CIS 580, Machine Perception, Spring 2019

Homework 4

Juncheng Li

March 25, 2018

CIS 580 HW4

1.1 $P_w = RP_c + T$

$$T = \begin{bmatrix} b\sin\alpha \\ -b\cos\alpha \\ h \end{bmatrix}$$

$$Rot(z,\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{inv} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$Rot(x,\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}$$

$$R = Rot(z,\alpha) \cdot R_{inv} \cdot Rot(x,\beta) = \begin{bmatrix} -\sin\alpha & \cos\alpha\sin\beta & \cos\alpha\cos\beta \\ \cos\alpha & \sin\alpha\sin\beta & \sin\alpha\cos\beta \\ 0 & \cos\beta & -\sin\beta \end{bmatrix}$$

1.2

(a) Parral to the image plane

$$R = Rot(z,\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix}, \quad \hat{T} = \begin{bmatrix} 0 & 0 & y_0 \\ 0 & 0 & -x_0 \\ -y_0 & x_0 & 0 \end{bmatrix}$$

$$E = \hat{T} \cdot R = \begin{bmatrix} 0 & 0 & y_0 \\ 0 & 0 & -x_0 \\ x_0\sin\alpha - y_0\cos\alpha & x_0\cos\alpha + y_0\sin\alpha & 0 \end{bmatrix}$$

Satify $E = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ 0 & d & 0 \end{bmatrix}$

(b)

$$E = \begin{bmatrix} 0 & 0 & y_0 \\ 0 & 0 & -x_0 \\ x_0\sin\alpha - y_0\cos\alpha & x_0\cos\alpha + y_0\sin\alpha & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & 0 \end{bmatrix}$$

$x_0 = -b, \ y_0 = a.$

$\sin\alpha = \frac{-bc}{a^2+b^2}, \ \cos\alpha = \frac{-ac}{a^2+b^2}.$

$$R = \begin{bmatrix} \frac{-ac}{a^2+b^2} & \frac{bc}{a^2+b^2} & 0 \\ \frac{-bc}{a^2+b^2} & \frac{-ac}{a^2+b^2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} -b \\ a \\ 0 \end{bmatrix}$$

1.3

$$R = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}$$

$$Ra = \begin{bmatrix} R_1 a \\ R_2 a \\ R_3 a \end{bmatrix}$$

$$R_b = \begin{bmatrix} R_1 b \\ R_2 b \\ R_3 b \end{bmatrix}$$

$$Ra \times R_b = \begin{bmatrix} (R_2 \cdot a)(R_3 \cdot b) - (R_3 \cdot a)(R_2 \cdot b) \\ (R_3 \cdot a)(R_1 \cdot b) - (R_1 \cdot a)(R_3 \cdot b) \\ (R_1 \cdot a)(R_2 \cdot b) - (R_2 \cdot a)(R_1 \cdot b) \end{bmatrix}$$

$$(A \cdot C)(B \cdot D) - (A \cdot D)(B \cdot C) = (A \times B)(C \times D)$$

$$Ra \times R_b = \begin{bmatrix} (R_2 \cdot a)(R_3 \cdot b) - (R_3 \cdot a)(R_2 \cdot b) \\ (R_3 \cdot a)(R_1 \cdot b) - (R_1 \cdot a)(R_3 \cdot b) \\ (R_1 \cdot a)(R_2 \cdot b) - (R_2 \cdot a)(R_1 \cdot b) \end{bmatrix} = \begin{bmatrix} (R_2 \times R_3)(a \times b) \\ (R_3 \times R_1)(a \times b) \\ (R_1 \times R_2)(a \times b) \end{bmatrix}$$

$$(R_1 \times R_2) = R_3, \quad (R_2 \times R_3) = R_1, \quad (R_3 \times R_1) = R_2$$

$$\begin{bmatrix} (R_2 \times R_3)(a \times b) \\ (R_3 \times R_1)(a \times b) \\ (R_1 \times R_2)(a \times b) \end{bmatrix} = \begin{bmatrix} R_1 (a \times b) \\ R_2 (a \times b) \\ R_3 (a \times b) \end{bmatrix} = R(a \times b)$$

