3DCV&DL HW1

R11922196 林佑鑫

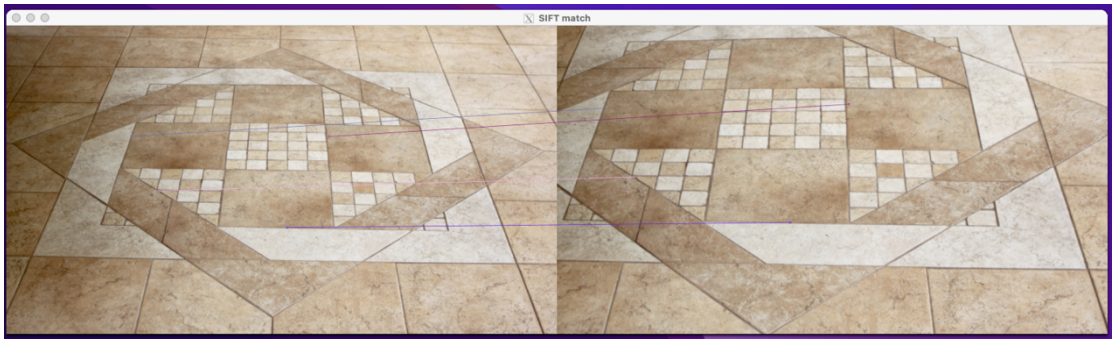
Problem 1: Homography estimation

Screenshots:

ImageA to ImageB with 4 correspondences:

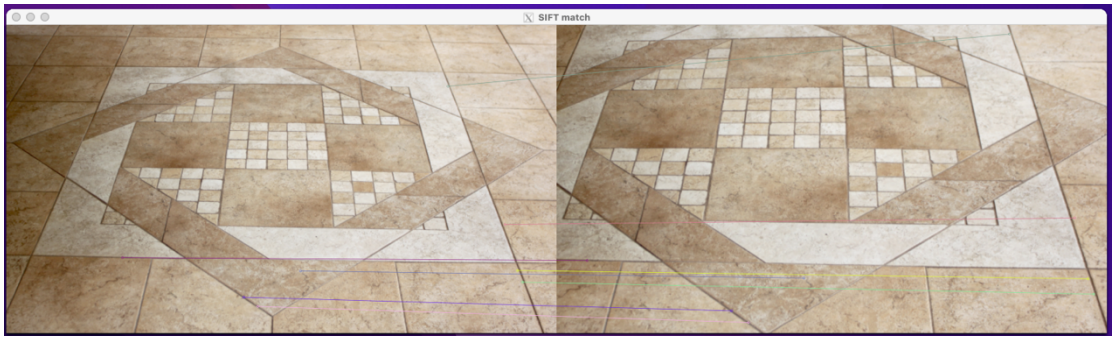
match 圖太大張，所以 show result 時我把長寬都縮成一半，才截得到兩邊對比，

只有在 show result 時有縮小，故不會影響數值的結果。



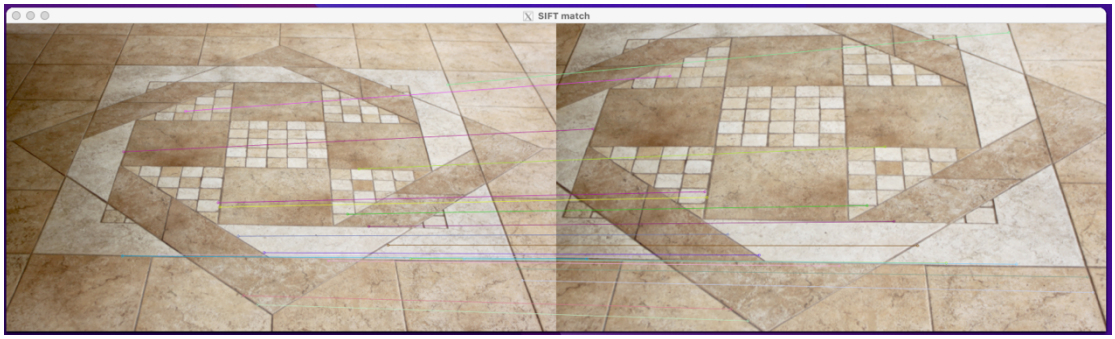

```
Chosen points in image1:
[[426.65710449 460.55276489]
 [350.75933838 308.04724121]
 [902.00769043 309.03338623]
 [785.0255127  562.96728516]]
Chosen points in image2:
[[215.7739563  421.60299683]
 [131.82395935 230.34104919]
 [816.68505859 218.7766571 ]
 [651.76733398 548.9085083 ]]
Matrix H:
[[-3.84349609e-03  1.46772680e-04  8.79129528e-01]
 [ 1.15167184e-04 -4.08202378e-03  4.76538768e-01]
 [ 1.84893148e-07 -7.49682795e-08 -3.25665051e-03]]
Matrix H with points normalization:
[[ 6.39882824e-01 -2.44353878e-02 -1.46361508e+02]
 [-1.91735601e-02  6.79594007e-01 -7.93363557e+01]
 [-3.07818577e-05  1.24810624e-05  5.42182086e-01]]
DLT error: 1.2819116002906004
Normalized DLT error: 1.281911600092404
```

