

# NTU CSIE 2022 Spring, Digital Visual Effects

## Project #2 Image Stitching

Group 33  
B07902078 沈韋辰  
B07902072 陳光裕

### 1. Feature Detection

#### 1.1 Cylindrical Projection

首先我們做圓柱投影，把每張圖片按照上課提到的盒子測量法所計算出來的focal length做圓柱座標轉換，即可得到投影後的圖片。



Before

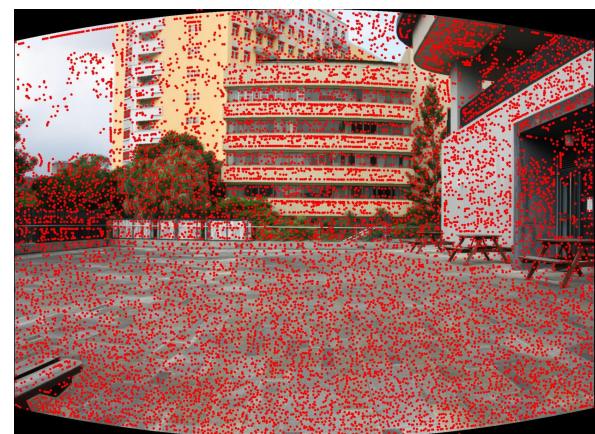


After

#### 1.2 Harris Corner Detection

首先我們使用 $5 \times 5$ 的Sobel operator去計算圖片的gradient，並用 $11 \times 11$ 的Gaussian kernel先對整張圖片做convolution。

接著就用上課提到的方式去計算R值。我們標記所有足夠大且為區域最大值的R值，作為我們的corner



### 1.3 Adaptive Non-Maximum Suppression

由於特徵點的數量過多，因此我們只需要取R值最大的前幾名就好了。右圖為n=300時的特徵點分布。

不過這邊可以注意到，那些因為做完投影而留下的corner被誤認為特徵點了。



因此我們限制了那些，和投影後產生的陰影附近相接的點，讓特徵點能更集中在圖片上。

不過這邊的另一個問題是，特徵點過於集中在一些特定的位置，像是照片右方的桌椅上。



因此我們使用上課提到的  
Adaptive Non-Maximum  
Suppression ( $r=36$ , here)。

對於任意特徵點，半徑 $r$ 之內的  
local maxima 都忽略不計。

半徑會隨著detect到的特徵點  
逐漸減少，我們組是將找完全  
部的特徵點作為 $r$ 的半衰期，隨  
著找到的點數來減少半徑



## 2. Feature Description

## 2.1 Major orientation

這次作業中，我們是使用Oriented Patch作為Feature Description的演算法。因此我們會需要得到每一個特徵點的major orientation。計算方式跟SIFT內的方式雷同，對於特徵點周圍各點的gradient去投票，gradient的角度作為投票對象（共把360度分成36個bin），而量值做為投票的權重，且權重會再因為與特徵點的距離遠近，套用一個Gaussian kernel。

我們會去找到最多票數的bin作為peak，若有其他的bin內的票數達到peak的80%，我們就會給予該點多個major orientation，實做上是將該特徵點在同個位置再複製一個出來。



## 2.2 Oriented patch

接著，我們相對於每個特徵點的major orientation去裁切一個 $40 \times 40$ 像素的patch，將它先做Gaussian blur，再resize成 $8 \times 8$ 。最後我們對這個 $8 \times 8$  patch做normalize，然後拉成一個64為地向量，作為該特徵點的description vector。



