Jindong_Shi
2022-01-23

第 15 节课习题

1. 证明式 (15) 中，取 $y = u_4$ 是该问题的最优解。提示: 设 $y' = u_4 + v$，其中 v 正交于u_4，证明

$$y'^T D^T D y' \geq y^T D^T D y$$

该方法基于奇异值构造矩阵零空间的理论。

证明:

首先对对称矩阵 D^TD 做 SVD 分解:

$D^TD=\sum_{i=1}^{4}\sigma_i^2 u_i u_j^T$，其中奇异值 σ_i 由大到小排列，u_i, u_j 为标准正交基，即 $\|u_i\|_2^2=1$，$u_i^Tu_j=0$，当 $i\neq j$

根据奇异值构造零空间理论，矩阵 D 的零空间可以由 D 的奇异值向量线性组合得到，

而 $Dy=0$，y 位于 D 的零空间内，故

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

假设 u_k 为其中某一个奇异值向量，那么

$$\begin{aligned} y &= a_k u_k + \underbrace{\sum_{i=1,i\neq k}^{4} a_i u_i}_{\triangleq v} \\ &= a_k u_k + v \end{aligned}$$

由于奇异值向量彼此正交，故 $a_k u_k$ 与 v 也正交，将 $y=a_k u_k+v$ 代入 $\min_y \|Dy\|_2^2$, s.t $\|y\|=1$，得

$$\begin{aligned} \min_y \|Dy\|_2^2 &= \min_y \left(y^T D^T D y\right) \\ &= \min_y \left((a_k u_k+v)^T D^T D (a_k u_k+v)\right) \\ &= \min_y \left(a_k u_k^T D^T D a_k u_k + \underbrace{a_k u_k^T D^T D v}_{\overset{!}{=}0} + \underbrace{v^T D^T D a_k u_k}_{\overset{!}{=}0} + v^T D^T D v \right) \\ &= \min_y \left(a_k^2 u_k^T D^T D u_k + v^T D^T D v \right) \end{aligned}$$

将 $D^TD=\sum_{i=1}^{4}\sigma_i^2 u_i u_j^T$ 代入，

$$\begin{aligned} \min_y \|Dy\|_2^2 &= \min_y \left(a_k^2 u_k^T \Big(\sum_{i=1}^{4}\sigma_i^2 u_i u_j^T\Big) u_k + v^T D^T D v \right) \\ &= \min_y \left(\underbrace{a_k^2 \sigma_k^2}_{\geq 0} + \underbrace{v^T D^T D v}_{\geq 0} \right) \\ &\geq a_k^2 \sigma_k^2 \text{，（当且仅当 } v=0 \text{，取到等号）} \end{aligned}$$

所以当且仅当 $v=0$，$\|Dy\|_2^2$ 取到最小值 $a_k^2\sigma_k^2$，

又因为当 $v=0$ 时，$y=a_k u_k$，而 u_k 为标准正交基，且 $\|y\|=1$，故 $a_k=1$，所以

$$\min_y \|Dy\|_2^2 \geq \sigma_k^2$$

当 σ_k 取最小奇异值 σ_4 时，$\|Dy\|_2^2$ 取到最小值 σ_4^2，此时

$$y = a_k u_k + v = 1\cdot u_4 + 0 = u_4$$ $\Longrightarrow$ 故 $y=u_4$ 为该最小二乘问题最优解

证毕。

Jindong_Shi
2022-01-23

2. 完成特征点三角化代码，并通过仿真测试

1) 三角化代码

根据课堂中给出的如下三角化公式:

