

1、证明如下

1. 证明:

$$\min_y \|Dy\|^2 = (Dy)^T(Dy) = y^T D^T D y, \quad \|y\|=1$$

$$y = \sum_{i=1}^4 k_i u_i = k_i u_i + v$$

将 y 代入 $y^T D^T D y$,

$$\begin{aligned} \min_y \|Dy\|^2 &= (k_i u_i + v)^T D^T D (k_i u_i + v) \\ &= k_i^2 u_i^T D^T D u_i + v^T D^T D v + k_i u_i^T D^T D v + k_i v^T D^T D u_i \end{aligned}$$

由于 u_i 和 v 正交, $D u_i = \sigma_i u_i$

$$\begin{aligned} \min_y \|Dy\|^2 &= k_i^2 u_i^T D^T D u_i + v^T D^T D v \\ &= k_i^2 \sigma_i^2 \|u_i\|^2 + v^T D^T D v \\ &\geq k_i^2 u_i^T D^T D u_i + v^T D^T D v = k_i^2 \sigma_i^2 \|u_i\|^2 \end{aligned}$$

当 $v=0$ 时等号成立.

取最小值, 则 $\sigma_i = 64$

$$\text{此时 } y = k_4 u_4 + v = k_4 u_4$$

由于 $\|y\|=1$, 则 $k_4=1$, 所以 $y=u_4$

2.1、代码如下

```

int k = 0 ;
for(int i = start_frame_id; i < end_frame_id; ++i)
{
    Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
    Eigen::Vector3d tcw = -Rcw * camera_pose[i].twc;
    P1 << Rcw.block<1,3>(0,0), tcw.x();
    P2 << Rcw.block<1,3>(1,0), tcw.y();
    P3 << Rcw.block<1,3>(2,0), tcw.z();
    D.block<1,4>(k,0) = camera_pose[i].uv.x() * P3 - P1;
    D.block<1,4>(k+1, 0) = camera_pose[i].uv.y() * P3 - P2;
    k+=2;
}

```

把相机位姿从世界坐标系变换到相机坐标系，得到投影矩阵 P_k ，并计算矩阵 D

```

Eigen::MatrixXd DTD = D.transpose() * D;
Eigen::JacobiSVD<Eigen::MatrixXd> svd(DTD, Eigen::ComputeThinU | Eigen::ComputeThinV);
Eigen::MatrixXd U = svd.matrixU();
P_est = U.block<3,1>(0,3) / U(3,3);

```

对 DTD 进行 SVD 分解，得到估计值，取齐次坐标的前三维，并同时除以第四维坐标
结果如下

```

ground truth:
-2.9477 -0.330799 8.43792
your result:
-2.9477 -0.330799 8.43792

