

第14組 期末報告

組員:0611078張祐誠、0611017施君瀚、0616035林均翰

1. 使用的方法:

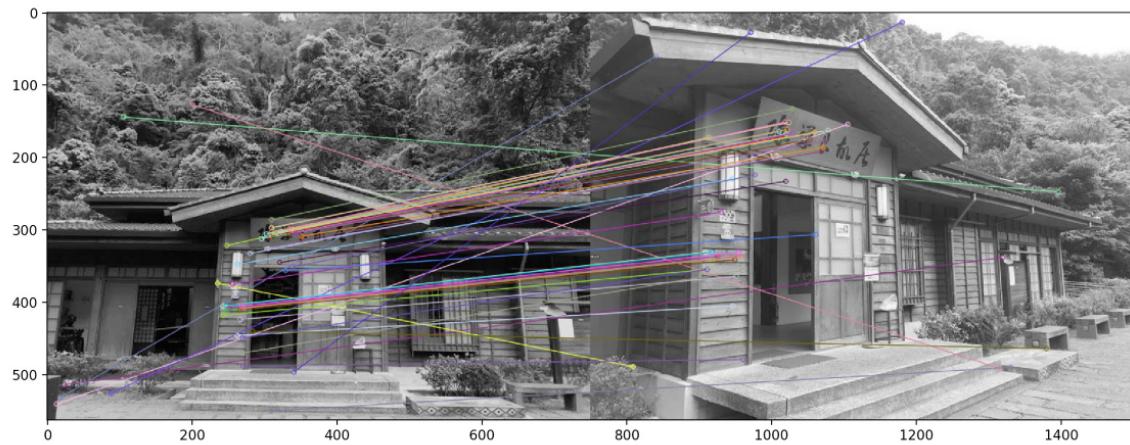
先把兩張要比對的圖，讀進來後轉成灰階影像；從opencv讀取要使用的特徵點擷取方法，即本次的SIFT, SURF, ORB三個方法。利用我們所選用的方法分別對兩張照片個別提取特徵點。

之後利用BFMatcher.knnMatch()把特徵點互相對應，接著透過D.Lowe提出的檢測方法，挑選distance比值小於0.75的對應點；最後把特徵點相連，顯示於結果上。我們使用的是張學良故居不同角度的照片做測試。

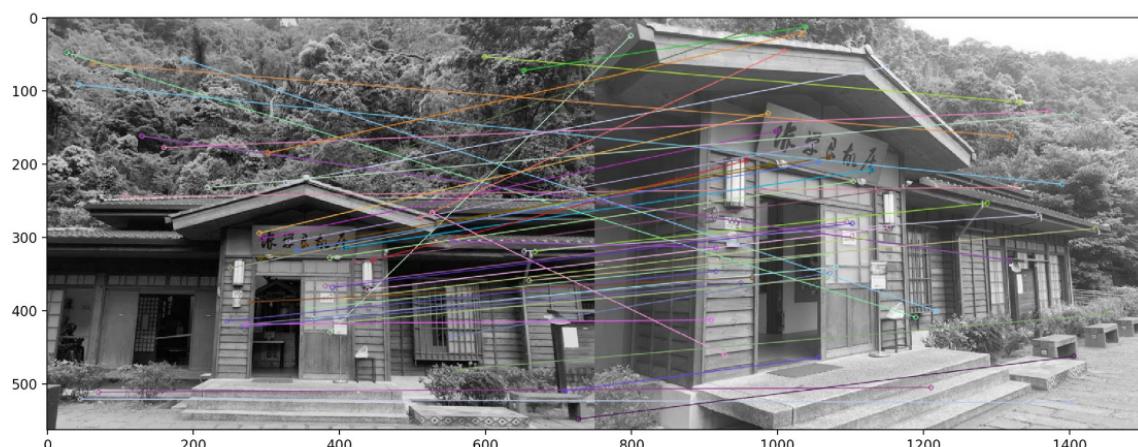
```
1 import numpy as np
2 import cv2
3 import matplotlib.pyplot as plt
4 import time
5
6 start = time.perf_counter() #timestart
7
8 img1 = cv2.imread('1.jpg',cv2.IMREAD_GRAYSCALE)           # queryImage
9 img2 = cv2.imread('2.jpg',cv2.IMREAD_GRAYSCALE)           # trainImage
10
11 # Initiate detector
12 #detector = cv2.xfeatures2d.SURF_create()
13 #detector = cv2.xfeatures2d.SIFT_create()
14 detector = cv2.ORB_create()
15
16 # find the keypoints and descriptors with SIFT
17 kp1, des1 = detector.detectAndCompute(img1,None)
18 kp2, des2 = detector.detectAndCompute(img2,None)
19
20
21 bf = cv2.BFMatcher()
22 matches = bf.knnMatch(des1,des2,k=2)
23
24 # Apply ratio test
25 good = []
26 for m,n in matches:
27     if m.distance < 0.75*n.distance:
28         good.append([m])
29
30 end = time.perf_counter()
31 print("Time spent by mothod of ORB:",end - start)
32
33 #cv.drawMatchesKnn expects list of lists as matches.
34 img3 = cv2.drawMatchesKnn(img1,kp1,img2,kp2,good[:51],None,flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
35
36 plt.imshow(img3),plt.show()
```

2.結果

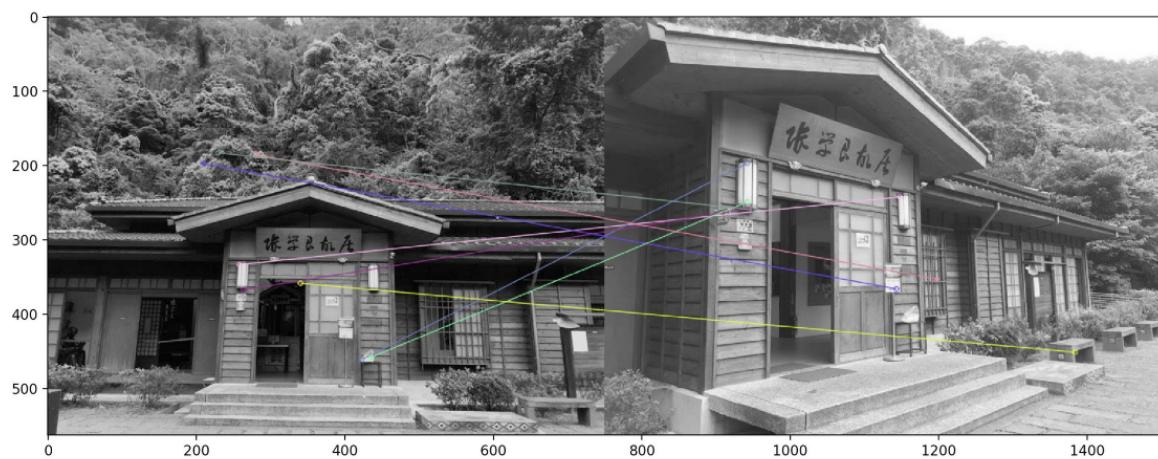
SIFT:



SURF:



ORB:



三種方法花費時間：

```
Time spent by mothod of SURF: 0.5504733489999999
(base) syje98@x86_64-apple-darwin13 lab11 % /opt
Time spent by mothod of SIFT: 0.743562758
(base) syje98@x86_64-apple-darwin13 lab11 % /opt
Time spent by mothod of ORB: 0.04126813600000001
```

3.三種方法的比較

Sift:

SIFT(Scale-invariant feature transform)

針對每個keypoint取 16×16 大小的像素，再平均切為 4×4 的cells，每個cells取梯度及角度值統計為8個bins的Histogram，總共16個cells會得到16個histogram (8bins)，可合併成 $16 \times 8 = 128$ 維度的資料，最後針對這些資料進行L2-Normalizing，即可得到代表該keypoint的Feature vectors(包含位置、尺度、旋轉不變數)。每個keypoint經過SIFT descriptor可得到128維的資料，是三種方法中最多資料量的，因此也是三者之中處理速度最久、效果也最好的一種特徵提取方法

在本次實驗結果中，也能看出吻合的特徵點數多於另外兩種，且精確度也明顯為三者之最，但相對的處理時間則為三者中最久的，約為Surf的1.3倍

SURF:

SURF (Speeded Up Robust Features, 加速穩健特徵)，這個算法可被用於計算機視覺任務，如物件識別和3D重構。SURF標準的版本比SIFT要快數倍，並且其作者聲稱在不同圖像變換方面比SIFT更加穩健。SURF使用海森矩陣的行列式值作特徵點偵測並用積分圖加速運算。

因為其使用的是方形濾波器(box filter)，所以可以通過積分圖(integral image)的方式來計算，所以相對於Sift的高斯濾波器(Gaussian filter)來說，Surf只需用積分圖的概念進行一加一減兩次單純的四則運算，比起高斯濾波器的加權平均算法，每次使用Filter都需要重新計算一遍，運算式更簡單快速。

但是也因為使用方形濾波器(box filter)的原因，只能達到高斯濾波器的近似值，所以在演算法表現上會比使用高斯濾波器的Sift來的差。

在我們的實驗結果中，Surf的花費的時間比Sift快了約26%，詳細的去比對特徵點可以發現，相較於Sift來說，Surf比對吻合的特徵點更少，而且更容易連接到不相關的物體上，特別是色差不大的樹林區域，可以看到許多不相關的特徵點比對。

ORB:

ORB(Oriented FAST and Rotated BRIEF), 從名字上可以理解到這個方法是建立在對FAST特徵點與BRIEF特徵描述子的一種結合與改進。在利用FAST特徵點檢測的方法來檢測特徵點，然後利用上課講述的Harris角點的度量方法，從FAST特徵點從挑選出Harris角點響應值最大的N個特徵點。接着透過計算每個特徵點的主方向，灰度質心法，計算特徵點圓形裡鄰域範圍內的灰度中心位置，定義出該特徵點的主方向。接下來旋轉每個特徵點的Patch到主方向，找出最優的256個點。

在我們的結果中可以明顯看到兩件事：第一是這個方法明顯快很多，時間約為SIFT的二十分之一；但我們也可以看到他的正確率相當不好。

從上述我們可以理解跟另外兩者的不同，ORB在尺度有變換時，效果就會非常差，正如同我們測試的照片裡，是透過不同的角度拍攝的；不過正式因為他對基於FAST方法對於尖角的偵測，加上改善的找尋主方向的方法，所以可以使他在偵測特徵點相當迅速。