B. M. D. S. Karunarathna

180308C

EN2550 – Fundamentals of Image Processing & Machine Vision

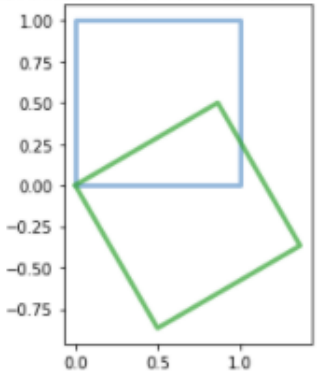
Assignment 02

1) 2D Transformations

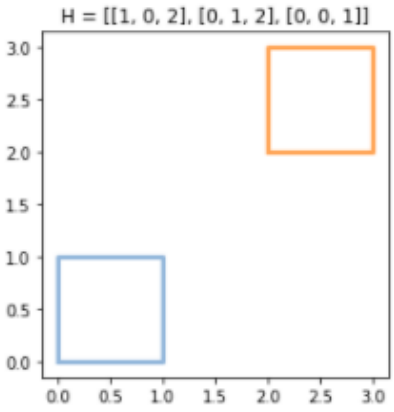
```
a, b, c, d = (0, 0, 1), (0, 1, 1), (1, 1, 1), (1, 0, 1)
P = np.array([a, b, c, d]).T
P = P/P[-1, :]
P = np.insert(P, 4, P[:, 0], axis=1)
x = P[0, :]
y = P[1, :]
def rotation(t):
    H=[[np.cos(t), np.sin(t), 0], [-np.sin(t), np.cos(t), 0], [0., 0., 1.]]
    return H
def translation(tx,ty):
    H=[[1,0,tx],[0,1,ty],[0,0,1]]
    return H
def scaling(sx,sy):
    H=[[sx,0,0],[0,sy,0],[0,0,1]]
    return H
def x_shear(shx):
    H=[[1,shx,0],[0,1,0],[0,0,1]]
    return H
def y_shear(shy):
    H=[[1,0,0],[shy,1,0],[0,0,1]]
    return H
def y_reflection():
    H=[[-1., 0., 0.], [0., 1., 0.], [0., 0., 1.]]
    return H
def x_reflection():
    H=[[1., 0., 0.], [0.,-1., 0.], [0., 0., 1.]]
    return H
transforms=[[rotation(np.pi/3),'#2ca02c'],[translation(2,2),'#ff7f0e'],[scaling(2,2),'#d62728'],
[x_shear(3),'#8c564b'],[y_shear(2),'#e377c2'],[y_reflection(),'#ff7f0e'],[x_reflection(),'#2ca02c']]
for i in transforms:
    Pt = np.matmul(i[0],P)
    Pt = np.insert(Pt,4,Pt[:,0],axis=1)
    xt = Pt[0, :]
    yt = Pt[1, :]
    fig, ax = plt.subplots(1, 1, sharex=True, sharey=True)
    ax.plot(x, y, color='#6699cc', alpha=0.7,linewidth=3, solid_capstyle='round', zorder=2)
    ax.plot(xt, yt, color=i[1], alpha=0.7,linewidth=3,solid_capstyle='round', zorder=2)
```

Rotation

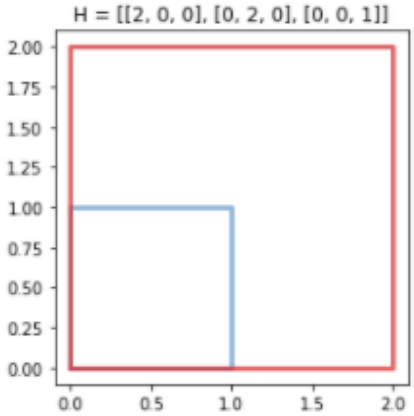
```
H = [[0.5000000000000001, 0.8660254037844386, 0], [-0.8660254037844386, 0.5000000000000001, 0], [0.0, 0.0, 1.0]]
```



