# VIO 第六章作业

曾卓

06 2021

# 1 基础部分

## 1.1 理论部分

**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

对 D 进行 SVD 得到 $D = \sum_{i=1}^{4} \sigma_i v_i u_i^\top$，得到 $D^\top D = \sum_{i=1}^{4} \sigma_i^2 u_i u_i^\top$ 另取一向量 v 与 $u_4$ 正交，设 $y = u_4, y' = u_4 + v$。

$$\begin{aligned}
y'^\top D^\top D y' &= (u_4 + v)^\top D^\top D (u_4 + v) \\
&= (u_4 + v)^\top \sum_{i=1}^{4} \sigma_i^2 u_i u_i^\top (u_4 + v) \\
&= y^\top D^\top D y + v^\top \sum_{i=1}^{4} \sigma_i^2 u_i u_i^\top v + u_4^\top \sum_{i=1}^{4} \sigma_i^2 u_i u_i^\top v + v^\top \sum_{i=1}^{4} \sigma_i^2 u_i u_i^\top u_4 \\
&= y^\top D^\top D y + v^\top \sum_{i=1}^{3} \sigma_i^2 u_i u_i^\top v + 0 + 0 \\
&= y^\top D^\top D y + \sum_{i=1}^{3} \sigma_i^2 (u_i^\top v)^2 \\
&\geq y^\top D^\top D y
\end{aligned} \tag{1}$$

当且仅当 $\sigma_i = 0$ 的时候取等号，否则由于 $\sigma^2 > 0$ 取大于号。
所以当 $y = u_4$ 为最优解，加上任意正交分量都会使得希望最小化的量增加。

## 1.2 代码部分

由于 $D^\top D$ 是对称矩阵，故 SVD 和 EVD 的结果相同，在代码中分别用 selfadjointEingensolver 和 JacobiSVD 进行了求解，除了一个无限接近于 0 的数以外，其它结果均相同 (排序是反的)，比较了一下求解时间也差不多（一般 JacobiSVD 快一点）。

```cpp
    // step 1 construct matrix D
    Eigen::MatrixX4d D;
    for (int i = start_frame_id; i < end_frame_id; ++i)
    {
        double u = camera_pose[i].uv[0];
        double v = camera_pose[i].uv[1];
        Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
        Eigen::Vector3d tcw = -Rcw * camera_pose[i].twc;

        D.conservativeResize(D.rows() + 2, Eigen::NoChange);
        Eigen::Matrix<double, 3, 4> P;
        P << Rcw, tcw;
        D.row(D.rows() - 2) = u * P.row(2) - P.row(0);
        D.row(D.rows() - 1) = v * P.row(2) - P.row(1);
    }
    // step 2 compute D^T * D and apply SVD
    Eigen::MatrixXd DTD_temp = D.transpose() * D;
    Eigen::MatrixXd DTD = 0.5 * (DTD_temp.transpose() + DTD_temp);
    // DTD is symmetric so we can use SelfAdjointEVD
    auto start = std::chrono::steady_clock::now();
    Eigen::SelfAdjointEigenSolver<Eigen::MatrixXd> saes(DTD);
    auto end = std::chrono::steady_clock::now();
    std::chrono::duration<double> elapsed_seconds = end - start;
    std::cout << "EVD used time = " << 1e3 * elapsed_seconds.count() << " ms" << std::endl;
    auto eigenVal = saes.eigenvalues();
    std::cout << ((saes.info() == 0) ? "success" : "fail") << std::endl;
    std::cout << eigenVal << std::endl;
    // std::cout << saes.eigenvectors().col(0) << std::endl;
    Eigen::Vector4d P_est4 = saes.eigenvectors().col(0);
    // std::cout << P_est4 << std::endl;
    P_est = (P_est4 / P_est4(3, 0)).head(3);

    // same result for SVD
    auto start1 = std::chrono::steady_clock::now();
    Eigen::JacobiSVD<Eigen::MatrixXd> svd(DTD, Eigen::ComputeThinU | Eigen::ComputeThinV);
    auto end1 = std::chrono::steady_clock::now();
    std::chrono::duration<double> elapsed_seconds1 = end1 - start1;
    std::cout << "SVD used time = " << 1e3 * elapsed_seconds1.count() << " ms" << std::endl;
    std::cout << svd.singularValues() << std::endl;
    // std::cout << svd.matrixU() << '\n'
    //   << svd.matrixV() << std::endl;
```