.4 line l: $ax+by+c=0$

$$l=\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$x'=(x_1, y_1)$$

$$d=\|\vec{xx'}\|\cos\theta$$

$$d=\frac{\|\vec{xx'}\|\|\vec{n}\|\cos\theta}{\|\vec{n}\|}$$

$$d=\frac{\vec{xx'}\cdot\vec{n}}{\|\vec{n}\|}$$

$$\vec{xx'}\cdot\vec{n}=(x_0-x_1,\ y_0-y_1)\cdot(a,b)=a(x_0-x_1)+b(y_0-y_1)$$

$$\|\vec{n}\|=\sqrt{a^2+b^2}$$

$$d=\frac{|a(x_0-x_1)+b(y_0-y_1)|}{\sqrt{a^2+b^2}}=\frac{|ax_0+by_0-ax_1-by_1|}{\sqrt{a^2+b^2}}$$

$$c=-ax_1-by_1$$

$$d^2=\frac{(ax_0+by_0+c)^2}{a^2+b^2}$$

$$(\tilde{x}^T l)^2=[x_0\ y_0\ 1]\begin{bmatrix} a \\ b \\ c \end{bmatrix}^2=(ax_0+by_0+c)^2$$

$$\|\hat{e}_3 l\|^2=\text{norm}\left(\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix}\right)^2=\text{norm}\left(\begin{bmatrix} -b \\ a \\ c \end{bmatrix}\right)^2=a^2+b^2$$

$$d^2=\frac{(ax_0+by_0+c)^2}{a^2+b^2}=\frac{(\tilde{x}^T l)^2}{\|\hat{e}_3 l\|^2}$$

2. 3D Reconstruction from two 2D images

2.2.1 Estimation of the essential matrix

In the reconstruct_script.m file, U1 and U2 is matching from two images by using SIFT. Then we compute the calibrated coordinates. By using:

$$x_1 = \frac{U1 - u_0}{f}, x_2 = \frac{U2 - u_0}{f}$$

The constrain can be formulated:

$$P1^T * E * P2 = 0$$

we want to formulate and solve a linear system Ae = 0 where A is some matrix containing the point positions and e contains the parameters of E, to estimate. Then we use the 8-point algorithm. Aes=0, es is the null space of A, hence, es=v9, Then project the E onto the space of essential matrices by using svd and returning U diag(1,1,0)V'.

E

3x3 double

	1	2	3
1	0.3881	-0.2843	0.8040
2	0.3845	0.9097	0.0767
3	-0.2030	-0.0268	-0.3145
4			

Figure 1 Essential matrix without Ransac

2.2.2 RANSAC estimation

- Pick a random set of 8 pairs
- estimate E using them and compute the individual residuals for all the other pairs (x1, x2): d(x2, epi(x1))2 +d(x1, epi(x2))2
- Count how many residuals are lower than ε = 10−5(consensus set), and if this count is the largest so far, store the current estimate of E as the best estimate so far, Iterate as many times as needed according to the probability of failure , I use 2000 iteration instead of 500 to get a more steady result.

2.2.3 Drawing the Epipolar lines

- Calculate the fundamental matrix, F = inv(K')*E*inv(K);
- epiLines1 = (U2*F)';
- epiLines2 = (F*U1');

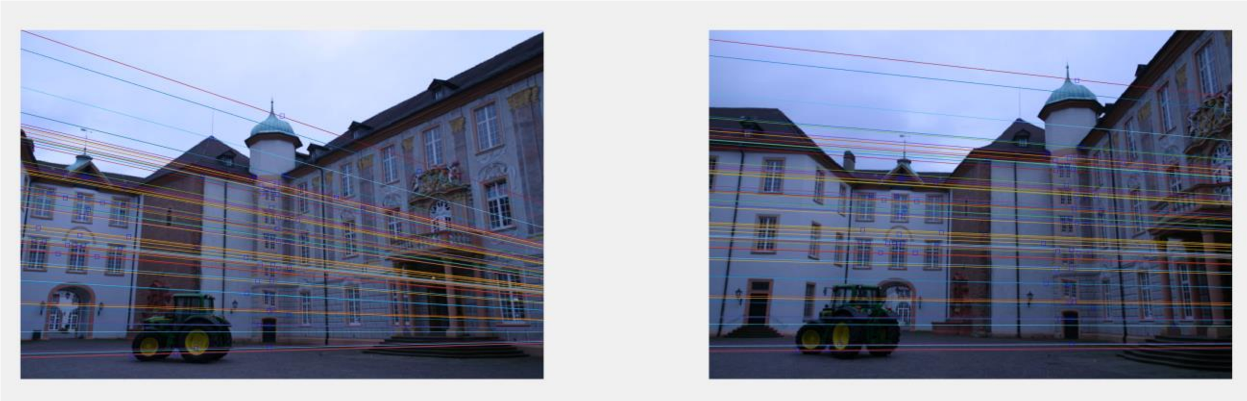

Figure 2 Epipolar lines with ransac

2.3 Pose recovery and 3D reconstruction

There are four 2 camera pose configurations given an essential matrix. Let E=UDV'. W=$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$,

The four configuration are enumerated below:

1. t1 = U(:; 3) and R1 = UWVT
2. t2 = -U(:; 3) and R2 = UWVT
3. t3 = U(:; 3) and R3 = UW'VT
4. t4 = -U(:; 3) and R4 = UW'VT.

