CS-E4850 Computer Vision, Answers to Exercise Round 11

Yangzhe Kong, Student number: 765756

November 28, 2019

Exercise 1. Camera calibration

The definition of function **camcalibDLT** goes as follows.

```python
def camcalibDLT(Xworld, Xim):
    N = Xworld.shape[0]
    A = np.zeros((2*N,12))
    for i in range(N):
        A[2*i,:] = np.hstack((np.zeros((4)), Xworld[i,:], -Xim[i,1]*Xworld[i,:]))
        A[2*i+1,:] = np.hstack((Xworld[i,:], np.zeros(4), -Xim[i,0]*Xworld[i,:]))

    M = np.dot(A.T, A)

    u,s,v = np.linalg.svd(M)
    ev = v[-1,:]
    P = np.reshape(ev, (3,4))

    return P
```

And the plots are shown in Figure 1. We can see that the projected points exactly overlap with manually localized points.

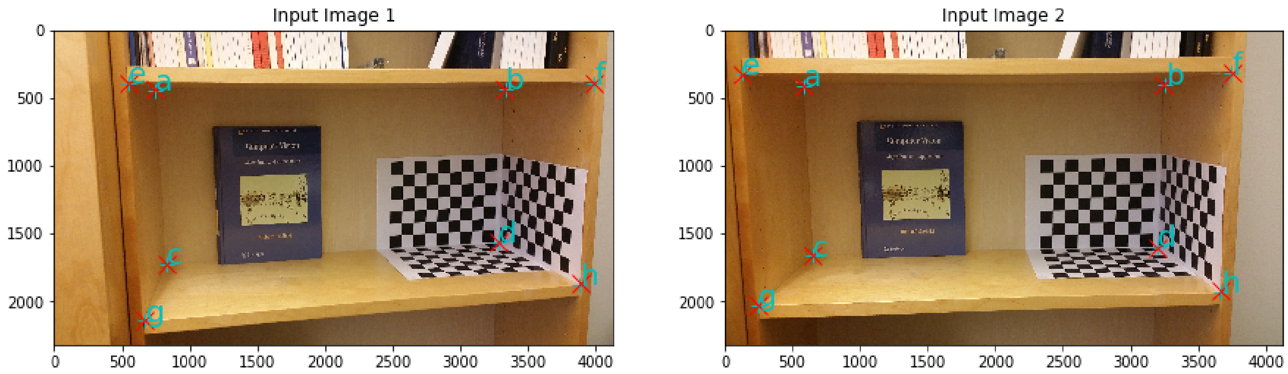

Figure 1: Visualization of the the projected points and the manually localized points

In addition, the calculation results of the intrinsic camera calibration matrice for first camera K_1 and the intrinsic camera calibration matrice for first camera K_2 are:

$$K_1 = \begin{bmatrix} 3840.34 & 5.19 & 2050.42 \\ 0 & 3820.54 & 1170.11 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 3848.79 & -20.12 & 2062.03 \\ 0 & 3844.46 & 1156.92 \\ 0 & 0 & 1 \end{bmatrix}$$

Exercise 2. Triangulation

The definition of function **trianglin** goes as follows.

```
def trianglin(P1, P2, x1, x2):

     A = np.vstack((x1[0]*P1[2] - P1[0],
                    x1[1]*P1[2] - P1[1],
                    x2[0]*P2[2] - P2[0],
                    x2[1]*P2[2] - P2[1],
                   ))
     M = np.dot(A.T,A)

     u,s,v = np.linalg.svd(M)
     ev = v[-1,:]
     return ev
```

And the results of the picture width and the picture heights are:
Picture width: 138.49 mm
Picture height: 106.26 mm