记投影矩阵 $\mathbf{P}_k = [\mathbf{R}_k, \mathbf{t}_k] \in \mathbb{R}^{3\times4}$，为 World 系到 Camera 系投影关系：

$$\forall k, \lambda_k \mathbf{x}_k = \mathbf{P}_k \mathbf{y}.$$

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

我们在代码中构造了矩阵 $P_k = [R_{cw}, \ t_{cw}]$, 这里需要先将 R_{wc} 和 t_{wc}分别转化为R_{cw} 和 t_{cw}。
然后根据 P_k 和 路标点归一化坐标 uv 构造矩阵 D, 矩阵 D 的行数为该路标点的观测帧个数乘以 2，列数为 4。

```
/// 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 */
// 构造矩阵D
Eigen::Matrix<double, Eigen::Dynamic, 4> D;
int size = (end_frame_id - start_frame_id) * 2;
D.conservativeResize( rows: size,  cols: 4);
D.setZero();
for(int i = start_frame_id; i < end_frame_id; ++i)
{
    Eigen::Matrix<double, 3, 4> Pk;
    // From World to Camera:   Pw = Rwc * Pc + twc    ->     Pc = Rcw * (Pw - twc)
    auto Rcw :Transpose<...> = camera_pose[i].Rwc.transpose();
    auto tcw :Product<...> =  -Rcw * camera_pose[i].twc;
    Pk << Rcw, tcw;
    D.block( startRow: 2*(i-start_frame_id),  startCol: 0,  blockRows: 1,  blockCols: 4)
      = camera_pose[i].uv[0] * Pk.block( startRow: 2, startCol: 0, blockRows: 1, blockCols: 4) - Pk.block( startRow: 0, startCol: 0, blockRows: 1, blockCols: 4);
    D.block( startRow: 2*(i-start_frame_id)+1,  startCol: 0,  blockRows: 1,  blockCols: 4)
      = camera_pose[i].uv[1] * Pk.block( startRow: 2, startCol: 0, blockRows: 1, blockCols: 4) - Pk.block( startRow: 1, startCol: 0, blockRows: 1, blockCols: 4);
}
```

接着根据第 1 题中证明的结论，对矩阵 D^TD 做 SVD 分解 $D^TD = U\Sigma V^T = U\Sigma U^T$之后，$Dy = 0$ 的最小二乘解 y 就是 D^TD 的最小奇异值对应的奇异值向量，即矩阵 U 的最右边一列。由于这里解出的解满足 $\|y\| = 1$, 所以我们需要将该向量 y 除以 y 的最后一个元素来转化为齐次坐标，取齐次化的 y 的前三个元素，即路标点通过三角化估计得到的三维坐标。

```
// SVD分解
Eigen::JacobiSVD<Eigen::Matrix4d> svd( matrix: D.transpose() * D,  computationOptions: Eigen::ComputeFullU | Eigen::ComputeFullV);
Eigen::Vector4d singular_values = svd.singularValues();       // Singular values are always sorted in decreasing order
Eigen::Matrix4d U = svd.matrixU();
Eigen::Matrix4d V = svd.matrixV();

std::cout << "Singular values of Matrix DT*D: \n" << singular_values.transpose() << std::endl;
P_est = U.block( startRow: 0, startCol: 3, blockRows: 3, blockCols: 1) / U( row: 3, col: 3);

std::cout <<"ground truth: \n"<< Pw.transpose() <<std::endl;
std::cout <<"your result: \n"<< P_est.transpose() <<std::endl;

// 计算位移的RMSE
auto e :CwiseBinaryOp<...> = Pw - P_est;
double RMSE_translation = std::sqrt( x: e.transpose() * e);
std::cout <<"RMSE of translation: \n"<< RMSE_translation <<std::endl;

// TODO:: 请如课程讲解中提到的判断三角化结果好坏的方式，绘制奇异值比值变化曲线
// 最小奇异值和第二小奇异值的比值
double singular_value_ratio = singular_values[3] / singular_values[2];
std::cout << "singular_value_ratio: \n" << singular_value_ratio << std::endl;
```