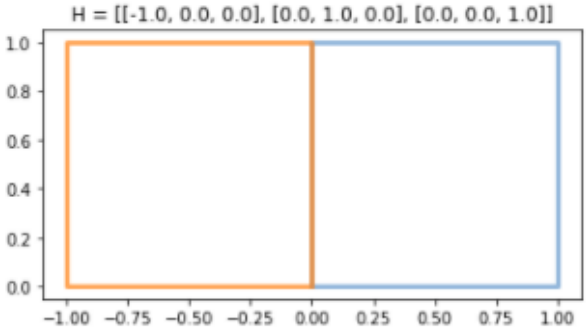
Translation



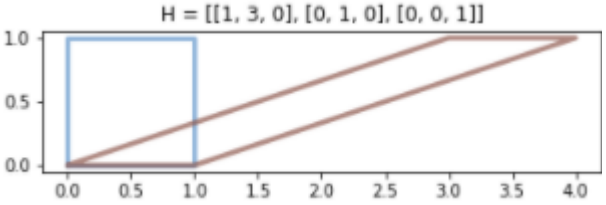
Scaling



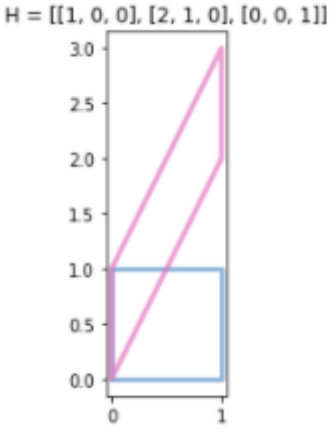
Reflection on x axis



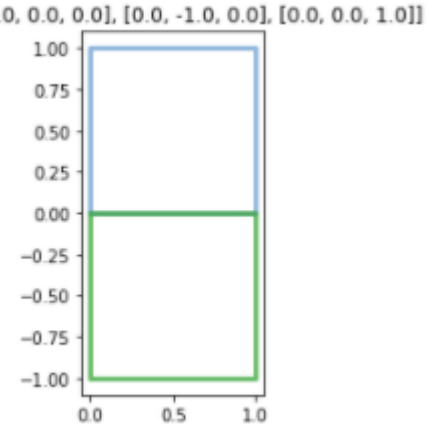
Vertical Shear



Horizontal Shear



Reflection on y axis



2) Warping Using a Given Homography

Image 01

Image 05

Stiched image

3) Computing the Homography Using Mouse-Clicked Points and Warping

```
Homography,s=cv.findHomography(p1,p2)
im5_warped = cv.warpPerspective(im5, np.linalg.inv(Homography), (1000,1000))
im5_warped[0:im1.shape[0], 0:im1.shape[1]] = im1
```

Image 01

```
[[334. 192.]
 [405. 206.]
 [527. 223.]
 [641. 294.]
 [516. 346.]]
```

Clicked points

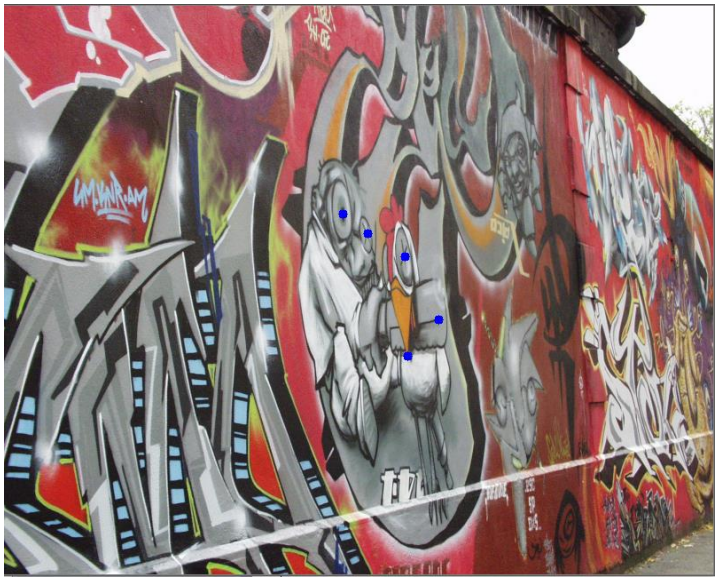


Image 05

```
[[382. 234.]
 [410. 256.]
 [452. 282.]
 [490. 353.]
 [455. 394.]]
```

Clicked points

Stiched Image

Homography

```
[[ 7.31316845e-01  3.35884985e-01  1.91153170e+02]
 [ 2.70947028e-01  1.53986590e+00 -7.97262805e+01]
 [ 5.92205674e-04  5.75433833e-04  1.00000000e+00]]
```

