深蓝学院
shenlanxueyuan.com

机器人学中的状态估计

第七次作业讲评

主讲人 顾津铭

习题1

当 $\frac{\partial J}{\partial \boldsymbol{r}^{\mathrm{T}}}=\boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M}w_j\left(\boldsymbol{y}_j^{\oplus}-(\boldsymbol{p}_j-\boldsymbol{r})^{+}\right)\boldsymbol{q}=0$ 时，可得 $\boldsymbol{r}=\boldsymbol{p}-\boldsymbol{q}^{+}\boldsymbol{y}^{+}\boldsymbol{q}^{-1}$，

其中 $\boldsymbol{y}=\frac{1}{w}\sum_{j=1}^{M}w_j\boldsymbol{y}_j,\quad \boldsymbol{p}=\frac{1}{w}\sum_{j=1}^{M}w_j\boldsymbol{p}_j,\quad w=\frac{1}{w}\sum_{j=1}^{M}w_j$。

作业概况： 完成情况很好；

证明思路： 灵活运用四元数的左手算子与右手算子的性质进行证明。

已知 $\boldsymbol{u}^{+}\boldsymbol{v}\equiv\boldsymbol{v}^{\oplus}\boldsymbol{u},\ \boldsymbol{u}^{+}\boldsymbol{v}^{\oplus}\equiv\boldsymbol{v}^{\oplus}\boldsymbol{u}^{+}$

$$
\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{r}^{\mathrm{T}}} &= \boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M} w_j\left(\boldsymbol{y}_j^{\oplus}-(\boldsymbol{p}_j-\boldsymbol{r})^{+}\right)\boldsymbol{q} \\
&= \boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M} w_j\boldsymbol{y}_j^{\oplus}\boldsymbol{q}-\boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M} w_j(\boldsymbol{p}_j-\boldsymbol{r})^{+}\boldsymbol{q} \\
&= \boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M} w_j\boldsymbol{q}^{+}\boldsymbol{y}_j-\boldsymbol{q}^{-1\oplus}\sum_{j=1}^{M} w_j\boldsymbol{q}^{\oplus}(\boldsymbol{p}_j-\boldsymbol{r}) \\
&= \boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{+}\sum_{j=1}^{M} w_j\boldsymbol{y}_j-\boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{\oplus}\sum_{j=1}^{M} w_j\boldsymbol{p}_j+\boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{\oplus}\sum_{j=1}^{M} w_j\boldsymbol{r}
\end{aligned}
$$

$$
\boldsymbol{q}=\begin{bmatrix}\boldsymbol{\varepsilon}\\ \eta\end{bmatrix},\quad \boldsymbol{q}^{\oplus}=\begin{bmatrix}\eta\boldsymbol{I}+\boldsymbol{\varepsilon}^{\wedge} & \boldsymbol{\varepsilon}\\ -\boldsymbol{\varepsilon}^{\mathrm{T}} & \eta\end{bmatrix},\quad \boldsymbol{q}^{-1}=\begin{bmatrix}-\boldsymbol{\varepsilon}\\ \eta\end{bmatrix},\quad \boldsymbol{q}^{-1\oplus}=\begin{bmatrix}\eta\boldsymbol{I}-\boldsymbol{\varepsilon}^{\wedge} & -\boldsymbol{\varepsilon}\\ \boldsymbol{\varepsilon}^{\mathrm{T}} & \eta\end{bmatrix}
$$

$$
\boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{\oplus}=\begin{bmatrix}\eta\boldsymbol{I}-\boldsymbol{\varepsilon}^{\wedge} & -\boldsymbol{\varepsilon}\\ \boldsymbol{\varepsilon}^{\mathrm{T}} & \eta\end{bmatrix}\begin{bmatrix}\eta\boldsymbol{I}+\boldsymbol{\varepsilon}^{\wedge} & \boldsymbol{\varepsilon}\\ -\boldsymbol{\varepsilon}^{\mathrm{T}} & \eta\end{bmatrix}=\boldsymbol{I}
$$

