1.证明式(15)中，取$y=u_4$是该问题的最优解

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

y可以由D^TD的奇异值向量线性组合得到，即

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

其中 $v=\sum_{j=1,j\neq i}^{4} k_j u_j$ ，易知 u_i 与 v 正交

将 y 代入 y^TD^TDy 可得：

$$\min \|Dy\|^2 = (k_i u_i + v)^T D^T D (k_i u_i + v)$$

$$= k_i^2 u_i^T D^T D u_i + v^T D^T D v + k_i u_i^T D^T D v + k_i v^T D^T D u_i$$

由于 u_i 与 v 正交，后两项为 0 ，且 $Du_i=\sigma_i u_i$，带入有

$$\min_x \|Dy\|^2 = (k_i u_i + v)^T D^T D (k_i u_i + v)$$

$$= k_i^2 u_i^T D^T D u_i + v^T D^T D v$$

$$= k_i^2 \sigma_i^2 \|u_i\|^2 + v^T D^T D v \geqslant k_i^2 u_i^T D^T D u_i + v^T D^T D v$$

$$= k_i^2 \sigma_i^2 \|u_i\|^2$$，当且仅当 $v=0$ 等号成立

若要取最小值，则 $\sigma_i=\sigma_4$ ，也就是取最小奇异值的时候，

目标函数取最小值，此时

$$y = k_4 u_4 + v = k_4 u_4$$

由于 $\|y\|=1$ ，所以 $k_4=1$ ，故 $y=u_4$

2.完成特征点三角化代码，通过仿真测试

```
/* your code begin */
Eigen::Matrix<double, Eigen::Dynamic, 4> D(2 * (poseNums - start_frame_id), 4);
Eigen::RowVector4d P_1 = Eigen::RowVector4d::Zero(4);
Eigen::RowVector4d P_2 = Eigen::RowVector4d::Zero(4);
Eigen::RowVector4d P_3 = Eigen::RowVector4d::Zero(4);
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;
    P_1 << Rcw.block<1, 3>(0, 0), tcw.x();
    P_2 << Rcw.block<1, 3>(1, 0), tcw.y();
    P_3 << Rcw.block<1, 3>(2, 0), tcw.z();
    D.block<1, 4>(k, 0) = camera_pose[i].uv.x() * P_3 - P_1;
    D.block<1, 4>(k + 1, 0) = camera_pose[i].uv.y() * P_3 - P_2;
    k += 2;
}
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); //取齐次坐标的前三维，并同时除以第四维坐标
Eigen::Vector4d Singular_values = svd.singularValues();
std::cout << "Singular values:\n" << Singular_values << std::endl;
std::cout << "sigma4/sigma3:\n" << Singular_values[3] / Singular_values[2] << std::endl;
/* your code end */
```

仿真测试结果为：

```
xwl@xwl-Inspiron-15-7000-Gaming:~/Documents/VSLAM-fundamentals-and-VIO-learning/L15/course6_hw/build$ ./estimate_depth
Singular values:
    468.406
    7.74642
   0.723255
5.30104e-16
sigma4/sigma3:
7.32942e-16
ground truth:
  -2.9477 -0.330799   8.43792
your result:
  -2.9477 -0.330799   8.43792
```

3.对测量值增加不同噪声，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

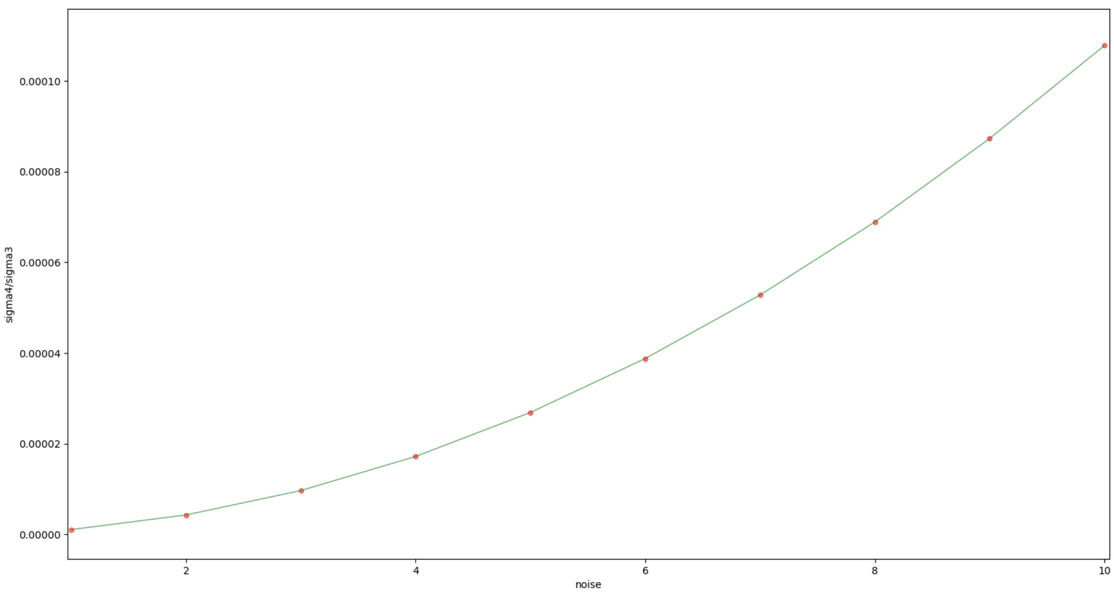
```
for (int i = start_frame_id; i < end_frame_id; ++i)
{
    Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
    Eigen::Vector3d Pc = Rcw * (Pw - camera_pose[i].twc);

    std::normal_distribution<double> noise_pdf(0, 1. / 2000.);

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

    // 给归一化坐标加上测量噪声
    camera_pose[i].uv = Eigen::Vector2d(x / z + noise_pdf(generator), y / z + noise_pdf(ge
}
```

比例值变化曲线为：



4.固定噪声方差参数，将观测图像帧扩成多帧，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

噪声误差设定为5个像素误差(5./2000.)，start_frame_id设为1-10，取为10时比例值变为nan，9时比例值为0.885821，明显增大