Homework 3

Yun-Jou Lin

ID : 0027875836

lin599@purdue.edu

I. Logic of the problems

The report will use three methods to generate the image without projective and affine distortion. The concept of the methods is shown in Figure 0. We are going to apply two different 2-step methods and remove the affine distortion and projective distortion separately. For the 2-step method, one is use the at least 4 corresponding points in both image and world to find the Hp (homography for remove the projective distortion). The other is to use 2 parallel line pairs to find the vanishing line for determining the Hp and remove the projective distortion. After removing the projective distortion, the second step is to find the Ha for removing the affine distortion. In this step, we need at least 2 orthogonal line pairs to identify the Ha. The other method is single step method to remove both projective and affine distortion by determining the dual degenerate conic to identify find the homography Hp+a. In order to identify the dual degenerate conic, we need at least 5 orthoganal line pairs to resolve 5 parameters.

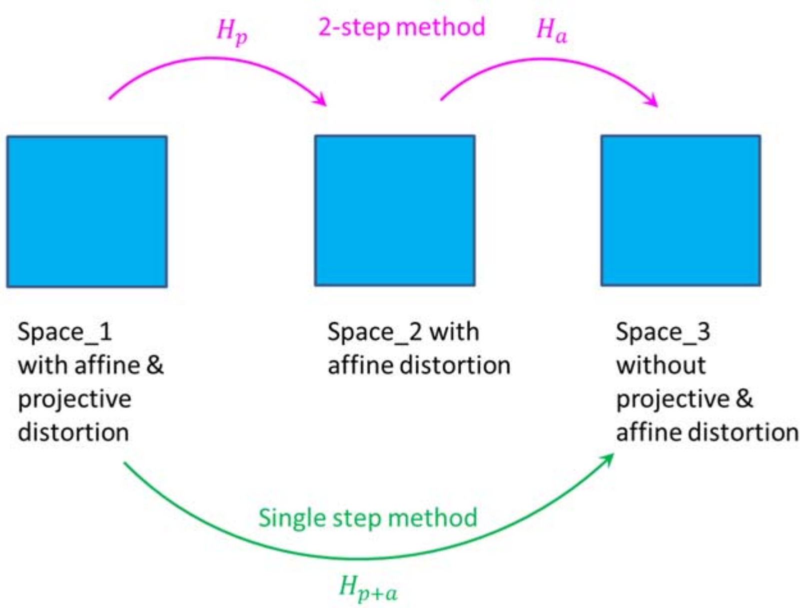

Figure 0. Concept of 2 step and single step homography

II. Steps and Equations

A.1 2-steps method – Remove projective distortion

(1) Use point-point to derive the homography (Hp) between image and world

1. The homography H between image and world without projective distortion can be written as following equations. Since H is a ratio, so we can put the h_{33} as 1. Then, we only have 8 parameters to be solved and we can use at least 4 corresponding points to find the parameters.

$$x' = H_p x \quad (1)$$

$$H_p = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & 1 \end{bmatrix} \quad (2)$$

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3)$$

The world coordinate can be written as physical coordinate in the following equations. Then, we can use at least 4 corresponding points to derive the parameters $(a_1 \quad a_2 \quad a_3 \quad b_1 \quad b_2 \quad b_3 \quad c_1 \quad c_2)$ through least squares method and determine the homography Hp.

$$X = \frac{x'_1}{x'_3} = \frac{a_1 x + b_1 y + c_1}{a_3 x + b_3 y + 1} \qquad (4)$$

$$Y = \frac{x'_2}{x'_3} = \frac{a_2 x + b_2 y + c_2}{a_3 x + b_3 y + 1} \qquad (5)$$

In my code, my Hp is image to world. After, deriving the Hp, we multiply the Hp to bounding box of image for finding the corresponding area in world. Since the corresponding area would be too large or too small, therefore, I use a scale to control the size and find the min coordinate fit to the new image (0,0) for generating the new image without projective. Then the area in the world are go through to find the color in the image with inv(Hp).

(2) Use 2 parallel line pairs to derive the homography (Hp) between image and world.

The homography, Hp, for removing the projective distortion can be estimated from the parameters of the vanishing line. So we only need to derive the vanishing line to establish the Hp. First, two pair of line that are physically parallel are taken. A line parameter can be derive from two points (p &q) (Equ. 6). Then, a vanishing points can be derived from 2 physically parallel lines (l &m) (Equ. 7). Two vanishing points will be used to find a vanishing line (Equ 8). Hp can be represented by the vanishing line in Equ 9.

$$l = p \times q \quad (6)$$

$$p_{van} = l \times m \quad (7)$$

$$l_{van} = p_{van} \times q_{van} \quad (8)$$

$$H_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_{van1} & l_{van2} & l_{van3} \end{bmatrix} \quad (9)$$

A.2 2-steps method – Remove affine distortion

We are going to use two physically orthogonal line pair to remove the affine distortion in this step. The angle between two lines (l & m) in world can be represent as Eq. (10) which can be represented by dual degenerate conic in Eq. (11). When l and m are orthogonal and the Eq. 11 can be written as Eq 12 by transform dual conics as $C^{*\prime} = HC^{*}H^{T}$. Let S = AA^T and Eq 12 can be written as Eq 13.

$$\cos\theta = \frac{l_1 m_1 + l_2 m_2}{\sqrt{(l_1^2 + l_2^2)(m_1^2 + m_2^2)}} \quad (10)$$

$$\cos\theta = \frac{l^T C_\infty^* m}{\sqrt{(l^T C_\infty^* m)(m^T C_\infty^* l)}} \quad (11)$$

$$l'^T H C_\infty^* H^T m' = l'^T \begin{bmatrix} A & \vec{t} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} I & \vec{0} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} A^T & \vec{0} \\ \vec{t}^T & 1 \end{bmatrix} m' = l'^T \begin{bmatrix} AA^T & \vec{0} \\ 0 & 0 \end{bmatrix} m' = 0 \quad (12)$$

$$\text{,where } H_a = \begin{bmatrix} A & \vec{0} \\ \vec{0} & 1 \end{bmatrix}$$

$$(l_1' \; l_2') \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{pmatrix} m_1' \\ m_2' \end{pmatrix} = 0 \quad (13)$$

For the S, only ratio is matter so we can take s_{22} = 1 and only 3 unknown (s_{11} & s_{12} & s_{21}). Eq 13 can be written as Eq 14. Moreover, S is symmetric so $s_{12} = s_{21}$. Therefore, we only need to derive s_{11} & s_{12} with Eq 15.

$$s_{11} l_1' m_1' + s_{12}(l_1' m_2' + l_2' m_1') + s_{22} l_2' m_2' = 0 \quad (14)$$

$$\begin{bmatrix} l_1' m_1' & l_1' m_2' + l_2' m_1' \end{bmatrix} \begin{bmatrix} s_{11} \\ s_{12} \end{bmatrix} = -l_2' m_2' \quad (15)$$

Since $A = VDV^T$, $S = AA^T = VD^2V^T$, we can use the SVD decomposition to solve the V & D and derive A. After determining A, Ha can be obtained.

In my code, the Ha is from world without projective and affine (pure world) to world with affine. Therefore, we need to multiply inv(Ha) with Hp to find the homography (H) from image with projective and affine distortion to pure world. The bounding box of image is multiply H to identify the area in pure world. Since the corresponding area would be too large or too small, therefore, I use a scale to control the size and find the min coordinate fit to the new image (0,0) for generating the new image without projective. Then the area in the world are go through to find the color in the image with inv(H).