ImageA to ImageB with 8 correspondences:




```
Chosen points in image1:
[[ 703.31494141  785.1206665 ]
 [ 822.95373535  686.16192627]
 [ 324.92495728  648.97821045]
 [ 663.90979004  759.64154053]
 [1444.26049805  717.37335205]
 [1394.82019043  556.7880249 ]
 [1430.12084961  685.02209473]
 [1235.96264648  170.8475647 ]]
Chosen points in image2:
[[ 536.49676514  830.14440918]
 [ 691.8638916   706.02575684]
 [  85.19613647  657.10717773]
 [ 489.32406616  797.56860352]
 [1502.89550781  751.02069092]
 [1449.60876465  538.21917725]
 [1487.03234863  707.9876709 ]
 [1267.09106445   25.19884491]]
Matrix H:
[[-3.84591536e-03  1.40337900e-04  8.81211250e-01]
 [ 1.20419586e-04 -4.08642223e-03  4.72678054e-01]
 [ 1.96741092e-07 -8.42387332e-08 -3.26588112e-03]]
Matrix H with points normalization:
[[-6.71791150e-01  2.45385391e-02  1.53909206e+02]
 [ 2.10345197e-02 -7.13800009e-01  8.25642175e+01]
 [ 3.43663523e-05 -1.46951231e-05 -5.70486905e-01]]
DLT error: 0.151790954988882 62
Normalized DLT error: 0.15636756609172817
```

ImageA to ImageB with 20 correspondences:

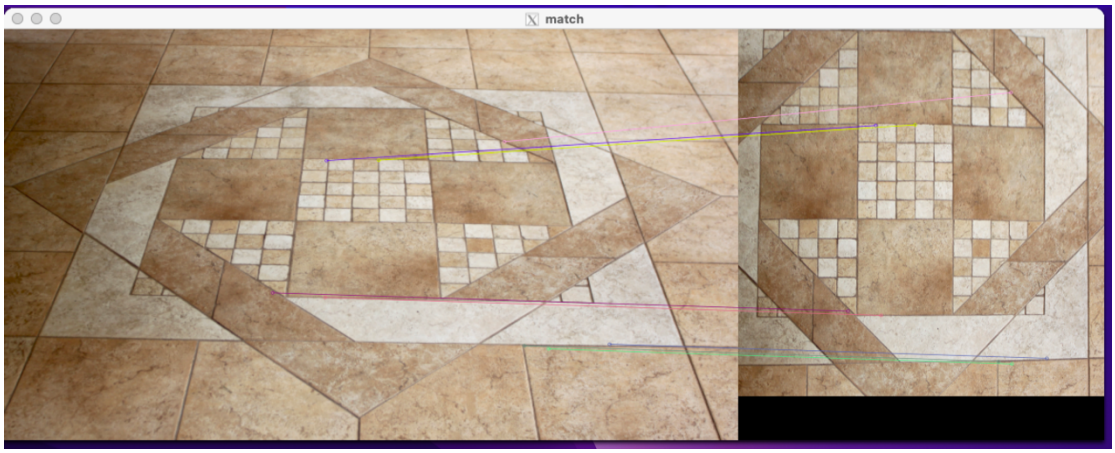



```
Chosen points in image1:
[[ 901.08184814  662.7623291 ]
 [ 649.21606445  593.78112793]
 [1015.01086426  566.66729736]
 [ 721.99084473  639.68774414]
 [1235.96264648  170.8475647 ]
 [ 663.90979004  759.64154053]
 [ 597.91271973  512.4664917 ]
 [1430.12084961  685.02209473]
 [ 703.31494141  785.1206665 ]
 [1132.6237793   656.28979492]
 [ 324.92495728  648.97821045]
 [ 987.26080322  405.96810913]
 [ 955.16021729  532.11676025]
 [ 328.19647217  358.10269165]
 [ 592.81170654  500.10784912]
 [1444.26049805  717.37335205]
 [1281.23449707  657.81903076]
 [1068.90197754  618.8447876 ]
 [ 427.5279541   644.98400879]
 [ 501.73800659  244.7421875 ]]
Chosen points in image2:
[[ 792.32409668  676.88543701]
 [ 480.35623169  588.61346436]
 [ 945.41094971  552.70892334]
 [ 568.17358398  646.72344971]
 [1267.09106445   25.19884491]
 [ 489.32406616  797.56860352]
 [ 421.9017334   485.29864502]
 [1487.03234863  707.9876709 ]
 [ 536.49676514  830.14440918]
 [1091.81604004  669.27502441]
 [  85.19613647  657.10717773]
 [ 920.14532471  344.05148315]
 [ 870.13897705  508.12762451]
 [ 102.2940979   294.31921387]
 [ 416.027771    469.76123047]
 [1502.89550781  751.02069092]
 [1288.2701416   671.92163086]
 [1011.69750977  620.10992432]
 [ 207.93588257  652.8168335 ]
 [ 318.00241089  146.50991821]]
Matrix H:
[[ 3.84231713e-03 -1.38845845e-04 -8.81372238e-01]
 [-1.20318586e-04  4.08310731e-03 -4.72377885e-01]
 [-1.96649751e-07  8.49225131e-08  3.26220059e-03]]
Matrix H with points normalization:
[[ 6.63642215e-01 -2.39890276e-02 -1.52224421e+02]
 [-2.07832860e-02  7.05230866e-01 -8.15859130e+01]
 [-3.39677616e-05  1.46607785e-05  5.63453098e-01]]
DLT error: 0.12137401883496164
Normalized DLT error: 0.1205854574196389
```

