深蓝学院
shenlanxueyuan.com

第五章作业提示

主讲人 会打篮球的猫

纲要

深蓝学院
shenlanxueyuan.com

- **第一部分：概述**
- 第二部分：基础题一
- 第三部分：基础题二
- 第三部分：提升题——论文总结
- 第四部分：提升题——添加先验

概述

- 第五章基本上是对之前两章的巩固与应用（重点关注如下几个问题）

——最小二乘的求解步骤？

①、信息矩阵如何构建？

$$\sum_{i=1}^{n} \mathbf{J}_i^\top \boldsymbol{\Sigma}_i^{-1} \mathbf{J}_i \delta\boldsymbol{\xi} = -\sum_{i=1}^{n} \mathbf{J}^\top \boldsymbol{\Sigma}_i^{-1} \mathbf{r}$$

连加

②、如何利用信息矩阵的稀疏性加速计算？

$$\begin{bmatrix} \mathbf{H}_{\mathrm{pp}} & \mathbf{H}_{\mathrm{pl}} \\ \mathbf{H}_{\mathrm{lp}} & \mathbf{H}_{\mathrm{ll}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_{\mathrm{p}}^{*} \\ \Delta\mathbf{x}_{\mathrm{l}}^{*} \end{bmatrix} = \begin{bmatrix} -\mathbf{b}_{\mathrm{p}} \\ -\mathbf{b}_{\mathrm{l}} \end{bmatrix}$$

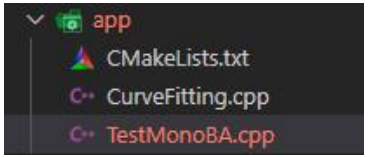
舒尔补

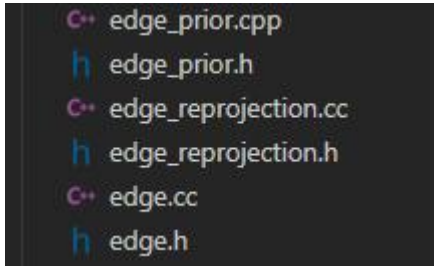
——边缘化时对信息矩阵是如何操作的？

变换行列、舒尔补

概述

- 多读代码，不仅仅是作业 TODO 所涉及的部分





纲要

- 第一部分：概述
- **第二部分：基础题一**
- 第三部分：基础题二
- 第三部分：提升题——论文总结
- 第四部分：提升题——添加先验

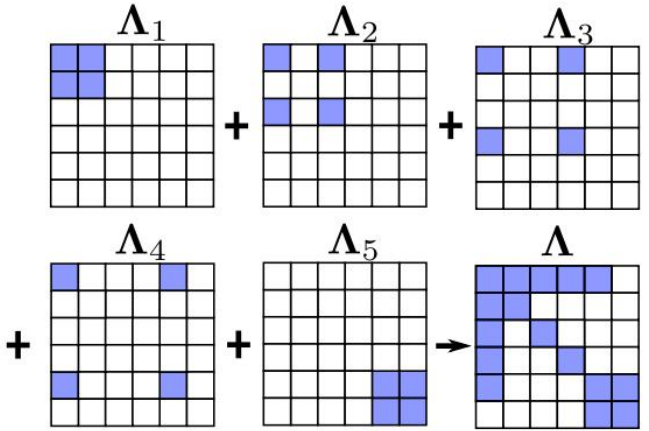
基础题一

深蓝学院 shenlanxueyuan.com

1. 完成单目 Bundle Adjustment (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。

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



基础题一

深蓝学院 shenlanxueyuan.com

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

```
// step1: schur marginalization --> Hpp, bpp
int reserve_size = ordering_poses_;
int marg_size = ordering_landmarks_;

// TODO:: home work. 完成矩阵块取值，Hmm，Hpm，Hmp，bpp，bmm
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);
```

```
// TODO:: home work. 完成舒尔补 Hpp, bpp 代码
MatXX tempH = Hpm * Hmm_inv;
// H_pp_schur_ = Hessian_.block(?,?,?,?) - tempH * Hmp;
// b_pp_schur_ = bpp - ? * ?;
H_pp_schur_ = Hessian_.block(0, 0, reserve_size, reserve_size) - tempH * Hmp;
b_pp_schur_ = bpp - tempH * bmm;
```

```
// TODO:: home work. step3: solve landmark
VecX delta_x_ll(marg_size);
// delta_x_ll = ???;
delta_x_ll = Hmm_inv * (bmm - Hmp * delta_x_pp);
```

$$\begin{pmatrix} H_{pp} & H_{pm} \\ H_{mp} & H_{mm} \end{pmatrix} \begin{pmatrix} X_p \\ X_l \end{pmatrix} = \begin{pmatrix} b_{pp} \\ b_{mm} \end{pmatrix}$$

$$\begin{pmatrix} I & -H_{pm}H_{mm}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} H_{pp} & H_{pm} \\ H_{mp} & H_{mm} \end{pmatrix} \begin{pmatrix} X_p \\ X_l \end{pmatrix} = \begin{pmatrix} I & -H_{pm}H_{mm}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} b_{pp} \\ b_{mm} \end{pmatrix}$$

$$\begin{pmatrix} H_{pp} - H_{pm}H_{mm}^{-1}H_{mp} & 0 \\ H_{mp} & H_{mm} \end{pmatrix} \begin{pmatrix} X_p \\ X_l \end{pmatrix} = \begin{pmatrix} b_{pp} - H_{pm}H_{mm}^{-1}b_{mm} \\ b_{mm} \end{pmatrix}$$

$$\begin{cases} (H_{pp} - H_{pm}H_{mm}^{-1}H_{mp})X_p = b_{pp} - H_{pm}H_{mm}^{-1}b_{mm} \\ H_{mp}X_p + H_{mm}X_l = b_{mm} \end{cases}$$

$$\begin{cases} X_p \quad \rightarrow \quad PCGSolver \\ X_l = H_{mm}^{-1}(b_{mm} - H_{mp}X_p) \end{cases}$$

基础题一

深蓝学院 shenlanxueyuan.com

- 测试结果与分析

```
lcx@lcx:~/Desktop/vio_homework/ch5/hw_course5_new/build$ ./app/testMonoBA
0 order: 0
1 order: 6
2 order: 12

 ordered_landmark_vertices_ size : 20
iter: 0 , chi= 5.35099 , Lambda= 0.00597396
iter: 1 , chi= 0.0289048 , Lambda= 0.00199132
iter: 2 , chi= 0.000109162 , Lambda= 0.000663774
problem solve cost: 0.609307 ms
   makeHessian cost: 0.345429 ms

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 :-0.00047801  0.00115904 0.000366507 || 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
```

利用LM法优化得到的结果，发现在收敛后，第一帧的相机位置不再是原点（0, 0, 0），这也对应了课件中提到的：求得的结果往零空间变化。

基础题一

深蓝学院 shenlanxueyuan.com

- 测试结果与分析

因此，将前两帧 pose 进行固定，即对应的雅克比矩阵设为 0，从而不去进行更新。

```
if(i < 2)
    vertexCam->SetFixed();
```

固定相机pose的操作增加了系统的可观性，这次结果可以看出，没有再出现上述结果往零空间变化的问题。

```
lcx@lcx:~/Desktop/vio_homework/ch5/hw_course5_new/build$ ./app/testMonoBA
0 order: 0
1 order: 6
2 order: 12

 ordered_landmark_vertices_ size : 20
iter: 0 , chi= 5.35099 , Lambda= 0.00597396
iter: 1 , chi= 0.0282599 , Lambda= 0.00199132
iter: 2 , chi= 0.000117497 , Lambda= 0.000663774
problem solve cost: 0.257207 ms
   makeHessian cost: 0.126673 ms

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
```

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：概述
- 第二部分：基础题一
- **第三部分：基础题二**
- 第三部分：提升题——论文总结
- 第四部分：提升题——添加先验

基础题二

深蓝学院 shenlanxueyuan.com

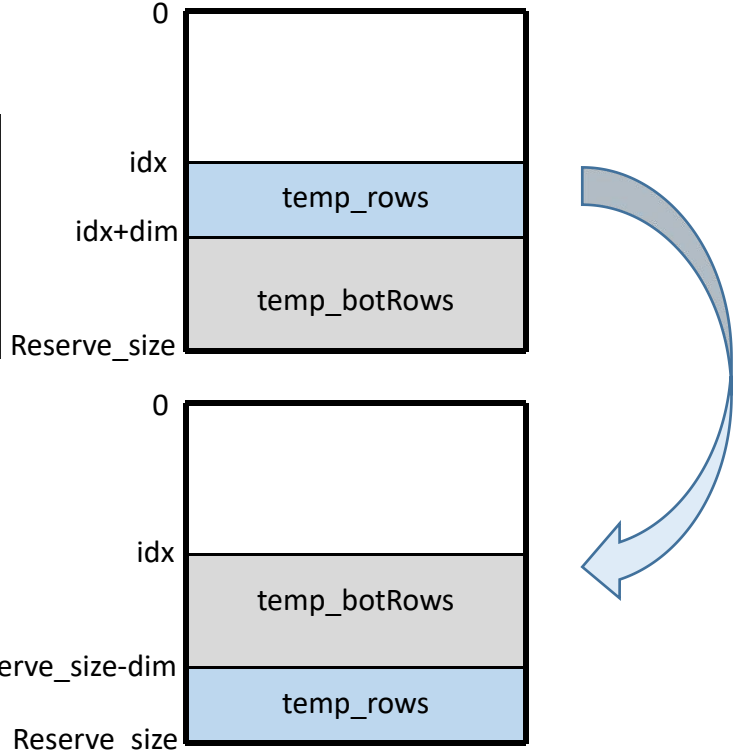
❷ 完成滑动窗口算法测试函数。

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

```
// TODO: home work. 将变量移动到右下角
/// 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);
Eigen::MatrixXd temp_botRows = H_marg.block(idx + dim, 0, reserve_size - idx - dim, reserve_size);
// H_marg.block(?,?,?,?) = temp_botRows;
// H_marg.block(?,?,?,?) = temp_rows;
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;
```

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

Eigen::MatrixXd tempB = Arm * Amm_inv;
Eigen::MatrixXd H_prior = Arr - tempB * Amr;
```



基础题二

深蓝学院 shenlanxueyuan.com

❷ 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。

```
---------- 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
```

- 从上图可以看出，我们将变量 2 在信息矩阵中成功转移到了右下角。
- 在 marg 之前，信息矩阵中的 0 表示变量 1 和 3 关于 2 条件独立，换句话说就是，在我们常画的因子图中，1 和 3 之间是没有连接边的。
- 当 marg 之后，2 留下的先验信息则传递给了 1 和 3，信息矩阵中相应位置不再是 0， 此时 1 和 3 之间产生了链接，这也成功对应了上节课的作业内容。

纲要

- 第一部分：概述
- 第二部分：基础题一
- 第三部分：基础题二
- **第三部分：提升题——论文总结**
- 第四部分：提升题——添加先验

提升题--论文总结

Zhang Z, Gallego G, Scaramuzza D. On the Comparison of Gauge Freedom Handling in Optimization-Based Visual-Inertial State Estimation[J]. IEEE Robotics and Automation Letters, 2018, 3(3):2710-2717.

- gauge freedom ？

  VINS系统中存在着4自由度的不可观量——偏航角yaw、全局位置p^{global}。

the unobservable DoF in VI systems. It is well known that for a VI system, global position and yaw are not observable [3], [10], which in this paper we call *gauge freedom* following the convention from the field of bundle adjustment [11]. Given this

- 它会造成什么影响？

  G-N法求解时，H矩阵是奇异的（不满秩，具有4维的零空间），稳定性较差，可能不收敛。

$$\underbrace{\mathbf{J}^{\top}\boldsymbol{\Sigma}^{-1}\mathbf{J}}_{\mathbf{H}\ or\ \boldsymbol{\Lambda}}\delta\boldsymbol{\xi}=\underbrace{-\mathbf{J}^{\top}\boldsymbol{\Sigma}^{-1}\mathbf{r}}_{\mathbf{b}}$$

提升题--论文总结

深蓝学院 shenlanxueyuan.com

- gauge fixation approach（对应基础题中固定的情况）

  即在优化过程中将第一帧的相机位置和yaw角固定住，等价于将相应参数所对应的雅可比矩阵设为0，因此残差也是0，从而实现不更新。

$$\mathbf{p}_0 = \mathbf{p}_0^0, \qquad \Delta\boldsymbol{\phi}_{0z} \doteq \mathbf{e}_z^\top \Delta\boldsymbol{\phi}_0 = 0, \tag{10}$$
$$\mathbf{J}_{\mathbf{p}_0} = \mathbf{0}, \ \mathbf{J}_{\Delta\boldsymbol{\phi}_{0z}} = \mathbf{0}.$$

- free gauge approach（对应基础题中未固定的情况）

  允许参数在优化时自由演变，使用伪逆或者添加阻尼的方法处理海森矩阵，保证问题有好的参数更新。例如L-M法。

$$(J^T J + \mu I)\,\Delta x = -J^T f$$

- gauge prior approach（对应提升题部分）

  为第一帧相机添加一个残差（惩罚项）。即为第一帧相机位姿增加一个先验约束。

$$\|\mathbf{r}_0^P\|_{\Sigma_0^P}^2, \quad \text{where} \quad \mathbf{r}_0^P(\boldsymbol{\theta}) \doteq (\mathbf{p}_0 - \mathbf{p}_0^0,\ \Delta\boldsymbol{\phi}_{0z}). \tag{11}$$

提升题--论文总结

深蓝学院 shenlanxueyuan.com

gauge prior approach中的权重设定

—— 由于gauge prior法中涉及到协方差矩阵的设定（为先验约束设置不同的权重），协方差一般设定为对线矩阵的形式，即一个权重w乘以单位阵的形式，因此首先讨论不同权重对于gauge prior法的影响。

Before comparing the three approaches from Section III, we need to choose the prior covariance Σ_0^P in the gauge prior approach. A common choice is $\Sigma_0^P = \sigma_0^2 \mathrm{I}$, for which the prior (11) becomes $\|\mathbf{r}_0^P\|_{\Sigma_0^P}^2 = w^P \|\mathbf{r}_0^P\|^2$, with $w^P = 1/\sigma_0^2$. We tested a wide range of the prior weight w^P on different configurations and the results were similar. Therefore, we will

特别注意的是，当w=0时，基本就是free gauge法，而w=无穷时则等价于gauge fixation法。

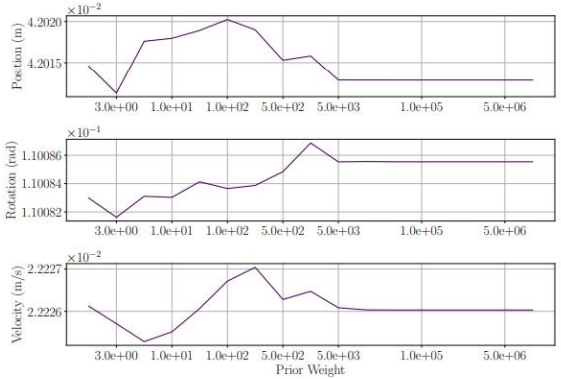


Fig. 4: RMSE in position, orientation and velocity for different prior weights

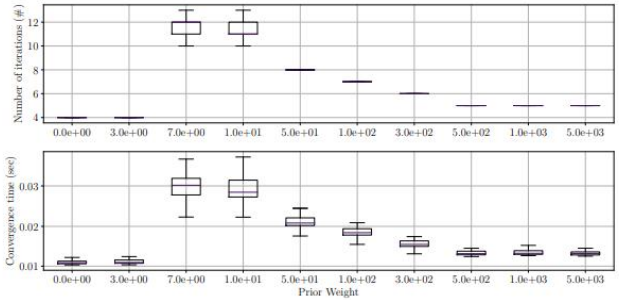


Fig. 5: Number of iterations and computing time for different prior weights.

- 精度

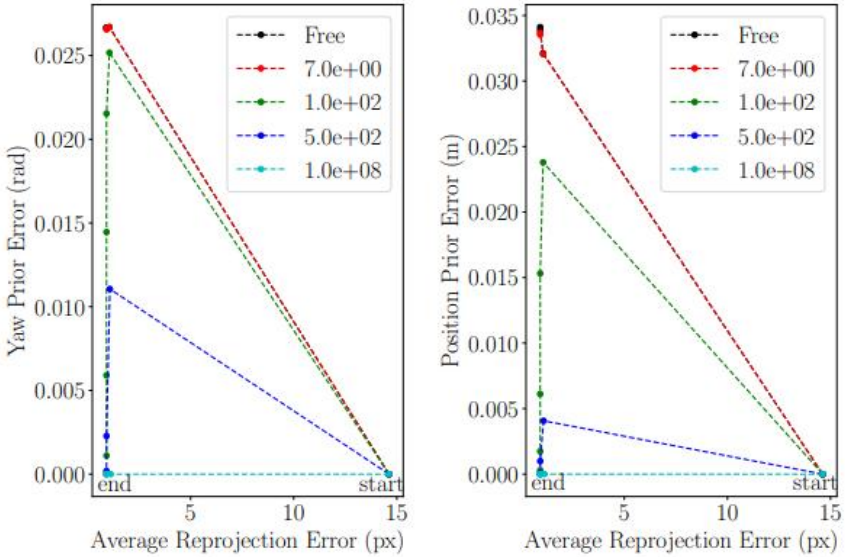
不同权重下所估计的误差是非常相似的，也就是说w的取值对于精度的影响不大。且当w增长至某个值以后，RMSE会稳定在某个值。

- 运行效率

迭代次数以及收敛时间同样会在w达到某个值以后稳定下来。但是w在从0增大的过程中，计算时间会出现一个峰值。

提升题--论文总结

针对上述这个情况，作者做了进一步研究。通过比较不同权重下，在迭代过程中，重投影误差以及先验误差的变化情况，得到如下结论：



当w较大时，在迭代计算过程中，算法在降低重投影误差时，能够保证先验误差几乎等于0。

反过来，当w较小时，在前几次迭代时，会以增加先验误差为代价来降低重投影误差。

优化算法花费多次迭代微调先验误差，同时保证重投影误差较小，因此就增加了计算量。

—— 选择适当大小的权重（其实也就是上述结果稳定以后的w取值）

- 对精度影响不大
- 保证迭代次数少
- 避免很大的权重（优化可能不稳定）

提升题--论文总结

——对比与分析（精度与效率）：

① 首先，三种方法在精度的表现上几乎相同；

② 如上文所说，gauge prior法需要选择一个合适的权重值来避免增加计算量；

③ 在选取合适的权重值的情况下，gauge prior法与gauge fixation法在精度以及计算量上有几乎相同的表现；

④ Free gauge法相比于其他两种速度略快，因为它迭代次数更少。

TABLE II: RMSE on different trajectories and 3D points configurations. The smallest errors (e.g., **p** gauge fixation vs. **p** free gauge) are highlighted.

Configuration	Gauge fixation			Free gauge		
	p	ϕ	**v**	**p**	ϕ	**v**
sine plane	0.04141	0.1084	**0.02182**	0.04141	0.1084	0.02183
arc plane	**0.02328**	0.6987	0.01303	0.02329	0.6987	0.01303
rec plane	**0.01772**	0.1668	0.01496	0.01774	0.1668	**0.01495**
sine random	0.03932	0.0885	0.01902	**0.03908**	**0.0874**	**0.01886**
arc random	0.02680	0.6895	0.01167	**0.02678**	0.6895	**0.01166**
rec random	**0.02218**	0.1330	0.009882	0.02220	0.1330	**0.009881**

Position, rotation and velocity RMSE are measured in m, deg and m/s, respectively.

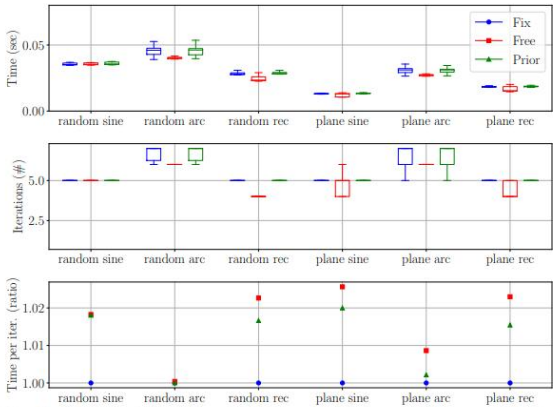


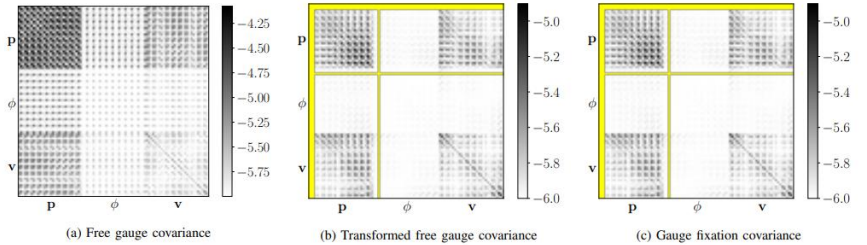
Fig. 7: Number of iterations, total convergence time and time per iteration for all configurations. The time per iteration is the ratio with respect to the gauge fixation approach (in blue), which takes least time per iteration.

提升题--论文总结

深蓝学院 shenlanxueyuan.com

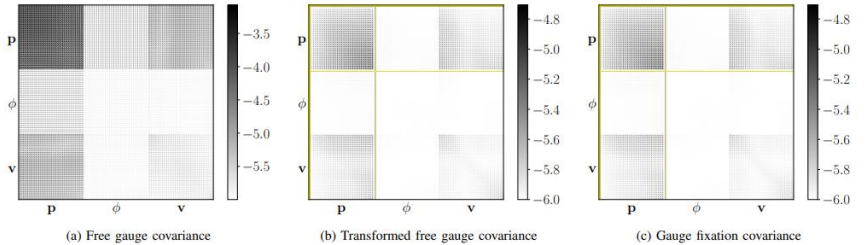
——对比与分析（协方差）：

free gauge法和其他方法相比得到的协方差有很大不同，但是通过线性变换，其协方差能够满足gauge fixation的条件，这也表明了不同方法的协方差是紧密相关的。



(a) Free gauge covariance (b) Transformed free gauge covariance (c) Gauge fixation covariance

Fig. 9: Covariance of free gauge (Fig. 9a) and gauge fixation (Fig. 9c) approaches using $N = 10$ keyframes. In the middle (Fig. 9b), the free gauge covariance transformed using (12) shows very good agreement with the gauge fixation covariance: the relative difference between them is $\|\Sigma_b - \Sigma_c\|_F / \|\Sigma_c\|_F \approx 0.11\%$ ($\|\cdot\|_F$ denotes Frobenius norm). For better visualization, the magnitude of the covariance entries is displayed in logarithmic scale. The yellow bands of the gauge fixation and transformed covariances indicate zero entries due to the fixed 4-DoFs (the position and the yaw angle of the first camera).



(a) Free gauge covariance (b) Transformed free gauge covariance (c) Gauge fixation covariance

Fig. 10: Covariance comparison and transformation using $N = 30$ keyframes of the EuRoC Vicon 1 sequence (VI1). Same color scheme as in Fig. 9. The relative difference between (b) and (c) is $\|\Sigma_b - \Sigma_c\|_F / \|\Sigma_c\|_F \approx 0.02\%$. Observe that, in the gauge fixation covariance, the uncertainty of the first position and yaw is zero, and it grows for the rest of the camera poses (darker color), as illustrated in Fig. 1b.

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：概述
- 第二部分：基础题一
- 第三部分：基础题二
- 第三部分：提升题——论文总结
- **第四部分：提升题——添加先验**

提升题——添加先验

- 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。(加分题，评选优秀)

```
double w = 1e6;
for (size_t i = 0; i < 2; ++i) {
    shared_ptr<EdgeSE3Prior> edge(new EdgeSE3Prior(cameras[i].twc, cameras[i].qwc));
    std::vector<std::shared_ptr<Vertex> > edge_vertex;
    edge_vertex.push_back(vertexCams_vec[i]);
    edge->SetVertex(edge_vertex);
    edge->SetInformation(w * edge->Information());

    problem.AddEdge(edge);
}
```

TestMonoBA.cpp

总体就是仿照上面添加边的代码，为前两帧相机位姿添加先验约束。同 时还要引入头文件。

记得引入头文件。

提升题--添加先验

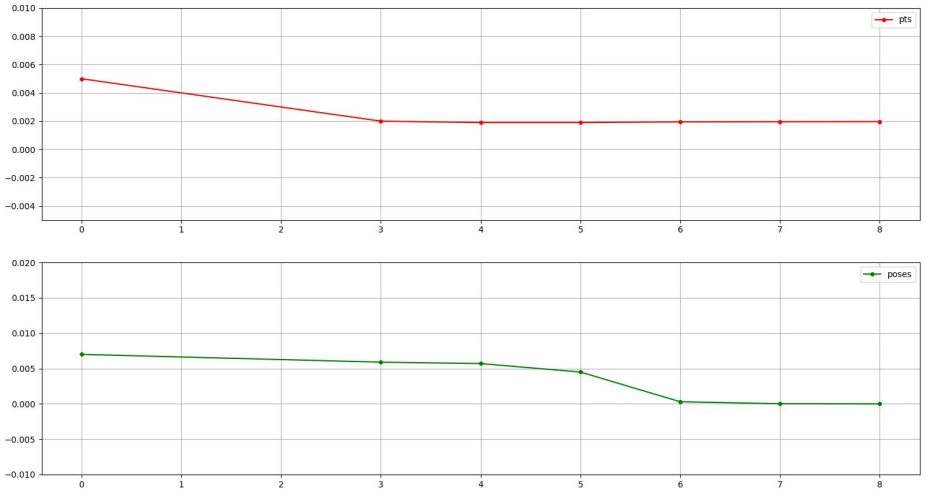
深蓝学院 shenlanxueyuan.com

——测试与结果分析（精度）

依旧仿照论文中的方式，从 0 开始取值，依次测试了 1000、10^4、 10^5、 10^6、 10^7、 10^8。