B Single Step Method – Remove both projective and affine distortion

We use dual conic to derive the parameters of homography (Hp+a), Eq. 16, that can remove both affine and projective distortion. With the fact that $C^{*\prime} = HC^{*}H^{T}$, the dual conic can be represented as Eq. 16. Take two physically orthogonal lines which $\cos\theta = 0$, in the image, the two lines can be represented in Eq 17. 6 parameters (a, b, c, d, e, f) are going to be derived. However, since only ratio is matter, we set f = 1 which is written in Eq 18. With at least 5 pairs physically orthogonal line, we can solve the parameters (a, b, c, d, e) to derive the conic.

$$\mathrm{H}_{p+a} = \begin{bmatrix} A & 0 \\ v^{T} & 1 \end{bmatrix} \quad (16)$$

$$C_{\infty}^{*\prime} = \begin{bmatrix} AA^{T} & Av \\ v^{T}A^{T} & v^{T}v \end{bmatrix} = HC_{\infty}^{*}H^{T} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \quad (17)$$

$$l'^{T}HC_{\infty}^{*}H^{T}m' = l'^{T}\begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}m' \quad (18)$$

$$\left(l_1'm_1',\ \frac{l_1'm_2'+l_2'm_1'}{2}, l_2'm_2', \frac{l_1'm_3'+l_3'm_1'}{2}, \frac{l_2'm_3'+l_3'm_2'}{2}\right)\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = -l_3'm_3' \quad (19)$$

After deriving the conic, the AA^{T} can be solved by SVD and to derive A and the v can be derived from $v^{T}A^{T}$. After getting A & v, the homography (Hp+a) can be derived.

Since we directly derive the Hp+a, we only need to multiply that with the bounding box of image to identify the area in pure world and apply a scale to control the size of output image. Finally, from the image in pure world to find the corresponding area in original image to fill the color.

III. Experimental Results

<u>Flatiron Image</u>

A. 2-Step Method –4 corresponding points

Figure 1. Four used Points corresponding to the world plane

Figure 2. Image without projective distortion

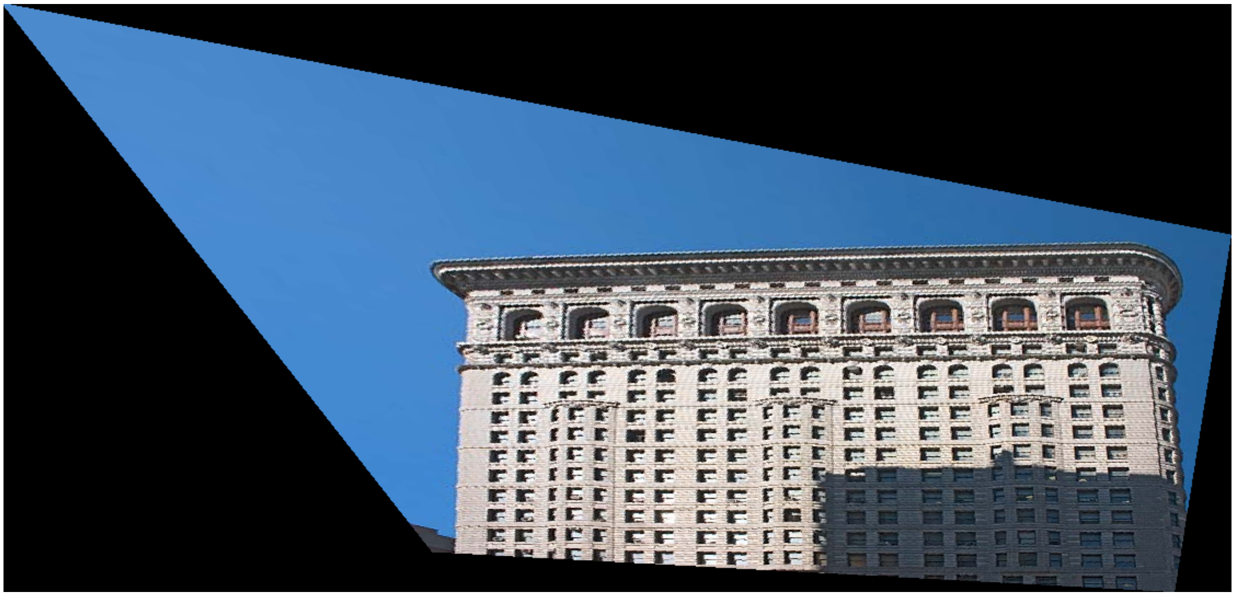

Figure 3. Image without both projective and affine distortion

B. 2-Step Method – 2 pair parallel lines

Figure 4. Four used lines (2 pair parallel lines)

Figure 5. Image without projective distortion

Figure 6. Image without both projective and affine distortion

C. Single Step Method

Figure 7. 6 Lines to form 5 orthogonal pairs (for the rectangle one we can form 4 orthogonal pairs)

Figure 8. Image without both projective and affine distortion

<u>Monalisa Image</u>

A. 2-Step Method – 4 corresponding points

Figure 9. Four used Points corresponding to the world plane

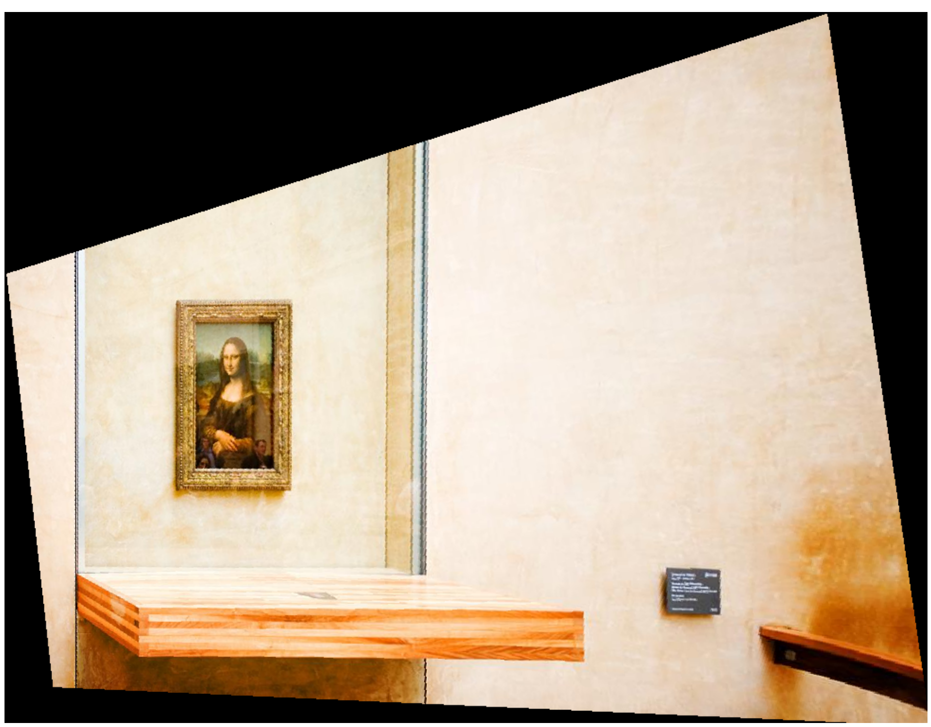

Figure 10. Image without projective distortion

Figure 11. Image without both projective and affine distortion

B. 2-Step Method – 2 pair parallel lines

Figure 12. Four used lines (2 pair parallel lines)

