深蓝学院
shenlanxueyuan.com

从零开始手写VIO_第六期

第6章作业思路分享

视觉前端 Frontend

0. 作业内容

作业

基础题

1. 证明式(15)中，取 $\mathbf{y} = \mathbf{u}_4$ 是该问题的最优解。提示：设 $\mathbf{y}\prime = \mathbf{u}_4 + \mathbf{v}$，其中 $\mathbf{v}$ 正交于 $\mathbf{u}_4$，证明

$$\mathbf{y}\prime^{\top}\mathbf{D}^{\top}\mathbf{D}\mathbf{y}\prime \geq \mathbf{y}^{\top}\mathbf{D}^{\top}\mathbf{D}\mathbf{y}$$

   该方法基于奇异值构造矩阵零空间的理论。
2. 请依据本节课公式，完成特征点三角化代码，并通过仿真测试

提升题

1. 请对测量值加上不同噪声 (增大测量噪声方差)，观察最小奇异值和第二小奇异值之间的比例变化，并绘制比例值的变化曲线。
2. 固定噪声方差参数，将观测图像帧扩成多帧 (如 3，4，5 帧等)，观察最小奇异值和第二小奇异值之间的比例变化，并绘制比例值的变化曲线。

1. 证明式（15）

由三角化方程可构建最小二乘问题，即：

$$\min_{\mathbf{y}} \|\mathbf{D}\mathbf{y}\|_2^2, \quad s.t. \|\mathbf{y}\| = 1$$

则将 y 表示为正交基分解，则有：

$$y = \sum_{i=1}^{4} \eta_i u_i$$

则将 y 代入目标函数，则有（注意 y 模长为 1）：

$$\begin{aligned} y'^T D^T D y' &= \sum_{i=1}^{4} \eta_i u_i^T \sum_{i=1}^{4} \sigma_i^2 u_i u_i^T \sum_{i=1}^{4} \eta_i u_i \\ &= \sum_{i=1}^{4} \eta_i^2 \sigma_i^2 \\ &= \eta_1^2\sigma_1^2 + \eta_2^2\sigma_2^2 + \eta_3^2\sigma_3^2 + \eta_4^2\sigma_4^2 \\ &\xrightarrow{\eta_1^2+\eta_2^2+\eta_3^2+\eta_4^2=1} \eta_1^2\sigma_1^2 + \eta_2^2\sigma_2^2 + \eta_3^2\sigma_3^2 + (1-\eta_1^2-\eta_2^2-\eta_3^2)\sigma_4^2 \\ &= \eta_1^2(\sigma_1^2-\sigma_4^2) + \eta_2^2(\sigma_2^2-\sigma_4^2) + \eta_3^2(\sigma_3^2-\sigma_4^2) + \sigma_4^2 \end{aligned}$$

由于奇异值由大到小排序，即：

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_4$$

故可得 $\eta_1=\eta_2=\eta_3=0, \eta_4=1$ 时可使得目标函数达到最小值，即：取 $y=u_4$ 是该问题的最优解，证毕。

2. 完成特征点三角化代码

深蓝学院 shenlanxueyuan.com

- 将(11)代入(10)的前两行：

$$\begin{aligned} u_k \mathbf{P}_{k,3}^\top \mathbf{y} &= \mathbf{P}_{k,1}^\top \mathbf{y} \\ v_k \mathbf{P}_{k,3}^\top \mathbf{y} &= \mathbf{P}_{k,2}^\top \mathbf{y} \end{aligned} \tag{12}$$

- 每次观测将提供两个这样的方程，视 $\mathbf{y}$ 为未知量，并将 $\mathbf{y}$ 移到等式一侧：

$$\begin{bmatrix} u_1 \mathbf{P}_{1,3}^\top - \mathbf{P}_{1,1}^\top \\ v_1 \mathbf{P}_{1,3}^\top - \mathbf{P}_{1,2}^\top \\ \vdots \\ u_n \mathbf{P}_{n,3}^\top - \mathbf{P}_{n,1}^\top \\ v_n \mathbf{P}_{n,3}^\top - \mathbf{P}_{n,2}^\top \end{bmatrix} \mathbf{y} = \mathbf{0} \rightarrow \mathbf{D}\mathbf{y} = \mathbf{0} \tag{13}$$

