Robert Gross - Robotics & Computer Vision

Project 1

```
% Problem 1
img = imread('brad_pitt_perfume_billboard.jpg');
img = im2double(img);
img = rgb2gray(img);
figure(1);
imshow(img);
[x,y] = ginput(4);      % original billboard image
img2 = imread('rob.jpg');
img2 = im2double(img2);
img2 = rgb2gray(img2);
img2 = imresize(img2, [250, 350]);  %just for a cleaner look
 figure(2);
imshow(img2);
 [xp,yp] = ginput(4);    % my pic

i=1;
A = [];
while (i<5)
    A1 = [x(i), y(i), 1, 0, 0, 0, -x(i)*xp(i), -y(i)*xp(i), -xp(i);
    0, 0, 0, x(i), y(i), 1, -x(i)*yp(i), -y(i)*yp(i), -yp(i)];
    A = [A;A1];
    i = i+1;
end
 [u,sig,v] = svd(A);
 v1 =  v(:,9);           % the 9th column forms the null space from n-
r
 figure(3);
 changed = homogwarp(img2, img, v1);
 imshow(changed);
 [length,height] = size(changed);
 for a = 1:length
     for b = 1:height
         check = inpolygon(a,b,[y;y(1)],[x;x(1)]);  %check if each
 pixel is
         % in the boundary- if so, submit that pixel to replace
 billboard
         % image's pixel
         if check == true
            img(a,b) = changed(a,b);
         end
     end
 end

 figure(4);
 imshow(img);   %final image
```

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

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

Published with MATLAB® R2015b

```
import numpy as num
import matplotlib.pyplot as matplt
from PIL import Image
import homography
# identify and import other relevant libraries
# read and display image 1 (face image).
im1 = num.array(Image.open("D:/robotics_cv/Proj. 1/rob.jpg").convert("L"))
matplt.figure()
matplt.imshow(im1)
tp = num.array(matplt.ginput(4))
im2 = num.array(Image.open("D:/robotics_cv/Proj.
1/brad_pitt_perfume_billboard.jpg").convert('L'))
matplt.figure();
matplt.imshow(im2)
fp = num.array(matplt.ginput(4))
H = homography.H_from_points(tp,fp)
print("Homography:")
print(H)
# Result gave:
#Homography:
#[[ 0.44782091 -0.14597362  0.14092835]
# [ 0.23610372  0.03637275  0.02359735]
# [ 0.0015813  -0.45766037  0.70086164]]
```

```
import numpy as num
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 = num.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 = num.linalg.svd(A)
   H = V[8].reshape((3,3))
   return H;
```