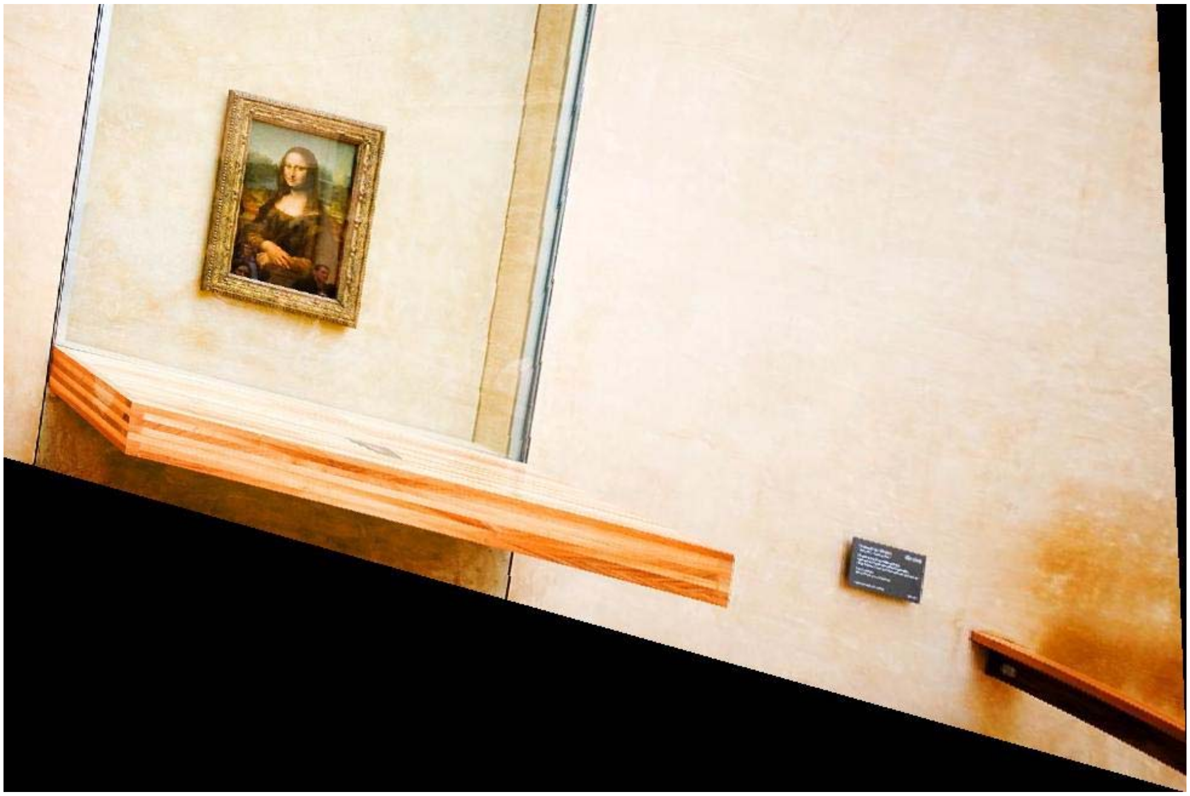

Figure 13. Image without projective distortion

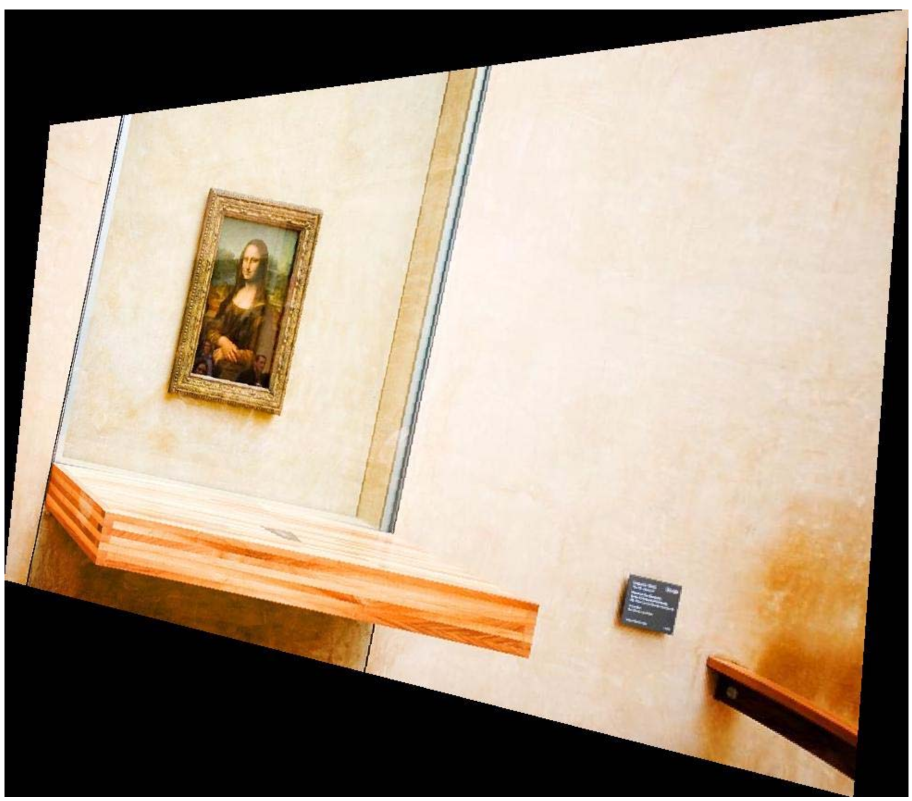

Figure 14. Image without both projective and affine distortion

C. Single Step Method

Figure 15. 6 Lines to form 5 orthogonal pairs (for the rectangle one we can form 4 orthogonal pairs)

Figure 16. Image without both projective and affine distortion

Wideangle Image

A. 2-Step Method – 4 corresponding points

Figure 17. Four used Points corresponding to the world plane

Figure 18. Image without projective distortion

Figure 19. Image without both projective and affine distortion

B. 2-Step Method – 2 pair parallel lines

Figure 20. Four used lines (2 pair parallel lines)

