

Image Stitching

資工三 B09902117 林榮盛

資工三 B09902108 石容居

作業內容

本次利用特徵點偵測和比對技術，並通過 stitching 和 blending 技術，嘗試將多張連續拍攝的圖片依據特徵點疊合，創造出一張 360 度環繞的圖片

實作項目

Harris corner detector

MSOP(detection/Descriptor/matching)

Image Stitch(cylindrical project/match/stitch/blend/crop)

Harris corner detector

1. Compute matrix:

首先將 image 轉成灰階並對整張 image 做 Gaussian blur，然後再算出 image 分別對 x 的偏微分 I_x 與對 y 的偏微分 I_y

$$I_x = G_\sigma^x \times I$$

$$I_y = G_\sigma^y \times I$$

然後算出對應的 matrix M

$$M = \begin{bmatrix} G_{\sigma'} I_x I_x & G_{\sigma'} I_x I_y \\ G_{\sigma'} I_x I_y & G_{\sigma'} I_y I_y \end{bmatrix}$$

2. Find corners:

算出每個 pixel 的 Response R

$$R = \det M - k(\text{trace } M)^2, \text{ where } 0.04 \leq k \leq 0.06$$

如果某個 pixel 的 R 大於 threshold 並且擁有周遭 8 格鄰居中最大的 R 則此 pixel 是 corner

3. Results:



Figure 1: 上圖的 threshold = 最大值的 0.5 倍，紅色十字為找出的 corner

MSOP(Multi-Scale Oriented Patches)

1. detection:

首先對轉為灰階的圖片做 rescale 和 Blur，其中第一層為原圖，每下層為前一層做 blur 後解析度縮 2 倍，如公式所示：

$$\begin{aligned} P_0(x, y) &= I(x, y) \\ P'_l(x, y) &= P_l(x, y) * g_{\sigma_p}(x, y) \\ P_{l+1}(x, y) &= P'_l(2x, 2y) \end{aligned}$$

其中 $\sigma_p = 1$ 接著對每一層做 harris corner detection

$$H_l(x, y) = \nabla_{\sigma_d} P_l(x, y) \nabla_{\sigma_d} P_l(x, y)^T * g_{\sigma_i}(x, y)$$

其中 $\sigma_i = 1.5$ ， $\sigma_d = 1$

MSOP 的 corner detection function 和 Harris corner detection 有些許不同，是以下算式得出：

$$f_{HM}(x, y) = \frac{\text{Det } H_l(x, y)}{\text{Tr } H_l(x, y)}$$

且我們會在此步驟額外計算 SIFT 的 Eliminating edge responses:

$$E_{HM}(x, y) = \frac{(Tr H_l(x, y))^2}{Det H_l(x, y)}$$

在計算 f 和 E 時，可能會遇到 divided by zero 的情況，因此會在分母加上 $\epsilon = 10^{-9}$

計算完 f 和 E 後，將取出 3×3 範圍內 f 最大值的點，並且該點須滿足下列條件：

$f > 10$ 且 $E < \frac{(r+1)^2}{r}$ ，此處 $r=10$

找出符合條件的點後，會進行 Sub-pixel refinement，做法和 SIFT 類似

最後進行 non maximal suppression，此步驟會先將所有點依 f 值由大至小做排序，只取前 r 比例 (此處 r 為 0.1) 的值做計算，再依據以下公式求出每一點的 radius:

$$r_i = \min_j |x_i - x_j|, \text{ s.t. } f(x_i) < c_{robust} f(x_j), x_j \in \mathfrak{L}$$

其中 $c_{robust} = 0.9$ ， \mathfrak{L} 為所有特徵點的 set

最終 radius 最大的前 700 個為找出的特徵點

2. descriptor

當找出特徵點位置後，會先將特徵點周圍 3×3 範圍做 Blur，再取出該點的 gradient，作為角度的特徵：

$$\begin{aligned} u_l(x, y) &= \nabla_{\sigma_o} P_l(x, y) \\ [\cos \theta, \sin \theta] &= u_l / |u_l| \end{aligned}$$

此處 $\sigma_o = 4.5$

求出角度後，先在原 rescale 後的圖上找出一正方形，其大小能包含以 (x, y) 做中心旋轉 $-\theta$ 後的 40×40 範圍正方形，找出後，將其依 (x, y) 為中心旋轉 $-\theta$ ，擷取 40×40 範圍，解析度調降 5 倍，獲得 8×8 正方形，將其依亮度做 normalization 後拉值為 64 維向量，加上其在原圖上的座標、倍率、 u_l 作為特徵點描述

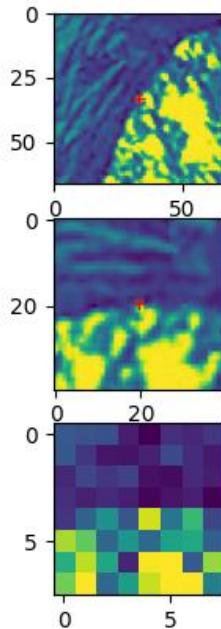


Figure 2: 上圖為旋轉前取出的正方形，中圖為旋轉後 40×40 的正方形，下圖為解析度降 5 倍的結果

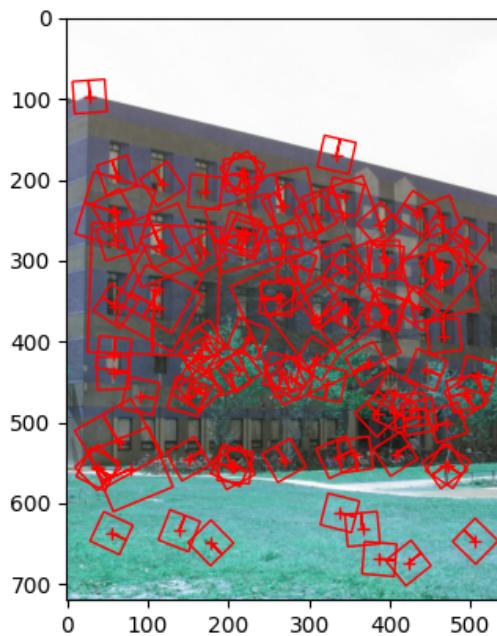


Figure 3: 最終結果，包含表示範圍和角度

3. matching

此處直接使用兩個向量的 euclidean distance 作為 matching 的標準，不過不考慮 u_l 和在原圖片的位置，我們假設相機不會做上下移動，因此會額外將兩特徵點的上下位置差納入考量，但權重不大

和 SIFT 相同，我們會將差距最小值除差距第二小值，若其值小於某比例（此處為 0.85），則判定找出特徵點配對

雖然我們分別實做了 Harris corner detector 和 MSOP，我們實作 stitching 時僅使用 MSOP 作為特徵點偵測和比對

Image Stitch

1. Cylindrical projecting:

以 inverse warping 的方式來做 cylindrical projecting

- (a) 對於每個在 projected image I' 中的座標 (x', y') 算出 (x, y)

$$x = f \times \tan\left(\frac{x'}{f}\right)$$

$$y = \sqrt{x^2 + f^2} \times \frac{y'}{f}$$

- (b) 取 x, y 相鄰的 4 個整數點，以對角的面積當權重做加權平均

$$\bar{x} = \lfloor x \rfloor$$

$$\bar{y} = \lfloor y \rfloor$$

$$a = x - \bar{x}$$

$$b = y - \bar{y}$$

$$\begin{aligned} I'(x', y') &= (1 - a)(1 - b) \times I(\bar{x}, \bar{y}) \\ &= a(1 - b) \times I(\bar{x} + 1, \bar{y}) \\ &= (1 - a)b \times I(\bar{x}, \bar{y} + 1) \\ &= ab \times I(\bar{x} + 1, \bar{y} + 1) \end{aligned}$$

2. Image match(Ransac):

重複做 N 次下列步驟

- (a) sample m matches:

對這 m 個 match pairs $((x_1^i, y_1^i), (x_2^i, y_2^i))$, $0 < i \leq m$ 算出平均的 shifts
 $[x_{\text{avg}}, y_{\text{avg}}]$

$$x_{\text{avg}} = \frac{\sum_{i=0}^m (x_2^i - x_1^i)}{m}$$

$$y_{\text{avg}} = \frac{\sum_{i=0}^m (y_2^i - y_1^i)}{m}$$

- (b) compute inliers:

對每個 matches $((x_1^i, y_1^i), (x_2^i, y_2^i))$, $0 < i \leq n$, number of matches = n 計算 distance d

$$d^i = \sqrt{(x_{\text{avg}} + x_1^i - x_2^i)^2 + (y_{\text{avg}} + y_1^i - y_2^i)^2}$$

如果 d^i 小於 threshold 則此 match 為 inliers

回傳 N 次中 inliers 最多的 average shift，此 average shift 為兩張 image 的最佳 shift

3. Image stitch:

- (a) 透過 image 之間的 shift 計算出每張 image 在 output image 中的位置，並記錄在 table 中。
- (b) 填 output image 的 pixel 時去 table 查此 pixel 有哪些 image 會提供，如果只有一張 image 會提供則直接填入，若有兩張以上則會做 blending

4. Image blending:

(a) Linear blending :

找出 image 重疊的邊界 l, r 再算出此 pixel 與 l, r 距離 l_d, r_d ，再透過 l, r 做加權平均

$$Pixel_{\text{output}} = \frac{Pixel_{\text{left}} \times r_d + Pixel_{\text{right}} \times l_d}{(l_d + r_d)}$$

(b) Linear blending with constant width:

與 linear blending 類似，只是將 l, r 改為兩張圖重疊的中心線左右各 N 個 pixel

- i. 先找出 image 重疊的中心點 mid ， $l = \text{mid} - N$ ， $r = \text{mid} + N$
- ii. 算出與 l 的距離 l_d ，與 r 的距離 r_d 然後與 linear blending 一樣做加權平均

(c) Results:



Figure 4: 上圖為 linear blending

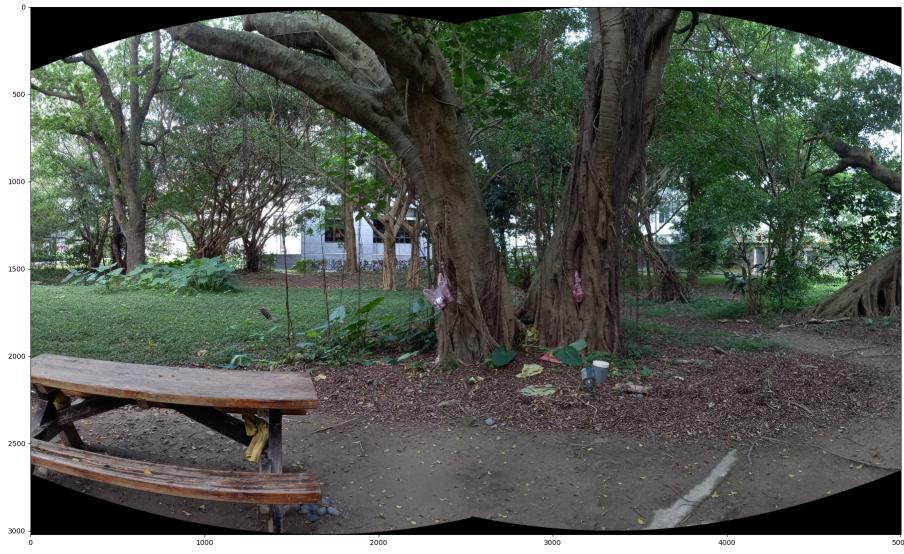


Figure 5: 上圖的爲 linear blending with constant width = 30 pixel

5. Crop :

重新算 focal length f ， y' 為 stitch image 最左或是最右邊兩端有色邊界的高度， y 為目標高度

$$x' = x = \frac{\text{image width}}{2}$$

$$f(y, y') = \sqrt{\frac{y'^2 x^2}{y^2 - y'^2}}$$

然後用新的 focal length 反向做 cylindrical projection 將 image 拉平，也就是對於 x', y' 找出 (x, y) ，用對角面積做加權平均，拉平後再做 crop 切掉黑色部分，這個作法是在做完 end-to-end alignment 後，或是 stitch image 的 drift 現象不明顯時才可以用



Figure 6: 上圖的爲 y 與 y' 的圖示

6. Result:

- (a) No cropping

(b) Cropping

以下為 MSOP 各參數討論
Eliminating edge 效果

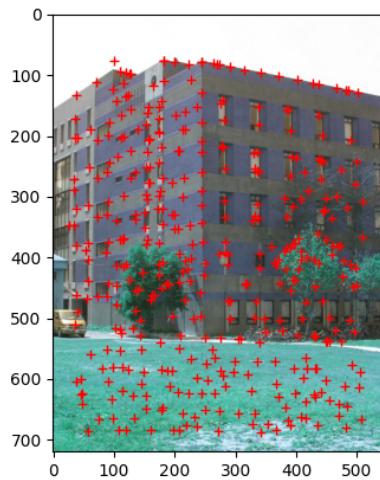


Figure 7: without edge elimination

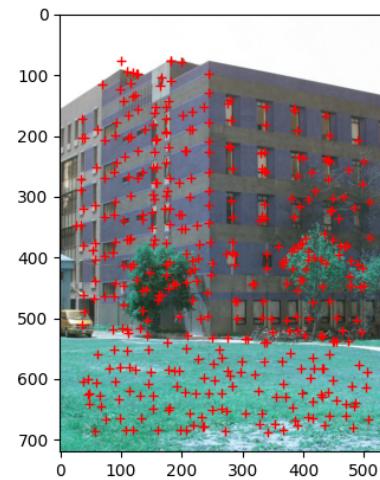


Figure 8: with edge elimination

可以看出在 edge elimination 的作用下，部分 edge 被消除
non maximal suppression 效果

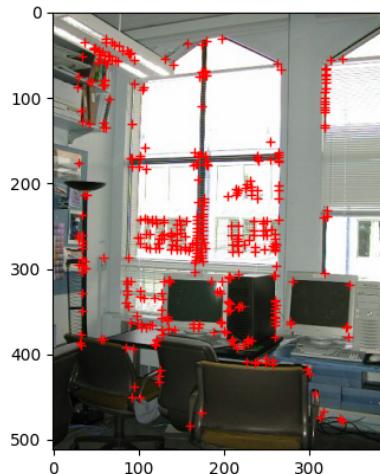


Figure 9: without non maximal suppression

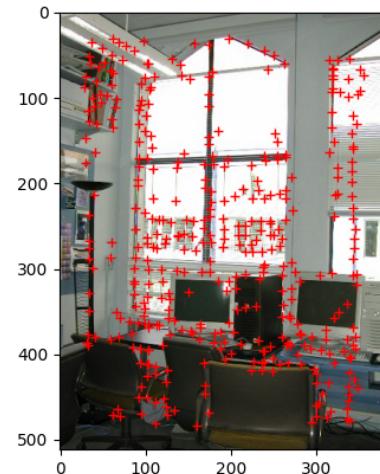


Figure 10: with non maximal suppression

可以看到，使用 non maximal suppression 的特徵點較為分散，不會過度集中於小範圍

只取前 r 比例的特徵點做 non maximal suppression 的影響

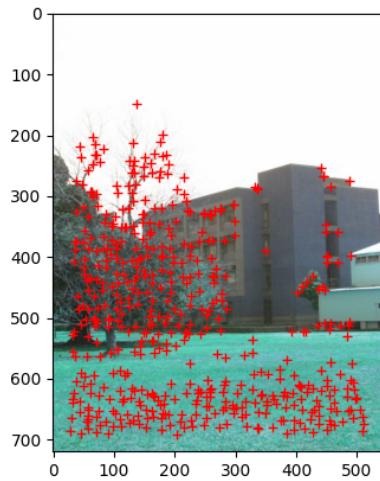


Figure 11: $r=0.5$

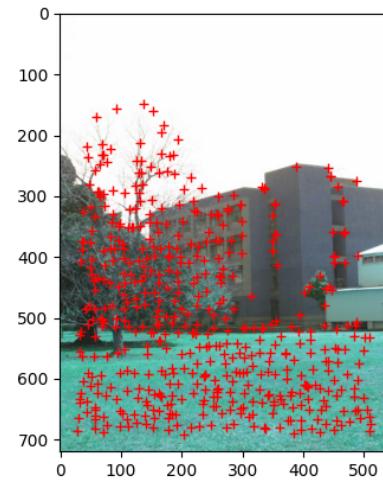


Figure 12: $r=1$

在 r 越高的情況下，特徵點會較為分散，但由於 non maximal suppression 為整段程式
的 bottleneck，因此在計算較大圖時，我們會把 $r=0.1$ ，犧牲部分效果以求更高的效率
matching 時最小值和第二小值的比例對 matching 的影響

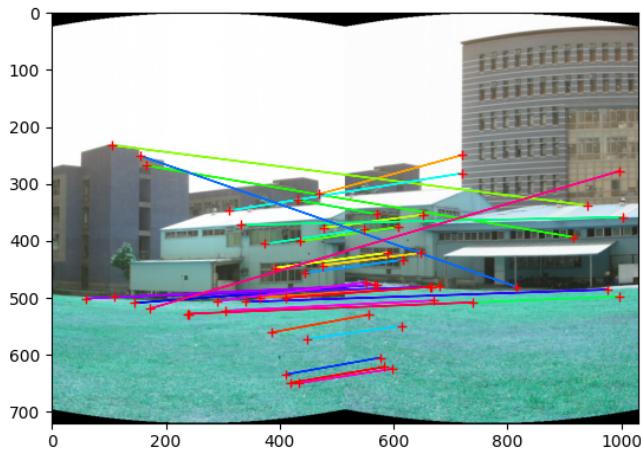


Figure 13: 比例為 0.9

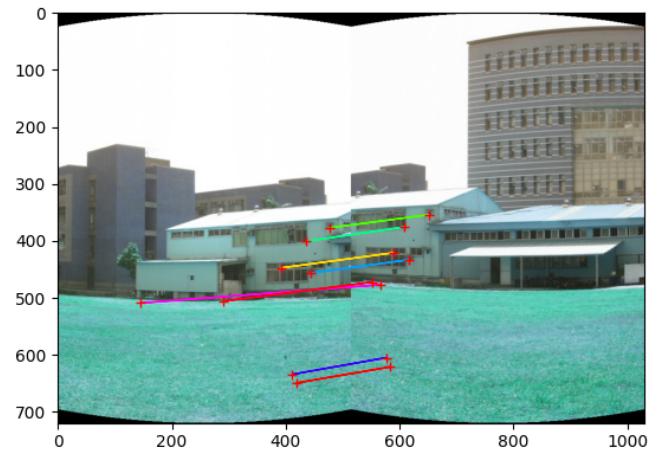


Figure 14: 比例為 0.7

可以看出當比例調低後，特徵點的對應會較為精準，但找出的對應點配對較少，可能
造成後續的 transition 無法找出最佳解或找出的解有誤差

將特徵點的上下距離差異加入考量

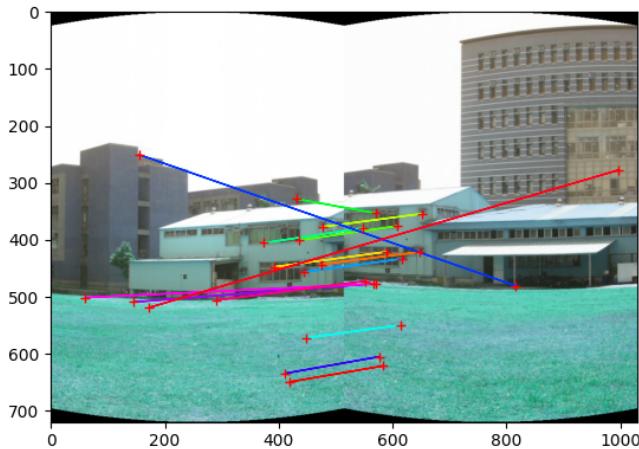


Figure 15: 無納入考量

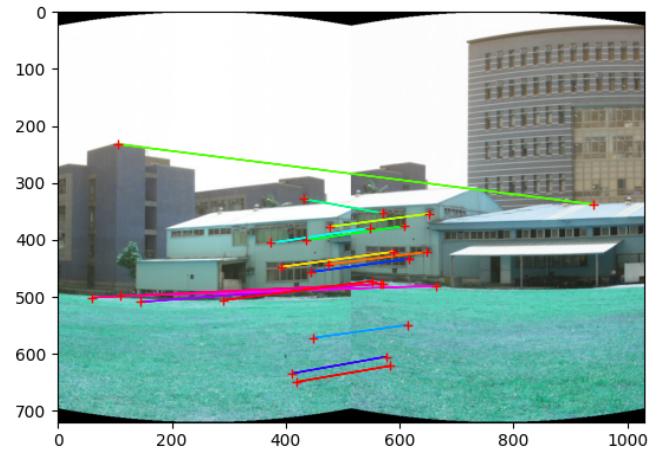


Figure 16: 有納入考量

將上下差距納入考量的理由，是因為我們發現在 matching 時會發生特徵點的對應明顯錯誤，而這個錯誤可藉由計算上下差距排除，但特徵點對仍然可能因拍攝角度和相機晃動等情況有小幅度的高低差異，因此會將其乘上較低權重計算，避免配對時僅取水平方向的對應。

在有納入上下差距為考量的情況下，可以發現其在一定程度上減少上下差距過大的對應錯誤情況，當圖片的像素越高時，其效果會相對明顯。

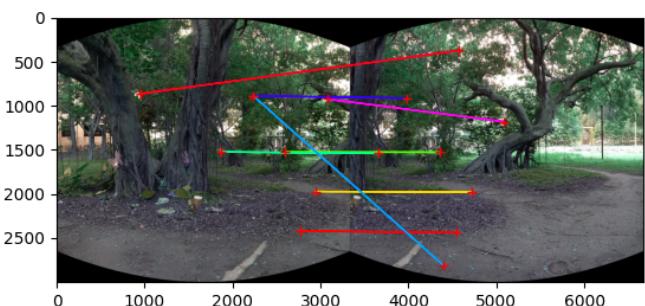


Figure 17: 像素較高，無納入考量

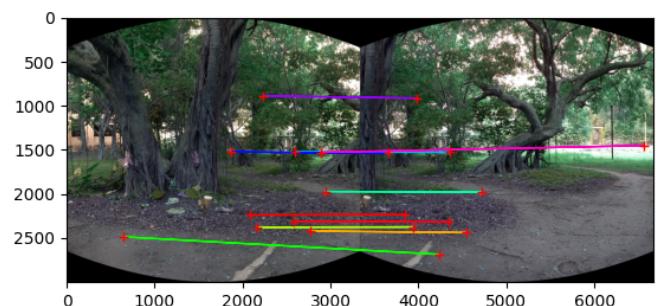


Figure 18: 像素較高，有納入考量

根據上述討論，我們可以因應不同需求調整不同參數，用以達到最佳效果，由於本次作業使用樹林作為環繞圖片，因此在特徵點對應上可能會較為困難（我們發現程式有機會選取樹葉或泥地草地作為特徵點，因此很容易和圖片中其他地點過於相像，理由可能是某些圖片僅拍攝到樹或泥地而沒有具有特徵的物體或人造建築，所以程式只能選取樹葉或泥地作為特徵點），我們最終選擇的參數如下：

有做 edge elimination

前 0.1 的特徵點進行 non maximal suppression(為求效率而做出的犧牲)

最大值和第二大值的比例為 0.85(若設定過小會導致某些圖片找不出對應)

將上下差距納入考量 (增加對應正確的機會)

這種設定是基於我們的拍攝圖片組的缺點而調整，因此可能會在其他圖片組的效果變差，套用其他圖片組前需再經過微調

reference

reference 1:[Multi-Image Matching using Multi-Scale Oriented Patches](#)

reference 2:<https://github.com/bchao1/image-stitching>

reference 3:<https://github.com/qa276390/image-stitching-msop>

reference 4:https://github.com/jpatts/image_matching

reference 5:<https://github.com/JCly-rikiu/Image-stitching>