

Image Stitching

Team 35 蔡政諺 郭蕙綺

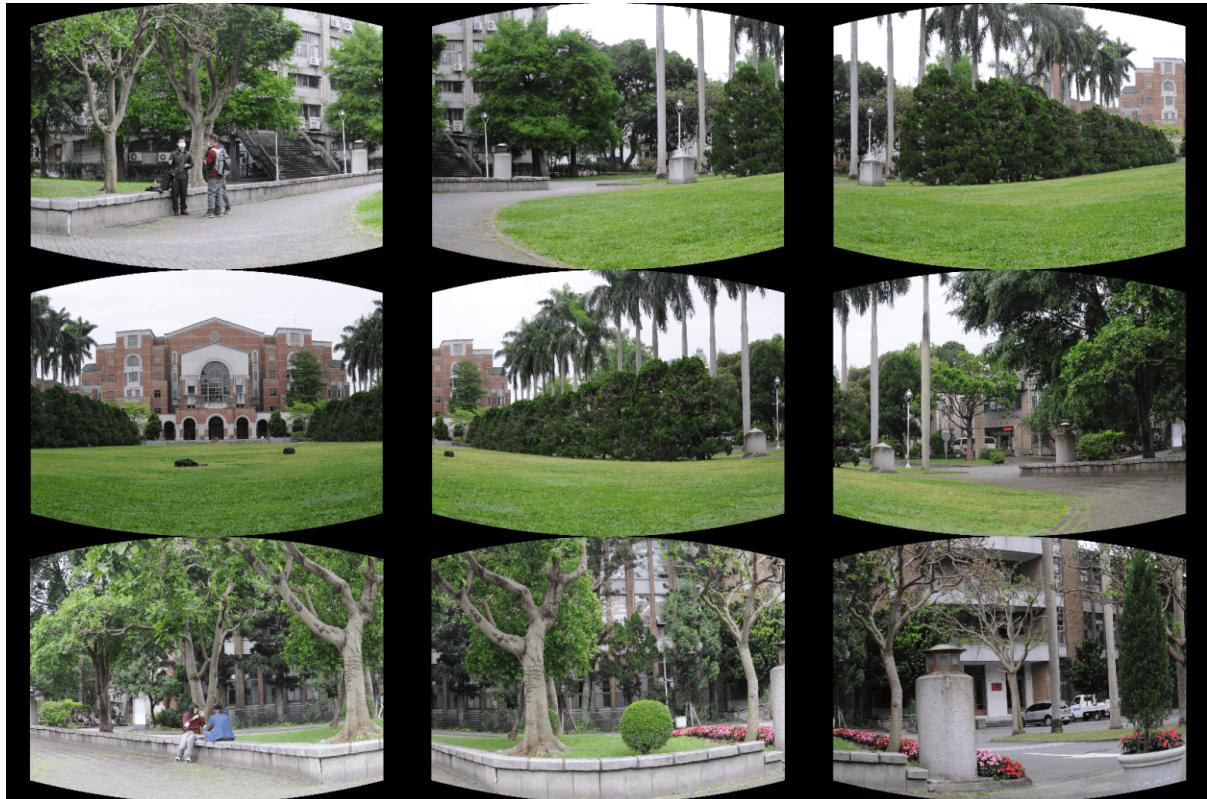
1. Taking Photographs

我們使用助教提供的 Nikon D90, 定點拍攝 9 張不同水平視角的照片。



2. Cylindrical Projection

我們根據上課講義的公式將照片 warp 到 cylindrical coordinate, 其中 focal length 是使用者作業網頁提供的 autostitch 套件所 output 的結果。下圖是做完 cylindrical projection 的結果。