Figure 21. Image without projective distortion

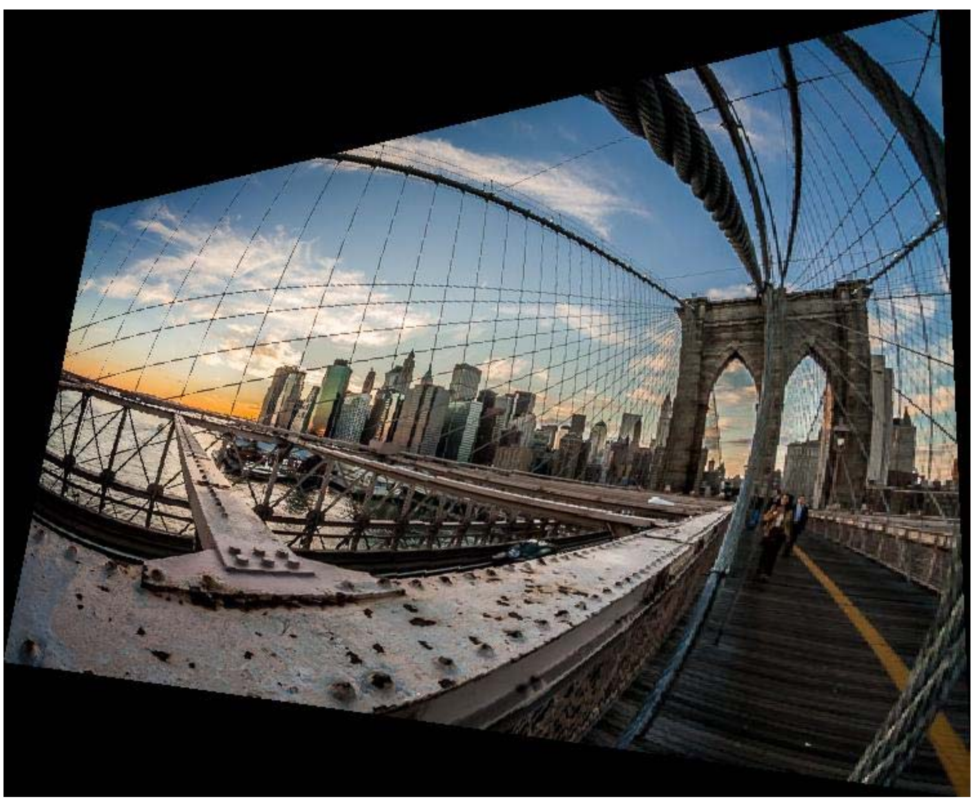

Figure 22. Image without both projective and affine distortion

C. Single Step Method

Figure 23. 6 Lines to form 5 orthogonal pairs (for the rectangle one we can form 4 orthogonal pairs)

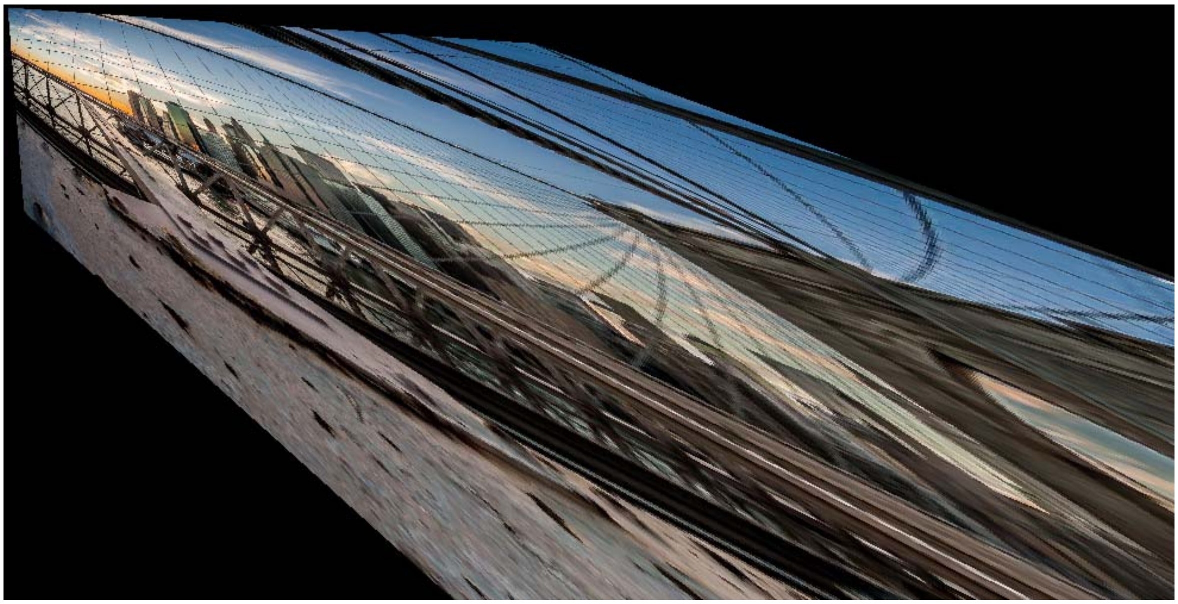

Figure 24. Image without both projective and affine distortion

Mypic1 Image

A. 2-Step Method – 2 pair parallel lines

Figure 25. Four used lines (2 pair parallel lines)

Figure 26. Image without projective distortion

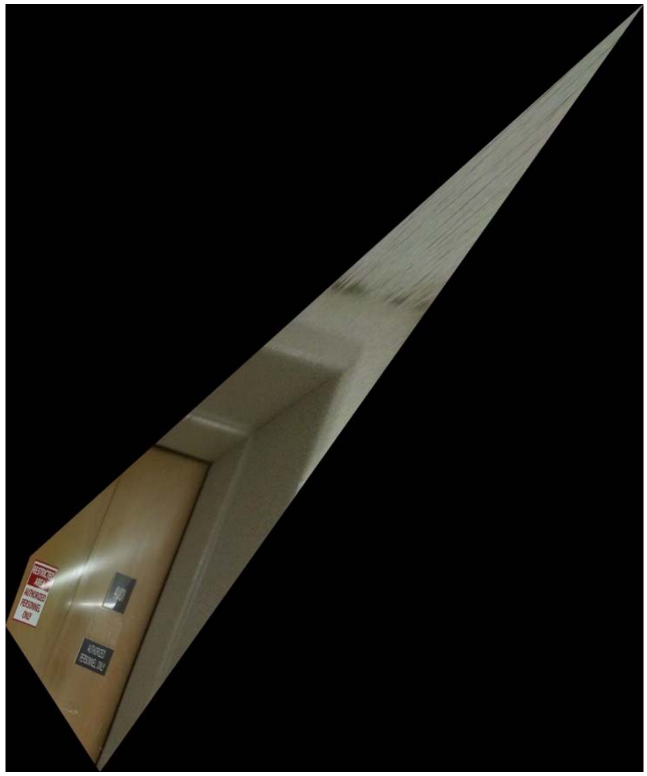

Figure 27. Image without both projective and affine distortion

B. Single Step Method

Figure 28. 6 Lines to form 5 orthogonal pairs (for the rectangle one we can form 4 orthogonal pairs)

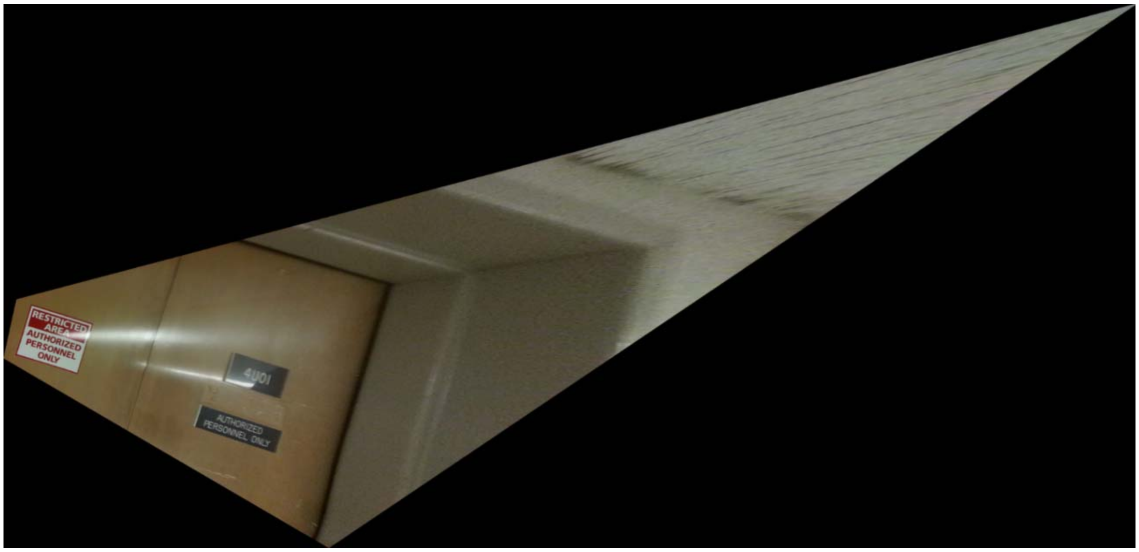

Figure 29. Image without both projective and affine distortion

<u>Mypic2 Image</u>

I. 2-Step Method – 2 pair parallel lines

Figure 30. Four used lines (2 pair parallel lines)

