

VFX HW2

學生：R06522106 陳馨慈、R06942039 何明倩

1. Taking photos with different angle

- 配備：Panasonic /LUMIX 數位單眼相機，型號 DC GF9，鏡頭為 LUMIX G VARIO 系列 H-FS12032，焦距 12-32mm，光圈 3.5-5.6。
- 地點：化學教室、台大草坪
- 拍攝條件：用三腳架固定相機，轉動相機(固定 optical center)
- 場景
 - 化學教室





2. Multi-Scale Harris

a. Multi-Scale Harris corner detection

BlurPyramid.m : 產生五個 level 的 pyramid image,由上一個 level 的 image 用 $\sigma = 1$ 做 gaussian blur , 再用 factor = 2 做 downsampling 。

CDF.m : 算出一張 image 的 corner detection function 。

FPLo.m : 算出任何一個 level 下 feature point 的 location , 用 3*3 的 window 做篩選,並將 corner detection function 的 threshold 設為 4000(為了控制 feature point 的數目不要過多)。

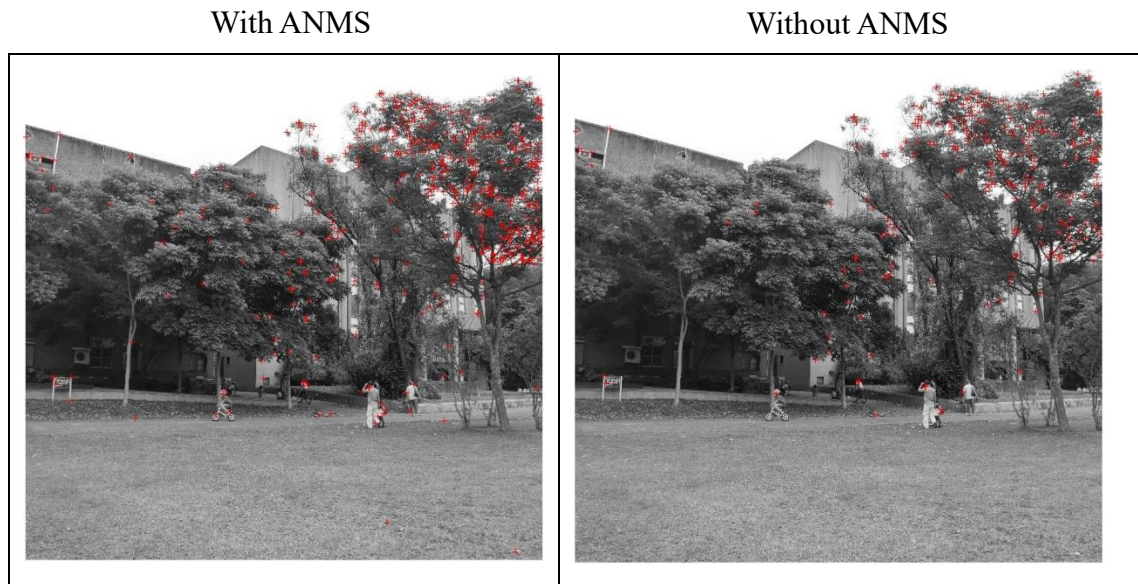
AllFpLo.m : 將每個 level 中算出的 feature point location 合併,儲存的位置以最高階析度(level 0)的 image 為基準,同時亦儲存每個 feature point 是哪一層 level 貢獻的,此步驟是為了後面的 matching 與 descriptor 。

b. Adaptive non-maximal suppression

ANMS.m : 得到所有 feature point 的位置後,欲取其中 250 個 feature points 做後續步驟,而此步驟是希望使取出的 250 個 feature point 在影像上能均勻分布。首先取整張影像上的 global maximum,以此 global maximum 的位置為圓心,畫一半徑化圓,使圓涵蓋整張影像,逐漸縮小半徑,並在圓以外的部分搜尋第二個 maximum,當找到下一個 maximum 時紀錄當前的縮小半徑,並以當前找到的兩個 maximum 為圓新開紀錄半徑下的圓,來找下一個 maximum,以此類推,直到搜尋到 250 個 maximum,這些 maximum 即為我們最後保留的 feature point 。

NextMax.m : 此程式用在 ANMS 中來協助每一次的 maximum 搜尋,併記錄下當前縮小半徑。

Compare : 以下為有/無 apply Adaptive non-maximal suppression 的結果，可以看出 apply Adaptive non-maximal suppression 後 feature point 的分佈在空間上較均勻。



3. Multi-Scale Oriented Patches

a. MSOP descriptor vector

eachfeature.m :

根據 feature 所計算出來的角度，將它所在的 level 圖片以 feature 的角度做旋轉，旋轉之後就取出以 feature 為中心 40×40 的 patch 在畫分成 8×8 的 grid。取出每個 grid 的中間點為 descriptor 其中的一個值。紀錄的順序是從左到右，從上到下，最後是個 64 維的 descriptor。

MSOP_descriptor.m

將每個 feature 套入 eachfeature.m 計算出 descriptor，再將整體 features 存成一個大矩陣

b. Feature matching

feature_coordinate_1.m:

將找到 feature 從 reauirectangle 轉到圓柱座標。

matching.m : 將兩張影像的 descriptor 用 cosine similarity 做 matching，每個 descriptor 為 64×1 為的向量。

4. Cylindrical panorama

equirec2cylind.m:

先以 thi 和 h 為座標基底建立出 warping 後的矩陣大小，接著利用 backward 的方式，推算出新座標與舊座標的位置對應，再將對應的點

的 pixel 值存入新座標中。

5. RANSAC

RANSAC.m:

將兩張圖片配對好的 features 的 x, y 座標作為 input。每次 iteration 都 random 選出一組 match feature 來計算出 translation matrix 的 $m1(x \text{ 座標的位移})$ 和 $m2(y \text{ 座標的位移})$ 。去計算出每組 $m1$ 和 $m2$ 會有多少符合 $m1$ 、 $m2$ 的 match features。經過 n 次的 iteration 後挑選出擁有最多組 match features 的 $m1$ 、 $m2$ 以及最後符合的 $m1$ 、 $m2$ 的 match features 的 x 、 y 座標

6. Image stitching

blending.m :

將兩張圖片的重疊的部分做 α mask，隨著慢慢重疊就將 α 值從 1 到 0。

stitch_img.m :

將前面的步驟進行整合，最後根據計算出來的 translate matrix 對第二張圖進行調整使其第一張座標相同，再套用 blending.m 來將兩張圖接合。

7. Result

我們一組 data 有十張，我們先兩兩 stitch 起來，也就是 1、2 圖接，2、3 接。之後再互相接。