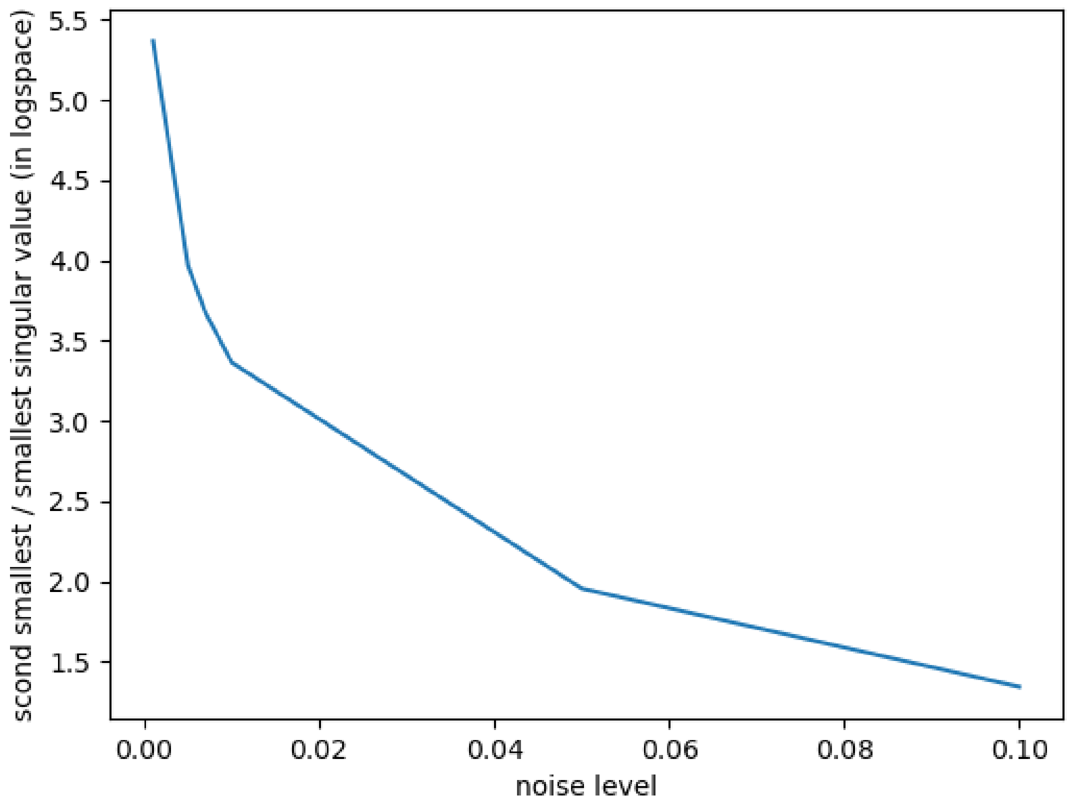
```
EVD used time = 0.07733 ms
success
-1.08765e-15
    0.723255
     7.74642
     468.406
SVD used time = 0.051253 ms

    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
```

```
if (std::abs(eigenVal[1] / eigenVal[0]) < 1e2)
{
    std::cout << "-------------solution not reliable------------\n"
              << std::endl;
}
else
    std::cout << "-------------good solution------------\n"
              << std::endl;
```

# 2 提高部分

## 2.1 加上了不同噪声

```
double w_sigma = sigma; // variance of Gaussian noise
std::normal_distribution<double> noise(0., w_sigma);
camera_pose[i].uv = Eigen::Vector2d(x / z + noise(generator), y / z + noise(generator));
```




