CIS 580 Machine Perception HW4

Chun Chang

CIS 580 Machine Perception HW4

Chun Chang

1. ① rotate to the pose shown in figure

② rotate along current y-axis.

③ rotate along current x-axis.

$$R = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R_Y(\alpha) \end{bmatrix} \begin{bmatrix} R_x(\beta) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c\alpha & 0 & s\alpha \\ 0 & 1 & 0 \\ -s\alpha & 0 & c\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix}$$

$$= \begin{bmatrix} -\sin\alpha, & \cos\alpha\sin\beta, & \cos\alpha\cos\beta \\ \cos\alpha, & \sin\alpha\sin\beta, & \sin\alpha\cos\beta \\ 0, & \cos\beta, & -\sin\beta \end{bmatrix}$$

$${}^{W}T_c = \begin{bmatrix} R_z(\alpha) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -b \\ h \end{bmatrix} = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -b \\ h \end{bmatrix} = \begin{bmatrix} b\sin\alpha \\ -b\cos\alpha \\ h \end{bmatrix}$$

#

2. move & rotate on a plane parallel to the image plane.

$\Rightarrow t_z = 0$, rotation along with z-axis.

$$E = \hat{t} \cdot R = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } t_z = 0$$

$$= \begin{bmatrix} 0 \; , & 0 \; , & t_y \\ 0 \; , & 0 \; , & -t_x \\ -t_y c\theta + t_x s\theta \; , & t_y s\theta + t_x c\theta \; , & 0 \end{bmatrix}$$

$$\begin{cases} a = t_y \; , & -a\cos\theta - b\sin\theta = c \\ b = -t_x \; , & a\sin\theta - b\cos\theta = d. \end{cases}$$

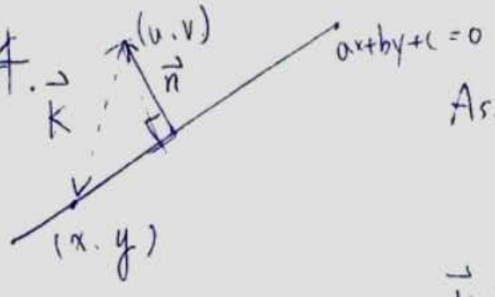
$$\underline{t = \begin{bmatrix} -b \\ a \\ 0 \end{bmatrix}} \quad \underline{R = \begin{bmatrix} \frac{-(ac+bd)}{a^2+b^2} \; , & -\frac{(ad-bc)}{a^2+b^2} & 0 \\ \frac{ad-bc}{a^2+b^2} \; , & \frac{-(ac+bd)}{a^2+b^2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ where } \theta = \sin^{-1}\left(\frac{-b^2}{\sqrt{c^2+d^2}}\right) - \cos^{-1}\left(\frac{c}{\sqrt{c^2+d^2}}\right)}$$

3. Since R is orthogonal $\Rightarrow R^T \cdot R = I$, $\widehat{Ra} = R\hat{a} \cdot R^T$

$$R(a \times b) = R\hat{b} \cdot a = R\hat{b} \cdot \underbrace{(R^T \cdot R)}_{}^{I} a = \widehat{Rb} \cdot Ra$$

$$\underline{= Ra \times Rb}$$

#

4. 

Assume . a line $\vec{l} = (a.b)$. $ax+by+c=0 \in \mathbb{R}^2$

anoutlier (u,v). . a point on the line $(x.y)$

$$\vec{K} = (x-u,\ y-v)$$

distance from outlier to the line

$$= \frac{\vec{l}\cdot\vec{K}}{|\vec{l}|} = \frac{(x-u\ .\ y-v.\ 0)(a\ b\ 0)}{\sqrt{a^2+b^2}}$$

$$= \frac{-(au+bv+c)}{\sqrt{a^2+b^2}}$$

$$e^3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$d^2 = \frac{(au+bv+c)^2}{a^2+b^2}$$

$$= \frac{\left[(u.v.1)\cdot(a.b.0)\right]^2}{\left\|\begin{pmatrix} -b \\ a \\ 0 \end{pmatrix}\right\|^2}$$

$$= \frac{(\tilde{x}^T\cdot l)^2}{\|\hat{e}_3\, l\|^2} \qquad \tilde{x}^T = (u.v.1)$$

