

VFX 2018 Project #2

Image Stitching

蔡健華 R06944042 葉日豐 R06921074

1. Feature Detection:

Features又被稱作interest points或salient points，指的是一張image中具有代表性的那些點，而我們可以很輕易地藉由local information，判斷出一張照片裡的一個feature point和另外一張照片裡的一個feature point是同一個點。

這部分我們是實作Harris corner detector，因為老師課堂中提到做panorama並不需要使用到SIFT那麼複雜的演算法，而我們在固定相機拍攝的前提下，並不會使特徵點在scale下有變化，因此也不需要使用到multi-scale的harris corner detector。

Feature detection的演算法如下：

- 算出x和y方向上的derivatives得到Ix 與 Iy :
- 計算 $I_{x2} = I_x * I_x$, $I_{y2} = I_y * I_y$, $I_{xy} = I_x * I_y$
- 對 I_{x2} , I_{y2} , I_{xy} 做 Gaussian Blur得到各個pixel的和 S_{x2} , S_{y2} , S_{xy}
- 定義 $M = [[S_{x2}, S_{xy}], [S_{xy}, S_{y2}]]$
- $R = \det(M) - k(\text{trace}(M))^2$ ，為其corner detection function。
- $\det(M) = S_{x2} \cdot S_{y2} - S_{xy} \cdot S_{xy} - \text{trace}(M) = S_{x2} + S_{y2}$
- R越大的地方越可能是Feature Point
- 定義Threshold，取 $R > \text{Threshold}$ 的點為Feature Point

找出來以後，我還做了一些處理來減少 feature points 的數目：

- 以最高corner response的0.01倍作 threshold，只保留corner response 高於 threshold 值的feature point。
- Image 的邊緣除掉。
- 對於image的feature point 做 local suppression，用一個bounding box 去掃描該區域並只留下corner response最大的feature point。

2. Feature Descriptor:

在找完feature point的位置後，我們需要為每一個feature point derive “feature description”，這些feature description可以說是feature point的unique特色，讓電腦得以根據這些feature description，決定不同張圖裡的兩個feature point到底是不是同一個點。在這次實作的是 MSOP feature descriptor，也就是將Bias/gain normalized sampling of local patch 做為我們的feature descriptor，具體作法如下：

- 根據feature detection 所找到的feature points的位置，以每個feature point為中心，框出一個40x40的patch - 對每個40x40的patch做Gaussian blur，即右式： $P_l(x, y) * g_{2 \times \sigma_p}(x, y)$
- 取得8x8 oriented patch sampled at 5 x scale
- 對pixel 的亮度做Bias/gain normalization : $I' = (I - \mu)/\sigma$
- 使其mean 為0，variance為1。
- subtract mean可以使其具有 shift invariance的特性；而divided by sigma則 可以使其具有 scale invariance的特性。
- 經由上面的作法可以取得64維的feature description
- 另外，為了方便取得每個feature description對應到的feature point，我將feature point的位置存在feature descriptor的第65和66維