$\boldsymbol{y}=\frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{y}_j,\quad \boldsymbol{p}=\frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{p}_j,\quad w=\frac{1}{w}\sum_{j=1}^{M} w_j$，可得

$$
\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{r}^{\mathrm{T}}} &= \boldsymbol{q}^{-1\oplus}\boldsymbol{q}^{+}w\boldsymbol{y}-w\boldsymbol{p}+w\boldsymbol{r} \\
&= w(\boldsymbol{q}^{+}\boldsymbol{q}^{-1\oplus}\boldsymbol{y}-\boldsymbol{p}+\boldsymbol{r}) \\
&= w(\boldsymbol{q}^{+}\boldsymbol{y}^{+}\boldsymbol{q}^{-1}-\boldsymbol{p}+\boldsymbol{r})
\end{aligned}
$$

令 $w(\boldsymbol{q}^{+}\boldsymbol{y}^{+}\boldsymbol{q}^{-1}-\boldsymbol{p}+\boldsymbol{r})=0$，可得 $\boldsymbol{r}=\boldsymbol{p}-\boldsymbol{q}^{+}\boldsymbol{y}^{+}\boldsymbol{q}^{-1}$。

习题2

深蓝学院 shenlanxueyuan.com

$$\frac{1}{\omega}\sum_{j=1}^{M}\omega_j \boldsymbol{z}_j^{\odot^{\mathrm{T}}}\boldsymbol{z}_j^{\odot}=\mathcal{T}_{op}^{-\mathrm{T}}\left(\frac{1}{\omega}\sum_{j=1}^{M}\omega_j \boldsymbol{p}_j^{\odot^{\mathrm{T}}}\boldsymbol{p}_j^{\odot}\right)\mathcal{T}_{op}^{-1}$$

作业概况：完成情况很好；

证明思路：1、使用上一章中证明出的结论进行证明；2、将每一项都写成矩阵的形式，通过逐步运算推导出左右相等。

已知 $\boldsymbol{z}_j = \boldsymbol{T}_{op}\boldsymbol{p}_j$， $\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^{\mathrm{T}}} \boldsymbol{z}_j^{\odot} = \frac{1}{w}\sum_{j=1}^{M} w_j \left(\boldsymbol{T}_{op}\boldsymbol{p}_j\right)^{\odot^{\mathrm{T}}} \left(\boldsymbol{T}_{op}\boldsymbol{p}_j\right)^{\odot}$

因为 $(\mathbf{T}\boldsymbol{p})^{\odot^{\mathrm{T}}}(\mathbf{T}\boldsymbol{p})^{\odot} = \mathcal{T}^{-\mathrm{T}}\boldsymbol{p}^{\odot^{\mathrm{T}}}\boldsymbol{p}^{\odot}\mathcal{T}^{-1}$，所以

$$
\begin{aligned}
\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^{\mathrm{T}}} \boldsymbol{z}_j^{\odot} &= \frac{1}{w}\sum_{j=1}^{M} w_j \mathcal{T}_{op}^{-\mathrm{T}} \boldsymbol{p}_j^{\odot^{\mathrm{T}}} \boldsymbol{p}_j^{\odot} \mathcal{T}_{op}^{-1} \\
&= \mathcal{T}_{op}^{-\mathrm{T}} \left(\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{p}_j^{\odot^{\mathrm{T}}} \boldsymbol{p}_j^{\odot} \right) \mathcal{T}_{op}^{-1}
\end{aligned}
$$

习题3

$$\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^{\mathrm{T}}}(\boldsymbol{y}_j-\boldsymbol{z}_j)=\begin{bmatrix}\boldsymbol{y}-\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})\\ \boldsymbol{b}-\boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})\end{bmatrix}$$