#

2.3

1. Since $E = U \cdot S \cdot V^T$, U, S, V are the components of sigular-value decomposition of essential matrix E

$$E = U \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot V^T = \sigma U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

$$\because \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \hat{T}_z^T \cdot R_z(\tfrac{\pi}{2}) \text{, here } T_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$E = \sigma U \hat{T}_z^T \cdot R_z(\tfrac{\pi}{2}) V^T = \sigma U \hat{T}_z^T \cdot \underbrace{[U^T \cdot U]}^{I} \cdot R_z(\tfrac{\pi}{2}) \cdot V^T$$

$$= -\sigma \, \widehat{UT_z} \cdot U R_z(\tfrac{\pi}{2}) V^T$$

$$UT_z = U \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \Rightarrow \widehat{UT_z} \text{ is only related last column of } U$$

b). Since $R_z(\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $R_z(-\frac{\pi}{2}) = R_z(\frac{\pi}{2})^T = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$E = -(-\widehat{UT_z'}) \cdot U R_z(-\tfrac{\pi}{2}) \cdot V^T \Rightarrow \hat{T}_z^{T\,'} = -\hat{T}_z = -\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow T_z' = -T_z$$

2.3.2.

from Rodrigue's formula

$$R_T(\pi) = I\cos\pi + T\cdot T^T\cdot(1-\cos\pi) + \hat{T}\sin\pi$$

$$= \left(2TT^T - I\right).$$

from last question $U\begin{bmatrix}0\\0\\1\end{bmatrix} = T$

$$R_T(\pi) = \left(2\,U\begin{bmatrix}0\\0\\1\end{bmatrix}\cdot[0,0,1]\,U^T - I\right) = \left(2\,U\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}U^T - I\right)$$

$$\hat{T} = U R_Z(\tfrac{\pi}{2})\cdot\Sigma U^T$$

$$R_T(\pi)\cdot\hat{T} = 2U\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}\underbrace{U^TU}_{\to I}R_Z(\tfrac{\pi}{2})\cdot\Sigma U^T - UR_Z(\tfrac{\pi}{2})\Sigma U^T$$

$$= U\left(\begin{bmatrix}0&0&0\\0&0&0\\0&0&2\end{bmatrix} - \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\right)R_Z(\tfrac{\pi}{2})\Sigma U^T$$

$$= U\begin{pmatrix}-1&0&0\\0&-1&0\\0&0&1\end{pmatrix}R_Z(\tfrac{\pi}{2})\Sigma U^T, \quad \because \begin{pmatrix}-1&0&0\\0&-1&0\\0&0&1\end{pmatrix}\cdot\underbrace{\begin{pmatrix}0&-1&0\\1&0&0\\0&0&1\end{pmatrix}}_{R_Z(\frac{\pi}{2})} = \begin{pmatrix}0&1&0\\-1&0&0\\0&0&1\end{pmatrix}$$

$$= R_Z(-\tfrac{\pi}{2})$$

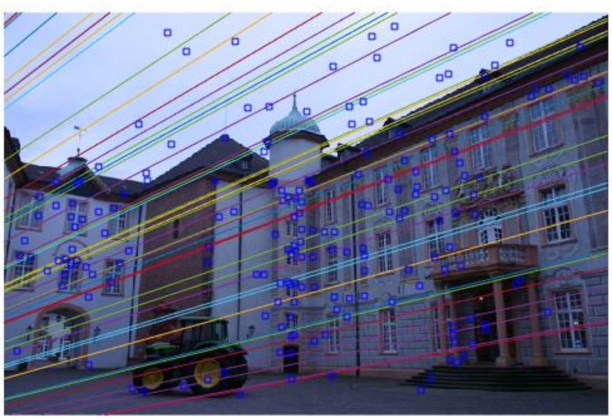
$$\Rightarrow U\cdot R_Z(-\tfrac{\pi}{2})\cdot\Sigma U^T \quad \#/$$