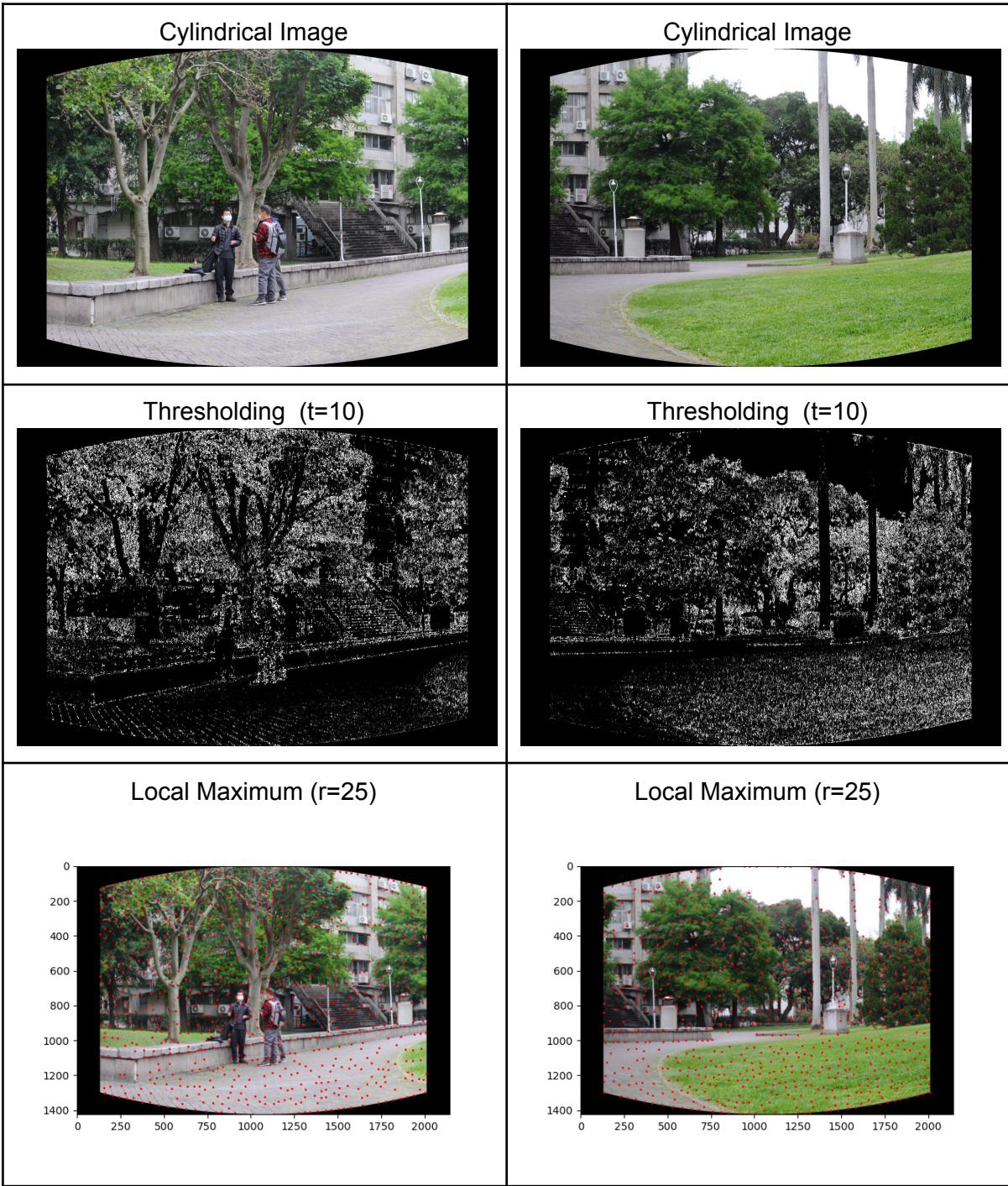


3. Feature Matching

Harris Corner Detector

首先要算出影像中的特徵點，我們實作的是 Harris Corner Detector，它是 Moravec Corner Detector 的延伸，概念是若以某個點為中心的 local window 經過小位移後，其亮度變化非常劇烈，那個點就是一個 corner。不同於 Moravec Corner Detector 只能考慮八個方向的位移，Harris Corner Detector 透過 Taylor expansion 考慮所有的位移，並且用 Gaussian window function 取代 binary window function。

根據講義，Harris Corner Detector 可以分成以下步驟：首先對原始影像做 Gaussian blur，以免在算梯度時受到雜訊干擾，再對 blurred image 取得 x 方向與 y 方向的梯度圖，稱為 I_x 與 I_y 。接著用 I_x 與 I_y 做 element-wise dot product, I_x 乘 I_x 得到 I_{x^2} , I_y 乘 I_y 得到 I_{y^2} , I_x 乘 I_y 得到 I_{xy} 。接著再對 I_{x^2} , I_{y^2} , I_{xy} 做 Gaussian blur，得到 S_{x^2} , S_{y^2} , S_{xy} 。對於每個點 (x, y) ，定義矩陣 $M = [[S_{x^2}(x, y), S_{xy}(x, y)], [S_{xy}(x, y), S_{y^2}(x, y)]]$ ，計算該點的 corner response $R = \det(M) - k(\text{trace}(M))^2$ ， R 愈大代表該點愈像是 corner，這裡我們透過 threshold 過濾 R 值太小的點，再留下 local maximum，做為影像的特徵點。



