ROBOTICS & CV

PROJECT 1

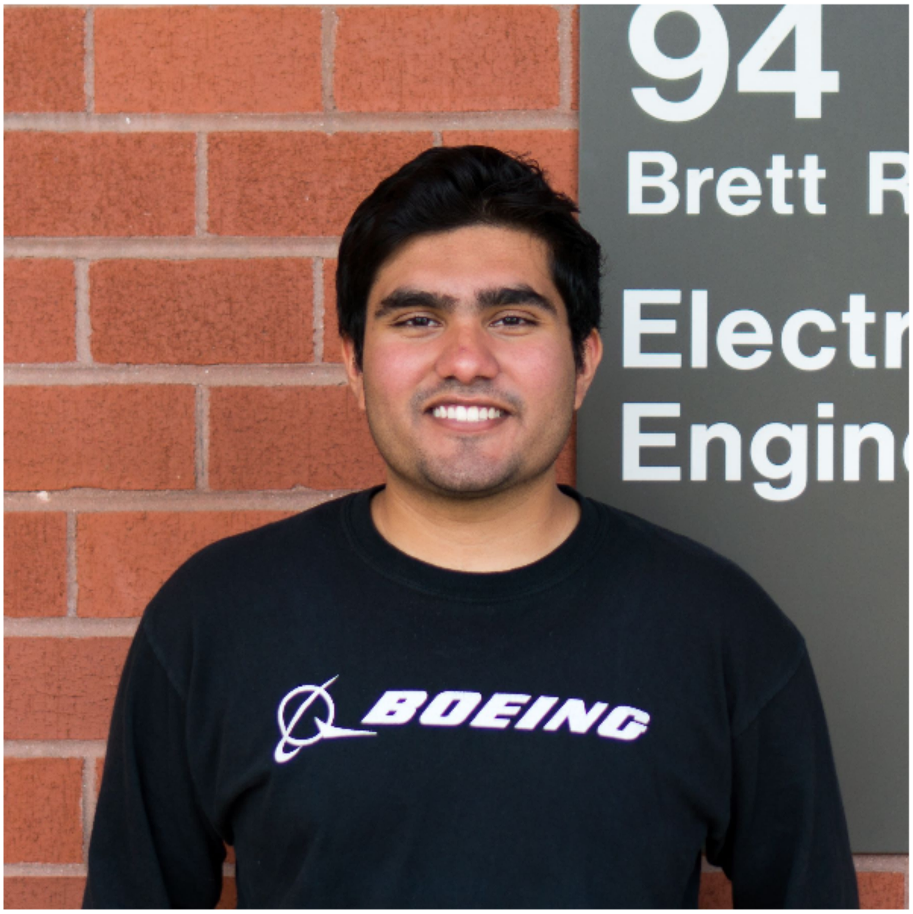
NiraL Shah

10/09/16 | RUID:150008870

Contents

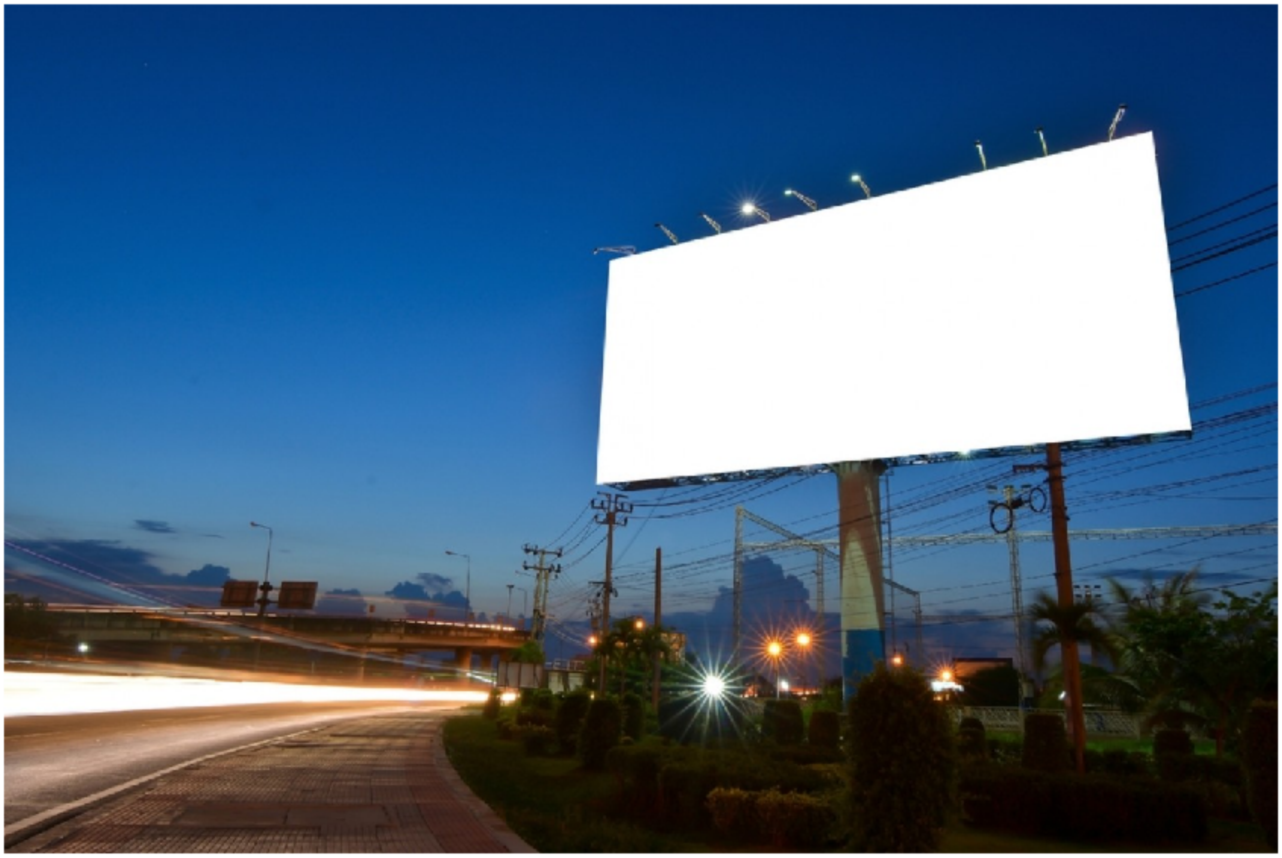
```
N = imread('NiralPhoto.jpg');
title('Original Photo');
%N = imresize(N, [500 NaN]);
imshow(N);
%[x y] = ginput(4);
```

```
Warning: Image is too big to fit on screen; displaying at 33%
```



```
B = imread('BillBoard.jpg');
title('Original Billboard Image');

%B = imresize(B, );
imshow(B);
%[x1 y1] = ginput(4);
```

