

Computer Vision HW3 Report

Student ID: R11921100

Name: 溫威領

Part 1.

- Result: output1.png



- Discussion:

由於 part 1 使用 forward warping，所以當 dst 比 src 的 image size 還要大的時候就會產生 hole，如右邊部分可看到仍有未被覆蓋的 dst。

Part 2.

● solve_homography function:

```
4 def solve_homography(u, v):
5     """..."""
12    N = u.shape[0]
13    H = None
14    if v.shape[0] is not N:
15        print('u and v should have the same size')
16        return None
17    if N < 4:
18        print('At least 4 points should be given')
19    # TODO: 1.forming A
20    A = np.zeros((2*N, 9))
21    for i in range(N):
22        A[i*2, :] = [u[i][0], u[i][1], 1, 0, 0, 0, -1*u[i][0]*v[i][0], -1*u[i][1]*v[i][0], -1*v[i][0]]
23        A[i*2+1, :] = [0, 0, 0, u[i][0], u[i][1], 1, -1*u[i][0]*v[i][1], -1*u[i][1]*v[i][1], -1*v[i][1]]
24    # TODO: 2.solve H with A
25    U, S, V = np.linalg.svd(A)
26    # print('U:', U.shape, 'S:', S.shape, 'V:', V.shape)
27    H = V.T[:, -1].reshape((3, 3))
28    # print(H)
29    return H
```

● warping function:

```
70 def warping(src, dst, H, ymin, ymax, xmin, xmax, direction='b', blending=False):
71     """..."""
102    h_src, w_src, ch = src.shape
103    h_dst, w_dst, ch = dst.shape
104    H_inv = np.linalg.inv(H)
105    # TODO: 1.meshgrid the (x,y) coordinate pairs
106    Ux, Uy = np.meshgrid(np.arange(xmin, xmax), np.arange(ymin, ymax))
107    # TODO: 2.reshape the destination pixels as N x 3(x) 3 x N(o) homogeneous coordinate
108    U = np.concatenate([(Ux.reshape(-1)), (Uy.reshape(-1)), (np.ones((xmax - xmin) * (ymax - ymin)))], axis=0)
109    # print('\nU.shape: ', U.shape, '\nU:\n', U.astype(int), '\n')
```

```
110 if direction == 'b':
111     ...
127    V = np.dot(H_inv, U)
128    V = np.around(V/V[2])
129    # print('\nV.shape: ', V.shape, '\nV:\n', V, '\n')
130    mask = (V[0] >= 0) & (V[0] <= w_src-1) & (V[1] >= 0) & (V[1] <= h_src-1)
131    mUx, mUy = U[0][mask], U[1][mask]
132    mU = np.concatenate([(mUx), (mUy), (np.ones(mUx.shape))], axis=0).astype(int)
133    mV = np.dot(H_inv, mU)
134    mV = np.around(mV/mV[2]).astype(int)
```

```
135 if blending:
136     ...
142     for y in range(ymax):
143         ym = mU[1] == y
144         if True in ym:
145             am = np.sum(dst[mU[1][ym], mU[0][ym]], axis=1) != 0
146             if True in am:
147                 u_max, u_min = max(mU[0][ym][am]), min(mU[0][ym][am])
148                 if u_max == u_min:
149                     a = 1 - (mU[0][ym][am]-u_min)
150                 else:
151                     a = 1 - (mU[0][ym][am]-u_min)/(u_max-u_min)
152                 a = np.array([a, a, a]).T
153                 dst[mU[1][ym][am], mU[0][ym][am]] = dst[mU[1][ym][am], mU[0][ym][am]]*a + \
154                     src[mV[1][ym][am], mV[0][ym][am]]*(1-a)
155             dst[mU[1][ym][~am], mU[0][ym][~am]] = src[mV[1][ym][~am], mV[0][ym][~am]]
156         else:
157             dst[mU[1], mU[0]] = src[mV[1], mV[0]]
158     ...
```

