以直接广义逆或者 LM 求解。
- 协方差：Fixed gauge 方法的协方差矩阵中，第一帧不确定度为零，之后依次增加。Free gauge 方法的不确定度平摊到了协方差矩阵中的每一帧，可以采用 Covariance Transformation 将其转换成有意义的形式。

作业提升题第二题

- 在代码中给第一帧和第二帧添加 prior 约束，并比较为 prior 设定不同权重时，BA 求解收敛精度和速度。
- 代码仿真的是单目视觉系统，因此要同时给前两帧添加先验约束。源码中已经提供了 EdgeSE3Prior 先验边，只需要仿照其他残差在前两帧中添加先验边即可。

作业提升题第二题

- Type edge_prior(…) // 使用智能指针初始化先验边
- Something->SetVertex(…) // 对边设置顶点
- Something->SetInformation(…) // 对边设置信息矩阵（权重）
- Something.AddEdge(…) // 为待求解问题添加边

作业提升题第二题

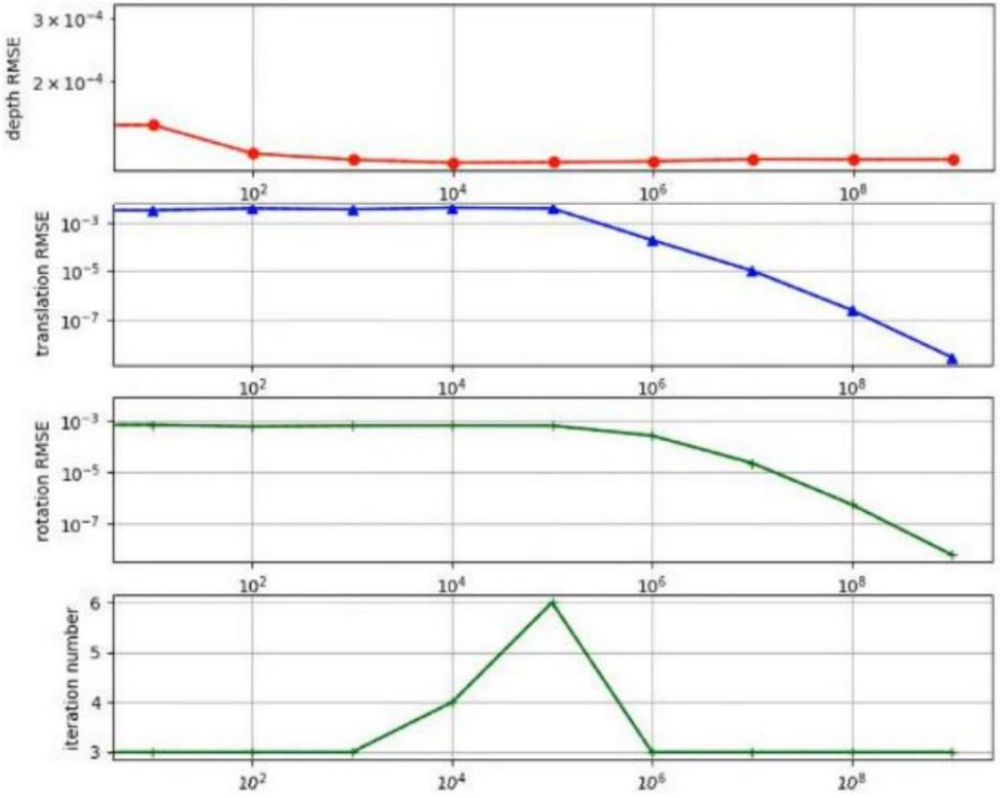
SE3 的先验边的残差计算与雅可比矩阵推导：

$$r^{prior} = \begin{bmatrix} r_R^{prior} \\ r_p^{prior} \end{bmatrix} = \begin{bmatrix} \Delta\phi \\ p - p^0 \end{bmatrix} = \begin{bmatrix} \ln\left(\left(R^0\right)^{-1} R\right)^{\vee} \\ p - p^0 \end{bmatrix}$$

$$J_{r^{prior}} = \frac{\partial r^{prior}}{\partial \begin{bmatrix} R \\ p \end{bmatrix}} = \begin{bmatrix} \frac{\partial r_R^{prior}}{\partial R} & \frac{\partial r_R^{prior}}{\partial p} \\ \frac{\partial r_p^{prior}}{\partial R} & \frac{\partial r_p^{prior}}{\partial p} \end{bmatrix} = \begin{bmatrix} J_r^{-1}\left(\ln\left(\left(R^0\right)^{-1} R\right)^{\vee}\right) & 0 \\ 0 & I \end{bmatrix}$$

作业提升题第二题

随着权重的增加，存在一个峰值：



深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening