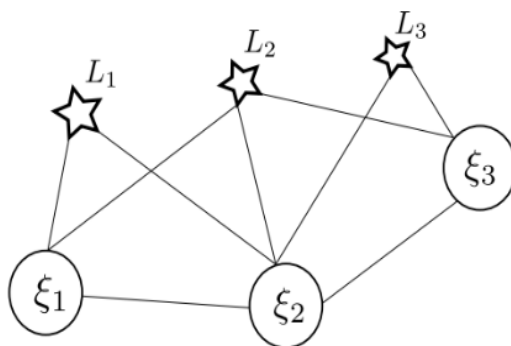


# 从零开始手写VIO 第四课作业

边城量子 2019.7.10

1. 某时刻, SLAM系统中相机和路标点的观测关系如下图所示, 其中  $\xi$  表示相机姿态,  $L$  表示观测到的路标点。当路标点  $L$  表示在世界坐标系下时, 第  $k$  个路标被第  $i$  时刻的相机观测到, 重投影误差为  $r(\xi_i, L_k)$ 。



1. 请绘制上述系统的信息矩阵  $\Lambda$ 。
2. 请绘制相机  $\xi_1$  被 marg 以后的信息矩阵  $\Lambda'$ 。

回答:

## • 目标1: 绘制信息矩阵 $\Lambda$

状态:  $\xi = [\xi_1 \quad \xi_2 \quad \xi_3 \quad L_1 \quad L_2 \quad L_3]^\top$

残差:  $r = [r_{\xi_1 L_1} \quad r_{\xi_1 L_2} \quad r_{\xi_1 \xi_2} \quad r_{\xi_2 L_1} \quad r_{\xi_2 L_2} \quad r_{\xi_2 L_3} \quad r_{\xi_2 \xi_3} \quad r_{\xi_3 L_2} \quad r_{\xi_3 L_3}]^\top$

Jacobian  $J^\top = [J_1^\top \quad J_2^\top \quad J_3^\top \quad J_4^\top \quad J_5^\top \quad J_6^\top \quad J_7^\top \quad J_8^\top \quad J_9^\top]$

其中的  $J_i$  为对应的残差  $r_{mn}$  对状态  $\xi$  求偏导得到, 例如

$$J_1 = \frac{\partial r_{\xi_1 L_1}}{\partial \xi} = \left( \frac{\partial r_{\xi_1 L_1}}{\partial \xi_1} \quad 0 \quad 0 \quad \frac{\partial r_{\xi_1 L_1}}{\partial L_1} \quad 0 \quad 0 \right)$$

以此类推, 计算得到所有的  $J_i \quad 1 \leq i \leq 9$ 。

然后根据公式  $\Lambda_i = J_i^\top \sum_i^{-1} J_i$ , 分别计算得到残差对应的信息矩阵  $\Lambda_{i(6 \times 6)}$ 。

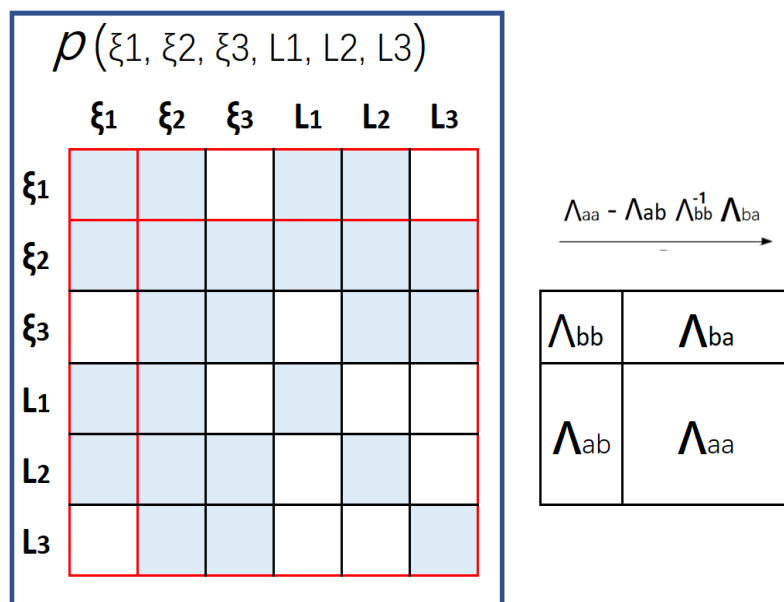
通过分析可知,  $\Lambda_i$  是对称的且跟求偏导的位置有关。

把所有残差的信息矩阵  $\Lambda_i$  加起来, 得到最终的信息矩阵  $\Lambda$ , 可视化如下:

	$\xi_1$	$\xi_2$	$\xi_3$	L1	L2	L3
$\xi_1$						
$\xi_2$						
$\xi_3$						
L1						
L2						
L3						

