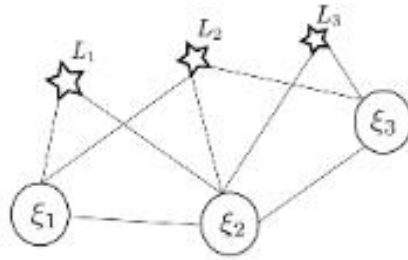


深蓝学院 VIO 课程第四课作业

作业

- 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 ξ 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 Λ 。
 - 2 请绘制相机 ξ_1 被 marg 以后的信息矩阵 Λ' 。
- 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。

1.1 绘制信息矩阵 Λ 和 Λ'

根据题中条件可绘制信息矩阵 Λ ，如图 1 左侧所示。当 ξ_1 被 marg 以后，与 ξ_1 相连的 L_1 和 L_2 便有了关联，故其信息矩阵如图 1 右侧所示。

	ξ_1	ξ_2	ξ_3	L_1	L_2	L_3
ξ_1	■	■		■	■	
ξ_2	■	■	■	■	■	■
ξ_3		■	■		■	■
L_1	■	■		■		
L_2	■	■	■		■	
L_3		■	■			■

	ξ_2	ξ_3	L_1	L_2	L_3
ξ_2	■	■	■	■	■
ξ_3	■	■		■	■
L_1	■		■	■	
L_2	■	■	■	■	
L_3	■	■			■

图 1 信息矩阵 Λ (左) 和 Λ' (右)

2.1 代码补充

NullSpaceTest 项目通过单目 BA 信息矩阵的奇异值来说明零空间的维度为 7，表明单目 V0 的不可观测度为 7 个维度。代码中模拟了相机的一段运动轨迹（水平方向上做圆弧运动，10 个位姿点组成一个 1/4 圆弧；垂直方向上做正弦运动），每个位姿点都可以观测到 20 个路标点，并产生了 20 个观测量。

首先需要构建误差。这里的误差就是重投影误差，公式如下：

$$\xi^* = \arg \min_{\xi} \frac{1}{2} \sum_{i=1}^n \|u_i - K \exp(\xi^{\wedge}) P_i\|_2^2$$

根据《视觉 SLAM 十四讲》的公式 (7.42)、(7.45)、(7.47) 分别得到了误差关于观测的雅克比（源码中的 jacobian_uv_Pc，2*3 维度）、关于位姿点的雅克比（源码中的 jacobian_uv_Pj，2*3 维度）和关于路标点的雅克比（源码中的 jacobian_uv_Ti，2*6 维度）。

再根据信息矩阵的特性和雅克比维度可知信息矩阵 H 的大概组成应如表 1 所示。表 1 将 H 分为左上、左下、右上、右下 4 个区域，每个区域对应的信息子矩阵和维度已列出。

表 1 信息矩阵 H 的构成

	10 个位姿点	20 个观测
10 个位姿点	H_left_up 维度：6*6 数量：10*10 个	H_right_up 维度：6*3 数量：10*20 个
20 个观测	H_left_down 维度：3*6 数量：20*10 个	H_right_down 维度：3*3 数量：20*20 个

其中左上角和左下角的信息子矩阵已经在源码中给出，则根据表 1 很容易可以计算出右上角和右下角的信息矩阵，具体内容见下图源码及注释：

```

56 for (int i = 0; i < poseNums; ++i) {
57     Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
58     Eigen::Vector3d Pc = Rcw * (Pw - camera_pose[i].twc);
59
60     double x = Pc.x();
61     double y = Pc.y();
62     double z = Pc.z();
63     double z_2 = z * z;
64     // 视觉SLAM十四讲公式(7.42), J_de_dPc
65     Eigen::Matrix<double, 2, 3> jacobian_uv_Pc;
66     jacobian_uv_Pc << fx/z, 0, -x * fx/z_2,
67                     0, fy/z, -y * fy/z_2;
68     // 视觉SLAM十四讲公式(7.47), J_de_dPw, 误差对观测点的雅克比
69     Eigen::Matrix<double, 2, 3> jacobian_Pj = jacobian_uv_Pc * Rcw;
70     // 视觉SLAM十四讲公式(7.45), J_de_dxi, 误差对位姿点的雅克比
71     Eigen::Matrix<double, 2, 6> jacobian_Ti;
72     jacobian_Ti << -x * y * fx/z_2, (1 + x*x/z_2)*fx, -y/z*fx, fx/z, 0, -x * fx/z_2,
73                  -(1+y*y/z_2)*fy, x*y/z_2 * fy, x/z * fy, 0, fy/z, -y * fy/z_2;
74
75     // 位姿点与位姿点的信息矩阵, H左上角, 10x10个6x6 block
76     H.block(i*6, i*6, 6, 6) += jacobian_Ti.transpose() * jacobian_Ti;
77
78     /// 请补充完整作业信息矩阵块的计算
79     // 观测点与观测点的信息矩阵, H右下角, 20x20个3x3 block
80     H.block(j*3 + 6*poseNums, j*3 + 6*poseNums, 3, 3) += jacobian_Pj.transpose() * jacobian_Pj;
81
82     // 位姿点与观测点的信息矩阵, H右上角, 10x20个6x3 block
83     H.block(i*6, j*3 + 6*poseNums, 6, 3) += jacobian_Ti.transpose() * jacobian_Pj;
84
85     // 观测点与位姿点的信息矩阵, H左下角, 20x10个3x6 block
86     H.block(j*3 + 6*poseNums, i*6, 3, 6) += jacobian_Pj.transpose() * jacobian_Ti;
87 }
88 }

```

图 2 补充代码部分截图

图 2 中 80 行与 83 行为补充内容。运行结果如图 3 所示，可看出输出结果的最后 7 维值均接近于 0，表明结论正确。

应用程序输出

```

NullSpaceTest ✕
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
21:30:13: /home/vance/vio_ws/src/vio_homework_code/L4/nullspace_test

```

图 3 完整代码的运行结果截图