EECS 442 Homework #2

Yi Yang

09/29/2016

Problem 1

(a)

Suppose there exists matrix H_1, H_2 and $H = H_1 H_2$ such that $\hat{M} = MH$, $\hat{M}' = M'H$. If we can solve for H_1 and H_2 respectively, then we can prove the statement is correct. Consider:

$$H_1 = \begin{bmatrix} A^{-1} & -A^{-1}b \\ 0 & 1 \end{bmatrix} \Rightarrow \quad MH_1 = \begin{bmatrix} I & 0 \end{bmatrix}, \quad M'H_1 = \begin{bmatrix} A'A^{-1} & -A'A^{-1}b + b' \end{bmatrix}$$

Suppose the general form of H_2 is:

$$H_2 = \begin{bmatrix} W_{3\times3} & X_{3\times1} \\ Y_{1\times3} & Z_{1\times1} \end{bmatrix}$$

Since

$$MH_1H_2 = \begin{bmatrix} I & 0 \end{bmatrix} \quad M'H_1H_2 = \begin{bmatrix} P_{2\times3} & Q_{2\times1} \\ 0 & 1 \end{bmatrix}$$

We can have:

$$W_{3\times3} = I \quad X_{3\times1} = 0$$

Hence, we need to prove the following relations:

$$M'H_1H_2 = \begin{bmatrix} A'A^{-1} & -A'A^{-1}b + b' \end{bmatrix} \begin{bmatrix} I & 0 \\ Y_{1\times3} & Z_{1\times1} \end{bmatrix} = \begin{bmatrix} P_{2\times3} & Q_{2\times1} \\ 0 & 1 \end{bmatrix}$$

If we represent the elements of $M'H_1H_2$ as n_{ij}, then we can deduce the following equation:

$$\begin{bmatrix} n_{3\times1} & n_{3\times2} & n_{3\times3} & n_{3\times4} \end{bmatrix} \begin{bmatrix} I & 0 \\ Y_{1\times3} & Z_{1\times1} \end{bmatrix} = \begin{bmatrix} P_{2\times3} & Q_{2\times1} \\ 0 & 1 \end{bmatrix}$$

Since we have the following relations:

$$e_3^T \left(-A'A^{-1}b + b'\right) \neq 0$$

Hence, we know that $n_{3\times4} \neq 0$, and we can formulate H_2 as:

$$H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{n_{31}}{n_{34}} & -\frac{n_{32}}{n_{34}} & -\frac{n_{33}}{n_{34}} & \frac{1}{n_{34}} \end{bmatrix}$$

According to the derivation above, we know if we set $H = H_1 H_2$, we will transform M and M' to $\hat{M}$ and $\hat{M}'$ respectively.

(b)

It is obvious that

$$MX = (MH)\left(H^{-1}X\right)$$
$$M'X = (M'H)\left(H^{-1}X\right)$$

Hence, if x and x' are matched points with respect to the pair of cameras (M, M'), corresponding to a 3D point, then they are also matched points with respect to the pair of cameras $(MH, M'H)$, corresponding to the point $H^{-1}X$, that is the fundamental matrix corresponding to two pairs are the same.

(c)

According to the conclusions drawn from (b), we know the fundamental matrix corresponding to pair $(M,\ M')$ and $(\hat{M},\ \hat{M}')$ are the same, which is$[b]_x A$, that is:

$$F = [b]_x A = \begin{bmatrix} 0 & -1 & b_2 \\ 1 & 0 & -b_1 \\ -b_2 & b_1 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & -a_{23} \\ a_{11} & a_{12} & a_{13} \\ -b_2 a_{11} + b_1 a_{21} & -b_2 a_{12} + b_1 a_{22} & -b_2 a_{13} + b_1 a_{23} \end{bmatrix}$$

Hence,

$$F = a_{12} \begin{bmatrix} -\frac{a_{21}}{a_{12}} & -\frac{a_{22}}{a_{12}} & -\frac{a_{23}}{a_{12}} \\ \frac{a_{11}}{a_{12}} & 1 & \frac{a_{13}}{a_{12}} \\ -b_2 \frac{a_{11}}{a_{12}} + b_1 \frac{a_{21}}{a_{12}} & -b_2 + b_1 \frac{a_{22}}{a_{12}} & -b_2 \frac{a_{13}}{a_{12}} + b_1 \frac{a_{23}}{a_{12}} \end{bmatrix}$$

From the final expression of F, we know the seven parameters for scaled matrix F is: $-\frac{a_{21}}{a_{12}}$, $-\frac{a_{22}}{a_{12}}$, $-\frac{a_{23}}{a_{12}}$, $\frac{a_{11}}{a_{12}}$, $\frac{a_{13}}{a_{12}}$, b_1 and b_2.

