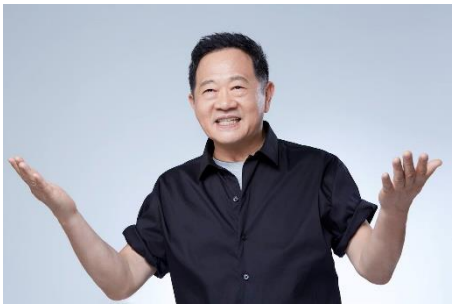
Computer Vision HW3 Report

B04901147 電機四 黃健祐

Part 1 – Estimating Homography

➢ Input Images

img1	img2	img3
		

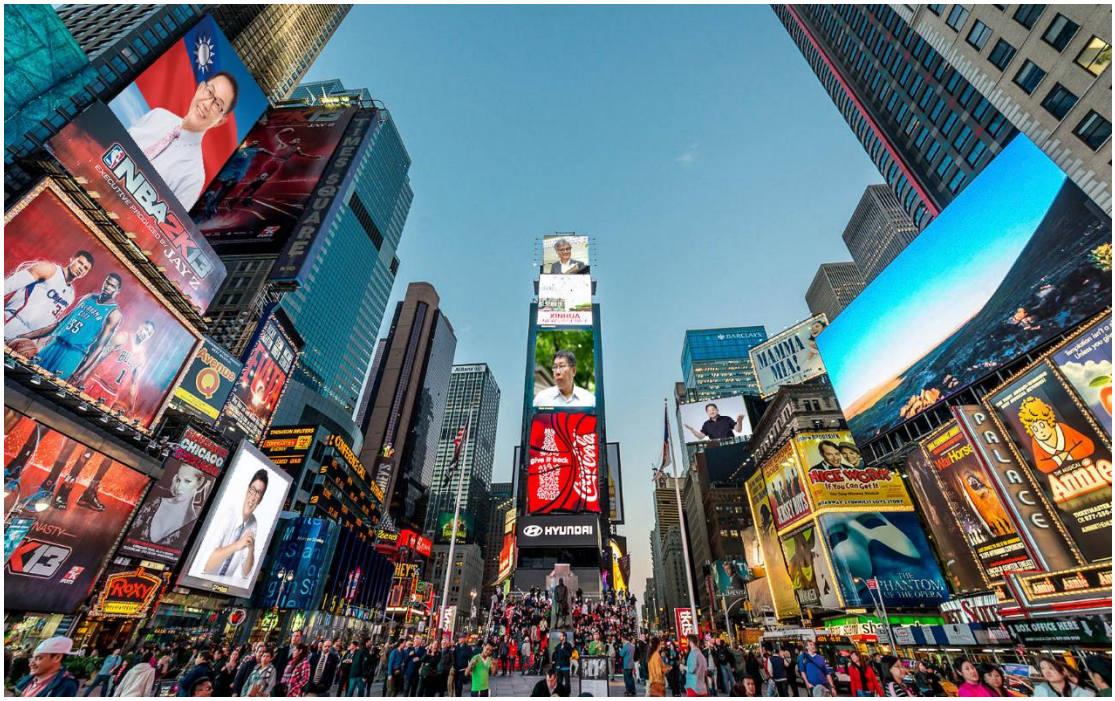
img4	img5
	

➢ Homography Implementation

```
def solve_homography(u, v):
    N = u.shape[0]
    if v.shape[0] is not N:
        print('u and v should have the same size')
        return None
    if N < 4:
        print('At least 4 points should be given')
    A = np.zeros((2*N, 8))
    b = np.zeros((2*N, 1))
    H = np.zeros((3, 3))
```

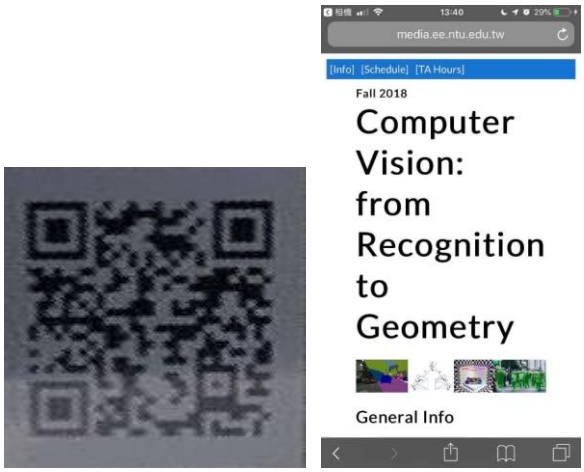
```
for i in range(N):
        ux = u[i,0]
        uy = u[i,1]
        vx = v[i,0]
        vy = v[i,1]
        A[2*i,:] = np.array([ux,uy,1,0,0,0,-ux*vx,-uy*vx])
        A[2*i+1,:] = np.array([0,0,0,ux,uy,1,-ux*vy,-uy*vy])
b = v.reshape(-1,1)
H = LA.solve(A,b)
H = np.concatenate((H,np.array([[1]])))
H = H.reshape(3,3)
return H
```

➢ Result



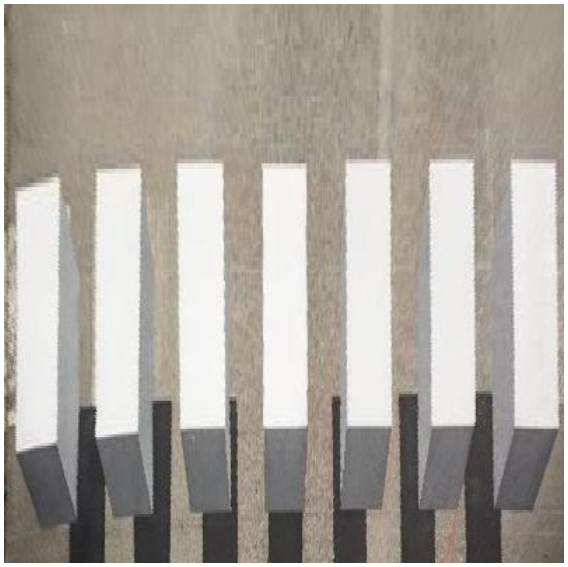
Part 2 – Unwarp the Screen

In backward warping, nearest neighbor method is applied. The output QR code can be identified by built-in camera in iOS and the corresponding website is http://media.ee.ntu.edu.tw/courses/cv/18F/ .



Part 3 – Unwarp the 3D Illusion

➢ Result



➢ Discussion

It is impossible to unwarp the input image to match the ground-truth top view by 2D homography. In the input image, the left-most bar is not parallel to the others, so it is not surprising that it is not parallel to the others in the output images. Moreover, while we are trying to unwarp the 3D illusion, the input image in fact contains only 2D information. Hence, in order to unwarp it more perfectly, other information must be given.