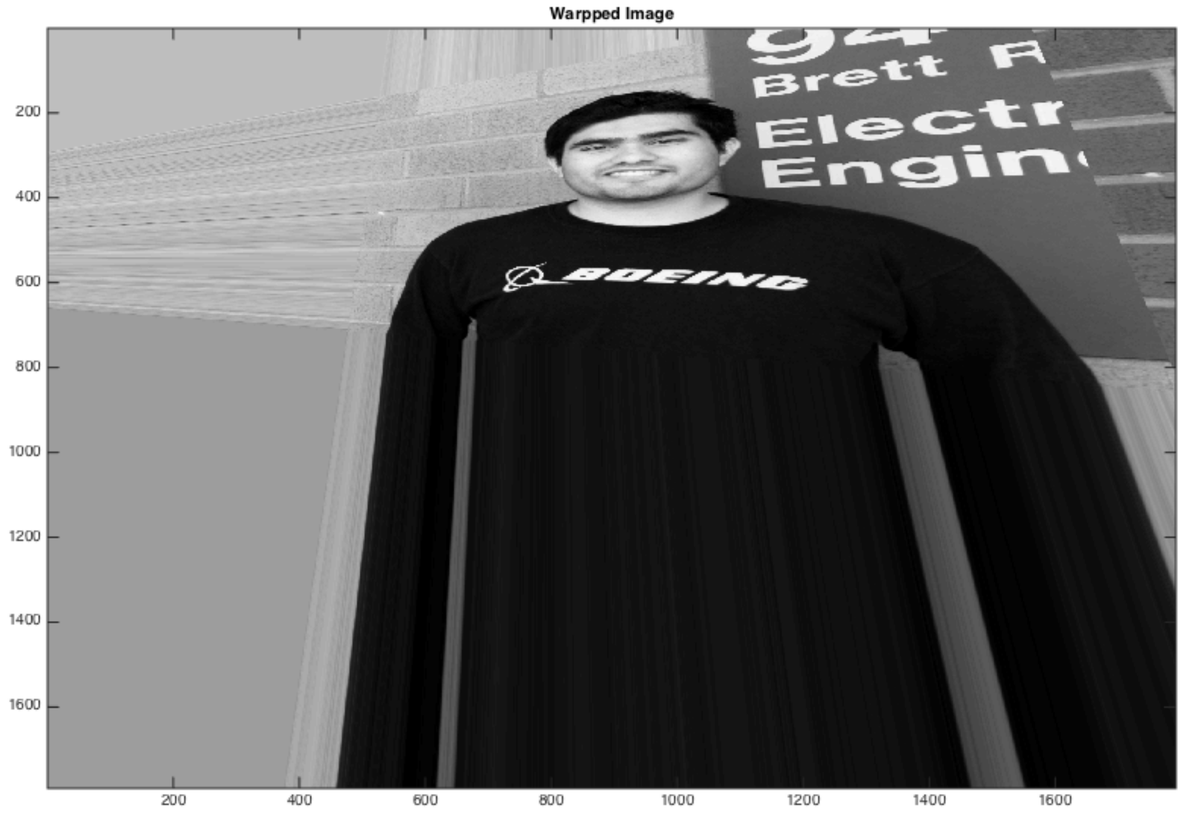
```
Warning: Image is too big to fit on screen; displaying at 67%
```



```
hh = homog([x y], [x1 y1]) % get the homography matrix:
regimg = homogwarp(N,N,hh);
```

```
hh =

   -0.0015
   -0.0001
    0.8160
   -0.0005
   -0.0020
    0.5781
   -0.0000
   -0.0000
   -0.0003
```

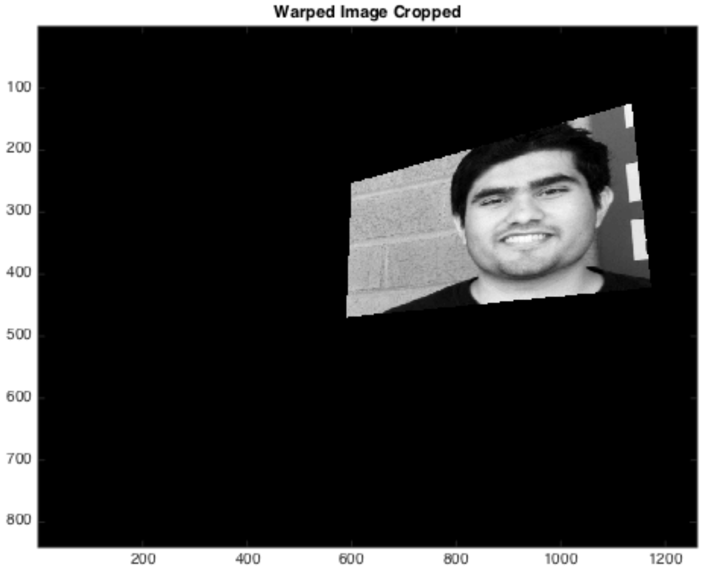
```
imagesc(regimg)
title('Warpped Image');
colormap(gray)
%imwrite(regimg,'warpedNiralImg.jpg');
```



crop my picture

```
BW = roipoly(B, x1,y1);
[X Y] = size(BW)


for i = 1:X
    for j= 1:Y
        if BW(i,j) == 1
           Out(i,j) = regimg(i,j);
        else
          Out(i,j)= 0;
        end
    end
end

figure;
imagesc(Out)
colormap(gray)
title('Warped Image Cropped');
```