Problem 2

According to the epipolar constraints, we have: $l' = Fx$. Suppose x is the intersection of line l and line k on the right image plane. Then, we have following formula:

$$x = k \times l = [k]_x l$$

Combine two equations we have derived above, we have the following relations, that is:

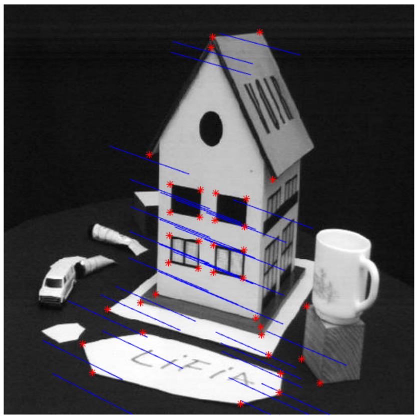
$$l' = F[k]_x l$$

Problem 3

3.1 Fundamental Matrix

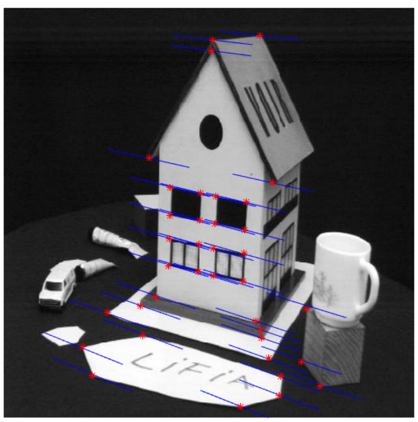
First, we apply 8 point logrithm without normalization
For data set1, we have:

$$dis_{img1} = 28.026, \quad dis_{img2} = 25.163$$

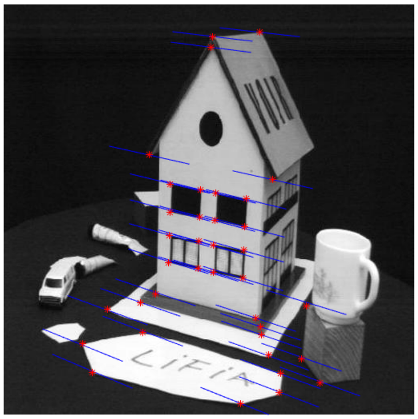
For data set2, we have:

$$dis_{img1} = 9.7014, \quad dis_{img2} = 14.568$$

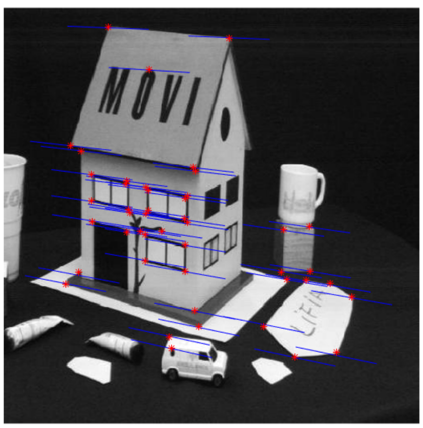
Secondly, we apply 8 point logrithm with normalization:
For data set1, we have:

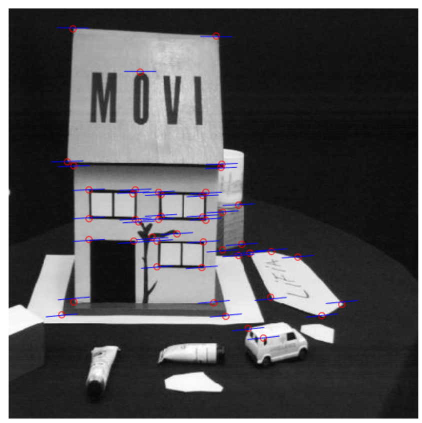
$$dis_{img1} = 0.8844, \quad dis_{img2} = 0.82422$$



For data set2, we have:

$$dis_{img1} = 0.8914, \quad dis_{img2} = 0.89353$$





3.2 Stereo Rectification

Stereo rectification, the results are shown below:
For set1,

$$H = \begin{bmatrix} 2.1065 \times 10^{-6} & -1.647 \times 10^{-6} & -3.354 \times 10^{-6} \\ 1.2335 \times 10^{-6} & -5.9127 \times 10^{-6} & -1.717 \times 10^{-3} \\ -6.2663 \times 10^{-9} & 2.013 \times 10^{-1} & 1.0212 \times 10^{-5} \end{bmatrix}$$

$$H' = \begin{bmatrix} -9.9489 \times 10^{-1} & -1.01 \times 10^{-1} & -2.8055 \times 10^{2} \\ 1.01 \times 10^{-1} & -9.9489 \times 10^{-1} & -2.884 \times 10^{2} \\ 9.3144 \times 10^{-4} & 9.4559 \times 10^{-5} & 1.2627 \end{bmatrix}$$