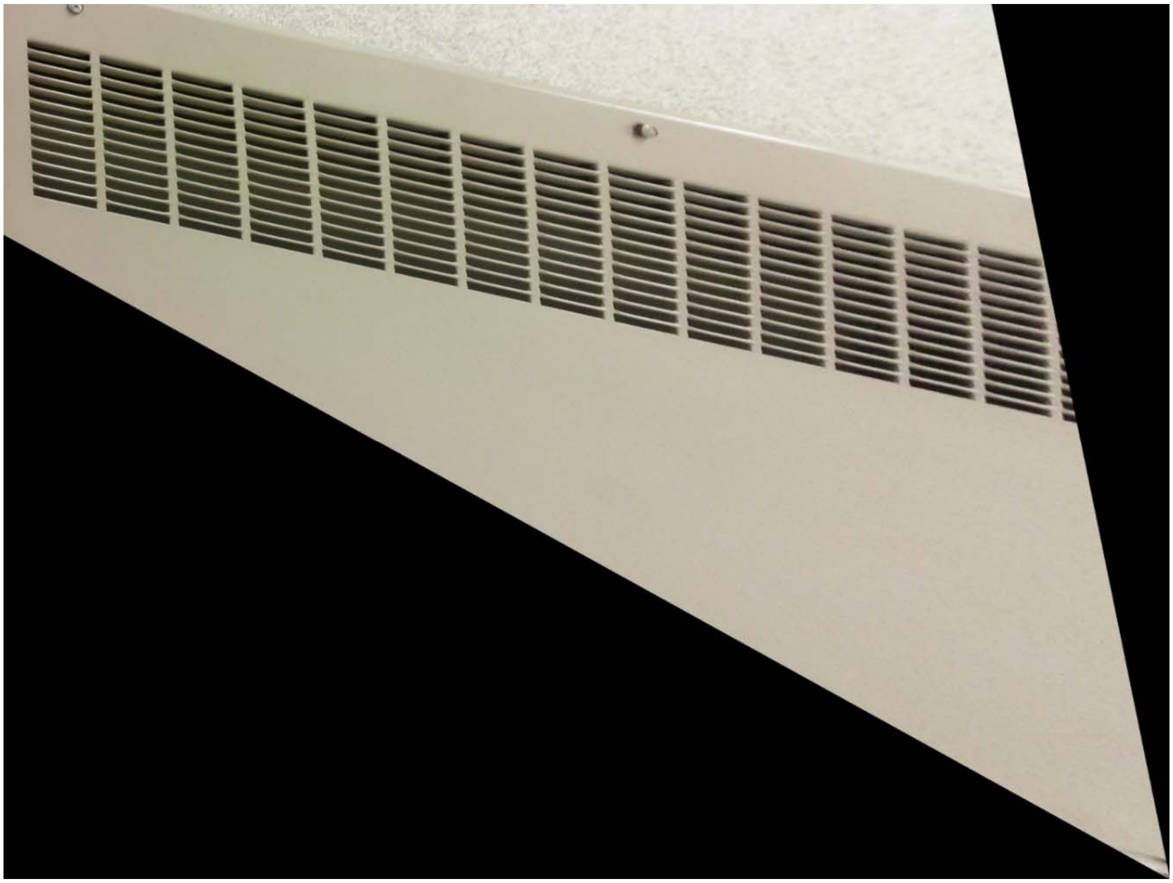

Figure 31. Image without projective distortion

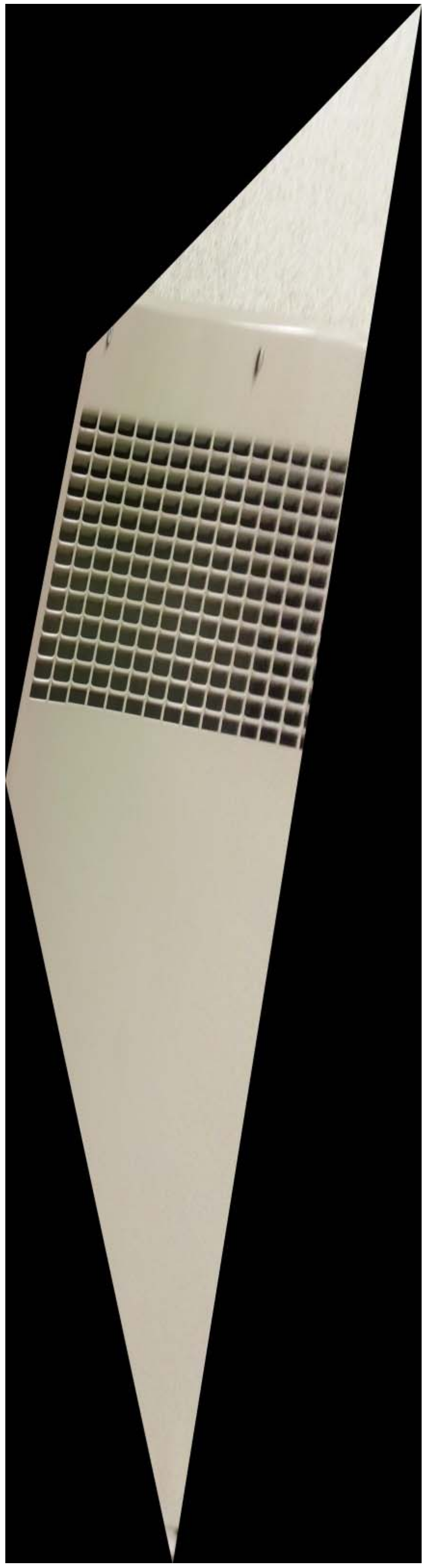

Figure 32. Image without both projective and affine distortion

II. Single Step Method

Figure 33. 6 Lines to form 5 orthogonal pairs (for the rectangle one we can form 4 orthogonal pairs)

