第四讲-习题

姜帆

2019 年 7 月 8 日

目录

1 推导

1.1 绘制信息矩阵 Λ

某时刻相机位姿 ξ_i 与路标点 L_k 如下图所示，重投影误差为 $r(\xi_i, L_k)$。

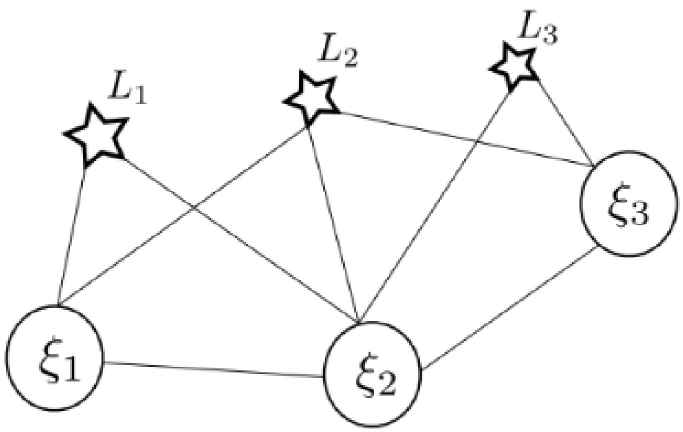


图 1: 相机位姿与路标点

此处残差总共有 9 维；状态变量有 3 个相机 pose，3 个路标点坐标共六维；所以雅克比矩阵为 6×9 维，信息矩阵为 6×6 维。

1.1.1 Λ_1

记 $r_1 = r(\xi_1, \xi_2)$

$$\begin{aligned} J_1 &= \begin{bmatrix} \frac{\partial r_1}{\partial \xi_1} & \frac{\partial r_1}{\partial \xi_2} & 0 & 0 & 0 & 0 \end{bmatrix} \\ \Lambda_1 &= J_1^T \Sigma_1^{-1} J_1 \end{aligned} \tag{1}$$

1.1.2 Λ_2

记 $r_2 = r(\xi_2, \xi_3)$

$$\begin{aligned} J_2 &= \begin{bmatrix} 0 & \frac{\partial r_2}{\partial \xi_2} & \frac{\partial r_2}{\partial \xi_3} & 0 & 0 & 0 \end{bmatrix} \\ \Lambda_2 &= J_2^T \Sigma_2^{-1} J_2 \end{aligned} \tag{2}$$

1.1.3 Λ_3

记 $r_3 = r(\xi_1, L_1)$

$$\begin{aligned} J_3 &= \begin{bmatrix} \frac{\partial r_3}{\partial \xi_1} & 0 & 0 & \frac{\partial r_3}{\partial L_1} & 0 & 0 \end{bmatrix} \\ \Lambda_3 &= J_3^T \Sigma_3^{-1} J_3 \end{aligned} \tag{3}$$

1.1.4 Λ_4

记 $r_4 = r(\xi_1, L_2)$

$$\begin{aligned} J_4 &= \begin{bmatrix} \frac{\partial r_4}{\partial \xi_1} & 0 & 0 & 0 & \frac{\partial r_4}{\partial L_2} & 0 \end{bmatrix} \\ \Lambda_4 &= J_4^T \Sigma_4^{-1} J_4 \end{aligned} \tag{4}$$

1.1.5 Λ_5

记 $r_5 = r(\xi_2, L_1)$

$$\begin{aligned} J_5 &= \begin{bmatrix} 0 & \frac{\partial r_5}{\partial \xi_2} & 0 & \frac{\partial r_5}{\partial L_1} & 0 & 0 \end{bmatrix} \\ \Lambda_5 &= J_5^T \Sigma_5^{-1} J_5 \end{aligned} \tag{5}$$

1.1.6 Λ_6

记 $r_6 = r(\xi_2, L_2)$

$$\begin{aligned} J_6 &= \begin{bmatrix} 0 & \frac{\partial r_6}{\partial \xi_2} & 0 & 0 & \frac{\partial r_6}{\partial L_2} & 0 \end{bmatrix} \\ \Lambda_6 &= J_6^T \Sigma_6^{-1} J_6 \end{aligned} \tag{6}$$

1.1.7 Λ_7

记 $r_7 = r(\xi_2, L_3)$

$$\begin{aligned} J_7 &= \begin{bmatrix} 0 & \frac{\partial r_7}{\partial \xi_2} & 0 & 0 & 0 & \frac{\partial r_7}{\partial L_3} \end{bmatrix} \\ \Lambda_7 &= J_7^T \Sigma_7^{-1} J_7 \end{aligned} \tag{7}$$

1.1.8 Λ_8

记 $r_8 = r(\xi_3, L_2)$

$$\begin{aligned} J_8 &= \begin{bmatrix} 0 & 0 & \frac{\partial r_8}{\partial \xi_2} & 0 & \frac{\partial r_8}{\partial L_2} & 0 \end{bmatrix} \\ \Lambda_8 &= J_8^T \Sigma_8^{-1} J_8 \end{aligned} \tag{8}$$

1.1.9 Λ_9

记 $r_9 = r(\xi_3, L_3)$

$$\begin{aligned} J_9 &= \begin{bmatrix} 0 & \frac{\partial r_9}{\partial \xi_2} & 0 & 0 & 0 & \frac{\partial r_9}{\partial L_3} \end{bmatrix} \\ \Lambda_9 &= J_9^T \Sigma_9^{-1} J_9 \end{aligned} \tag{9}$$

1.1.10 信息矩阵 Λ

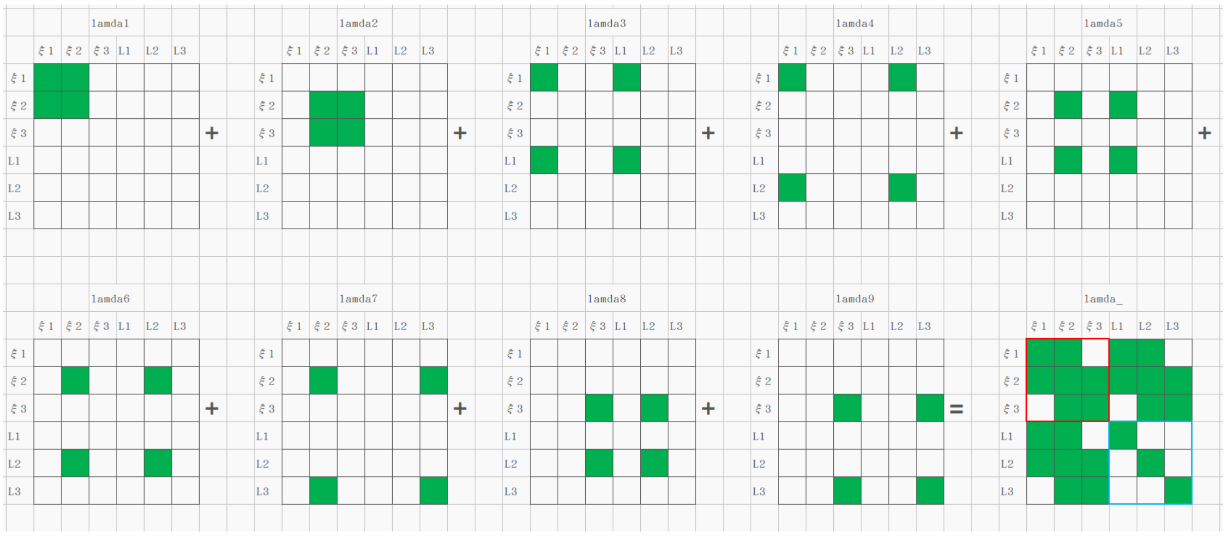


图 2: 信息矩阵

1.2 绘制相机位姿 ξ_1 被 marg 后的信息矩阵 Λ'

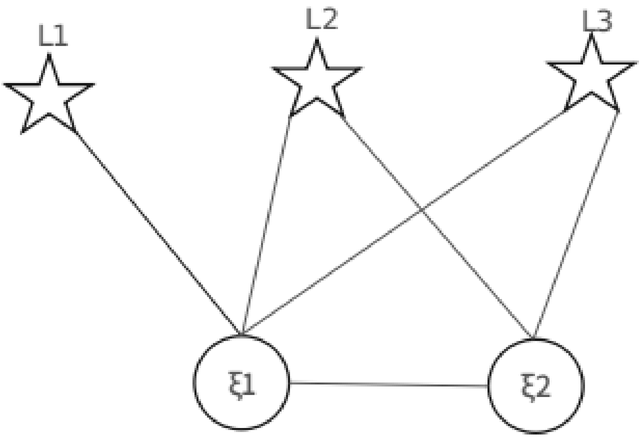


图 3: marg 后相机位姿与路标点

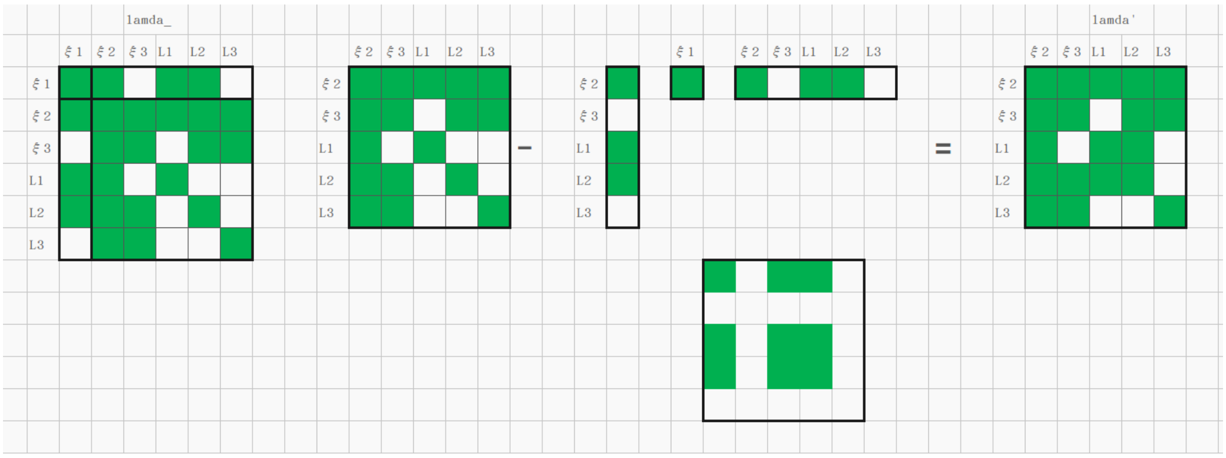


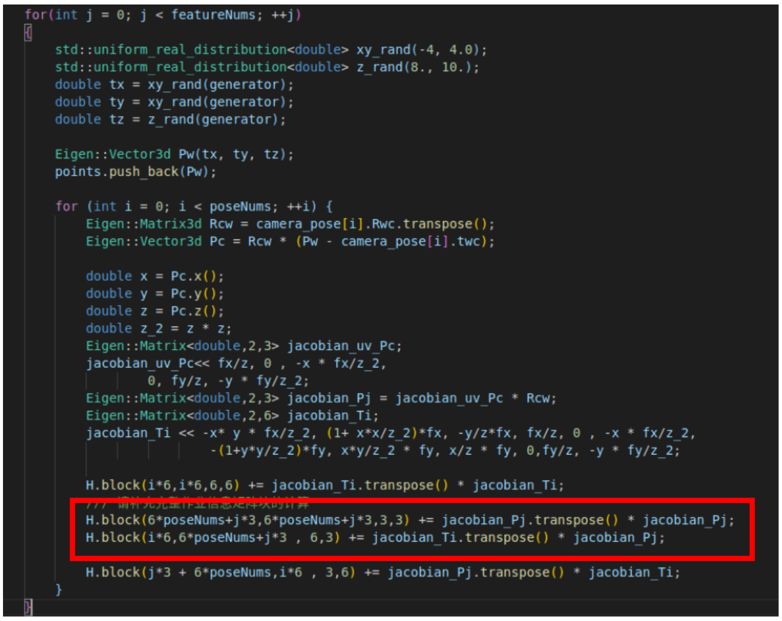
图 4: marg 后的信息矩阵

2 代码补充

补充代码中单目 bundle adjustment 信息矩阵计算部分。

```
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
H.block(6*poseNums+j*3,6*poseNums+j*3,3,3) += jacobian_Pj.transpose() * jacobian_Pj;
H.block(i*6,6*poseNums+j*3 , 6,3) += jacobian_Ti.transpose() * jacobian_Pj;

H.block(j*3 + 6*poseNums,i*6 , 3,6) += jacobian_Pj.transpose() * jacobian_Ti;
```



```
for(int j = 0; j < featureNums; ++j)
{
    std::uniform_real_distribution<double> xy_rand(-4, 4.0);
    std::uniform_real_distribution<double> z_rand(8., 10.);
    double tx = xy_rand(generator);
    double ty = xy_rand(generator);
    double tz = z_rand(generator);

    Eigen::Vector3d Pw(tx, ty, tz);
    points.push_back(Pw);

    for (int i = 0; i < poseNums; ++i) {
        Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
        Eigen::Vector3d Pc = Rcw * (Pw - camera_pose[i].twc);

        double x = Pc.x();
        double y = Pc.y();
        double z = Pc.z();
        double z_2 = z * z;
        Eigen::Matrix<double,2,3> jacobian_uv_Pc;
        jacobian_uv_Pc<< fx/z, 0 , -x * fx/z_2,
                0, fy/z, -y * fy/z_2;
        Eigen::Matrix<double,2,3> jacobian_Pj = jacobian_uv_Pc * Rcw;
        Eigen::Matrix<double,2,6> jacobian_Ti;
        jacobian_Ti << -x* y * fx/z_2, (1+ x*x/z_2)*fx, -y/z*fx, fx/z, 0 , -x * fx/z_2,
                        -(1+y*y/z_2)*fy, x*y/z_2 * fy, x/z * fy, 0,fy/z, -y * fy/z_2;

        H.block(i*6,i*6,6,6) += jacobian_Ti.transpose() * jacobian_Ti;
        H.block(6*poseNums+j*3,6*poseNums+j*3,3,3) += jacobian_Pj.transpose() * jacobian_Pj;
        H.block(i*6,6*poseNums+j*3 , 6,3) += jacobian_Ti.transpose() * jacobian_Pj;

        H.block(j*3 + 6*poseNums,i*6 , 3,6) += jacobian_Pj.transpose() * jacobian_Ti;
    }
}
```

图 5: 信息矩阵计算补充

信息矩阵 SVD 分解后结果:

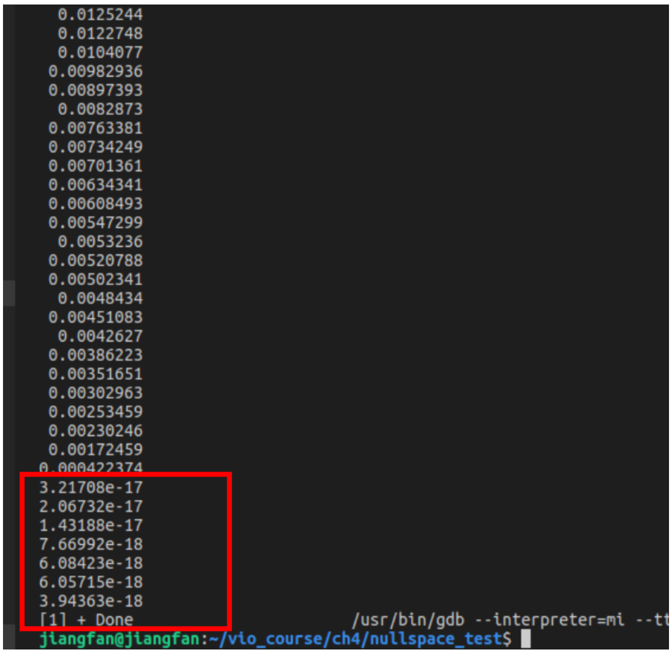


图 6: 信息矩阵 SVD 分解

奇异值分解后从图上可以看出后七维接近于 0，说明单目 slam 的 BA 求解零空间维度为 7。