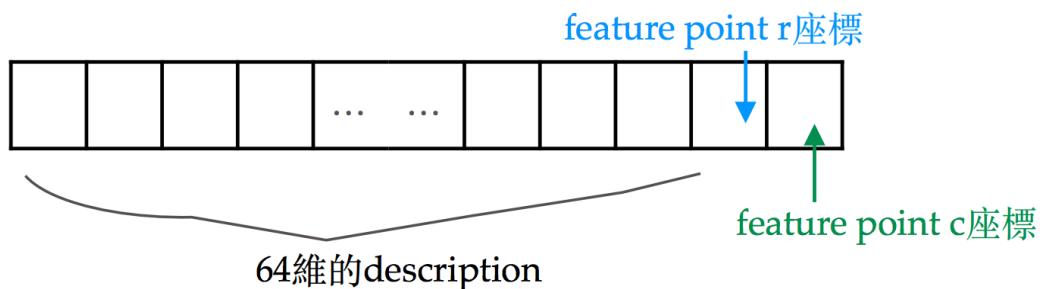


圖1. Descriptor 的格式

3. Feature Matching:

計算好每個feature point的descriptor後，我們便可以根據descriptor將不同image的feature point進行matching。這個部分，我們實作了Exhaustive Knn Search，Exhaustive Knn Search主要是計算兩張input image的所有feature points的descriptor間的L2 distance，將這些distance進行ascending sort，取得距離最小的前幾名作為我們的matched feature points。不過這樣的做法比較慢，複雜度為 $O(n^2)$ 。Matching結果如下：