```

在不加任何噪声的情况下三角化求得的空间点 P_{est} 与原空间点 P_w 完全相同

3.1 设置噪声如下

```

int start_frame_id = 0;
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);

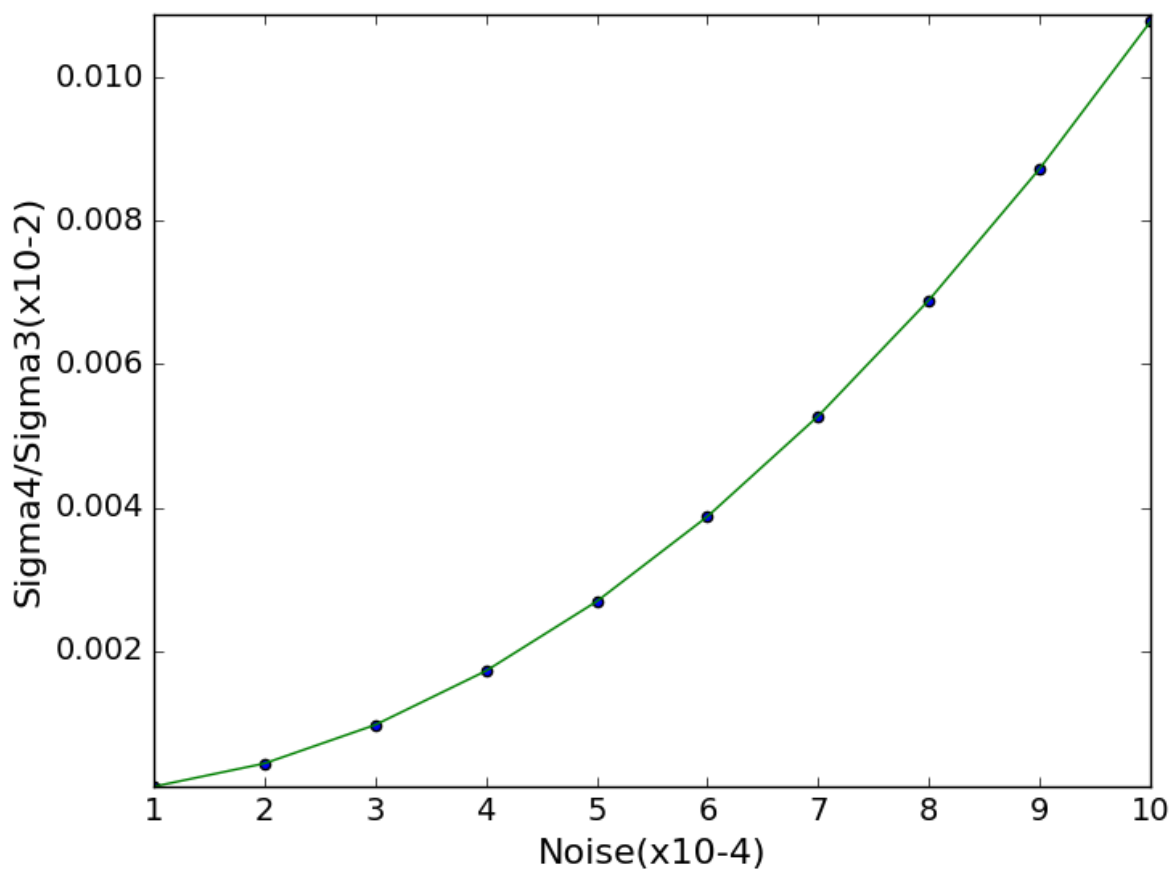
    std::normal_distribution<double> noise(0, 10./2000.);

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

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

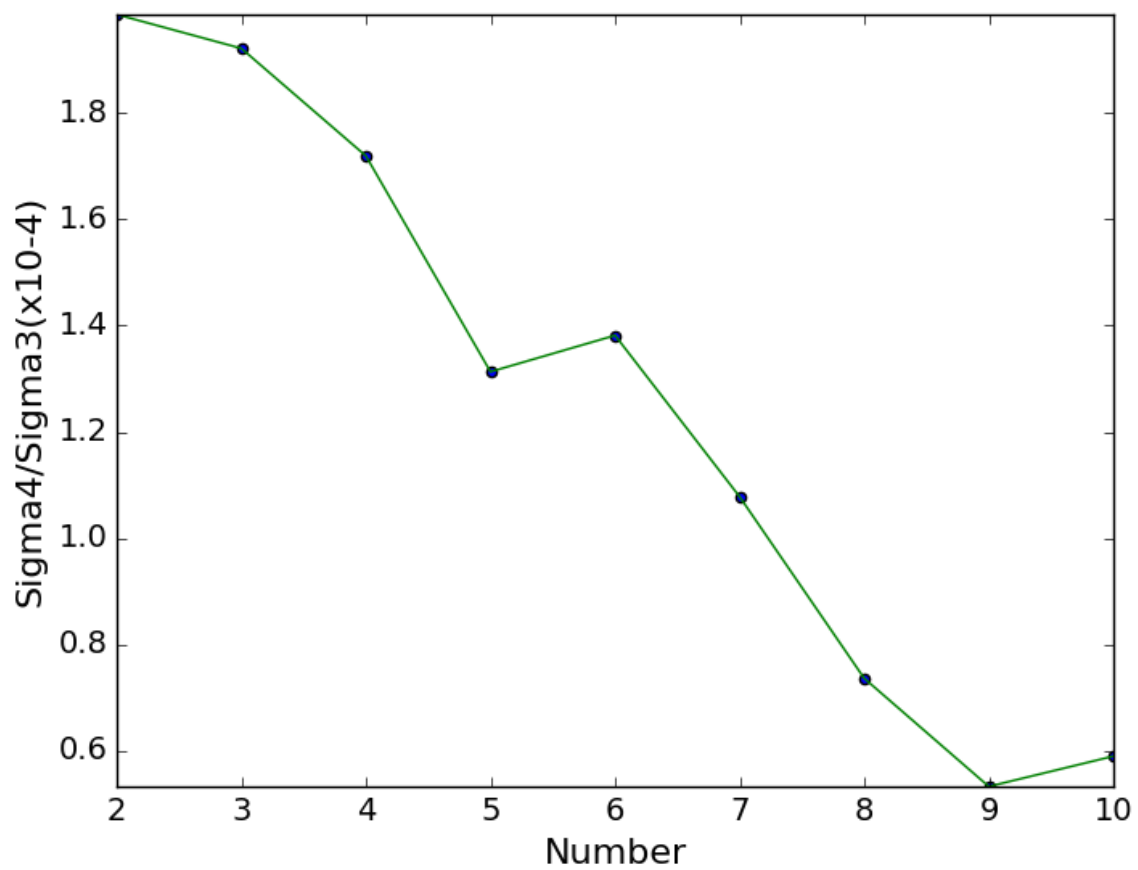
```

结果如下



随着方差的增大， $\text{Sigma4}/\text{Sigma3}$ 也在不断增大，说明三角化误差越来越大。

3.2 通过修改代码中的变量 `start_frame_id` 来设置观测帧数，结果如下



可以看出，所有 Sigma4/Sigma3 （最小奇异值和第二小奇异值之间的比例）都在 10^{-4} 量级，三角化都有效，随着观测帧数的增加， Sigma4/Sigma3 整体呈减小趋势，中间略有波动。