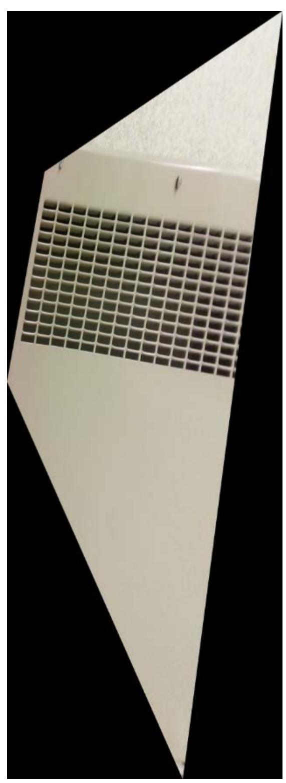

Figure 34. Image without both projective and affine distortion

IV. Code

2-Step Method – points-point

```
%this one is a two steps method
%first, the projective distortion is moved by 4 corresponding points in
%both image & world plane and then we use measured point to derive
%the two pair orthoganoal lines for remove affine distortion

clear all;
% world coordinate of image control points
flatironICP = [1 , 0 , 0;2 , 100, 0; 3, 0 , 150;4, 100, 150];
monalisaICP = [1, 0, 0; 2, 60, 0; 3, 0, 100; 4, 60, 100];
wildangleICP = [1, 0, 0; 2, 70, 0; 3, 0, 120; 4, 70, 120];

% load image and image coordinate
I = imread('flatiron.jpg');
icp0 = load('flatiron_ICP_WorldCoord.dat');
icp1 = flatironICP;

% show the measured coord
figure;
image(I);
hold on;
for i = 1:length(icp0)
    plot(icp0(i,2),icp0(i,3),'r*');
end
axis equal;

% //////////////////////////  First part : remove the projective distortion
from 4 corresponding points in the image &
world//////////////////////////////////////////////////%

%Derive the homography from icp0 to icp1
Un = zeros(8,1);
for i = 1:length(icp0)
    j = i * 2-1;
    B(j,:) =  [icp0(i,2), icp0(i,3), 1, 0, 0, 0, -icp1(i,2)*icp0(i,2), -
icp1(i,2)*icp0(i,3)];
    B(j+1,:) =  [0, 0, 0, icp0(i,2), icp0(i,3), 1, -icp1(i,3)*icp0(i,2), -
icp1(i,3)*icp0(i,3)];
    C(j) = icp1(i,2);
    C(j+1) = icp1(i,3);
end

%Use least square to find the homography
Un = inv(B'*B)*(B'*C');
V = B*Un -C';
%fill the homography
H1 = [Un(1), Un(2), Un(3); Un(4),Un(5),Un(6); Un(7), Un(8), 1];

Hinv = inv(H1);
% find the box of image that correspond to the world plane
Box = [1, 1; size(I,2),1; 1, size(I,1); size(I,2),size(I,1)];
for i = 1:length(Box)
```

```
    projCoord(i,1) =
(H1(1,1)*Box(i,1)+H1(1,2)*Box(i,2)+H1(1,3))/(H1(3,1)*Box(i,1)+H1(3,2)*Box(i,
2)+H1(3,3));
    projCoord(i,2) =
(H1(2,1)*Box(i,1)+H1(2,2)*Box(i,2)+H1(2,3))/(H1(3,1)*Box(i,1)+H1(3,2)*Box(i,
2)+H1(3,3));
end

% derive the scale for fix the size of image
scale1 = size(I,1)/(max(projCoord(:,2))-min(projCoord(:,2)));
scale2 = size(I,2)/(max(projCoord(:,1))-min(projCoord(:,1)));
if(scale1>scale2)
    scale = scale1;
else
    scale = scale2;
end

% find the shifting value for generate the figure
tx = round(min(projCoord(:,1)))-1;
ty = round(min(projCoord(:,2)))-1;

CI = uint8(zeros(round((max(projCoord(:,2))-
min(projCoord(:,2)))*scale),round((max(projCoord(:,1))-
min(projCoord(:,1)))*scale),3));



maxX = size(I,2);
maxY = size(I,1);

%generate the figure without projective distortion
for i = 1 : size(CI,2) %x
    for j = 1 : size(CI,1) %y
        k1 = i/scale + tx;
        k2 = j/scale + ty;
        Cx =
round((Hinv(1,1)*k1+Hinv(1,2)*k2+Hinv(1,3))/(Hinv(3,1)*k1+Hinv(3,2)*k2+Hinv(
3,3)));
        Cy =
round((Hinv(2,1)*k1+Hinv(2,2)*k2+Hinv(2,3))/(Hinv(3,1)*k1+Hinv(3,2)*k2+Hinv(
3,3)));
        if(Cx <= maxX && Cx > 0 && Cy <= maxY && Cy > 0)
            CI(j,i,:) = I(Cy,Cx,:);
        end
    end
end

figure; imshow(CI);
imwrite(CI,'PointtoPoint_1.png');



% /////////////////////////  Second part : remove the affine distortion
from 2 orthoganal lines ////////////////////////////////////////////////%

%   remove the projective distortion for the four measured coordinates
for i = 1:length(icp0)
```

```
    transCoord(i,1) =
(H1(1,1)*icp0(i,2)+H1(1,2)*icp0(i,3)+H1(1,3))/(H1(3,1)*icp0(i,2)+H1(3,2)*icp
0(i,3)+H1(3,3));
    transCoord(i,2) =
(H1(2,1)*icp0(i,2)+H1(2,2)*icp0(i,3)+H1(2,3))/(H1(3,1)*icp0(i,2)+H1(3,2)*icp
0(i,3)+H1(3,3));
end

transCoord(:,3) = 1;

%   derive the line parameters from measured coord
l1 = cross(transCoord(1,:)',transCoord(2,:)');
l2 = cross(transCoord(3,:)',transCoord(4,:)');
l3 = cross(transCoord(1,:)',transCoord(3,:)');
l4 = cross(transCoord(2,:)',transCoord(4,:)');

%   Use least squares to solve the s11 and s12
B = [];
B=[l1(1)*l3(1), (l1(2)*l3(1)+l1(1)*l3(2));
   l2(1)*l4(1), (l2(2)*l4(1)+l2(1)*l4(2));];
C = [-(l1(2)*l3(2)); -(l2(2)*l4(2))];

Un = inv(B'*B)*(B'*C);
V = B*Un - C;

% S == AA' -> use svd to derive eigenvalue and eigenvector
S = [Un(1) Un(2); Un(2),1];

[U,D,VT]=svd(S);
Ds = [sqrt(D(1,1)),0;
       0,sqrt(D(2,2))];

% find the affine
A=VT'*Ds*VT;

H2=[A(1,1) A(1,2) 0;A(2,1) A(2,2) 0;0 0 1];

% find the homography with both projective and affine distortion from image
% to pure world plane
Hinv = inv(H2)*H1;
H = inv(Hinv);


% find the box of image that correspond to the world plane
Box = [1, 1; size(I,2),1; 1, size(I,1); size(I,2),size(I,1)];
for i = 1:length(Box)
    projCoord(i,1) =
(Hinv(1,1)*Box(i,1)+Hinv(1,2)*Box(i,2)+Hinv(1,3))/(Hinv(3,1)*Box(i,1)+Hinv(3
,2)*Box(i,2)+Hinv(3,3));
    projCoord(i,2) =
(Hinv(2,1)*Box(i,1)+Hinv(2,2)*Box(i,2)+Hinv(2,3))/(Hinv(3,1)*Box(i,1)+Hinv(3
,2)*Box(i,2)+Hinv(3,3));
end
```

```
% derive the scale for fix the size of image
scale1 = size(I,1)/(max(projCoord(:,2))-min(projCoord(:,2)));
scale2 = size(I,2)/(max(projCoord(:,1))-min(projCoord(:,1)));
if(scale1>scale2)
    scale = scale1;
else
    scale = scale2;
end

% find the shifting value for generate the figure
tx = round(min(projCoord(:,1)))-1;
ty = round(min(projCoord(:,2)))-1;

CI = uint8(zeros(round((max(projCoord(:,2))-
min(projCoord(:,2)))*scale),round((max(projCoord(:,1))-
min(projCoord(:,1)))*scale),3));

maxX = size(I,2);
maxY = size(I,1);

%generate the figure without affine and projective distortion
for i = 1 : size(CI,2) %x
    for j = 1 : size(CI,1) %y
        k1 = i/scale + tx;
        k2 = j/scale + ty;
        Cx =
round((H(1,1)*k1+H(1,2)*k2+H(1,3))/(H(3,1)*k1+H(3,2)*k2+H(3,3)));
        Cy =
round((H(2,1)*k1+H(2,2)*k2+H(2,3))/(H(3,1)*k1+H(3,2)*k2+H(3,3)));
        if(Cx <= maxX && Cx > 0 && Cy <= maxY && Cy > 0)
            CI(j,i,:) = I(Cy,Cx,:);
        end
    end
end

figure; imshow(CI);
imwrite(CI,'PointtoPoint_2.png');
```