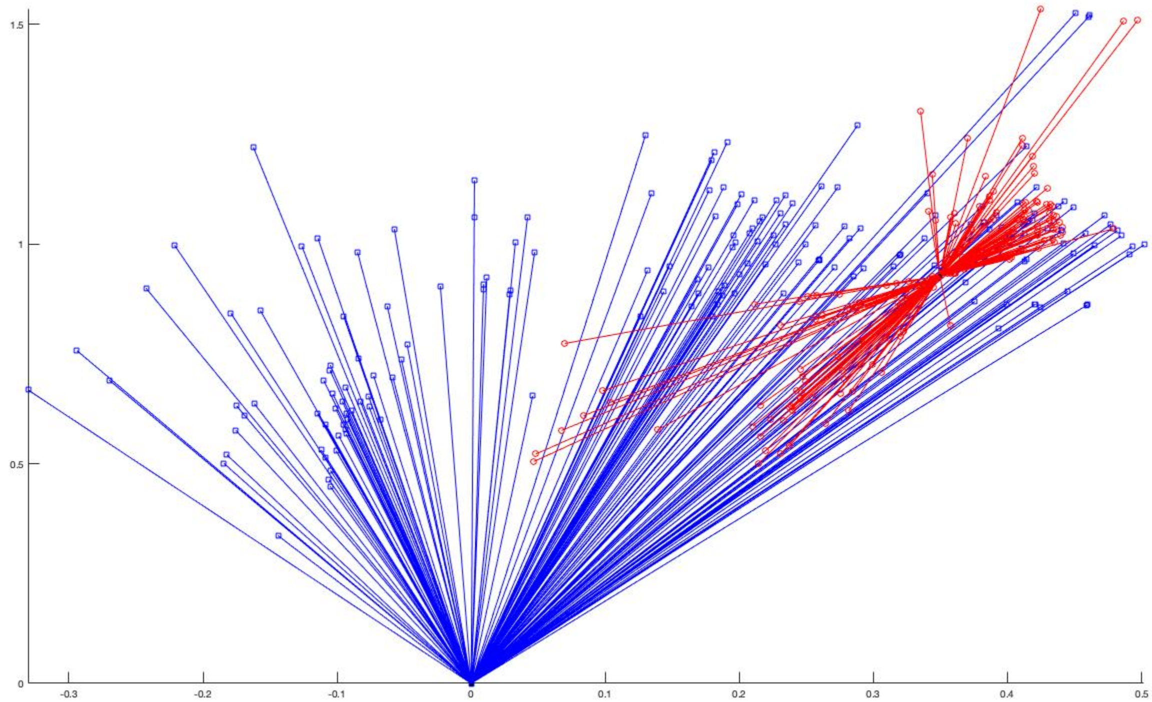
2-0 Raw data processing

```
U11 = [U1, ones(size(U1,1),1)];
U22 = [U2, ones(size(U2,1),1)];

X1 = inv(K)*U11';  % as a function of U1 and K
X2 = inv(K)*U22';  % as a function of U2 and K

X1 = X1 ./ X1(3,:);
X2 = X2 ./ X2(3,:);

X1 = X1(1:2,:)';
X2 = X2(1:2,:)';
```