作业概况： 完成情况很好；

证明思路： 将每一项都写成矩阵的形式，通过逐步运算推导出左右相等。

已知 $\boldsymbol{z}_j = \boldsymbol{T}_{op}\boldsymbol{p}_j = \begin{bmatrix} \boldsymbol{C}_{op} & -\boldsymbol{C}_{op}\boldsymbol{r}_{op} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{p}_j \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \\ 1 \end{bmatrix}$

$$\boldsymbol{z}_j^{\odot} = \begin{bmatrix} \boldsymbol{1} & -(\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}))^{\wedge} \\ \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} \end{bmatrix}, \ \boldsymbol{z}_j^{\odot^{\mathrm{T}}} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} \\ (\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}))^{\wedge} & \boldsymbol{0} \end{bmatrix}$$

$$\begin{aligned}
\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^{\mathrm{T}}}(\boldsymbol{y}_j - \boldsymbol{z}_j) &= \frac{1}{w}\sum_{j=1}^{M} w_j \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} \\ (\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}))^{\wedge} & \boldsymbol{0} \end{bmatrix}\left(\begin{bmatrix} \boldsymbol{y}_j \\ 1 \end{bmatrix} - \begin{bmatrix} \boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \\ 1 \end{bmatrix}\right) \\
&= \frac{1}{w}\sum_{j=1}^{M} w_j \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} \\ (\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}))^{\wedge} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{y}_j - \boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \\ 0 \end{bmatrix} \\
&= \frac{1}{w}\sum_{j=1}^{M} w_j \begin{bmatrix} \boldsymbol{y}_j - \boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \\ (\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}))^{\wedge}\boldsymbol{y}_j - (\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}))^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \end{bmatrix} \\
&= \frac{1}{w}\sum_{j=1}^{M} w_j \begin{bmatrix} \boldsymbol{y}_j - \boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) \\ -\boldsymbol{y}_j^{\wedge}\boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{w}\sum_{j=1}^{M} w_j \begin{bmatrix} \boldsymbol{y}_j - \boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) \\ -(\boldsymbol{y}_j^{\wedge} - \boldsymbol{y}^{\wedge})\boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) - \boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{y}_j - \boldsymbol{C}_{\mathrm{op}}\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) \\ -\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j - \boldsymbol{y})^{\wedge}\boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) - \frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j - \boldsymbol{r}_{op}) \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{y} - \boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p} - \boldsymbol{r}_{\mathrm{op}}) \\ -\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j - \boldsymbol{y})^{\wedge}\boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j - \boldsymbol{r}_{\mathrm{op}}) - \boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p} - \boldsymbol{r}_{op}) \end{bmatrix}
\end{aligned}
$$

则 $-\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j-\boldsymbol{y})^{\wedge}\boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}_j-\boldsymbol{r}_{\mathrm{op}})$ 的第 i 行可表示为

$$
\begin{aligned}
&\boldsymbol{I}_i^{\mathrm{T}}\left(-\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j-\boldsymbol{y})^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})\right)\\
&=\frac{1}{w}\sum_{j=1}^{M} w_j(\boldsymbol{y}_j-\boldsymbol{y})^{\mathrm{T}}\boldsymbol{I}_i^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}_j-\boldsymbol{p})\\
&=\frac{1}{w}\sum_{j=1}^{M} w_j\operatorname{tr}(\boldsymbol{I}_i^{\wedge}C_{op}(\boldsymbol{p}_j-\boldsymbol{p})(\boldsymbol{y}_j-\boldsymbol{y})^{\mathrm{T}})\\
&=\operatorname{tr}(\boldsymbol{I}_i^{\wedge}\boldsymbol{C}_{op}\boldsymbol{W}^{\mathrm{T}})
\end{aligned}
$$

$$
\boldsymbol{b}=\left[\operatorname{tr}(\boldsymbol{I}_i^{\wedge}\boldsymbol{C}_{op}\boldsymbol{W}^{\mathrm{T}})\right]_i
$$

$$
\frac{1}{w}\sum_{j=1}^{M} w_j\boldsymbol{z}_j^{\odot^{\mathrm{T}}}(\boldsymbol{y}_j-\boldsymbol{z}_j)=\begin{bmatrix}\boldsymbol{y}-\boldsymbol{C}_{\mathrm{op}}(\boldsymbol{p}-\boldsymbol{r}_{\mathrm{op}})\\ \boldsymbol{b}-\boldsymbol{y}^{\wedge}\boldsymbol{C}_{op}(\boldsymbol{p}-\boldsymbol{r}_{op})\end{bmatrix}
$$