2-Step Method – vanishing line

```
%this one is a two steps method
%first, the projective distortion is moved by two pair of parallel line
%that can generate the vanishing line and then we use measured point to
derive
%the two pair orthoganoal lines for remove affine distortion
clear all;
I = imread('flatiron.jpg');
icp = load('flatiron_ICP_Pair.dat');
icp(:,1) = [];
icp(:,3) = 1;

% show the measured coord
figure;
```

```
image(I);
hold on;
plot(icp(1,1),icp(1,2),'r*',icp(i,2),icp(i,2),'r*' );
plot(icp(3,1),icp(3,2),'r*',icp(i,4),icp(i,4),'r*' );
plot(icp(1,1),icp(1,2),'r*',icp(i,3),icp(i,3),'r*' );
plot(icp(2,1),icp(2,2),'r*',icp(i,4),icp(i,4),'r*' );
axis equal;


% //////////////////////////  First part : remove the projective distortion
from 2 pair of parallel lines////////////////////////////////////////////////
% use mesured 4 points to derive 4 lines
l1 = cross(icp(1,:)',icp(2,:)');
l2 = cross(icp(3,:)',icp(4,:)');
l3 = cross(icp(1,:)',icp(3,:)');
l4 = cross(icp(2,:)',icp(4,:)');

% use parallal lines to find the vanishing points
P = cross(l1,l2);
Q = cross(l3,l4);

P=P/P(3);
Q=Q/Q(3);

%derive vanishing line from 2 vanishing points and find the homography
vl = cross(P,Q);
vl=vl/vl(3);


H1 = [1, 0, 0; 0, 1, 0; vl'];


Hinv = inv(H1);

% find the box of image that correspond to the world plane
Box = [1, 1; size(I,2),1; 1, size(I,1); size(I,2),size(I,1)];
for i = 1:length(Box)
    projCoord(i,1) =
(H1(1,1)*Box(i,1)+H1(1,2)*Box(i,2)+H1(1,3))/(H1(3,1)*Box(i,1)+H1(3,2)*Box(i,
2)+H1(3,3));
    projCoord(i,2) =
(H1(2,1)*Box(i,1)+H1(2,2)*Box(i,2)+H1(2,3))/(H1(3,1)*Box(i,1)+H1(3,2)*Box(i,
2)+H1(3,3));
end

% derive the scale for fix the size of image
scale1 = size(I,1)/(max(projCoord(:,2))-min(projCoord(:,2)));
scale2 = size(I,2)/(max(projCoord(:,1))-min(projCoord(:,1)));
if(scale1>scale2)
    scale = scale1;
else
    scale = scale2;
end
```

```matlab
% find the shifting value for generating the figure
tx = round(min(projCoord(:,1)))-1;
ty = round(min(projCoord(:,2)))-1;

CI = uint8(zeros(round((max(projCoord(:,2))-
min(projCoord(:,2)))*scale),round((max(projCoord(:,1))-
min(projCoord(:,1)))*scale),3));

maxX = size(I,2);
maxY = size(I,1);

%generate the figure without projective distortion
for i = 1 : size(CI,2) %x
    for j = 1 : size(CI,1) %y
        k1 = i/scale + tx;
        k2 = j/scale + ty;
        Cx =
round((Hinv(1,1)*k1+Hinv(1,2)*k2+Hinv(1,3))/(Hinv(3,1)*k1+Hinv(3,2)*k2+Hinv(
3,3)));
        Cy =
round((Hinv(2,1)*k1+Hinv(2,2)*k2+Hinv(2,3))/(Hinv(3,1)*k1+Hinv(3,2)*k2+Hinv(
3,3)));
        if(Cx <= maxX && Cx > 0 && Cy <= maxY && Cy > 0)
            CI(j,i,:) = I(Cy,Cx,:);
        end
    end
end

figure; imshow(CI);
imwrite(CI,'VanishingLine_1.jpg');

% //////////////////////////  Second part : remove the affine distortion
from 2 orthoganal lines ///////////////////////////////////////////////%

%   remove the projective distortion for the four measured coordinates
for i = 1:length(icp)
    transCoord(i,1) =
(H1(1,1)*icp(i,1)+H1(1,2)*icp(i,2)+H1(1,3))/(H1(3,1)*icp(i,1)+H1(3,2)*icp(i,
2)+H1(3,3));
    transCoord(i,2) =
(H1(2,1)*icp(i,1)+H1(2,2)*icp(i,2)+H1(2,3))/(H1(3,1)*icp(i,1)+H1(3,2)*icp(i,
2)+H1(3,3));
end

transCoord(:,3) = 1;

%   derive the line parameters from measured coord
l1 = cross(transCoord(1,:)',transCoord(2,:)');
l2 = cross(transCoord(3,:)',transCoord(4,:)');
l3 = cross(transCoord(1,:)',transCoord(3,:)');
l4 = cross(transCoord(2,:)',transCoord(4,:)');

%  solve the s11 and s12
B = [];
B=[l1(1)*l3(1), (l1(2)*l3(1)+l1(1)*l3(2));
```

```
   l2(1)*l4(1), (l2(2)*l4(1)+l2(1)*l4(2));];
C = [-(l1(2)*l3(2)); -(l2(2)*l4(2))];

Un = B\C;
V = B*Un - C;

% S == AA' -> use svd to derive eigenvalue and eigenvector
S = [Un(1) Un(2); Un(2),1];

[U,D,V]=svd(S);
Ds = [sqrt(D(1,1)),0;
       0,sqrt(D(2,2))];

% find the affine
A=V*Ds*V';

H2=[A(1,1) A(1,2) 0;A(2,1) A(2,2) 0;0 0 1];

% find the homography with both projective and affine distortion from image
% to pure world plane
Hinv = H2\H1;
H = inv(Hinv);



% find the box of image that correspond to the world plane
Box = [1, 1; size(I,2),1; 1, size(I,1); size(I,2),size(I,1)];
for i = 1:length(Box)
    projCoord(i,1) =
(Hinv(1,1)*Box(i,1)+Hinv(1,2)*Box(i,2)+Hinv(1,3))/(Hinv(3,1)*Box(i,1)+Hinv(3
,2)*Box(i,2)+Hinv(3,3));
    projCoord(i,2) =
(Hinv(2,1)*Box(i,1)+Hinv(2,2)*Box(i,2)+Hinv(2,3))/(Hinv(3,1)*Box(i,1)+Hinv(3
,2)*Box(i,2)+Hinv(3,3));
end

% derive the scale for fix the size of image
scale1 = size(I,1)/(max(projCoord(:,2))-min(projCoord(:,2)));
scale2 = size(I,2)/(max(projCoord(:,1))-min(projCoord(:,1)));
if(scale1>scale2)
    scale = scale1;
else
    scale = scale2;
end

% find the shifting value for generate the figure
tx = round(min(projCoord(:,1)))-1;
ty = round(min(projCoord(:,2)))-1;

CI = uint8(zeros(round((max(projCoord(:,2))-
min(projCoord(:,2)))*scale),round((max(projCoord(:,1))-
min(projCoord(:,1)))*scale),3));



maxX = size(I,2);
```

```
maxY = size(I,1);

%generate the figure without affine and projective distortion
for i = 1 : size(CI,2) %x
    for j = 1 : size(CI,1) %y
        k1 = i/scale + tx;
        k2 = j/scale + ty;
        Cx =
round((H(1,1)*k1+H(1,2)*k2+H(1,3))/(H(3,1)*k1+H(3,2)*k2+H(3,3)));
        Cy =
round((H(2,1)*k1+H(2,2)*k2+H(2,3))/(H(3,1)*k1+H(3,2)*k2+H(3,3)));
        if(Cx <= maxX && Cx > 0 && Cy <= maxY && Cy > 0)
            CI(j,i,:) = I(Cy,Cx,:);
        end
    end
end

figure; imshow(CI);
imwrite(CI,'VanishingLine_2.jpg');
```