我们这里在观测值上加上一些噪声，令总共的相机帧数为 10 帧，从第 3 帧开始观测到该路标点，则该路标点被检测到的帧数为 7 帧。我们计算估计的路标点坐标的 RMSE，以及最小奇异值和第二小奇异值的比例如下:

```
/home/jindong/VIO/Exercise/L6/course6_hw/app/estimate_depth
Singular values of Matrix DT*D:
    467.03     7.78234    0.712333 0.00788922
ground truth:
  -2.9477 -0.330799    8.43792
your result:
 -3.02521 -0.747822    9.01257
RMSE of translation:
0.714245
singular_value_ratio:
0.0110752
```

我们发现最小奇异值和第二小奇异值的比例在 1e-2 左右，三角化可视为有效。

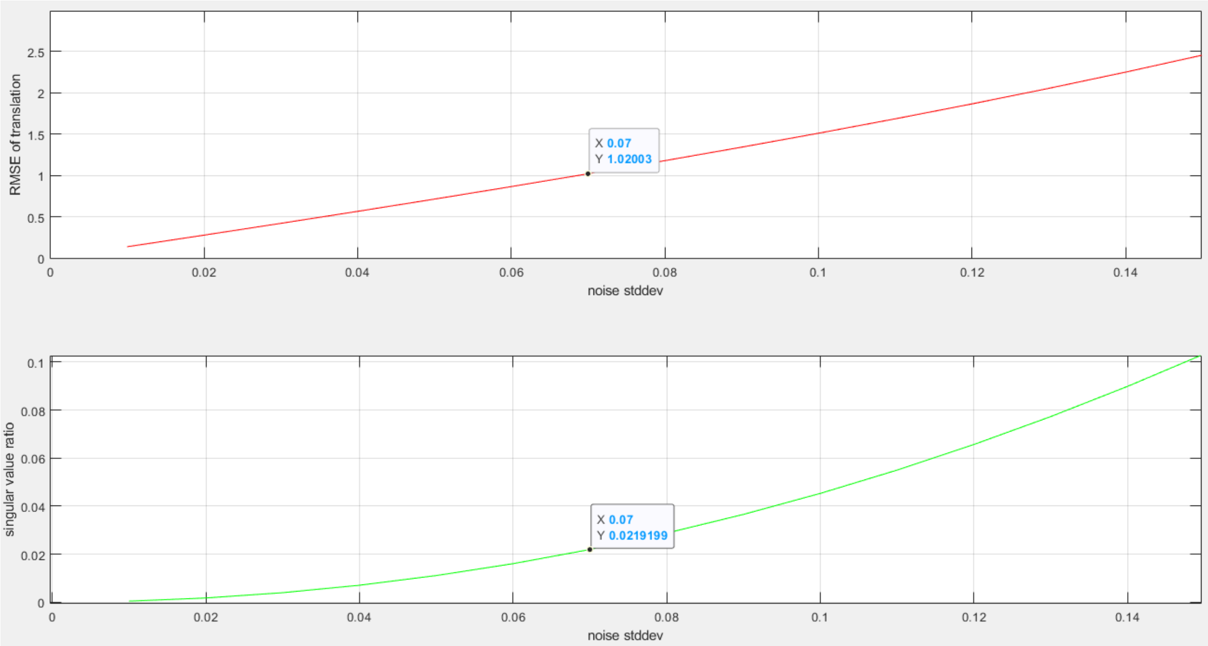
2) 测量值加上不同噪声(增大测量噪声方差)，观察最小奇异值和第二小奇异值之间的比例变化，并绘制变化曲线

我们固定该路标点被观测到的帧数为 7 帧，改变测量噪声的标准差:

```
// 相机测量噪声
double noise_stddev = 50./1000.;     // 相机噪声标准差
std::normal_distribution<double> noise_pdf( mean: 0.,  stddev: noise_stddev);

// 这个特征从第三帧相机开始被观测，i=3
int start_frame_id = 3;
int end_frame_id = poseNums;
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);        // Pw = Rwc * Pc + twc

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

    camera_pose[i].uv = Eigen::Vector2d( x: x/z + noise_pdf( &: generator),  y: y/z + noise_pdf( &: generator));
}
```