ImageA to ImageC with 4 correspondences:

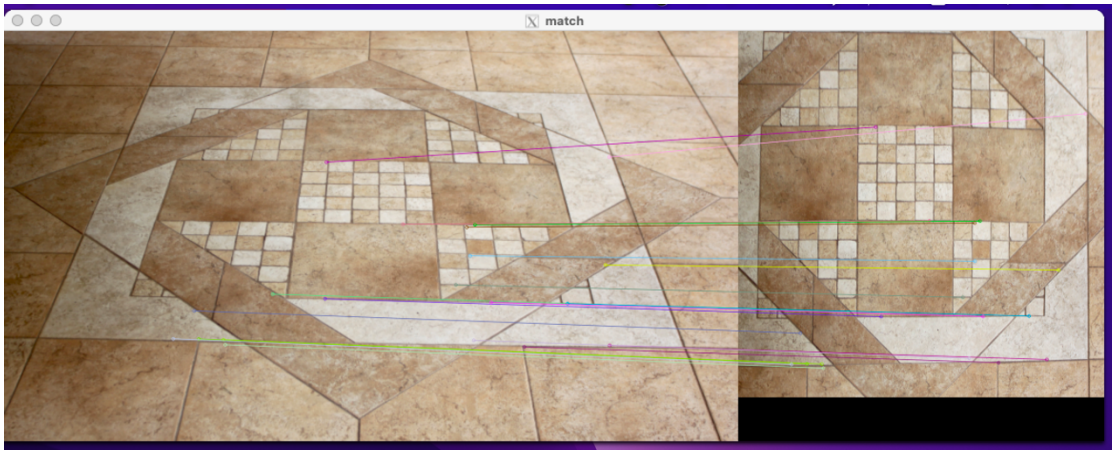

```
Chosen points in image1:
[[ 912.69470215  495.00030518]
 [ 895.94781494  373.03851318]
 [ 407.97912598  645.22753906]
 [1141.86999512  668.47357178]]
Chosen points in image2:
[[451.71792603 515.39569092]
 [446.80288696 353.56002808]
 [144.02682495 697.35791016]
 [574.19073486 700.01141357]]
Matrix H:
[[-1.76637048e-03 -6.43828127e-04  7.23274027e-01]
 [-8.81139860e-06 -4.16223933e-03  6.90543941e-01]
 [-1.31494740e-07 -1.72463524e-06 -1.69959167e-03]]
Matrix H with points normalization:
[[ 2.93054258e-01  1.06815969e-01 -1.19996646e+02]
 [ 1.46187785e-03  6.90547070e-01 -1.14566477e+02]
 [ 2.18159746e-05  2.86130065e-04  2.81975147e-01]]
DLT error: 4.003222436935783
Normalized DLT error: 4.003222436787387
```

ImageA to ImageC with 8 correspondences:




```
Chosen points in image1:
[[1048.98974609  235.13372803]
 [1269.8782959   658.6986084 ]
 [ 562.85437012  551.19067383]
 [ 675.83551025  274.6781311 ]
 [1141.86999512  668.47357178]
 [ 672.73461914  560.9407959 ]
 [ 785.53668213  273.89263916]
 [1090.94262695  661.92022705]]
Chosen points in image2:
[[571.92541504 132.34724426]
 [647.1940918  688.26647949]
 [230.18876648 589.43121338]
 [288.11401367 201.19953918]
 [574.19073486 700.01141357]
 [299.69616699 598.38458252]
 [370.89337158 199.80670166]
 [546.06085205 694.92907715]]
Matrix H:
[[ 1.77039648e-03  6.30002310e-04 -7.15540002e-01]
 [ 1.54738316e-05  4.17157783e-03 -6.98554681e-01]
 [ 1.33396062e-07  1.71226698e-06  1.71180085e-03]]
Matrix H with points normalization:
[[ 2.78721104e-01  9.91168917e-02 -1.12599074e+02]
 [ 2.38529894e-03  6.56773764e-01 -1.09924571e+02]
 [ 2.09371136e-05  2.69447345e-04  2.69663575e-01]]
DLT error: 3.5041918637209464
Normalized DLT error: 3.445579533116163
```