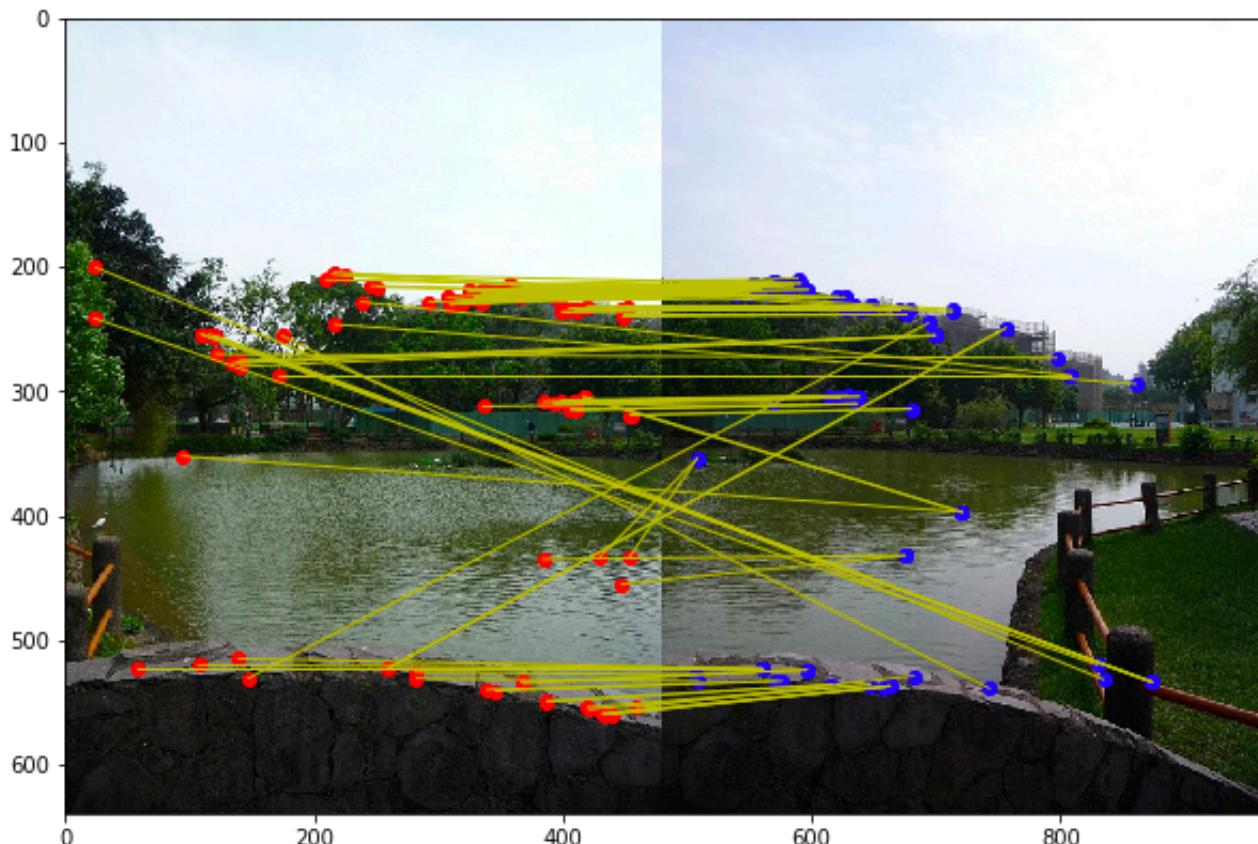


圖2. Feature matching後的結果

4. RANSAC/Image Stitching:

得到每張圖的matched feature pair後，就可以用RANSAC來找出一張圖要多少的位移量才能跟下一張圖做stitching。

RANSAC algorithm

DigiVFX

Run k times: ← How many times?
(1) draw n samples randomly ← How big?
 Smaller is better
(2) fit parameters Θ with these n samples
(3) for each of other $N-n$ points, calculate
 its distance to the fitted model, count the
 number of inlier points c
Output Θ with the largest c ← How to define?
 Depends on the problem.

圖3. RANSAC演算法

k 取300次，每次一組選feature pair，找出這組pair的兩個feature之間的x, y位移量，用該位移量去fit其他每一組pair，若誤差在threshold以內就記為inlier。300次後，找出inlier最多的那一組x,y位移，利用他來做image stitching。

Image stitching只要用剛剛計算出來的位移量去apply到該張圖，跟下一張圖疊在一起就可以了，重點是他們疊在一起的地方，會因為兩張相片的亮度色彩等因素造成有接縫的現象，這裡就要用到下面講的blending來解決。

5. Blending:

兩張image在得到對齊的關係後，因為有些區域會互相重疊，所以還需要透過 blending 的方式才能夠生成完整的影像，我們在這邊實作了 linear blending：

- 透過影像邊界遞減的方式來互相疊加，也就是交界區域越靠近誰，誰的影像像素乘以的權重就越大，示意圖如下：

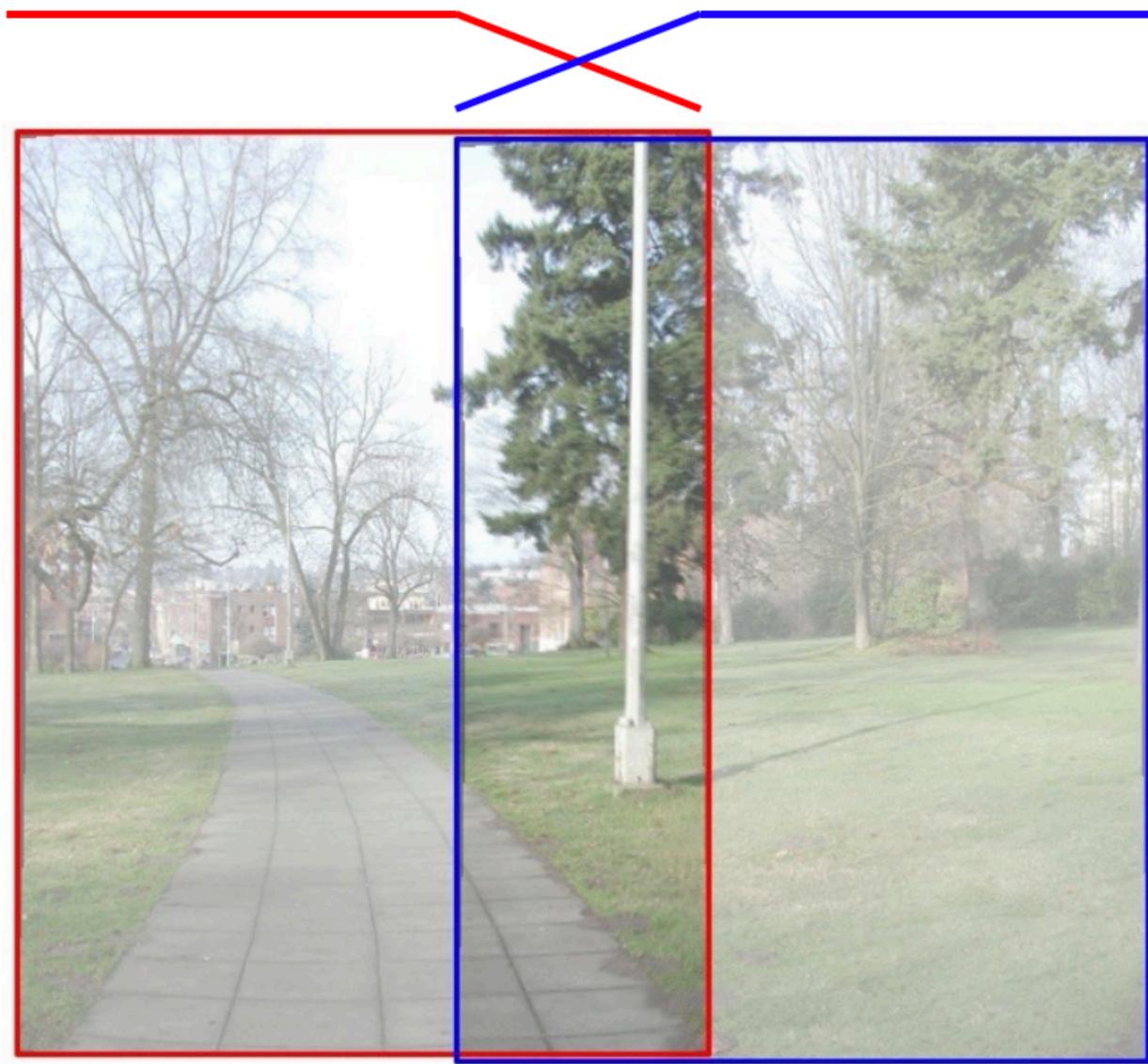


圖4. Linear Blending 示意圖

我們的實作方式是為重疊區域建立兩個linear mask，在組合圖的時候，兩張照片重疊的區域分別乘上這個mask，便可以得到blending的結果。但是，linear blending有一個很大缺點，那就是其重疊區的中間會因為兩張影像的權重差不多，所以，如果兩張image對齊的結果不是那麼完美的話，可能會有鬼影產生。

6. Drift:

由於誤差會逐漸疊加的關係，經過image stitch產生的panorama容易會出現逐漸偏移的現象，稱為Drift。如下圖，圖片變得傾斜，後來我嘗試了兩種簡單的方法來修正，第一種是計算出原始影像的高度 h ，和panorama的高度 H ，計算出斜率 $m=(H-h)/x$ ，然後用inverse warping的方式將傾斜的panorama warp成高度為 h 的影像。還有我嘗試了第二種做法，將panorama的高度 H 減掉原始高度 h ，得出 y 的總偏移量，除以影像張數得到 y 的平均偏移量，最後在stitch的時候，將每張影像的 y 軸位移減掉平均偏移量，就可以讓最終的panorama變成方正。最後在clip掉上下的黑邊即可。

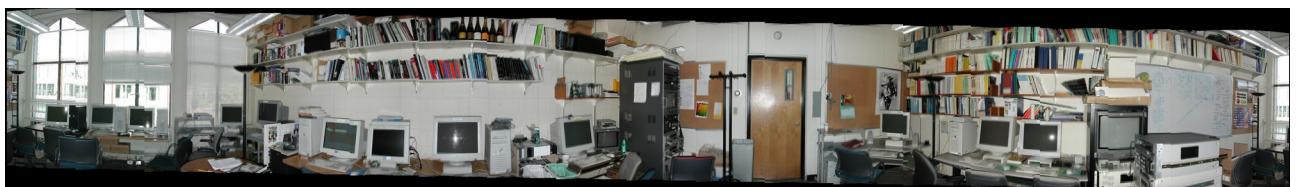


圖5. 未處理drift的panorama



圖6. 使用方法一修正偏移，但仍可明顯看出會有y軸的誤差



圖7. 方法二加上clip，修正偏移且y軸誤差較不明顯

7. Artifacts:

