

Project 2 Image-Stitching

1. Team Members

- 機械四 B06502052 蔡易儒
- 機械四 B06502011 陳姍苓

2. Task

- Combine a set of images into a larger image by registering, warping, resampling and blending them together.

3. Code

- Requirements

1. Warping

為了使照片接合效果更好、更自然，利用inverse warping將每張照片和每張照片的feature points投影到圓柱面，計算照片中每個點和feature points的圓柱座標，使用bilinear interpolation求出照片中每個點在圓柱面上的RGB值。

下列各圖為圓柱投影後的結果。



2. Detect feature

- **Harris corner detection**

首先我們使用sigma = 1將影像先做高斯模糊，接著用這張模糊的影像來找I_x與I_y，我們使用了兩種方法來計算I_x與I_y，一種是直接計算影像兩個方向的gradient，另一個則是使用sobel filter計算兩個方向的結果[1]。得到I_x與I_y之後我們就可以直接算得I_{xx}=I_x²、I_{xy} = I_x*I_y與I_{yy}=I_y²，接著就可以得到整張影像的corner response如下圖公式，而再下圖的R放大一點看可以看到有一些黑邊。接著設一個threshold，若corner response大於此threshold才將其視為可能的特徵點。而threshold我們設立一個變數threshold ratio來進行計算，threshold = max(corner response)*threshold ratio。

$$M(x, y) = \begin{bmatrix} S_{x^2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{y^2}(x, y) \end{bmatrix}$$
$$R = \det M - k(\text{trace} M)^2$$

- **Local maximum**

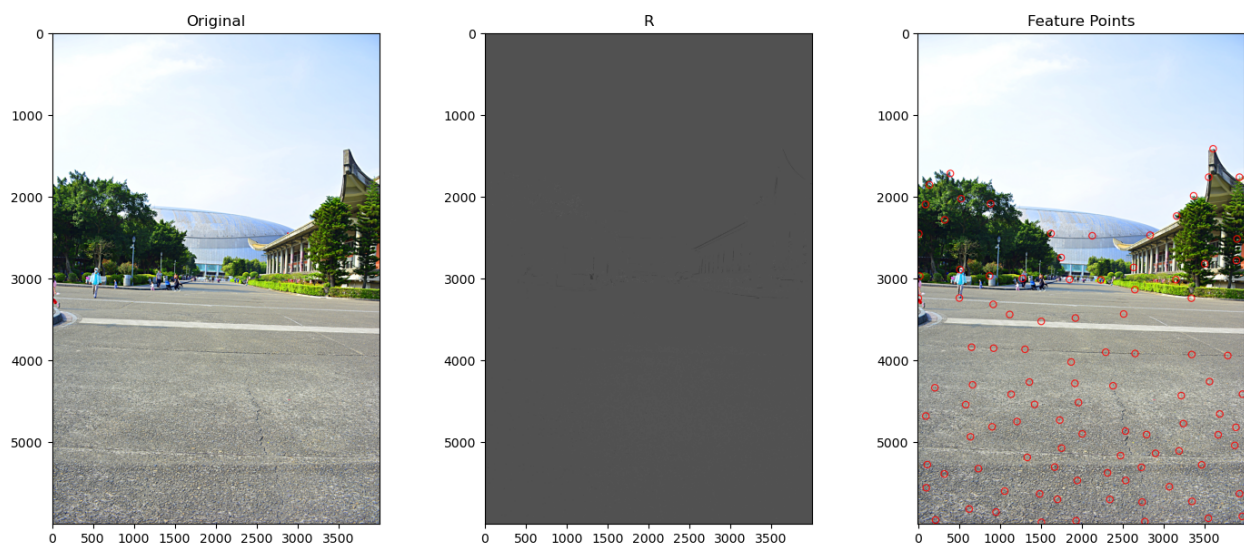
接著透過八個3*3的遮罩代表中心點往八個方向的filter如 [[-1,0,0],[0,1,0],[0,0,0]]、[[0,-1,0],[0,1,0],[0,0,0]]等，將corner response分別代入，並使用and確保往每個方向都會變小，如此就可以篩掉corner response雖然有過threshold但並不是local maximum的點。

- **Non-maximal suppression**

接著參考Multi-Image Matching using Multi-Scale Oriented Patches[2][3]，做Non-maximal suppression，先把corner response最大值的特徵點的radius設為極大值，接著對每一個特徵點找相鄰corner response乘上c_robust之後大於自己且與自己距離最小的特徵點距離，最後再將每個特徵點的radius做sort，取出suppression radius最大的num_feature個特徵點，其他的就捨棄掉不用，效果如下圖。

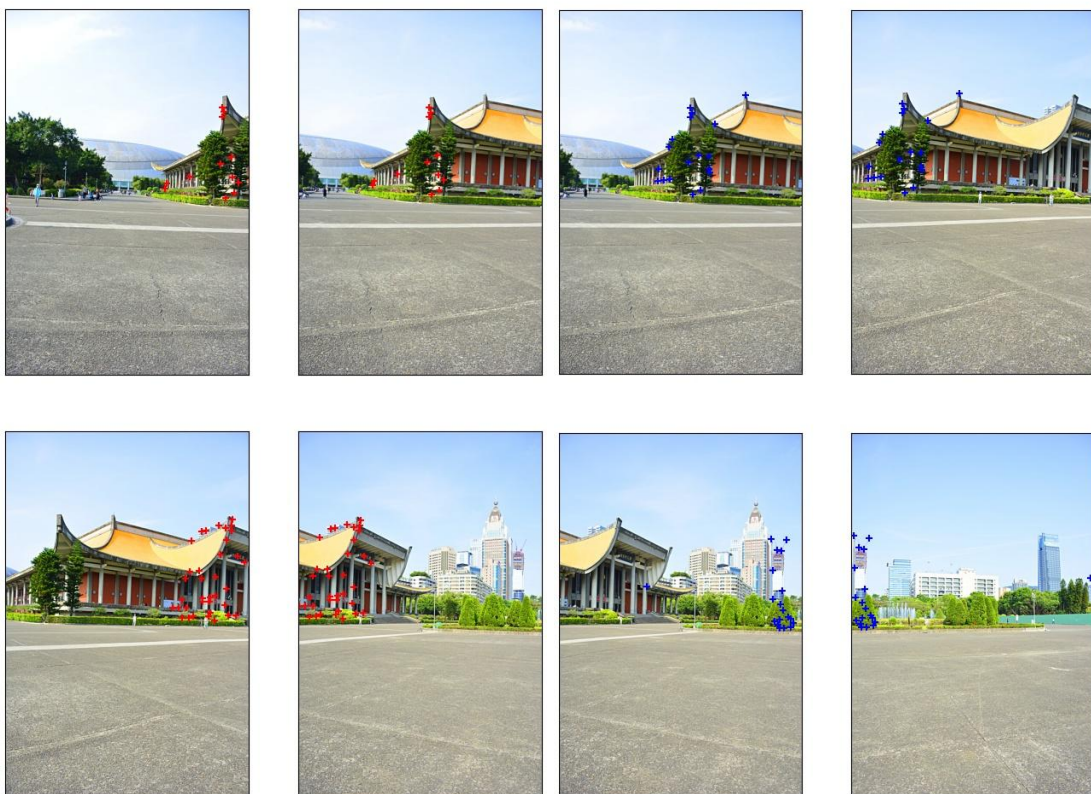
- **Feature description**

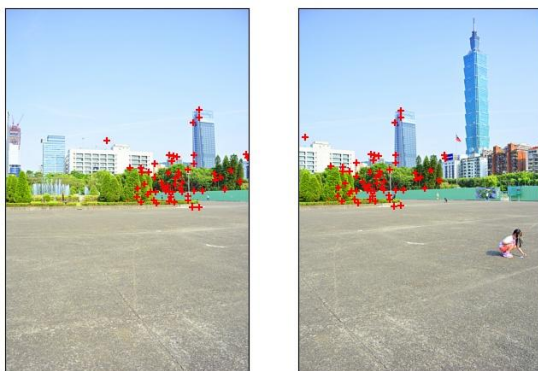
獲取特徵點之後，本組選取該特徵點往外擴張40個pixel的patch，因此每一個特徵點都會獲得一個80*80大小的patch，由於本次作業的影像來源是可控制的，因此本組固定來源影像之間不會有太多的自由度需要考慮，因此只會有translation，不會有角度的轉動，所以簡化了SIFT feature descriptor的步驟，直接存image block當作特徵點的描述。



3. Feature matching

計算兩圖中feature points之間的距離，利用暴力搜尋法找出兩張照片之間每個feature point相距最短的點作為最相似的對應點，並確保兩點間的距離與第二近的距離有一定比例的差距。下列各圖中的顏色標記即為對應的特徵點。





4. Image alignment

- **RANSAC**

從feature matching可知影像間需平移多少便能順利接合，為了避免matching結果錯誤、減少outliers的影響，使用RANSAC找出translation model的參數，經過多次遞迴找出兩點距離小於某threshold且滿足正確率為p、inliers數量最多的解作為最佳解。

- **Translation**

根據RANSAC結果可知影像間的平移關係，以此作為translation model的參數，將影像適當的平移以便之後的影像接合。

5. Blending

從上一步驟可知兩兩圖片間的位移關係，圖片接合時計算出每張照片對於第一張照片y方向的位移，以最小的位移(dy)作為基準，將每張照片平移dy對齊y=0，再各自平移y方向的位移。然後利用alpha blending處理兩兩照片間的重疊區域，根據像素在重疊區域的位置改變左右影像的權重，消除鬼影，使重疊區域的接合結果平滑而自然。

- Bonus

Recognizing panoramas (Feature matching)

(程式中的find_firstimg、bonus_align函式)

打亂圖片順序，得到特徵點後，計算每張圖(a)與剩餘所有圖(b1,...,bn)的對應特徵點數量，記錄對應特徵點在(b1,...,bn)右半邊的數量的最大值，由於第一張照片是在最左邊，其往左找的對應特徵點數量應為最小值，因此取各圖中往左對應特徵點數量的最小值，其對應的圖片即為第一張圖。

找到第一張圖後，便計算每張圖與第一張圖右半邊的對應特徵點數量，數量最多的即為第二張，接著再計算剩餘每張圖與第二張圖右半邊的對應特徵點數量，以此類推，就能找到正確的照片順序。

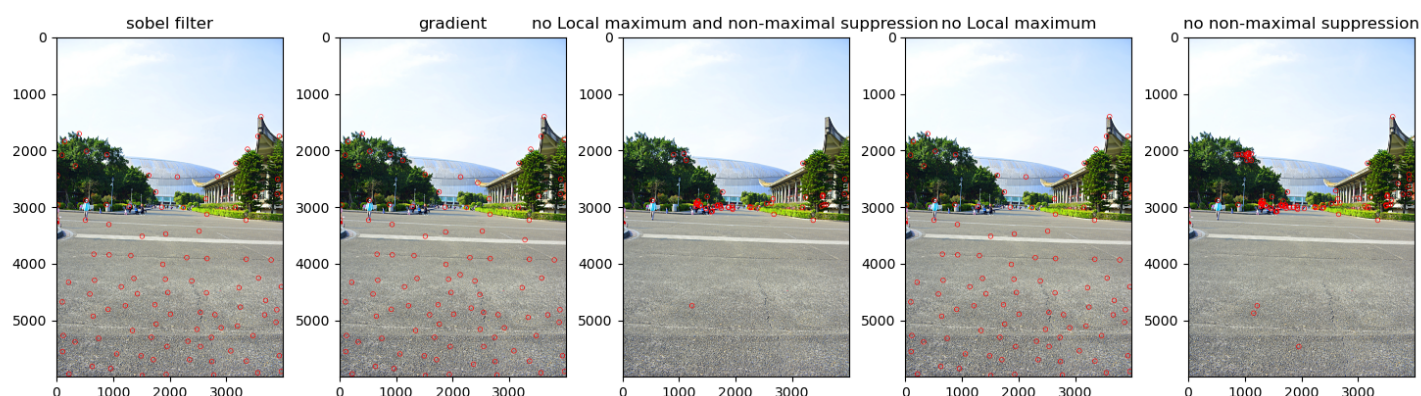
由於拍照時可能旋轉角度較小，照片之間的重疊區域多，因此程式可能會偵測第一張與第三張的對應特徵點數量大於第一張與第二張的對應特徵點數量，針對這樣的情形，我們讓程式捨棄第二張圖，因此最後接合圖片所用的照片數量可能少於原先輸入的照片數量。

4. Result



5. Parameter Discussion

- Sobel filter or gradient
- using non_maximal suppression
- using sobel or gradient to find features



左一與左二分別是使用sobel filter、gradient計算corner response。而右三張為得到corner response後有沒有濾掉不是local maximum及non-maximal suppression，從上圖可以明顯發現gradient與sobel filter都可以抓到還不錯的特徵點，比較明顯的特徵點都有抓到。而右邊local maximum很明顯可以過濾掉一些比較不好的corner response，但是仍無法避免許多特徵點都落在附近的區域，而使用non-maximal suppression後就可以很有效地把所有特徵點分開。

6. References

- [1] stackoverflow:
<https://stackoverflow.com/questions/52947492/sobel-filter-implementation-in-scipy>
- [2] <http://matthewalunbrown.com/papers/cvpr05.pdf>
- [3] <https://github.com/bchao1/Image-Stitching>
- [4] <https://github.com/JCly-rikiu/Image-stitching>
- [5] <https://github.com/qhan1028/Image-Stitching>

- [6] <https://blog.csdn.net/leonardohaig/article/details/104570965>
- [7] <https://www.programmersought.com/article/26154888679/>