——std_variance of Gauss noise is 0 ———-

EVD used time = 0.170782 ms success

-1.08765e-15 0.723255 7.74642 468.406

SVD used time = 0.100784 ms

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

scecond smallest sigular value compared with smallest one :-6.64968e+14

————–-good solution————–-

——std_variance of Gauss noise is 0.001 ———-

EVD used time = 0.079452 ms success

3.10923e-06 0.723003 7.74711 468.379

SVD used time = 0.080734 ms

468.379 7.74711 0.723003 3.10923e-06

ground truth: -2.9477 -0.330799 8.43792

your result: -2.94906 -0.33886 8.44899

scecond smallest sigular value compared with smallest one :232534

—————-good solution—————-

——std_variance of Gauss noise is 0.005 ———-

EVD used time = 0.071436 ms success

7.78384e-05 0.722011 7.74988 468.268

SVD used time = 0.047502 ms

468.268 7.74988 0.722011 7.78384e-05

ground truth: -2.9477 -0.330799 8.43792

your result: -2.95459 -0.371189 8.49342

scecond smallest sigular value compared with smallest one :9275.77

—————-good solution—————-

——std_variance of Gauss noise is 0.007 ———-

EVD used time = 0.070444 ms success

0.000152667 0.721524 7.75127 468.212

SVD used time = 0.049241 ms

468.212 7.75127 0.721524 0.000152667

ground truth: -2.9477 -0.330799 8.43792

your result: -2.95741 -0.387408 8.51572

scecond smallest sigular value compared with smallest one :4726.13

—————-good solution—————-

——std_variance of Gauss noise is 0.01 ———-

EVD used time = 0.067193 ms success

0.000311878 0.720803 7.75337 468.13

SVD used time = 0.04815 ms

468.13 7.75337 0.720803 0.000311878

ground truth: -2.9477 -0.330799 8.43792

your result: -2.96168 -0.411807 8.54928

scecond smallest sigular value compared with smallest one :2311.17

—————-good solution—————-

——std_variance of Gauss noise is 0.05 ———-

EVD used time = 0.15171 ms success

0.00788922 0.712333 7.78234 467.03

SVD used time = 0.070236 ms

467.03 7.78234 0.712333 0.00788922

ground truth: -2.9477 -0.330799 8.43792

your result: -3.02521 -0.747822 9.01257

scecond smallest sigular value compared with smallest one :90.292

————-solution not reliable————-

——std_variance of Gauss noise is 0.1 ———-

EVD used time = 0.16632 ms success

0.031899 0.704401 7.8214 465.674

SVD used time = 0.113002 ms

465.674 7.8214 0.704401 0.031899

ground truth: -2.9477 -0.330799 8.43792

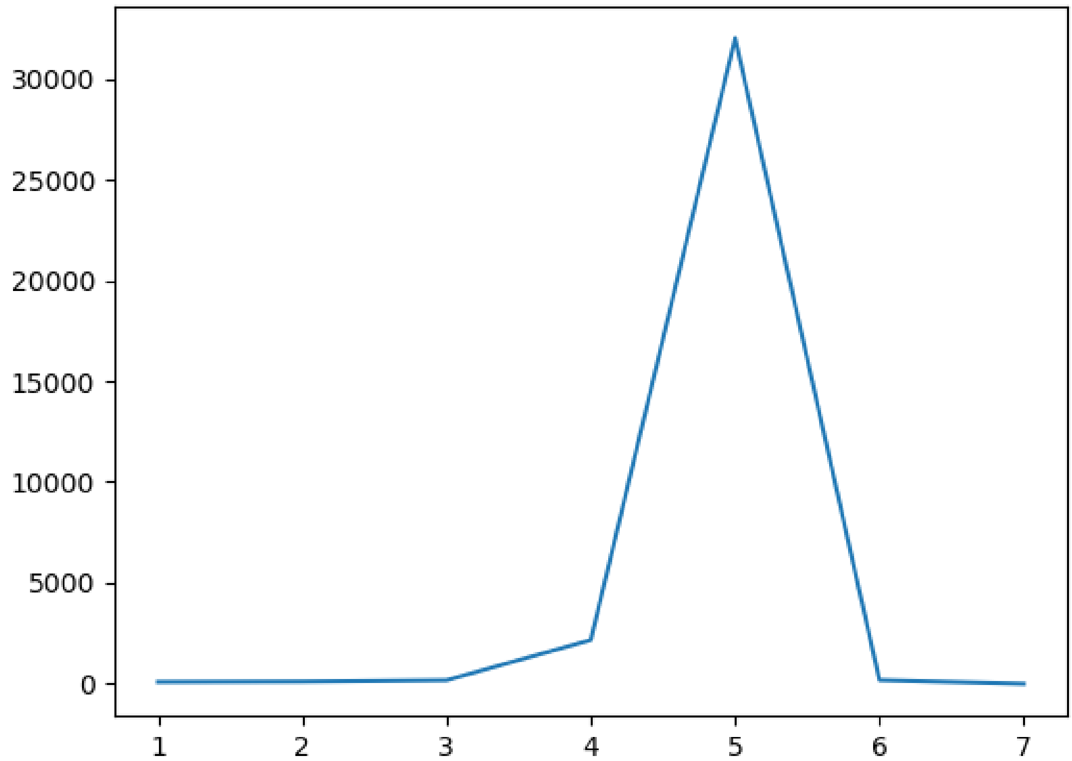
your result: -3.12246 -1.2126 9.65179

scecond smallest sigular value compared with smallest one :22.0822

————-solution not reliable————-