然后绘制不同噪声标准差下 (noise stddev = 0.01 ~ 0.15)，路标点三角化的 RMSE 和 最小奇异值和第二小奇异值的比例的变化情况，如下:

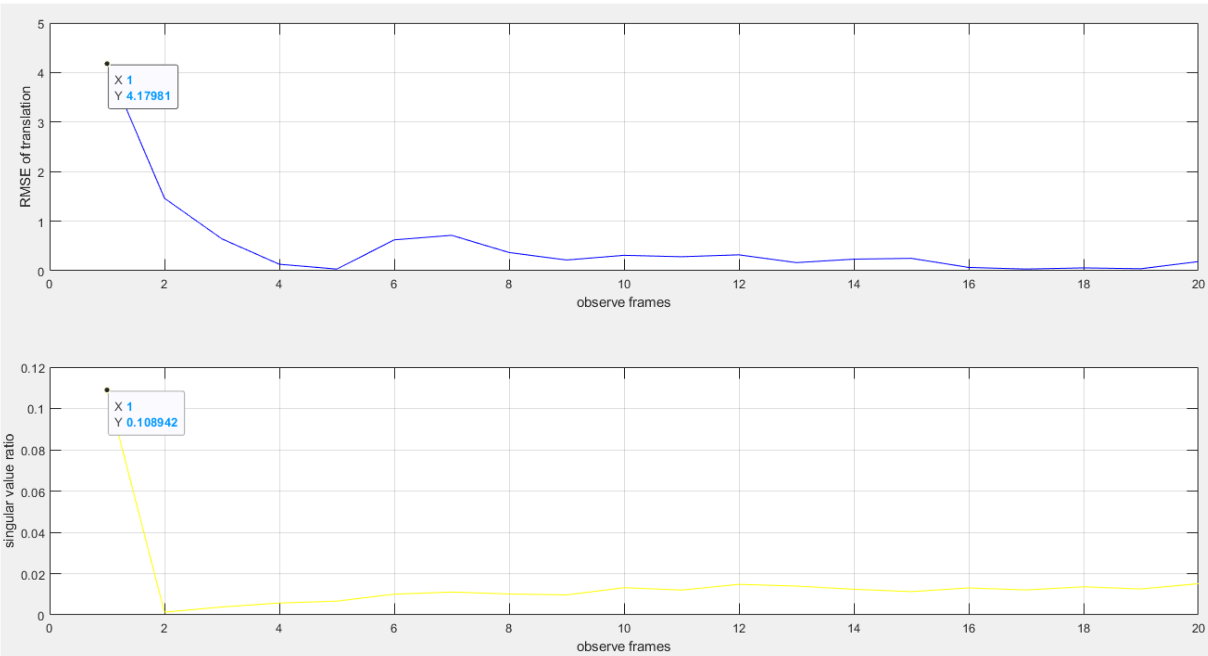
Jindong_Shi
2022-01-23



我们发现随着观测噪声增大，路标点三角化的 RMSE 增大，同时最小奇异值和第二小奇异值的比例也逐渐变大，可能最终使得三角化无效。

3) 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，并绘制变化曲线

我们固定测量噪声的标准差为 0.05, 改变该路标被观测到的帧数为 1 ~ 20 帧，绘制出路标点估计的 RMSE 以及最小奇异值和第二小奇异值的比例如下:



我们发现当该路标点只被 1 帧图像观测到时，由于矩阵 D 存在列不满秩，导致三角化的结果产生退化，该情况下的路标点估计的 RMSE 以及最小奇异值和第二小奇异值的比例 (≈ 0.11)都较大， 当该路标点被两帧即以上帧数的图像观测到后，三角化不产生退化，最小奇异值和第二小奇异值的比例都较小，在 1e-2 左右。随着该路标点被观测到的图像帧数的增加，该比例缓缓增加，但变化并不明显。