Assignment
CAP 6419 - 3D Computer Vision

Affine and Metric Rectification of Image

The aim of this experiment is to demonstrate the idea of removing projective distortion, once the image of infinity points is specified, and the affine distortion can be removed once the image of the circular points is specified.

Procedure:
Affine Rectification of an Image:

1. First find the pairs of parallel lines on the world image.
2. Then find the imaged line (l) at infinity
3. Using this image line form Matrix H, which is [1 0 0, 0 1 0, l1, l2, l3].
4. Use H and input image to get affined rectify image.

Metric Rectification of an Image:

1. Locate pair of orthogonal lines on the affined rectify image.
2. Define $S = KK^T$

$$\begin{pmatrix} l'_1 & l'_2 & l'_3 \end{pmatrix} \begin{bmatrix} \mathrm{KK}^\mathsf{T} & \mathbf{0} \\ \mathbf{0}^\mathsf{T} & 0 \end{bmatrix} \begin{pmatrix} m'_1 \\ m'_2 \\ m'_3 \end{pmatrix} = 0$$

3. Obtain homography matrix.
4. Apply this matrix on image to get metric rectified image.

Code:

% Function for affine rectification

```
function H = affine_rectification(inputImg,outImg)

% to loccate the pair of parallel lines
figure(1);
imshow(inputImg);
title('Select a pair of parallel lines by selecting 4 points:');
xlabel('Press Enter once done');
```

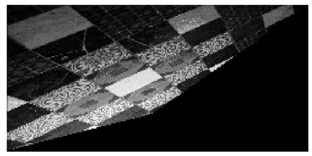
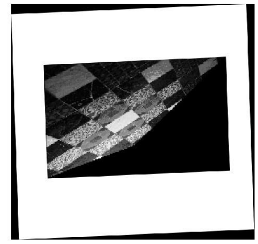
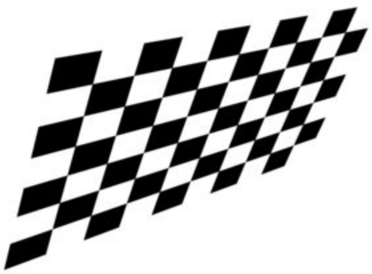
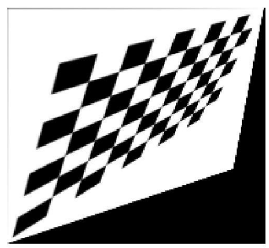
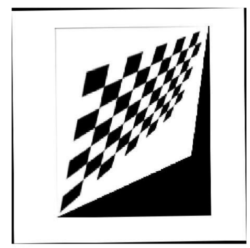
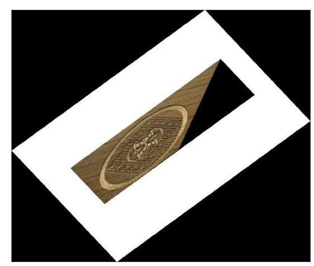
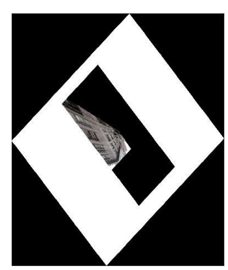
```
[x,y] = getpts;
close Figure 1;

% assigning selected points to variables
p1 = [x(1) y(1) 1];
p2 = [x(2) y(2) 1];
p3 = [x(3) y(3) 1];
p4 = [x(4) y(4) 1];

% 2 pairs of parallel lines obtained from the points by cross products
l1 = cross(p1,p2);
l2 = cross(p3,p4);
l3 = cross(p2, p3);
l4 = cross(p1, p4);


% to get intersection point of line l1 and l2 , then l3 and l4 at infinity
a = cross(l1, l2);
a = a/a(1,3); %The Euclidean coordinate dividing by 3rd value
b = cross(l3, l4);
b = b/ b(1,3); % the Euclidean cordinate
l = cross(a, b); % required imaged line at infinity
disp("the line at infinity is:");
disp (l);

%The H matfix
H = [1 0 0; 0 1 0; l(1, 1)/l(1,3) l(1, 2)/l(1,3) 1];
temp = maketform('projective',H');


Iout = imtransform(inputImg, temp);
% required affine rectified image.
myout = imshow(Iout);
saveas(myout,outImg);
end
```

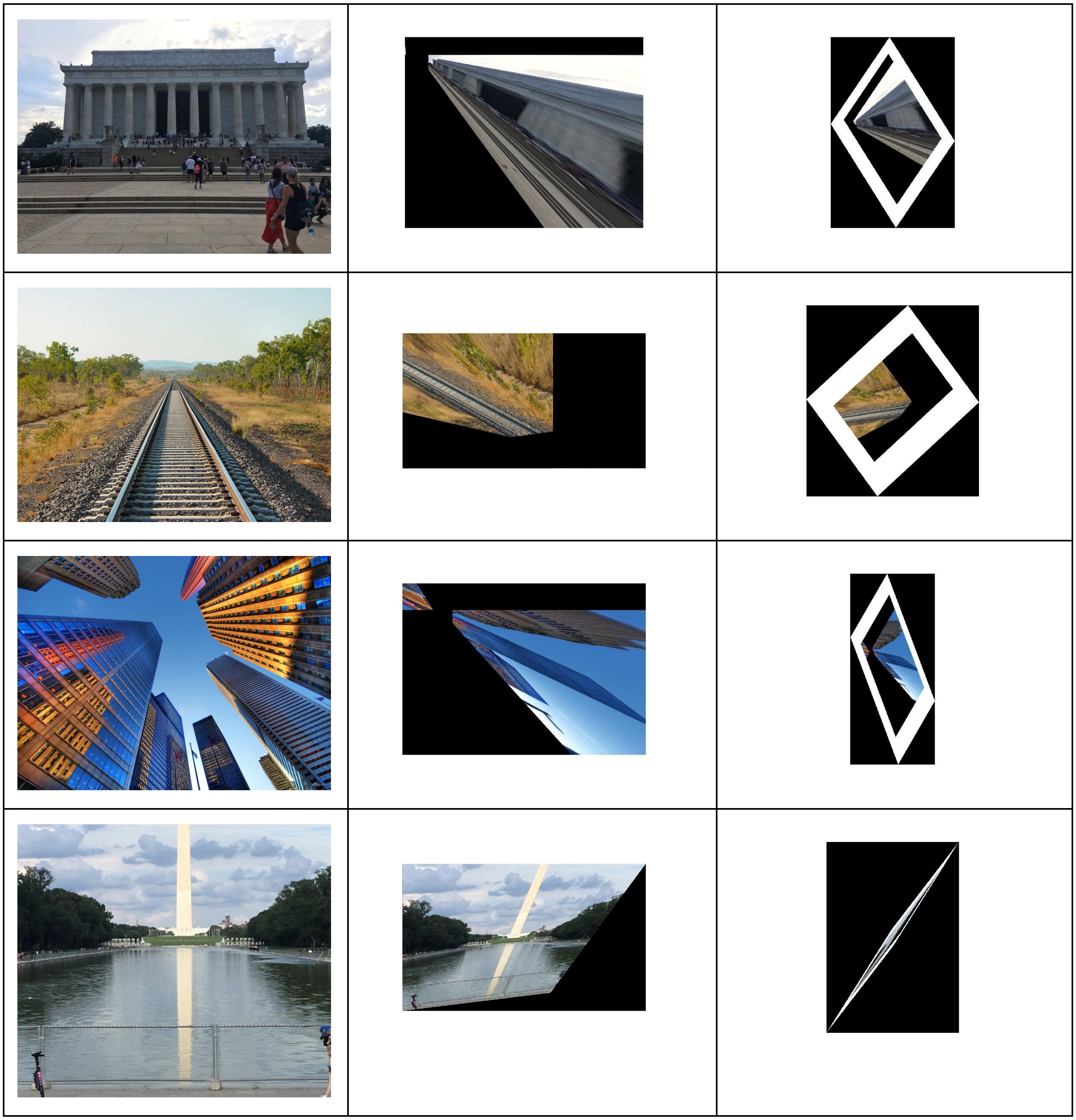
%Function for Metric Rectification on affine rectified image

```
function H = metric_rectification(affineImage,metricImage)

% to locate the pair of orthogonal lines
figure(1);
imshow(affineImage);
title('Select 2 pairs of orthogonal lines from the image');
xlabel('Press Enter once done');

[x,y] = getpts;
close Figure 1;
% assigning variables
l11 = x(1);
l12 = y(1);
l21 = x(3);
l22 = y(3);
m11 = x(2);
m12 = y(2);
m21 = x(4);
m22 = y(4);

% Defining variables to find S
M = [l11*m11 (l11*m12 + l12*m11) ; l21*m21 (l21*m22 + l22*m21)];
b = [-l12*m12;-l22*m22 ];
% solving for the linear constraint on 2 × 2 matrix S
x = linsolve(M,b);
% defining S
S = eye(2);
S(1,1) = x(1);
S(1,2) = x(2);
S(2,1) = x(2);

disp(S);
% obtaining a suitable rectifying homography
[U,D,V] = svd(S);
sqrtD = sqrt(D);
U_T = transpose(U);
A = U_T*sqrtD;
A = A*V;
H2 = eye(3);
```

```
H2(1,1) = A(1,1);
H2(1,2) = A(1,2);
H2(2,1) = A(2,1);
H2(2,2) = A(2,2);
if H2(1,1) < 0
H2(1,1) = -H2(1,1);

elseif H2(2,2) < 0
H2(2,2) = -H2(2,2);
end
H = H2';

temp = maketform('projective',H);
Output_Image = imtransform(affineImage, temp);
saveImg = imshow(Output_Image); % required metric rectified image
saveas(saveImg,metricImage);

end


%Call above function to get result like:


image_filename="building.jpg";
[filepath,name,ext] = fileparts(image_filename);

input = imread(image_filename);
b= name+"_aff.jpg";
disp(b);

affine_rectification(input,b);

affinedInput = imread(b);
c= name+"_met.jpg";
metric_rectification(affinedInput,c);
```

Sample Results:

Original Image	**Affine Rectified Image**	**Metric Rectified Image**
		
		
		
		



Conclusion:

The affined image can be achieved by simply to transform the identified l∞ to its canonical position of l∞ = (0, 0, 1)T. The (projective) matrix which achieves this transformation can be applied to every point in the image in order to affinely rectify the image, i.e. after the transformation, affine measurements can be made directly from the rectified image.

But in this implementation, the result is very much dependent on the points of selection for parallel lines in affine transformation and orthogonal points in Metric transformation. I observed the little change in point coordinates gives dramatically different results in each run for the same image. But for some of the images, I got required rectified images.