2-1. Estimate Essential Matrix

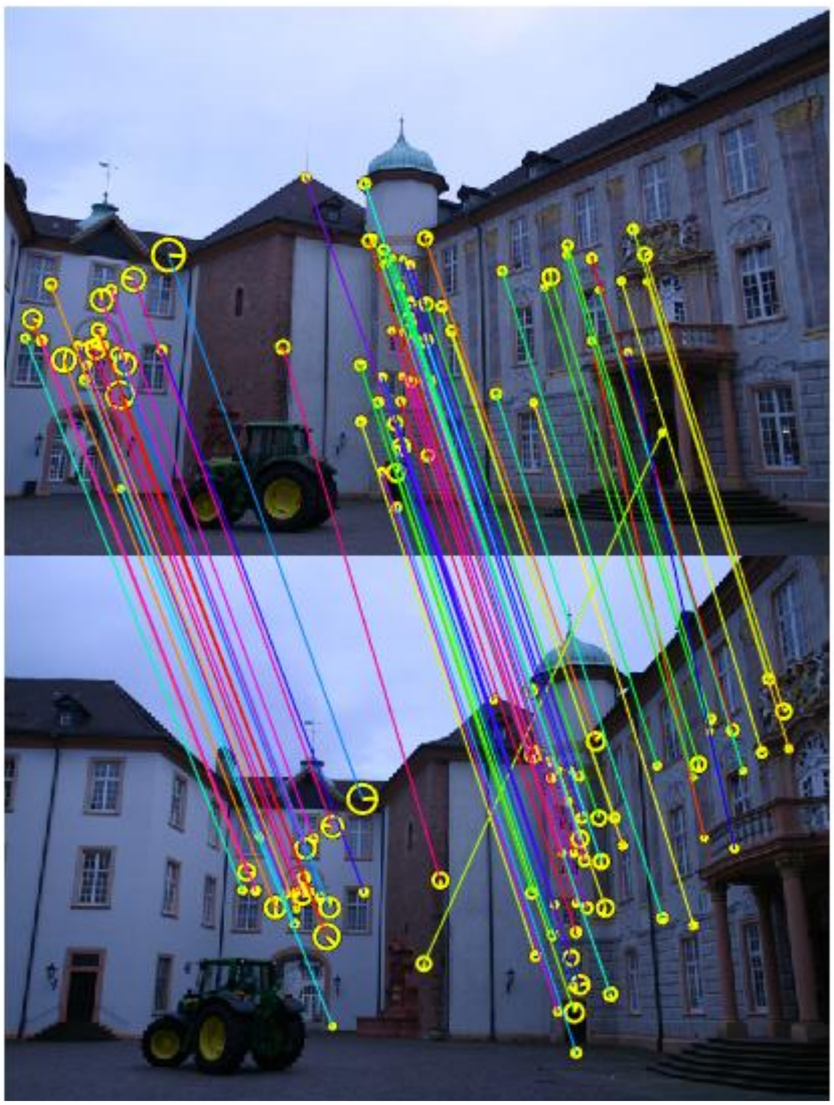
```
X11 = [X1, ones(size(X1,1),1)];
X22 = [X2, ones(size(X2,1),1)];
a = [X11(:,1) .* X22, X11(:,2).* X22, X11(:,3).*X22 ]; % Kronecker products of points
in C1 and C2

[U,S,V] = svd(a);

E = reshape(V(:,9),3,3); % extract E' from V matrix

% Project E on the space of essential matrices
[U1, S1, V1] = svd(E);

E = U1*diag([1,1,0])* V1';
```




1.5
1
0.5
0
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

2-2. Estimate Essential Matrix by Ransac

```
x1_s = X1(sampleInd,1:2);
x2_s = X2(sampleInd,1:2);
E_sample = estimateEmatrix(x1_s, x2_s);
X1_r = X1(testInd,:);
X2_r = X2(testInd,:);
l1 = X2_r * E_sample; % epiline1
l2 = (E_sample * X1_r')'; % epiline2
e3 = [0, -1, 0;
    1, 0, 0;
    0, 0, 0];
% vectorized distance to epi line
d1 = sum((X1_r .* l1),2).^2 ./ sum((l1 * e3').^2,2);
d2 = sum((X2_r .* l2),2).^2 ./ sum((l2 * e3').^2,2);

residuals = d1 + d2;

rr(testInd,:) = residuals;
[curInliers,~] = find(rr < eps);
```