Single Step Method

```
% this is a one step method that use five orthogonal line pairs for derive
%the homography that can remove both projective and affine distortion at the
same time

clear all;
I = imread('flatiron.jpg');
icp = load('flatiron_ICP_onestep.txt');
icp(:,1) = [];
icp(:,4)= [];


figure;
image(I);
hold on;
%generate the line parameters from the measured coord & shows measured coord
for i = 1:length(icp)
    plot(icp(i,1),icp(i,2),'r*');
    plot(icp(i,4),icp(i,5),'r*');
    L(i,:)=cross(icp(i,1:3)',icp(i,4:6)');
end
axis equal;

% l and m are the orthogal line pairs
l = L(1:5,:);
m = L(6:10,:);
```

```
% derive the conic parameters (a, b, c, d, e) from least squares method (f
is 1)
B = [l(1,1)*m(1,1), (l(1,1)*m(1,2)+l(1,2)*m(1,1))/2, l(1,2)*m(1,2),
(l(1,1)*m(1,3)+l(1,3)*m(1,1))/2, (l(1,2)*m(1,3)+l(1,3)*m(1,2))/2;
     l(2,1)*m(2,1), (l(2,1)*m(2,2)+l(2,2)*m(2,1))/2, l(2,2)*m(2,2),
(l(2,1)*m(2,3)+l(2,3)*m(2,1))/2, (l(2,2)*m(2,3)+l(2,3)*m(2,2))/2;
     l(3,1)*m(3,1), (l(3,1)*m(3,2)+l(3,2)*m(3,1))/2, l(3,2)*m(3,2),
(l(3,1)*m(3,3)+l(3,3)*m(3,1))/2, (l(3,2)*m(3,3)+l(3,3)*m(3,2))/2;
     l(4,1)*m(4,1), (l(4,1)*m(4,2)+l(4,2)*m(4,1))/2, l(4,2)*m(4,2),
(l(4,1)*m(4,3)+l(4,3)*m(4,1))/2, (l(4,2)*m(4,3)+l(4,3)*m(4,2))/2;
     l(5,1)*m(5,1), (l(5,1)*m(5,2)+l(5,2)*m(5,1))/2, l(5,2)*m(5,2),
(l(5,1)*m(5,3)+l(5,3)*m(5,1))/2, (l(5,2)*m(5,3)+l(5,3)*m(5,2))/2;];
C = [-(l(1,3)*m(1,3));
     -(l(2,3)*m(2,3));
     -(l(3,3)*m(3,3));
     -(l(4,3)*m(4,3));
     -(l(5,3)*m(5,3));];

Un = inv(B'*B)*(B'*C);
V = B*Un - C;

% fill the conic
Con = [Un(1) Un(2)/2 Un(4)/2; Un(2)/2 Un(3) Un(5)/2; Un(4)/2 Un(5)/2 1];
Con = Con/max(max(Con));

% derive the affine from S = AA'
S = Con(1:2,1:2);

[U,D,V]=svd(S);
Ds = [sqrt(D(1,1)),0;
       0,sqrt(D(2,2))];
A=V*Ds*V';

% get v
VT = [Con(3,1), Con(3,2)]/A';

% get the homography that can remove both projective and affine distortion
H=[A(1,1), A(1,2), 0; A(2,1), A(2,2), 0; VT(1), VT(2) 1];

Hinv = inv(H);

% find the box of image that correspond to the world plane
Box = [1, 1; size(I,2),1; 1, size(I,1); size(I,2),size(I,1)];
for i = 1:length(Box)
    projCoord(i,1) =
(Hinv(1,1)*Box(i,1)+Hinv(1,2)*Box(i,2)+Hinv(1,3))/(Hinv(3,1)*Box(i,1)+Hinv(3
,2)*Box(i,2)+Hinv(3,3));
    projCoord(i,2) =
(Hinv(2,1)*Box(i,1)+Hinv(2,2)*Box(i,2)+Hinv(2,3))/(Hinv(3,1)*Box(i,1)+Hinv(3
,2)*Box(i,2)+Hinv(3,3));
end

% derive the scale for fix the size of image
scale1 = size(I,1)/(max(projCoord(:,2))-min(projCoord(:,2)));
```

```
scale2 = size(I,2)/(max(projCoord(:,1))-min(projCoord(:,1)));
if(scale1>scale2)
    scale = scale1;
else
    scale = scale2;
end

% find the shifting value for generating the figure
tx = round(min(projCoord(:,1)))-1;
ty = round(min(projCoord(:,2)))-1;

CI = uint8(zeros(round((max(projCoord(:,2))-
min(projCoord(:,2)))*scale),round((max(projCoord(:,1))-
min(projCoord(:,1)))*scale),3));


maxX = size(I,2);
maxY = size(I,1);

%generate the figure without both affine and projective distortion
for i = 1 : size(CI,2) %x
    for j = 1 : size(CI,1) %y
        k1 = i/scale + tx;
        k2 = j/scale + ty;
        Cx =
round((H(1,1)*k1+H(1,2)*k2+H(1,3))/(H(3,1)*k1+H(3,2)*k2+H(3,3)));
        Cy =
round((H(2,1)*k1+H(2,2)*k2+H(2,3))/(H(3,1)*k1+H(3,2)*k2+H(3,3)));
        if(Cx <= maxX && Cx > 0 && Cy <= maxY && Cy > 0)
            CI(j,i,:) = I(Cy,Cx,:);
        end
    end
end

figure; imshow(CI);
imwrite(CI,'oneStep.png');
```

V. Observation

(a) Comparing two different methods (2-steps and single step), the single step method would be more sensitive to the measured coordinates. Since I have implemented the 2-steps (point-point & vanishing line) and single step method, I found the point-point could be the most robust way to remove the projective distortion which are not much sensitive to the picking points.

(b) For removing the projective distortion, when the lines pair are not parallel in the physical space the result may not be ideal. It means this way is sensitive to the given dataset.

(c) Since both 2-step methods need to use orthogonal lines for solving the affine distortion, when the picking lines are not orthogonal in the physical space, the results would be also not ideal.

Sometimes, even a pixel will affect the results, especially when the size of image is small but it covers large area in the physical world, a pixel could be huge different in the physical space.

(d) For me, the single step method is the most sensitive one. Not only those lines should be really represented the orthogonal lines in physical space but also those line should be consistent. Since we are going to use at least five pair to solve the problem, if one pair of line is not good, it will totally affect the results.

(e) All the methods can derive the image without projective and affine distortion. However, there may exist some scaling problem that are not solved (similarity transformation.) Moreover, when the measured points are not very ideal, the scale would shows on both x and y directions.

(5) The wideangle imagery has other distortion. After removing affine and projective distortion we still can see the distortion there.