- 于是，$\mathbf{y}$ 为 $\mathbf{D}$ 零空间中的一个非零元素。

$$\min_{\mathbf{y}} \|\mathbf{D}\mathbf{y}\|_2^2, \quad s.t. \|\mathbf{y}\| = 1$$

```
/// TODO::homework; 请完成三角化估计深度的代码
// 遍历所有的观测数据，并三角化
Eigen::Vector3d P_est;           // 结果保存到这个变量
P_est.setZero();
/* your code begin */
Eigen::MatrixX4d D = Eigen::MatrixX4d::Zero(2 * (end_frame_id - start_frame_id), 4);
for (int i = start_frame_id; i < end_frame_id; ++i) {
    int num_D_row = i - start_frame_id;
    Eigen::Matrix<double, 3, 4> P_n;
    P_n.block(0, 0, 3, 3) = camera_pose[i].Rwc.transpose();
    P_n.block(0, 3, 3, 1) = -1.0 * camera_pose[i].Rwc.transpose() * camera_pose[i].twc;

    D.block(2*num_D_row, 0, 1, 4) = camera_pose[i].uv(0) * P_n.row(2) - P_n.row(0);
    D.block(2*num_D_row+1, 0, 1, 4) = camera_pose[i].uv(1) * P_n.row(2) - P_n.row(1);
}
Eigen::Matrix<double, 4, 4> DTD = D.transpose() * D;
Eigen::JacobiSVD<Eigen::Matrix4d> svd(DTD, Eigen::ComputeFullU | Eigen::ComputeFullV);
Eigen::MatrixXd eigen_u = svd.matrixU();
Eigen::MatrixXd eigen_v = svd.matrixV();
Eigen::Vector4d eigen_values = svd.singularValues();

std::cout << "Singular values of D^TD = \n" << eigen_values << std::endl;
std::cout << "Sing(4) / Sing(3) = " << eigen_values(3) / eigen_values(2) << std::endl;

Eigen::Vector4d u4 = eigen_u.col(3);
if (u4(2)/u4(3) <= 0 || u4(3) == 0) {
    std::cerr << "Invalid triangulation. " << std::endl;
    return 1;
}
P_est(0) = u4(0) / u4(3);
P_est(1) = u4(1) / u4(3);
P_est(2) = u4(2) / u4(3);
/* your code end */
```

```
[redacted]$ ./estimate_depth
Singular values of D^TD =
    468.406
    7.74642
   0.723255
5.30104e-16
Sing(4) / Sing(3) = 7.32942e-16
ground truth:
  -2.9477 -0.330799   8.43792
your result:
  -2.9477 -0.330799   8.43792
```