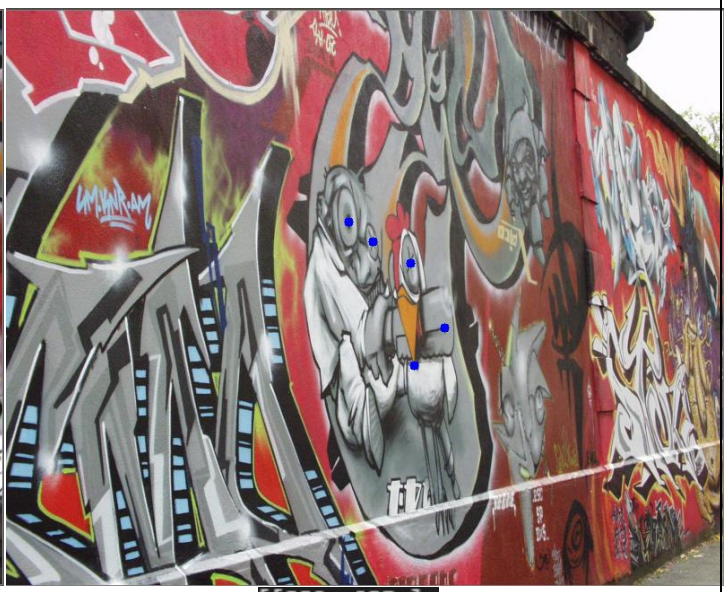
4) Computing the Homogrpahy Using Mouse-Clicked Points without OpenCV

```
A = np.empty((2*N, 9))
for i in range(N):
    A[2*i,:]=[-p1[i][0],-p1[i][1],-1, 0, 0, 0, p1[i][0]*p2[i][0], p1[i][1]*p2[i][0], p2[i][0]]
    A[2*i+1,:]=[0, 0, 0, -p1[i][0], -p1[i][1], -1,  p1[i][0]*p2[i][1], p1[i][1]*p2[i][1], p2[i][1]]
u,s,v=np.linalg.svd(A)
L=v[-1,:]/v[-1,-1]
Homography=L.reshape(3,3)
im5_warped = cv.warpPerspective(im5, np.linalg.inv(Homography), (1000,1000))
im5_warped[0:im1.shape[0], 0:im1.shape[1]] = im1
cv.namedWindow("Image 5 Warped",cv.WINDOW_AUTOSIZE)
cv.imshow("Image 5 Warped", im5_warped)
cv.waitKey(0)
cv.destroyAllWindows()
```

Image 01

```
[[334. 193.]
 [404. 208.]
 [526. 223.]
 [515. 346.]
 [640. 295.]]
```

Image 05



```
[[382. 235.]
 [409. 257.]
 [451. 281.]
 [455. 395.]
 [489. 353.]]
```

Homography

```
[[ 7.14114110e-01  2.77313708e-01  1.96804029e+02]
 [ 2.61825323e-01  1.47242347e+00 -7.08791252e+01]
 [ 5.93715821e-04  4.21763914e-04  1.00000000e+00]]
```

Switched Image

5) Switching More than Two Images Using Mouse Clicked Points

```
Homography01t02,mask1=cv.findHomography(p1,p2)
Homography01t03,mask2=cv.findHomography(p1,p3)
im2_warped = cv.warpPerspective(im2, np.linalg.inv(Homography01t02), (1000,1000))
im2_warped[0:im1.shape[0], 0:im1.shape[1]] = im1
cv.namedWindow("Image 2 Warped",cv.WINDOW_AUTOSIZE)
cv.imshow("Image 2 Warped", im2_warped)
cv.waitKey(0)
im3_warped = cv.warpPerspective(im3, np.linalg.inv(Homography01t03), (1000,1000))
cv.namedWindow("Image 3 Warped",cv.WINDOW_AUTOSIZE)
cv.imshow("Image 3 Warped", im3_warped)
cv.waitKey(0)
im2_warped[0:im3_warped.shape[0],0:200]=im3_warped[:,0:200]
cv.namedWindow("Stiched Image",cv.WINDOW_AUTOSIZE)
cv.imshow("Stiched Image", im2_warped)
cv.waitKey(0)
cv.destroyAllWindows()
```