$$err = 74.279$$

For set2,

$$H = \begin{bmatrix} -7.3225 \times 10^{-7} & 2.7156 \times 10^{-6} & 2.2629 \times 10^{-3} \\ -5.4272 \times 10^{-7} & 5.2278 \times 10^{-6} & 1.4334 \times 10^{-3} \\ 2.2287 \times 10^{-9} & -9.5275 \times 10^{-11} & -7.5068 \times 10^{-6} \end{bmatrix}$$

$$H' = \begin{bmatrix} -9.9279 \times 10^{-1} & -1.1988 \times 10^{-1} & -2.8484 \times 10^{2} \\ 1.1988 \times 10^{-1} & -9.9279 \times 10^{-1} & -2.2347 \times 10^{-2} \\ 5.8878 \times 10^{-4} & 7.1092 \times 10^{-5} & 1.1689 \end{bmatrix}$$

$$err = 58.188$$

Matlab codes are attached:

```
function [F,dist1,dist2] = FMatrix(pathdata1,pathdata2,pathimg1,pathimg2)
% EECS 442 HW2 Q3a
% fundenmantal matrix without normalization

% load feature points from two images. column wise data
% format: dimention(3) * N
[X1,X2] = readTextFiles(pathdata1,pathdata2);

% construction of W, every row is a set of corresponding points
W=ones(9,size(X1,2));
W(1,:) = X2(1,:).*X1(1,:);
W(2,:) = X2(1,:).*X1(2,:);
W(3,:) = X2(1,:);
W(4,:) = X2(2,:).*X1(1,:);
W(5,:) = X2(2,:).*X1(2,:);
W(6,:) = X2(2,:);
W(7,:) = X1(1,:);
W(8,:) = X1(2,:);
% calculate SVD of A
W=W';
[U, D, V] = svd (W);
f = V(:,end); % f is the last column of V
F = reshape(f,3,3);
F = F';
[U, D, V] = svd(F);
D(:,3:end) = 0; % only use the first two columns
F = U * D * V';

% calculate distances
% epipolar lines for X1
F = F';
l1 = F * X2; % 3-3 * 3-n
% epipolar lines for X2
l2 = F' * X1;

% distance between X1 and l1
temp = abs(l1(1,:).*X1(1,:)+l1(2,:).*X1(2,:)+l1(3,:));
dist1 = temp./(l1(1,:).^2+l1(2,:).^2).^.5;
dist1 = mean(dist1(:));
% distance between X2 and l2
temp = abs(l2(1,:).*X2(1,:)+l2(2,:).*X2(2,:)+l2(3,:));
dist2 = temp./(l2(1,:).^2+l2(2,:).^2).^.5;
dist2 = mean(dist2(:));

% draw feature points and epipolar lines
figure;
img1 = imread(pathimg1);
imshow(img1);
hold on;
for i=1:size(X1,2)
    x = X1(1,i)-50:X1(1,i)+50;
    y = -(l1(1,i).*x+l1(3,i))./l1(2,i);
    plot(x,y,'b');
end
plot(X1(1,:),X1(2,:),'r*');
hold off;
```

```

figure;
img2 = imread(pathimg2);
imshow(img2);
hold on;
plot(X2(1,:),X2(2,:),'ro');
for i=1:size(X2,2)
    x = X2(1,i)-20:X2(1,i)+20;
    y = -(l2(1,i).*x+l2(3,i))./l2(2,i);
    plot(x,y,'b');
end
hold off;
```

```
function [F,dist1,dist2] = FMatrix_normalization(pathdata1,pathdata2,pathimg1,pathimg2)
% EECS 442 HW2 Q3a
% fundenmantal matrix with normalization

% load feature points from two images. column wise data
% format: dimention(3) * N
[X1,X2] = readTextFiles(pathdata1,pathdata2);
% normalization
% centroid
X1Trans = [-mean(X1(1,:)); -mean(X1(2,:)); 1];
X2Trans = [-mean(X2(1,:)); -mean(X2(2,:)); 1];
% Isotropic scaling
X1Scale = sqrt(2)./std(X1,1,2);
X2Scale = sqrt(2)./std(X2,1,2);
% transform matrix
transM1 = eye(3,3);
transM1(:,3) = X1Trans;
transM2 = eye(3,3);
transM2(:,3) = X2Trans;

scaleM1 = eye(3,3);
scaleM1(1,1) = X1Scale(1);
scaleM1(2,2) = X1Scale(2);

scaleM2 = eye(3,3);
scaleM2(1,1) = X2Scale(1);
scaleM2(2,2) = X2Scale(2);


normM1 = scaleM1 * transM1;
normM2 = scaleM2 * transM2;

normX1 = normM1 * X1;
normX2 = normM2 * X2;

% construction of W, every row is a set of corresponding points
W=ones(9,size(X1,2));
W(1,:) = normX2(1,:).*normX1(1,:);
W(2,:) = normX2(1,:).*normX1(2,:);
W(3,:) = normX2(1,:);
W(4,:) = normX2(2,:).*normX1(1,:);
W(5,:) = normX2(2,:).*normX1(2,:);
W(6,:) = normX2(2,:);
W(7,:) = normX1(1,:);
W(8,:) = normX1(2,:);
% calculate SVD of A
W=W';
[U, D, V] = svd (W);
f = V(:,end); % f is the last column of V
F = reshape(f,3,3);
```

```
F = F';
[U, D, V] = svd(F);
D(:,3:end) = 0; % only use the first two columns
F = U * D * V';

% denormalization
F  = normM2' * F * normM1;
F = F';
% calculate distances
% epipolar lines for X1
l1 = F * X2; % 3-3 * 3-n
% epipolar lines for X2
l2 = F' * X1;

% distance between X1 and l1
temp = abs(l1(1,:).*X1(1,:)+l1(2,:).*X1(2,:)+l1(3,:));
dist1 = temp./(l1(1,:).^2+l1(2,:).^2).^.5;
dist1 = mean(dist1(:));
% distance between X2 and l2
temp = abs(l2(1,:).*X2(1,:)+l2(2,:).*X2(2,:)+l2(3,:));
dist2 = temp./(l2(1,:).^2+l2(2,:).^2).^.5;
dist2 = mean(dist2(:));

% draw feature points and epipolar lines
figure;
img1 = imread(pathimg1);
imshow(img1);
hold on;
for i=1:size(X1,2)
    x = X1(1,i)-50:X1(1,i)+50;
    y = -(l1(1,i).*x+l1(3,i))./l1(2,i);
    plot(x,y,'b');
end
plot(X1(1,:),X1(2,:),'r*');
hold off;

figure;
img2 = imread(pathimg2);
imshow(img2);
hold on;
plot(X2(1,:),X2(2,:),'ro');
for i=1:size(X2,2)
    x = X2(1,i)-20:X2(1,i)+20;
    y = -(l2(1,i).*x+l2(3,i))./l2(2,i);
    plot(x,y,'b');
end
hold off;
```

```
function [HL,HR,err] = imageRect(pathdata1,pathdata2,pathimg1,pathimg2)
% get fundamental matrix
F = FMatrix_normalization(pathdata1,pathdata2,pathimg1,pathimg2);
img1 = imread(pathimg1);
img2 = imread(pathimg2);

% calculate epipoles
[U, D, V] = svd (F);
epL = V(:,end); % last column of V
epL = epL/epL(3);
[U, D, V] = svd(F');
epR = V(:,end);
epR = epR/epR(3);

% map epi_prime to [1,0,1]
```

```
% x0 is center of the image
pc = [size(img2,2)/2 ; size(img2,1)/2];
TH=[
    1, 0, pc(1);
    0, 1, pc(2);
    0, 0, 1
    ];
epR_prime = TH * epR;
len = sqrt(epR_prime(1)^2+epR_prime(2)^2);
sine = epR_prime(2) / len;
cose = epR_prime(1) / len;
RH=[
    cose, -sine, 0;
    sine, cose, 0;
    0, 0, 1
    ];
% map e_prime to the x-axis at location [length,0,1]
H1=RH * TH;
% send epipole to infinity
H2 = [
    1 0 0;
    0 1 0;
    -1/len 0 1
    ];
% transformation for one image
HR = H2*H1;

% find H to align epipolar lines
% calculate M
% [e'_]xM = F
exM = pinv([0, -epR(3), epR(2); epR(3),0,-epR(1);-epR(2),epR(1),0]);
M = exM * F;
H0 = HR * M;
% find HA =  I + HR * epR * a'
[X1,X2] =  readTextFiles(pathdata1,pathdata2);
X2_hat = HR * X2;
X1_hat = H0 * X1;
X1_hat = X1_hat./repmat(X1_hat(3,:),3,1);
X2_hat = X2_hat./repmat(X2_hat(3,:),3,1);
A = X1_hat';
b = X2_hat(1,:)';
% use the method solving least square parameter
lsp = pinv(A) * b;
HA = [lsp(1),lsp(2),lsp(3);0 1 0;0 0 1];
% get H
HL = HA * H0;

% apply transform and find errors
X1H = HL*X1;
X1H = X1H./repmat(X1H(3,:),3,1);
X2H = HR*X2;
X2H = X2H./repmat(X2H(3,:),3,1);
err = sum((X1H(1:2,:)-X2H(1:2,:)).^2);
err = mean(sqrt(err));
```