If the determinant of the rotation matrix is negative 1. The camera pose must be corrected with t=-t, R=-R.

$$\mathrm{U} = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix}$$

$$r13 = r21r32 - r22r31$$

$$r23 = r11r32 - r12r31$$

$$r33 = r11r22 - r12r21$$

$$\hat{T} = \begin{bmatrix} 0 & r12r21 - r11r22 & r12r31 - r11r32 \\ r11r22 - r12r21 & 0 & r22r31 - r21r32 \\ r11r32 - r12r31 & r21r32 - r22r31 & 0 \end{bmatrix}$$

$\mathrm{T} = \begin{bmatrix} r31 \\ r32 \\ r33 \end{bmatrix}$, which is the last column of U

When we have determined a decomposition E = SR we need to compute a translation vector t from S such that [t]× = S. For such a t we have St = [t]×t = t × t = 0. Therefore the vector t is in the null space of S. The null space of the two matrices S1 and S2 are the same and we can find it by looking in the third column of U.

$$\mathrm{R} = \mathrm{U} * \mathrm{Rz}\left(\frac{\pi}{2}\right)^{T} * V^{T}$$

$$\mathrm{q}^{T}(-T \times Rp) = 0$$

$$\mathrm{T} \times RT(\pi)\mathrm{Rp} = (1 - 2\mathrm{T}^{T}T)(T \times Rp) = 0$$

So we can get the conclusion.

2.4 Triangulation

we can reconstruct the 3D point by computing the intersection of the two rays coming from camera 1 and 2: λ2x2 = λ1Rx1 + T, In practice, the equality doesn't hold (the two rays don't intersect perfectly) and you need to do a least square estimation of λ 1, λ2.

$$\begin{bmatrix} X2 & -R * X1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = T$$

T=T/norm(T)

Now we solve the pseudo-inverse by using:

```
lambdas{i}(:,pt) = pinv([[X2(pt,:),1]',-R*[X1(pt,:),1]'])*(T/norm(T));
```