# 3. Feature Matching

## 3.1 Second Closest Neighbor

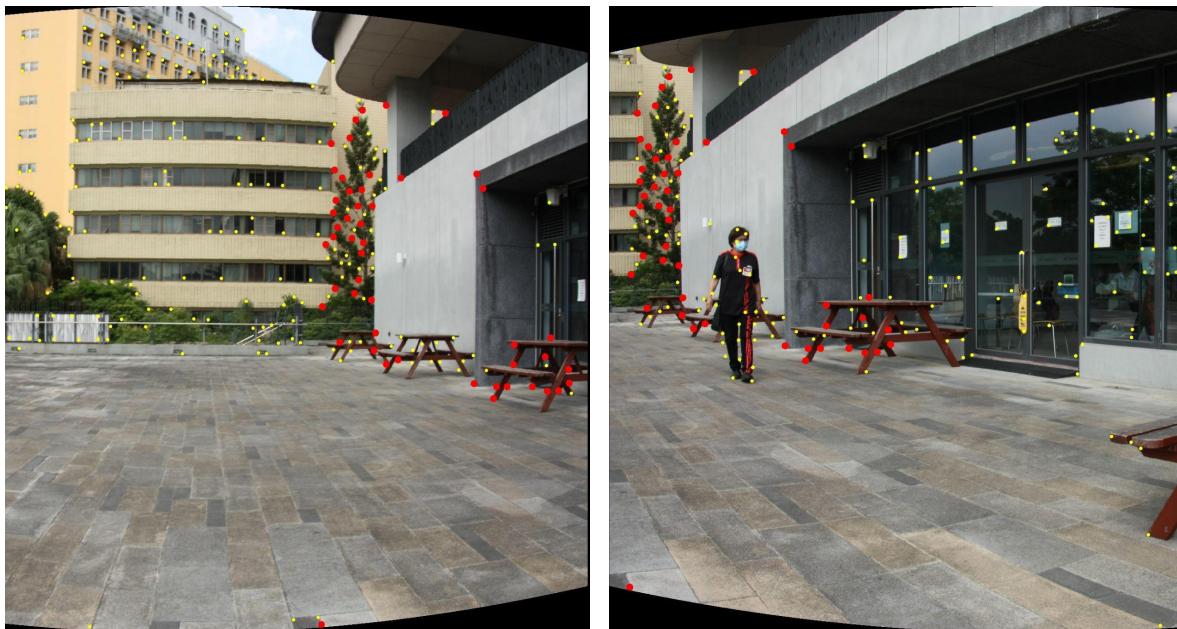
在做Matching時，我們使用second closest neighbor法來保證對應到的特徵點對是足夠好的，我們對兩張圖內的特徵點對去計算Sum of Square Differences(SSD)，SSD越小，代表兩個特徵點很有可能是對應點。但純粹使用SSD也可能發生，雖然真實對應點並不存在在另一張圖中，但還是產生了錯誤的match。

所以我們要去計算SSD最小以及次小的特徵點，因為每對特徵點相對其他對特徵點都要有獨特性，因此若最小與次小的特徵點，SSD比例過小，代表這對match太過相似，因此不會是好的match。我們經由反覆實驗決定出(最小的SSD/次小的SSD)  $< 0.6$  為較好的比例。經由以上的計算，我們可以得到兩張圖片中高機率是對應特徵點的pairs了。此外

，如果原先的圖片並沒有排序過，這邊會match的數量去比對每一張圖的下一張要接著誰。

### 3.2 RANSAC

經由Second Closest Neighbor雖然已經排除掉大部分的mismatch，但還是有一些些的mismatch的特徵點沒有被發現。我們實作了RANSAC演算法，這個演算法主要是用一個 $3 \times 3$ 矩陣作為模型(我們使用affine transform，共六個未知數)。每次從match pairs中隨機挑三組去計算候選模型，並計算inliers的數量。我們定義inliers為轉換後的點與實際配對的點的距離差的平方要小於某個閾值 (1000 in our case)，然後將以上的計算重複



進行k次 (700 in our case)，選出有最多inliers的那個做為我們要的模型。  
上圖的黃點為所有的特徵點，紅點是和別張圖match到的點。可以看到左圖的右半部和右圖的左半部可以看出大致上是match的

## 4. Image Stitching

### 4.1 Warping

當我們有了我們現在有由左至右，逐張圖片相對前一張的transform model。由於我們採用的是affine transform，因此可以很親易地計算出每一張圖相對第一張的transfrom，就是前一張相對第一張的transfrom乘上當前這張相對前一張的transform。

接著如下圖，我們將轉換後的圖片一張張貼上畫布。

### 4.2 Blending



由於圖片和圖片之間重疊處的亮度會疊加，因此我們需要做blending。我們會去計算圖片間重疊的範圍做linear的blending，結果如下：



#### 4.3 Drifting

由於圖片對接可能受到環境傾斜與鏡頭穩定度等因素，導致圖片有drifting，因此我們會去計算能將panorama的四個頂點warp成一個矩形的prospective transform，套用在圖片上



#### 4.4 Cropping

這邊我們選擇直接裁去黑色的部分，若切面不整齊，則去計算該行或該列的圖片部分比例，只保留該行圖片比例超過95%的部分。



#### 4.5 Smoothing

由於圖片疊合可能會造成一些噪點與疊合痕跡，因此最後我們還會再對panorama做一個Gaussian blur



Before

After

## 5. Results



## 6. Reference

- Course lectures and slides
- <https://ir.nctu.edu.tw/bitstream/11536/38307/1/259801.pdf>
- <https://www.csie.ntu.edu.tw/~cyy/courses/vfx/papers/Brown2003RP.pdf>
- <https://tigercosmos.xyz/post/2020/05/cv/image-stitching/>
- <https://ssarcandy.tw/2017/05/26/panorama-image-stitching/>
- <https://stackoverflow.com/questions/16702966/rotate-image-and-crop-out-black-borders>