## 2.2 观测图像扩展成多帧

我不是很理解这个扩展成多帧的含义，我理解的是，增加关键帧之间的间隔，即不将每一帧都视为关键帧（或者说，不对每一帧都进行三角化），这样虽然会使得最后获得的总数据量变少，但是在噪声方差比较大的时候，这样反而可以使得 D 更加不容易满秩，也就是第二小奇异值与最小奇异值之比增大了，从下面的例子里就可以看出来，在间隔为 5 帧的时候，反而是这个比值最大的时候。
但是从结果来看，即使这个比值变大了，也不代表这个结果就更加精准了，反而是间隔为 2 或者 3 的时候更接近真实值，所以依据奇异值之比来判断结果好坏并不完全可靠！




——interval is 1 ———-

EVD used time = 0.076526 ms success

0.00788922 0.712333 7.78234 467.03

SVD used time = 0.047147 ms

467.03 7.78234 0.712333 0.00788922

ground truth: -2.9477 -0.330799 8.43792

your result: -3.02521 -0.747822 9.01257

scecond smallest sigular value compared with smallest one :90.292

———–-solution not reliable———–-

——interval is 2 ———-

EVD used time = 0.060129 ms success

0.00464101 0.504309 4.44015 267.568 SVD used time = 0.044849 ms

267.568 4.44015 0.504309 0.00464101

ground truth: -2.9477 -0.330799 8.43792

your result: -3.01961 -0.520352 8.71717

scecond smallest sigular value compared with smallest one :108.664

———–-good solution———–-

——interval is 3 ———-

EVD used time = 0.078905 ms success

0.00258029 0.448076 3.36703 201.184

SVD used time = 0.048146 ms

201.184 3.36703 0.448076 0.00258029

ground truth: -2.9477 -0.330799 8.43792

your result: -3.16883 -0.524578 8.61687

scecond smallest sigular value compared with smallest one :173.653

———–-good solution———–-

——interval is 4 ———-

EVD used time = 0.118972 ms success

0.000139171 0.30154 2.20296 102.215

SVD used time = 0.080991 ms

102.215 2.20296 0.30154 0.000139171

ground truth: -2.9477 -0.330799 8.43792

your result: -2.89518 -0.0605566 8.38062

scecond smallest sigular value compared with smallest one :2166.68

———–-good solution———–-

——interval is 5 ———-

EVD used time = 0.067309 ms success

1.1462e-05 0.367118 2.21435 118.44

SVD used time = 0.047641 ms

118.44 2.21435 0.367118 1.1462e-05

ground truth: -2.9477 -0.330799 8.43792

your result: -3.13642 -0.463303 9.12417

scecond smallest sigular value compared with smallest one :32029.1

———–-good solution———–-

——interval is 6 ———-

EVD used time = 0.169675 ms success

0.00242278 0.431692 2.23918 134.447

SVD used time = 0.047002 ms

134.447 2.23918 0.431692 0.00242278

ground truth: -2.9477 -0.330799 8.43792

your result: -3.20287 -0.504016 8.63133

scecond smallest sigular value compared with smallest one :178.18

————-good solution————-

——interval is 7 ———-

EVD used time = 0.061622 ms success

-2.14709e-15 -4.62983e-16 1.12061 17.8799

SVD used time = 0.039683 ms

17.8799 1.12061 5.4326e-17 3.68863e-17

ground truth: -2.9477 -0.330799 8.43792

your result: -1.86853 1.8334 4.44486

scecond smallest sigular value compared with smallest one :0.215633

————-solution not reliable————-