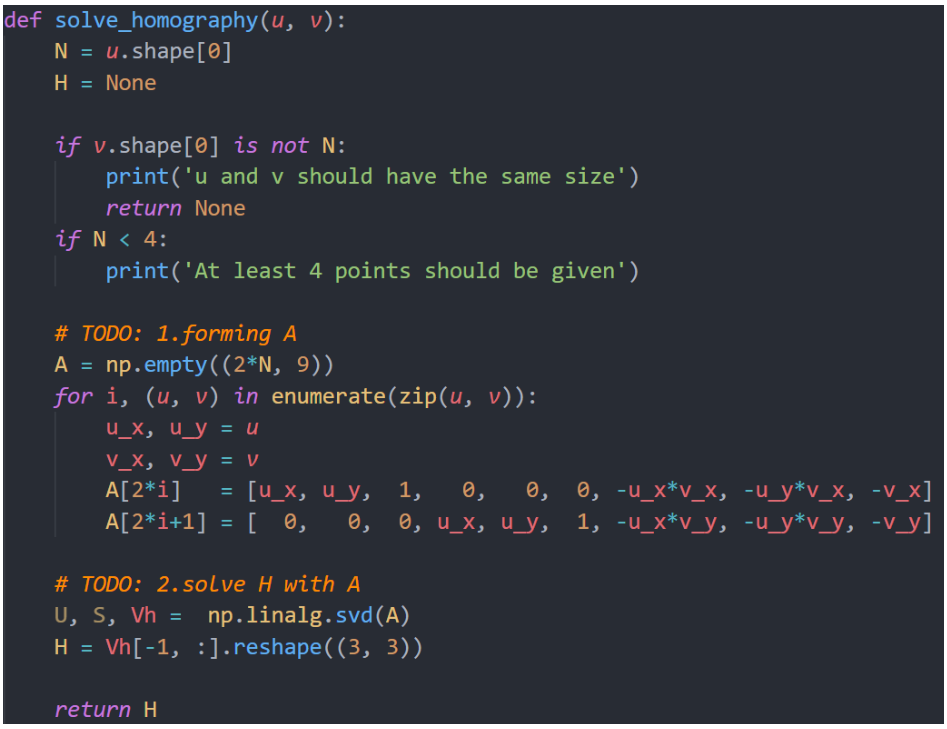
Computer Vision HW3 Report

Student ID: R10943117
Name: 陳昱仁

Part 1.

• Paste the function solve_homography( )

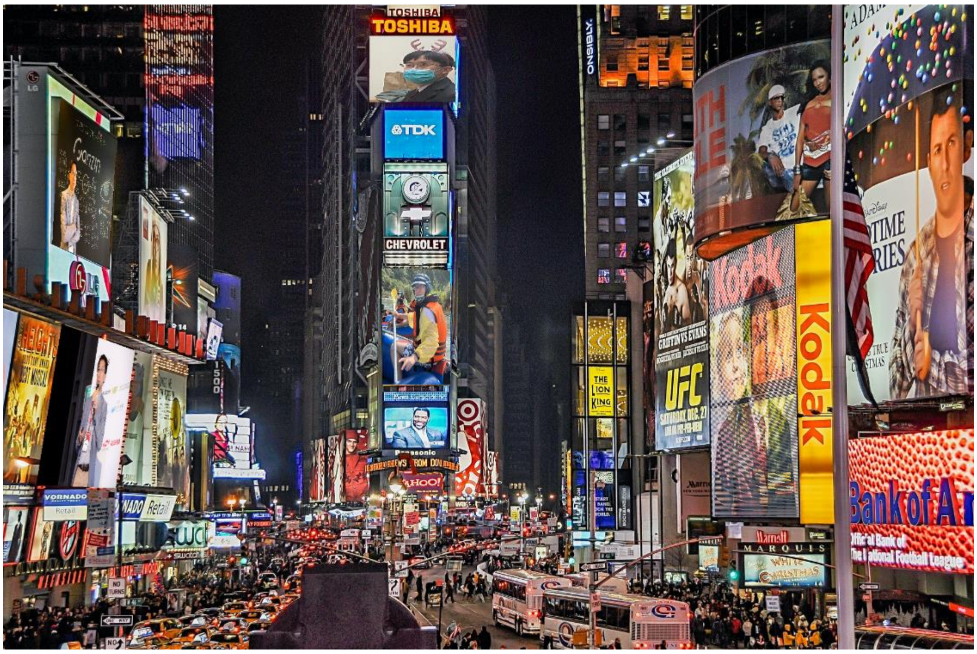
Ans:



```
def solve_homography(u, v):
    N = u.shape[0]
    H = None

    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')

    # TODO: 1.forming A
    A = np.empty((2*N, 9))
    for i, (u, v) in enumerate(zip(u, v)):
        u_x, u_y = u
        v_x, v_y = v
        A[2*i]   = [u_x, u_y,  1,   0,   0,  0, -u_x*v_x, -u_y*v_x, -v_x]
        A[2*i+1] = [  0,   0,  0, u_x, u_y,  1, -u_x*v_y, -u_y*v_y, -v_y]

    # TODO: 2.solve H with A
    U, S, Vh =  np.linalg.svd(A)
    H = Vh[-1, :].reshape((3, 3))

    return H
```

• Paste your warped canvas

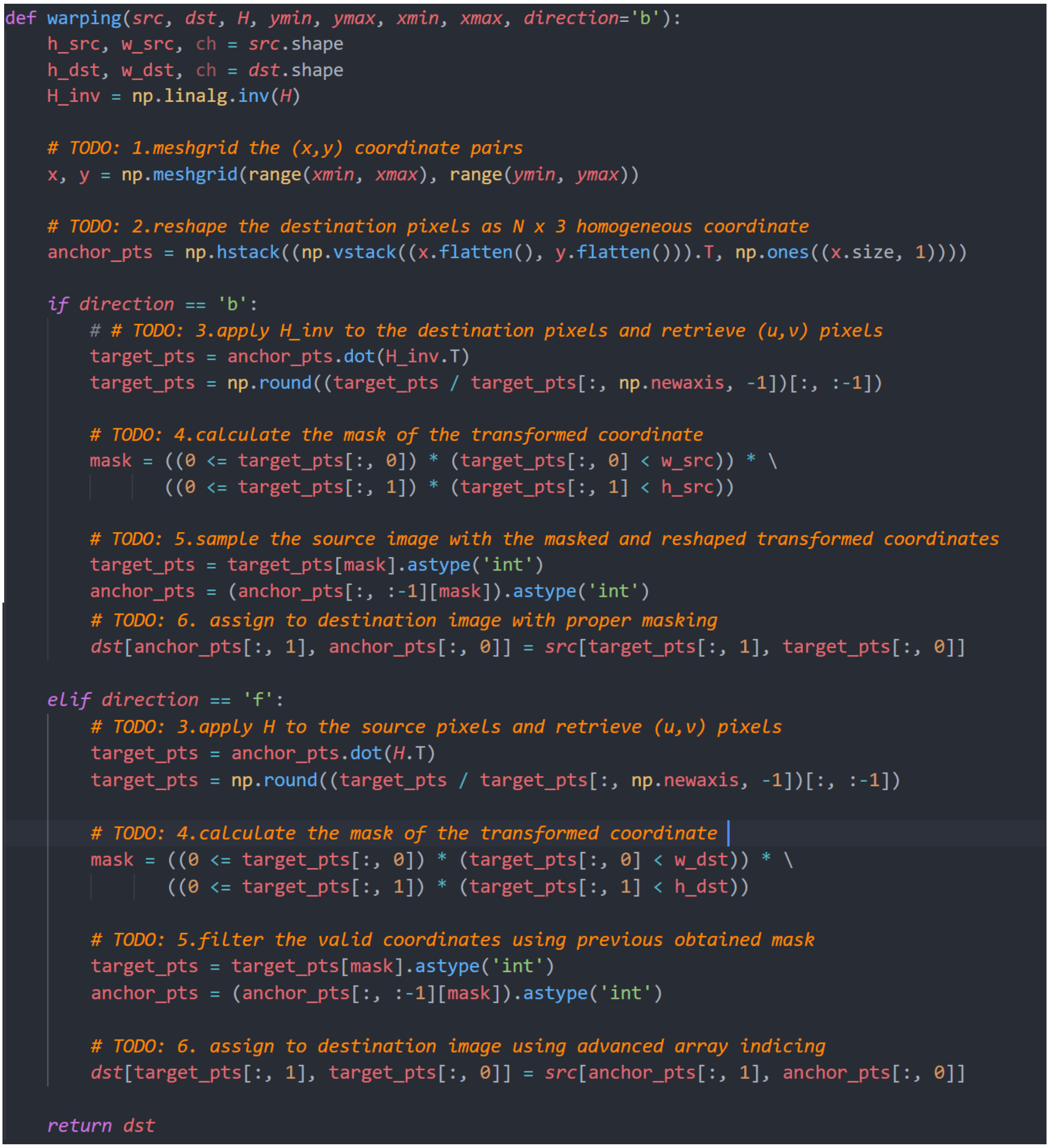
Ans:



Part 2.

• Paste the function code *warping( )* (both forward & backward)

Ans:



```python
def warping(src, dst, H, ymin, ymax, xmin, xmax, direction='b'):
    h_src, w_src, ch = src.shape
    h_dst, w_dst, ch = dst.shape
    H_inv = np.linalg.inv(H)

    # TODO: 1.meshgrid the (x,y) coordinate pairs
    x, y = np.meshgrid(range(xmin, xmax), range(ymin, ymax))

    # TODO: 2.reshape the destination pixels as N x 3 homogeneous coordinate
    anchor_pts = np.hstack((np.vstack((x.flatten(), y.flatten())).T, np.ones((x.size, 1))))

    if direction == 'b':
        # # TODO: 3.apply H_inv to the destination pixels and retrieve (u,v) pixels
        target_pts = anchor_pts.dot(H_inv.T)
        target_pts = np.round((target_pts / target_pts[:, np.newaxis, -1])[:, :-1])

        # TODO: 4.calculate the mask of the transformed coordinate
        mask = ((0 <= target_pts[:, 0]) * (target_pts[:, 0] < w_src)) * \
               ((0 <= target_pts[:, 1]) * (target_pts[:, 1] < h_src))

        # TODO: 5.sample the source image with the masked and reshaped transformed coordinates
        target_pts = target_pts[mask].astype('int')
        anchor_pts = (anchor_pts[:, :-1][mask]).astype('int')
        # TODO: 6. assign to destination image with proper masking
        dst[anchor_pts[:, 1], anchor_pts[:, 0]] = src[target_pts[:, 1], target_pts[:, 0]]

    elif direction == 'f':
        # TODO: 3.apply H to the source pixels and retrieve (u,v) pixels
        target_pts = anchor_pts.dot(H.T)
        target_pts = np.round((target_pts / target_pts[:, np.newaxis, -1])[:, :-1])

        # TODO: 4.calculate the mask of the transformed coordinate
        mask = ((0 <= target_pts[:, 0]) * (target_pts[:, 0] < w_dst)) * \
               ((0 <= target_pts[:, 1]) * (target_pts[:, 1] < h_dst))

        # TODO: 5.filter the valid coordinates using previous obtained mask
        target_pts = target_pts[mask].astype('int')
        anchor_pts = (anchor_pts[:, :-1][mask]).astype('int')

        # TODO: 6. assign to destination image using advanced array indicing
        dst[target_pts[:, 1], target_pts[:, 0]] = src[anchor_pts[:, 1], anchor_pts[:, 0]]

    return dst
```

• Briefly introduce the interpolation method you use

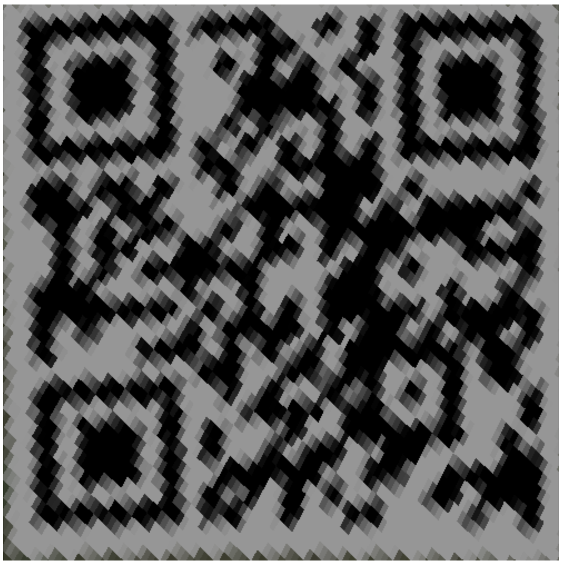
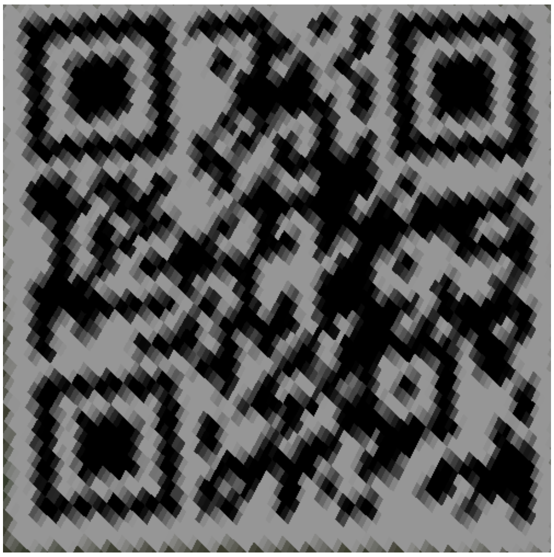
Ans:

使用的方法是 Nearest neighbor，對座標四捨五入找最近的 pixel 來內插。

Part 3.

• Paste the 2 warped QR code and the link you find

Ans:

Output3-1	Output3-2
	

QR code result: http://media.ee.ntu.edu.tw/courses/cv/21S/

• Discuss the difference between 2 source images, are the warped results the same or different?

Ans:

第一張 source image 相較於第二張較為方正，第二張可看出不自然的彎曲。兩者 warped result 都可正確掃出 2021 CV 網站，但第二張 QR code 結果相較第一張模糊。

• If the results are the same, explain why. If the results are different, explain why?

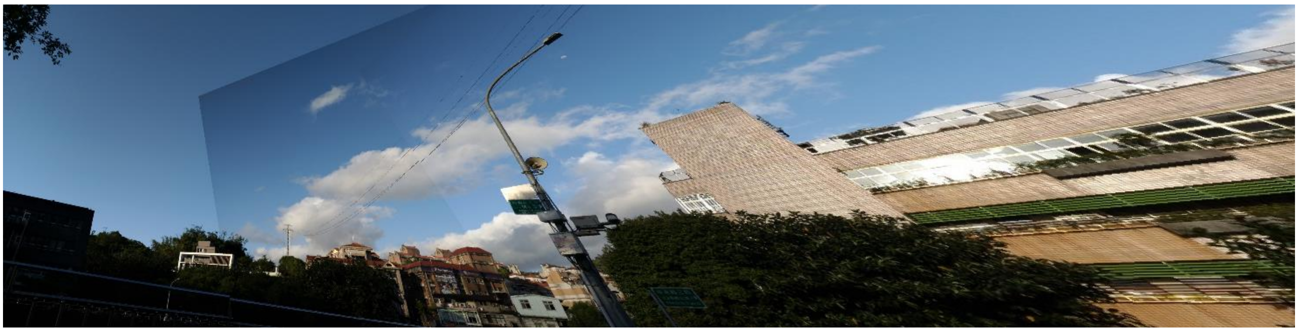
Ans:

兩者結果有些微不同，因 homography 只適用於 planer 上，若非 planer，自然無法完美的還原原圖，第二張圖因有 distortion，非 planer，所以會相對模糊。

Part 4.

• Paste your stitched panorama.

Ans:



• Can all consecutive images be stitched into a panorama?

Ans:

不行，理由如下。

• If yes, explain your reason. If not, explain under what conditions will result in failure?

Ans:

連續的圖片需有相同位置的 optical center，意即 camera 只能旋轉，不能有任何平移，若有平移則會無法還原全景圖。