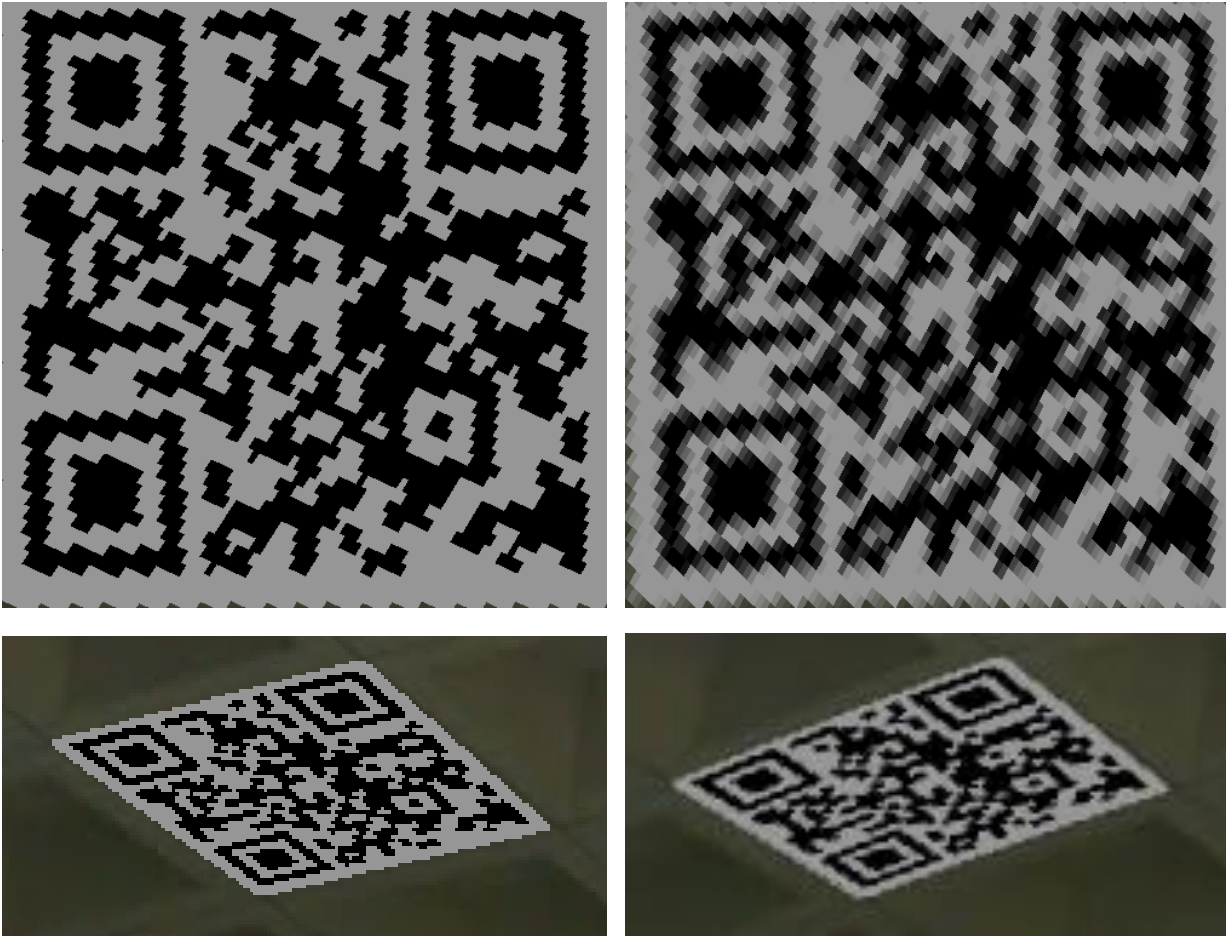
```
162 elif direction == 'f':
163     ...
186     V = np.dot(H, U)
187     V = np.around(V/V[2]).astype(int)
188     mask = ((V[0] >= 0) & (V[0] < w_dst)) & ((V[1] >= 0) & (V[1] < h_dst))
189     mVx, mVy = V[0][mask], V[1][mask]
190     dst[mVy, mVx] = src[mask.reshape(ymax-ymin, xmax-xmin)]
191 return dst
```

- Interpolation Method:

使用 nearest neighbor interpolation，將小數座標經 `np.around()` 四捨五入到整數位。

Part 3.

- Result: 左 output3_1.png , 右 output3_2.png



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

可以，上面四張圖都可以連結到 <https://qrgo.page.link/jc2Y9>。

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

左圖 src 邊界是直的，而右圖 src 邊界則為弧線，如結果顯示，左右圖 warping 後的結果不相同。

- If the results are different, explain why.

推測因為 homography 是將 pixel 的 xy 座標做線性轉換，所以無法很好地將右圖 src 的弧線(非線性)轉成 dst 的直線(線性)，所以右圖 dst 仍有些弧度，而左圖是直線轉換所以有較好的結果。

Part 4.

- Result Without Blending



- Result With Blending



- Blending Method:

將 dst 非零區塊與 src warping 區塊的重疊區域進行 linear alpha blending。重疊區域 y 軸上每條 x 方向的 dst 不透明度 alpha 從左到右漸變小，意旨該區域越左邊越接近 dst，而越右邊越接近 src，所以不會有明顯的邊界。

- Can all consecutive images be stitched into a panorama?

我認為不是所有連續的圖片都可以拼接成一張全景圖。

- If not, explain under what conditions will result in a failure?

1)當相機有平移的情況下，會看到需重疊物品的不同角度，所以無法完整的連續場景重疊。例如需重疊之正方體，第一張圖片看到左面，而第二張圖片看到右面，顯然無法經由 homography 投影座標達到貼合。

2)當圖片有景深發生變形時，如 part 3 無法將有非線性曲線轉為直線，所以無法將各個景深變形的連續圖片拼接成一張全景圖。