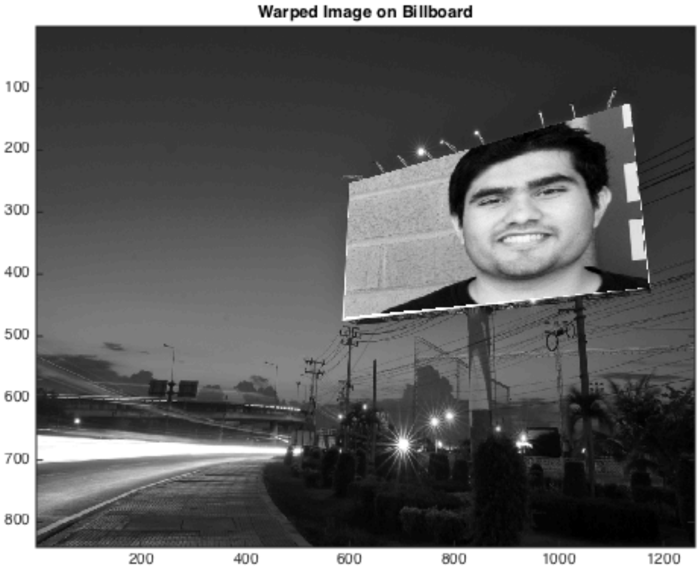
```
X =

   840


Y =

        1260
```



overlay my picture on to bill board

```
BW = roipoly(B, x1,y1);
[X Y] = size(BW)
billboard = rgb2gray(B);

for i = 1:X
    for j= 1:Y
        if BW(i,j) == 1
           scene(i,j) = Out(i,j);
        else
          scene(i,j) = billboard(i,j);
        end
    end
end

figure;
imagesc(scene)
colormap(gray)
title('Warped Image on Billboard');
```

```
X =

   840


Y =

        1260
```



Published with MATLAB® R2014b

Problem 2

```python

import homography
import matplotlib.pyplot as plt
import numpy
from PIL import Image


# identify and import other relevant libraries
# read and display image 1 (face image).
# Hint: dependencies: PIL
my_img = numpy.array(Image.open("/Users/niralshah/Desktop/NiralPhoto.jpg").convert('L'))
plt.figure()
plt.imshow(my_img)
tp = numpy.array(plt.ginput(4))
print("Points TP Selected Are:");
print(tp);

# read and display image 2 (billboard image)

billboard_img = numpy.array(Image.open("/Users/niralshah/Desktop/Billboard.jpg").convert('L'))
plt.figure();
plt.imshow(billboard_img)
fp = numpy.array(plt.ginput(4))

print("Points FP Selected Are:");
print(fp);

# compute homography matrix
H = homography.H_from_points(tp,fp)
print("homography matrix:");
print(H)

# homography matrix:
# [[  2.68207171e-04  -5.85968330e-04   2.56439718e-03]
#  [ -4.14582041e-04  -1.21910130e-04   3.42026854e-07]
#  [ -2.55780810e-03   8.01827515e-04   9.99992818e-01]]

```

```python
import numpy as np

def H_from_points(fp,tp):
#""" find homography H."""
    if fp.shape != tp.shape:
        raise RuntimeError, "number of points do not match"
    #create matrix for linear method, 2 rows for each correspondence pair
    nbr_correspondences = fp.shape[1]
    A = np.zeros((2*nbr_correspondences,9))

    for i in range(nbr_correspondences):
        A[2*i] = [-fp[0][i],-fp[1][i],-1,0,0,0,tp[0][i]*fp[0][i],tp[0][i]*fp[1][i],tp[0][i]]
        A[2*i+1] = [0,0,0,-fp[0][i],-fp[1][i],-1,tp[1][i]*fp[0][i],tp[1][i]*fp[1][i],tp[1][i]]

    U,S,V = np.linalg.svd(A)
    H = V[8].reshape((3,3))
    return H
```