2-3. Draw epipolar line

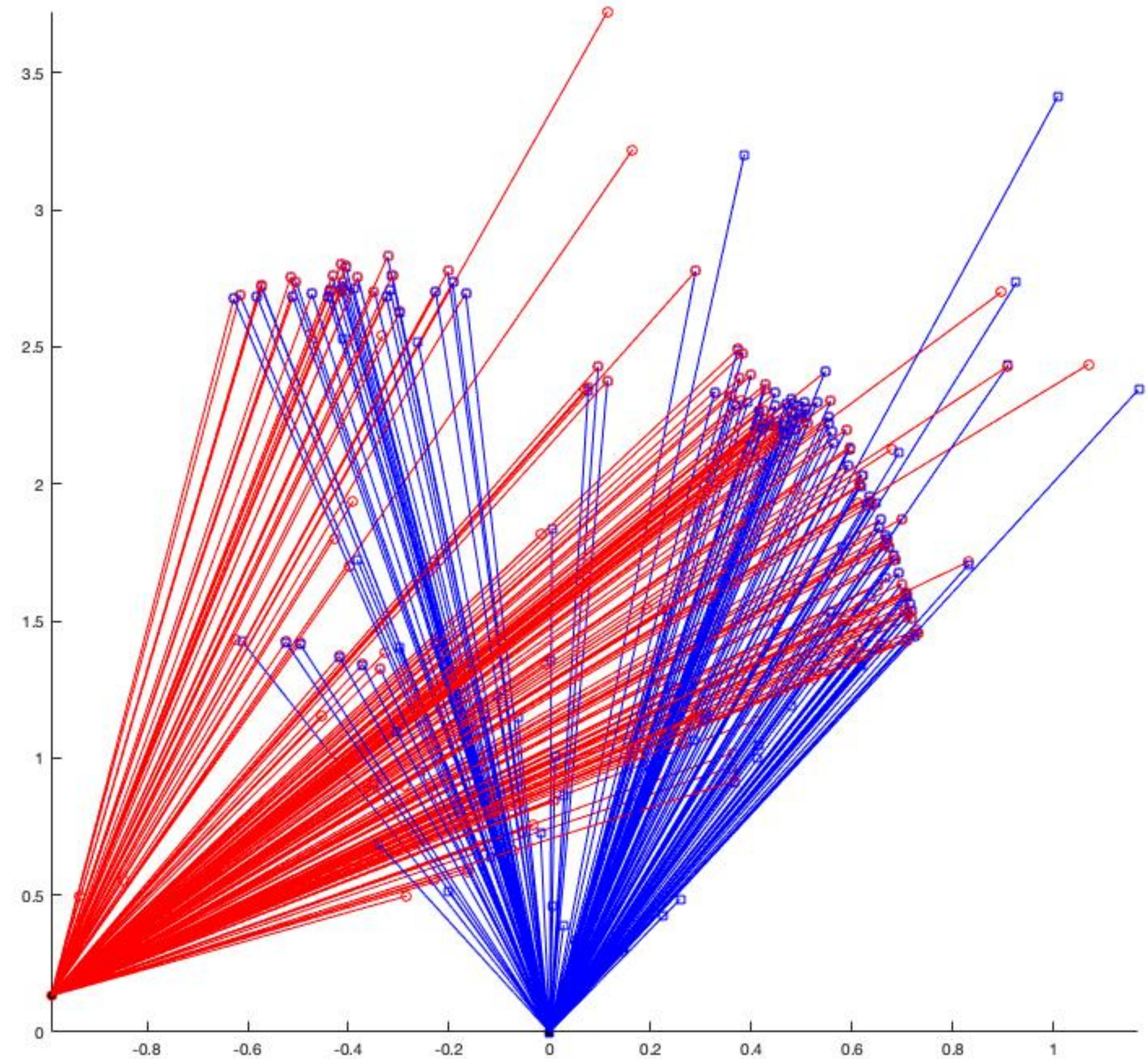
```
F = inv(K)' * E * inv(K);

U11 = [U1, ones(size(U1,1),1)]'; % 3xn
U22 = [U2, ones(size(U2,1),1)]; % nx3

epiLines1 = (U22 * F)';

epiLines2 = (F * U11);
```

2-3. Candidate of rotation and translation

```
z_r = [0, -1, 0; % rotation along z axis for pi/2
    1, 0, 0;
    0, 0, 1];
z_r2 = [0, 1, 0; % rotation along z axis for -pi/2
    -1, 0, 0;
    0, 0, 1];
[U,S,V] = svd(E);
T1_hat = U*z_r*S*U';
R1 = U*z_r*V';
T2_hat = U*z_r2*S*U';
R2 = U*z_r2*V';
T1 = [-T1_hat(2,3),T1_hat(1,3),-T1_hat(1,2)]'; % extract translation from hat operator
T2 = [-T2_hat(2,3),T2_hat(1,3),-T2_hat(1,2)]';

transfoCandidates(1).T = T1;
transfoCandidates(1).R = R1;
transfoCandidates(2).T = T2;
transfoCandidates(2).R = R2;
transfoCandidates(3).T = T1;
transfoCandidates(3).R = R2;
transfoCandidates(4).T = T2;
transfoCandidates(4).R = R1;
```

2-4. reconstruct3D

```
lambdas{i}(:,pt) = pinv([X22(:,pt), - R*X11(:,pt)])*(T/norm(T));
```



2-6. showReprojection

```
p2 = K * (R' * P2' - R'*T); %project point in camera 2 to camera 1
P2proj = p2';

p1 = K * (R * P1' + T); %project point in camera 1 to camera 2
P1proj = p1';
```

Discussion:
The estimation of essential matrix by SVD was quite reliable due to computation of whole datasets. But the results were not realistic, the reason might be insufficient features and datasets.

The usage of Ransac produced satisfying results. But the performance was not reliable due to dependence on random choosing sample sets. If the iteration was high enough, the performance would be nearly perfect.