3. 变化噪声并绘制曲线

深蓝学院 shenlanxueyuan.com

a. 特征图像坐标 (u, v) 添加噪声

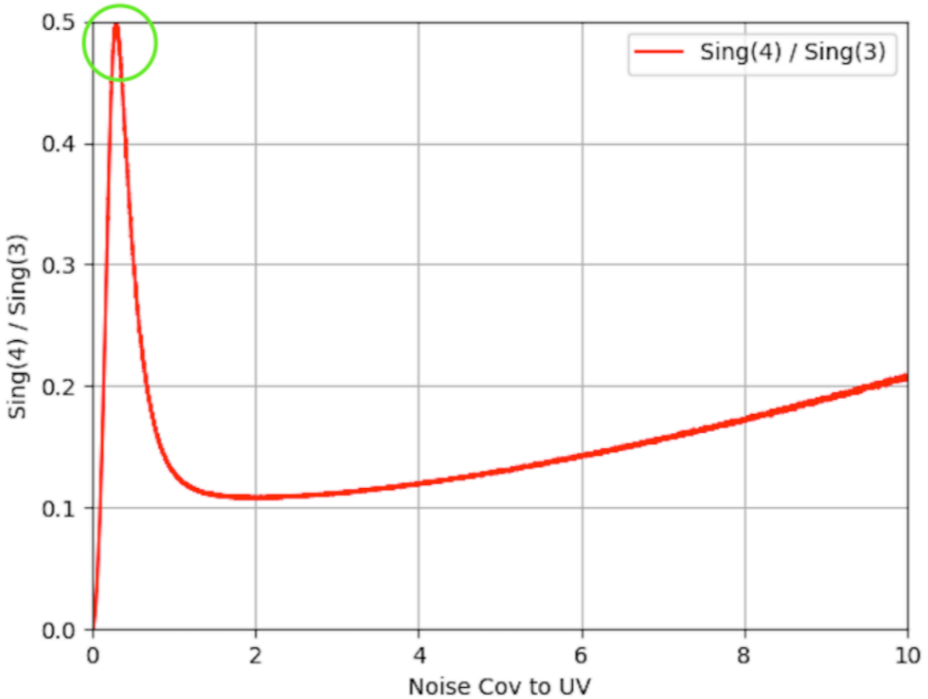
```
void AddNoise2UV(Pose &camera_pose, Eigen::Vector3d &Pw, double cov) {
    unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();
    std::default_random_engine rng(seed);
    std::normal_distribution<double> normalDis(0.0, cov);

    Eigen::Matrix3d Rcw = camera_pose.Rwc.transpose();
    Eigen::Vector3d Pc = Rcw * (Pw - camera_pose.twc);

    double x = Pc.x();
    double y = Pc.y();
    double z = Pc.z();

    camera_pose.uv = Eigen::Vector2d(x/z + normalDis(rng), y/z + normalDis(rng));
}
```

```
Cov = 0, Sing(4)/Sing(3) = 7.32942e-16
Cov = 0.001, Sing(4)/Sing(3) = 1.10254e-05
Cov = 0.002, Sing(4)/Sing(3) = 4.37024e-05
Cov = 0.003, Sing(4)/Sing(3) = 9.90077e-05
Cov = 0.004, Sing(4)/Sing(3) = 0.000174783
Cov = 0.005, Sing(4)/Sing(3) = 0.000275842
Cov = 0.006, Sing(4)/Sing(3) = 0.000395941
Cov = 0.007, Sing(4)/Sing(3) = 0.000538281
Cov = 0.008, Sing(4)/Sing(3) = 0.000702382
Cov = 0.009, Sing(4)/Sing(3) = 0.000887033
```



3. 变化噪声并绘制曲线

b. 特征 3 维空间坐标 (x, y, z) 添加噪声

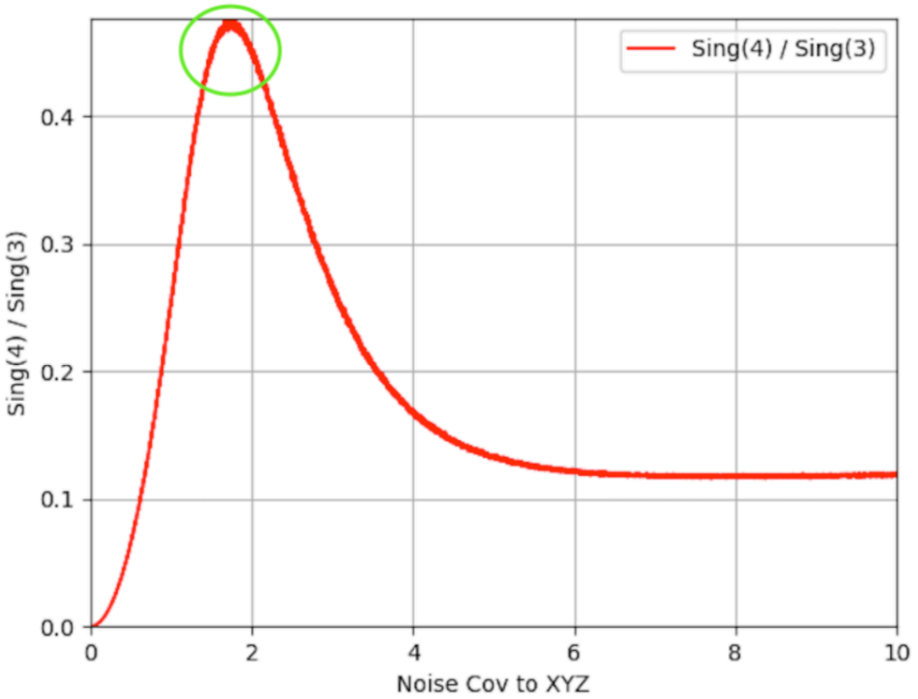
```
void AddNoise2XYZ(Pose &camera_pose, Eigen::Vector3d &Pw, double cov) {
    unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();
    std::default_random_engine rng(seed);
    std::normal_distribution<double> normalDis(0.0, cov);

    Eigen::Matrix3d Rcw = camera_pose.Rwc.transpose();
    Eigen::Vector3d Pc = Rcw * (Pw - camera_pose.twc);

    double x = Pc.x() + normalDis(rng);
    double y = Pc.y() + normalDis(rng);
    double z = Pc.z() + normalDis(rng);

    camera_pose.uv = Eigen::Vector2d(x/z, y/z);
}
```

```
Cov = 0, Sing(4)/Sing(3) = 7.32942e-16
Cov = 0.001, Sing(4)/Sing(3) = 2.6675e-07
Cov = 0.002, Sing(4)/Sing(3) = 1.05894e-06
Cov = 0.003, Sing(4)/Sing(3) = 2.38749e-06
Cov = 0.004, Sing(4)/Sing(3) = 4.24082e-06
Cov = 0.005, Sing(4)/Sing(3) = 6.59319e-06
Cov = 0.006, Sing(4)/Sing(3) = 9.51725e-06
Cov = 0.007, Sing(4)/Sing(3) = 1.29261e-05
Cov = 0.008, Sing(4)/Sing(3) = 1.68123e-05
Cov = 0.009, Sing(4)/Sing(3) = 2.13253e-05
```

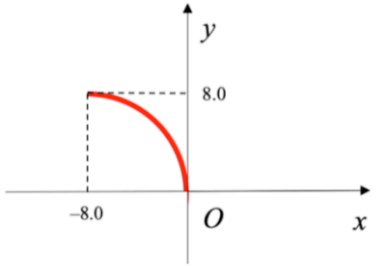


4. 变化帧数并绘制曲线

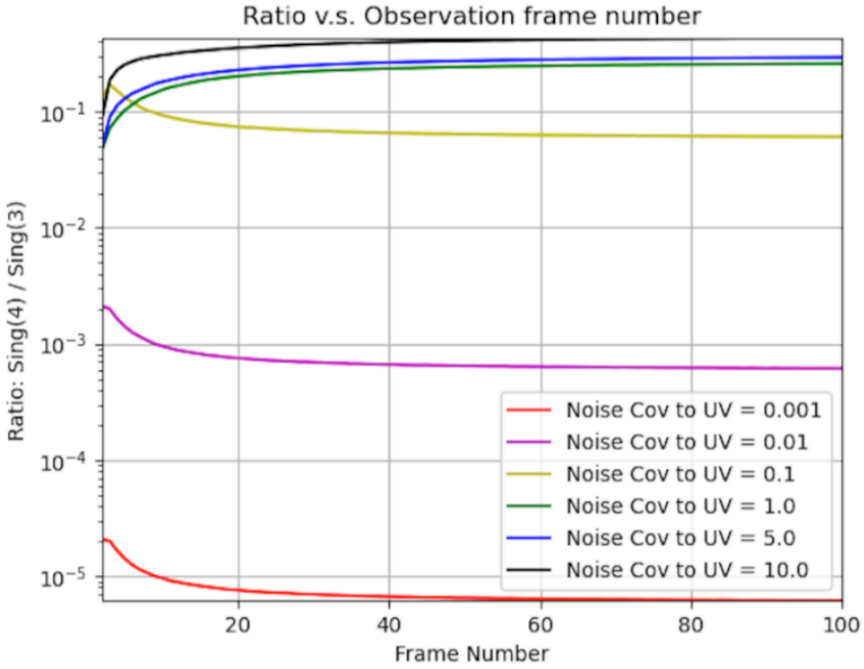
深蓝学院 shenlanxueyuan.com

a. 特征图像坐标 (u, v) 添加噪声

对于本题添加多帧的方法，此处选择**固定轨迹路线（即固定轨迹起终点），并在轨迹中加密观测帧数量**的方法。下图红色 1/4 圆弧为本题的轨迹在 x-y 平面的投影：



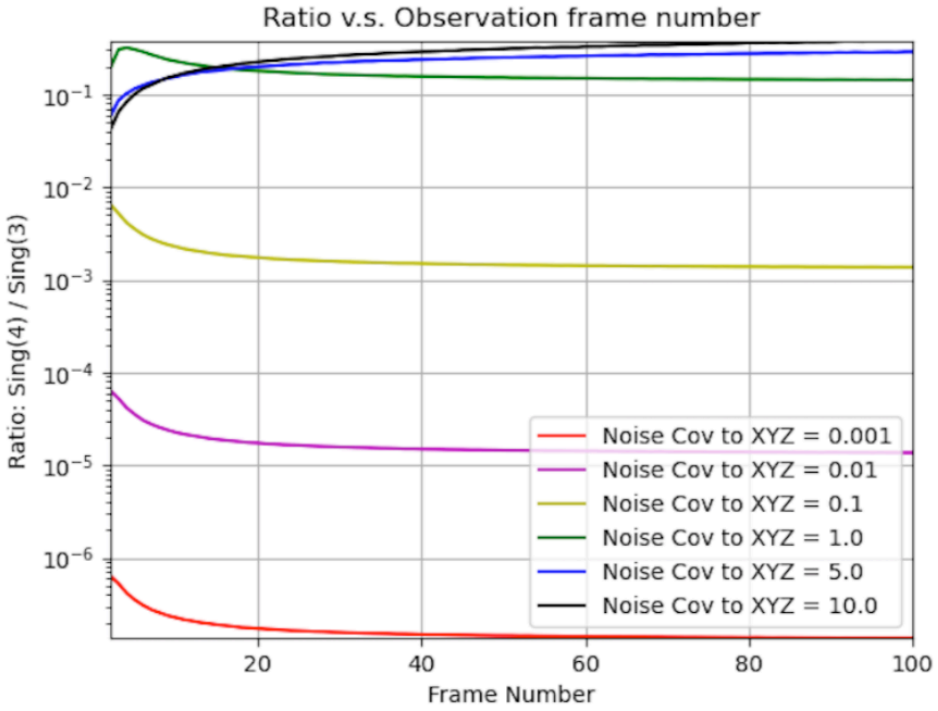

```
Pose Num = 2, Sing(4)/Sing(3) = 2.07192e-05
Pose Num = 3, Sing(4)/Sing(3) = 1.99968e-05
Pose Num = 4, Sing(4)/Sing(3) = 1.64773e-05
Pose Num = 5, Sing(4)/Sing(3) = 1.40879e-05
Pose Num = 6, Sing(4)/Sing(3) = 1.24474e-05
Pose Num = 7, Sing(4)/Sing(3) = 1.13593e-05
Pose Num = 8, Sing(4)/Sing(3) = 1.06313e-05
Pose Num = 9, Sing(4)/Sing(3) = 9.95823e-06
Pose Num = 10, Sing(4)/Sing(3) = 9.58587e-06
Pose Num = 11, Sing(4)/Sing(3) = 9.10303e-06
```



4. 变化帧数并绘制曲线

b. 特征 3 维空间坐标 (x, y, z) 添加噪声

```
Pose Num = 2, Sing(4)/Sing(3) = 6.36889e-07
Pose Num = 3, Sing(4)/Sing(3) = 5.26347e-07
Pose Num = 4, Sing(4)/Sing(3) = 4.15505e-07
Pose Num = 5, Sing(4)/Sing(3) = 3.50915e-07
Pose Num = 6, Sing(4)/Sing(3) = 3.07572e-07
Pose Num = 7, Sing(4)/Sing(3) = 2.76284e-07
Pose Num = 8, Sing(4)/Sing(3) = 2.56062e-07
Pose Num = 9, Sing(4)/Sing(3) = 2.38332e-07
Pose Num = 10, Sing(4)/Sing(3) = 2.26721e-07
Pose Num = 11, Sing(4)/Sing(3) = 2.15319e-07
```



深蓝学院
shenlanxueyuan.com

感谢各位聆听

Thanks for Listening