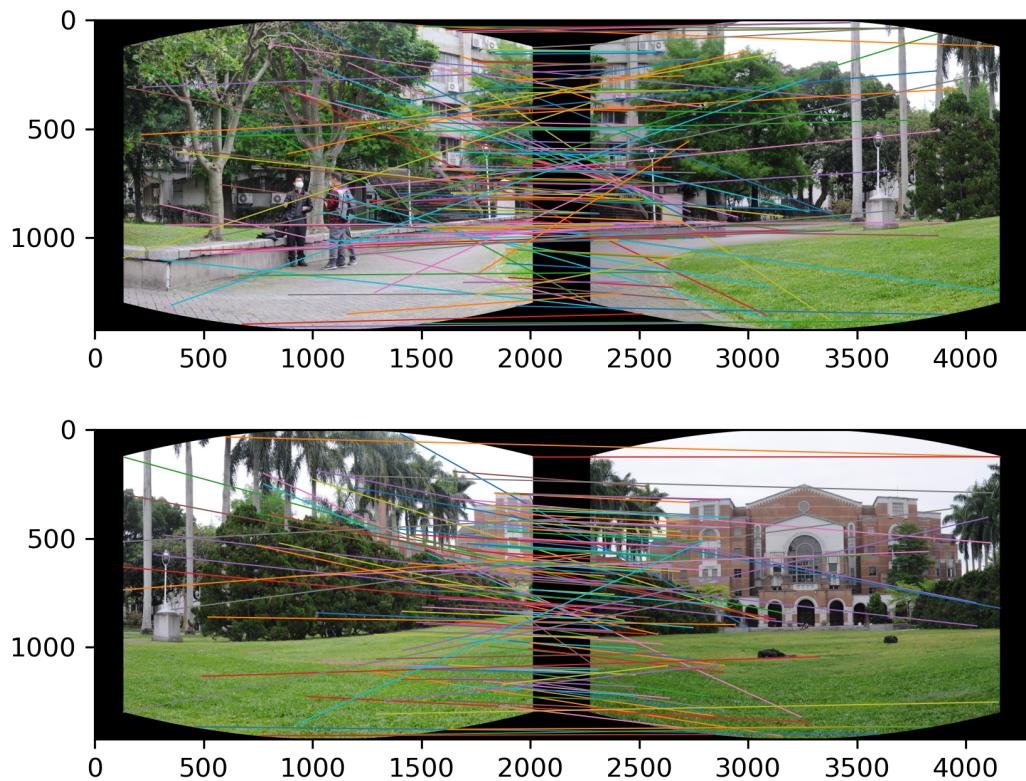
Feature Description

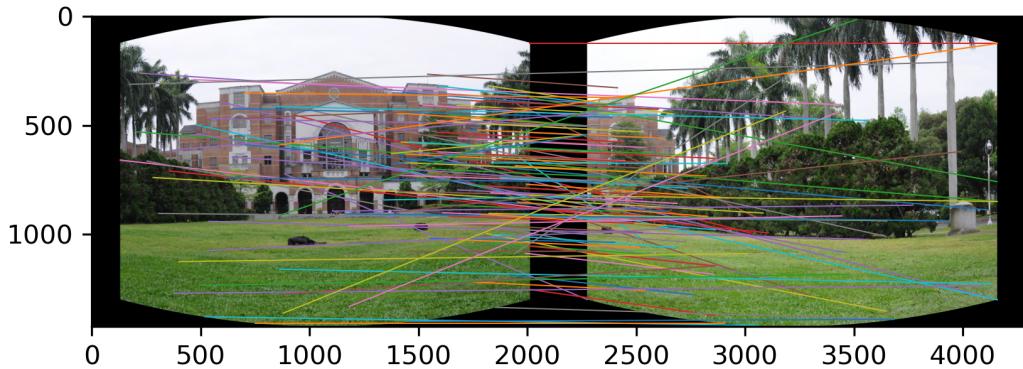
算出特徵點後，接著要將每個特徵點的內容描述成一個高維向量，這裡我們實作 SIFT Descriptor。首先將影像中每一個點的梯度大小 m 與角度 θ 算出來。計算 θ 時，由於 \arctan 函數沒有考慮 x 方向的正負，值域均落在 -90 度至 90 度，因此我將 x 方向梯度小於 0 的 θ 加上 180 度，讓 x 方向梯度小於 0 的 θ 值域落在 90 度至 270 度，再將所有 θ 加上 360 並取 360 的餘數，讓值域落在 0 度至 360 度。有了每個點的角度後，再根據 bin 的數量，將每個點的 orientation 算出。接著，對於每一個特徵點，我們取一個 16×16 的 patch，每 4×4 為單位算一個 histogram，記錄每個 orientation 的梯度值總和。由於作業的情