习题4

深蓝学院 shenlanxueyuan.com

作业说明： 写一段ICP程序，完成两个PCD文件的Pose计算，不允许使用第三方点云库 。

编程思路： 1、使用PCL库实现点云的加载，显示，保存和旋转平移等操作；

2、使用“**pcl::registration::CorrespondenceEstimation**”、“**pcl::KdTreeFLANN**”或其他方法实现点云的匹配，匹配好的点云作为ICP算法的输入量；

3、不借助第三方点云库，实现ICP算法，可以使用以SVD为代表的代数方法，也可以使用非线性优化的方法求解点云Pose。

ICP算法流程

算法：迭代最近点算法（Iterative Closest Point, ICP）

输入：源点云 $\boldsymbol{P}_s$，目标点云 $\boldsymbol{P}_t$，两组点云之间的对应点集合 cor

$\mathbf{T}_{ts}{}^{1} \leftarrow \boldsymbol{I}_{4\times4}$

while 未达到迭代次数或迭代精度

$\boldsymbol{P}_s{}^{r} \leftarrow \{\boldsymbol{p} \mid \boldsymbol{P}_s \cap \mathrm{cor}\}$，$\boldsymbol{P}_t{}^{r} \leftarrow \{\boldsymbol{p} \mid \boldsymbol{P}_t \cap \mathrm{cor}\}$ //使用匹配集合中的点

$c_s \leftarrow \mathrm{getCentroid}(\boldsymbol{P}_s{}^{r})$，$c_t \leftarrow \mathrm{getCentroid}(\boldsymbol{P}_t{}^{r})$ //计算质心

$\boldsymbol{Q}_s \leftarrow \boldsymbol{P}_s{}^{r} - c_s$，$\boldsymbol{Q}_t \leftarrow \boldsymbol{P}_t{}^{r} - c_t$ //去除质心

$\boldsymbol{W} = \boldsymbol{Q}_s \boldsymbol{Q}_s{}^{\mathrm{T}}$ //计算 W

$\boldsymbol{U}, \boldsymbol{V} \leftarrow \mathrm{SVD}(\boldsymbol{W})$ //对 W 进行 SVD 分解

$\boldsymbol{R} \leftarrow \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$，$\boldsymbol{t} \leftarrow c_t - \boldsymbol{R}c_s$ //计算旋转矩阵 R 和平移向量 t

$\mathbf{T}_{ts} \leftarrow \boldsymbol{R}, \boldsymbol{t}$ //变换矩阵

$\mathbf{T}_{ts}{}^{k} \leftarrow \mathbf{T}_{ts}{}^{k-1} \cdot \mathbf{T}_{ts}$ //迭代更新

