**Computer Vision HW3 Report**

Student ID: R11522810

Name: 曾建堯

**Part1:**

* **Paste output1.png in your report.**



**Part2:**

* Paste the function code *solve\_homography(u, v)* and *warping( )*in your report. (Both forward & backward)

def solve\_homography(u, v):

    """

    This function should return a 3-by-3 homography matrix,

    u, v are N-by-2 matrices, representing N corresponding points for v = T(u)

    :param u: N-by-2 source pixel location matrices

    :param v: N-by-2 destination pixel location matrices

    :return:

    """

    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.zeros((2\*N, 9))

    B = np.zeros((2\*N, 1))

    for i in range(N):

        A[2\*i, 0] = u[i, 0]

        A[2\*i, 1] = u[i, 1]

        A[2\*i, 2] = 1

        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, 6] = -1 \* u[i, 0] \* v[i, 0]

        A[2\*i, 7] = -1 \* u[i, 1] \* v[i, 0]

        A[2\*i, 8] = -1 \* v[i, 0]

        A[2\*i+1, 6] = -1 \* u[i, 0] \* v[i, 1]

        A[2\*i+1, 7] = -1 \* u[i, 1] \* v[i, 1]

        A[2\*i+1, 8] = -1 \* v[i, 1]

    # TODO: 2.solve H with A

    \_, \_, V = np.linalg.svd(A)

    # print("V\n",V)

    h = V[-1, :] / V[-1,-1]

    H = h.reshape(3, 3)

    return H

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 = np.arange(xmin, xmax)

    y = np.arange(ymin, ymax)

    ux, uy = np.meshgrid(x, y)

    ux = ux.reshape(-1).astype(int)

    uy = uy.reshape(-1).astype(int)

    # TODO: 2.reshape the destination pixels as N x 3 homogeneous coordinate

    desPixel = np.array([ux, uy, np.ones(ux.shape[0])])

    if direction == 'b':

        # TODO: 3.apply H\_inv to the destination pixels and retrieve (u,v) pixels, then reshape to (ymax-ymin),(xmax-xmin)

        newpixel = H\_inv.dot(desPixel)

        newpixel = newpixel / newpixel[-1]

        vx = np.floor(newpixel[0, :]).flatten().astype(int)

        vy = np.floor(newpixel[1, :]).flatten().astype(int)

        # TODO: 4.calculate the mask of the transformed coordinate (should not exceed the boundaries of source image)

        mask = np.where((vx >=0) & (vx < w\_src) & (vy >=0) & (vy < h\_src))

        # TODO: 5.sample the source image with the masked and reshaped transformed coordinates

        ux, uy = ux[mask], uy[mask]

        vx, vy = vx[mask], vy[mask]

        # TODO: 6. assign to destination image with proper masking

        dst[uy, ux] = src[vy, vx]

    elif direction == 'f':

        # TODO: 3.apply H to the source pixels and retrieve (u,v) pixels, then reshape to (ymax-ymin),(xmax-xmin)

        newpixel = H.dot(desPixel)

        newpixel = newpixel / newpixel[-1]

        vx = np.floor(newpixel[0, :]).reshape(-1).astype(int)

        vy = np.floor(newpixel[1, :]).reshape(-1).astype(int)

        # TODO: 4.calculate the mask of the transformed coordinate (should not exceed the boundaries of destination image)

        mask = np.where((vx >=0) & (vx < w\_dst) & (vy >=0) & (vy < h\_dst))

        # TODO: 5.filter the valid coordinates using previous obtained mask

        ux, uy = ux[mask], uy[mask]

        vx, vy = vx[mask], vy[mask]

        # TODO: 6. assign to destination image using advanced array indicing

        dst[vy, vx] = src[uy, ux]

    return dst

* **Briefly introduce the interpolation method you use**

在本題的插植(interpolation)方法是直接將數值取下高斯，在尋找在src中對應的整數

**Part3:**

* Can you get the correct QR code link from both images?

|  |  |
| --- | --- |
|  |  |
| Source3-1 | Source3-2 |
| <https://qrgo.page.link/jc2Y9>  可利用<https://qrcodescan.in/> , iphone掃出 | <https://qrgo.page.link/jc2Y9>  僅能利用iphone內建 |

皆可以利用QR code連接到 <http://media.ee.ntu.edu.tw/courses/cv/21S/> 上

* **Discuss the difference between 2 source images, are the warped results the same or different? If the results are the same, explain why. If the results are different, explain why.**

由上圖的結果可以看到，雖然兩張圖所導引到的網站皆相同，但可以觀察到後者較為模糊，推論可能的原因是因為前者的圖片較為方正，而後者的原圖因為視角的因素變得較為扭曲，故圖片經過轉換後變得較為模糊。

**Part4:**

****

**Q: Can all consecutive images be stitched into a panorama? If yes, explain your reason. If not, explain under what conditions will result in a failure?**

否，1) 在光源變換較大的情況下，因為本題所採用的ORB feature易受到光線明暗變化的影響，顧若採用自動曝光(例如手機等)，便無法成功匹配。2) homography適用的時機是在planer的情況，若非此情況，所求出的結果並非正確，便會造成影像歪曲。3) 相機平移，平移容易造成視角與影像比例的變換，造成feature匹配時產生問題。