2.5

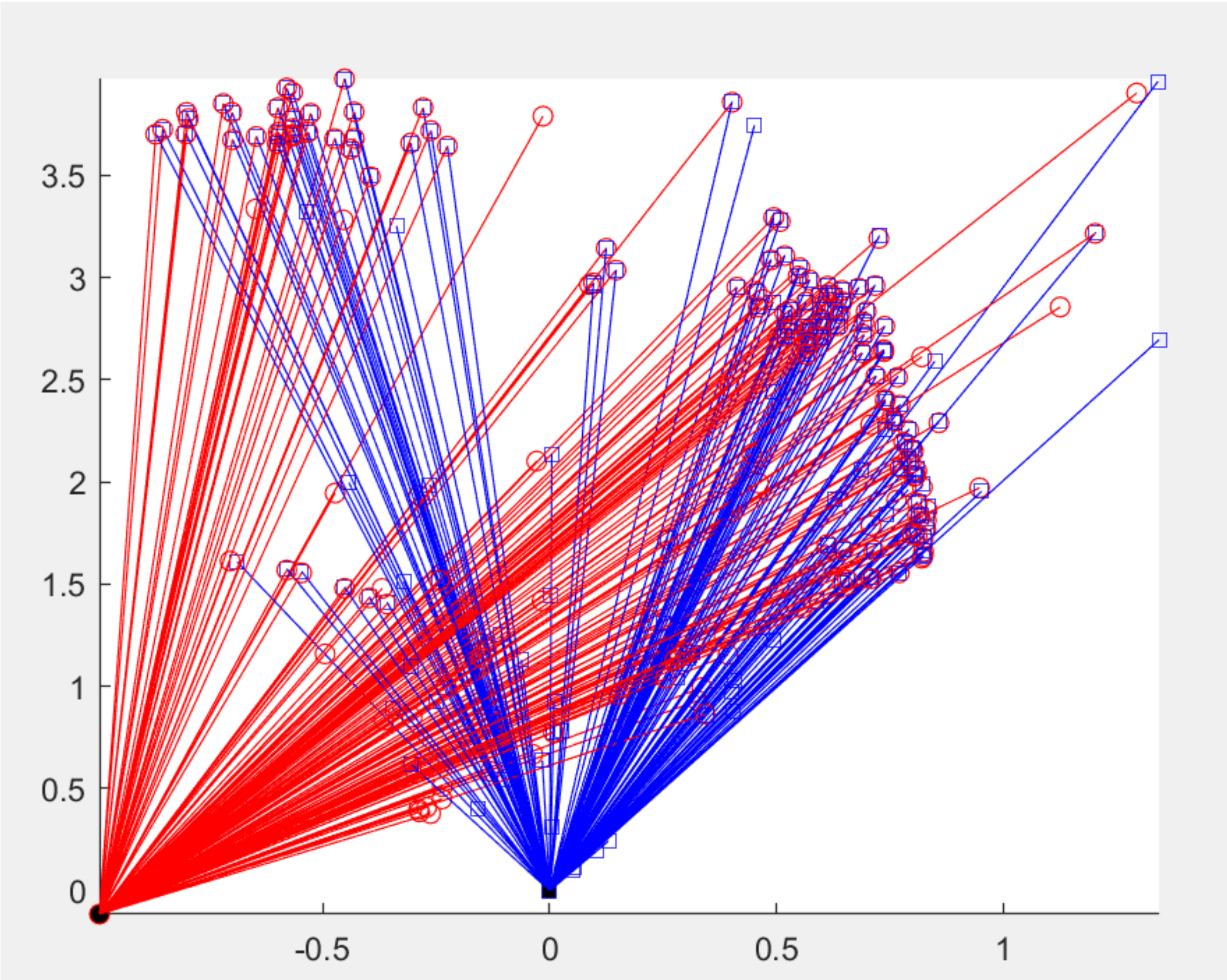

Figure 3 reconstruct

2.6 show Reprojections

P2proj =(K*((R)'*(P2'-T)))';

P1proj =(K*(R*P1'+T))';

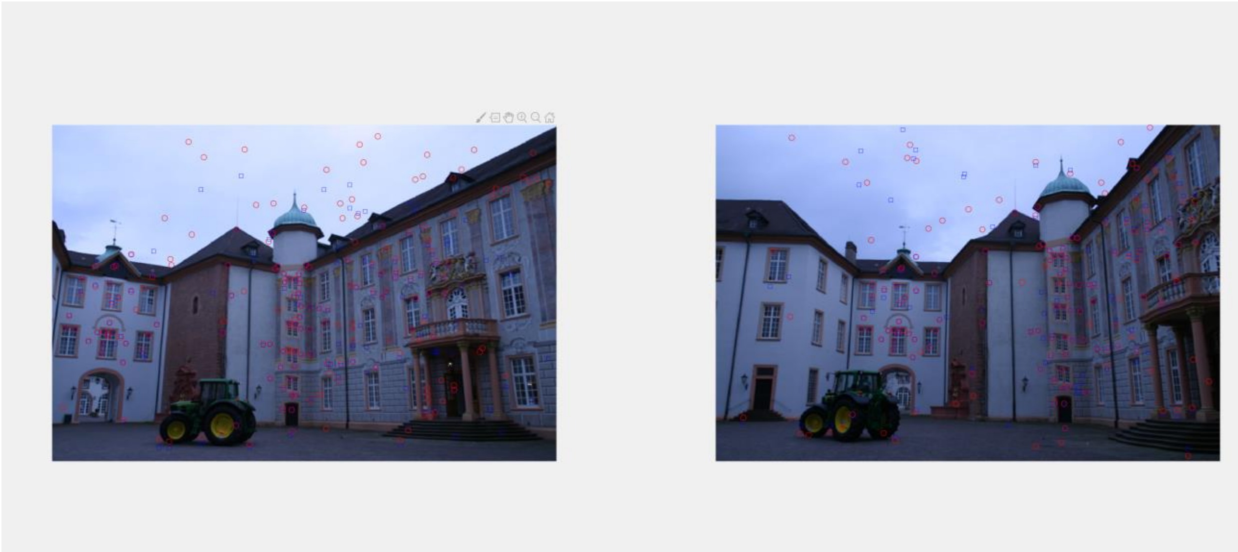

Figure 4 reprojection