end while

输出：变换矩阵 $\mathbf{T}_{ts}{}^{k}$

作业示例

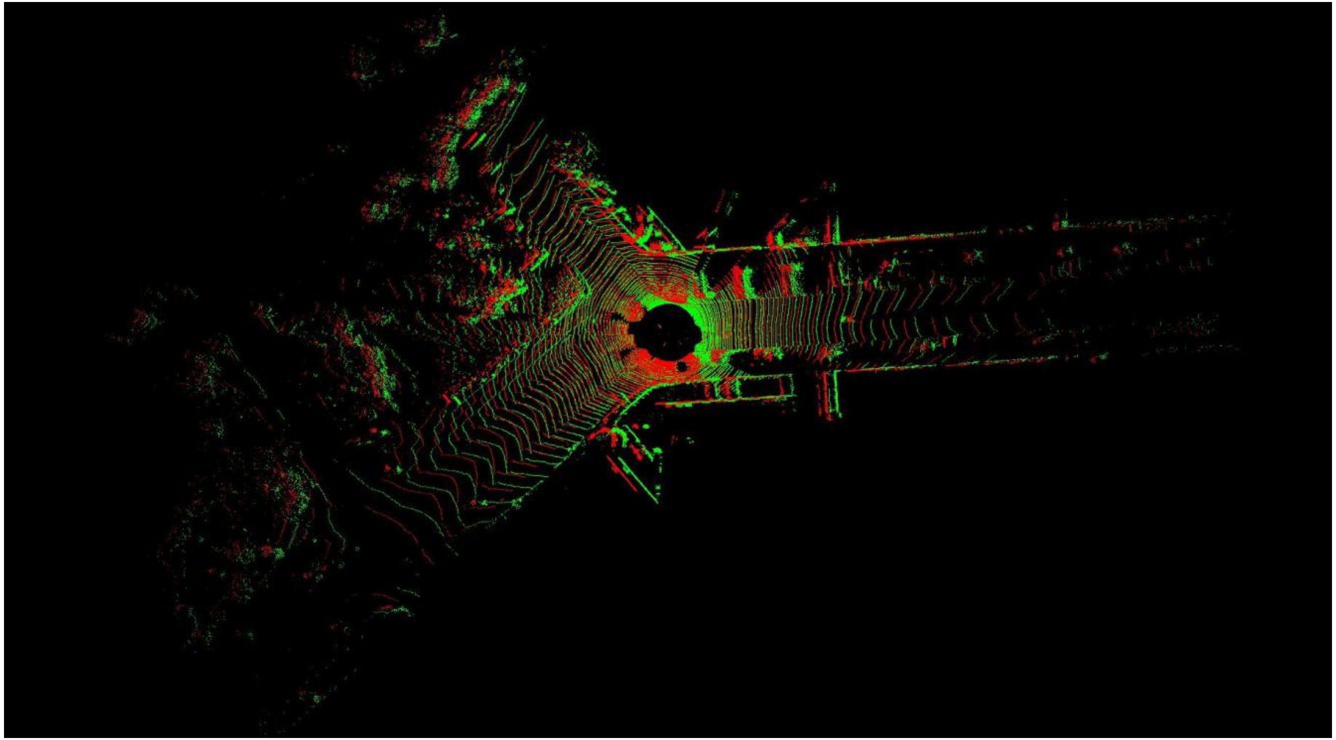
```
void testICP::calculateRegister(PointType::Ptr first,PointType::Ptr second,pcl::Correspondences& cor)
{
    Eigen::Vector3d p1(0,0,0), p2(0,0,0);
    int N = cor.size();
    //第一：计算质心
    for (size_t i = 0; i < N; ++i)
    {
        pcl::PointXYZ& f1 = first->at (cor[i].index_query);
        pcl::PointXYZ& f2 = second->at (cor[i].index_match);
        p1 += Eigen::Vector3d(f1.x,f1.y,f1.z);
        p2 += Eigen::Vector3d(f2.x,f2.y,f2.z);
    }
    p1 = p1 / N;
    p2 = p2 / N;
    // 第二：去质心
    std::vector<Eigen::Vector3d> q1(N), q2(N);
    for ( int i = 0; i < N; i++ )
    {
        pcl::PointXYZ& f1 = first->at (cor[i].index_query);
        pcl::PointXYZ& f2 = second->at (cor[i].index_match);
        q1[i] = Eigen::Vector3d(f1.x,f1.y,f1.z) - p1;
        q2[i] = Eigen::Vector3d(f2.x,f2.y,f2.z) - p2;
    }
```

作业示例

```
    // 第三：计算W
    Eigen::Matrix3d W = Eigen::Matrix3d::Zero();
    for ( int i=0; i<N; i++ )
    {
        W += q1[i]*q2[i].transpose();
    }
    // 第四：通过SVD分解W，为U V
    Eigen::JacobiSVD<Eigen::Matrix3d> svd ( W, Eigen::ComputeFullU|Eigen::ComputeFullV );
    Eigen::Matrix3d U = svd.matrixU();
    Eigen::Matrix3d V = svd.matrixV();
    // 第五：计算R = U*S*V',t = p - R*y
    if (U.determinant () * V.determinant () < 0)
    {
        for (int x = 0; x < 3; ++x)
            V (x, 2) *= -1;
    }
    Eigen::Matrix3d R = U * V.transpose();
    Eigen::Vector3d  t   = p1 - R * p2;
    //第六：得到transformation_matrix
    transformation_matrix.topLeftCorner (3, 3) = R;
    transformation_matrix.block (0, 3, 3, 1) = t;
}
```

运行结果

0.999975	0.00712412	0.00228177	-1.27099
-0.00712773	0.999977	0.00158097	-0.00147516
-0.00227045	-0.0015972	0.999999	-0.00269849
0	0	0	1



深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening