Assignment-1

Avinash Vadlamudi
201501164

1)

A) Histogram Matching:

Code:

```
function q1a(image1,image2)
    im1= imread(image1);
    im2 = imread(image2);
    im1 = rgb2gray(im1);
    im2 = rgb2gray(im2);

    v1 = imhist(im1,256);
    v2 = imhist(im2,256);

    mat = tril(ones(256,256),0);
    cum_v1 = mat*v1;
    cum_v2 = mat*v2;

    cum_v1(:,1) =
round(((cum_v1(:,1)*1.0)/cum_v1(256,1))*255);
    cum_v2(:,1) = round((cum_v2(:,1)*1.0/cum_v2(256,1))*255);

    for i= 1:256
        I(i,1) = i-1+find(cum_v2(i:256,1)>=cum_v1(i,1),1,'first')-1;
    end
    im3 = im1;
    im3 = I(im1(:,:)+1);
    figure;
    subplot(1,3,1);
```

```
    imshow(uint8(im1));
    title('actual image');
    subplot(1,3,2);
    imshow(uint8(im2));
    title('reference image');
    subplot(1,3,3);
    imshow(uint8(im3));
    title('final image');
end
```

B) Local Histogram Equalization:

Code:

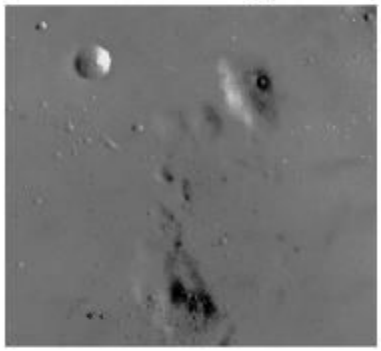
```
function q1b(image)
    im1= (imread(image));
    im3 = zeros(size(im1));
    parts = 3;
    if size(im1,3)==3
        len = size(im1,1);
        wid = size(im1,2);
        for i = 1:round(len/(1*parts)):len
            for j = 1:round(wid/(1*parts)):wid
                for k = 1:3
                    v1 =
imhist(im1(i:min(i+round(len/parts),len),j:min(j+round(wid/parts),
wid),k),256);
                    mat = tril(ones(256,256),0);

                    cum_v1 = mat*v1;
                    cum_v1(:,1) =
round(((cum_v1(:,1)*1.0)/cum_v1(256,1))*255);

%im3(i:min(i+round(len/parts),len),j:min(j+round(wid/parts),wid),
```

```
k) =
(im3(i:min(i+round(len/parts),len),j:min(j+round(wid/parts),wid),k
) +
cum_v1(im1(i:min(i+round(len/parts),len),j:min(j+round(wid/part
s),wid),k)+1))/2;

im3(i:min(i+round(len/parts),len),j:min(j+round(wid/parts),wid),k)
= (
cum_v1(im1(i:min(i+round(len/parts),len),j:min(j+round(wid/part
s),wid),k)+1));
            end
         end
      end
   else
      v1 = imhist(im1,256);
      mat = tril(ones(256,256),0);

      cum_v1 = mat*v1;
      cum_v1(:,1) =
round(((cum_v1(:,1)*1.0)/cum_v1(256,1))*255);
      %im3 = (im3(:,:) + cum_v1(im1(:,:)+1))/2;
      im3 = cum_v1(im1(:,:)+1);
   end
   figure;
   subplot(1,2,1);
   imshow(uint8(im1));
   title('Actual Image');
   subplot(1,2,2);
   imshow(uint8(im3));
   title('Final Image');
end
```

Actual Image

Final Image

Actual Image

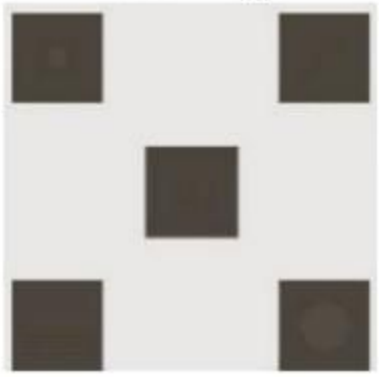


Final Image

Actual Image

Final Image

Actual Image



Final Image

2)

A) <u>Gaussian Filtering:</u>

Code:

```
function q2a(image)
    im1 = double(imread(image));
    filter1 = zeros(3,3);
    filter2 = zeros(5,5);
    filter3 = zeros(8,8);

    s = 100;
    for i=1:3
        for j = 1:3
            filter1(i,j) = exp(-1*((i-2).^2 + (j-2).^2)/(2*s*s));
        end
    end
    filter1 = filter1/sum(filter1(:));

    for i=1:5
        for j = 1:5
            filter2(i,j) = exp(-1*((i-3).^2 + (j-3).^2)/(2*s*s));
        end
    end
    filter2 = filter2/sum(filter2(:));

    for i=1:8
        for j = 1:8
            filter3(i,j) = exp(-1*(min(abs(i-4),abs(i-5)).^2 +
min(abs(j-4),abs(j-5)).^2)/(2*s*s));
        end
    end
    filter3 = filter3/sum(filter3(:));
```

```
    im11 = imfilter(im1,filter1);
    im12 = imfilter(im1,filter2);
    im13 = imfilter(im1,filter3);

    figure;
    subplot(2,2,1);
    imshow(uint8(im1));
    title('Original image');
    subplot(2,2,2);
    imshow(uint8(im11));
    title('Filter of size 3');
    subplot(2,2,3);
    imshow(uint8(im12));
    title('Filter of size 5');
    subplot(2,2,4);
    imshow(uint8(im13));
    title('Filter of size 8');

end
```

For sigma = 2

Original image

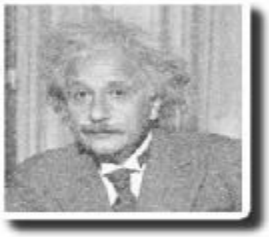
Filter of size 3

Filter of size 5

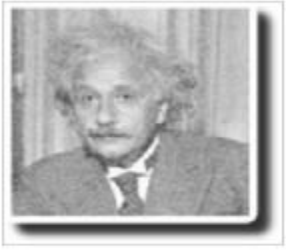
Filter of size 8

Original image

Filter of size 3



Filter of size 5



Filter of size 8

For sigma = 10

Original image

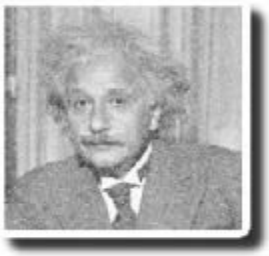
Filter of size 3

Filter of size 5

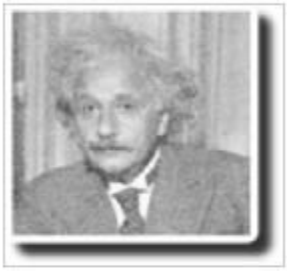
Filter of size 8

Original image

Filter of size 3



Filter of size 5



Filter of size 8

For sigma = 100

Original image

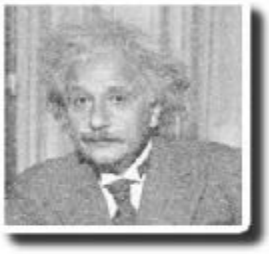
Filter of size 3

Filter of size 5

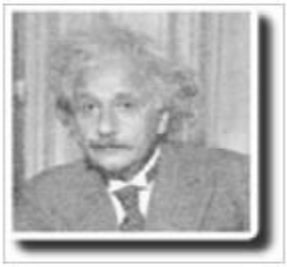
Filter of size 8

Original image

Filter of size 3



Filter of size 5



Filter of size 8

B) Median Filtering:

Code:

```
function q2b(image)
im1 = double(imread(image));

im2 = padarray(im1,[5,5]);
im11 = im1;
im12 = im1;
im13 = im1;

for i = 6:size(im2,1)-5
   for j = 6:size(im2,2)-5
      im11(i-5,j-5,1) = median(median(im2(i-1:i+1,j-1:j+1,1),2));
      im11(i-5,j-5,2) = median(median(im2(i-1:i+1,j-1:j+1,2),2));
      im11(i-5,j-5,3) = median(median(im2(i-1:i+1,j-1:j+1,3),2));

      im12(i-5,j-5,1) = median(median(im2(i-2:i+2,j-2:j+2,1),2));
      im12(i-5,j-5,2) = median(median(im2(i-2:i+2,j-2:j+2,2),2));
      im12(i-5,j-5,3) = median(median(im2(i-2:i+2,j-2:j+2,3),2));

      im13(i-5,j-5,1) = median(median(im2(i-4:i+4,j-4:j+4,1),2));
      im13(i-5,j-5,2) = median(median(im2(i-4:i+4,j-4:j+4,2),2));
      im13(i-5,j-5,3) = median(median(im2(i-4:i+4,j-4:j+4,3),2));

   end
end
figure;
subplot(2,2,1);
imshow(uint8(im1));
title('Original image');
subplot(2,2,2);
imshow(uint8(im11));
```

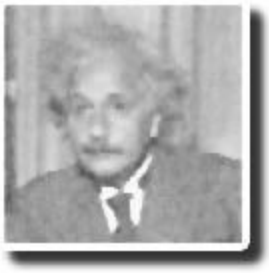
```
title('Filter of size 3');
subplot(2,2,3);
imshow(uint8(im12));
title('Filter of size 5');
subplot(2,2,4);
imshow(uint8(im13));
title('Filter of size 8');
end
```

Original image

Filter of size 3

Filter of size 5

Filter of size 8



Original image

Filter of size 3

Filter of size 5

Filter of size 8

C) High Boost Filtering:

Code:

```
function q2c(image)
im1 = (imread(image));
filter1 = zeros(3,3);
filter2 = zeros(5,5);
filter3 = zeros(8,8);

lambda = 3;
s = 30;

for i=1:3
   for j = 1:3
      filter1(i,j) = exp(-1*((i-2).^2 + (j-2).^2)/(2*s*s));
   end
end
filter1 = filter1/sum(filter1(:));

for i=1:5
   for j = 1:5
      filter2(i,j) = exp(-1*((i-3).^2 + (j-3).^2)/(2*s*s));
   end
end
filter2 = filter2/sum(filter2(:));

for i=1:8
   for j = 1:8
      filter3(i,j) = exp(-1*(min(abs(i-4),abs(i-5)).^2 +
min(abs(j-4),abs(j-5)).^2)/(2*s*s));
   end
end
filter3 = filter3/sum(filter3(:));
```

```
im11 = imfilter(im1,filter1);
im12 = imfilter(im1,filter2);
im13 = imfilter(im1,filter3);

im21 = abs(im1 - im11);
im22 = abs(im1 - im12);
im23 = abs(im1 - im13);

im21 = uint8(im21);
im22 = uint8(im22);
im23 = uint8(im23);


im31 = (im1) + lambda*((im21));
im32 = (im1) + lambda*((im22));
im33 = (im1) + lambda*((im23));

figure;
subplot(3,3,1);
imshow(uint8(im1));
title('Original image');
subplot(3,3,2);
imshow(uint8(lambda*im21));
title('Laplacian image');
subplot(3,3,3);
imshow(uint8(im31));
title('Filter of size 3');

subplot(3,3,4);
imshow(uint8(im1));
title('Original image');
subplot(3,3,5);
```

```
imshow(uint8(lambda*im22));
title('Laplacian image');
subplot(3,3,6);
imshow(uint8(im32));
title('Filter of size 5');

subplot(3,3,7);
imshow(uint8(im1));
title('Original image');
subplot(3,3,8);
imshow(uint8(lambda*im23));
title('Laplacian image');
subplot(3,3,9);
imshow(uint8(im33));
title('Filter of size 8');
end
```

Original image

Laplacian image

Filter of size 3

Original image

Laplacian image

Filter of size 5

Original image

Laplacian image

Filter of size 8

Original image

Laplacian image

Filter of size 3

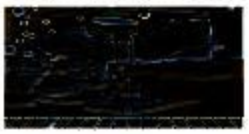
Original image

Laplacian image

Filter of size 5

Original image

Laplacian image



Filter of size 8

D) Bilateral Filtering:

Code:

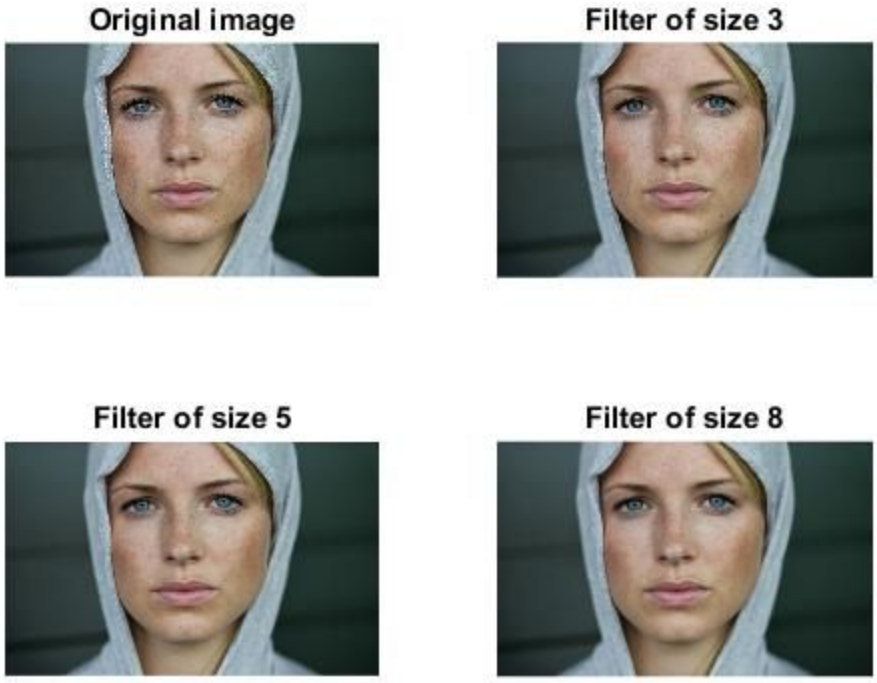
```
sigma1 = 2;
sigma2 = 400;

im1 = double(imread('./Assign1_imgs/portraits2.jpg'));
filter1 = zeros(3,3);
filter2 = zeros(5,5);
filter3 = zeros(8,8);

for i=1:3
   for j = 1:3
      filter1(i,j) = exp(-1*((i-2).^2 +
(j-2).^2)/(2*sigma1*sigma1));
   end
end
%filter1 = filter1/sum(filter1(:));

for i=1:5
   for j = 1:5
      filter2(i,j) = exp(-1*((i-3).^2 +
(j-3).^2)/(2*sigma1*sigma1));
   end
end
%filter2 = filter2/sum(filter2(:));

for i=1:8
   for j = 1:8
      filter3(i,j) = exp(-1*(min(abs(i-4),abs(i-5)).^2 +
min(abs(j-4),abs(j-5)).^2)/(2*sigma1*sigma1));
   end
end
```

```
%filter3 = filter3/sum(filter3(:));

dim = size(im1);
im11 = zeros(dim);
im12 = zeros(dim);
im13 = zeros(dim);

for i = 1:dim(1)
  for j = 1:dim(2)
      I1 =
im1(max(i-1,1):min(i+1,dim(1)),max(j-1,1):min(j+1,dim(2)),
:);
      I2 =
im1(max(i-2,1):min(i+2,dim(1)),max(j-2,1):min(j+2,dim(2)),
:);
      I3 =
im1(max(i-3,1):min(i+4,dim(1)),max(j-3,1):min(j+4,dim(2)),
:);
% size -3
      H =
exp((-1*((I1(:,:,1)-im1(i,j,1)).^2))/(2*sigma2*sigma2));
      F =
H.*filter1((max(i-1,1):min(i+1,dim(1)))-i+1+1,(max(j-1,1):mi
n(j+1,dim(2)))-j+1+1);
      norm_F = sum(F(:));
      im11(i,j,1) = sum(sum(F.*I1(:,:,1)))/norm_F;

      H =
exp(-((I1(:,:,2)-im1(i,j,2)).^2)/(2*sigma2*sigma2));
      F =
H.*filter1((max(i-1,1):min(i+1,dim(1)))-i+1+1,(max(j-1,1):mi
n(j+1,dim(2)))-j+1+1);
      norm_F = sum(F(:));
```

```
        im11(i,j,2) = sum(sum(F.*I1(:,:,2)))/norm_F;

        H =
exp(-((I1(:,:,3)-im1(i,j,3)).^2)/(2*sigma2*sigma2));
        F =
H.*filter1((max(i-1,1):min(i+1,dim(1)))-i+1+1,(max(j-1,1):mi
n(j+1,dim(2)))-j+1+1);
        norm_F = sum(F(:));
        im11(i,j,3) = sum(sum(F.*I1(:,:,3)))/norm_F;

% size -5
        H =
exp(-((I2(:,:,1)-im1(i,j,1)).^2)/(2*sigma2*sigma2));
        F =
H.*filter2((max(i-2,1):min(i+2,dim(1)))-i+2+1,(max(j-2,1):mi
n(j+2,dim(2)))-j+2+1);
        norm_F = sum(F(:));
        im12(i,j,1) = sum(sum(F.*I2(:,:,1)))/norm_F;

        H =
exp(-((I2(:,:,2)-im1(i,j,2)).^2)/(2*sigma2*sigma2));
        F =
H.*filter2((max(i-2,1):min(i+2,dim(1)))-i+2+1,(max(j-2,1):mi
n(j+2,dim(2)))-j+2+1);
        norm_F = sum(F(:));
        im12(i,j,2) = sum(sum(F.*I2(:,:,2)))/norm_F;

        H =
exp(-((I2(:,:,3)-im1(i,j,3)).^2)/(2*sigma2*sigma2));
        F =
H.*filter2((max(i-2,1):min(i+2,dim(1)))-i+2+1,(max(j-2,1):mi
n(j+2,dim(2)))-j+2+1);
        norm_F = sum(F(:));
```

```
        im12(i,j,3) = sum(sum(F.*I2(:,:,3)))/norm_F;

% size -8
        H =
exp(-((I3(:,:,1)-im1(i,j,1)).^2)/(2*sigma2*sigma2));
        F =
H.*filter3((max(i-3,1):min(i+4,dim(1)))-i+3+1,(max(j-3,1):mi
n(j+4,dim(2)))-j+3+1);
        norm_F = sum(F(:));
        im13(i,j,1) = sum(sum(F.*I3(:,:,1)))/norm_F;

        H =
exp(-((I3(:,:,2)-im1(i,j,2)).^2)/(2*sigma2*sigma2));
        F =
H.*filter3((max(i-3,1):min(i+4,dim(1)))-i+3+1,(max(j-3,1):mi
n(j+4,dim(2)))-j+3+1);
        norm_F = sum(F(:));
        im13(i,j,2) = sum(sum(F.*I3(:,:,2)))/norm_F;

        H =
exp(-((I3(:,:,3)-im1(i,j,3)).^2)/(2*sigma2*sigma2));
        F =
H.*filter3((max(i-3,1):min(i+4,dim(1)))-i+3+1,(max(j-3,1):mi
n(j+4,dim(2)))-j+3+1);
        norm_F = sum(F(:));
        im13(i,j,3) = sum(sum(F.*I3(:,:,3)))/norm_F;

   end
end
figure;
subplot(2,2,1);
imshow(uint8(im1));
title('Original image');
```

```
subplot(2,2,2);
imshow(uint8(im11));
title('Filter of size 3');
subplot(2,2,3);
imshow(uint8(im12));
title('Filter of size 5');
subplot(2,2,4);
imshow(uint8(im13));
title('Filter of size 8');
```



Original image

Filter of size 3

Filter of size 5

Filter of size 8

Original image

Filter of size 3

Filter of size 5

Filter of size 8

3)

A) Ripple Transform:

Code:

```
function q3a(image)
im1 = double(imread(image));
X_max = size(im1,1);
Y_max = size(im1,2);
C_max = size(im1,3);
im2 = im1;

ax = 10;
ay = 15;
tx = 120;
ty = 150;
val = 0;
for val = 1:20
for i =1:X_max
   for j = 1:Y_max
      for chan = 1:C_max
         l = i + (ax+val)*(sin((2*pi*j)/(tx+val*3)));
         r = j + (ay+val)*(sin((2*pi*i)/(ty+val*3)));

         im2(i,j,:) =
im1(min(max(floor(l),1),X_max),min(max(floor(r),1),Y_ma
x),:);

      end
   end
end
figure;
imshow(uint8(im2));
end
```

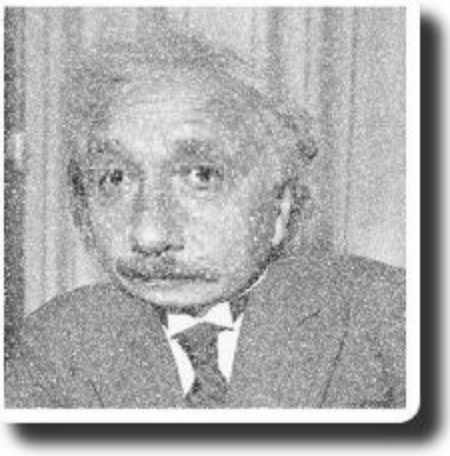
```
%figure;
%subplot(1,2,1);
%imshow(uint8(im1));
%subplot(1,2,2);
%imshow(uint8(im2));
end
```

B) Spherical Transform:

Code:

```
function q3b(image)
im1 = double(imread(image));
X_max = size(im1,1);
Y_max = size(im1,2);
im2 = im1;

Y1_c = round(size(im1,2)/2);
X1_c = round(size(im1,1)/2);
r1_max = min(X1_c,Y1_c)/2;
val = 10;
rho1 = 1.5;
for val = 1:20
for i = 1:X_max
   for j = 1:Y_max
      X_c = X1_c+ 2*val-30;
      Y_c = Y1_c + 2*val-30;
      r_max = r1_max+ 2*val;
      rho = rho1 + 0.01*val;

      rad = sqrt((i-X_c).^2 + (j-Y_c).^2);
      dx = i-X_c;
      dy = j-Y_c;
      if rad<=r_max
         z = sqrt((r_max).^2 - (rad).^2);
         bx = ((rho-1)/rho)*(asin(dx/(sqrt(dx*dx + z*z))));
         by = ((rho-1)/rho)*(asin(dy/(sqrt(dy*dy + z*z))));
```

```
            l = i-z*tan(bx);
            r = j-z*tan(by);
        else
            l = i;
            r = j;
        end
        im2(i,j,:) =
im1(min(max(floor(l),1),X_max),min(max(floor(r),1),Y_ma
x),:);
    end
end
figure;
imshow(uint8(im2));
end
figure;
subplot(1,2,1);
imshow(uint8(im1));
subplot(1,2,2);
imshow(uint8(im2));
end
```



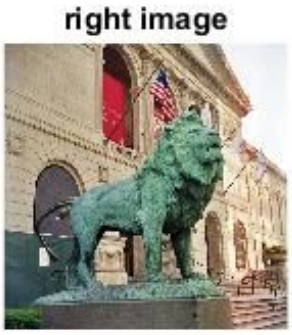
4) Homography Estimation:

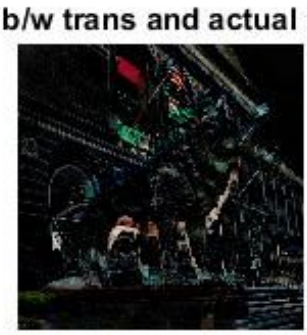
Code:

```
imshow('./Assign1_imgs/stereo_left.jpg');
p1=ginput(4);
t = ones(4,1);
p2 = [p1,t];

imshow('./Assign1_imgs/stereo_right.jpg');
p3=ginput(4);
t = ones(4,1);
p4 = [p3,t];

p41 = transpose(p4);
p21 = transpose(p2);
```

```
I = [1,0,0;0,1,0;0,0,1];

mat = p41*(p21\I);

im1 = imread('./Assign1_imgs/stereo_left.jpg');
im2 = imread('./Assign1_imgs/stereo_right.jpg');
im22 = im1;
for i = 1:size(im1,1)
    for j = 1:size(im1,1)
        temp = [i;j;1];
        val = mat*temp;
        val(1,1) = floor(val(1,1));
        val(2,1) = floor(val(2,1));
        if(val(1,1)<=0 || val(1,1)>size(im2,1)|| val(2,1)<=0 ||
val(2,1)>size(im2,2))
            continue;
        else
            im22(i,j,:) = im2(val(1,1),val(2,1),:);
        end
    end
end
im21 = imresize(im2,[size(im22,1) size(im22,2)]);
im_diff = abs(im22-im21);
figure;
subplot(2,2,1);
imshow(uint8(im1));
title('left image');
subplot(2,2,2);
imshow(uint8(im2));
title('right image');
subplot(2,2,3);
imshow(uint8(im22));
title('after trans of left to right');
```

```
subplot(2,2,4);
imshow(uint8(im_diff));
title('diff b/w trans and actual right');
```





Here the edges are visible in difference image b/w transformed image and right image because I selected the points manually using ginput() which might contains errors.