Computer Vision HW3 Report

B05901182 電機三 潘彥銘

Part1:

1. solve_homography(u, v):

```python
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')
    # if you take solution 2:
    A = np.zeros((2*N, 9))
    b = np.zeros((2*N, 1))
    # TODO: compute H from A and b
    for i in range(4):
        A[2 * i][0] = u[i][0]
        A[2 * i][1] = u[i][1]
        A[2 * i][2] = 1
        A[2 * i][6] = -( u[i][0] * v[i][0] )
        A[2 * i][7] = -( u[i][1] * v[i][0] )
        A[2 * i][8] = -v[i][0]
        A[2 * i + 1][3] = u[i][0]
        A[2 * i + 1][4] = u[i][1]
        A[2 * i + 1][5] = 1
        A[2 * i + 1][6] = -( u[i][0] * v[i][1] )
        A[2 * i + 1][7] = -( u[i][1] * v[i][1] )
        A[2 * i + 1][8] = -v[i][1]
    U, s, vh = np.linalg.svd(A.transpose() @ A, full_matrices = False)
    H = U[:, 8].reshape(3,3)
    return H
```

2. Answer:

Part2:

1. Answer:

2. Decoded link:
   http://media.ee.ntu.edu.tw/courses/cv/18F/

Part3:

1. Question: Can you get the parallel bars from the top view?
2. Answer: No, I can't. Because the red line in Fig.1 is not a straight line(like yellow line).

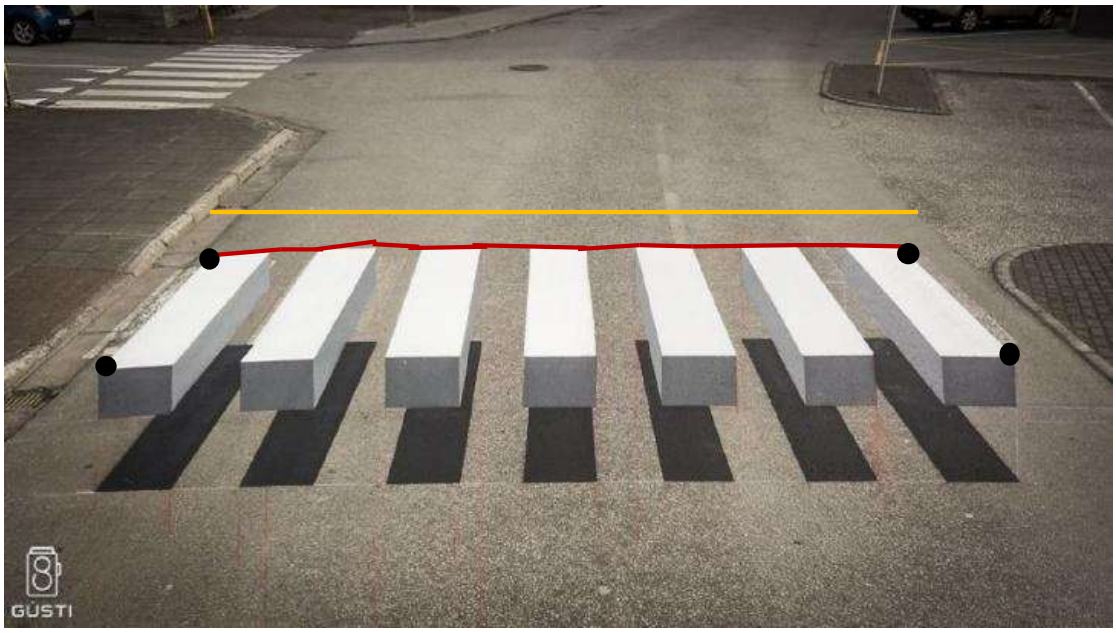

Fig.1

So if we use the four black corners in Fig.1 to do homography, we will get Fig.2 below, whose white bars is unparalleled.

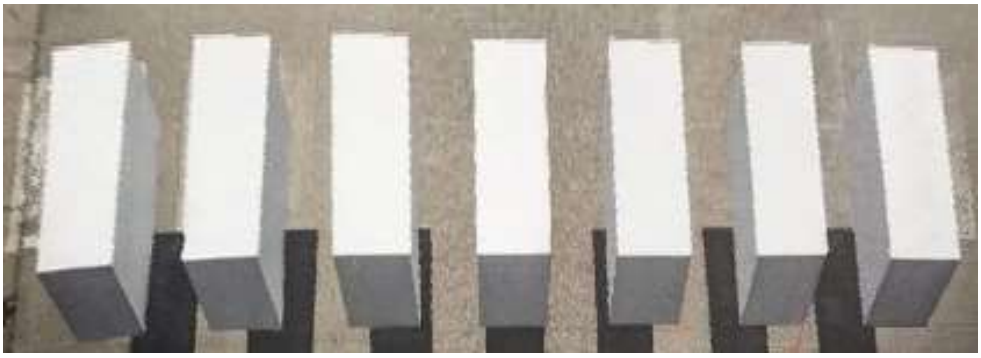

Fig.2