境是水平影像的 panorama, 不會發生 rotation, 故我們省略計算 major orientation 的步驟, 直接取正方形的 patch。此外, 若 patch 取到影像外的位置, 我們會當作梯度為 0 去處理。由於我們定義 bin 的數量為 8, 所以每個 histogram 會是一個 8 維向量, 這樣的 histogram 共有 16 個, 故最終會得到一個 128 維的向量表達一個特徵點。得到向量後, 根據上課講義, 我們對他做 normalization, 將大於 0.2 的值 clip 為 0.2, 再做一次 normalization, 便得到該特徵點的 descriptor。

Feature Matching

有了每個特徵點的描述向量後, 就可以將兩張影像的特徵點 match。假設兩張影像的特徵點分別是 k_1 與 k_2 , 對於 k_1 中的每個特徵點 $k_{1,i}$, 我們計算 $k_{1,i}$ 與 k_2 中的每個特徵點的 L2 distance, 取 k_2 中距離 $k_{1,i}$ 最近的特徵點與第二近的特徵點 $k_{2,1}, k_{2,2}$, 兩者距離比例若小於 0.8, 表示 $k_{2,1}$ 足夠像, 且比其他特徵點都還要像 $k_{1,i}$ 得多, 我們便將其視作 match。下方展示幾個我們的影像經過 feature matching 的結果, 圖中的連線代表兩者的特徵點發生 match。可以發現大部分的 matched pair 呈現相同的趨勢, 而少數會是 outlier, 最適合的位移將透過 RANSAC 決定。





4. Image Stitching

RANSAC

由於每個 matched pairs 的 shift 不一定相同，我們使用上課教的 RANSAC algorithm (圖左) 去找到最佳的 shift，根據講義內容去計算 k 值 (圖右)，($P=0.99, p=0.7, n=1$) k 約莫等於 4，有點太小結果不太理想，加上 matched pair 數量不多，所以直接將 k 設為 matched pair 的數量，最後 output 出最佳 shift。

RANSAC algorithm

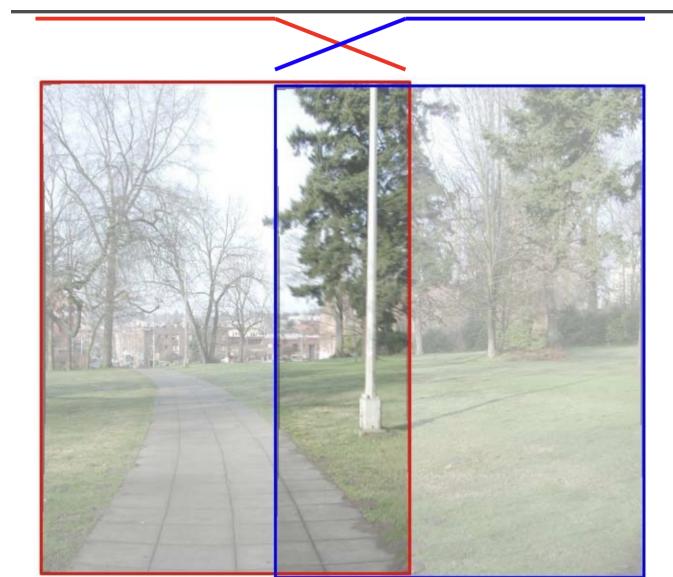
DigiVFX

- 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.

$$k = \frac{\log(1 - P)}{\log(1 - p^n)}$$

Stitching and blending

最後要根據前面找出的 best shift 做 stitching，在接合兩張圖片時，重疊的部分需要做 blending，讓整張圖片的亮度在接縫不要有太明顯的差異，這邊我們採用上課提到的 linear blending (如圖)，但由於對整個重疊的部分做 blending 需要很久的時間，所以採用當重疊 width 超過 600 時，只對最中間 600 的圖片做 blending (在 width 重疊正中間位置的左右各延伸 300 做 blending)，也就是 blending 的寬度只會小於等於 600，最後輸出 panorama image。



5. Result

Library



Library (cropped)



Palm



Palm (cropped)