ImageA to ImageC with 20 correspondences:




```
Chosen points in image1:
[[1269.03710938  263.94390869]
 [ 398.29418945  586.2442627 ]
 [1090.94262695  661.92022705]
 [ 672.73461914  560.9407959 ]
 [ 562.85437012  551.19067383]
 [ 836.08514404  404.74911499]
 [1261.13720703  490.11593628]
 [ 947.23242188  531.92974854]
 [ 463.56973267  657.32147217]
 [ 407.97912598  645.22753906]
 [1181.60876465  571.10180664]
 [ 459.54742432  646.80114746]
 [ 987.26080322  405.96810913]
 [1269.8782959   658.6986084 ]
 [ 675.83551025  274.6781311 ]
 [ 354.22363281  645.0045166 ]
 [ 977.34118652  470.94390869]
 [ 970.21118164  411.14651489]
 [ 984.7677002   648.13238525]
 [1021.23345947  568.54534912]]
Chosen points in image2:
[[728.13812256 173.19007874]
 [130.07743835 633.13317871]
 [546.06085205 694.92907715]
 [299.69616699 598.38458252]
 [230.18876648 589.43121338]
 [400.93927002 397.47052002]
 [671.90649414 500.54681396]
 [471.17596436 557.91442871]
 [178.22705078 706.40484619]
 [144.02682495 697.35791016]
 [609.97802734 597.07983398]
 [175.05117798 697.78900146]
 [506.95843506 398.0987854 ]
 [647.1940918  688.26647949]
 [288.11401367 201.19953918]
 [111.28975677 698.49298096]
 [496.2204895  482.20440674]
 [495.67572021 402.36273193]
 [484.29406738 685.68023682]
 [513.42102051 598.07507324]]
Matrix H:
[[ 1.79096012e-03  6.35783582e-04 -7.16374706e-01]
 [-1.37036947e-05  4.27346448e-03 -6.97697798e-01]
 [ 9.44013446e-08  1.74040999e-06  1.78580298e-03]]
Matrix H with points normalization:
[[ 2.88565761e-01  1.01834730e-01 -1.14959499e+02]
 [-2.55198277e-03  6.88574696e-01 -1.11922311e+02]
 [ 1.46682156e-05  2.79410704e-04  2.89134287e-01]]
DLT error: 3.2122543593954513
Normalized DLT error: 2.876599785225214
```

Compare the errors:

ImageA to ImageB \ points	4	8	20
DLT	1.2819116002906004	0.15179095498888262	0.12137401883496164
Normalized DLT	1.281911600092404	0.15636756609172817	0.1205854574196389

ImageA to ImageC \ points	4	8	20
DLT	4.003222436935783	3.5041918637209464	3.2122543593954513
Normalized DLT	4.003222436787387	3.445579533116163	2.876599785225214

(Bonus):

My method:

1. Sample k pairs: 首先留下 good_matches 中 distance 前 50 高的 matches，再經由 RANSAC，每次從裡面取 k 個點，求出 H。對每個.npy 裡的 p_s，用 H 跟 p_s 與算出 p_t_hat，再算出 p_t_hat 與眞正的值 p_t 的 norm2 distance，當這個 distances 值小於 threshold = 1.5 時這個點爲 inlier，算所有 inlier 的數量，若大於目前 H 所得到的 inlier 數量則更新 H，共做 N=100000 次，最後 inlier 最多的極爲最佳解 H。
2. Normalize DLT: 先分別對 points1, points2 做 normalize 並得到 transformation matrix T, T'，用 normalized 過的 points1_norm, points2_norm 做 DLT 得到 H_hat，再照公式互相內積得到 H。
3. DLT: 用講義中的方法，用 k 個 matches 的點得到 A，再對 A 做 SVD，V 的最後一行(VH 的最後一列)爲 h，再將 h reshape 成 3X3 matrix。
4. Compute reprojection error with given GT pairs: 求出 p_t(眞實值)與 p_t_hat(估測值)的 norm2 distance，回傳 error_vector 供 RANSAC 判斷 inlier，error 是 error_vector 的 sum 爲最後判斷之 error 大小。

Screenshot: correspondences of other local features:

我加入了 ORB 來做不同的 feature matching

若 match_method 爲 ORB 則將 images 都轉成灰階，用 orb 抓取特徵點並 match 後，sort 取分數前 50 高的當作 good_matches。

```python
def get_correspondences(match_method, img1, img2):

    if match_method == 'sift':
        sift = cv.SIFT_create()
        kp1, des1 = sift.detectAndCompute(img1, None)
        kp2, des2 = sift.detectAndCompute(img2, None)

        matcher = cv.BFMatcher()
        matches = matcher.knnMatch(des1, des2, k=2)
        good_matches = []
        for m, n in matches:
            if m.distance < 0.75 * n.distance:
                good_matches.append(m)

        good_matches = sorted(good_matches, key=lambda x: x.distance)
        good_matches = good_matches[:50]

        points1 = np.array([kp1[m.queryIdx].pt for m in good_matches])
        points2 = np.array([kp2[m.trainIdx].pt for m in good_matches])

    elif match_method == 'orb':
        orb = cv.ORB_create()
        img1_gray = cv.cvtColor(img1, cv.COLOR_BGR2GRAY)
        img2_gray = cv.cvtColor(img2, cv.COLOR_BGR2GRAY)

        kp1, des1 = orb.detectAndCompute(img1_gray, None)
        kp2, des2 = orb.detectAndCompute(img2_gray, None)

        matcher = cv.BFMatcher(cv.NORM_HAMMING, crossCheck = True)
        matches = matcher.match(des1, des2)

        good_matches = sorted(matches, key = lambda x : x.distance)
        good_matches = good_matches[:50]

        points1 = np.array([kp1[m.queryIdx].pt for m in good_matches])
        points2 = np.array([kp2[m.trainIdx].pt for m in good_matches])

    return points1, points2, kp1, kp2, good_matches
```

下圖(上)爲 SIFT matches, (下)爲 ORB matches, 都是 imageA to imageB with 4 correspondences

Experimental comparisons:

上面 imageA to imageB with 4 correspondences 的 error table

imageA to imageB 4 correspondences	SIFT	ORB
DLT	0.6053000048203058	9.659821530185646
Normalized DLT	0.6053000040596236	9.659821530508513

這邊 SIFT 的 error 跟第一題的 error 不同，因爲 RANSAC 會隨機取不同的點，故 error 不同。

反覆測試多次之後發現，ORB 的 error 幾乎都大於 SIFT，不論是 DLT 或 Normalized DLT。

不過剛好選到很多 outliers 的話 SIFT 的 error 還是有可能比 ORB 還要高，

下表是其中一次 imageA to imageC with 4 correspondences 的結果。

```
Match method: sift
Chosen points in image1:
[[ 360.62591553  561.59747314]
 [ 781.77819824  370.82913208]
 [ 972.33239746  466.60058594]
 [1281.1940918   255.94462585]]
Chosen points in image2:
[[102.49594116 606.08148193]
 [368.51876831 353.91574097]
 [492.09860229 477.68658447]
 [739.72479248 167.16596985]]
Matrix H:
[[-2.21889211e-03 -5.43073493e-04  7.36670743e-01]
 [ 1.05377502e-04 -5.15190138e-03  6.76222768e-01]
 [ 5.83713085e-08 -1.65704930e-06 -2.68579744e-03]]
Matrix H with points normalization:
[[ 2.46599583e-01  6.03552090e-02 -8.18709019e+01]
 [-1.17112716e-02  5.72563546e-01 -7.51529343e+01]
 [-6.48717453e-06  1.84158421e-04  2.98489741e-01]]
DLT error: 22.33660289946204
Normalized DLT error: 22.33660289924257

Match method: orb
Chosen points in image1:
[[533.95202637 489.02404785]
 [592.         529.        ]
 [788.40002441 294.        ]
 [967.         409.        ]]
Chosen points in image2:
[[203.90402222 514.94403076]
 [246.         562.        ]
 [373.24804688 233.2800293 ]
 [493.         400.        ]]
Matrix H:
[[-1.78838883e-03 -6.07216823e-04  7.24608742e-01]
 [-3.04805583e-05 -4.09876556e-03  6.89143694e-01]
 [-2.04965463e-07 -1.65842305e-06 -1.66531867e-03]]
Matrix H with points normalization:
[[ 2.81217337e-01  9.54825342e-02 -1.13941967e+02]
 [ 4.79295177e-03  6.44515283e-01 -1.08365223e+02]
 [ 3.22300388e-05  2.60780712e-04  2.61865022e-01]]
DLT error: 9.5408143996285
Normalized DLT error: 9.54081439890445
```

Error table:

imageA to imageC 4 correspondences	SIFT	ORB
DLT	22.33660289946204	9.5408143996285
Normalized DLT	22.33660289924257	9.5408143989045

Discussion:

interesting finding:

1. 對 imageA to imageB 來說，DLT 與 normalized DLT 的 error 並沒有太大的差別，且互有高低，可能是因爲.npy 給的 100 個點數量不足。
2. imageA to imageB 的 outliers 相對 imageA to imageC 的 outliers 數量相對少，因爲後者的 error 浮動範圍更大，而前者 error 幾乎沒有太大的變化。

Difficulties I encountered:

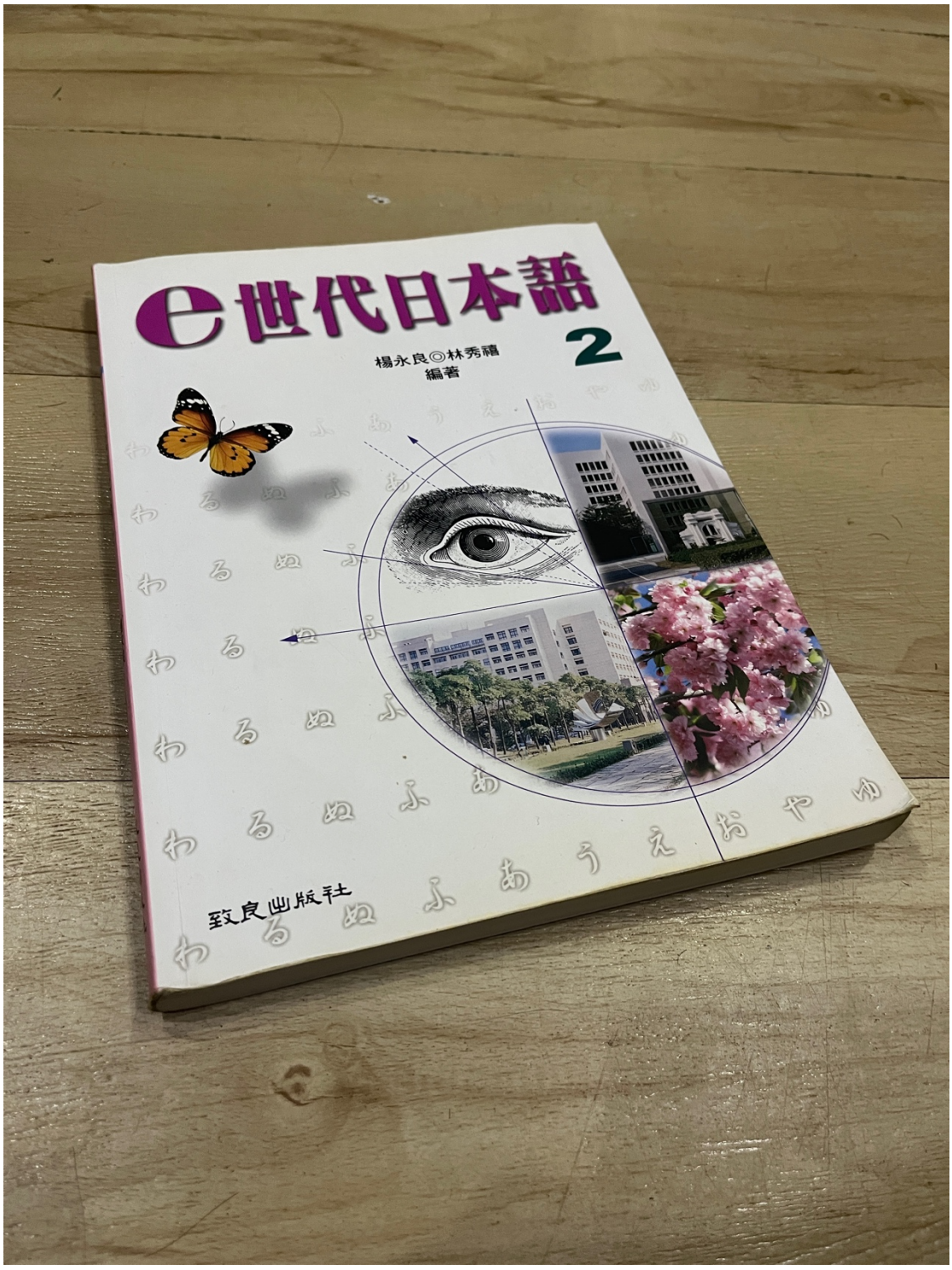
1. 在 imageA to imageC 中有許多 outliers，若選到這些 match，則 error 容易飆超高，所以我嘗試用 RANSAC 來解決這個問題。
2. 但用 RANSAC 會碰到另一個問題，就是 loop 的次數 N 若不夠大，則 error 結果起伏很大，可能剛好都選到好 matches 或剛好都選到壞 matches。欲使每次運算 error 值接近、則 N 必須設定一個相對大的數值，這使得運算時間大幅增加。

Insights I observe:

1. 給定的 matches 數越多，error 就越小，因爲給定 A 的方程式越多、h 就越準。
2. 照理來說，有做 points normalization 再做 DLT error 會較小。

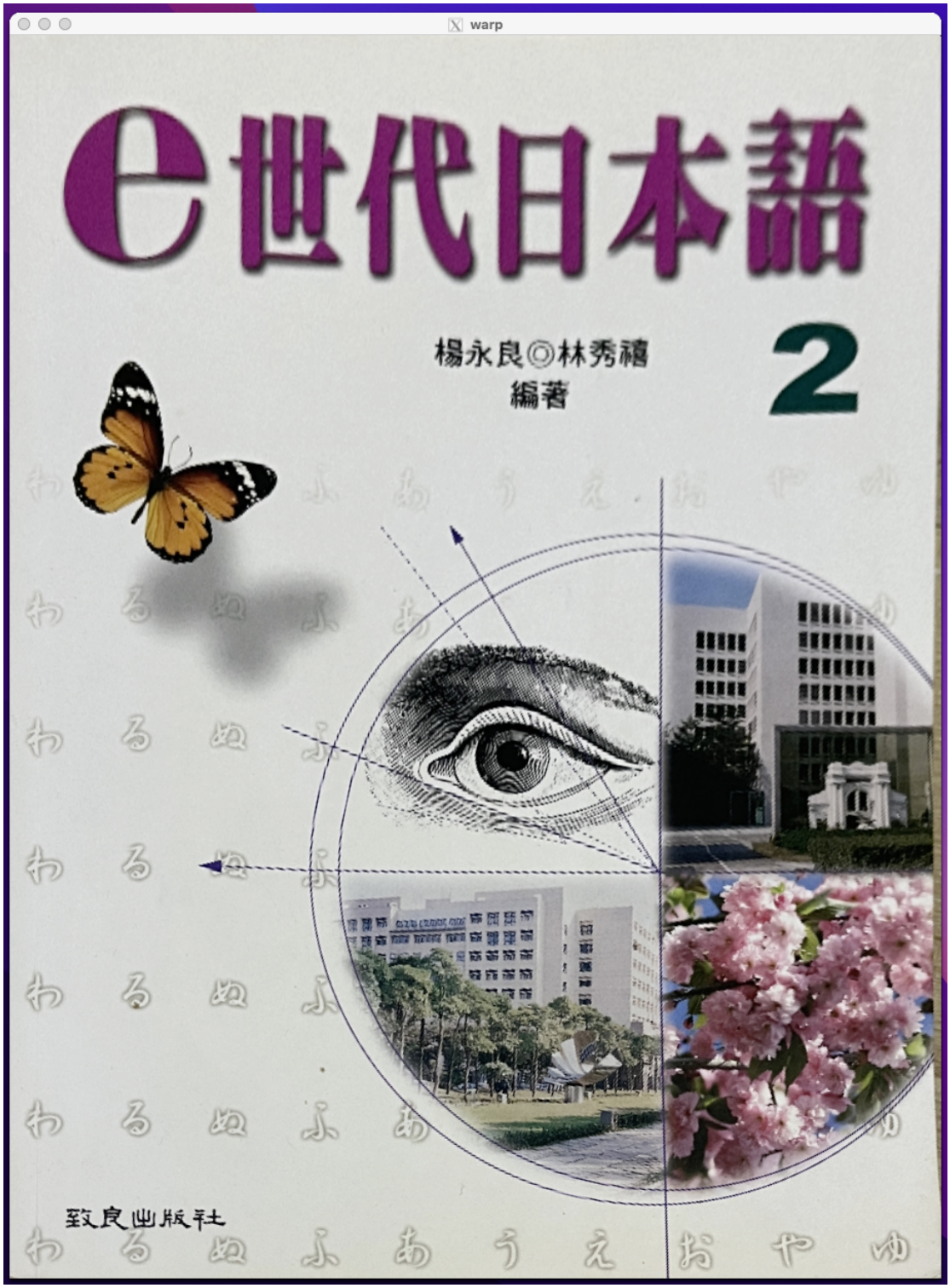
Problem 2: Document Rectification:

The input document image:



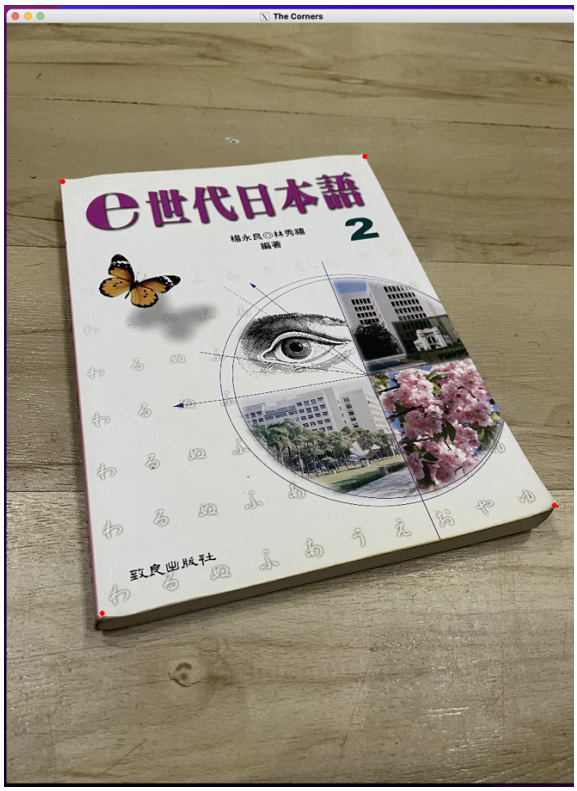
Rectified results:

右上糊糊的是因爲拍的時候對焦在下半部，可參考上一張圖的右上也糊糊的。

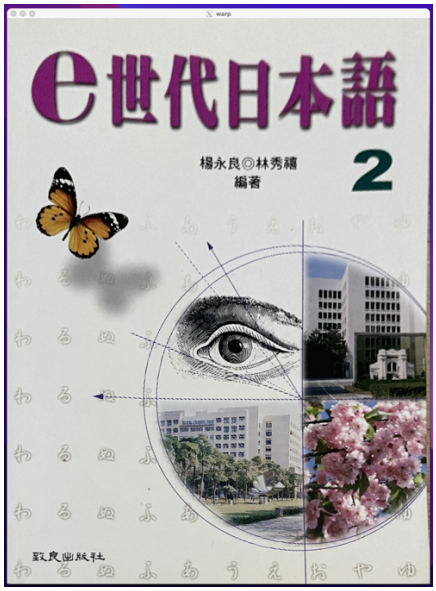


Briefly explain my method:

1. 首先將 image 長寬縮成 0.3 倍，因爲解析度太高圖太大張，會無法手動點出四個角落。
2. 手動點出四個角落，順序爲左上 → 右上 → 左下 → 右下。



3. 得到原圖四個角落的位置，並欲分別轉換到
   [0, 0], [0, width-1], [height-1, 0], [height-1, width-1]
4. 有四組對應點，做 normalized DLT 得 H。
5. 做 backward warping。將 inverse(H)對圖中每一點(i, j)做內積，可得到有小數的 x, y，用假想點(x, y)的旁邊四個鄰居做 bilinear interpolation 可得到(i, j)的值
6. 對圖中所有點都做完 backward warping 後，結果就是 rectified image。



How to execute:

Hardware:

1. CPU: AMD 2950X
2. GPU: Nvidia RTX 2080 Ti * 2 & RTX 1080 * 1

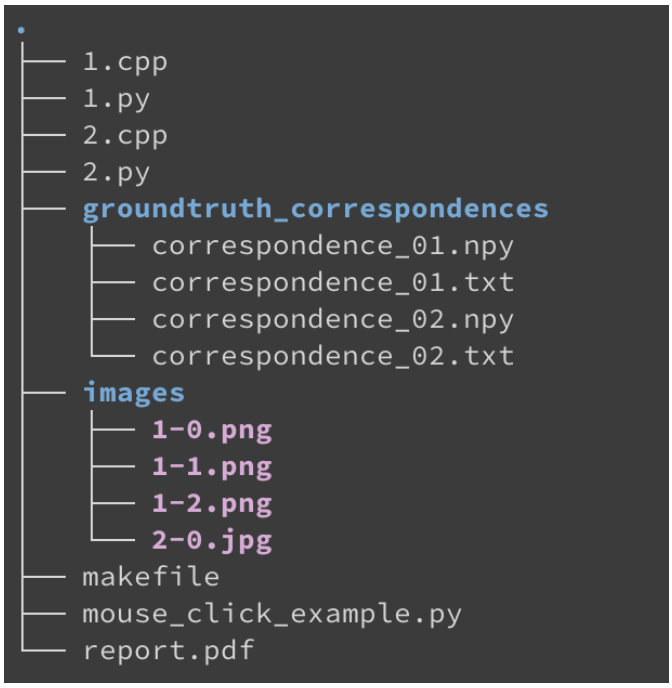
Environment:

Ubuntu 20.04 / Python 3.8.10

Modules:

1. OpenCV 4.5.1
2. Numpy 1.19.0

File structures:



1.cpp, 2.cpp, groundtruth_correspondences/*.txt, makefile, mouse_click_example.py
以上都沒用到

Problem1 command:

1-0 to 1-1:

python3 1.py images/1-0.png images/1-1.png groundtruth_correspondences/correspondence_01.npy

1-0 to 1-2:

python3 1.py images/1-0.png images/1-2.png groundtruth_correspondences/correspondence_02.npy

接著請輸入 k 值並耐心等待約 20-30 秒，因爲我 RANSAC 的 N 設定蠻大的，分別算出 SIFT 跟 ORB 的 error 後會一次畫出兩張 match 圖。

Problem2 command:

python3 2.py images/2-0.jpg

接著請點出四個角落，並等待結果。