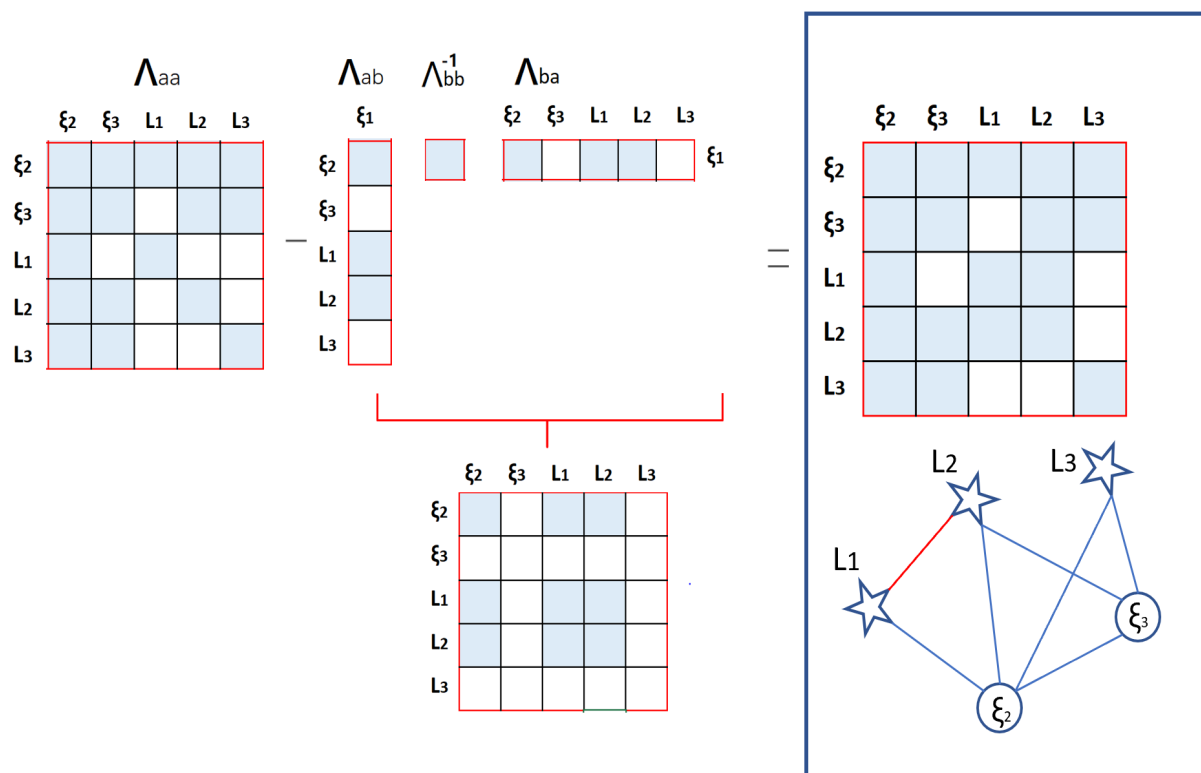
• 目标2: 绘制相机  $\xi_1$  被 marg 以后的信息矩阵  $\Lambda'$

- 第一步，先将信息矩阵  $\Lambda$  分为四个部分，如下图：



备注：其中  $\Lambda_{bb}$  是由矩阵中所有跟  $\xi_1$  有关的行和列（一般是求导）抽取组成的方阵，维数不定，主要是看  $\xi_1$  相连的边有多少条。

- 第二步，根据公式  $\Lambda_{aa} - \Lambda_{ab} \Lambda_{bb}^{-1} \Lambda_{ba}$  来计算最后的  $marg$  之后的信息矩阵  $\Lambda'$ ，整个过程如下图所示。其中四个矩阵块分解后如下图左边和中间，以及三个矩阵块相乘的结果，最右边是  $\Lambda_{aa}$  和根据它重新画的。



备注：从右下角的图可以看出，由于把  $\xi_1$  移除了，导致跟它相连的、原来是条件独立的  $L_1$  和  $L_2$  之间也产生了相关性，信息矩阵也变得稠密。而之前未跟  $\xi_1$  相连的  $L_3$  则未受影响。

2. 请补充作业代码中单目 Bundle Adjustment 信息矩阵的计算，并输出正确的结果。正确的结果为：奇异值最后 7 维接近于 0，表明零空间的维度为 7。

回答：

1. 在文件 `hessian_nullspace_test.cpp` 中增加两行代码如下：

```
// 对应H矩阵的左上角
H.block(i*6,i*6,6,6) += jacobian_Ti.transpose() * jacobian_Ti;

/// 请补充完整作业信息矩阵块的计算
// ---- 新增代码 Start ----
// 对应H矩阵的右下角
H.block(j*3 + 6*poseNums,j*3 + 6*poseNums,3,3) += jacobian_Pj.transpose() *
jacobian_Pj;
// 对应H矩阵的左下角
H.block(i*6,j*3 + 6*poseNums, 6,3) += jacobian_Ti.transpose() * jacobian_Pj;
// ---- 新增代码 End ----

// 对应H矩阵的右上角
H.block(j*3 + 6*poseNums,i*6 , 3,6) += jacobian_Pj.transpose() * jacobian_Ti;
```

2. 保存/编译并执行，得到执行结果：

.....

```
0.00451083
0.0042627
0.00386223
0.00351651
0.00302963
0.00253459
0.00230246
0.00172459
0.000422374
3.21708e-17
2.06732e-17
1.43188e-17
7.66992e-18
6.08423e-18
6.05715e-18
3.94363e-18
hadoop@ubuntu:~/Documents/nullspace_test/build$
```

可以看到最后7个奇异值接近于0。