```
errPt += fabs(1. / points[k].z() - allPoints[k]->Parameters().norm());
```

```
errPose += (vertexCams_vec[i]->Parameters().head(3) - cameras[i].twc).norm();
```



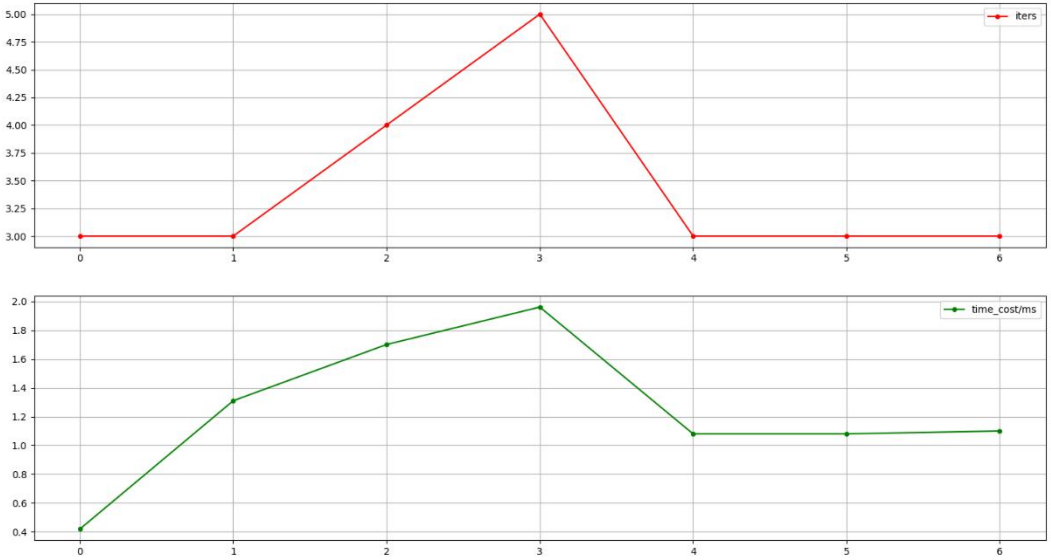
关于poses（绿）和points（红）的精度，在不同w的取值下，纵轴的误差差别非常地小，且误差最后趋于稳定，与论文中的图趋势相同。

提升题--添加先验

深蓝学院 shenlanxueyuan.com

——测试与结果分析（效率）

依旧仿照论文中的方式，从 0 开始取值，依次测试了 1000、10^4、 10^5、 10^6、 10^7、 10^8。



从图中可以看出，随着w值增加，迭代次数（红）与计算时间（绿）会首先达到一个峰值，随后在w增加到某个值以后开始趋于稳定，与论文中的结论相同。

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening