深蓝学院
shenlanxueyuan.com

第十五章 前端Frontend作业评讲

主讲人 于子平

作业内容

深蓝学院
shenlanxueyuan.com

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。

基础题一

- 此题目具有较强发散性，答案不唯一。

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

寻找最小二乘解：

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

基础题一

对D做SVD分解，$D = V\Sigma U^T$ ，UV都是正交阵，Σ 是奇异矩阵。

$$\begin{aligned} \|Dy\|_2^2 &= (Dy)^T(Dy) \\ &= y^T D^T D y \\ &= y^T U \Sigma^T V^T V \Sigma U^T y \\ &= y^x U \Sigma^T \Sigma U^\tau y \end{aligned} \quad \ldots (1)$$

其中，$\Sigma^T\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix} \ldots (2)$ ，σ_i 是奇异值，且有 $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_4$。

设y由正交基U线性组合而得，则有：

$$y = \alpha_1 u_1 + \alpha_2 u_2 + \alpha_3 u_3 + \alpha_4 u_4 = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \ldots (3)$$

基础题一

将y和$\Sigma^{\tau}\Sigma$ 代入$\|Dy\|_2^2$，可得:

$$\begin{aligned}
\|Dv\|_2^2 &= y^T U \Sigma^T \Sigma U^T y \\
&= \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & a_4 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_1^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \ldots (4) \\
&= \alpha_1^2\sigma_1^2 + \alpha_2^2\sigma_2^2 + \alpha_3^2\sigma_3^2 + \alpha_4^2\sigma_4^2 \\
&\geq \sigma_4^2\left(\alpha_1^2 + \alpha_2^2 + \alpha_1^2 + a_4^2\right)
\end{aligned}$$

因为||y||=1，所以有$y^{\tau} y = a_1^2 + a_2^2 + a_1^2 + a_4^2 = 1$

则有$\|Dy\|_2^2 = y^{\tau} U \Sigma^T \Sigma U^{\tau} y \geq \sigma_4^2$

基础题一

当且仅当 $y = u_4$ 时（根据(4)式$a_4^2 = 1$，$a_1^2 = 0$，$a_2^2 = 0$，$a_3^2 = 0$ 时，才会取到σ_4^2，然后反代回(3)就可以得到y了），即y取最小奇异值对应的奇异向量时，目标函数值取上述最小值。

拉格朗日乘子法

对 $\mathbf{D}^\top\mathbf{D}$ 进行 SVD

基础题二

```
    /// 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 */
    auto loop_times = end_frame_id - start_frame_id;
    Eigen::MatrixXd D(2*loop_times, 4);
    //填充D矩阵
    for(int i=0;i<loop_times;++i){
        //构建T(R,t)矩阵，即pdf中的P矩阵
        Eigen::MatrixXd P_k(3, 4);
        P_k.block<3, 3>(0, 0) = camera_pose[i+start_frame_id].Rwc.transpose();
        P_k.block<3, 1>(0, 3) = -camera_pose[i+start_frame_id].Rwc.transpose() *
camera_pose[i+start_frame_id].twc;
        //表示P矩阵中的三行，来消除未知深度
        auto P_k1 = P_k.block<1, 4>(0, 0);
        auto P_k2 = P_k.block<1, 4>(1, 0);
        auto P_k3 = P_k.block<1, 4>(2, 0);

        //得到消除深度信息后的两个方程，即构建D矩阵
        D.block<1, 4>(2*i, 0) = camera_pose[i+start_frame_id].uv[0] * P_k3 -
P_k1;
        D.block<1, 4>(2*i+1, 0) = camera_pose[i+start_frame_id].uv[1] * P_k3 -
P_k2;
    }
```

```
    //根据证明题的结论，y = u4
    Eigen::Matrix4d D_res = D.transpose() * D;
    Eigen::JacobiSVD<Eigen::Matrix4d> svd(D_res, Eigen::ComputeFullU |
Eigen::ComputeFullV);
    auto res_U = svd.matrixU();
    std::cout << "U = " << res_U << std::endl;
    //奇异值
    Eigen::Vector4d Singular_values = svd.singularValues();
    std::cout << "奇异值：\n" << Singular_values << std::endl;

    auto tmp = res_U.rightCols(1);
    P_est.x() = tmp(0) / tmp(3);
    P_est.y() = tmp(1) / tmp(3);
    P_est.z() = tmp(2) / tmp(3);
    /* your code end */
```

参照课件的三角化部分公式即可

基础题二

```
krasjet@krasjet-Lenovo-Legion-Y7000P-2020H:~/Documents/course6_hw/build$ ./estim
ate_depth
U =  0.0530721    0.846878     0.41558   -0.327528
 -0.103079    0.431629   -0.895388 -0.0367562
 -0.102585    0.309021    0.122288    0.937565
  0.987945   0.0316285  -0.103049    0.111113
奇异值：
    468.406
    7.74642
   0.723255
5.30104e-16
ground truth:
  -2.9477 -0.330799   8.43792
your result:
  -2.9477 -0.330799   8.43792
```

提升题一

噪声处理方式不同、噪声选取范围不同等差异，因此结果不唯一，逻辑自洽，言之有理即可。

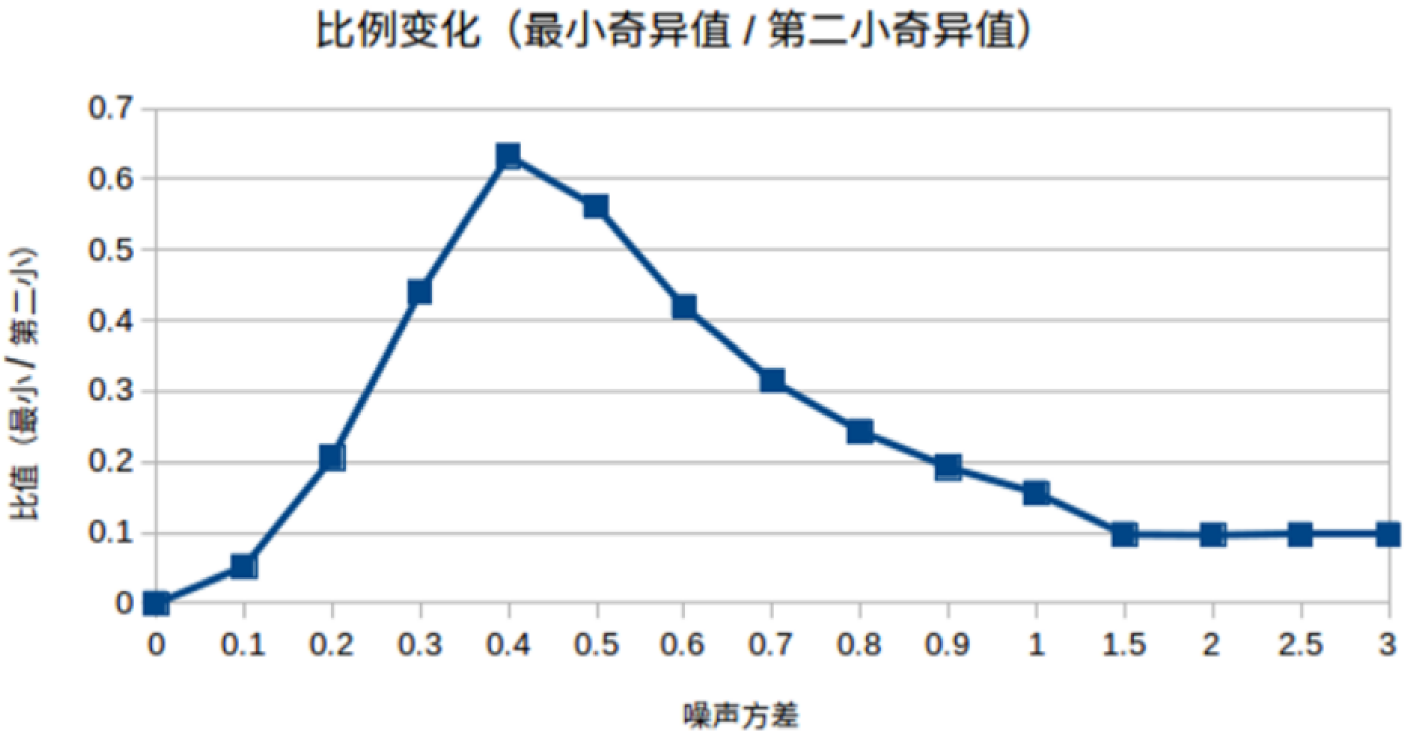
```
//生成正太高斯分布噪声
double mu = 0.0, sigma = 0.;
std::normal_distribution<double> norm(mu, sigma);
//从第三帧开始，计算这一个特征点在每一帧图像里的归一化坐标
for (int i = start_frame_id; i < end_frame_id; ++i) {
    Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
    Eigen::Vector3d Pc = Rcw * (Pw - camera_pose[i].twc);

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

    double u = x / z + noise(generator);
    double v = y / z + noise(generator);

    camera_pose[i].uv = Eigen::Vector2d(u, v);
}
```

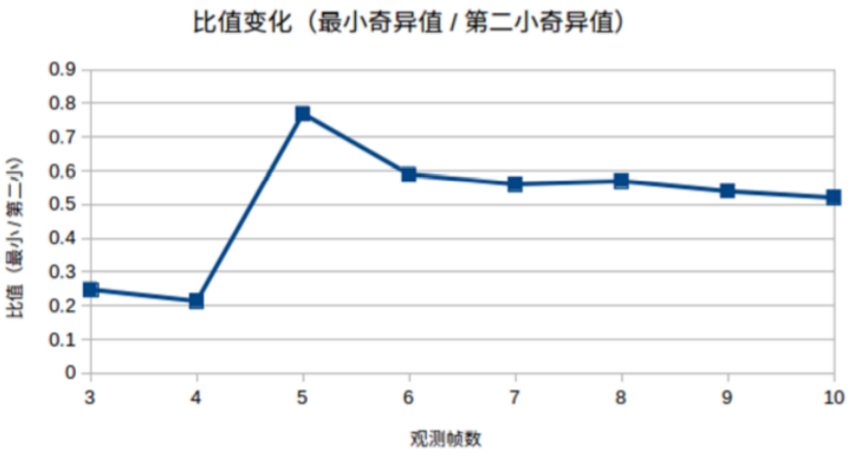
提升题一


比例变化（最小奇异值 / 第二小奇异值）
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
比值（最小 / 第二小）
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.5
2
2.5
3
噪声方差

提升题二

在固定噪声大小后（σ = 0.5），通过测试不同观测帧数对最小奇异值和第二小奇异值比例的影响，统计如下表，并绘制了对应的比例变化曲线

噪声方差	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
观测帧数	3	4	5	6	7	8	9	10
最小奇异值	0.0270532	0.04546	0.192775	0.358927	0.63029	0.837918	1.10706	1.70157
第二小奇异值	0.109126	0.21275	0.2504	0.608317	1.12412	1.46937	2.05068	3.26857
比例(最小 / 第二小)	0.24790792	0.213679	0.769868	0.5900328	0.560696	0.570257	0.53985	0.520585



感谢各位聆听！

Thanks for Listening